HOL/Import

HOL-Light SVN revision 152 (2012-11-19)
Flyspeck SVN revision 3006 (2012-12-02)
(with types)

December 19, 2012

thm T_DEF:

$True = ((\lambda p::bool.\ p) = (\lambda p::bool.\ p))$

thm Tactics_jordan.unify_exists_tac_example:

$True = True$

thm TRUTH:

$True$

thm AND_DEF:

$op \wedge = (\lambda(p::bool)\ q::bool.\ (\lambda f::bool \Rightarrow bool \Rightarrow bool.\ f\ p\ q) = (\lambda f::bool \Rightarrow bool \Rightarrow bool.\ f\ True\ True))$

thm IMP_DEF:

$op \longrightarrow = (\lambda(p::bool)\ q::bool.\ (p \wedge q) = p)$

thm FORALL_DEF:

$All = (\lambda P::?'a::type \Rightarrow bool.\ P = (\lambda x::?'a::type.\ True))$

thm EXISTS_DEF:

$Ex = (\lambda P::?'a::type \Rightarrow bool.\ \forall q::bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow q) \longrightarrow q)$

thm OR_DEF:

$op \vee = (\lambda(p::bool)\ q::bool.\ \forall r::bool.\ (p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow r)$

thm F_DEF:

$False = (\forall p::bool.\ p)$

thm NOT_DEF:

$Not = (\lambda p::bool.\ p \longrightarrow False)$

thm EXISTS_UNIQUE_DEF:

$Ex1 = (\lambda P::?'a::type \Rightarrow bool.\ Ex\ P \wedge (\forall (x::?'a::type)\ y::?'a::type.\ P\ x \wedge P\ y \longrightarrow x = y))$

thm IMP_IMP:

$((?p::bool) \longrightarrow (?q::bool) \longrightarrow (?r::bool)) = (?p \wedge ?q \longrightarrow ?r)$

thm EQ_REFL:

$\forall x::?'a::type.\ x = x$

thm REFL_CLAUSE:

$\forall x::?'a::type.\ (x = x) = True$

thm EQ_SYM:

$\forall (x::?'a::type)\ y::?'a::type.\ x = y \longrightarrow y = x$

thm EQ_SYM_EQ:

$\forall (x::?'a::type)\ y::?'a::type.\ (x = y) = (y = x)$

thm EQ_TRANS:

$\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ x = y \wedge y = z \longrightarrow x = z$

thm BETA_THM:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ y::?'b::type.\ f\ y = f\ y$

thm ABS_SIMP:

$\forall (t1::?'b::type)\ t2::?'a::type.\ t1 = t1$

thm CONJ_ASSOC:

$\forall (t1::bool)\ (t2::bool)\ t3::bool.\ (t1 \wedge t2 \wedge t3) = ((t1 \wedge t2) \wedge t3)$

thm CONJ_SYM:

$\forall (t1::bool)\ t2::bool.\ (t1 \wedge t2) = (t2 \wedge t1)$

thm CONJ_ACI_conjunct0:

$((?p::bool) \wedge (?q::bool)) = (?q \wedge ?p)$

thm CONJ_ACI_conjunct1:

$(((?p::bool) \wedge (?q::bool)) \wedge (?r::bool)) = (?p \wedge ?q \wedge ?r)$

thm CONJ_ACI_conjunct2:

$((?p::bool) \wedge (?q::bool) \wedge (?r::bool)) = (?q \wedge ?p \wedge ?r)$

thm CONJ_ACI_conjunct3:

$((?p::bool) \wedge ?p) = ?p$

thm CONJ_ACI_conjunct4:

$((?p::bool) \wedge ?p \wedge (?q::bool)) = (?p \wedge ?q)$

thm CONJ_ACI:

$((?p::bool) \wedge (?q::bool)) = (?q \wedge ?p) \wedge ((?p \wedge ?q) \wedge (?r::bool)) = (?p \wedge ?q \wedge ?r) \wedge (?p \wedge ?q \wedge ?r) = (?q \wedge ?p \wedge ?r) \wedge (?p \wedge ?p) = ?p \wedge (?p \wedge ?p \wedge ?q) = (?p \wedge ?q)$

thm DISJ_ASSOC:

$\forall (t1::bool)\ (t2::bool)\ t3::bool.\ (t1 \vee t2 \vee t3) = ((t1 \vee t2) \vee t3)$

thm DISJ_SYM:

$\forall (t1::bool)\ t2::bool.\ (t1 \vee t2) = (t2 \vee t1)$

thm DISJ_ACI_conjunct0:

$((?p::bool) \vee (?q::bool)) = (?q \vee ?p)$

thm DISJ_ACI_conjunct1:

$(((?p::bool) \vee (?q::bool)) \vee (?r::bool)) = (?p \vee ?q \vee ?r)$

thm DISJ_ACI_conjunct2:

$((?p::bool) \vee (?q::bool) \vee (?r::bool)) = (?q \vee ?p \vee ?r)$

thm DISJ_ACI_conjunct3:

$((?p::bool) \vee ?p) = ?p$

thm DISJ_ACI_conjunct4:

$((?p::bool) \vee ?p \vee (?q::bool)) = (?p \vee ?q)$

thm DISJ_ACI:

$((?p::bool) \vee (?q::bool)) = (?q \vee ?p) \wedge ((?p \vee ?q) \vee (?r::bool)) = (?p \vee ?q \vee ?r) \wedge (?p \vee ?q \vee ?r) = (?q \vee ?p \vee ?r) \wedge (?p \vee ?p) = ?p \wedge (?p \vee ?p \vee ?q) = (?p \vee ?q)$

thm IMP_CONJ:

$((?p::bool) \wedge (?q::bool) \longrightarrow (?r::bool)) = (?p \longrightarrow ?q \longrightarrow ?r)$

thm IMP_CONJ_ALT:

$((?p::bool) \wedge (?q::bool) \longrightarrow (?r::bool)) = (?q \longrightarrow ?p \longrightarrow ?r)$

thm LEFT_OR_DISTRIB:

$\forall (p::bool)\ (q::bool)\ r::bool.\ (p \wedge (q \vee r)) = (p \wedge q \vee p \wedge r)$

thm RIGHT_OR_DISTRIB:

$\forall (p::bool)\ (q::bool)\ r::bool.\ ((p \vee q) \wedge r) = (p \wedge r \vee q \wedge r)$

thm FORALL_SIMP:

$\forall\, t{::}bool.\ (\forall\, x{::}?'a{::}type.\ t) = t$

thm EXISTS_SIMP:

$\forall\, t{::}bool.\ (\exists\, x{::}?'a{::}type.\ t) = t$

thm EQ_IMP:

$(?a{::}bool) = (?b{::}bool) \longrightarrow ?a \longrightarrow ?b$

thm EQ_CLAUSES:

$\forall\, t{::}bool.\ (True = t) = t \land (t = True) = t \land (False = t) = (\neg\ t) \land (t = False) = (\neg\ t)$

thm NOT_CLAUSES_WEAK_conjunct1:

$(\neg\ False) = True$

thm NOT_CLAUSES_WEAK_conjunct0:

$(\neg\ True) = False$

thm NOT_CLAUSES_WEAK:

$(\neg\ True) = False \land (\neg\ False) = True$

thm AND_CLAUSES:

$\forall\, t{::}bool.\ (True \land t) = t \land (t \land True) = t \land (False \land t) = False \land (t \land False) = False \land (t \land t) = t$

thm OR_CLAUSES:

$\forall\, t{::}bool.\ (True \lor t) = True \land (t \lor True) = True \land (False \lor t) = t \land (t \lor False) = t \land (t \lor t) = t$

thm IMP_CLAUSES:

$\forall\, t{::}bool.\ (True \longrightarrow t) = t \land (t \longrightarrow True) = True \land (False \longrightarrow t) = True \land (t \longrightarrow t) = True \land (t \longrightarrow False) = (\neg\ t)$

thm EXISTS_UNIQUE_THM:

$\forall\, P{::}?'a{::}type \Rightarrow bool.\ (\exists!x{::}?'a{::}type.\ P\ x) = ((\exists\, x{::}?'a{::}type.\ P\ x) \land (\forall\, (x{::}?'a{::}type)\ x'{::}?'a{::}type.\ P\ x \land P\ x' \longrightarrow x = x'))$

thm EXISTS_REFL:

$\forall\, a{::}?'a{::}type.\ \exists\, x{::}?'a{::}type.\ x = a$

thm EXISTS_UNIQUE_REFL:

$\forall\, a{::}?'a{::}type.\ \exists!x{::}?'a{::}type.\ x = a$

thm UNWIND_THM1:

$\forall\, (P{::}?'a{::}type \Rightarrow bool)\ a{::}?'a{::}type.\ (\exists\, x{::}?'a{::}type.\ a = x \land P\ x) = P\ a$

thm UNWIND_THM2:

$\forall (P::?'a::type \Rightarrow bool)\ a::?'a::type.\ (\exists x::?'a::type.\ x = a \wedge P\ x) = P\ a$

thm FORALL_UNWIND_THM2:

$\forall (P::?'a::type \Rightarrow bool)\ a::?'a::type.\ (\forall x::?'a::type.\ x = a \longrightarrow P\ x) = P\ a$

thm FORALL_UNWIND_THM1:

$\forall (P::?'a::type \Rightarrow bool)\ a::?'a::type.\ (\forall x::?'a::type.\ a = x \longrightarrow P\ x) = P\ a$

thm SWAP_FORALL_THM:

$\forall P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'a::type.\ P\ x\ y) = (\forall (y::?'a::type)\ x::?'b::type.\ P\ x\ y)$

thm SWAP_EXISTS_THM:

$\forall P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\exists (x::?'b::type)\ y::?'a::type.\ P\ x\ y) = (\exists (y::?'a::type)\ x::?'b::type.\ P\ x\ y)$

thm FORALL_AND_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P\ x \wedge Q\ x) = ((\forall x::?'a::type.\ P\ x) \wedge (\forall x::?'a::type.\ Q\ x))$

thm AND_FORALL_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ ((\forall x::?'a::type.\ P\ x) \wedge (\forall x::?'a::type.\ Q\ x)) = (\forall x::?'a::type.\ P\ x \wedge Q\ x)$

thm LEFT_AND_FORALL_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\forall x::?'a::type.\ P\ x) \wedge Q) = (\forall x::?'a::type.\ P\ x \wedge Q)$

thm RIGHT_AND_FORALL_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \wedge (\forall x::?'a::type.\ Q\ x)) = (\forall x::?'a::type.\ P \wedge Q\ x)$

thm EXISTS_OR_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ P\ x \vee Q\ x) = ((\exists x::?'a::type.\ P\ x) \vee (\exists x::?'a::type.\ Q\ x))$

thm OR_EXISTS_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ ((\exists x::?'a::type.\ P\ x) \vee (\exists x::?'a::type.\ Q\ x)) = (\exists x::?'a::type.\ P\ x \vee Q\ x)$

thm LEFT_OR_EXISTS_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\exists x::?'a::type.\ P\ x) \vee Q) = (\exists x::?'a::type.\ P\ x \vee Q)$

thm RIGHT_OR_EXISTS_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \vee (\exists x::?'a::type.\ Q\ x)) = (\exists x::?'a::type.\ P \vee Q\ x)$

thm LEFT_EXISTS_AND_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ (\exists x::?'a::type.\ P\ x \wedge Q) = ((\exists x::?'a::type.\ P\ x) \wedge Q)$

thm RIGHT_EXISTS_AND_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ P \wedge Q\ x) = (P \wedge (\exists x::?'a::type.\ Q\ x))$

thm TRIV_EXISTS_AND_THM:

$\forall (P::bool)\ Q::bool.\ (\exists x::?'a::type.\ P \wedge Q) = ((\exists x::?'a::type.\ P) \wedge (\exists x::?'a::type.\ Q))$

thm LEFT_AND_EXISTS_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\exists x::?'a::type.\ P\ x) \wedge Q) = (\exists x::?'a::type.\ P\ x \wedge Q)$

thm RIGHT_AND_EXISTS_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \wedge (\exists x::?'a::type.\ Q\ x)) = (\exists x::?'a::type.\ P \wedge Q\ x)$

thm TRIV_AND_EXISTS_THM:

$\forall (P::bool)\ Q::bool.\ ((\exists x::?'a::type.\ P) \wedge (\exists x::?'a::type.\ Q)) = (\exists x::?'a::type.\ P \wedge Q)$

thm TRIV_FORALL_OR_THM:

$\forall (P::bool)\ Q::bool.\ (\forall x::?'a::type.\ P \vee Q) = ((\forall x::?'a::type.\ P) \vee (\forall x::?'a::type.\ Q))$

thm TRIV_OR_FORALL_THM:

$\forall (P::bool)\ Q::bool.\ ((\forall x::?'a::type.\ P) \vee (\forall x::?'a::type.\ Q)) = (\forall x::?'a::type.\ P \vee Q)$

thm RIGHT_IMP_FORALL_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \longrightarrow (\forall x::?'a::type.\ Q\ x)) = (\forall x::?'a::type.\ P \longrightarrow Q\ x)$

thm RIGHT_FORALL_IMP_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P \longrightarrow Q\ x) = (P \longrightarrow (\forall x::?'a::type.\ Q\ x))$

thm LEFT_IMP_EXISTS_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\exists x::?'a::type.\ P\ x) \longrightarrow Q) = (\forall x::?'a::type.\ P\ x \longrightarrow Q)$

thm LEFT_FORALL_IMP_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow Q) = ((\exists x::?'a::type.\ P\ x) \longrightarrow Q)$

thm TRIV_FORALL_IMP_THM:

$\forall (P::bool)\ Q::bool.\ (\forall x::?'a::type.\ P \longrightarrow Q) = ((\exists x::?'a::type.\ P) \longrightarrow (\forall x::?'a::type.\ Q))$

thm TRIV_EXISTS_IMP_THM:

$\forall (P::bool)\ Q::bool.\ (\exists x::?'a::type.\ P \longrightarrow Q) = ((\forall x::?'a::type.\ P) \longrightarrow (\exists x::?'a::type.\ Q))$

thm EXISTS_UNIQUE_ALT:

$\forall P::?'a::type \Rightarrow bool.\ (\exists !x::?'a::type.\ P\ x) = (\exists x::?'a::type.\ \forall y::?'a::type.\ P\ y = (x = y))$

thm Geomdetail.EQ_EXPAND:

$((?a::bool) = (?b::bool)) = ((?a \longrightarrow ?b) \wedge (?b \longrightarrow ?a))$

thm EXISTS_UNIQUE:

$\forall P::?'a::type \Rightarrow bool.\ (\exists !x::?'a::type.\ P\ x) = (\exists x::?'a::type.\ P\ x \wedge (\forall y::?'a::type.\ P\ y \longrightarrow y = x))$

thm MONO_AND:

$((?A::bool) \longrightarrow (?B::bool)) \wedge ((?C::bool) \longrightarrow (?D::bool)) \longrightarrow ?A \wedge ?C \longrightarrow ?B \wedge ?D$

thm MONO_OR:

$((?A::bool) \longrightarrow (?B::bool)) \wedge ((?C::bool) \longrightarrow (?D::bool)) \longrightarrow ?A \vee ?C \longrightarrow ?B \vee ?D$

thm MONO_IMP:

$((?B::bool) \longrightarrow (?A::bool)) \wedge ((?C::bool) \longrightarrow (?D::bool)) \longrightarrow (?A \longrightarrow ?C) \longrightarrow ?B \longrightarrow ?D$

thm MONO_NOT:

$((?B::bool) \longrightarrow (?A::bool)) \longrightarrow \neg\ ?A \longrightarrow \neg\ ?B$

thm MONO_FORALL:

$(\forall x::?'a::type.\ (?P::?'a::type \Rightarrow bool)\ x \longrightarrow (?Q::?'a::type \Rightarrow bool)\ x) \longrightarrow (\forall x::?'a::type.\ ?P\ x) \longrightarrow (\forall x::?'a::type.\ ?Q\ x)$

thm MONO_EXISTS:

$(\forall x::?'a::type.\ (?P::?'a::type \Rightarrow bool)\ x \longrightarrow (?Q::?'a::type \Rightarrow bool)\ x) \longrightarrow (\exists x::?'a::type.\ ?P\ x) \longrightarrow (\exists x::?'a::type.\ ?Q\ x)$

thm ETA_AX:

$\forall t::?'b::type \Rightarrow ?'a::type.\ t = t$

thm EQ_EXT:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ g::?'b::type \Rightarrow ?'a::type.\ (\forall x::?'b::type.\ f\,x = g\ x) \longrightarrow f = g$

thm FUN_EQ_THM:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ g::?'b::type \Rightarrow ?'a::type.\ (f = g) = (\forall x::?'b::type.\ f\,x = g\ x)$

thm SELECT_AX:

$\forall (P::?'a::type \Rightarrow bool)\ x::?'a::type.\ P\ x \longrightarrow P\ (Eps\ P)$

thm EXISTS_THM:

$Ex = (\lambda P::?'a::type \Rightarrow bool.\ P\ (Eps\ P))$

thm SELECT_REFL:

$\forall x::?'a::type.\ (SOME\ y::?'a::type.\ y = x) = x$

thm SELECT_UNIQUE:

$\forall (P::?'a::type \Rightarrow bool)\ x::?'a::type.\ (\forall y::?'a::type.\ P\ y = (y = x)) \longrightarrow Eps\ P = x$

thm EXCLUDED_MIDDLE:

$\forall t::bool.\ t \vee \neg\ t$

thm BOOL_CASES_AX:

$\forall t::bool.\ t = True \vee t = False$

thm DE_MORGAN_THM:

$\forall (t1::bool)\ t2::bool.\ (\neg\ (t1 \wedge t2)) = (\neg\ t1 \vee \neg\ t2) \wedge (\neg\ (t1 \vee t2)) = (\neg\ t1 \wedge \neg\ t2)$

thm NOT_CLAUSES_conjunct0:

$\forall t::bool.\ (\neg\ \neg\ t) = t$

thm NOT_CLAUSES:

$(\forall t::bool.\ (\neg\ \neg\ t) = t) \wedge (\neg\ True) = False \wedge (\neg\ False) = True$

thm NOT_IMP:

$\forall (t1::bool)\ t2::bool.\ (\neg\ (t1 \longrightarrow t2)) = (t1 \wedge \neg\ t2)$

thm CONTRAPOS_THM:

$\forall (t1::bool)\ t2::bool.\ (\neg\ t1 \longrightarrow \neg\ t2) = (t2 \longrightarrow t1)$

thm NOT_EXISTS_THM:

$\forall P::?'a::type \Rightarrow bool.\ (\neg\ (\exists x::?'a::type.\ P\ x)) = (\forall x::?'a::type.\ \neg\ P\ x)$

thm EXISTS_NOT_THM:

$\forall P::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ \neg\ P\ x) = (\neg\ (\forall x::?'a::type.\ P\ x))$

thm NOT_FORALL_THM:

$\forall P::?'a::type \Rightarrow bool.\ (\neg\ (\forall x::?'a::type.\ P\ x)) = (\exists x::?'a::type.\ \neg\ P\ x)$

thm FORALL_NOT_THM:

$\forall P::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ \neg\ P\ x) = (\neg\ (\exists x::?'a::type.\ P\ x))$

thm FORALL_BOOL_THM:

$(\forall b::bool.\ (?P::bool \Rightarrow bool)\ b) = (?P\ True \wedge ?P\ False)$

thm EXISTS_BOOL_THM:

$(\exists b::bool.\ (?P::bool \Rightarrow bool)\ b) = (?P\ True \vee ?P\ False)$

thm LEFT_FORALL_OR_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ (\forall x::?'a::type.\ P\ x \vee Q) = ((\forall x::?'a::type.\ P\ x) \vee Q)$

thm RIGHT_FORALL_OR_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P \vee Q\ x) = (P \vee (\forall x::?'a::type.\ Q\ x))$

thm LEFT_OR_FORALL_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\forall x::?'a::type.\ P\ x) \vee Q) = (\forall x::?'a::type.\ P\ x \vee Q)$

thm RIGHT_OR_FORALL_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \vee (\forall x::?'a::type.\ Q\ x)) = (\forall x::?'a::type.\ P \vee Q\ x)$

thm LEFT_IMP_FORALL_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ ((\forall x::?'a::type.\ P\ x) \longrightarrow Q) = (\exists x::?'a::type.\ P\ x \longrightarrow Q)$

thm LEFT_EXISTS_IMP_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::bool.\ (\exists x::?'a::type.\ P\ x \longrightarrow Q) = ((\forall x::?'a::type.\ P\ x) \longrightarrow Q)$

thm RIGHT_IMP_EXISTS_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (P \longrightarrow (\exists x::?'a::type.\ Q\ x)) = (\exists x::?'a::type.\ P \longrightarrow Q\ x)$

thm RIGHT_EXISTS_IMP_THM:

$\forall (P::bool)\ Q::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ P \longrightarrow Q\ x) = (P \longrightarrow (\exists x::?'a::type.\ Q\ x))$

thm COND_DEF:

$If = (\lambda(t::bool)\ (t1::?'a::type)\ t2::?'a::type.\ SOME\ x::?'a::type.\ (t = True \longrightarrow x = t1) \wedge (t = False \longrightarrow x = t2))$

thm COND_CLAUSES:

$\forall (t1::?'a::type)\ t2::?'a::type.\ (if\ True\ then\ t1\ else\ t2) = t1 \wedge (if\ False\ then\ t1\ else\ t2) = t2$

thm COND_EXPAND:

$\forall (b::bool)\ (t1::bool)\ t2::bool.\ (if\ b\ then\ t1\ else\ t2) = ((\neg\ b \vee t1) \wedge (b \vee t2))$

thm COND_ID:

$\forall (b::bool)\ t::?'a::type.\ (if\ b\ then\ t\ else\ t) = t$

thm COND_RAND:

$\forall (b::bool)\ (f::?'b::type \Rightarrow ?'a::type)\ (x::?'b::type)\ y::?'b::type.\ f\ (if\ b\ then\ x\ else\ y) = (if\ b\ then\ f\ x\ else\ f\ y)$

thm COND_RATOR:

$\forall (b::bool)\ (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ (if\ b\ then\ f\ else\ g)\ x = (if\ b\ then\ f\ x\ else\ g\ x)$

thm COND_ABS:

$\forall (b::bool)\ (f::?'b::type \Rightarrow ?'a::type)\ g::?'b::type \Rightarrow ?'a::type.\ (\lambda x::?'b::type.\ if\ b\ then\ f\ x\ else\ g\ x) = (if\ b\ then\ f\ else\ g)$

thm MONO_COND:

$((?A::bool) \longrightarrow (?B::bool)) \wedge ((?C::bool) \longrightarrow (?D::bool)) \longrightarrow (if\ ?b::bool\ then\ ?A\ else\ ?C) \longrightarrow (if\ ?b\ then\ ?B\ else\ ?D)$

thm COND_ELIM_THM:

$(?P::?'a::type \Rightarrow bool)\ (if\ ?c::bool\ then\ ?x::?'a::type\ else\ (?y::?'a::type)) = ((?c \longrightarrow ?P\ ?x) \wedge (\neg\ ?c \longrightarrow ?P\ ?y))$

thm SKOLEM_THM:

$\forall P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'b::type.\ \exists y::?'a::type.\ P\ x\ y) = (\exists y::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ P\ x\ (y\ x))$

thm UNIQUE_SKOLEM_ALT:

$\forall P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'b::type.\ \exists!y::?'a::type.\ P\ x\ y) = (\exists f::?'b::type \Rightarrow ?'a::type.\ \forall (x::?'b::type)\ y::?'a::type.\ P\ x\ y = (f\ x = y))$

thm UNIQUE_SKOLEM_THM:

$\forall P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'b::type.\ \exists!y::?'a::type.\ P\ x\ y) = (\exists!f::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ P\ x\ (f\ x))$

thm bool_INDUCT:

$\forall P::bool \Rightarrow bool.\ P\ False \wedge P\ True \longrightarrow (\forall x::bool.\ P\ x)$

thm bool_RECURSION:

$\forall (a::?'a::type)\ b::?'a::type.\ \exists f::bool \Rightarrow ?'a::type.\ f\ False = a \wedge f\ True = b$

thm DEF_o:

$op \circ = (\lambda(f::?'c::type \Rightarrow ?'b::type)\ (g::?'a::type \Rightarrow ?'c::type)\ x::?'a::type.\ f\ (g\ x))$

thm o_DEF:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ g::?'a::type \Rightarrow ?'c::type.\ f \circ g = (\lambda x::?'a::type.\ f\ (g\ x))$

thm I_DEF:

$id = (\lambda x::?'a::type.\ x)$

thm o_THM:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'a::type \Rightarrow ?'c::type)\ x::?'a::type.\ (f \circ g)\ x = f\ (g\ x)$

thm o_ASSOC:

$\forall (f::?'d::type \Rightarrow ?'c::type)\ (g::?'b::type \Rightarrow ?'d::type)\ h::?'a::type \Rightarrow ?'b::type.\ f \circ (g \circ h) = f \circ g \circ h$

thm I_THM:

$\forall x::?'a::type.\ id\ x = x$

thm I_O_ID:

$\forall f::?'b::type \Rightarrow ?'a::type.\ id \circ f = f \wedge f \circ id = f$

thm EXISTS_ONE_REP:

$\exists b::bool.\ b$

thm TYDEF_1:

$Abs\_unit\ (Rep\_unit\ (?a::unit)) = ?a \wedge (?r::bool) = (Rep\_unit\ (Abs\_unit\ ?r) = ?r)$

thm one_tydef_conjunct1:

$\forall r::bool.\ r = (Rep\_unit\ (Abs\_unit\ r) = r)$

thm one_tydef_conjunct0:

$\forall a::unit.\ Abs\_unit\ (Rep\_unit\ a) = a$

thm one_tydef:

$(\forall\, a{::}unit.\ Abs\_unit\ (Rep\_unit\ a) = a) \wedge (\forall\, r{::}bool.\ r = (Rep\_unit\ (Abs\_unit\ r) = r))$

thm one:

$\forall\, v{::}unit.\ v = ()$

thm one_axiom:

$\forall\, (f{::}?'a{::}type \Rightarrow unit)\ g{::}?'a{::}type \Rightarrow unit.\ f = g$

thm one_INDUCT:

$\forall\, P{::}unit \Rightarrow bool.\ P\ () \longrightarrow (\forall\, x{::}unit.\ P\ x)$

thm one_RECURSION:

$\forall\, e{::}?'a{::}type.\ \exists\, fn{::}unit \Rightarrow ?'a{::}type.\ fn\ () = e$

thm one_Axiom:

$\forall\, e{::}?'a{::}type.\ \exists!fn{::}unit \Rightarrow ?'a{::}type.\ fn\ () = e$

thm DEF_LET:

$LET = (\lambda f{::}?'b{::}type \Rightarrow ?'a{::}type.\ f)$

thm LET_DEF:

$\forall\, (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ x{::}?'b{::}type.\ LET\ f\ x = f\ x$

thm DEF_LET_END:

$LET\_END = (\lambda t{::}?'a{::}type.\ t)$

thm LET_END_DEF:

$\forall\, t{::}?'a{::}type.\ LET\_END\ t = t$

thm DEF_GABS:

$GABS = Eps$

thm GABS_DEF:

$\forall\, P{::}?'a{::}type \Rightarrow bool.\ GABS\ P = Eps\ P$

thm DEF_GEQ:

$GEQ = op =$

thm GEQ_DEF:

$\forall\, (a{::}?'a{::}type)\ b{::}?'a{::}type.\ GEQ\ a\ b = (a = b)$

thm _SEQPATTERN:

$\_SEQPATTERN = (\lambda (r{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool)\ (s{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool)\ x{::}?'b{::}type.\ if\ \exists\, y{::}?'a{::}type.\ r\ x\ y\ then\ r\ x\ else\ s\ x)$

thm _UNGUARDED_PATTERN:

$\_UNGUARDED\_PATTERN = op \wedge$

thm _GUARDED_PATTERN:

$\_GUARDED\_PATTERN = (\lambda(p::bool)\ (g::bool)\ r::bool.\ p \wedge g \wedge r)$

thm _MATCH:

$\_MATCH = (\lambda(e::?'b::type)\ r::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ if\ Ex1\ (r\ e)\ then\ Eps\ (r\ e)\ else\ SOME\ z::?'a::type.\ False)$

thm _FUNCTION:

$\_FUNCTION = (\lambda(r::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'b::type.\ if\ Ex1\ (r\ x)\ then\ Eps\ (r\ x)\ else\ SOME\ z::?'a::type.\ False)$

thm DEF_mk_pair:

$Pair\_Rep = (\lambda(x::?'b::type)\ (y::?'a::type)\ (a::?'b::type)\ b::?'a::type.\ a = x \wedge b = y)$

thm mk_pair_def:

$\forall (x::?'b::type)\ y::?'a::type.\ Pair\_Rep\ x\ y = (\lambda(a::?'b::type)\ b::?'a::type.\ a = x \wedge b = y)$

thm PAIR_EXISTS_THM:

$\exists (x::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (a::?'b::type)\ b::?'a::type.\ x = Pair\_Rep\ a\ b$

thm TYDEF_prod:

$Abs\_prod\ (Rep\_prod\ (?a::?'b::type \times ?'a::type)) = ?a \wedge (\exists (a::?'b::type)\ b::?'a::type.\ (?r::?'b::type \Rightarrow ?'a::type \Rightarrow bool) = Pair\_Rep\ a\ b) = (Rep\_prod\ (Abs\_prod\ ?r) = ?r)$

thm prod_tybij_conjunct1:

$\forall r::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\exists (a::?'b::type)\ b::?'a::type.\ r = Pair\_Rep\ a\ b) = (Rep\_prod\ (Abs\_prod\ r) = r)$

thm prod_tybij_conjunct0:

$\forall a::?'b::type \times ?'a::type.\ Abs\_prod\ (Rep\_prod\ a) = a$

thm prod_tybij:

$(\forall a::?'b::type \times ?'a::type.\ Abs\_prod\ (Rep\_prod\ a) = a) \wedge (\forall r::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\exists (a::?'b::type)\ b::?'a::type.\ r = Pair\_Rep\ a\ b) = (Rep\_prod\ (Abs\_prod\ r) = r))$

thm REP_ABS_PAIR:

$\forall (x::?'b::type)\ y::?'a::type.\ Rep\_prod\ (Abs\_prod\ (Pair\_Rep\ x\ y)) = Pair\_Rep\ x\ y$

thm DEF_,:

$Pair = (\lambda(x::?'b::type)\ y::?'a::type.\ Abs\_prod\ (Pair\_Rep\ x\ y))$

thm COMMA_DEF:

$\forall (x::?'b::type)\ y::?'a::type.\ (x,\ y) = Abs\_prod\ (Pair\_Rep\ x\ y)$

thm DEF_FST:

$fst = (\lambda p::?'b::type \times ?'a::type.\ SOME\ x::?'b::type.\ \exists\, y::?'a::type.\ p = (x,\ y))$

thm FST_DEF:

$\forall\, p::?'b::type \times ?'a::type.\ fst\ p = (SOME\ x::?'b::type.\ \exists\, y::?'a::type.\ p = (x,\ y))$

thm DEF_SND:

$snd = (\lambda p::?'b::type \times ?'a::type.\ SOME\ y::?'a::type.\ \exists\, x::?'b::type.\ p = (x,\ y))$

thm SND_DEF:

$\forall\, p::?'b::type \times ?'a::type.\ snd\ p = (SOME\ y::?'a::type.\ \exists\, x::?'b::type.\ p = (x,\ y))$

thm PAIR_EQ:

$\forall (x::?'b::type)\ (y::?'a::type)\ (a::?'b::type)\ b::?'a::type.\ ((x,\ y) = (a,\ b)) = (x = a \wedge y = b)$

thm PAIR_SURJECTIVE:

$\forall\, p::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ p = (x,\ y)$

thm FST:

$\forall (x::?'b::type)\ y::?'a::type.\ fst\ (x,\ y) = x$

thm SND:

$\forall (x::?'b::type)\ y::?'a::type.\ snd\ (x,\ y) = y$

thm PAIR:

$\forall\, x::?'b::type \times ?'a::type.\ (fst\ x,\ snd\ x) = x$

thm pair_INDUCT:

$\forall\, P::?'b::type \times ?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'a::type.\ P\ (x,\ y)) \longrightarrow (\forall\, p::?'b::type \times ?'a::type.\ P\ p)$

thm pair_RECURSION:

$\forall\, PAIR'::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type.\ \exists fn::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (a0::?'c::type)\ a1::?'b::type.\ fn\ (a0,\ a1) = PAIR'\ a0\ a1$

thm DEF_UNCURRY:

$UNCURRY = (\lambda(\_1082::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ \_1083::?'c::type \times ?'b::type.\ \_1082\ (fst\ \_1083)\ (snd\ \_1083))$

thm UNCURRY_DEF:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (x::?'c::type)\ y::?'b::type.\ UNCURRY\ f\ (x,\ y) = f\ x\ y$

thm DEF_PASSOC:

$PASSOC = (\lambda(\_1099::(?'d::type \times ?'c::type) \times ?'b::type \Rightarrow ?'a::type)\ \_1100::?'d::type \times ?'c::type \times ?'b::type.\ \_1099\ ((fst\ \_1100, fst\ (snd\ \_1100)), snd\ (snd\ \_1100)))$

thm PASSOC_DEF:

$\forall (f::(?'d::type \times ?'c::type) \times ?'b::type \Rightarrow ?'a::type)\ (x::?'d::type)\ (y::?'c::type)\ z::?'b::type.\ PASSOC\ f\ (x,\ y,\ z) = f\ ((x,\ y),\ z)$

thm FORALL_PAIR_THM:

$\forall P::?'b::type \times ?'a::type \Rightarrow bool.\ (\forall p::?'b::type \times ?'a::type.\ P\ p) = (\forall (p1::?'b::type)\ p2::?'a::type.\ P\ (p1,\ p2))$

thm EXISTS_PAIR_THM:

$\forall P::?'b::type \times ?'a::type \Rightarrow bool.\ (\exists p::?'b::type \times ?'a::type.\ P\ p) = (\exists (p1::?'b::type)\ p2::?'a::type.\ P\ (p1,\ p2))$

thm LAMBDA_PAIR_THM:

$\forall t::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ t = GABS\ (\lambda f::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (x::?'c::type)\ y::?'b::type.\ GEQ\ (f\ (x,\ y))\ (t\ (x,\ y)))$

thm PAIRED_ETA_THM_conjunct2:

$\forall f::?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \Rightarrow ?'a::type.\ GABS\ (\lambda f'::?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (w::?'e::type)\ (x::?'d::type)\ (y::?'c::type)\ z::?'b::type.\ GEQ\ (f'\ (w,\ x,\ y,\ z))\ (f\ (w,\ x,\ y,\ z))) = f$

thm PAIRED_ETA_THM_conjunct1:

$\forall f::?'d::type \times ?'c::type \times ?'b::type \Rightarrow ?'a::type.\ GABS\ (\lambda f'::?'d::type \times ?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (x::?'d::type)\ (y::?'c::type)\ z::?'b::type.\ GEQ\ (f'\ (x,\ y,\ z))\ (f\ (x,\ y,\ z))) = f$

thm PAIRED_ETA_THM_conjunct0:

$\forall f::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ GABS\ (\lambda f'::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (x::?'c::type)\ y::?'b::type.\ GEQ\ (f'\ (x,\ y))\ (f\ (x,\ y))) = f$

thm PAIRED_ETA_THM:

$(\forall f::?'l::type \times ?'k::type \Rightarrow ?'j::type.\ GABS\ (\lambda f'::?'l::type \times ?'k::type \Rightarrow ?'j::type.\ \forall (x::?'l::type)\ y::?'k::type.\ GEQ\ (f'\ (x,\ y))\ (f\ (x,\ y))) = f) \wedge (\forall f::?'i::type \times ?'h::type \times ?'g::type \Rightarrow ?'f::type.\ GABS\ (\lambda f'::?'i::type \times ?'h::type \times ?'g::type \Rightarrow ?'f::type.\ \forall (x::?'i::type)\ (y::?'h::type)\ z::?'g::type.$

$GEQ\ (f'\ (x,\ y,\ z))\ (f\ (x,\ y,\ z))) = f) \wedge (\forall f{::}?'e{::}type \times ?'d{::}type \times ?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ GABS\ (\lambda f'{::}?'e{::}type \times ?'d{::}type \times ?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ \forall (w{::}?'e{::}type)\ (x{::}?'d{::}type)\ (y{::}?'c{::}type)\ z{::}?'b{::}type.\ GEQ\ (f'\ (w,\ x,\ y,\ z))\ (f\ (w,\ x,\ y,\ z))) = f)$

thm FORALL_UNCURRY:

$\forall P{::}(?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\forall f{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ P\ f) = (\forall f{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ P\ (\lambda(a{::}?'c{::}type)\ b{::}?'b{::}type.\ f\ (a,\ b)))$

thm EXISTS_UNCURRY:

$\forall P{::}(?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\exists f{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ P\ f) = (\exists f{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ P\ (\lambda(a{::}?'c{::}type)\ b{::}?'b{::}type.\ f\ (a,\ b)))$

thm EXISTS_CURRY:

$\forall P{::}(?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\exists f{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ P\ f) = (\exists f{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ P\ (GABS\ (\lambda fa{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ \forall (a{::}?'c{::}type)\ b{::}?'b{::}type.\ GEQ\ (fa\ (a,\ b))\ (f\ a\ b))))$

thm FORALL_CURRY:

$\forall P{::}(?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\forall f{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ P\ f) = (\forall f{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ P\ (GABS\ (\lambda fa{::}?'c{::}type \times ?'b{::}type \Rightarrow ?'a{::}type.\ \forall (a{::}?'c{::}type)\ b{::}?'b{::}type.\ GEQ\ (fa\ (a,\ b))\ (f\ a\ b))))$

thm FORALL_PAIRED_THM:

$\forall P{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ All\ (GABS\ (\lambda f{::}?'b{::}type \times ?'a{::}type \Rightarrow bool.\ \forall (x{::}?'b{::}type)\ y{::}?'a{::}type.\ GEQ\ (f\ (x,\ y))\ (P\ x\ y))) = (\forall (x{::}?'b{::}type)\ y{::}?'a{::}type.\ P\ x\ y)$

thm EXISTS_PAIRED_THM:

$\forall P{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ Ex\ (GABS\ (\lambda f{::}?'b{::}type \times ?'a{::}type \Rightarrow bool.\ \forall (x{::}?'b{::}type)\ y{::}?'a{::}type.\ GEQ\ (f\ (x,\ y))\ (P\ x\ y))) = (\exists (x{::}?'b{::}type)\ y{::}?'a{::}type.\ P\ x\ y)$

thm FORALL_TRIPLED_THM:

$\forall P{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ All\ (GABS\ (\lambda f{::}?'c{::}type \times ?'b{::}type \times ?'a{::}type \Rightarrow bool.\ \forall (x{::}?'c{::}type)\ (y{::}?'b{::}type)\ z{::}?'a{::}type.\ GEQ\ (f\ (x,\ y,\ z))\ (P\ x\ y\ z))) = (\forall (x{::}?'c{::}type)\ (y{::}?'b{::}type)\ z{::}?'a{::}type.\ P\ x\ y\ z)$

thm EXISTS_TRIPLED_THM:

$\forall P{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ Ex\ (GABS\ (\lambda f{::}?'c{::}type \times ?'b{::}type \times ?'a{::}type \Rightarrow bool.\ \forall (x{::}?'c{::}type)\ (y{::}?'b{::}type)\ z{::}?'a{::}type.\ GEQ\ (f\ (x,\ y,\ z))\ (P\ x\ y\ z))) = (\exists (x{::}?'c{::}type)\ (y{::}?'b{::}type)\ z{::}?'a{::}type.\ P\ x\ y\ z)$

thm DEF_ONE_ONE:

$inj = (\lambda f::?'b::type \Rightarrow ?'a::type. \forall (x1::?'b::type)\ x2::?'b::type. f\,x1 = f\,x2 \longrightarrow x1 = x2)$

thm ONE_ONE:

$\forall f::?'b::type \Rightarrow ?'a::type.\ inj\ f = (\forall (x1::?'b::type)\ x2::?'b::type.\ f\ x1 = f\ x2 \longrightarrow x1 = x2)$

thm DEF_ONTO:

$surj = (\lambda f::?'b::type \Rightarrow ?'a::type.\ \forall y::?'a::type.\ \exists x::?'b::type.\ y = f\ x)$

thm ONTO:

$\forall f::?'b::type \Rightarrow ?'a::type.\ surj\ f = (\forall y::?'a::type.\ \exists x::?'b::type.\ y = f\ x)$

thm INFINITY_AX:

$\exists f::ind \Rightarrow ind.\ inj\ f \wedge \neg\ surj\ f$

thm IND_SUC_0_EXISTS:

$\exists (f::ind \Rightarrow ind)\ z::ind.\ (\forall (x1::ind)\ x2::ind.\ (f\ x1 = f\ x2) = (x1 = x2)) \wedge (\forall x::ind.\ f\ x \neq z)$

thm DEF_IND_SUC:

$IND\_SUC = (SOME\ f::ind \Rightarrow ind.\ \exists z::ind.\ (\forall (x1::ind)\ x2::ind.\ (f\,x1 = f\,x2) = (x1 = x2)) \wedge (\forall x::ind.\ f\ x \neq z))$

thm DEF_IND_0:

$IND\_0 = (SOME\ z::ind.\ (\forall (x1::ind)\ x2::ind.\ (IND\_SUC\ x1 = IND\_SUC\ x2) = (x1 = x2)) \wedge (\forall x::ind.\ IND\_SUC\ x \neq z))$

thm IND_SUC_SPEC:

$(\forall (x1::ind)\ x2::ind.\ (IND\_SUC\ x1 = IND\_SUC\ x2) = (x1 = x2)) \wedge (\forall x::ind.\ IND\_SUC\ x \neq IND\_0)$

thm IND_SUC_0:

$\forall x::ind.\ IND\_SUC\ x \neq IND\_0$

thm IND_SUC_INJ:

$\forall (x1::ind)\ x2::ind.\ (IND\_SUC\ x1 = IND\_SUC\ x2) = (x1 = x2)$

thm DEF_NUM_REP:

$NUM\_REP = (\lambda a::ind.\ \forall NUM\_REP'::ind \Rightarrow bool.\ (\forall a::ind.\ a = IND\_0 \vee (\exists i::ind.\ a = IND\_SUC\ i \wedge NUM\_REP'\ i) \longrightarrow NUM\_REP'\ a) \longrightarrow NUM\_REP'\ a)$

thm NUM_REP_RULES:

$NUM\_REP\ IND\_0 \wedge (\forall i::ind.\ NUM\_REP\ i \longrightarrow NUM\_REP\ (IND\_SUC\ i))$

thm NUM_REP_CASES:

$\forall\, a{::}ind.\ NUM\_REP\ a = (a = IND\_0 \vee (\exists\, i{::}ind.\ a = IND\_SUC\ i \wedge NUM\_REP\ i))$

thm NUM_REP_INDUCT:

$\forall\, NUM\_REP'{::}ind \Rightarrow bool.\ NUM\_REP'\ IND\_0 \wedge (\forall\, i{::}ind.\ NUM\_REP'\ i \longrightarrow NUM\_REP'\ (IND\_SUC\ i)) \longrightarrow (\forall\, a{::}ind.\ NUM\_REP\ a \longrightarrow NUM\_REP'\ a)$

thm NUM_REP_RULES_conjunct0:

$NUM\_REP\ IND\_0$

thm NUM_REP_RULES_conjunct1:

$\forall\, i{::}ind.\ NUM\_REP\ i \longrightarrow NUM\_REP\ (IND\_SUC\ i)$

thm SUC_INJ:

$\forall\, (m{::}nat)\ n{::}nat.\ (Suc\ m = Suc\ n) = (m = n)$

thm ARITH_EQ_conjunct1:

$((0{::}nat) = (0{::}nat)) = True$

thm DEF_NUMERAL:

$NUM = (\lambda x{::}nat.\ x)$

thm NUMERAL:

$\forall\, n{::}nat.\ NUM\ n = n$

thm NOT_SUC:

$\forall\, n{::}nat.\ Suc\ n \neq (0{::}nat)$

thm num_INDUCTION:

$\forall\, P{::}nat \Rightarrow bool.\ P\ (0{::}nat) \wedge (\forall\, n{::}nat.\ P\ n \longrightarrow P\ (Suc\ n)) \longrightarrow (\forall\, n{::}nat.\ P\ n)$

thm num_Axiom:

$\forall\, (e{::}?'a{::}type)\ f{::}?'a{::}type \Rightarrow nat \Rightarrow ?'a{::}type.\ \exists!fn{::}nat \Rightarrow ?'a{::}type.\ fn\ (0{::}nat) = e \wedge (\forall\, n{::}nat.\ fn\ (Suc\ n) = f\ (fn\ n)\ n)$

thm num_RECURSION:

$\forall\, (e{::}?'a{::}type)\ f{::}?'a{::}type \Rightarrow nat \Rightarrow ?'a{::}type.\ \exists\, fn{::}nat \Rightarrow ?'a{::}type.\ fn\ (0{::}nat) = e \wedge (\forall\, n{::}nat.\ fn\ (Suc\ n) = f\ (fn\ n)\ n)$

thm num_CASES:

$\forall\, m{::}nat.\ m = (0{::}nat) \vee (\exists\, n{::}nat.\ m = Suc\ n)$

thm num_RECURSION_STD:

$\forall\, (e{::}?'a{::}type)\ f{::}nat \Rightarrow ?'a{::}type \Rightarrow ?'a{::}type.\ \exists\, fn{::}nat \Rightarrow ?'a{::}type.\ fn\ (0{::}nat) = e \wedge (\forall\, n{::}nat.\ fn\ (Suc\ n) = f\ n\ (fn\ n))$

thm DEF_BIT0:

$bit0 = (SOME\ fn::nat \Rightarrow nat.\ fn\ (0::nat) = (0::nat) \wedge (\forall n::nat.\ fn\ (Suc\ n) = Suc\ (Suc\ (fn\ n))))$

thm BIT0_DEF:

$bit0\ (0::nat) = (0::nat) \wedge (\forall n::nat.\ bit0\ (Suc\ n) = Suc\ (Suc\ (bit0\ n)))$

thm DEF_BIT1:

$bit1 = (\lambda x::nat.\ Suc\ (bit0\ x))$

thm BIT1_DEF:

$\forall n::nat.\ bit1\ n = Suc\ (bit0\ n)$

thm PRE:

$pred\ (0::nat) = (0::nat) \wedge (\forall n::nat.\ pred\ (Suc\ n) = n)$

thm PRE_conjunct1:

$\forall n::nat.\ pred\ (Suc\ n) = n$

thm PRE_conjunct0:

$pred\ (0::nat) = (0::nat)$

thm ADD:

$(\forall n::nat.\ (0::nat) + n = n) \wedge (\forall (m::nat)\ n::nat.\ Suc\ m + n = Suc\ (m + n))$

thm ADD_conjunct1:

$\forall (m::nat)\ n::nat.\ Suc\ m + n = Suc\ (m + n)$

thm ADD_conjunct0:

$\forall n::nat.\ (0::nat) + n = n$

thm ADD_0:

$\forall m::nat.\ m + (0::nat) = m$

thm ADD_SUC:

$\forall (m::nat)\ n::nat.\ m + Suc\ n = Suc\ (m + n)$

thm ADD_CLAUSES:

$(\forall n::nat.\ (0::nat) + n = n) \wedge (\forall m::nat.\ m + (0::nat) = m) \wedge (\forall (m::nat)\ n::nat.\ Suc\ m + n = Suc\ (m + n)) \wedge (\forall (m::nat)\ n::nat.\ m + Suc\ n = Suc\ (m + n))$

thm ADD_SYM:

$\forall (m::nat)\ n::nat.\ m + n = n + m$

thm ADD_ASSOC:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + (n + p) = m + n + p$

thm ADD_AC:

$(?m::nat) + (?n::nat) = ?n + ?m \wedge ?m + ?n + (?p::nat) = ?m + (?n + ?p)$
$\wedge\ ?m + (?n + ?p) = ?n + (?m + ?p)$

thm ADD_EQ_0:

$\forall (m::nat)\ n::nat.\ (m + n = (0::nat)) = (m = (0::nat) \wedge n = (0::nat))$

thm EQ_ADD_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + n = m + p) = (n = p)$

thm ADD_AC_conjunct0:

$(?m::nat) + (?n::nat) = ?n + ?m$

thm EQ_ADD_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + p = n + p) = (m = n)$

thm EQ_ADD_LCANCEL_0:

$\forall (m::nat)\ n::nat.\ (m + n = m) = (n = (0::nat))$

thm EQ_ADD_RCANCEL_0:

$\forall (m::nat)\ n::nat.\ (m + n = n) = (m = (0::nat))$

thm BIT0_DEF_conjunct1:

$\forall n::nat.\ bit0\ (Suc\ n) = Suc\ (Suc\ (bit0\ n))$

thm BIT0_DEF_conjunct0:

$bit0\ (0::nat) = (0::nat)$

thm BIT0:

$\forall n::nat.\ bit0\ n = n + n$

thm BIT1:

$\forall n::nat.\ bit1\ n = Suc\ (n + n)$

thm BIT0_THM:

$\forall n::nat.\ NUM\ (bit0\ n) = NUM\ n + NUM\ n$

thm BIT1_THM:

$\forall n::nat.\ NUM\ (bit1\ n) = Suc\ (NUM\ n + NUM\ n)$

thm ARITH_ADD_conjunct1:

$(0::nat) + (0::nat) = (0::nat)$

thm ONE:

$(1::nat) = Suc\ (0::nat)$

thm TWO:

$(2::nat) = Suc\ (1::nat)$

thm ADD1:

$\forall m::nat.\ Suc\ m = m + (1::nat)$

thm MULT:

$(\forall n::nat.\ (0::nat) * n = (0::nat)) \wedge (\forall (m::nat)\ n::nat.\ Suc\ m * n = m * n + n)$

thm MULT_CLAUSES_conjunct4:

$\forall (m::nat)\ n::nat.\ Suc\ m * n = m * n + n$

thm MULT_CLAUSES_conjunct0:

$\forall n::nat.\ (0::nat) * n = (0::nat)$

thm MULT_0:

$\forall m::nat.\ m * (0::nat) = (0::nat)$

thm MULT_SUC:

$\forall (m::nat)\ n::nat.\ m * Suc\ n = m + m * n$

thm MULT_CLAUSES:

$(\forall n::nat.\ (0::nat) * n = (0::nat)) \wedge (\forall m::nat.\ m * (0::nat) = (0::nat)) \wedge (\forall n::nat.\ (1::nat) * n = n) \wedge (\forall m::nat.\ m * (1::nat) = m) \wedge (\forall (m::nat)\ n::nat.\ Suc\ m * n = m * n + n) \wedge (\forall (m::nat)\ n::nat.\ m * Suc\ n = m + m * n)$

thm MULT_CLAUSES_conjunct3:

$\forall m::nat.\ m * (1::nat) = m$

thm MULT_CLAUSES_conjunct2:

$\forall n::nat.\ (1::nat) * n = n$

thm MULT_SYM:

$\forall (m::nat)\ n::nat.\ m * n = n * m$

thm LEFT_ADD_DISTRIB:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m * (n + p) = m * n + m * p$

thm MULT_AC_conjunct0:

$(?m::nat) * (?n::nat) = ?n * ?m$

thm RIGHT_ADD_DISTRIB:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + n) * p = m * p + n * p$

thm MULT_ASSOC:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m * (n * p) = m * n * p$

thm MULT_AC:

$(?m::nat) * (?n::nat) = ?n * ?m \wedge ?m * ?n * (?p::nat) = ?m * (?n * ?p) \wedge$
$?m * (?n * ?p) = ?n * (?m * ?p)$

thm MULT_EQ_0:

$\forall (m::nat)\ n::nat.\ (m * n = (0::nat)) = (m = (0::nat) \vee n = (0::nat))$

thm ADD_AC_conjunct1:

$(?m::nat) + (?n::nat) + (?p::nat) = ?m + (?n + ?p)$

thm EQ_MULT_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * n = m * p) = (m = (0::nat) \vee n = p)$

thm EQ_MULT_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * p = n * p) = (m = n \vee p = (0::nat))$

thm MULT_2:

$\forall n::nat.\ (2::nat) * n = n + n$

thm MULT_EQ_1:

$\forall (m::nat)\ n::nat.\ (m * n = (1::nat)) = (m = (1::nat) \wedge n = (1::nat))$

thm EXP:

$(\forall m::nat.\ m^{0::nat} = (1::nat)) \wedge (\forall (m::nat)\ n::nat.\ m^{Suc\ n} = m * m^n)$

thm EXP_conjunct1:

$\forall (m::nat)\ n::nat.\ m^{Suc\ n} = m * m^n$

thm EXP_conjunct0:

$\forall m::nat.\ m^{0::nat} = (1::nat)$

thm EXP_EQ_0:

$\forall (m::nat)\ n::nat.\ (m^n = (0::nat)) = (m = (0::nat) \wedge n \neq (0::nat))$

thm EXP_EQ_1:

$\forall (x::nat)\ n::nat.\ (x^n = (1::nat)) = (x = (1::nat) \vee n = (0::nat))$

thm EXP_ZERO:

$\forall n::nat.\ (0::nat)^n = (if\ n = (0::nat)\ then\ 1::nat\ else\ (0::nat))$

thm MULT_AC_conjunct2:

$(?m::nat) * ((?n::nat) * (?p::nat)) = ?n * (?m * ?p)$

thm MULT_AC_conjunct1:

$(?m::nat) * (?n::nat) * (?p::nat) = ?m * (?n * ?p)$

thm EXP_ADD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m^{n + p} = m^n * m^p$

thm EXP_ONE:

$\forall n::nat.\ (1::nat)^n = (1::nat)$

thm EXP_1:

$\forall n::nat.\ n^{1::nat} = n$

thm EXP_2:

$\forall n::nat.\ n^2 = n * n$

thm MULT_EXP:

$\forall (p::nat)\ (m::nat)\ n::nat.\ (m * n)^p = m^p * n^p$

thm EXP_MULT:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m^{n * p} = m^{np}$

thm LE:

$(\forall m::nat.\ (m \leq (0::nat)) = (m = (0::nat))) \wedge (\forall (m::nat)\ n::nat.\ (m \leq Suc\ n) = (m = Suc\ n \vee m \leq n))$

thm LE_conjunct1:

$\forall (m::nat)\ n::nat.\ (m \leq Suc\ n) = (m = Suc\ n \vee m \leq n)$

thm LE_conjunct0:

$\forall m::nat.\ (m \leq (0::nat)) = (m = (0::nat))$

thm LT:

$(\forall m::nat.\ (m < (0::nat)) = False) \wedge (\forall (m::nat)\ n::nat.\ (m < Suc\ n) = (m = n \vee m < n))$

thm LT_conjunct1:

$\forall (m::nat)\ n::nat.\ (m < Suc\ n) = (m = n \vee m < n)$

thm LT_conjunct0:

$\forall m::nat.\ (m < (0::nat)) = False$

thm DEF_>=:

$(\lambda(x::nat)\ y::nat.\ y \leq x) = (\lambda(x::nat)\ y::nat.\ y \leq x)$

thm GE:

$\forall (n::nat)\ m::nat.\ (n \leq m) = (n \leq m)$

thm DEF_>:

$(\lambda(x::nat)\ y::nat.\ y < x) = (\lambda(x::nat)\ y::nat.\ y < x)$

thm GT:

$\forall (n::nat)\ m::nat.\ (n < m) = (n < m)$

thm DEF_MAX:

$max = (\lambda(x::nat)\ y::nat.\ if\ x \leq y\ then\ y\ else\ x)$

thm MAX:

$\forall (m::nat)\ n::nat.\ max\ m\ n = (if\ m \leq n\ then\ n\ else\ m)$

thm DEF_MIN:

$min = (\lambda(x::nat)\ y::nat.\ if\ x \leq y\ then\ x\ else\ y)$

thm MIN:

$\forall (m::nat)\ n::nat.\ min\ m\ n = (if\ m \leq n\ then\ m\ else\ n)$

thm LE_SUC_LT:

$\forall (m::nat)\ n::nat.\ (Suc\ m \leq n) = (m < n)$

thm LT_SUC_LE:

$\forall (m::nat)\ n::nat.\ (m < Suc\ n) = (m \leq n)$

thm LE_SUC:

$\forall (m::nat)\ n::nat.\ (Suc\ m \leq Suc\ n) = (m \leq n)$

thm LT_SUC:

$\forall (m::nat)\ n::nat.\ (Suc\ m < Suc\ n) = (m < n)$

thm LE_0:

$\forall n::nat.\ (0::nat) \leq n$

thm LT_0:

$\forall n::nat.\ (0::nat) < Suc\ n$

thm LE_REFL:

$\forall n::nat.\ n \leq n$

thm LT_REFL:

$\forall n::nat.\ \neg\ n < n$

thm LE_ANTISYM:

$\forall (m::nat)\ n::nat.\ (m \leq n \land n \leq m) = (m = n)$

thm LT_ANTISYM:

$\forall (m::nat)\ n::nat.\ \neg\ (m < n \wedge n < m)$

thm LET_ANTISYM:

$\forall (m::nat)\ n::nat.\ \neg\ (m \leq n \wedge n < m)$

thm LTE_ANTISYM:

$\forall (m::nat)\ n::nat.\ \neg\ (m < n \wedge n \leq m)$

thm LE_TRANS:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m \leq n \wedge n \leq p \longrightarrow m \leq p$

thm LT_TRANS:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m < n \wedge n < p \longrightarrow m < p$

thm LET_TRANS:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m \leq n \wedge n < p \longrightarrow m < p$

thm LTE_TRANS:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m < n \wedge n \leq p \longrightarrow m < p$

thm LE_CASES:

$\forall (m::nat)\ n::nat.\ m \leq n \vee n \leq m$

thm LT_CASES:

$\forall (m::nat)\ n::nat.\ m < n \vee n < m \vee m = n$

thm LET_CASES:

$\forall (m::nat)\ n::nat.\ m \leq n \vee n < m$

thm LTE_CASES:

$\forall (m::nat)\ n::nat.\ m < n \vee n \leq m$

thm LE_LT:

$\forall (m::nat)\ n::nat.\ (m \leq n) = (m < n \vee m = n)$

thm LT_LE:

$\forall (m::nat)\ n::nat.\ (m < n) = (m \leq n \wedge m \neq n)$

thm NOT_LE:

$\forall (m::nat)\ n::nat.\ (\neg\ m \leq n) = (n < m)$

thm NOT_LT:

$\forall (m::nat)\ n::nat.\ (\neg\ m < n) = (n \leq m)$

thm LT_IMP_LE:

$\forall (m::nat)\ n::nat.\ m < n \longrightarrow m \leq n$

thm EQ_IMP_LE:

$\forall (m::nat)\ n::nat.\ m = n \longrightarrow m \leq n$

thm LT_NZ:

$\forall n::nat.\ ((0::nat) < n) = (n \neq (0::nat))$

thm LE_1:

$(\forall n::nat.\ n \neq (0::nat) \longrightarrow (0::nat) < n) \wedge (\forall n::nat.\ n \neq (0::nat) \longrightarrow (1::nat) \leq n) \wedge (\forall n{>}0::nat.\ n \neq (0::nat)) \wedge (\forall n{>}0::nat.\ (1::nat) \leq n) \wedge (\forall n{\geq}1::nat.\ (0::nat) < n) \wedge (\forall n{\geq}1::nat.\ n \neq (0::nat))$

thm LE_EXISTS:

$\forall (m::nat)\ n::nat.\ (m \leq n) = (\exists d::nat.\ n = m + d)$

thm LT_EXISTS:

$\forall (m::nat)\ n::nat.\ (m < n) = (\exists d::nat.\ n = m + Suc\ d)$

thm LE_ADD:

$\forall (m::nat)\ n::nat.\ m \leq m + n$

thm LE_ADDR:

$\forall (m::nat)\ n::nat.\ n \leq m + n$

thm LT_ADD:

$\forall (m::nat)\ n::nat.\ (m < m + n) = ((0::nat) < n)$

thm LT_ADDR:

$\forall (m::nat)\ n::nat.\ (n < m + n) = ((0::nat) < m)$

thm LE_ADD_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + n \leq m + p) = (n \leq p)$

thm LE_ADD_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + p \leq n + p) = (m \leq n)$

thm LT_ADD_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + n < m + p) = (n < p)$

thm LT_ADD_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m + p < n + p) = (m < n)$

thm ADD_AC_conjunct2:

$(?m::nat) + ((?n::nat) + (?p::nat)) = ?n + (?m + ?p)$

thm LE_ADD2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m \leq p \wedge n \leq q \longrightarrow m + n \leq p + q$

thm LET_ADD2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m \leq p \wedge n < q \longrightarrow m + n < p + q$

thm LTE_ADD2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m < p \wedge n \leq q \longrightarrow m + n < p + q$

thm LT_ADD2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m < p \wedge n < q \longrightarrow m + n < p + q$

thm LT_MULT:

$\forall (m::nat)\ n::nat.\ ((0::nat) < m * n) = ((0::nat) < m \wedge (0::nat) < n)$

thm LE_MULT2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m \leq n \wedge p \leq q \longrightarrow m * p \leq n * q$

thm LT_LMULT:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m \neq (0::nat) \wedge n < p \longrightarrow m * n < m * p$

thm LE_MULT_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * n \leq m * p) = (m = (0::nat) \vee n \leq p)$

thm LE_MULT_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * p \leq n * p) = (m \leq n \vee p = (0::nat))$

thm LT_MULT_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * n < m * p) = (m \neq (0::nat) \wedge n < p)$

thm LT_MULT_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * p < n * p) = (m < n \wedge p \neq (0::nat))$

thm LT_MULT2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m < n \wedge p < q \longrightarrow m * p < n * q$

thm LE_SQUARE_REFL:

$\forall n::nat.\ n \leq n * n$

thm WLOG_LE:

$(\forall (m::nat)\ n::nat.\ (?P::nat \Rightarrow nat \Rightarrow bool)\ m\ n = ?P\ n\ m) \wedge (\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow ?P\ m\ n) \longrightarrow (\forall (m::nat)\ n::nat.\ ?P\ m\ n)$

thm WLOG_LT:

$(\forall m::nat.\ (?P::nat \Rightarrow nat \Rightarrow bool)\ m\ m) \wedge (\forall (m::nat)\ n::nat.\ ?P\ m\ n = ?P\ n\ m) \wedge (\forall (m::nat)\ n::nat.\ m < n \longrightarrow ?P\ m\ n) \longrightarrow (\forall (m::nat)\ y::nat.\ ?P\ m\ y)$

thm num_WF:

$\forall P::nat \Rightarrow bool. (\forall n::nat. (\forall m<n. P\ m) \longrightarrow P\ n) \longrightarrow (\forall n::nat. P\ n)$

thm num_WOP:

$\forall P::nat \Rightarrow bool. (\exists n::nat. P\ n) = (\exists n::nat. P\ n \wedge (\forall m<n. \neg\ P\ m))$

thm num_MAX:

$\forall P::nat \Rightarrow bool. ((\exists x::nat. P\ x) \wedge (\exists M::nat. \forall x::nat. P\ x \longrightarrow x \leq M)) = (\exists m::nat. P\ m \wedge (\forall x::nat. P\ x \longrightarrow x \leq m))$

thm EVEN:

$even\ (0::nat) = True \wedge (\forall n::nat. even\ (Suc\ n) = odd\ n)$

thm EVEN_conjunct1:

$\forall n::nat. even\ (Suc\ n) = odd\ n$

thm EVEN_conjunct0:

$even\ (0::nat) = True$

thm DEF_ODD:

$ODD = (SOME\ ODD::nat \Rightarrow nat \Rightarrow bool. \forall\ \_2184::nat. ODD\ \_2184\ (0::nat) = False \wedge (\forall n::nat. ODD\ \_2184\ (Suc\ n) = (\neg\ ODD\ \_2184\ n)))\ (7::nat)$

thm ODD:

$ODD\ (0::nat) = False \wedge (\forall n::nat. ODD\ (Suc\ n) = (\neg\ ODD\ n))$

thm ODD_conjunct1:

$\forall n::nat. ODD\ (Suc\ n) = (\neg\ ODD\ n)$

thm ODD_conjunct0:

$ODD\ (0::nat) = False$

thm NOT_EVEN:

$\forall n::nat. odd\ n = ODD\ n$

thm NOT_ODD:

$\forall n::nat. (\neg\ ODD\ n) = even\ n$

thm EVEN_OR_ODD:

$\forall n::nat. even\ n \vee ODD\ n$

thm EVEN_AND_ODD:

$\forall n::nat. \neg\ (even\ n \wedge ODD\ n)$

thm EVEN_ADD:

$\forall (m::nat)\ n::nat. even\ (m + n) = (even\ m = even\ n)$

thm EVEN_MULT:

$\forall (m::nat)\ n::nat.\ even\ (m * n) = (even\ m \vee even\ n)$

thm EVEN_EXP:

$\forall (m::nat)\ n::nat.\ even\ m^n = (even\ m \wedge n \neq (0::nat))$

thm ODD_ADD:

$\forall (m::nat)\ n::nat.\ ODD\ (m + n) = (ODD\ m \neq ODD\ n)$

thm ODD_MULT:

$\forall (m::nat)\ n::nat.\ ODD\ (m * n) = (ODD\ m \wedge ODD\ n)$

thm ODD_EXP:

$\forall (m::nat)\ n::nat.\ ODD\ m^n = (ODD\ m \vee n = (0::nat))$

thm EVEN_DOUBLE:

$\forall n::nat.\ even\ ((2::nat) * n)$

thm ODD_DOUBLE:

$\forall n::nat.\ ODD\ (Suc\ ((2::nat) * n))$

thm EVEN_EXISTS_LEMMA:

$\forall n::nat.\ (even\ n \longrightarrow (\exists m::nat.\ n = (2::nat) * m)) \wedge (odd\ n \longrightarrow (\exists m::nat.\ n = Suc\ ((2::nat) * m)))$

thm EVEN_EXISTS:

$\forall n::nat.\ even\ n = (\exists m::nat.\ n = (2::nat) * m)$

thm ODD_EXISTS:

$\forall n::nat.\ ODD\ n = (\exists m::nat.\ n = Suc\ ((2::nat) * m))$

thm EVEN_ODD_DECOMPOSITION:

$\forall n::nat.\ (\exists (k::nat)\ m::nat.\ ODD\ m \wedge n = (2::nat)^k * m) = (n \neq (0::nat))$

thm SUB:

$(\forall m::nat.\ m - (0::nat) = m) \wedge (\forall (m::nat)\ n::nat.\ m - Suc\ n = pred\ (m - n))$

thm SUB_conjunct1:

$\forall (m::nat)\ n::nat.\ m - Suc\ n = pred\ (m - n)$

thm SUB_conjunct0:

$\forall m::nat.\ m - (0::nat) = m$

thm SUB_0:

$\forall m::nat.\ (0::nat) - m = (0::nat) \wedge m - (0::nat) = m$

thm SUB_PRESUC:

$\forall (m::nat)\ n::nat.\ pred\ (Suc\ m - n) = m - n$

thm SUB_SUC:

$\forall (m::nat)\ n::nat.\ Suc\ m - Suc\ n = m - n$

thm SUB_REFL:

$\forall n::nat.\ n - n = (0::nat)$

thm ADD_SUB:

$\forall (m::nat)\ n::nat.\ m + n - n = m$

thm ADD_SUB2:

$\forall (m::nat)\ n::nat.\ m + n - m = n$

thm SUB_EQ_0:

$\forall (m::nat)\ n::nat.\ (m - n = (0::nat)) = (m \leq n)$

thm ADD_SUBR2:

$\forall (m::nat)\ n::nat.\ m - (m + n) = (0::nat)$

thm ADD_SUBR:

$\forall (m::nat)\ n::nat.\ n - (m + n) = (0::nat)$

thm SUB_ADD:

$\forall (m::nat)\ n::nat.\ n \leq m \longrightarrow m - n + n = m$

thm SUB_ADD_LCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + n - (m + p) = n - p$

thm SUB_ADD_RCANCEL:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + p - (n + p) = m - n$

thm LEFT_SUB_DISTRIB:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m * (n - p) = m * n - m * p$

thm RIGHT_SUB_DISTRIB:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m - n) * p = m * p - n * p$

thm SUC_SUB1:

$\forall n::nat.\ Suc\ n - (1::nat) = n$

thm EVEN_SUB:

$\forall (m::nat)\ n::nat.\ even\ (m - n) = (m \leq n \vee even\ m = even\ n)$

thm ODD_SUB:

$\forall (m::nat)\ n::nat.\ ODD\ (m - n) = (n < m \wedge ODD\ m \neq ODD\ n)$

thm FACT:

$fact\ (0::nat) = (1::nat) \wedge (\forall n::nat.\ fact\ (Suc\ n) = Suc\ n * fact\ n)$

thm FACT_conjunct1:

$\forall n::nat.\ fact\ (Suc\ n) = Suc\ n * fact\ n$

thm FACT_conjunct0:

$fact\ (0::nat) = (1::nat)$

thm Hypermap.ZR_LT_1:

$(0::nat) < (1::nat)$

thm FACT_LT:

$\forall n::nat.\ (0::nat) < fact\ n$

thm FACT_LE:

$\forall n::nat.\ (1::nat) \leq fact\ n$

thm FACT_NZ:

$\forall n::nat.\ fact\ n \neq (0::nat)$

thm FACT_MONO:

$\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow fact\ m \leq fact\ n$

thm EXP_LT_0:

$\forall (n::nat)\ x::nat.\ ((0::nat) < x^n) = (x \neq (0::nat) \vee n = (0::nat))$

thm LT_EXP:

$\forall (x::nat)\ (m::nat)\ n::nat.\ (x^m < x^n) = ((2::nat) \leq x \wedge m < n \vee x = (0::nat) \wedge m \neq (0::nat) \wedge n = (0::nat))$

thm LE_EXP:

$\forall (x::nat)\ (m::nat)\ n::nat.\ (x^m \leq x^n) = (if\ x = (0::nat)\ then\ m = (0::nat) \longrightarrow n = (0::nat)\ else\ x = (1::nat) \vee m \leq n)$

thm EQ_EXP:

$\forall (x::nat)\ (m::nat)\ n::nat.\ (x^m = x^n) = (if\ x = (0::nat)\ then\ (m = (0::nat)) = (n = (0::nat))\ else\ x = (1::nat) \vee m = n)$

thm EXP_MONO_LE_IMP:

$\forall (x::nat)\ (y::nat)\ n::nat.\ x \leq y \longrightarrow x^n \leq y^n$

thm EXP_MONO_LT_IMP:

$\forall (x::nat)\ (y::nat)\ n::nat.\ x < y \wedge n \neq (0::nat) \longrightarrow x^n < y^n$

thm EXP_MONO_LE:

$\forall (x::nat)\ (y::nat)\ n::nat.\ (x^n \leq y^n) = (x \leq y \vee n = (0::nat))$

thm EXP_MONO_LT:

$\forall (x::nat)\ (y::nat)\ n::nat.\ (x^n < y^n) = (x < y \wedge n \neq (0::nat))$

thm EXP_MONO_EQ:

$\forall (x::nat)\ (y::nat)\ n::nat.\ (x^n = y^n) = (x = y \vee n = (0::nat))$

thm DIVMOD_EXIST:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow (\exists (q::nat)\ r::nat.\ m = q * n + r \wedge r < n)$

thm DIVMOD_EXIST_0:

$\forall (m::nat)\ n::nat.\ \exists (q::nat)\ r::nat.\ \textit{if}\ n = (0::nat)\ \textit{then}\ q = (0::nat) \wedge r = m\ \textit{else}\ m = q * n + r \wedge r < n$

thm DIVISION_0:

$\forall (m::nat)\ n::nat.\ \textit{if}\ n = (0::nat)\ \textit{then}\ m\ \textit{div}\ n = (0::nat) \wedge m\ \textit{mod}\ n = m\ \textit{else}\ m = m\ \textit{div}\ n * n + m\ \textit{mod}\ n \wedge m\ \textit{mod}\ n < n$

thm DIVISION:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow m = m\ \textit{div}\ n * n + m\ \textit{mod}\ n \wedge m\ \textit{mod}\ n < n$

thm DIVISION_SIMP:

$(\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow m\ \textit{div}\ n * n + m\ \textit{mod}\ n = m) \wedge (\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow n * (m\ \textit{div}\ n) + m\ \textit{mod}\ n = m)$

thm DIVMOD_UNIQ_LEMMA:

$\forall (m::nat)\ (n::nat)\ (q1::nat)\ (r1::nat)\ (q2::nat)\ r2::nat.\ (m = q1 * n + r1 \wedge r1 < n) \wedge m = q2 * n + r2 \wedge r2 < n \longrightarrow q1 = q2 \wedge r1 = r2$

thm DIVMOD_UNIQ:

$\forall (m::nat)\ (n::nat)\ (q::nat)\ r::nat.\ m = q * n + r \wedge r < n \longrightarrow m\ \textit{div}\ n = q \wedge m\ \textit{mod}\ n = r$

thm MOD_UNIQ:

$\forall (m::nat)\ (n::nat)\ (q::nat)\ r::nat.\ m = q * n + r \wedge r < n \longrightarrow m\ \textit{mod}\ n = r$

thm DIV_UNIQ:

$\forall (m::nat)\ (n::nat)\ (q::nat)\ r::nat.\ m = q * n + r \wedge r < n \longrightarrow m\ \textit{div}\ n = q$

thm MOD_MULT:

$\forall (m::nat)\ n::nat.\ m \neq (0::nat) \longrightarrow m * n\ \textit{mod}\ m = (0::nat)$

thm DIV_MULT:

$\forall (m::nat)\ n::nat.\ m \neq (0::nat) \longrightarrow m * n\ div\ m = n$

thm MOD_LT:

$\forall (m::nat)\ n::nat.\ m < n \longrightarrow m\ mod\ n = m$

thm MOD_EQ:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ m = n + q * p \longrightarrow m\ mod\ p = n\ mod\ p$

thm DIV_LE:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow m\ div\ n \leq m$

thm DIV_MUL_LE:

$\forall (m::nat)\ n::nat.\ n * (m\ div\ n) \leq m$

thm MOD_0:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow (0::nat)\ mod\ n = (0::nat)$

thm DIV_0:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow (0::nat)\ div\ n = (0::nat)$

thm MOD_1:

$\forall n::nat.\ n\ mod\ (1::nat) = (0::nat)$

thm DIV_1:

$\forall n::nat.\ n\ div\ (1::nat) = n$

thm DIV_LT:

$\forall (m::nat)\ n::nat.\ m < n \longrightarrow m\ div\ n = (0::nat)$

thm MOD_MOD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n * p \neq (0::nat) \longrightarrow m\ mod\ (n * p)\ mod\ n = m\ mod\ n$

thm MOD_MOD_REFL:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow m\ mod\ n\ mod\ n = m\ mod\ n$

thm DIV_MULT2:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m * p \neq (0::nat) \longrightarrow m * n\ div\ (m * p) = n\ div\ p$

thm MOD_MULT2:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m * p \neq (0::nat) \longrightarrow m * n\ mod\ (m * p) = m * (n\ mod\ p)$

thm MOD_EXISTS:

$\forall (m::nat)\ n::nat.\ (\exists\, q::nat.\ m = n * q) = (if\ n = (0::nat)\ then\ m = (0::nat)\ else\ m\ mod\ n = (0::nat))$

thm LE_RDIV_EQ:

$\forall (a::nat)\ (b::nat)\ n::nat.\ a \neq (0::nat) \longrightarrow (n \leq b\ div\ a) = (a * n \leq b)$

thm LE_LDIV_EQ:

$\forall (a::nat)\ (b::nat)\ n::nat.\ a \neq (0::nat) \longrightarrow (b\ div\ a \leq n) = (b < a * (n + (1::nat)))$

thm LE_LDIV:

$\forall (a::nat)\ (b::nat)\ n::nat.\ a \neq (0::nat) \wedge b \leq a * n \longrightarrow b\ div\ a \leq n$

thm DIV_MONO:

$\forall (m::nat)\ (n::nat)\ p::nat.\ p \neq (0::nat) \wedge m \leq n \longrightarrow m\ div\ p \leq n\ div\ p$

thm DIV_MONO_LT:

$\forall (m::nat)\ (n::nat)\ p::nat.\ p \neq (0::nat) \wedge m + p \leq n \longrightarrow m\ div\ p < n\ div\ p$

thm DIV_EQ_0:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow (m\ div\ n = (0::nat)) = (m < n)$

thm MOD_EQ_0:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow (m\ mod\ n = (0::nat)) = (\exists\, q::nat.\ m = q * n)$

thm EVEN_MOD:

$\forall\, n::nat.\ even\ n = (n\ mod\ (2::nat) = (0::nat))$

thm ODD_MOD:

$\forall\, n::nat.\ ODD\ n = (n\ mod\ (2::nat) = (1::nat))$

thm MOD_MULT_RMOD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n \neq (0::nat) \longrightarrow m * (p\ mod\ n)\ mod\ n = m * p\ mod\ n$

thm MOD_MULT_LMOD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n \neq (0::nat) \longrightarrow m\ mod\ n * p\ mod\ n = m * p\ mod\ n$

thm MOD_MULT_MOD2:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n \neq (0::nat) \longrightarrow m\ mod\ n * (p\ mod\ n)\ mod\ n = m * p\ mod\ n$

thm MOD_EXP_MOD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n \neq (0::nat) \longrightarrow (m\ mod\ n)^p\ mod\ n = m^p\ mod\ n$

thm MOD_MULT_ADD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ (m * n + p)\ mod\ n = p\ mod\ n$

thm DIV_MULT_ADD:

$\forall (a::nat)\ (b::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow (a * n + b)\ div\ n = a + b\ div\ n$

thm MOD_ADD_MOD:

$\forall (a::nat)\ (b::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow (a\ mod\ n + b\ mod\ n)\ mod\ n = (a + b)\ mod\ n$

thm DIV_ADD_MOD:

$\forall (a::nat)\ (b::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow ((a + b)\ mod\ n = a\ mod\ n + b\ mod\ n) = ((a + b)\ div\ n = a\ div\ n + b\ div\ n)$

thm LE_1_conjunct0:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow (0::nat) < n$

thm DIV_REFL:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow n\ div\ n = (1::nat)$

thm MOD_LE:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow m\ mod\ n \leq m$

thm DIV_MONO2:

$\forall (m::nat)\ (n::nat)\ p::nat.\ p \neq (0::nat) \wedge p \leq m \longrightarrow n\ div\ m \leq n\ div\ p$

thm DIV_LE_EXCLUSION:

$\forall (a::nat)\ (b::nat)\ (c::nat)\ d::nat.\ b \neq (0::nat) \wedge b * c < (a + (1::nat)) * d \longrightarrow c\ div\ d \leq a\ div\ b$

thm DIV_EQ_EXCLUSION:

$(?b::nat) * (?c::nat) < ((?a::nat) + (1::nat)) * (?d::nat) \wedge ?a * ?d < (?c + (1::nat)) * ?b \longrightarrow ?a\ div\ ?b = ?c\ div\ ?d$

thm MULT_DIV_LE:

$\forall (m::nat)\ (n::nat)\ p::nat.\ p \neq (0::nat) \longrightarrow m * (n\ div\ p) \leq m * n\ div\ p$

thm DIV_DIV:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n * p \neq (0::nat) \longrightarrow m\ div\ n\ div\ p = m\ div\ (n * p)$

thm DIV_MOD:

$\forall (m::nat)\ (n::nat)\ p::nat.\ n * p \neq (0::nat) \longrightarrow m\ div\ n\ mod\ p = m\ mod\ (n * p)\ div\ n$

thm MOD_MOD_EXP_MIN:

$\forall (x::nat)\ (p::nat)\ (m::nat)\ n::nat.\ p \neq (0::nat) \longrightarrow x\ mod\ p^{m}\ mod\ p^{n} = x\ mod\ p^{min\ m\ n}$

thm PRE_ELIM_THM:

$(?P::nat \Rightarrow bool)\ (pred\ (?n::nat)) = (\forall m::nat.\ ?n = Suc\ m \lor m = (0::nat) \land ?n = (0::nat) \longrightarrow ?P\ m)$

thm PRE_ELIM_THM′:

$(?P::nat \Rightarrow bool)\ (pred\ (?n::nat)) = (\exists m::nat.\ (?n = Suc\ m \lor m = (0::nat) \land ?n = (0::nat)) \land ?P\ m)$

thm SUB_ELIM_THM:

$(?P::nat \Rightarrow bool)\ ((?a::nat) - (?b::nat)) = (\forall d::nat.\ ?a = ?b + d \lor ?a < ?b \land d = (0::nat) \longrightarrow ?P\ d)$

thm SUB_ELIM_THM′:

$(?P::nat \Rightarrow bool)\ ((?a::nat) - (?b::nat)) = (\exists d::nat.\ (?a = ?b + d \lor ?a < ?b \land d = (0::nat)) \land ?P\ d)$

thm DIVMOD_ELIM_THM:

$(?P::nat \Rightarrow nat \Rightarrow bool)\ ((?m::nat)\ div\ (?n::nat))\ (?m\ mod\ ?n) = (\forall (q::nat)\ r::nat.\ ?n = (0::nat) \land q = (0::nat) \land r = ?m \lor ?m = q * ?n + r \land r < ?n \longrightarrow ?P\ q\ r)$

thm DIVMOD_ELIM_THM′:

$(?P::nat \Rightarrow nat \Rightarrow bool)\ ((?m::nat)\ div\ (?n::nat))\ (?m\ mod\ ?n) = (\exists (q::nat)\ r::nat.\ (?n = (0::nat) \land q = (0::nat) \land r = ?m \lor ?m = q * ?n + r \land r < ?n) \land ?P\ q\ r)$

thm DEF_minimal:

$minimal = (\lambda\_4978::nat \Rightarrow bool.\ SOME\ n::nat.\ \_4978\ n \land (\forall m<n.\ \neg\ \_4978\ m))$

thm minimal:

$\forall P::nat \Rightarrow bool.\ minimal\ P = (SOME\ n::nat.\ P\ n \land (\forall m<n.\ \neg\ P\ m))$

thm MINIMAL:

$\forall P::nat \Rightarrow bool.\ (\exists n::nat.\ P\ n) = (P\ (minimal\ P) \land (\forall m<minimal\ P.\ \neg\ P\ m))$

thm TRANSITIVE_STEPWISE_LT_EQ:

$\forall R::nat \Rightarrow nat \Rightarrow bool.\ (\forall (x::nat)\ (y::nat)\ z::nat.\ R\ x\ y \land R\ y\ z \longrightarrow R\ x\ z) \longrightarrow (\forall (m::nat)\ n::nat.\ m < n \longrightarrow R\ m\ n) = (\forall n::nat.\ R\ n\ (Suc\ n))$

thm TRANSITIVE_STEPWISE_LT:

$\forall R::nat \Rightarrow nat \Rightarrow bool.\ (\forall (x::nat)\ (y::nat)\ z::nat.\ R\ x\ y \land R\ y\ z \longrightarrow R\ x\ z) \land (\forall n::nat.\ R\ n\ (Suc\ n)) \longrightarrow (\forall (m::nat)\ n::nat.\ m < n \longrightarrow R\ m\ n)$

thm TRANSITIVE_STEPWISE_LE_EQ:

$\forall R{::}nat \Rightarrow nat \Rightarrow bool.\ (\forall x{::}nat.\ R\ x\ x) \wedge (\forall (x{::}nat)\ (y{::}nat)\ z{::}nat.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z) \longrightarrow (\forall (m{::}nat)\ n{::}nat.\ m \leq n \longrightarrow R\ m\ n) = (\forall n{::}nat.\ R\ n\ (Suc\ n))$

thm TRANSITIVE_STEPWISE_LE:

$\forall R{::}nat \Rightarrow nat \Rightarrow bool.\ (\forall x{::}nat.\ R\ x\ x) \wedge (\forall (x{::}nat)\ (y{::}nat)\ z{::}nat.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z) \wedge (\forall n{::}nat.\ R\ n\ (Suc\ n)) \longrightarrow (\forall (m{::}nat)\ n{::}nat.\ m \leq n \longrightarrow R\ m\ n)$

thm DEF_WF:

$WF = (\lambda\_5021{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ \forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'a{::}type.\ P\ x) \longrightarrow (\exists x{::}?'a{::}type.\ P\ x \wedge (\forall y{::}?'a{::}type.\ \_5021\ y\ x \longrightarrow \neg\ P\ y)))$

thm WF:

$\forall <<{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ WF\ << = (\forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'a{::}type.\ P\ x) \longrightarrow (\exists x{::}?'a{::}type.\ P\ x \wedge (\forall y{::}?'a{::}type.\ <<\ y\ x \longrightarrow \neg\ P\ y)))$

thm WF_EQ:

$WF\ (?<<{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool) = (\forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'a{::}type.\ P\ x) = (\exists x{::}?'a{::}type.\ P\ x \wedge (\forall y{::}?'a{::}type.\ ?<<\ y\ x \longrightarrow \neg\ P\ y)))$

thm WF_IND:

$WF\ (?<<{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool) = (\forall P{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ (\forall y{::}?'a{::}type.\ ?<<\ y\ x \longrightarrow P\ y) \longrightarrow P\ x) \longrightarrow (\forall x{::}?'a{::}type.\ P\ x))$

thm WF_DCHAIN:

$WF\ (?<<{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool) = (\neg\ (\exists s{::}nat \Rightarrow ?'a{::}type.\ \forall n{::}nat.\ ?<<\ (s\ (Suc\ n))\ (s\ n)))$

thm WF_UREC:

$WF\ (?<<{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow bool) \longrightarrow (\forall H{::}(?'b{::}type \Rightarrow ?'a{::}type) \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ (\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ (g{::}?'b{::}type \Rightarrow ?'a{::}type)\ x{::}?'b{::}type.\ (\forall z{::}?'b{::}type.\ ?<<\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow H\ f\ x = H\ g\ x) \longrightarrow (\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ g{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall x{::}?'b{::}type.\ f\ x = H\ f\ x) \wedge (\forall x{::}?'b{::}type.\ g\ x = H\ g\ x) \longrightarrow f = g))$

thm WF_UREC_WF:

$(\forall H{::}(?'a{::}type \Rightarrow bool) \Rightarrow ?'a{::}type \Rightarrow bool.\ (\forall (f{::}?'a{::}type \Rightarrow bool)\ (g{::}?'a{::}type \Rightarrow bool)\ x{::}?'a{::}type.\ (\forall z{::}?'a{::}type.\ (?<<{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow bool)\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow H\ f\ x = H\ g\ x) \longrightarrow (\forall (f{::}?'a{::}type \Rightarrow bool)\ g{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ f\ x = H\ f\ x) \wedge (\forall x{::}?'a{::}type.\ g\ x = H\ g\ x) \longrightarrow f = g)) \longrightarrow WF\ ?<<$

thm WF_REC_INVARIANT:

$WF\ (?{<<}::?'b::type \Rightarrow ?'b::type \Rightarrow bool) \longrightarrow (\forall (H::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type)\ S::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ (\forall z::?'b::type.\ ?{<<}\ z\ x \longrightarrow f\ z = g\ z \land S\ z\ (f\ z)) \longrightarrow H\ f\ x = H\ g\ x \land S\ x\ (H\ f\ x)) \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ f\ x = H\ f\ x))$

thm WF_REC:

$WF\ (?{<<}::?'b::type \Rightarrow ?'b::type \Rightarrow bool) \longrightarrow (\forall H::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ (\forall z::?'b::type.\ ?{<<}\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow H\ f\ x = H\ g\ x) \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ f\ x = H\ f\ x))$

thm WF_REC_WF:

$(\forall H::(?'a::type \Rightarrow nat) \Rightarrow ?'a::type \Rightarrow nat.\ (\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ x::?'a::type.\ (\forall z::?'a::type.\ (?{<<}::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow H\ f\ x = H\ g\ x) \longrightarrow (\exists f::?'a::type \Rightarrow nat.\ \forall x::?'a::type.\ f\ x = H\ f\ x)) \longrightarrow WF\ ?{<<}$

thm WF_EREC:

$WF\ (?{<<}::?'b::type \Rightarrow ?'b::type \Rightarrow bool) \longrightarrow (\forall H::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ (\forall z::?'b::type.\ ?{<<}\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow H\ f\ x = H\ g\ x) \longrightarrow (\exists! f::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ f\ x = H\ f\ x))$

thm WF_SUBSET:

$(\forall (x::?'a::type)\ y::?'a::type.\ (?{<<}::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ y \longrightarrow (?{<<<}::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ y) \land WF\ ?{<<<} \longrightarrow WF\ ?{<<}$

thm WF_MEASURE_GEN:

$\forall m::?'b::type \Rightarrow ?'a::type.\ WF\ (?{<<}::?'a::type \Rightarrow ?'a::type \Rightarrow bool) \longrightarrow WF\ (\lambda (x::?'b::type)\ x'::?'b::type.\ ?{<<}\ (m\ x)\ (m\ x'))$

thm WF_LEX_DEPENDENT:

$\forall (R::?'b::type \Rightarrow ?'b::type \Rightarrow bool)\ S::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ WF\ R \land (\forall a::?'b::type.\ WF\ (S\ a)) \longrightarrow WF\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow ?'b::type \times ?'a::type \Rightarrow bool.\ \forall (r1::?'b::type)\ s1::?'a::type.\ GEQ\ (f\ (r1, s1))\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow bool.\ \forall (r2::?'b::type)\ s2::?'a::type.\ GEQ\ (f\ (r2, s2))\ (R\ r1\ r2 \lor r1 = r2 \land S\ r1\ s1\ s2)))))$

thm WF_LEX:

$\forall (R::?'b::type \Rightarrow ?'b::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ WF\ R \land WF\ S \longrightarrow WF\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow ?'b::type \times ?'a::type \Rightarrow bool.\ \forall (r1::?'b::type)\ s1::?'a::type.\ GEQ\ (f\ (r1, s1))\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow bool.\ \forall (r2::?'b::type)\ s2::?'a::type.\ GEQ\ (f\ (r2, s2))\ (R\ r1\ r2 \lor r1 = r2 \land S\ s1\ s2)))))$

thm WF_POINTWISE:

WF (?<<::?'b::type ⇒ ?'b::type ⇒ bool) ∧ WF (?<<<::?'a::type ⇒ ?'a::type ⇒ bool) ⟶ WF (GABS (λf::?'b::type × ?'a::type ⇒ ?'b::type × ?'a::type ⇒ bool. ∀(x1::?'b::type) y1::?'a::type. GEQ (f (x1, y1)) (GABS (λf::?'b::type × ?'a::type ⇒ bool. ∀(x2::?'b::type) y2::?'a::type. GEQ (f (x2, y2)) (?<< x1 x2 ∧ ?<<< y1 y2)))))

thm WF_num:

WF op <

thm WF_REC_num:

∀H::(nat ⇒ ?'a::type) ⇒ nat ⇒ ?'a::type. (∀(f::nat ⇒ ?'a::type) (g::nat ⇒ ?'a::type) n::nat. (∀m<n. f m = g m) ⟶ H f n = H g n) ⟶ (∃f::nat ⇒ ?'a::type. ∀n::nat. f n = H f n)

thm DEF_MEASURE:

MEASURE = (λ(_6120::?'a::type ⇒ nat) (x::?'a::type) y::?'a::type. _6120 x < _6120 y)

thm MEASURE:

∀m::?'a::type ⇒ nat. MEASURE m = (λ(x::?'a::type) y::?'a::type. m x < m y)

thm WF_MEASURE:

∀m::?'a::type ⇒ nat. WF (MEASURE m)

thm MEASURE_LE:

(∀y::?'a::type. MEASURE (?m::?'a::type ⇒ nat) y (?a::?'a::type) ⟶ MEASURE ?m y (?b::?'a::type)) = (?m ?a ≤ ?m ?b)

thm WF_REFL:

∀x::?'a::type. WF (?<<::?'a::type ⇒ ?'a::type ⇒ bool) ⟶ ¬ ?<< x x

thm WF_FALSE:

WF (λ(x::?'a::type) y::?'a::type. False)

thm WF_REC_TAIL:

∀(P::?'b::type ⇒ bool) (g::?'b::type ⇒ ?'b::type) h::?'b::type ⇒ ?'a::type. ∃f::?'b::type ⇒ ?'a::type. ∀x::?'b::type. f x = (if P x then f (g x) else h x)

thm WF_REC_TAIL_GENERAL:

∀(P::(?'b::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ bool) (G::(?'b::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ ?'b::type) H::(?'b::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ ?'a::type. WF (?<<::?'b::type ⇒ ?'b::type ⇒ bool) ∧ (∀(f::?'b::type ⇒ ?'a::type) (g::?'b::type ⇒ ?'a::type) x::?'b::type. (∀z::?'b::type. ?<< z x ⟶ f z = g z) ⟶ P f x = P g x ∧ G f x = G g x ∧ H f x = H g x) ∧ (∀(f::?'b::type ⇒ ?'a::type) (g::?'b::type ⇒ ?'a::type) x::?'b::type. (∀z::?'b::type. ?<< z x ⟶ f z = g z)

$\longrightarrow H\,f\,x = H\,g\,x) \wedge (\forall\,(f::?'b::type \Rightarrow ?'a::type)\;(x::?'b::type)\;y::?'b::type.\; P\,f\;x \wedge ?<<\,y\;(G\,f\,x) \longrightarrow ?<<\,y\,x) \longrightarrow (\exists\,f::?'b::type \Rightarrow ?'a::type.\;\forall\,x::?'b::type.\; f\,x = (if\;P\;f\;x\;then\;f\;\;(G\;f\;x)\;\;else\;H\;f\;x))$

thm ARITH_ZERO_conjunct0:

$(0::nat) = (0::nat)$

thm ARITH_ZERO_conjunct1:

$bit0\;\;(0::nat) = (0::nat)$

thm ARITH_ZERO:

$(0::nat) = (0::nat) \wedge bit0\;\;(0::nat) = (0::nat)$

thm ARITH_SUC:

$(\forall\,n::nat.\;Suc\;(NUM\;n) = NUM\;(Suc\;n)) \wedge Suc\;(0::nat) = bit1\;(0::nat) \wedge (\forall\,n::nat.\;Suc\;(bit0\;n) = bit1\;n) \wedge (\forall\,n::nat.\;Suc\;(bit1\;n) = bit0\;(Suc\;n))$

thm ARITH_PRE_conjunct1:

$pred\;(0::nat) = (0::nat)$

thm ARITH_PRE:

$(\forall\,n::nat.\;pred\;(NUM\;n) = NUM\;(pred\;n)) \wedge pred\;(0::nat) = (0::nat) \wedge (\forall\,n::nat.\;pred\;(bit0\;n) = (if\;n = (0::nat)\;then\;0::nat\;else\;bit1\;(pred\;n))) \wedge (\forall\,n::nat.\;pred\;(bit1\;n) = bit0\;n)$

thm ARITH_ADD:

$(\forall\,(m::nat)\;n::nat.\;NUM\,m + NUM\,n = NUM\;(m + n)) \wedge (0::nat) + (0::nat) = (0::nat) \wedge (\forall\,n::nat.\;(0::nat) + bit0\,n = bit0\,n) \wedge (\forall\,n::nat.\;(0::nat) + bit1\;n = bit1\;n) \wedge (\forall\,n::nat.\;bit0\;n + (0::nat) = bit0\;n) \wedge (\forall\,n::nat.\;bit1\;n + (0::nat) = bit1\;n) \wedge (\forall\,(m::nat)\;n::nat.\;bit0\;m + bit0\,n = bit0\;(m + n)) \wedge (\forall\,(m::nat)\;n::nat.\;bit0\;m + bit1\;n = bit1\;(m + n)) \wedge (\forall\,(m::nat)\;n::nat.\;bit1\;m + bit0\,n = bit1\;(m + n)) \wedge (\forall\,(m::nat)\;n::nat.\;bit1\;m + bit1\;n = bit0\;(Suc\;(m + n)))$

thm ARITH_MULT_conjunct1:

$(0::nat) * (0::nat) = (0::nat)$

thm ARITH_MULT:

$(\forall\,(m::nat)\;n::nat.\;NUM\;m * NUM\;n = NUM\;(m * n)) \wedge (0::nat) * (0::nat) = (0::nat) \wedge (\forall\,n::nat.\;(0::nat) * bit0\;n = (0::nat)) \wedge (\forall\,n::nat.\;(0::nat) * bit1\;n = (0::nat)) \wedge (\forall\,n::nat.\;bit0\;n * (0::nat) = (0::nat)) \wedge (\forall\,n::nat.\;bit1\;n * (0::nat) = (0::nat)) \wedge (\forall\,(m::nat)\;n::nat.\;bit0\;m * bit0\;n = bit0\;(bit0\;(m * n))) \wedge (\forall\,(m::nat)\;n::nat.\;bit0\;m * bit1\;n = bit0\;m + bit0\;(bit0\;(m * n))) \wedge (\forall\,(m::nat)\;n::nat.\;bit1\;m * bit0\;n = bit0\;n + bit0\;(bit0\;(m * n))) \wedge (\forall\,(m::nat)\;n::nat.\;bit1\;m * bit1\;n = bit1\;m + (bit0\;n + bit0\;(bit0\;(m * n))))$

thm ARITH_EXP_conjunct1:

$(0::nat)^{0::nat} = bit1\ (0::nat)$

thm ARITH_EXP_conjunct9:

$\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit1\ n} = bit1\ m * ((bit1\ m)^n * (bit1\ m)^n)$

thm ARITH_EXP_conjunct8:

$\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit1\ n} = bit0\ m * ((bit0\ m)^n * (bit0\ m)^n)$

thm ARITH_EXP_conjunct7:

$\forall n::nat.\ (0::nat)^{bit1\ n} = (0::nat)$

thm ARITH_EXP_conjunct6:

$\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit0\ n} = (bit1\ m)^n * (bit1\ m)^n$

thm ARITH_EXP_conjunct5:

$\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit0\ n} = (bit0\ m)^n * (bit0\ m)^n$

thm ARITH_EXP_conjunct4:

$\forall n::nat.\ (0::nat)^{bit0\ n} = (0::nat)^n * (0::nat)^n$

thm ARITH_EXP_conjunct3:

$\forall m::nat.\ (bit1\ m)^{0::nat} = bit1\ (0::nat)$

thm ARITH_EXP_conjunct2:

$\forall m::nat.\ (bit0\ m)^{0::nat} = bit1\ (0::nat)$

thm ARITH_EXP:

$(\forall (m::nat)\ n::nat.\ (NUM\ m)^{NUM\ n} = NUM\ m^n) \land (0::nat)^{0::nat} = bit1\ (0::nat) \land (\forall m::nat.\ (bit0\ m)^{0::nat} = bit1\ (0::nat)) \land (\forall m::nat.\ (bit1\ m)^{0::nat} = bit1\ (0::nat)) \land (\forall n::nat.\ (0::nat)^{bit0\ n} = (0::nat)^n * (0::nat)^n) \land (\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit0\ n} = (bit0\ m)^n * (bit0\ m)^n) \land (\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit0\ n} = (bit1\ m)^n * (bit1\ m)^n) \land (\forall n::nat.\ (0::nat)^{bit1\ n} = (0::nat)) \land (\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit1\ n} = bit0\ m * ((bit0\ m)^n * (bit0\ m)^n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit1\ n} = bit1\ m * ((bit1\ m)^n * (bit1\ m)^n))$

thm ARITH_EVEN_conjunct1:

$even\ (0::nat) = True$

thm ARITH_EVEN:

$(\forall n::nat.\ even\ (NUM\ n) = even\ n) \land even\ (0::nat) = True \land (\forall n::nat.\ even\ (bit0\ n) = True) \land (\forall n::nat.\ even\ (bit1\ n) = False)$

thm ARITH_ODD_conjunct1:

$ODD\ (0::nat) = False$

thm ARITH_ODD:

$(\forall n::nat.\ ODD\ (NUM\ n) = ODD\ n) \wedge ODD\ (0::nat) = False \wedge (\forall n::nat.\ ODD\ (bit0\ n) = False) \wedge (\forall n::nat.\ ODD\ (bit1\ n) = True)$

thm ARITH_LE_conjunct1:

$((0::nat) \leq (0::nat)) = True$

thm ARITH_LE:

$(\forall (m::nat)\ n::nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)) \wedge ((0::nat) \leq (0::nat)) = True \wedge (\forall n::nat.\ (bit0\ n \leq (0::nat)) = (n \leq (0::nat))) \wedge (\forall n::nat.\ (bit1\ n \leq (0::nat)) = False) \wedge (\forall n::nat.\ ((0::nat) \leq bit0\ n) = True) \wedge (\forall n::nat.\ ((0::nat) \leq bit1\ n) = True) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit1\ n) = (m \leq n))$

thm ARITH_LE_conjunct9:

$\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit1\ n) = (m \leq n)$

thm ARITH_LE_conjunct8:

$\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit0\ n) = (m < n)$

thm ARITH_LE_conjunct7:

$\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)$

thm ARITH_LE_conjunct6:

$\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)$

thm ARITH_LE_conjunct5:

$\forall n::nat.\ ((0::nat) \leq bit1\ n) = True$

thm ARITH_LE_conjunct4:

$\forall n::nat.\ ((0::nat) \leq bit0\ n) = True$

thm ARITH_LE_conjunct3:

$\forall n::nat.\ (bit1\ n \leq (0::nat)) = False$

thm ARITH_LE_conjunct2:

$\forall n::nat.\ (bit0\ n \leq (0::nat)) = (n \leq (0::nat))$

thm ARITH_LE_conjunct0:

$\forall (m::nat)\ n::nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)$

thm ARITH_LT_conjunct1:

$((0::nat) < (0::nat)) = False$

thm ARITH_LT:

$(\forall (m::nat)\ n::nat.\ (NUM\ m < NUM\ n) = (m < n)) \land ((0::nat) < (0::nat)) = False \land (\forall n::nat.\ (bit0\ n < (0::nat)) = False) \land (\forall n::nat.\ (bit1\ n < (0::nat)) = False) \land (\forall n::nat.\ ((0::nat) < bit0\ n) = ((0::nat) < n)) \land (\forall n::nat.\ ((0::nat) < bit1\ n) = True) \land (\forall (m::nat)\ n::nat.\ (bit0\ m < bit0\ n) = (m < n)) \land (\forall (m::nat)\ n::nat.\ (bit0\ m < bit1\ n) = (m \leq n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m < bit0\ n) = (m < n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m < bit1\ n) = (m < n))$

thm ARITH_GE:

$(\forall (m::nat)\ n::nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)) \land (0::nat) \leq (0::nat) \land (\forall n::nat.\ (bit0\ n \leq (0::nat)) = (n \leq (0::nat))) \land (\forall n::nat.\ \neg\ bit1\ n \leq (0::nat)) \land (\forall n::nat.\ (0::nat) \leq bit0\ n) \land (\forall n::nat.\ (0::nat) \leq bit1\ n) \land (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)) \land (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit0\ n) = (m < n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit1\ n) = (m \leq n))$

thm ARITH_GT:

$(\forall (m::nat)\ n::nat.\ (NUM\ m < NUM\ n) = (m < n)) \land \neg\ (0::nat) < (0::nat) \land (\forall n::nat.\ \neg\ bit0\ n < (0::nat)) \land (\forall n::nat.\ \neg\ bit1\ n < (0::nat)) \land (\forall n::nat.\ ((0::nat) < bit0\ n) = ((0::nat) < n)) \land (\forall n::nat.\ (0::nat) < bit1\ n) \land (\forall (m::nat)\ n::nat.\ (bit0\ m < bit0\ n) = (m < n)) \land (\forall (m::nat)\ n::nat.\ (bit0\ m < bit1\ n) = (m \leq n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m < bit0\ n) = (m < n)) \land (\forall (m::nat)\ n::nat.\ (bit1\ m < bit1\ n) = (m < n))$

thm ARITH_EQ:

$(\forall (m::nat)\ n::nat.\ (NUM\ m = NUM\ n) = (m = n)) \land ((0::nat) = (0::nat)) = True \land (\forall n::nat.\ (bit0\ n = (0::nat)) = (n = (0::nat))) \land (\forall n::nat.\ (bit1\ n = (0::nat)) = False) \land (\forall n::nat.\ ((0::nat) = bit0\ n) = ((0::nat) = n)) \land (\forall n::nat.\ ((0::nat) = bit1\ n) = False) \land (\forall (m::nat)\ n::nat.\ (bit0\ m = bit0\ n) = (m = n)) \land (\forall (m::nat)\ n::nat.\ (bit0\ m = bit1\ n) = False) \land (\forall (m::nat)\ n::nat.\ (bit1\ m = bit0\ n) = False) \land (\forall (m::nat)\ n::nat.\ (bit1\ m = bit1\ n) = (m = n))$

thm ARITH_SUB_conjunct1:

$(0::nat) - (0::nat) = (0::nat)$

thm ARITH_EQ_conjunct9:

$\forall (m::nat)\ n::nat.\ (bit1\ m = bit1\ n) = (m = n)$

thm ARITH_EQ_conjunct8:

$\forall (m::nat)\ n::nat.\ (bit1\ m = bit0\ n) = False$

thm ARITH_EQ_conjunct7:

$\forall (m::nat)\ n::nat.\ (bit0\ m = bit1\ n) = False$

thm ARITH_EQ_conjunct6:

$\forall (m::nat)\ n::nat.\ (bit0\ m = bit0\ n) = (m = n)$

thm ARITH_EQ_conjunct5:

$\forall n{::}nat.\ ((0{::}nat) = bit1\ n) = False$

thm ARITH_EQ_conjunct4:

$\forall n{::}nat.\ ((0{::}nat) = bit0\ n) = ((0{::}nat) = n)$

thm ARITH_EQ_conjunct3:

$\forall n{::}nat.\ (bit1\ n = (0{::}nat)) = False$

thm ARITH_EQ_conjunct2:

$\forall n{::}nat.\ (bit0\ n = (0{::}nat)) = (n = (0{::}nat))$

thm ARITH_EQ_conjunct0:

$\forall (m{::}nat)\ n{::}nat.\ (NUM\ m = NUM\ n) = (m = n)$

thm ARITH_SUB:

$(\forall (m{::}nat)\ n{::}nat.\ NUM\ m - NUM\ n = NUM\ (m - n)) \wedge (0{::}nat) - (0{::}nat) = (0{::}nat) \wedge (\forall n{::}nat.\ (0{::}nat) - bit0\ n = (0{::}nat)) \wedge (\forall n{::}nat.\ (0{::}nat) - bit1\ n = (0{::}nat)) \wedge (\forall n{::}nat.\ bit0\ n - (0{::}nat) = bit0\ n) \wedge (\forall n{::}nat.\ bit1\ n - (0{::}nat) = bit1\ n) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m - bit0\ n = bit0\ (m - n)) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m - bit1\ n = pred\ (bit0\ (m - n))) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m - bit0\ n = (if\ n \le m\ then\ bit1\ (m - n)\ else\ (0{::}nat))) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m - bit1\ n = bit0\ (m - n))$

thm ARITH:

$((0{::}nat) = (0{::}nat) \wedge bit0\ (0{::}nat) = (0{::}nat)) \wedge ((\forall n{::}nat.\ Suc\ (NUM\ n) = NUM\ (Suc\ n)) \wedge Suc\ (0{::}nat) = bit1\ (0{::}nat) \wedge (\forall n{::}nat.\ Suc\ (bit0\ n) = bit1\ n) \wedge (\forall n{::}nat.\ Suc\ (bit1\ n) = bit0\ (Suc\ n))) \wedge ((\forall n{::}nat.\ pred\ (NUM\ n) = NUM\ (pred\ n)) \wedge pred\ (0{::}nat) = (0{::}nat) \wedge (\forall n{::}nat.\ pred\ (bit0\ n) = (if\ n = (0{::}nat)\ then\ 0{::}nat\ else\ bit1\ (pred\ n))) \wedge (\forall n{::}nat.\ pred\ (bit1\ n) = bit0\ n)) \wedge ((\forall (m{::}nat)\ n{::}nat.\ NUM\ m + NUM\ n = NUM\ (m + n)) \wedge (0{::}nat) + (0{::}nat) = (0{::}nat) \wedge (\forall n{::}nat.\ (0{::}nat) + bit0\ n = bit0\ n) \wedge (\forall n{::}nat.\ (0{::}nat) + bit1\ n = bit1\ n) \wedge (\forall n{::}nat.\ bit0\ n + (0{::}nat) = bit0\ n) \wedge (\forall n{::}nat.\ bit1\ n + (0{::}nat) = bit1\ n) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m + bit0\ n = bit0\ (m + n)) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m + bit1\ n = bit1\ (m + n)) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m + bit0\ n = bit1\ (m + n)) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m + bit1\ n = bit0\ (Suc\ (m + n)))) \wedge ((\forall (m{::}nat)\ n{::}nat.\ NUM\ m * NUM\ n = NUM\ (m * n)) \wedge (0{::}nat) * (0{::}nat) = (0{::}nat) \wedge (\forall n{::}nat.\ (0{::}nat) * bit0\ n = (0{::}nat)) \wedge (\forall n{::}nat.\ (0{::}nat) * bit1\ n = (0{::}nat)) \wedge (\forall n{::}nat.\ bit0\ n * (0{::}nat) = (0{::}nat)) \wedge (\forall n{::}nat.\ bit1\ n * (0{::}nat) = (0{::}nat)) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m * bit0\ n = bit0\ (bit0\ (m * n))) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit0\ m * bit1\ n = bit0\ m + bit0\ (bit0\ (m * n))) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m * bit0\ n = bit0\ n + bit0\ (bit0\ (m * n))) \wedge (\forall (m{::}nat)\ n{::}nat.\ bit1\ m * bit1\ n = bit1\ m + (bit0\ n + bit0\ (bit0\ (m * n))))) \wedge ((\forall (m{::}nat)\ n{::}nat.\ (NUM\ m)^{NUM\ n} = NUM\ m^{n}) \wedge (0{::}nat)^{0{::}nat} = bit1\ (0{::}nat) \wedge (\forall m{::}nat.\ (bit0\ m)^{0{::}nat} = bit1\ (0{::}nat)) \wedge (\forall m{::}nat.\ (bit1\ m)^{0{::}nat} = bit1\ (0{::}nat)) \wedge (\forall n{::}nat.\ (0{::}nat)^{bit0\ n} = (0{::}nat)^{n} * (0{::}nat)^{n})$

$\wedge$ $(\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit0\ n} = (bit0\ m)^n * (bit0\ m)^n)$ $\wedge$ $(\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit0\ n} = (bit1\ m)^n * (bit1\ m)^n)$ $\wedge$ $(\forall n::nat.\ (0::nat)^{bit1\ n} = (0::nat))$ $\wedge$ $(\forall (m::nat)\ n::nat.\ (bit0\ m)^{bit1\ n} = bit0\ m * ((bit0\ m)^n * (bit0\ m)^n))$ $\wedge$ $(\forall (m::nat)\ n::nat.\ (bit1\ m)^{bit1\ n} = bit1\ m * ((bit1\ m)^n * (bit1\ m)^n)))$ $\wedge$ $((\forall n::nat.\ even\ (NUM\ n) = even\ n) \wedge even\ (0::nat) = True \wedge (\forall n::nat.\ even\ (bit0\ n) = True) \wedge (\forall n::nat.\ even\ (bit1\ n) = False))$ $\wedge$ $((\forall n::nat.\ ODD\ (NUM\ n) = ODD\ n) \wedge ODD\ (0::nat) = False \wedge (\forall n::nat.\ ODD\ (bit0\ n) = False) \wedge (\forall n::nat.\ ODD\ (bit1\ n) = True))$ $\wedge$ $((\forall (m::nat)\ n::nat.\ (NUM\ m = NUM\ n) = (m = n)) \wedge ((0::nat) = (0::nat)) = True \wedge (\forall n::nat.\ (bit0\ n = (0::nat)) = (n = (0::nat))) \wedge (\forall n::nat.\ (bit1\ n = (0::nat)) = False) \wedge (\forall n::nat.\ ((0::nat) = bit0\ n) = ((0::nat) = n)) \wedge (\forall n::nat.\ ((0::nat) = bit1\ n) = False) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m = bit0\ n) = (m = n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m = bit1\ n) = False) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m = bit0\ n) = False) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m = bit1\ n) = (m = n)))$ $\wedge$ $((\forall (m::nat)\ n::nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)) \wedge ((0::nat) \leq (0::nat)) = True \wedge (\forall n::nat.\ (bit0\ n \leq (0::nat)) = (n \leq (0::nat))) \wedge (\forall n::nat.\ (bit1\ n \leq (0::nat)) = False) \wedge (\forall n::nat.\ ((0::nat) \leq bit0\ n) = True) \wedge (\forall n::nat.\ ((0::nat) \leq bit1\ n) = True) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit1\ n) = (m \leq n)))$ $\wedge$ $((\forall (m::nat)\ n::nat.\ (NUM\ m < NUM\ n) = (m < n)) \wedge ((0::nat) < (0::nat)) = False \wedge (\forall n::nat.\ (bit0\ n < (0::nat)) = False) \wedge (\forall n::nat.\ (bit1\ n < (0::nat)) = False) \wedge (\forall n::nat.\ ((0::nat) < bit0\ n) = ((0::nat) < n)) \wedge (\forall n::nat.\ ((0::nat) < bit1\ n) = True) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m < bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m < bit1\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m < bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m < bit1\ n) = (m < n)))$ $\wedge$ $((\forall (m::nat)\ n::nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)) \wedge (0::nat) \leq (0::nat) \wedge (\forall n::nat.\ (bit0\ n \leq (0::nat)) = (n \leq (0::nat))) \wedge (\forall n::nat.\ \neg\ bit1\ n \leq (0::nat)) \wedge (\forall n::nat.\ (0::nat) \leq bit0\ n) \wedge (\forall n::nat.\ (0::nat) \leq bit1\ n) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m \leq bit1\ n) = (m \leq n)))$ $\wedge$ $((\forall (m::nat)\ n::nat.\ (NUM\ m < NUM\ n) = (m < n)) \wedge \neg\ (0::nat) < (0::nat) \wedge (\forall n::nat.\ \neg\ bit0\ n < (0::nat)) \wedge (\forall n::nat.\ \neg\ bit1\ n < (0::nat)) \wedge (\forall n::nat.\ ((0::nat) < bit0\ n) = ((0::nat) < n)) \wedge (\forall n::nat.\ (0::nat) < bit1\ n) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m < bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit0\ m < bit1\ n) = (m \leq n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m < bit0\ n) = (m < n)) \wedge (\forall (m::nat)\ n::nat.\ (bit1\ m < bit1\ n) = (m < n)))$ $\wedge$ $(\forall (m::nat)\ n::nat.\ NUM\ m - NUM\ n = NUM\ (m - n)) \wedge (0::nat) - (0::nat) = (0::nat) \wedge (\forall n::nat.\ (0::nat) - bit0\ n = (0::nat)) \wedge (\forall n::nat.\ (0::nat) - bit1\ n = (0::nat)) \wedge (\forall n::nat.\ bit0\ n - (0::nat) = bit0\ n) \wedge (\forall n::nat.\ bit1\ n - (0::nat) = bit1\ n) \wedge (\forall (m::nat)\ n::nat.\ bit0\ m - bit0\ n = bit0\ (m - n)) \wedge (\forall (m::nat)\ n::nat.\ bit0\ m - bit1\ n = pred\ (bit0\ (m - n))) \wedge (\forall (m::nat)\ n::nat.\ bit1\ m - bit0\ n = (if\ n \leq m\ then\ bit1\ (m - n)\ else\ (0::nat))) \wedge (\forall (m::nat)\ n::nat.\ bit1\ m - bit1\ n = bit0\ (m - n))$

thm ARITH_LT_conjunct9:

$\forall (m{::}nat)\ n{::}nat.\ (bit1\ m < bit1\ n) = (m < n)$

thm ARITH_LT_conjunct8:

$\forall (m{::}nat)\ n{::}nat.\ (bit1\ m < bit0\ n) = (m < n)$

thm ARITH_LT_conjunct7:

$\forall (m{::}nat)\ n{::}nat.\ (bit0\ m < bit1\ n) = (m \leq n)$

thm ARITH_LT_conjunct6:

$\forall (m{::}nat)\ n{::}nat.\ (bit0\ m < bit0\ n) = (m < n)$

thm ARITH_LT_conjunct5:

$\forall n{::}nat.\ ((0{::}nat) < bit1\ n) = True$

thm ARITH_LT_conjunct4:

$\forall n{::}nat.\ ((0{::}nat) < bit0\ n) = ((0{::}nat) < n)$

thm ARITH_LT_conjunct3:

$\forall n{::}nat.\ (bit1\ n < (0{::}nat)) = False$

thm ARITH_LT_conjunct2:

$\forall n{::}nat.\ (bit0\ n < (0{::}nat)) = False$

thm ARITH_LT_conjunct0:

$\forall (m{::}nat)\ n{::}nat.\ (NUM\ m < NUM\ n) = (m < n)$

thm ARITH_EVEN_conjunct0:

$\forall n{::}nat.\ even\ (NUM\ n) = even\ n$

thm ARITH_EVEN_conjunct3:

$\forall n{::}nat.\ even\ (bit1\ n) = False$

thm ARITH_EVEN_conjunct2:

$\forall n{::}nat.\ even\ (bit0\ n) = True$

thm ARITH_ODD_conjunct0:

$\forall n{::}nat.\ ODD\ (NUM\ n) = ODD\ n$

thm ARITH_ODD_conjunct3:

$\forall n{::}nat.\ ODD\ (bit1\ n) = True$

thm ARITH_ODD_conjunct2:

$\forall n{::}nat.\ ODD\ (bit0\ n) = False$

thm ARITH_SUC_conjunct1:

$Suc\ (0{::}nat) = bit1\ (0{::}nat)$

thm ARITH_SUB_conjunct9:

$\forall (m::nat)\ n::nat.\ bit1\ m - bit1\ n = bit0\ (m - n)$

thm ARITH_SUB_conjunct8:

$\forall (m::nat)\ n::nat.\ bit1\ m - bit0\ n = (if\ n \leq m\ then\ bit1\ (m - n)\ else\ (0::nat))$

thm ARITH_SUB_conjunct7:

$\forall (m::nat)\ n::nat.\ bit0\ m - bit1\ n = pred\ (bit0\ (m - n))$

thm ARITH_SUB_conjunct6:

$\forall (m::nat)\ n::nat.\ bit0\ m - bit0\ n = bit0\ (m - n)$

thm ARITH_SUB_conjunct5:

$\forall n::nat.\ bit1\ n - (0::nat) = bit1\ n$

thm ARITH_SUB_conjunct4:

$\forall n::nat.\ bit0\ n - (0::nat) = bit0\ n$

thm ARITH_SUB_conjunct3:

$\forall n::nat.\ (0::nat) - bit1\ n = (0::nat)$

thm ARITH_SUB_conjunct2:

$\forall n::nat.\ (0::nat) - bit0\ n = (0::nat)$

thm ARITH_SUB_conjunct0:

$\forall (m::nat)\ n::nat.\ NUM\ m - NUM\ n = NUM\ (m - n)$

thm ARITH_GT_conjunct9:

$\forall (m::nat)\ n::nat.\ (bit1\ m < bit1\ n) = (m < n)$

thm ARITH_GT_conjunct8:

$\forall (m::nat)\ n::nat.\ (bit1\ m < bit0\ n) = (m < n)$

thm ARITH_GT_conjunct7:

$\forall (m::nat)\ n::nat.\ (bit0\ m < bit1\ n) = (m \leq n)$

thm ARITH_GT_conjunct6:

$\forall (m::nat)\ n::nat.\ (bit0\ m < bit0\ n) = (m < n)$

thm ARITH_GT_conjunct5:

$\forall n::nat.\ (0::nat) < bit1\ n$

thm ARITH_GT_conjunct4:

$\forall n::nat.\ ((0::nat) < bit0\ n) = ((0::nat) < n)$

thm ARITH_GT_conjunct3:

$\forall\, n{::}nat.\ \neg\ bit1\ n < (0{::}nat)$

thm ARITH_GT_conjunct2:

$\forall\, n{::}nat.\ \neg\ bit0\ n < (0{::}nat)$

thm ARITH_GT_conjunct1:

$\neg\ (0{::}nat) < (0{::}nat)$

thm ARITH_GT_conjunct0:

$\forall\, (m{::}nat)\ n{::}nat.\ (NUM\ m < NUM\ n) = (m < n)$

thm ARITH_GE_conjunct9:

$\forall\, (m{::}nat)\ n{::}nat.\ (bit1\ m \leq bit1\ n) = (m \leq n)$

thm ARITH_GE_conjunct8:

$\forall\, (m{::}nat)\ n{::}nat.\ (bit1\ m \leq bit0\ n) = (m < n)$

thm ARITH_GE_conjunct7:

$\forall\, (m{::}nat)\ n{::}nat.\ (bit0\ m \leq bit1\ n) = (m \leq n)$

thm ARITH_GE_conjunct6:

$\forall\, (m{::}nat)\ n{::}nat.\ (bit0\ m \leq bit0\ n) = (m \leq n)$

thm ARITH_GE_conjunct5:

$\forall\, n{::}nat.\ (0{::}nat) \leq bit1\ n$

thm ARITH_GE_conjunct4:

$\forall\, n{::}nat.\ (0{::}nat) \leq bit0\ n$

thm ARITH_GE_conjunct3:

$\forall\, n{::}nat.\ \neg\ bit1\ n \leq (0{::}nat)$

thm ARITH_GE_conjunct2:

$\forall\, n{::}nat.\ (bit0\ n \leq (0{::}nat)) = (n \leq (0{::}nat))$

thm ARITH_GE_conjunct1:

$(0{::}nat) \leq (0{::}nat)$

thm ARITH_GE_conjunct0:

$\forall\, (m{::}nat)\ n{::}nat.\ (NUM\ m \leq NUM\ n) = (m \leq n)$

thm ARITH_EXP_conjunct0:

$\forall\, (m{::}nat)\ n{::}nat.\ (NUM\ m)^{NUM\ n} = NUM\ m^{n}$

thm ARITH_MULT_conjunct9:

$\forall\, (m{::}nat)\ n{::}nat.\ bit1\ m * bit1\ n = bit1\ m + (bit0\ n + bit0\ (bit0\ (m * n)))$

thm ARITH_MULT_conjunct8:

$\forall (m::nat)\ n::nat.\ bit1\ m * bit0\ n = bit0\ n + bit0\ (bit0\ (m * n))$

thm ARITH_MULT_conjunct7:

$\forall (m::nat)\ n::nat.\ bit0\ m * bit1\ n = bit0\ m + bit0\ (bit0\ (m * n))$

thm ARITH_MULT_conjunct6:

$\forall (m::nat)\ n::nat.\ bit0\ m * bit0\ n = bit0\ (bit0\ (m * n))$

thm ARITH_MULT_conjunct5:

$\forall n::nat.\ bit1\ n * (0::nat) = (0::nat)$

thm ARITH_MULT_conjunct4:

$\forall n::nat.\ bit0\ n * (0::nat) = (0::nat)$

thm ARITH_MULT_conjunct3:

$\forall n::nat.\ (0::nat) * bit1\ n = (0::nat)$

thm ARITH_MULT_conjunct2:

$\forall n::nat.\ (0::nat) * bit0\ n = (0::nat)$

thm ARITH_MULT_conjunct0:

$\forall (m::nat)\ n::nat.\ NUM\ m * NUM\ n = NUM\ (m * n)$

thm ARITH_ADD_conjunct9:

$\forall (m::nat)\ n::nat.\ bit1\ m + bit1\ n = bit0\ (Suc\ (m + n))$

thm ARITH_ADD_conjunct8:

$\forall (m::nat)\ n::nat.\ bit1\ m + bit0\ n = bit1\ (m + n)$

thm ARITH_ADD_conjunct7:

$\forall (m::nat)\ n::nat.\ bit0\ m + bit1\ n = bit1\ (m + n)$

thm ARITH_ADD_conjunct6:

$\forall (m::nat)\ n::nat.\ bit0\ m + bit0\ n = bit0\ (m + n)$

thm ARITH_ADD_conjunct5:

$\forall n::nat.\ bit1\ n + (0::nat) = bit1\ n$

thm ARITH_ADD_conjunct4:

$\forall n::nat.\ bit0\ n + (0::nat) = bit0\ n$

thm ARITH_ADD_conjunct3:

$\forall n::nat.\ (0::nat) + bit1\ n = bit1\ n$

thm ARITH_ADD_conjunct2:

$\forall\, n{::}nat.\ (0{::}nat) + bit0\ n = bit0\ n$

thm ARITH_ADD_conjunct0:

$\forall\, (m{::}nat)\ n{::}nat.\ NUM\ m + NUM\ n = NUM\ (m + n)$

thm ARITH_PRE_conjunct3:

$\forall\, n{::}nat.\ pred\ (bit1\ n) = bit0\ n$

thm ARITH_PRE_conjunct2:

$\forall\, n{::}nat.\ pred\ (bit0\ n) = (if\ n = (0{::}nat)\ then\ 0{::}nat\ else\ bit1\ (pred\ n))$

thm ARITH_PRE_conjunct0:

$\forall\, n{::}nat.\ pred\ (NUM\ n) = NUM\ (pred\ n)$

thm ARITH_SUC_conjunct3:

$\forall\, n{::}nat.\ Suc\ (bit1\ n) = bit0\ (Suc\ n)$

thm ARITH_SUC_conjunct2:

$\forall\, n{::}nat.\ Suc\ (bit0\ n) = bit1\ n$

thm ARITH_SUC_conjunct0:

$\forall\, n{::}nat.\ Suc\ (NUM\ n) = NUM\ (Suc\ n)$

thm INJ_INVERSE2:

$\forall\, P{::}?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type.\ (\forall\, (x1{::}?'c{::}type)\ (y1{::}?'b{::}type)\ (x2{::}?'c{::}type)\ y2{::}?'b{::}type.\ (P\ x1\ y1 = P\ x2\ y2) = (x1 = x2 \wedge y1 = y2)) \longrightarrow (\exists\, (X{::}?'a{::}type \Rightarrow ?'c{::}type)\ Y{::}?'a{::}type \Rightarrow ?'b{::}type.\ \forall\, (x{::}?'c{::}type)\ y{::}?'b{::}type.\ X\ (P\ x\ y) = x \wedge Y\ (P\ x\ y) = y)$

thm DEF_NUMPAIR:

$NUMPAIR = (\lambda(\_9570{::}nat)\ \_9571{::}nat.\ (2{::}nat)^{-9570} * ((2{::}nat) * \_9571 + (1{::}nat)))$

thm NUMPAIR:

$\forall\, (x{::}nat)\ y{::}nat.\ NUMPAIR\ x\ y = (2{::}nat)^{x} * ((2{::}nat) * y + (1{::}nat))$

thm NUMPAIR_INJ_LEMMA:

$\forall\, (x1{::}nat)\ (y1{::}nat)\ (x2{::}nat)\ y2{::}nat.\ NUMPAIR\ x1\ y1 = NUMPAIR\ x2\ y2 \longrightarrow x1 = x2$

thm NUMPAIR_INJ:

$\forall\, (x1{::}nat)\ (y1{::}nat)\ (x2{::}nat)\ y2{::}nat.\ (NUMPAIR\ x1\ y1 = NUMPAIR\ x2\ y2) = (x1 = x2 \wedge y1 = y2)$

thm DEF_NUMFST:

$NUMFST = (SOME\ X{::}nat \Rightarrow nat \Rightarrow nat.\ \forall\ \_9586{::}nat.\ \exists\ Y{::}nat \Rightarrow nat.\ \forall (x{::}nat)\ y{::}nat.\ X\ \_9586\ (NUMPAIR\ x\ y) = x \wedge Y\ (NUMPAIR\ x\ y) = y)\ (12{::}nat)$

thm DEF_NUMSND:

$NUMSND = (SOME\ Y{::}nat \Rightarrow nat \Rightarrow nat.\ \forall (\_9587{::}nat)\ (x{::}nat)\ y{::}nat.\ NUMFST\ (NUMPAIR\ x\ y) = x \wedge Y\ \_9587\ (NUMPAIR\ x\ y) = y)\ (13{::}nat)$

thm NUMPAIR_DEST:

$\forall (x{::}nat)\ y{::}nat.\ NUMFST\ (NUMPAIR\ x\ y) = x \wedge NUMSND\ (NUMPAIR\ x\ y) = y$

thm DEF_NUMSUM:

$NUMSUM = (\lambda(\_9588{::}bool)\ \_9589{::}nat.\ if\ \_9588\ then\ Suc\ ((2{::}nat) * \_9589)\ else\ (2{::}nat) * \_9589)$

thm NUMSUM:

$\forall (b{::}bool)\ x{::}nat.\ NUMSUM\ b\ x = (if\ b\ then\ Suc\ ((2{::}nat) * x)\ else\ (2{::}nat) * x)$

thm NUMSUM_INJ:

$\forall (b1{::}bool)\ (x1{::}nat)\ (b2{::}bool)\ x2{::}nat.\ (NUMSUM\ b1\ x1 = NUMSUM\ b2\ x2) = (b1 = b2 \wedge x1 = x2)$

thm DEF_NUMLEFT:

$NUMLEFT = (SOME\ X{::}nat \Rightarrow nat \Rightarrow bool.\ \forall\ \_9618{::}nat.\ \exists\ Y{::}nat \Rightarrow nat.\ \forall (x{::}bool)\ y{::}nat.\ X\ \_9618\ (NUMSUM\ x\ y) = x \wedge Y\ (NUMSUM\ x\ y) = y)\ (14{::}nat)$

thm DEF_NUMRIGHT:

$NUMRIGHT = (SOME\ Y{::}nat \Rightarrow nat \Rightarrow nat.\ \forall (\_9619{::}nat)\ (x{::}bool)\ y{::}nat.\ NUMLEFT\ (NUMSUM\ x\ y) = x \wedge Y\ \_9619\ (NUMSUM\ x\ y) = y)\ (15{::}nat)$

thm NUMSUM_DEST:

$\forall (x{::}bool)\ y{::}nat.\ NUMLEFT\ (NUMSUM\ x\ y) = x \wedge NUMRIGHT\ (NUMSUM\ x\ y) = y$

thm DEF_INJN:

$INJN = (\lambda(\_9620{::}nat)\ (n{::}nat)\ a{::}?'a{::}type.\ n = \_9620)$

thm INJN:

$\forall\ m{::}nat.\ INJN\ m = (\lambda(n{::}nat)\ a{::}?'a{::}type.\ n = m)$

thm INJN_INJ:

$\forall (n1{::}nat)\ n2{::}nat.\ (INJN\ n1 = INJN\ n2) = (n1 = n2)$

thm DEF_INJA:

$INJA = (\lambda(\_9625::?'a::type)\ (n::nat)\ b::?'a::type.\ b = \_9625)$

thm INJA:

$\forall\, a::?'a::type.\ INJA\ a = (\lambda(n::nat)\ b::?'a::type.\ b = a)$

thm INJA_INJ:

$\forall\,(a1::?'a::type)\ a2::?'a::type.\ (INJA\ a1 = INJA\ a2) = (a1 = a2)$

thm DEF_INJF:

$INJF = (\lambda(\_9632::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool)\ n::nat.\ \_9632\ (NUMFST\ n)\ (NUMSND\ n))$

thm INJF:

$\forall f::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool.\ INJF\ f = (\lambda n::nat.\ f\ (NUMFST\ n)\ (NUMSND\ n))$

thm INJF_INJ:

$\forall\,(f1::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool)\ f2::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool.\ (INJF\ f1 = INJF\ f2) = (f1 = f2)$

thm DEF_INJP:

$INJP = (\lambda(\_9637::nat \Rightarrow ?'a::type \Rightarrow bool)\ (\_9638::nat \Rightarrow ?'a::type \Rightarrow bool)\ (n::nat)\ a::?'a::type.\ if\ NUMLEFT\ n\ then\ \_9637\ (NUMRIGHT\ n)\ a\ else\ \_9638\ (NUMRIGHT\ n)\ a)$

thm INJP:

$\forall\,(f1::nat \Rightarrow ?'a::type \Rightarrow bool)\ f2::nat \Rightarrow ?'a::type \Rightarrow bool.\ INJP\ f1\ f2 = (\lambda(n::nat)\ a::?'a::type.\ if\ NUMLEFT\ n\ then\ f1\ (NUMRIGHT\ n)\ a\ else\ f2\ (NUMRIGHT\ n)\ a)$

thm INJP_INJ:

$\forall\,(f1::nat \Rightarrow ?'a::type \Rightarrow bool)\ (f1'::nat \Rightarrow ?'a::type \Rightarrow bool)\ (f2::nat \Rightarrow ?'a::type \Rightarrow bool)\ f2'::nat \Rightarrow ?'a::type \Rightarrow bool.\ (INJP\ f1\ f2 = INJP\ f1'\ f2') = (f1 = f1' \wedge f2 = f2')$

thm DEF_ZCONSTR:

$ZCONSTR = (\lambda(\_9649::nat)\ (\_9650::?'a::type)\ \_9651::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool.\ INJP\ (INJN\ (Suc\ \_9649))\ (INJP\ (INJA\ \_9650)\ (INJF\ \_9651)))$

thm ZCONSTR:

$\forall\,(c::nat)\ (i::?'a::type)\ r::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool.\ ZCONSTR\ c\ i\ r = INJP\ (INJN\ (Suc\ c))\ (INJP\ (INJA\ i)\ (INJF\ r))$

thm ZBOT:

$ZBOT = INJP\ (INJN\ (0::nat))\ (SOME\ z::nat \Rightarrow ?'a::type \Rightarrow bool.\ True)$

thm ZCONSTR_ZBOT:

$\forall\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow nat \Rightarrow ?'a{::}type \Rightarrow bool.\ ZCONSTR\ c\ i\ r \neq ZBOT$

thm DEF_ZRECSPACE:

$ZRECSPACE = (\lambda a{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ \forall\, ZRECSPACE'{::}(nat \Rightarrow ?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall\, a{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ a = ZBOT \vee (\exists\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow nat \Rightarrow ?'a{::}type \Rightarrow bool.\ a = ZCONSTR\ c\ i\ r \wedge (\forall\, n{::}nat.\ ZRECSPACE'\ (r\ n))) \longrightarrow ZRECSPACE'\ a) \longrightarrow ZRECSPACE'\ a)$

thm ZRECSPACE_RULES:

$ZRECSPACE\ ZBOT \wedge (\forall\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow nat \Rightarrow ?'a{::}type \Rightarrow bool.\ (\forall\, n{::}nat.\ ZRECSPACE\ (r\ n)) \longrightarrow ZRECSPACE\ (ZCONSTR\ c\ i\ r))$

thm ZRECSPACE_CASES:

$\forall\, a{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ ZRECSPACE\ a = (a = ZBOT \vee (\exists\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow nat \Rightarrow ?'a{::}type \Rightarrow bool.\ a = ZCONSTR\ c\ i\ r \wedge (\forall\, n{::}nat.\ ZRECSPACE\ (r\ n))))$

thm ZRECSPACE_INDUCT:

$\forall\, ZRECSPACE'{::}(nat \Rightarrow ?'a{::}type \Rightarrow bool) \Rightarrow bool.\ ZRECSPACE'\ ZBOT \wedge (\forall\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow nat \Rightarrow ?'a{::}type \Rightarrow bool.\ (\forall\, n{::}nat.\ ZRECSPACE'\ (r\ n)) \longrightarrow ZRECSPACE'\ (ZCONSTR\ c\ i\ r)) \longrightarrow (\forall\, a{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ ZRECSPACE\ a \longrightarrow ZRECSPACE'\ a)$

thm ZRECSPACE_RULES_conjunct0:

$ZRECSPACE\ ZBOT$

thm TYDEF_recspace:

$\_mk\_rec\ (\_dest\_rec\ (?a{::}?'a{::}type\ recspace)) = ?a \wedge ZRECSPACE\ (?r{::}nat \Rightarrow ?'a{::}type \Rightarrow bool) = (\_dest\_rec\ (\_mk\_rec\ ?r) = ?r)$

thm BOTTOM:

$BOTTOM = \_mk\_rec\ ZBOT$

thm DEF_CONSTR:

$CONSTR = (\lambda(\_9674{::}nat)\ (\_9675{::}?'a{::}type)\ \_9676{::}nat \Rightarrow ?'a{::}type\ recspace.\ \_mk\_rec\ (ZCONSTR\ \_9674\ \_9675\ (\lambda n{::}nat.\ \_dest\_rec\ (\_9676\ n))))$

thm CONSTR:

$\forall\,(c{::}nat)\ (i{::}?'a{::}type)\ r{::}nat \Rightarrow ?'a{::}type\ recspace.\ CONSTR\ c\ i\ r = \_mk\_rec\ (ZCONSTR\ c\ i\ (\lambda n{::}nat.\ \_dest\_rec\ (r\ n)))$

thm MK_REC_INJ:

$\forall\,(x{::}nat \Rightarrow ?'a{::}type \Rightarrow bool)\ y{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ \_mk\_rec\ x = \_mk\_rec\ y \longrightarrow ZRECSPACE\ x \wedge ZRECSPACE\ y \longrightarrow x = y$

thm DEST_REC_INJ:

$\forall (x::?'a::type\ recspace)\ y::?'a::type\ recspace.\ (\_dest\_rec\ x = \_dest\_rec\ y) = (x = y)$

thm ZRECSPACE_RULES_conjunct1:

$\forall (c::nat)\ (i::?'a::type)\ r::nat \Rightarrow nat \Rightarrow ?'a::type \Rightarrow bool.\ (\forall n::nat.\ ZRECSPACE\ (r\ n)) \longrightarrow ZRECSPACE\ (ZCONSTR\ c\ i\ r)$

thm CONSTR_BOT:

$\forall (c::nat)\ (i::?'a::type)\ r::nat \Rightarrow ?'a::type\ recspace.\ CONSTR\ c\ i\ r \neq BOTTOM$

thm CONSTR_INJ:

$\forall (c1::nat)\ (i1::?'a::type)\ (r1::nat \Rightarrow ?'a::type\ recspace)\ (c2::nat)\ (i2::?'a::type)\ r2::nat \Rightarrow ?'a::type\ recspace.\ (CONSTR\ c1\ i1\ r1 = CONSTR\ c2\ i2\ r2) = (c1 = c2 \wedge i1 = i2 \wedge r1 = r2)$

thm CONSTR_IND:

$\forall P::?'a::type\ recspace \Rightarrow bool.\ P\ BOTTOM \wedge (\forall (c::nat)\ (i::?'a::type)\ r::nat \Rightarrow ?'a::type\ recspace.\ (\forall n::nat.\ P\ (r\ n)) \longrightarrow P\ (CONSTR\ c\ i\ r)) \longrightarrow (\forall x::?'a::type\ recspace.\ P\ x)$

thm CONSTR_REC:

$\forall Fn::nat \Rightarrow ?'b::type \Rightarrow (nat \Rightarrow ?'b::type\ recspace) \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow ?'a::type.\ \exists f::?'b::type\ recspace \Rightarrow ?'a::type.\ \forall (c::nat)\ (i::?'b::type)\ r::nat \Rightarrow ?'b::type\ recspace.\ f\ (CONSTR\ c\ i\ r) = Fn\ c\ i\ r\ (\lambda n::nat.\ f\ (r\ n))$

thm DEF_FCONS:

$FCONS = (SOME\ FCONS::nat \Rightarrow ?'a::type \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow ?'a::type.\ \forall \_9706::nat.\ (\forall (a::?'a::type)\ f::nat \Rightarrow ?'a::type.\ FCONS\ \_9706\ a\ f\ (0::nat) = a) \wedge (\forall (a::?'a::type)\ (f::nat \Rightarrow ?'a::type)\ n::nat.\ FCONS\ \_9706\ a\ f\ (Suc\ n) = f\ n))\ (16::nat)$

thm FCONS:

$(\forall (a::?'a::type)\ f::nat \Rightarrow ?'a::type.\ FCONS\ a\ f\ (0::nat) = a) \wedge (\forall (a::?'a::type)\ (f::nat \Rightarrow ?'a::type)\ n::nat.\ FCONS\ a\ f\ (Suc\ n) = f\ n)$

thm FCONS_conjunct1:

$\forall (a::?'a::type)\ (f::nat \Rightarrow ?'a::type)\ n::nat.\ FCONS\ a\ f\ (Suc\ n) = f\ n$

thm FCONS_conjunct0:

$\forall (a::?'a::type)\ f::nat \Rightarrow ?'a::type.\ FCONS\ a\ f\ (0::nat) = a$

thm FCONS_UNDO:

$\forall f::nat \Rightarrow ?'a::type.\ f = FCONS\ (f\ (0::nat))\ (f \circ Suc)$

thm DEF_FNIL:

$FNIL = (\lambda\_9707::nat.\ SOME\ x::?'a::type.\ True)$

thm FNIL:

$\forall\, n::nat.\ FNIL\ n = (SOME\ x::?'a::type.\ True)$

thm sum_INDUCT:

$\forall\, P::?'b::type + ?'a::type \Rightarrow bool.\ (\forall\, a::?'b::type.\ P\ (Inl\ a)) \wedge (\forall\, a::?'a::type.\ P\ (Inr\ a)) \longrightarrow (\forall\, x::?'b::type + ?'a::type.\ P\ x)$

thm sum_RECURSION:

$\forall\, (Inl'::?'c::type \Rightarrow ?'b::type)\ Inr'::?'a::type \Rightarrow ?'b::type.\ \exists\, fn::?'c::type + ?'a::type \Rightarrow ?'b::type.\ (\forall\, a::?'c::type.\ fn\ (Inl\ a) = Inl'\ a) \wedge (\forall\, a::?'a::type.\ fn\ (Inr\ a) = Inr'\ a)$

thm TYDEF_option:

$\_mk\_option\ (\_dest\_option\ (?a::?'a::type\ HOL\_Light\_Import.option)) = ?a \wedge (\forall\, option'::?'a::type\ recspace \Rightarrow bool.\ (\forall\, a::?'a::type\ recspace.\ a = CONSTR\ (0::nat)\ (SOME\ v::?'a::type.\ True)\ (\lambda n::nat.\ BOTTOM) \vee (\exists\, aa::?'a::type.\ a = CONSTR\ (Suc\ (0::nat))\ aa\ (\lambda n::nat.\ BOTTOM)) \longrightarrow option'\ a) \longrightarrow option'\ (?r::?'a::type\ recspace)) = (\_dest\_option\ (\_mk\_option\ ?r) = ?r)$

thm DEF_NONE:

$NONE = \_mk\_option\ (CONSTR\ (0::nat)\ (SOME\ v::?'a::type.\ True)\ (\lambda n::nat.\ BOTTOM))$

thm DEF_SOME:

$SOME = (\lambda a::?'a::type.\ \_mk\_option\ (CONSTR\ (Suc\ (0::nat))\ a\ (\lambda n::nat.\ BOTTOM)))$

thm option_INDUCT:

$\forall\, P::?'a::type\ HOL\_Light\_Import.option \Rightarrow bool.\ P\ NONE \wedge (\forall\, a::?'a::type.\ P\ (SOME\ a)) \longrightarrow (\forall\, x::?'a::type\ HOL\_Light\_Import.option.\ P\ x)$

thm option_RECURSION:

$\forall\, (NONE'::?'b::type)\ SOME'::?'a::type \Rightarrow ?'b::type.\ \exists\, fn::?'a::type\ HOL\_Light\_Import.option \Rightarrow ?'b::type.\ fn\ NONE = NONE' \wedge (\forall\, a::?'a::type.\ fn\ (SOME\ a) = SOME'\ a)$

thm list_INDUCT:

$\forall\, P::?'a::type\ list \Rightarrow bool.\ P\ [] \wedge (\forall\, (a0::?'a::type)\ a1::?'a::type\ list.\ P\ a1 \longrightarrow P\ (a0\ \#\ a1)) \longrightarrow (\forall\, x::?'a::type\ list.\ P\ x)$

thm list_RECURSION:

$\forall\, (nil'::?'b::type)\ cons'::?'a::type \Rightarrow ?'a::type\ list \Rightarrow ?'b::type \Rightarrow ?'b::type.\ \exists\, fn::?'a::type\ list \Rightarrow ?'b::type.\ fn\ [] = nil' \wedge (\forall\, (a0::?'a::type)\ a1::?'a::type\ list.\ fn\ (a0\ \#\ a1) = cons'\ a0\ a1\ (fn\ a1))$

thm sum_DISTINCT:

$\forall (a::?'b::type)\ a'::?'a::type.\ Inl\ a \neq Inr\ a'$

thm sum_INJECTIVE_conjunct1:

$\forall (a::?'b::type)\ a'::?'b::type.\ (Inr\ a = Inr\ a') = (a = a')$

thm sum_INJECTIVE_conjunct0:

$\forall (a::?'b::type)\ a'::?'b::type.\ (Inl\ a = Inl\ a') = (a = a')$

thm sum_INJECTIVE:

$(\forall (a::?'b::type)\ a'::?'b::type.\ (Inl\ a = Inl\ a') = (a = a')) \wedge (\forall (a::?'a::type)\ a'::?'a::type.\ (Inr\ a = Inr\ a') = (a = a'))$

thm DEF_ISO:

$ISO = (\lambda(\_9809::?'b::type \Rightarrow ?'a::type)\ \_9810::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'a::type.\ \_9809\ (\_9810\ x) = x) \wedge (\forall y::?'b::type.\ \_9810\ (\_9809\ y) = y))$

thm ISO:

$\forall (g::?'b::type \Rightarrow ?'a::type)\ f::?'a::type \Rightarrow ?'b::type.\ ISO\ f\ g = ((\forall x::?'b::type.\ f\ (g\ x) = x) \wedge (\forall y::?'a::type.\ g\ (f\ y) = y))$

thm ISO_REFL:

$ISO\ (\lambda x::?'a::type.\ x)\ (\lambda x::?'a::type.\ x)$

thm ISO_FUN:

$ISO\ (?f::?'d::type \Rightarrow ?'c::type)\ (?f'::?'c::type \Rightarrow ?'d::type) \wedge ISO\ (?g::?'b::type \Rightarrow ?'a::type)\ (?g'::?'a::type \Rightarrow ?'b::type) \longrightarrow ISO\ (\lambda(h::?'d::type \Rightarrow ?'b::type)\ a'::?'c::type.\ ?g\ (h\ (?f'\ a')))\ (\lambda(h::?'c::type \Rightarrow ?'a::type)\ a::?'d::type.\ ?g'\ (h\ (?f\ a)))$

thm ISO_USAGE:

$ISO\ (?f::?'b::type \Rightarrow ?'a::type)\ (?g::?'a::type \Rightarrow ?'b::type) \longrightarrow (\forall P::?'b::type \Rightarrow bool.\ (\forall x::?'b::type.\ P\ x) = (\forall x::?'a::type.\ P\ (?g\ x))) \wedge (\forall P::?'b::type \Rightarrow bool.\ (\exists x::?'b::type.\ P\ x) = (\exists x::?'a::type.\ P\ (?g\ x))) \wedge (\forall (a::?'b::type)\ b::?'a::type.\ (a = ?g\ b) = (?f\ a = b))$

thm HD:

$hd\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = ?h$

thm TL:

$tl\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = ?t$

thm APPEND:

$(\forall l::?'a::type\ list.\ []\ @\ l = l) \wedge (\forall (h::?'a::type)\ (t::?'a::type\ list)\ l::?'a::type\ list.\ (h\ \#\ t)\ @\ l = h\ \#\ t\ @\ l)$

thm APPEND_conjunct1:

$\forall (h::?'a::type)\ (t::?'a::type\ list)\ l::?'a::type\ list.\ (h\ \#\ t)\ @\ l = h\ \#\ t\ @\ l$

thm APPEND_conjunct0:

$\forall l::?'a::type\ list.\ []\ @\ l = l$

thm REVERSE_conjunct0:

$rev\ [] = []$

thm REVERSE_conjunct1:

$rev\ ((?x::?'a::type)\ \#\ (?l::?'a::type\ list)) = rev\ ?l\ @\ [?x]$

thm REVERSE:

$rev\ [] = [] \wedge rev\ ((?x::?'a::type)\ \#\ (?l::?'a::type\ list)) = rev\ ?l\ @\ [?x]$

thm LENGTH:

$length\ [] = (0::nat) \wedge (\forall (h::?'a::type)\ t::?'a::type\ list.\ length\ (h\ \#\ t) = Suc\ (length\ t))$

thm LENGTH_conjunct1:

$\forall (h::?'a::type)\ t::?'a::type\ list.\ length\ (h\ \#\ t) = Suc\ (length\ t)$

thm LENGTH_conjunct0:

$length\ [] = (0::nat)$

thm MAP:

$(\forall f::?'b::type \Rightarrow ?'a::type.\ map\ f\ [] = []) \wedge (\forall (f::?'b::type \Rightarrow ?'a::type)\ (h::?'b::type)\ t::?'b::type\ list.\ map\ f\ (h\ \#\ t) = f\ h\ \#\ map\ f\ t)$

thm MAP_conjunct1:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (h::?'b::type)\ t::?'b::type\ list.\ map\ f\ (h\ \#\ t) = f\ h\ \#\ map\ f\ t$

thm MAP_conjunct0:

$\forall f::?'b::type \Rightarrow ?'a::type.\ map\ f\ [] = []$

thm LAST:

$last\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (if\ ?t = []\ then\ ?h\ else\ last\ ?t)$

thm BUTLAST_conjunct0:

$butlast\ [] = []$

thm BUTLAST_conjunct1:

$butlast\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (if\ ?t = []\ then\ []\ else\ ?h\ \#\ butlast\ ?t)$

thm BUTLAST:

$butlast\ [] = [] \wedge butlast\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (if\ ?t = []\ then\ []\ else\ ?h\ \#\ butlast\ ?t)$

thm REPLICATE_conjunct0:

$replicate\ (0::nat)\ (?x::?'a::type) = []$

thm REPLICATE_conjunct1:

$replicate\ (Suc\ (?n::nat))\ (?x::?'a::type) = ?x\ \#\ replicate\ ?n\ ?x$

thm REPLICATE:

$replicate\ (0::nat)\ (?x::?'a::type) = [] \wedge replicate\ (Suc\ (?n::nat))\ ?x = ?x\ \#\ replicate\ ?n\ ?x$

thm NULL_conjunct0:

$List.null\ [] = True$

thm NULL_conjunct1:

$List.null\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = False$

thm NULL:

$List.null\ [] = True \wedge List.null\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = False$

thm ALL_conjunct0:

$list\_all\ (?P::?'a::type \Rightarrow bool)\ [] = True$

thm ALL_conjunct1:

$list\_all\ (?P::?'a::type \Rightarrow bool)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?P\ ?h \wedge list\_all\ ?P\ ?t)$

thm ALL:

$list\_all\ (?P::?'a::type \Rightarrow bool)\ [] = True \wedge list\_all\ ?P\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?P\ ?h \wedge list\_all\ ?P\ ?t)$

thm EX_conjunct0:

$list\_ex\ (?P::?'a::type \Rightarrow bool)\ [] = False$

thm EX_conjunct1:

$list\_ex\ (?P::?'a::type \Rightarrow bool)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?P\ ?h \vee list\_ex\ ?P\ ?t)$

thm EX:

$list\_ex\ (?P::?'a::type \Rightarrow bool)\ [] = False \wedge list\_ex\ ?P\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?P\ ?h \vee list\_ex\ ?P\ ?t)$

thm ITLIST_conjunct0:

$foldr\ (?f::?'a::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ []\ (?b::?'b::type) = ?b$

thm ITLIST_conjunct1:

$foldr\ (?f::?'a::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list))\ (?b::?'b::type) = ?f\ ?h\ (foldr\ ?f\ ?t\ ?b)$

thm ITLIST:

$foldr\ (?f::?'a::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ []\ (?b::?'b::type) = ?b \wedge foldr\ ?f\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list))\ ?b = ?f\ ?h\ (foldr\ ?f\ ?t\ ?b)$

thm DEF_MEM:

$MEM = (SOME\ MEM::nat \Rightarrow ?'a::type \Rightarrow ?'a::type\ list \Rightarrow bool.\ \forall\ \_10235::nat.\ (\forall\ x::?'a::type.\ MEM\ \_10235\ x\ [] = False) \wedge (\forall\ (h::?'a::type)\ (x::?'a::type)\ t::?'a::type\ list.\ MEM\ \_10235\ x\ (h\ \#\ t) = (x = h \vee MEM\ \_10235\ x\ t)))\ (32::nat)$

thm MEM_conjunct0:

$MEM\ (?x::?'a::type)\ [] = False$

thm MEM_conjunct1:

$MEM\ (?x::?'a::type)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?x = ?h \vee MEM\ ?x\ ?t)$

thm MEM:

$MEM\ (?x::?'a::type)\ [] = False \wedge MEM\ ?x\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (?x = ?h \vee MEM\ ?x\ ?t)$

thm ALL2_DEF_conjunct0:

$list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ []\ (?l2.0::?'a::type\ list) = (?l2.0 = [])$

thm ALL2_DEF_conjunct1:

$list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ ((?h1.0::?'b::type)\ \#\ (?t1.0::?'b::type\ list))\ (?l2.0::?'a::type\ list) = (if\ ?l2.0 = []\ then\ False\ else\ ?P\ ?h1.0\ (hd\ ?l2.0) \wedge list\_all2\ ?P\ ?t1.0\ (tl\ ?l2.0))$

thm ALL2_DEF:

$list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ []\ (?l2.0::?'a::type\ list) = (?l2.0 = []) \wedge list\_all2\ ?P\ ((?h1.0::?'b::type)\ \#\ (?t1.0::?'b::type\ list))\ ?l2.0 = (if\ ?l2.0 = []\ then\ False\ else\ ?P\ ?h1.0\ (hd\ ?l2.0) \wedge list\_all2\ ?P\ ?t1.0\ (tl\ ?l2.0))$

thm ALL2_conjunct0:

$list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ []\ [] = True$

thm ALL2_conjunct1:

$list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ ((?h1.0::?'b::type)\ \#\ (?t1.0::?'b::type\ list))\ [] = False$

thm ALL2_conjunct2:

list_all2 (?P::?'b::type ⇒ ?'a::type ⇒ bool) [] ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = False

thm ALL2_conjunct3:

list_all2 (?P::?'b::type ⇒ ?'a::type ⇒ bool) ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = (?P ?h1.0 ?h2.0 ∧ list_all2 ?P ?t1.0 ?t2.0)

thm ALL2:

list_all2 (?P::?'b::type ⇒ ?'a::type ⇒ bool) [] [] = True ∧ list_all2 ?P ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) [] = False ∧ list_all2 ?P [] ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = False ∧ list_all2 ?P (?h1.0 # ?t1.0) (?h2.0 # ?t2.0) = (?P ?h1.0 ?h2.0 ∧ list_all2 ?P ?t1.0 ?t2.0)

thm DEF_MAP2:

MAP2 = (SOME MAP2::nat ⇒ (?'c::type ⇒ ?'b::type ⇒ ?'a::type) ⇒ ?'c::type list ⇒ ?'b::type list ⇒ ?'a::type list. ∀ _10251::nat. (∀ (f::?'c::type ⇒ ?'b::type ⇒ ?'a::type) l::?'b::type list. MAP2 _10251 f [] l = []) ∧ (∀ (h1::?'c::type) (f::?'c::type ⇒ ?'b::type ⇒ ?'a::type) (t1::?'c::type list) l::?'b::type list. MAP2 _10251 f (h1 # t1) l = f h1 (hd l) # MAP2 _10251 f t1 (tl l))) (34::nat)

thm MAP2_DEF_conjunct0:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) [] (?l::?'a::type list) = []

thm MAP2_DEF_conjunct1:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) (?l::?'a::type list) = ?f ?h1.0 (hd ?l) # MAP2 ?f ?t1.0 (tl ?l)

thm MAP2_DEF:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) [] (?l::?'a::type list) = [] ∧ MAP2 ?f ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ?l = ?f ?h1.0 (hd ?l) # MAP2 ?f ?t1.0 (tl ?l)

thm MAP2_conjunct0:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) [] [] = []

thm MAP2_conjunct1:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = ?f ?h1.0 ?h2.0 # MAP2 ?f ?t1.0 ?t2.0

thm MAP2:

MAP2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type) [] [] = [] ∧ MAP2 ?f ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = ?f ?h1.0 ?h2.0 # MAP2 ?f ?t1.0 ?t2.0

thm DEF_EL:

$EL = (SOME\ EL::nat \Rightarrow nat \Rightarrow ?'a::type\ list \Rightarrow ?'a::type.\ \forall \_10255::nat.\ (\forall l::?'a::type\ list.\ EL\ \_10255\ (0::nat)\ l = hd\ l) \wedge (\forall (n::nat)\ l::?'a::type\ list.\ EL\ \_10255\ (Suc\ n)\ l = EL\ \_10255\ n\ (tl\ l)))\ (35::nat)$

thm EL_conjunct0:

$EL\ (0::nat)\ (?l::?'a::type\ list) = hd\ ?l$

thm EL_conjunct1:

$EL\ (Suc\ (?n::nat))\ (?l::?'a::type\ list) = EL\ ?n\ (tl\ ?l)$

thm EL:

$EL\ (0::nat)\ (?l::?'a::type\ list) = hd\ ?l \wedge EL\ (Suc\ (?n::nat))\ ?l = EL\ ?n\ (tl\ ?l)$

thm FILTER_conjunct0:

$filter\ (?P::?'a::type \Rightarrow bool)\ [] = []$

thm FILTER_conjunct1:

$filter\ (?P::?'a::type \Rightarrow bool)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (if\ ?P\ ?h\ then\ ?h\ \#\ filter\ ?P\ ?t\ else\ filter\ ?P\ ?t)$

thm FILTER:

$filter\ (?P::?'a::type \Rightarrow bool)\ [] = [] \wedge filter\ ?P\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (if\ ?P\ ?h\ then\ ?h\ \#\ filter\ ?P\ ?t\ else\ filter\ ?P\ ?t)$

thm DEF_ASSOC:

$ASSOC = (SOME\ ASSOC::nat \Rightarrow ?'b::type \Rightarrow (?'b::type \times ?'a::type)\ list \Rightarrow ?'a::type.\ \forall (\_10269::nat)\ (h::?'b::type \times ?'a::type)\ (a::?'b::type)\ t::(?'b::type \times ?'a::type)\ list.\ ASSOC\ \_10269\ a\ (h\ \#\ t) = (if\ fst\ h = a\ then\ snd\ h\ else\ ASSOC\ \_10269\ a\ t))\ (37::nat)$

thm ASSOC:

$ASSOC\ (?a::?'a::type)\ ((?h::?'a::type \times ?'b::type)\ \#\ (?t::(?'a::type \times ?'b::type)\ list)) = (if\ fst\ ?h = ?a\ then\ snd\ ?h\ else\ ASSOC\ ?a\ ?t)$

thm DEF_ITLIST2:

$ITLIST2 = (SOME\ ITLIST2::nat \Rightarrow (?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type) \Rightarrow ?'c::type\ list \Rightarrow ?'b::type\ list \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall \_10278::nat.\ (\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (l2::?'b::type\ list)\ b::?'a::type.\ ITLIST2\ \_10278\ f\ []\ l2\ b = b) \wedge (\forall (h1::?'c::type)\ (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (t1::?'c::type\ list)\ (l2::?'b::type\ list)\ b::?'a::type.\ ITLIST2\ \_10278\ f\ (h1\ \#\ t1)\ l2\ b = f\ h1\ (hd\ l2)\ (ITLIST2\ \_10278\ f\ t1\ (tl\ l2)\ b)))\ (38::nat)$

thm ITLIST2_DEF_conjunct0:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) [] (?l2.0::?'a::type list) (?b::?'c::type) = ?b

thm ITLIST2_DEF_conjunct1:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) (?l2.0::?'a::type list) (?b::?'c::type) = ?f ?h1.0 (hd ?l2.0) (ITLIST2 ?f ?t1.0 (tl ?l2.0) ?b)

thm ITLIST2_DEF:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) [] (?l2.0::?'a::type list) (?b::?'c::type) = ?b ∧ ITLIST2 ?f ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ?l2.0 ?b = ?f ?h1.0 (hd ?l2.0) (ITLIST2 ?f ?t1.0 (tl ?l2.0) ?b)

thm ITLIST2_conjunct0:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) [] [] (?b::?'c::type) = ?b

thm ITLIST2_conjunct1:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) (?b::?'c::type) = ?f ?h1.0 ?h2.0 (ITLIST2 ?f ?t1.0 ?t2.0 ?b)

thm ITLIST2:

ITLIST2 (?f::?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ ?'c::type) [] [] (?b::?'c::type) = ?b ∧ ITLIST2 ?f ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) ?b = ?f ?h1.0 ?h2.0 (ITLIST2 ?f ?t1.0 ?t2.0 ?b)

thm ZIP_conjunct0:

zip [] [] = []

thm ZIP_conjunct1:

zip ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = (?h1.0, ?h2.0) # zip ?t1.0 ?t2.0

thm ZIP:

zip [] [] = [] ∧ zip ((?h1.0::?'b::type) # (?t1.0::?'b::type list)) ((?h2.0::?'a::type) # (?t2.0::?'a::type list)) = (?h1.0, ?h2.0) # zip ?t1.0 ?t2.0

thm NOT_CONS_NIL:

∀(h::?'a::type) t::?'a::type list. h # t ≠ []

thm LAST_CLAUSES_conjunct0:

last [?h::?'a::type] = ?h

thm LAST_CLAUSES:

last [?h::?'a::type] = ?h ∧ last (?h # (?k::?'a::type) # (?t::?'a::type list)) = last (?k # ?t)

thm APPENDNIL:

$\forall l::?'a::type\ list.\ l\ @\ []\ =\ l$

thm APPEND_ASSOC:

$\forall (l::?'a::type\ list)\ (m::?'a::type\ list)\ n::?'a::type\ list.\ l\ @\ m\ @\ n = (l\ @\ m)\ @\ n$

thm REVERSE_APPEND:

$\forall (l::?'a::type\ list)\ m::?'a::type\ list.\ rev\ (l\ @\ m) = rev\ m\ @\ rev\ l$

thm REVERSE_REVERSE:

$\forall l::?'a::type\ list.\ rev\ (rev\ l) = l$

thm CONS_11:

$\forall (h1::?'a::type)\ (h2::?'a::type)\ (t1::?'a::type\ list)\ t2::?'a::type\ list.\ (h1\ \#\ t1 = h2\ \#\ t2) = (h1 = h2 \wedge t1 = t2)$

thm list_CASES:

$\forall l::?'a::type\ list.\ l = []\ \vee (\exists (h::?'a::type)\ t::?'a::type\ list.\ l = h\ \#\ t)$

thm LENGTH_APPEND:

$\forall (l::?'a::type\ list)\ m::?'a::type\ list.\ length\ (l\ @\ m) = length\ l + length\ m$

thm MAP_APPEND:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (l1::?'b::type\ list)\ l2::?'b::type\ list.\ map\ f\ (l1\ @\ l2) = map\ f\ l1\ @\ map\ f\ l2$

thm LENGTH_MAP:

$\forall (l::?'b::type\ list)\ f::?'b::type \Rightarrow ?'a::type.\ length\ (map\ f\ l) = length\ l$

thm LENGTH_EQ_NIL:

$\forall l::?'a::type\ list.\ (length\ l = (0::nat)) = (l = [])$

thm LENGTH_EQ_CONS:

$\forall (l::?'a::type\ list)\ n::nat.\ (length\ l = Suc\ n) = (\exists (h::?'a::type)\ t::?'a::type\ list.\ l = h\ \#\ t \wedge length\ t = n)$

thm MAP_o:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ l::?'c::type\ list.\ map\ (g \circ f)\ l = map\ g\ (map\ f\ l)$

thm MAP_EQ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ l::?'b::type\ list.\ list\_all\ (\lambda x::?'b::type.\ f\ x = g\ x)\ l \longrightarrow map\ f\ l = map\ g\ l$

thm ALL_IMP:

$\forall (P::?'a::type \Rightarrow bool)\ (Q::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\forall x::?'a::type.\ MEM\ x\ l \wedge P\ x \longrightarrow Q\ x) \wedge list\_all\ P\ l \longrightarrow list\_all\ Q\ l$

thm NOT_EX:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\neg\ list\_ex\ P\ l) = list\_all\ (\lambda x::?'a::type.\ \neg\ P\ x)\ l$

thm NOT_ALL:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\neg\ list\_all\ P\ l) = list\_ex\ (\lambda x::?'a::type.\ \neg\ P\ x)\ l$

thm ALL_MAP:

$\forall (P::?'b::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ l::?'a::type\ list.\ list\_all\ P\ (map\ f\ l) = list\_all\ (P \circ f)\ l$

thm ALL_T:

$\forall l::?'a::type\ list.\ list\_all\ (\lambda x::?'a::type.\ True)\ l$

thm MAP_EQ_ALL2:

$\forall (l::?'b::type\ list)\ m::?'b::type\ list.\ list\_all2\ (\lambda(x::?'b::type)\ y::?'b::type.\ (?f::?'b::type \Rightarrow ?'a::type)\ x = ?f\ y)\ l\ m \longrightarrow map\ ?f\ l = map\ ?f\ m$

thm ALL2_MAP:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ l::?'a::type\ list.\ list\_all2\ P\ (map\ f\ l)\ l = list\_all\ (\lambda a::?'a::type.\ P\ (f\ a)\ a)\ l$

thm MAP_EQ_DEGEN:

$\forall (l::?'a::type\ list)\ f::?'a::type \Rightarrow ?'a::type.\ list\_all\ (\lambda x::?'a::type.\ f\ x = x)\ l \longrightarrow map\ f\ l = l$

thm ALL2_AND_RIGHT:

$\forall (l::?'b::type\ list)\ (m::?'a::type\ list)\ (P::?'b::type \Rightarrow bool)\ Q::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ list\_all2\ (\lambda(x::?'b::type)\ y::?'a::type.\ P\ x \wedge Q\ x\ y)\ l\ m = (list\_all\ P\ l \wedge list\_all2\ Q\ l\ m)$

thm ITLIST_APPEND:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ (l1::?'b::type\ list)\ l2::?'b::type\ list.\ foldr\ f\ (l1\ @\ l2)\ a = foldr\ f\ l1\ (foldr\ f\ l2\ a)$

thm ITLIST_EXTRA:

$\forall l::?'b::type\ list.\ foldr\ (?f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (l\ @\ [?a::?'b::type])\ (?b::?'a::type) = foldr\ ?f\ l\ (?f\ ?a\ ?b)$

thm ALL_MP:

$\forall (P::?'a::type \Rightarrow bool)\ (Q::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ list\_all\ (\lambda x::?'a::type.\ P\ x \longrightarrow Q\ x)\ l \wedge list\_all\ P\ l \longrightarrow list\_all\ Q\ l$

thm AND_ALL:

$\forall l::?'a::type\ list.\ (list\_all\ (?P::?'a::type \Rightarrow bool)\ l \wedge list\_all\ (?Q::?'a::type \Rightarrow bool)\ l) = list\_all\ (\lambda x::?'a::type.\ ?P\ x \wedge ?Q\ x)\ l$

thm EX_IMP:

$\forall (P::?'a::type \Rightarrow bool)\ (Q::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\forall x::?'a::type.\ MEM\ x\ l \wedge P\ x \longrightarrow Q\ x) \wedge list\_ex\ P\ l \longrightarrow list\_ex\ Q\ l$

thm ALL_MEM:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\forall x::?'a::type.\ MEM\ x\ l \longrightarrow P\ x) = list\_all\ P\ l$

thm LENGTH_REPLICATE:

$\forall (n::nat)\ x::?'a::type.\ length\ (replicate\ n\ x) = n$

thm EX_MAP:

$\forall (P::?'b::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ l::?'a::type\ list.\ list\_ex\ P\ (map\ f\ l) = list\_ex\ (P \circ f)\ l$

thm EXISTS_EX:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\exists x::?'b::type.\ list\_ex\ (P\ x)\ l) = list\_ex\ (\lambda s::?'a::type.\ \exists x::?'b::type.\ P\ x\ s)\ l$

thm FORALL_ALL:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\forall x::?'b::type.\ list\_all\ (P\ x)\ l) = list\_all\ (\lambda s::?'a::type.\ \forall x::?'b::type.\ P\ x\ s)\ l$

thm MEM_APPEND:

$\forall (x::?'a::type)\ (l1::?'a::type\ list)\ l2::?'a::type\ list.\ MEM\ x\ (l1\ @\ l2) = (MEM\ x\ l1 \vee MEM\ x\ l2)$

thm MEM_MAP:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (y::?'a::type)\ l::?'b::type\ list.\ MEM\ y\ (map\ f\ l) = (\exists x::?'b::type.\ MEM\ x\ l \wedge y = f\ x)$

thm FILTER_APPEND:

$\forall (P::?'a::type \Rightarrow bool)\ (l1::?'a::type\ list)\ l2::?'a::type\ list.\ filter\ P\ (l1\ @\ l2) = filter\ P\ l1\ @\ filter\ P\ l2$

thm FILTER_MAP:

$\forall (P::?'b::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ l::?'a::type\ list.\ filter\ P\ (map\ f\ l) = map\ f\ (filter\ (P \circ f)\ l)$

thm MEM_FILTER:

$\forall (P::?'a::type \Rightarrow bool)\ (l::?'a::type\ list)\ x::?'a::type.\ MEM\ x\ (filter\ P\ l) = (P\ x \wedge MEM\ x\ l)$

thm EX_MEM:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\exists x::?'a::type.\ P\ x \wedge MEM\ x\ l) = list\_ex\ P\ l$

thm MAP_FST_ZIP:

$\forall (l1::?'b::type\ list)\ l2::?'a::type\ list.\ length\ l1 = length\ l2 \longrightarrow map\ fst\ (zip\ l1\ l2) = l1$

thm MAP_SND_ZIP:

$\forall (l1::?'b::type\ list)\ l2::?'a::type\ list.\ length\ l1 = length\ l2 \longrightarrow map\ snd\ (zip\ l1\ l2) = l2$

thm MEM_ASSOC:

$\forall (l::(?'b::type \times ?'a::type)\ list)\ x::?'b::type.\ MEM\ (x,\ ASSOC\ x\ l)\ l = MEM\ x\ (map\ fst\ l)$

thm ALL_APPEND:

$\forall (P::?'a::type \Rightarrow bool)\ (l1::?'a::type\ list)\ l2::?'a::type\ list.\ list\_all\ P\ (l1\ @\ l2) = (list\_all\ P\ l1 \wedge list\_all\ P\ l2)$

thm MEM_EL:

$\forall (l::?'a::type\ list)\ n::nat.\ n < length\ l \longrightarrow MEM\ (EL\ n\ l)\ l$

thm MEM_EXISTS_EL:

$\forall (l::?'a::type\ list)\ x::?'a::type.\ MEM\ x\ l = (\exists i<length\ l.\ x = EL\ i\ l)$

thm ALL_EL:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ (\forall i<length\ l.\ P\ (EL\ i\ l)) = list\_all\ P\ l$

thm ALL2_MAP2:

$\forall (l::?'d::type\ list)\ m::?'c::type\ list.\ list\_all2\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (map\ (?f::?'d::type \Rightarrow ?'b::type)\ l)\ (map\ (?g::?'c::type \Rightarrow ?'a::type)\ m) = list\_all2\ (\lambda(x::?'d::type)\ y::?'c::type.\ ?P\ (?f\ x)\ (?g\ y))\ l\ m$

thm AND_ALL2:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (Q::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (l::?'b::type\ list)\ m::?'a::type\ list.\ (list\_all2\ P\ l\ m \wedge list\_all2\ Q\ l\ m) = list\_all2\ (\lambda(x::?'b::type)\ y::?'a::type.\ P\ x\ y \wedge Q\ x\ y)\ l\ m$

thm ALL2_ALL:

$\forall (P::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ list\_all2\ P\ l\ l = list\_all\ (\lambda x::?'a::type.\ P\ x\ x)\ l$

thm APPEND_EQ_NIL:

$\forall (l::?'a::type\ list)\ m::?'a::type\ list.\ (l\ @\ m = []) = (l = [] \wedge m = [])$

thm LENGTH_MAP2:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (l::?'c::type\ list)\ m::?'b::type\ list.\ length\ l = length\ m \longrightarrow length\ (MAP2\ f\ l\ m) = length\ m$

thm MAP_EQ_NIL:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ l::?'b::type\ list.\ (map\ f\ l = []) = (l = [])$

thm INJECTIVE_MAP:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall (l::?'b::type\ list)\ m::?'b::type\ list.\ map\ f\ l = map\ f\ m \longrightarrow l = m) = (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y)$

thm SURJECTIVE_MAP:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall m::?'a::type\ list.\ \exists l::?'b::type\ list.\ map\ f\ l = m) = (\forall y::?'a::type.\ \exists x::?'b::type.\ f\ x = y)$

thm MAP_ID:

$\forall l::?'a::type\ list.\ map\ (\lambda x::?'a::type.\ x)\ l = l$

thm MAP_I:

$map\ id = id$

thm APPEND_BUTLAST_LAST:

$\forall l::?'a::type\ list.\ l \neq [] \longrightarrow butlast\ l\ @\ [last\ l] = l$

thm LAST_APPEND:

$\forall (p::?'a::type\ list)\ q::?'a::type\ list.\ last\ (p\ @\ q) = (if\ q = []\ then\ last\ p\ else\ last\ q)$

thm LENGTH_TL:

$\forall l::?'a::type\ list.\ l \neq [] \longrightarrow length\ (tl\ l) = length\ l - (1::nat)$

thm EL_APPEND:

$\forall (k::nat)\ (l::?'a::type\ list)\ m::?'a::type\ list.\ EL\ k\ (l\ @\ m) = (if\ k < length\ l\ then\ EL\ k\ l\ else\ EL\ (k - length\ l)\ m)$

thm EL_TL:

$\forall n::nat.\ EL\ n\ (tl\ (?l::?'a::type\ list)) = EL\ (n + (1::nat))\ ?l$

thm EL_CONS:

$\forall (n::nat)\ (h::?'a::type)\ t::?'a::type\ list.\ EL\ n\ (h\ \#\ t) = (if\ n = (0::nat)\ then\ h\ else\ EL\ (n - (1::nat))\ t)$

thm LAST_EL:

$\forall l::?'a::type\ list.\ l \neq [] \longrightarrow last\ l = EL\ (length\ l - (1::nat))\ l$

thm HD_APPEND:

$\forall (l::?'a::type\ list)\ m::?'a::type\ list.\ hd\ (l\ @\ m) = (if\ l = []\ then\ hd\ m\ else\ hd\ l)$

thm CONS_HD_TL:

$\forall l::?'a::type\ list.\ l \neq [] \longrightarrow l = hd\ l\ \#\ tl\ l$

thm EL_MAP:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (n::nat)\ l::?'b::type\ list.\ n < length\ l \longrightarrow EL\ n\ (map\ f\ l) = f\ (EL\ n\ l)$

thm MAP_REVERSE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ l::?'b::type\ list.\ rev\ (map\ f\ l) = map\ f\ (rev\ l)$

thm ALL_FILTER:

$\forall (P::?'a::type \Rightarrow bool)\ (Q::?'a::type \Rightarrow bool)\ l::?'a::type\ list.\ list\_all\ P\ (filter\ Q\ l) = list\_all\ (\lambda x::?'a::type.\ Q\ x \longrightarrow P\ x)\ l$

thm MONO_ALL:

$(\forall x::?'a::type.\ (?P::?'a::type \Rightarrow bool)\ x \longrightarrow (?Q::?'a::type \Rightarrow bool)\ x) \longrightarrow list\_all\ ?P\ (?l::?'a::type\ list) \longrightarrow list\_all\ ?Q\ ?l$

thm MONO_ALL2:

$(\forall (x::?'b::type)\ y::?'a::type.\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ y \longrightarrow (?Q::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ y) \longrightarrow list\_all2\ ?P\ (?l::?'b::type\ list)\ (?l'::?'a::type\ list) \longrightarrow list\_all2\ ?Q\ ?l\ ?l'$

thm TYDEF_char:

$\_mk\_char\ (\_dest\_char\ (?a::HOL\_Light\_Import.char)) = ?a \wedge (\forall char'::(bool \times bool \times bool \times bool \times bool \times bool \times bool \times bool)\ recspace \Rightarrow bool.\ (\forall a::(bool \times bool \times bool \times bool \times bool \times bool \times bool \times bool)\ recspace.\ (\exists (a0::bool)\ (a1::bool)\ (a2::bool)\ (a3::bool)\ (a4::bool)\ (a5::bool)\ (a6::bool)\ a7::bool.\ a = CONSTR\ (0::nat)\ (a0, a1, a2, a3, a4, a5, a6, a7)\ (\lambda n::nat.\ BOTTOM)) \longrightarrow char'\ a) \longrightarrow char'\ (?r::(bool \times bool \times bool \times bool \times bool \times bool \times bool \times bool)\ recspace)) = (\_dest\_char\ (\_mk\_char\ ?r) = ?r)$

thm DEF__12186:

$\_12186 = (\lambda (a0::bool)\ (a1::bool)\ (a2::bool)\ (a3::bool)\ (a4::bool)\ (a5::bool)\ (a6::bool)\ a7::bool.\ \_mk\_char\ (CONSTR\ (0::nat)\ (a0, a1, a2, a3, a4, a5, a6, a7)\ (\lambda n::nat.\ BOTTOM)))$

thm DEF_ASCII:

$ASCII = \_12186$

thm char_INDUCT:

$\forall P::HOL\_Light\_Import.char \Rightarrow bool.\ (\forall (a0::bool)\ (a1::bool)\ (a2::bool)\ (a3::bool)\ (a4::bool)\ (a5::bool)\ (a6::bool)\ a7::bool.\ P\ (ASCII\ a0\ a1\ a2\ a3\ a4\ a5\ a6\ a7)) \longrightarrow (\forall x::HOL\_Light\_Import.char.\ P\ x)$

thm char_RECURSION:

$\forall f$::*bool* ⇒ *bool* ⇒ *bool* ⇒ *bool* ⇒ *bool* ⇒ *bool* ⇒ *bool* ⇒ *bool* ⇒ *?'a*::*type*. $\exists$ *fn*::*HOL_Light_Import.char* ⇒ *?'a*::*type*. $\forall$ (*a0*::*bool*) (*a1*::*bool*) (*a2*::*bool*) (*a3*::*bool*) (*a4*::*bool*) (*a5*::*bool*) (*a6*::*bool*) *a7*::*bool*. *fn* (*ASCII a0 a1 a2 a3 a4 a5 a6 a7*) = *f a0 a1 a2 a3 a4 a5 a6 a7*

thm DEF_dist:

HOL_Light_Import.dist = (λ*_12266*::*nat* × *nat*. *fst _12266* − *snd _12266* + (*snd _12266* − *fst _12266*))

thm DIST_LZERO:

$\forall$ *n*::*nat*. *HOL_Light_Import.dist* (*0*::*nat*, *n*) = *n*

thm DIST_RZERO:

$\forall$ *n*::*nat*. *HOL_Light_Import.dist* (*n*, *0*::*nat*) = *n*

thm DIST_LADD:

$\forall$ (*m*::*nat*) (*p*::*nat*) *n*::*nat*. *HOL_Light_Import.dist* (*m* + *n*, *m* + *p*) = *HOL_Light_Import.dist* (*n*, *p*)

thm DIST_RADD:

$\forall$ (*m*::*nat*) (*p*::*nat*) *n*::*nat*. *HOL_Light_Import.dist* (*m* + *p*, *n* + *p*) = *HOL_Light_Import.dist* (*m*, *n*)

thm DIST_LADD_0:

$\forall$ (*m*::*nat*) *n*::*nat*. *HOL_Light_Import.dist* (*m* + *n*, *m*) = *n*

thm DIST_RADD_0:

$\forall$ (*m*::*nat*) *n*::*nat*. *HOL_Light_Import.dist* (*m*, *m* + *n*) = *n*

thm DIST_LMUL:

$\forall$ (*m*::*nat*) (*n*::*nat*) *p*::*nat*. *m* ∗ *HOL_Light_Import.dist* (*n*, *p*) = *HOL_Light_Import.dist* (*m* ∗ *n*, *m* ∗ *p*)

thm DIST_RMUL:

$\forall$ (*m*::*nat*) (*n*::*nat*) *p*::*nat*. *HOL_Light_Import.dist* (*m*, *n*) ∗ *p* = *HOL_Light_Import.dist* (*m* ∗ *p*, *n* ∗ *p*)

thm DIST_ELIM_THM:

(*?P*::*nat* ⇒ *bool*) (*HOL_Light_Import.dist* (*?x*::*nat*, *?y*::*nat*)) = ($\forall$ *d*::*nat*. (*?x* = *?y* + *d* ⟶ *?P d*) ∧ (*?y* = *?x* + *d* ⟶ *?P d*))

thm DIST_LE_CASES:

$\forall$ (*m*::*nat*) (*n*::*nat*) *p*::*nat*. (*HOL_Light_Import.dist* (*m*, *n*) ≤ *p*) = (*m* ≤ *n* + *p* ∧ *n* ≤ *m* + *p*)

thm DIST_ADDBOUND:

$\forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (m,\ n) \leq m + n$

thm DIST_ADD2_REV:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ HOL\_Light\_Import.dist\ (m,\ p) \leq HOL\_Light\_Import.dist\ (m + n,\ p + q) + HOL\_Light\_Import.dist\ (n,\ q)$

thm DIST_ADD2:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ HOL\_Light\_Import.dist\ (m + n,\ p + q) \leq HOL\_Light\_Import.dist\ (m,\ p) + HOL\_Light\_Import.dist\ (n,\ q)$

thm DIST_TRIANGLES_LE:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ (q::nat)\ (r::nat)\ s::nat.\ HOL\_Light\_Import.dist\ (m,\ n) \leq r \wedge HOL\_Light\_Import.dist\ (p,\ q) \leq s \longrightarrow HOL\_Light\_Import.dist\ (m,\ p) \leq HOL\_Light\_Import.dist\ (n,\ q) + (r + s)$

thm BOUNDS_LINEAR:

$\forall (A::nat)\ (B::nat)\ C::nat.\ (\forall n::nat.\ A * n \leq B * n + C) = (A \leq B)$

thm BOUNDS_LINEAR_0:

$\forall (A::nat)\ B::nat.\ (\forall n::nat.\ A * n \leq B) = (A = (0::nat))$

thm BOUNDS_DIVIDED:

$\forall P::nat \Rightarrow nat.\ (\exists B::nat.\ \forall n::nat.\ P\ n \leq B) = (\exists (A::nat)\ B::nat.\ \forall n::nat.\ n * P\ n \leq A * n + B)$

thm BOUNDS_NOTZERO:

$\forall (P::nat \Rightarrow nat \Rightarrow nat)\ (A::nat)\ B::nat.\ P\ (0::nat)\ (0::nat) = (0::nat) \wedge (\forall (m::nat)\ n::nat.\ P\ m\ n \leq A * (m + n) + B) \longrightarrow (\exists B::nat.\ \forall (m::nat)\ n::nat.\ P\ m\ n \leq B * (m + n))$

thm BOUNDS_IGNORE:

$\forall (P::nat \Rightarrow nat)\ Q::nat \Rightarrow nat.\ (\exists B::nat.\ \forall i::nat.\ P\ i \leq Q\ i + B) = (\exists (B::nat)\ N::nat.\ \forall i \geq N.\ P\ i \leq Q\ i + B)$

thm DEF_is_nadd:

$is\_nadd = (\lambda\_12576::nat \Rightarrow nat.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (m * \_12576\ n,\ n * \_12576\ m) \leq B * (m + n))$

thm is_nadd:

$\forall x::nat \Rightarrow nat.\ is\_nadd\ x = (\exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (m * x\ n,\ n * x\ m) \leq B * (m + n))$

thm is_nadd_0:

$is\_nadd\ (\lambda n::nat.\ 0::nat)$

thm TYDEF_nadd:

$mk\_nadd\ (dest\_nadd\ (?a::nadd)) = ?a \wedge is\_nadd\ (?r::nat \Rightarrow nat) = (dest\_nadd\ (mk\_nadd\ ?r) = ?r)$

thm nadd_abs:

$mk\_nadd\ (dest\_nadd\ (?a::nadd)) = ?a$

thm nadd_rep:

$is\_nadd\ (?r::nat \Rightarrow nat) = (dest\_nadd\ (mk\_nadd\ ?r) = ?r)$

thm NADD_CAUCHY:

$\forall x::nadd.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (m * dest\_nadd\ x\ n,\ n * dest\_nadd\ x\ m) \leq B * (m + n)$

thm NADD_BOUND:

$\forall x::nadd.\ \exists (A::nat)\ B::nat.\ \forall n::nat.\ dest\_nadd\ x\ n \leq A * n + B$

thm NADD_MULTIPLICATIVE:

$\forall x::nadd.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ (m * n),\ m * dest\_nadd\ x\ n) \leq B * m + B$

thm NADD_ADDITIVE:

$\forall x::nadd.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ (m + n),\ dest\_nadd\ x\ m + dest\_nadd\ x\ n) \leq B$

thm NADD_SUC:

$\forall x::nadd.\ \exists B::nat.\ \forall n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ (Suc\ n),\ dest\_nadd\ x\ n) \leq B$

thm NADD_DIST_LEMMA:

$\forall x::nadd.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ (m + n),\ dest\_nadd\ x\ m) \leq B * n$

thm NADD_DIST:

$\forall x::nadd.\ \exists B::nat.\ \forall (m::nat)\ n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ m,\ dest\_nadd\ x\ n) \leq B * HOL\_Light\_Import.dist\ (m,\ n)$

thm NADD_ALTMUL:

$\forall (x::nadd)\ y::nadd.\ \exists (A::nat)\ B::nat.\ \forall n::nat.\ HOL\_Light\_Import.dist\ (n * dest\_nadd\ x\ (dest\_nadd\ y\ n),\ dest\_nadd\ x\ n * dest\_nadd\ y\ n) \leq A * n + B$

thm DEF_nadd_eq:

$nadd\_eq = (\lambda(\_12595::nadd)\ \_12596::nadd.\ \exists B::nat.\ \forall n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ \_12595\ n,\ dest\_nadd\ \_12596\ n) \leq B)$

thm nadd_eq:

$\forall (x::nadd)\ y::nadd.\ nadd\_eq\ x\ y = (\exists B::nat.\ \forall n::nat.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ n,\ dest\_nadd\ y\ n) \leq B)$

thm NADD_EQ_REFL:

$\forall x::nadd.\ nadd\_eq\ x\ x$

thm NADD_EQ_SYM:

$\forall (x::nadd)\ y::nadd.\ nadd\_eq\ x\ y = nadd\_eq\ y\ x$

thm NADD_EQ_TRANS:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ x\ y \wedge nadd\_eq\ y\ z \longrightarrow nadd\_eq\ x\ z$

thm DEF_nadd_of_num:

$nadd\_of\_num = (\lambda\_12607::nat.\ mk\_nadd\ (op * \_12607))$

thm nadd_of_num:

$\forall k::nat.\ nadd\_of\_num\ k = mk\_nadd\ (op * k)$

thm NADD_OF_NUM:

$\forall k::nat.\ dest\_nadd\ (nadd\_of\_num\ k) = op * k$

thm NADD_OF_NUM_WELLDEF:

$\forall (m::nat)\ n::nat.\ m = n \longrightarrow nadd\_eq\ (nadd\_of\_num\ m)\ (nadd\_of\_num\ n)$

thm NADD_OF_NUM_EQ:

$\forall (m::nat)\ n::nat.\ nadd\_eq\ (nadd\_of\_num\ m)\ (nadd\_of\_num\ n) = (m = n)$

thm DEF_nadd_le:

$nadd\_le = (\lambda(\_12614::nadd)\ \_12615::nadd.\ \exists B::nat.\ \forall n::nat.\ dest\_nadd\ \_12614\ n \leq dest\_nadd\ \_12615\ n + B)$

thm nadd_le:

$\forall (x::nadd)\ y::nadd.\ nadd\_le\ x\ y = (\exists B::nat.\ \forall n::nat.\ dest\_nadd\ x\ n \leq dest\_nadd\ y\ n + B)$

thm NADD_LE_WELLDEF_LEMMA:

$\forall (x::nadd)\ (x'::nadd)\ (y::nadd)\ y'::nadd.\ nadd\_eq\ x\ x' \wedge nadd\_eq\ y\ y' \wedge nadd\_le\ x\ y \longrightarrow nadd\_le\ x'\ y'$

thm NADD_LE_WELLDEF:

$\forall (x::nadd)\ (x'::nadd)\ (y::nadd)\ y'::nadd.\ nadd\_eq\ x\ x' \wedge nadd\_eq\ y\ y' \longrightarrow nadd\_le\ x\ y = nadd\_le\ x'\ y'$

thm NADD_LE_REFL:

$\forall x::nadd.\ nadd\_le\ x\ x$

thm NADD_LE_TRANS:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_le\ x\ y \wedge nadd\_le\ y\ z \longrightarrow nadd\_le\ x\ z$

thm NADD_LE_ANTISYM:

$\forall (x::nadd)\ y::nadd.\ (nadd\_le\ x\ y \wedge nadd\_le\ y\ x) = nadd\_eq\ x\ y$

thm NADD_LE_TOTAL_LEMMA:

$\forall (x::nadd)\ y::nadd.\ \neg\ nadd\_le\ x\ y \longrightarrow (\forall B::nat.\ \exists n::nat.\ n \neq (0::nat) \wedge dest\_nadd\ y\ n + B < dest\_nadd\ x\ n)$

thm NADD_LE_TOTAL:

$\forall (x::nadd)\ y::nadd.\ nadd\_le\ x\ y \vee nadd\_le\ y\ x$

thm NADD_ARCH:

$\forall x::nadd.\ \exists n::nat.\ nadd\_le\ x\ (nadd\_of\_num\ n)$

thm NADD_OF_NUM_LE:

$\forall (m::nat)\ n::nat.\ nadd\_le\ (nadd\_of\_num\ m)\ (nadd\_of\_num\ n) = (m \leq n)$

thm DEF_nadd_add:

$nadd\_add = (\lambda(\_12630::nadd)\ \_12631::nadd.\ mk\_nadd\ (\lambda n::nat.\ dest\_nadd\ \_12630\ n + dest\_nadd\ \_12631\ n))$

thm nadd_add:

$\forall (x::nadd)\ y::nadd.\ nadd\_add\ x\ y = mk\_nadd\ (\lambda n::nat.\ dest\_nadd\ x\ n + dest\_nadd\ y\ n)$

thm NADD_ADD:

$\forall (x::nadd)\ y::nadd.\ dest\_nadd\ (nadd\_add\ x\ y) = (\lambda n::nat.\ dest\_nadd\ x\ n + dest\_nadd\ y\ n)$

thm NADD_ADD_WELLDEF:

$\forall (x::nadd)\ (x'::nadd)\ (y::nadd)\ y'::nadd.\ nadd\_eq\ x\ x' \wedge nadd\_eq\ y\ y' \longrightarrow nadd\_eq\ (nadd\_add\ x\ y)\ (nadd\_add\ x'\ y')$

thm NADD_ADD_SYM:

$\forall (x::nadd)\ y::nadd.\ nadd\_eq\ (nadd\_add\ x\ y)\ (nadd\_add\ y\ x)$

thm NADD_ADD_ASSOC:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ (nadd\_add\ x\ (nadd\_add\ y\ z))\ (nadd\_add\ (nadd\_add\ x\ y)\ z)$

thm NADD_ADD_LID:

$\forall x::nadd.\ nadd\_eq\ (nadd\_add\ (nadd\_of\_num\ (0::nat))\ x)\ x$

thm NADD_ADD_LCANCEL:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ (nadd\_add\ x\ y)\ (nadd\_add\ x\ z) \longrightarrow nadd\_eq\ y\ z$

thm NADD_LE_ADD:

$\forall (x::nadd)\ y::nadd.\ nadd\_le\ x\ (nadd\_add\ x\ y)$

thm NADD_LE_EXISTS:

$\forall (x::nadd)\ y::nadd.\ nadd\_le\ x\ y \longrightarrow (\exists\, d::nadd.\ nadd\_eq\ y\ (nadd\_add\ x\ d))$

thm NADD_OF_NUM_ADD:

$\forall (m::nat)\ n::nat.\ nadd\_eq\ (nadd\_add\ (nadd\_of\_num\ m)\ (nadd\_of\_num\ n))\ (nadd\_of\_num\ (m + n))$

thm DEF_nadd_mul:

$nadd\_mul = (\lambda(\_12644::nadd)\ \_12645::nadd.\ mk\_nadd\ (\lambda n::nat.\ dest\_nadd\ \_12644\ (dest\_nadd\ \_12645\ n)))$

thm nadd_mul:

$\forall (x::nadd)\ y::nadd.\ nadd\_mul\, x\, y = mk\_nadd\ (\lambda n::nat.\ dest\_nadd\, x\ (dest\_nadd\ y\ n))$

thm NADD_MUL:

$\forall (x::nadd)\ y::nadd.\ dest\_nadd\ (nadd\_mul\, x\, y) = (\lambda n::nat.\ dest\_nadd\, x\ (dest\_nadd\ y\ n))$

thm NADD_MUL_SYM:

$\forall (x::nadd)\ y::nadd.\ nadd\_eq\ (nadd\_mul\ x\ y)\ (nadd\_mul\ y\ x)$

thm NADD_MUL_ASSOC:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ (nadd\_mul\, x\ (nadd\_mul\, y\, z))\ (nadd\_mul\ (nadd\_mul\ x\ y)\ z)$

thm NADD_MUL_LID:

$\forall\, x::nadd.\ nadd\_eq\ (nadd\_mul\ (nadd\_of\_num\ (1::nat))\ x)\ x$

thm NADD_LDISTRIB:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ (nadd\_mul\, x\ (nadd\_add\, y\, z))\ (nadd\_add\ (nadd\_mul\ x\ y)\ (nadd\_mul\ x\ z))$

thm NADD_MUL_WELLDEF_LEMMA:

$\forall (x::nadd)\ (y::nadd)\ y'::nadd.\ nadd\_eq\ y\ y' \longrightarrow nadd\_eq\ (nadd\_mul\ x\ y)\ (nadd\_mul\ x\ y')$

thm NADD_MUL_WELLDEF:

$\forall (x::nadd)\ (x'::nadd)\ (y::nadd)\ y'::nadd.\ nadd\_eq\ x\ x' \wedge nadd\_eq\ y\ y' \longrightarrow nadd\_eq\ (nadd\_mul\ x\ y)\ (nadd\_mul\ x'\ y')$

thm NADD_OF_NUM_MUL:

$\forall (m::nat)\ n::nat.\ nadd\_eq\ (nadd\_mul\ (nadd\_of\_num\ m)\ (nadd\_of\_num\ n))\ (nadd\_of\_num\ (m * n))$

thm NADD_LE_0:

$\forall x::nadd.\ nadd\_le\ (nadd\_of\_num\ (0::nat))\ x$

thm NADD_EQ_IMP_LE:

$\forall (x::nadd)\ y::nadd.\ nadd\_eq\ x\ y \longrightarrow nadd\_le\ x\ y$

thm NADD_LE_LMUL:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_le\ y\ z \longrightarrow nadd\_le\ (nadd\_mul\ x\ y)\ (nadd\_mul\ x\ z)$

thm NADD_LE_RMUL:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_le\ x\ y \longrightarrow nadd\_le\ (nadd\_mul\ x\ z)\ (nadd\_mul\ y\ z)$

thm NADD_LE_RADD:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_le\ (nadd\_add\ x\ z)\ (nadd\_add\ y\ z) = nadd\_le\ x\ y$

thm NADD_LE_LADD:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_le\ (nadd\_add\ x\ y)\ (nadd\_add\ x\ z) = nadd\_le\ y\ z$

thm NADD_RDISTRIB:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ nadd\_eq\ (nadd\_mul\ (nadd\_add\ x\ y)\ z)\ (nadd\_add\ (nadd\_mul\ x\ z)\ (nadd\_mul\ y\ z))$

thm NADD_ARCH_MULT:

$\forall (x::nadd)\ k::nat.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0::nat)) \longrightarrow (\exists N::nat.\ nadd\_le\ (nadd\_of\_num\ k)\ (nadd\_mul\ (nadd\_of\_num\ N)\ x))$

thm NADD_ARCH_ZERO:

$\forall (x::nadd)\ k::nadd.\ (\forall n::nat.\ nadd\_le\ (nadd\_mul\ (nadd\_of\_num\ n)\ x)\ k) \longrightarrow nadd\_eq\ x\ (nadd\_of\_num\ (0::nat))$

thm NADD_ARCH_LEMMA:

$\forall (x::nadd)\ (y::nadd)\ z::nadd.\ (\forall n::nat.\ nadd\_le\ (nadd\_mul\ (nadd\_of\_num\ n)\ x)\ (nadd\_add\ (nadd\_mul\ (nadd\_of\_num\ n)\ y)\ z)) \longrightarrow nadd\_le\ x\ y$

thm NADD_COMPLETE:

$\forall P::nadd \Rightarrow bool.\ (\exists x::nadd.\ P\ x) \land (\exists M::nadd.\ \forall x::nadd.\ P\ x \longrightarrow nadd\_le\ x\ M) \longrightarrow (\exists M::nadd.\ (\forall x::nadd.\ P\ x \longrightarrow nadd\_le\ x\ M) \land (\forall M'::nadd.\ (\forall x::nadd.\ P\ x \longrightarrow nadd\_le\ x\ M') \longrightarrow nadd\_le\ M\ M'))$

thm NADD_UBOUND:

$\forall x{::}nadd.\ \exists (B{::}nat)\ N{::}nat.\ \forall n{\geq}N.\ dest\_nadd\ x\ n \leq B * n$

thm NADD_NONZERO:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists N{::}nat.\ \forall n{\geq}N.\ dest\_nadd\ x\ n \neq (0{::}nat))$

thm NADD_LBOUND:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists (A{::}nat)\ N{::}nat.\ \forall n{\geq}N.\ n \leq A * dest\_nadd\ x\ n)$

thm DEF_nadd_rinv:

$nadd\_rinv = (\lambda(\_12781{::}nadd)\ n{::}nat.\ n * n\ div\ dest\_nadd\ \_12781\ n)$

thm nadd_rinv:

$\forall x{::}nadd.\ nadd\_rinv\ x = (\lambda n{::}nat.\ n * n\ div\ dest\_nadd\ x\ n)$

thm NADD_MUL_LINV_LEMMA0:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists (A{::}nat)\ B{::}nat.\ \forall n{::}nat.\ nadd\_rinv\ x\ n \leq A * n + B)$

thm NADD_MUL_LINV_LEMMA1:

$\forall (x{::}nadd)\ n{::}nat.\ dest\_nadd\ x\ n \neq (0{::}nat) \longrightarrow HOL\_Light\_Import.dist\ (dest\_nadd\ x\ n * nadd\_rinv\ x\ n,\ n * n) \leq dest\_nadd\ x\ n$

thm NADD_MUL_LINV_LEMMA2:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists N{::}nat.\ \forall n{\geq}N.\ HOL\_Light\_Import.dist\ (dest\_nadd\ x\ n * nadd\_rinv\ x\ n,\ n * n) \leq dest\_nadd\ x\ n)$

thm NADD_MUL_LINV_LEMMA3:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists N{::}nat.\ \forall (m{::}nat)\ n{::}nat.\ N \leq n \longrightarrow HOL\_Light\_Import.dist\ (m * (dest\_nadd\ x\ m * (dest\_nadd\ x\ n * nadd\_rinv\ x\ n)),\ m * (dest\_nadd\ x\ m * (n * n))) \leq m * (dest\_nadd\ x\ m * dest\_nadd\ x\ n))$

thm NADD_MUL_LINV_LEMMA4:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists N{::}nat.\ \forall (m{::}nat)\ n{::}nat.\ N \leq m \wedge N \leq n \longrightarrow dest\_nadd\ x\ m * dest\_nadd\ x\ n * HOL\_Light\_Import.dist\ (m * nadd\_rinv\ x\ n,\ n * nadd\_rinv\ x\ m) \leq m * n * HOL\_Light\_Import.dist\ (m * dest\_nadd\ x\ n,\ n * dest\_nadd\ x\ m) + dest\_nadd\ x\ m * dest\_nadd\ x\ n * (m + n))$

thm NADD_MUL_LINV_LEMMA5:

$\forall x{::}nadd.\ \neg\ nadd\_eq\ x\ (nadd\_of\_num\ (0{::}nat)) \longrightarrow (\exists (B{::}nat)\ N{::}nat.\ \forall (m{::}nat)\ n{::}nat.\ N \leq m \wedge N \leq n \longrightarrow dest\_nadd\ x\ m * dest\_nadd\ x\ n * HOL\_Light\_Import.dist\ (m * nadd\_rinv\ x\ n,\ n * nadd\_rinv\ x\ m) \leq B * (m * n * (m + n)))$

thm NADD_MUL_LINV_LEMMA6:

$\forall\, x$::*nadd*. ¬ *nadd_eq* x (*nadd_of_num* (*0*::*nat*)) ⟶ (∃ (*B*::*nat*) *N*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. $N \leq m \wedge N \leq n \longrightarrow m * n *$ *HOL_Light_Import.dist* ($m *$ *nadd_rinv* *x n*, $n *$ *nadd_rinv x m*) $\leq B * (m * n * (m + n))$)

thm NADD_MUL_LINV_LEMMA7:

$\forall\, x$::*nadd*. ¬ *nadd_eq* x (*nadd_of_num* (*0*::*nat*)) ⟶ (∃ (*B*::*nat*) *N*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. $N \leq m \wedge N \leq n \longrightarrow$ *HOL_Light_Import.dist* ($m *$ *nadd_rinv x n*, $n *$ *nadd_rinv x m*) $\leq B * (m + n)$)

thm NADD_MUL_LINV_LEMMA7a:

$\forall\, x$::*nadd*. ¬ *nadd_eq* x (*nadd_of_num* (*0*::*nat*)) ⟶ (∀ *N*::*nat*. ∃ (*A*::*nat*) *B*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. $m \leq N \longrightarrow$ *HOL_Light_Import.dist* ($m *$ *nadd_rinv x n*, $n *$ *nadd_rinv x m*) $\leq A * n + B$)

thm NADD_MUL_LINV_LEMMA8:

$\forall\, x$::*nadd*. ¬ *nadd_eq* x (*nadd_of_num* (*0*::*nat*)) ⟶ (∃ *B*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. *HOL_Light_Import.dist* ($m *$ *nadd_rinv x n*, $n *$ *nadd_rinv x m*) $\leq B * (m + n)$)

thm DEF_nadd_inv:

nadd_inv = (λ*_12795*::*nadd*. *if nadd_eq _12795* (*nadd_of_num* (*0*::*nat*)) *then nadd_of_num* (*0*::*nat*) *else mk_nadd* (*nadd_rinv _12795*))

thm nadd_inv:

$\forall\, x$::*nadd*. *nadd_inv* x = (*if nadd_eq x* (*nadd_of_num* (*0*::*nat*)) *then nadd_of_num* (*0*::*nat*) *else mk_nadd* (*nadd_rinv x*))

thm NADD_INV:

$\forall\, x$::*nadd*. *dest_nadd* (*nadd_inv x*) = (*if nadd_eq x* (*nadd_of_num* (*0*::*nat*)) *then* λ*n*::*nat*. *0*::*nat else nadd_rinv x*)

thm NADD_MUL_LINV:

$\forall\, x$::*nadd*. ¬ *nadd_eq* x (*nadd_of_num* (*0*::*nat*)) ⟶ *nadd_eq* (*nadd_mul* (*nadd_inv x*) *x*) (*nadd_of_num* (*1*::*nat*))

thm NADD_INV_0:

nadd_eq (*nadd_inv* (*nadd_of_num* (*0*::*nat*))) (*nadd_of_num* (*0*::*nat*))

thm NADD_INV_WELLDEF:

∀ (*x*::*nadd*) *y*::*nadd*. *nadd_eq x y* ⟶ *nadd_eq* (*nadd_inv x*) (*nadd_inv y*)

thm TYDEF_hreal:

mk_hreal (*dest_hreal* (*?a*::*hreal*)) = *?a* ∧ (∃ *x*::*nadd*. (*?r*::*nadd* ⇒ *bool*) = *nadd_eq x*) = (*dest_hreal* (*mk_hreal ?r*) = *?r*)

thm DEF_hreal_of_num:

$hreal\_of\_num = (\lambda m::nat.\ mk\_hreal\ (nadd\_eq\ (nadd\_of\_num\ m)))$

thm hreal_of_num:

$hreal\_of\_num\ (?m::nat) = mk\_hreal\ (nadd\_eq\ (nadd\_of\_num\ ?m))$

thm hreal_of_num_th:

$mk\_hreal\ (nadd\_eq\ (nadd\_of\_num\ (?m::nat))) = hreal\_of\_num\ ?m$

thm DEF_hreal_add:

$hreal\_add = (\lambda(x::hreal)\ y::hreal.\ mk\_hreal\ (\lambda u::nadd.\ \exists(xa::nadd)\ ya::nadd.\ nadd\_eq\ (nadd\_add\ xa\ ya)\ u \wedge dest\_hreal\ x\ xa \wedge dest\_hreal\ y\ ya))$

thm hreal_add:

$hreal\_add\ (?x::hreal)\ (?y::hreal) = mk\_hreal\ (\lambda u::nadd.\ \exists(x::nadd)\ y::nadd.\ nadd\_eq\ (nadd\_add\ x\ y)\ u \wedge dest\_hreal\ ?x\ x \wedge dest\_hreal\ ?y\ y)$

thm hreal_add_th:

$mk\_hreal\ (nadd\_eq\ (nadd\_add\ (?x::nadd)\ (?y::nadd))) = hreal\_add\ (mk\_hreal\ (nadd\_eq\ ?x))\ (mk\_hreal\ (nadd\_eq\ ?y))$

thm DEF_hreal_mul:

$hreal\_mul = (\lambda(x::hreal)\ y::hreal.\ mk\_hreal\ (\lambda u::nadd.\ \exists(xa::nadd)\ ya::nadd.\ nadd\_eq\ (nadd\_mul\ xa\ ya)\ u \wedge dest\_hreal\ x\ xa \wedge dest\_hreal\ y\ ya))$

thm hreal_mul:

$hreal\_mul\ (?x::hreal)\ (?y::hreal) = mk\_hreal\ (\lambda u::nadd.\ \exists(x::nadd)\ y::nadd.\ nadd\_eq\ (nadd\_mul\ x\ y)\ u \wedge dest\_hreal\ ?x\ x \wedge dest\_hreal\ ?y\ y)$

thm hreal_mul_th:

$mk\_hreal\ (nadd\_eq\ (nadd\_mul\ (?x::nadd)\ (?y::nadd))) = hreal\_mul\ (mk\_hreal\ (nadd\_eq\ ?x))\ (mk\_hreal\ (nadd\_eq\ ?y))$

thm DEF_hreal_le:

$hreal\_le = (\lambda(x::hreal)\ y::hreal.\ SOME\ u::bool.\ \exists(xa::nadd)\ ya::nadd.\ nadd\_le\ xa\ ya = u \wedge dest\_hreal\ x\ xa \wedge dest\_hreal\ y\ ya)$

thm hreal_le:

$hreal\_le\ (?x::hreal)\ (?y::hreal) = (SOME\ u::bool.\ \exists(x::nadd)\ y::nadd.\ nadd\_le\ x\ y = u \wedge dest\_hreal\ ?x\ x \wedge dest\_hreal\ ?y\ y)$

thm hreal_le_th:

$nadd\_le\ (?x::nadd)\ (?y::nadd) = hreal\_le\ (mk\_hreal\ (nadd\_eq\ ?x))\ (mk\_hreal\ (nadd\_eq\ ?y))$

thm DEF_hreal_inv:

$hreal\_inv = (\lambda x::hreal.\ mk\_hreal\ (\lambda u::nadd.\ \exists xa::nadd.\ nadd\_eq\ (nadd\_inv\ xa)\ u \wedge dest\_hreal\ x\ xa))$

thm hreal_inv:

$hreal\_inv\ (?x::hreal) = mk\_hreal\ (\lambda u::nadd.\ \exists\, x::nadd.\ nadd\_eq\ (nadd\_inv\ x)\ u \wedge dest\_hreal\ ?x\ x)$

thm hreal_inv_th:

$mk\_hreal\ (nadd\_eq\ (nadd\_inv\ (?x::nadd))) = hreal\_inv\ (mk\_hreal\ (nadd\_eq\ ?x))$

thm HREAL_COMPLETE:

$\forall\, P::hreal \Rightarrow bool.\ (\exists\, x::hreal.\ P\ x) \wedge (\exists\, M::hreal.\ \forall\, x::hreal.\ P\ x \longrightarrow hreal\_le\ x\ M) \longrightarrow (\exists\, M::hreal.\ (\forall\, x::hreal.\ P\ x \longrightarrow hreal\_le\ x\ M) \wedge (\forall\, M'::hreal.\ (\forall\, x::hreal.\ P\ x \longrightarrow hreal\_le\ x\ M') \longrightarrow hreal\_le\ M\ M'))$

thm HREAL_OF_NUM_EQ:

$\forall\, (m::nat)\ n::nat.\ (hreal\_of\_num\ m = hreal\_of\_num\ n) = (m = n)$

thm HREAL_OF_NUM_LE:

$\forall\, (m::nat)\ n::nat.\ hreal\_le\ (hreal\_of\_num\ m)\ (hreal\_of\_num\ n) = (m \leq n)$

thm HREAL_OF_NUM_ADD:

$\forall\, (m::nat)\ n::nat.\ hreal\_add\ (hreal\_of\_num\ m)\ (hreal\_of\_num\ n) = hreal\_of\_num\ (m + n)$

thm HREAL_OF_NUM_MUL:

$\forall\, (m::nat)\ n::nat.\ hreal\_mul\ (hreal\_of\_num\ m)\ (hreal\_of\_num\ n) = hreal\_of\_num\ (m * n)$

thm HREAL_LE_REFL:

$\forall\, x::hreal.\ hreal\_le\ x\ x$

thm HREAL_LE_TRANS:

$\forall\, (x::hreal)\ (y::hreal)\ z::hreal.\ hreal\_le\ x\ y \wedge hreal\_le\ y\ z \longrightarrow hreal\_le\ x\ z$

thm HREAL_LE_ANTISYM:

$\forall\, (x::hreal)\ y::hreal.\ (hreal\_le\ x\ y \wedge hreal\_le\ y\ x) = (x = y)$

thm HREAL_LE_TOTAL:

$\forall\, (x::hreal)\ y::hreal.\ hreal\_le\ x\ y \vee hreal\_le\ y\ x$

thm HREAL_LE_ADD:

$\forall\, (x::hreal)\ y::hreal.\ hreal\_le\ x\ (hreal\_add\ x\ y)$

thm HREAL_LE_EXISTS:

$\forall\, (x::hreal)\ y::hreal.\ hreal\_le\ x\ y \longrightarrow (\exists\, d::hreal.\ y = hreal\_add\ x\ d)$

thm HREAL_ARCH:

$\forall x::hreal.\ \exists n::nat.\ hreal\_le\ x\ (hreal\_of\_num\ n)$

thm HREAL_ADD_SYM:

$\forall (x::hreal)\ y::hreal.\ hreal\_add\ x\ y = hreal\_add\ y\ x$

thm HREAL_ADD_ASSOC:

$\forall (x::hreal)\ (y::hreal)\ z::hreal.\ hreal\_add\ x\ (hreal\_add\ y\ z) = hreal\_add\ (hreal\_add\ x\ y)\ z$

thm HREAL_ADD_LID:

$\forall x::hreal.\ hreal\_add\ (hreal\_of\_num\ (0::nat))\ x = x$

thm HREAL_ADD_LCANCEL:

$\forall (x::hreal)\ (y::hreal)\ z::hreal.\ hreal\_add\ x\ y = hreal\_add\ x\ z \longrightarrow y = z$

thm HREAL_MUL_SYM:

$\forall (x::hreal)\ y::hreal.\ hreal\_mul\ x\ y = hreal\_mul\ y\ x$

thm HREAL_MUL_ASSOC:

$\forall (x::hreal)\ (y::hreal)\ z::hreal.\ hreal\_mul\ x\ (hreal\_mul\ y\ z) = hreal\_mul\ (hreal\_mul\ x\ y)\ z$

thm HREAL_MUL_LID:

$\forall x::hreal.\ hreal\_mul\ (hreal\_of\_num\ (1::nat))\ x = x$

thm HREAL_ADD_LDISTRIB:

$\forall (x::hreal)\ (y::hreal)\ z::hreal.\ hreal\_mul\ x\ (hreal\_add\ y\ z) = hreal\_add\ (hreal\_mul\ x\ y)\ (hreal\_mul\ x\ z)$

thm HREAL_MUL_LINV:

$\forall x::hreal.\ x \neq hreal\_of\_num\ (0::nat) \longrightarrow hreal\_mul\ (hreal\_inv\ x)\ x = hreal\_of\_num\ (1::nat)$

thm HREAL_INV_0:

$hreal\_inv\ (hreal\_of\_num\ (0::nat)) = hreal\_of\_num\ (0::nat)$

thm HREAL_LE_EXISTS_DEF:

$\forall (m::hreal)\ n::hreal.\ hreal\_le\ m\ n = (\exists\ d::hreal.\ n = hreal\_add\ m\ d)$

thm HREAL_EQ_ADD_LCANCEL:

$\forall (m::hreal)\ (n::hreal)\ p::hreal.\ (hreal\_add\ m\ n = hreal\_add\ m\ p) = (n = p)$

thm HREAL_EQ_ADD_RCANCEL:

$\forall (m::hreal)\ (n::hreal)\ p::hreal.\ (hreal\_add\ m\ p = hreal\_add\ n\ p) = (m = n)$

thm HREAL_LE_ADD_LCANCEL:

$\forall (m::hreal)\ (n::hreal)\ p::hreal.\ hreal\_le\ (hreal\_add\ m\ n)\ (hreal\_add\ m\ p) = hreal\_le\ n\ p$

thm HREAL_LE_ADD_RCANCEL:

$\forall (m::hreal)\ (n::hreal)\ p::hreal.\ hreal\_le\ (hreal\_add\ m\ p)\ (hreal\_add\ n\ p) = hreal\_le\ m\ n$

thm HREAL_ADD_RID:

$\forall n::hreal.\ hreal\_add\ n\ (hreal\_of\_num\ (0::nat)) = n$

thm HREAL_ADD_RDISTRIB:

$\forall (m::hreal)\ (n::hreal)\ p::hreal.\ hreal\_mul\ (hreal\_add\ m\ n)\ p = hreal\_add\ (hreal\_mul\ m\ p)\ (hreal\_mul\ n\ p)$

thm HREAL_MUL_LZERO:

$\forall m::hreal.\ hreal\_mul\ (hreal\_of\_num\ (0::nat))\ m = hreal\_of\_num\ (0::nat)$

thm HREAL_MUL_RZERO:

$\forall m::hreal.\ hreal\_mul\ m\ (hreal\_of\_num\ (0::nat)) = hreal\_of\_num\ (0::nat)$

thm HREAL_ADD_AC_conjunct0:

$hreal\_add\ (?m::hreal)\ (?n::hreal) = hreal\_add\ ?n\ ?m$

thm HREAL_ADD_AC:

$hreal\_add\ (?m::hreal)\ (?n::hreal) = hreal\_add\ ?n\ ?m \wedge hreal\_add\ (hreal\_add\ ?m\ ?n)\ (?p::hreal) = hreal\_add\ ?m\ (hreal\_add\ ?n\ ?p) \wedge hreal\_add\ ?m\ (hreal\_add\ ?n\ ?p) = hreal\_add\ ?n\ (hreal\_add\ ?m\ ?p)$

thm HREAL_ADD_AC_conjunct2:

$hreal\_add\ (?m::hreal)\ (hreal\_add\ (?n::hreal)\ (?p::hreal)) = hreal\_add\ ?n\ (hreal\_add\ ?m\ ?p)$

thm HREAL_ADD_AC_conjunct1:

$hreal\_add\ (hreal\_add\ (?m::hreal)\ (?n::hreal))\ (?p::hreal) = hreal\_add\ ?m\ (hreal\_add\ ?n\ ?p)$

thm HREAL_LE_ADD2:

$\forall (a::hreal)\ (b::hreal)\ (c::hreal)\ d::hreal.\ hreal\_le\ a\ b \wedge hreal\_le\ c\ d \longrightarrow hreal\_le\ (hreal\_add\ a\ c)\ (hreal\_add\ b\ d)$

thm HREAL_LE_MUL_RCANCEL_IMP:

$\forall (a::hreal)\ (b::hreal)\ c::hreal.\ hreal\_le\ a\ b \longrightarrow hreal\_le\ (hreal\_mul\ a\ c)\ (hreal\_mul\ b\ c)$

thm DEF_treal_of_num:

$treal\_of\_num = (\lambda\_13040::nat.\ (hreal\_of\_num\ \_13040,\ hreal\_of\_num\ (0::nat)))$

thm treal_of_num:

$\forall\, n{::}nat.\ treal\_of\_num\ n = (hreal\_of\_num\ n,\ hreal\_of\_num\ (0{::}nat))$

thm DEF_treal_neg:

$treal\_neg = (\lambda\_13045{::}hreal \times hreal.\ (snd\ \_13045,\ fst\ \_13045))$

thm treal_neg:

$\forall\, (y{::}hreal)\ x{::}hreal.\ treal\_neg\ (x,\ y) = (y,\ x)$

thm DEF_treal_add:

$treal\_add = (\lambda(\_13054{::}hreal \times hreal)\ \_13055{::}hreal \times hreal.\ (hreal\_add\ (fst\ \_13054)\ (fst\ \_13055),\ hreal\_add\ (snd\ \_13054)\ (snd\ \_13055)))$

thm treal_add:

$\forall\, (x1{::}hreal)\ (x2{::}hreal)\ (y1{::}hreal)\ y2{::}hreal.\ treal\_add\ (x1,\ y1)\ (x2,\ y2) = (hreal\_add\ x1\ x2,\ hreal\_add\ y1\ y2)$

thm DEF_treal_mul:

$treal\_mul = (\lambda(\_13076{::}hreal \times hreal)\ \_13077{::}hreal \times hreal.\ (hreal\_add\ (hreal\_mul\ (fst\ \_13076)\ (fst\ \_13077))\ (hreal\_mul\ (snd\ \_13076)\ (snd\ \_13077)),\ hreal\_add\ (hreal\_mul\ (fst\ \_13076)\ (snd\ \_13077))\ (hreal\_mul\ (snd\ \_13076)\ (fst\ \_13077))))$

thm treal_mul:

$\forall\, (x1{::}hreal)\ (y2{::}hreal)\ (y1{::}hreal)\ x2{::}hreal.\ treal\_mul\ (x1,\ y1)\ (x2,\ y2) = (hreal\_add\ (hreal\_mul\ x1\ x2)\ (hreal\_mul\ y1\ y2),\ hreal\_add\ (hreal\_mul\ x1\ y2)\ (hreal\_mul\ y1\ x2))$

thm DEF_treal_le:

$treal\_le = (\lambda(\_13098{::}hreal \times hreal)\ \_13099{::}hreal \times hreal.\ hreal\_le\ (hreal\_add\ (fst\ \_13098)\ (snd\ \_13099))\ (hreal\_add\ (fst\ \_13099)\ (snd\ \_13098)))$

thm treal_le:

$\forall\, (x1{::}hreal)\ (y2{::}hreal)\ (x2{::}hreal)\ y1{::}hreal.\ treal\_le\ (x1,\ y1)\ (x2,\ y2) = hreal\_le\ (hreal\_add\ x1\ y2)\ (hreal\_add\ x2\ y1)$

thm DEF_treal_inv:

$treal\_inv = (\lambda\_13120{::}hreal \times hreal.\ if\ fst\ \_13120 = snd\ \_13120\ then\ (hreal\_of\_num\ (0{::}nat),\ hreal\_of\_num\ (0{::}nat))\ else\ if\ hreal\_le\ (snd\ \_13120)\ (fst\ \_13120)\ then\ (hreal\_inv\ (SOME\ d{::}hreal.\ fst\ \_13120 = hreal\_add\ (snd\ \_13120)\ d),\ hreal\_of\_num\ (0{::}nat))\ else\ (hreal\_of\_num\ (0{::}nat),\ hreal\_inv\ (SOME\ d{::}hreal.\ snd\ \_13120 = hreal\_add\ (fst\ \_13120)\ d)))$

thm treal_inv:

$\forall\, (y{::}hreal)\ x{::}hreal.\ treal\_inv\ (x,\ y) = (if\ x = y\ then\ (hreal\_of\_num\ (0{::}nat),\ hreal\_of\_num\ (0{::}nat))\ else\ if\ hreal\_le\ y\ x\ then\ (hreal\_inv\ (SOME\ d{::}hreal.\ x =$

$hreal\_add\ y\ d$), $hreal\_of\_num\ (0::nat)$) $else$ ($hreal\_of\_num\ (0::nat)$, $hreal\_inv$ ($SOME\ d::hreal.\ y = hreal\_add\ x\ d$)))

thm DEF_treal_eq:

$treal\_eq = (\lambda(\_13129::hreal \times hreal)\ \_13130::hreal \times hreal.\ hreal\_add\ (fst\ \_13129)\ (snd\ \_13130) = hreal\_add\ (fst\ \_13130)\ (snd\ \_13129))$

thm treal_eq:

$\forall (x1::hreal)\ (y2::hreal)\ (x2::hreal)\ y1::hreal.\ treal\_eq\ (x1, y1)\ (x2, y2) = (hreal\_add\ x1\ y2 = hreal\_add\ x2\ y1)$

thm TREAL_EQ_REFL:

$\forall x::hreal \times hreal.\ treal\_eq\ x\ x$

thm TREAL_EQ_SYM:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ x\ y = treal\_eq\ y\ x$

thm TREAL_EQ_TRANS:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_eq\ x\ y \wedge treal\_eq\ y\ z \longrightarrow treal\_eq\ x\ z$

thm TREAL_EQ_AP:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ x = y \longrightarrow treal\_eq\ x\ y$

thm TREAL_OF_NUM_EQ:

$\forall (m::nat)\ n::nat.\ treal\_eq\ (treal\_of\_num\ m)\ (treal\_of\_num\ n) = (m = n)$

thm TREAL_OF_NUM_LE:

$\forall (m::nat)\ n::nat.\ treal\_le\ (treal\_of\_num\ m)\ (treal\_of\_num\ n) = (m \leq n)$

thm TREAL_OF_NUM_ADD:

$\forall (m::nat)\ n::nat.\ treal\_eq\ (treal\_add\ (treal\_of\_num\ m)\ (treal\_of\_num\ n))\ (treal\_of\_num\ (m + n))$

thm TREAL_OF_NUM_MUL:

$\forall (m::nat)\ n::nat.\ treal\_eq\ (treal\_mul\ (treal\_of\_num\ m)\ (treal\_of\_num\ n))\ (treal\_of\_num\ (m * n))$

thm TREAL_ADD_SYM_EQ:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_add\ x\ y = treal\_add\ y\ x$

thm TREAL_MUL_SYM_EQ:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_mul\ x\ y = treal\_mul\ y\ x$

thm TREAL_ADD_SYM:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ (treal\_add\ x\ y)\ (treal\_add\ y\ x)$

thm TREAL_ADD_ASSOC:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_eq\ (treal\_add\ x\ (treal\_add\ y\ z))\ (treal\_add\ (treal\_add\ x\ y)\ z)$

thm TREAL_ADD_LID:

$\forall x::hreal \times hreal.\ treal\_eq\ (treal\_add\ (treal\_of\_num\ (0::nat))\ x)\ x$

thm TREAL_ADD_LINV:

$\forall x::hreal \times hreal.\ treal\_eq\ (treal\_add\ (treal\_neg\ x)\ x)\ (treal\_of\_num\ (0::nat))$

thm TREAL_MUL_SYM:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ (treal\_mul\ x\ y)\ (treal\_mul\ y\ x)$

thm TREAL_MUL_ASSOC:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_eq\ (treal\_mul\ x\ (treal\_mul\ y\ z))\ (treal\_mul\ (treal\_mul\ x\ y)\ z)$

thm TREAL_MUL_LID:

$\forall x::hreal \times hreal.\ treal\_eq\ (treal\_mul\ (treal\_of\_num\ (1::nat))\ x)\ x$

thm TREAL_ADD_LDISTRIB:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_eq\ (treal\_mul\ x\ (treal\_add\ y\ z))\ (treal\_add\ (treal\_mul\ x\ y)\ (treal\_mul\ x\ z))$

thm TREAL_LE_REFL:

$\forall x::hreal \times hreal.\ treal\_le\ x\ x$

thm TREAL_LE_ANTISYM:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ (treal\_le\ x\ y \wedge treal\_le\ y\ x) = treal\_eq\ x\ y$

thm TREAL_LE_TRANS:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_le\ x\ y \wedge treal\_le\ y\ z \longrightarrow treal\_le\ x\ z$

thm TREAL_LE_TOTAL:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_le\ x\ y \vee treal\_le\ y\ x$

thm TREAL_LE_LADD_IMP:

$\forall (x::hreal \times hreal)\ (y::hreal \times hreal)\ z::hreal \times hreal.\ treal\_le\ y\ z \longrightarrow treal\_le\ (treal\_add\ x\ y)\ (treal\_add\ x\ z)$

thm TREAL_LE_MUL:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_le\ (treal\_of\_num\ (0::nat))\ x \wedge treal\_le\ (treal\_of\_num\ (0::nat))\ y \longrightarrow treal\_le\ (treal\_of\_num\ (0::nat))\ (treal\_mul\ x\ y)$

thm TREAL_INV_0:

$treal\_eq\ (treal\_inv\ (treal\_of\_num\ (0::nat)))\ (treal\_of\_num\ (0::nat))$

thm TREAL_MUL_LINV:

$\forall x::hreal \times hreal.\ \neg\ treal\_eq\ x\ (treal\_of\_num\ (0::nat)) \longrightarrow treal\_eq\ (treal\_mul\ (treal\_inv\ x)\ x)\ (treal\_of\_num\ (1::nat))$

thm TREAL_OF_NUM_WELLDEF:

$\forall (m::nat)\ n::nat.\ m = n \longrightarrow treal\_eq\ (treal\_of\_num\ m)\ (treal\_of\_num\ n)$

thm TREAL_NEG_WELLDEF:

$\forall (x1::hreal \times hreal)\ x2::hreal \times hreal.\ treal\_eq\ x1\ x2 \longrightarrow treal\_eq\ (treal\_neg\ x1)\ (treal\_neg\ x2)$

thm TREAL_ADD_WELLDEFR:

$\forall (x1::hreal \times hreal)\ (x2::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ x1\ x2 \longrightarrow treal\_eq\ (treal\_add\ x1\ y)\ (treal\_add\ x2\ y)$

thm TREAL_ADD_WELLDEF:

$\forall (x1::hreal \times hreal)\ (x2::hreal \times hreal)\ (y1::hreal \times hreal)\ y2::hreal \times hreal.\ treal\_eq\ x1\ x2 \wedge treal\_eq\ y1\ y2 \longrightarrow treal\_eq\ (treal\_add\ x1\ y1)\ (treal\_add\ x2\ y2)$

thm TREAL_MUL_WELLDEFR:

$\forall (x1::hreal \times hreal)\ (x2::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ x1\ x2 \longrightarrow treal\_eq\ (treal\_mul\ x1\ y)\ (treal\_mul\ x2\ y)$

thm TREAL_MUL_WELLDEF:

$\forall (x1::hreal \times hreal)\ (x2::hreal \times hreal)\ (y1::hreal \times hreal)\ y2::hreal \times hreal.\ treal\_eq\ x1\ x2 \wedge treal\_eq\ y1\ y2 \longrightarrow treal\_eq\ (treal\_mul\ x1\ y1)\ (treal\_mul\ x2\ y2)$

thm TREAL_EQ_IMP_LE:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ x\ y \longrightarrow treal\_le\ x\ y$

thm TREAL_LE_WELLDEF:

$\forall (x1::hreal \times hreal)\ (x2::hreal \times hreal)\ (y1::hreal \times hreal)\ y2::hreal \times hreal.\ treal\_eq\ x1\ x2 \wedge treal\_eq\ y1\ y2 \longrightarrow treal\_le\ x1\ y1 = treal\_le\ x2\ y2$

thm TREAL_INV_WELLDEF:

$\forall (x::hreal \times hreal)\ y::hreal \times hreal.\ treal\_eq\ x\ y \longrightarrow treal\_eq\ (treal\_inv\ x)\ (treal\_inv\ y)$

thm REAL_ADD_SYM:

$\forall (x::real)\ y::real.\ x + y = y + x$

thm REAL_ADD_ASSOC:

$\forall (x::real)\ (y::real)\ z::real.\ x + (y + z) = x + y + z$

thm REAL_ADD_LID:

$\forall\, x{::}real.\ (0{::}real) + x = x$

thm REAL_ADD_LINV:

$\forall\, x{::}real.\ -x + x = (0{::}real)$

thm REAL_MUL_SYM:

$\forall\, (x{::}real)\ y{::}real.\ x * y = y * x$

thm REAL_MUL_ASSOC:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ x * (y * z) = x * y * z$

thm REAL_MUL_LID:

$\forall\, x{::}real.\ (1{::}real) * x = x$

thm REAL_ADD_LDISTRIB:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ x * (y + z) = x * y + x * z$

thm REAL_LE_REFL:

$\forall\, x{::}real.\ x \leq x$

thm REAL_LE_ANTISYM:

$\forall\, (x{::}real)\ y{::}real.\ (x \leq y \wedge y \leq x) = (x = y)$

thm REAL_LE_TRANS:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ x \leq y \wedge y \leq z \longrightarrow x \leq z$

thm REAL_LE_TOTAL:

$\forall\, (x{::}real)\ y{::}real.\ x \leq y \vee y \leq x$

thm REAL_LE_LADD_IMP:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ y \leq z \longrightarrow x + y \leq x + z$

thm REAL_LE_MUL:

$\forall\, (x{::}real)\ y{::}real.\ (0{::}real) \leq x \wedge (0{::}real) \leq y \longrightarrow (0{::}real) \leq x * y$

thm REAL_INV_0:

$inverse\ (0{::}real) = (0{::}real)$

thm REAL_MUL_LINV:

$\forall\, x{::}real.\ x \neq (0{::}real) \longrightarrow inverse\ x * x = (1{::}real)$

thm REAL_OF_NUM_EQ:

$\forall\, (m{::}nat)\ n{::}nat.\ (real\_of\_nat\ m = real\_of\_nat\ n) = (m = n)$

thm REAL_OF_NUM_LE:

$\forall (m::nat)\ n::nat.\ (real\_of\_nat\ m \leq real\_of\_nat\ n) = (m \leq n)$

thm REAL_OF_NUM_ADD:

$\forall (m::nat)\ n::nat.\ real\_of\_nat\ m + real\_of\_nat\ n = real\_of\_nat\ (m + n)$

thm REAL_OF_NUM_MUL:

$\forall (m::nat)\ n::nat.\ real\_of\_nat\ m * real\_of\_nat\ n = real\_of\_nat\ (m * n)$

thm real_sub:

$\forall (x::real)\ y::real.\ x - y = x + - y$

thm real_lt:

$\forall (y::real)\ x::real.\ (x < y) = (\neg\ y \leq x)$

thm real_ge:

$\forall (y::real)\ x::real.\ (y \leq x) = (y \leq x)$

thm real_gt:

$\forall (y::real)\ x::real.\ (y < x) = (y < x)$

thm real_abs:

$\forall x::real.\ |x| = (if\ (0::real) \leq x\ then\ x\ else\ - x)$

thm REAL_POLY_CLAUSES_conjunct9:

$\forall (x::real)\ n::nat.\ x^{Suc\ n} = x * x^n$

thm REAL_POLY_CLAUSES_conjunct8:

$\forall x::real.\ x^{0::nat} = (1::real)$

thm real_pow_conjunct0:

$(?x::real)^{0::nat} = (1::real)$

thm real_pow_conjunct1:

$\forall n::nat.\ (?x::real)^{Suc\ n} = ?x * ?x^n$

thm real_pow:

$(?x::real)^{0::nat} = (1::real) \wedge (\forall n::nat.\ ?x^{Suc\ n} = ?x * ?x^n)$

thm real_div:

$\forall (x::real)\ y::real.\ x / y = x * inverse\ y$

thm real_max:

$\forall (n::real)\ m::real.\ max\ m\ n = (if\ m \leq n\ then\ n\ else\ m)$

thm real_min:

$\forall (m::real)\ n::real.\ min\ m\ n = (if\ m \leq n\ then\ m\ else\ n)$

thm REAL_COMPLETE_SOMEPOS:

$\forall P{::}real \Rightarrow bool.\ (\exists x{::}real.\ P\ x \wedge (0{::}real) \leq x) \wedge (\exists M{::}real.\ \forall x{::}real.\ P\ x \longrightarrow x \leq M) \longrightarrow (\exists M{::}real.\ (\forall x{::}real.\ P\ x \longrightarrow x \leq M) \wedge (\forall M'{::}real.\ (\forall x{::}real.\ P\ x \longrightarrow x \leq M') \longrightarrow M \leq M'))$

thm REAL_ADD_RID:

$\forall x{::}real.\ x + (0{::}real) = x$

thm REAL_ADD_RINV:

$\forall x{::}real.\ x + - x = (0{::}real)$

thm REAL_COMPLETE:

$\forall P{::}real \Rightarrow bool.\ (\exists x{::}real.\ P\ x) \wedge (\exists M{::}real.\ \forall x{::}real.\ P\ x \longrightarrow x \leq M) \longrightarrow (\exists M{::}real.\ (\forall x{::}real.\ P\ x \longrightarrow x \leq M) \wedge (\forall M'{::}real.\ (\forall x{::}real.\ P\ x \longrightarrow x \leq M') \longrightarrow M \leq M'))$

thm REAL_ADD_AC:

$(?m{::}real) + (?n{::}real) = ?n + ?m \wedge ?m + ?n + (?p{::}real) = ?m + (?n + ?p) \wedge ?m + (?n + ?p) = ?n + (?m + ?p)$

thm REAL_EQ_ADD_LCANCEL:

$\forall (x{::}real)\ (y{::}real)\ z{::}real.\ (x + y = x + z) = (y = z)$

thm REAL_EQ_ADD_RCANCEL:

$\forall (x{::}real)\ (y{::}real)\ z{::}real.\ (x + z = y + z) = (x = y)$

thm REAL_MUL_RZERO:

$\forall x{::}real.\ x * (0{::}real) = (0{::}real)$

thm REAL_MUL_LZERO:

$\forall x{::}real.\ (0{::}real) * x = (0{::}real)$

thm REAL_NEGNEG:

$\forall x{::}real.\ - (- x) = x$

thm REAL_MUL_RNEG:

$\forall (x{::}real)\ y{::}real.\ x * - y = - (x * y)$

thm REAL_MUL_LNEG:

$\forall (x{::}real)\ y{::}real.\ - x * y = - (x * y)$

thm REAL_ADD_AC_conjunct2:

$(?m{::}real) + ((?n{::}real) + (?p{::}real)) = ?n + (?m + ?p)$

thm REAL_ADD_AC_conjunct1:

$(?m{::}real) + (?n{::}real) + (?p{::}real) = ?m + (?n + ?p)$

thm REAL_ADD_AC_conjunct0:

$(?m::real) + (?n::real) = ?n + ?m$

thm REAL_NEG_ADD:

$\forall (x::real)\ y::real.\ -\ (x + y) = -\ x + -\ y$

thm REAL_NEG_0:

$-\ (0::real) = (0::real)$

thm REAL_LE_LNEG:

$\forall (x::real)\ y::real.\ (-\ x \le y) = ((0::real) \le x + y)$

thm REAL_LE_NEG2:

$\forall (x::real)\ y::real.\ (-\ x \le -\ y) = (y \le x)$

thm REAL_LE_RNEG:

$\forall (x::real)\ y::real.\ (x \le -\ y) = (x + y \le (0::real))$

thm REAL_OF_NUM_POW:

$\forall (x::nat)\ n::nat.\ (real\_of\_nat\ x)^n = real\_of\_nat\ x^n$

thm REAL_POW_NEG:

$\forall (x::real)\ n::nat.\ (-\ x)^n = (if\ even\ n\ then\ x^n\ else\ -\ x^n)$

thm REAL_ABS_NUM:

$\forall n::nat.\ |real\_of\_nat\ n| = real\_of\_nat\ n$

thm REAL_ABS_NEG:

$\forall x::real.\ |-\ x| = |x|$

thm REAL_LTE_TOTAL:

$\forall (x::real)\ y::real.\ x < y \vee y \le x$

thm REAL_LET_TOTAL:

$\forall (x::real)\ y::real.\ x \le y \vee y < x$

thm REAL_LT_IMP_LE:

$\forall (x::real)\ y::real.\ x < y \longrightarrow x \le y$

thm REAL_LTE_TRANS:

$\forall (x::real)\ (y::real)\ z::real.\ x < y \wedge y \le z \longrightarrow x < z$

thm REAL_LET_TRANS:

$\forall (x::real)\ (y::real)\ z::real.\ x \le y \wedge y < z \longrightarrow x < z$

thm REAL_LT_TRANS:

$\forall (x::real)\ (y::real)\ z::real.\ x < y \wedge y < z \longrightarrow x < z$

thm REAL_LE_ADD:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (0::real) \leq x + y$

thm REAL_LTE_ANTISYM:

$\forall (x::real)\ y::real.\ \neg\ (x < y \wedge y \leq x)$

thm REAL_SUB_LE:

$\forall (x::real)\ y::real.\ ((0::real) \leq x - y) = (y \leq x)$

thm REAL_NEG_SUB:

$\forall (x::real)\ y::real.\ -\ (x - y) = y - x$

thm REAL_LE_LT:

$\forall (x::real)\ y::real.\ (x \leq y) = (x < y \vee x = y)$

thm REAL_SUB_LT:

$\forall (x::real)\ y::real.\ ((0::real) < x - y) = (y < x)$

thm REAL_NOT_LT:

$\forall (x::real)\ y::real.\ (\neg\ x < y) = (y \leq x)$

thm REAL_SUB_0:

$\forall (x::real)\ y::real.\ (x - y = (0::real)) = (x = y)$

thm REAL_LT_LE:

$\forall (x::real)\ y::real.\ (x < y) = (x \leq y \wedge x \neq y)$

thm REAL_LT_REFL:

$\forall x::real.\ \neg\ x < x$

thm REAL_LTE_ADD:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) \leq y \longrightarrow (0::real) < x + y$

thm REAL_LET_ADD:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) < y \longrightarrow (0::real) < x + y$

thm REAL_LT_ADD:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow (0::real) < x + y$

thm REAL_ENTIRE:

$\forall (x::real)\ y::real.\ (x * y = (0::real)) = (x = (0::real) \vee y = (0::real))$

thm REAL_LE_NEGTOTAL:

$\forall x::real.\ (0::real) \leq x \vee (0::real) \leq -\ x$

thm REAL_LE_SQUARE:

$\forall x{::}real.\ (0{::}real) \leq x * x$

thm REAL_MUL_RID:

$\forall x{::}real.\ x * (1{::}real) = x$

thm REAL_POW_2:

$\forall x{::}real.\ x^2 = x * x$

thm REAL_POLY_CLAUSES:

$(\forall (x{::}real)\ (y{::}real)\ z{::}real.\ x + (y + z) = x + y + z) \wedge (\forall (x{::}real)\ y{::}real.\ x + y = y + x) \wedge (\forall x{::}real.\ (0{::}real) + x = x) \wedge (\forall (x{::}real)\ (y{::}real)\ z{::}real.\ x * (y * z) = x * y * z) \wedge (\forall (x{::}real)\ y{::}real.\ x * y = y * x) \wedge (\forall x{::}real.\ (1{::}real) * x = x) \wedge (\forall x{::}real.\ (0{::}real) * x = (0{::}real)) \wedge (\forall (x{::}real)\ (y{::}real)\ z{::}real.\ x * (y + z) = x * y + x * z) \wedge (\forall x{::}real.\ x^{0{::}nat} = (1{::}real)) \wedge (\forall (x{::}real)\ n{::}nat.\ x^{Suc\ n} = x * x^n)$

thm REAL_POLY_NEG_CLAUSES:

$(\forall x{::}real.\ - x = - (1{::}real) * x) \wedge (\forall (x{::}real)\ y{::}real.\ x - y = x + - (1{::}real) * y)$

thm REAL_POS:

$\forall n{::}nat.\ (0{::}real) \leq real\_of\_nat\ n$

thm REAL_NEG_MINUS1:

$\forall x{::}real.\ - x = - (1{::}real) * x$

thm REAL_POLY_NEG_CLAUSES_conjunct1:

$\forall (x{::}real)\ y{::}real.\ x - y = x + - (1{::}real) * y$

thm REAL_OF_NUM_LT:

$\forall (m{::}nat)\ n{::}nat.\ (real\_of\_nat\ m < real\_of\_nat\ n) = (m < n)$

thm REAL_OF_NUM_GE:

$\forall (m{::}nat)\ n{::}nat.\ (real\_of\_nat\ n \leq real\_of\_nat\ m) = (n \leq m)$

thm REAL_OF_NUM_GT:

$\forall (m{::}nat)\ n{::}nat.\ (real\_of\_nat\ n < real\_of\_nat\ m) = (n < m)$

thm REAL_OF_NUM_MAX:

$\forall (m{::}nat)\ n{::}nat.\ max\ (real\_of\_nat\ m)\ (real\_of\_nat\ n) = real\_of\_nat\ (max\ m\ n)$

thm REAL_OF_NUM_MIN:

$\forall (m{::}nat)\ n{::}nat.\ min\ (real\_of\_nat\ m)\ (real\_of\_nat\ n) = real\_of\_nat\ (min\ m\ n)$

thm REAL_OF_NUM_SUC:

$\forall\, n{::}nat.\ real\_of\_nat\ n + (1{::}real) = real\_of\_nat\ (Suc\ n)$

thm REAL_OF_NUM_SUB:

$\forall\, (m{::}nat)\ n{::}nat.\ m \le n \longrightarrow real\_of\_nat\ n - real\_of\_nat\ m = real\_of\_nat\ (n - m)$

thm REAL_MUL_AC_conjunct0:

$(?m{::}real) * (?n{::}real) = ?n * ?m$

thm REAL_MUL_AC:

$(?m{::}real) * (?n{::}real) = ?n * ?m \wedge ?m * ?n * (?p{::}real) = ?m * (?n * ?p) \wedge ?m * (?n * ?p) = ?n * (?m * ?p)$

thm REAL_ADD_RDISTRIB:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ (x + y) * z = x * z + y * z$

thm REAL_LT_LADD_IMP:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ y < z \longrightarrow x + y < x + z$

thm REAL_LT_MUL:

$\forall\, (x{::}real)\ y{::}real.\ (0{::}real) < x \wedge (0{::}real) < y \longrightarrow (0{::}real) < x * y$

thm REAL_EQ_ADD_LCANCEL_0:

$\forall\, (x{::}real)\ y{::}real.\ (x + y = x) = (y = (0{::}real))$

thm REAL_EQ_ADD_RCANCEL_0:

$\forall\, (x{::}real)\ y{::}real.\ (x + y = y) = (x = (0{::}real))$

thm REAL_LNEG_UNIQ:

$\forall\, (x{::}real)\ y{::}real.\ (x + y = (0{::}real)) = (x = -\ y)$

thm REAL_RNEG_UNIQ:

$\forall\, (x{::}real)\ y{::}real.\ (x + y = (0{::}real)) = (y = -\ x)$

thm REAL_NEG_LMUL:

$\forall\, (x{::}real)\ y{::}real.\ -\ (x * y) = -\ x * y$

thm REAL_NEG_RMUL:

$\forall\, (x{::}real)\ y{::}real.\ -\ (x * y) = x * -\ y$

thm REAL_NEG_MUL2:

$\forall\, (x{::}real)\ y{::}real.\ -\ x * -\ y = x * y$

thm REAL_LT_LADD:

$\forall\, (x{::}real)\ (y{::}real)\ z{::}real.\ (x + y < x + z) = (y < z)$

thm REAL_LT_RADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x + z < y + z) = (x < y)$

thm REAL_LT_ANTISYM:

$\forall (x::real)\ y::real.\ \neg\ (x < y \wedge y < x)$

thm REAL_LT_GT:

$\forall (x::real)\ y::real.\ x < y \longrightarrow \neg\ y < x$

thm REAL_NOT_EQ:

$\forall (x::real)\ y::real.\ (x \neq y) = (x < y \vee y < x)$

thm REAL_NOT_LE:

$\forall (x::real)\ y::real.\ (\neg\ x \leq y) = (y < x)$

thm REAL_LET_ANTISYM:

$\forall (x::real)\ y::real.\ \neg\ (x \leq y \wedge y < x)$

thm REAL_NEG_LT0:

$\forall x::real.\ (-\ x < (0::real)) = ((0::real) < x)$

thm REAL_NEG_GT0:

$\forall x::real.\ ((0::real) < -\ x) = (x < (0::real))$

thm REAL_NEG_LE0:

$\forall x::real.\ (-\ x \leq (0::real)) = ((0::real) \leq x)$

thm REAL_NEG_GE0:

$\forall x::real.\ ((0::real) \leq -\ x) = (x \leq (0::real))$

thm REAL_LT_TOTAL:

$\forall (x::real)\ y::real.\ x = y \vee x < y \vee y < x$

thm REAL_LT_NEGTOTAL:

$\forall x::real.\ x = (0::real) \vee (0::real) < x \vee (0::real) < -\ x$

thm REAL_LE_01:

$(0::real) \leq (1::real)$

thm REAL_LT_01:

$(0::real) < (1::real)$

thm REAL_LE_LADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x + y \leq x + z) = (y \leq z)$

thm REAL_LE_RADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x + z \leq y + z) = (x \leq y)$

thm REAL_LT_ADD2:

$\forall (w::real)\ (x::real)\ (y::real)\ z::real.\ w < x \wedge y < z \longrightarrow w + y < x + z$

thm REAL_LE_ADD2:

$\forall (w::real)\ (x::real)\ (y::real)\ z::real.\ w \leq x \wedge y \leq z \longrightarrow w + y \leq x + z$

thm REAL_LT_LNEG:

$\forall (x::real)\ y::real.\ (-\ x < y) = ((0::real) < x + y)$

thm REAL_LT_RNEG:

$\forall (x::real)\ y::real.\ (x < -\ y) = (x + y < (0::real))$

thm REAL_LT_ADDNEG:

$\forall (x::real)\ (y::real)\ z::real.\ (y < x + -\ z) = (y + z < x)$

thm REAL_LT_ADDNEG2:

$\forall (x::real)\ (y::real)\ z::real.\ (x + -\ y < z) = (x < z + y)$

thm REAL_LT_ADD1:

$\forall (x::real)\ y::real.\ x \leq y \longrightarrow x < y + (1::real)$

thm REAL_SUB_ADD:

$\forall (x::real)\ y::real.\ x - y + y = x$

thm REAL_SUB_ADD2:

$\forall (x::real)\ y::real.\ y + (x - y) = x$

thm REAL_SUB_REFL:

$\forall x::real.\ x - x = (0::real)$

thm REAL_LE_DOUBLE:

$\forall x::real.\ ((0::real) \leq x + x) = ((0::real) \leq x)$

thm REAL_LE_NEGL:

$\forall x::real.\ (-\ x \leq x) = ((0::real) \leq x)$

thm REAL_LE_NEGR:

$\forall x::real.\ (x \leq -\ x) = (x \leq (0::real))$

thm REAL_NEG_EQ_0:

$\forall x::real.\ (-\ x = (0::real)) = (x = (0::real))$

thm REAL_ADD_SUB:

$\forall (x::real)\ y::real.\ x + y - x = y$

thm REAL_NEG_EQ:

$\forall (x::real)\ y::real.\ (-\ x = y) = (x = -\ y)$

thm REAL_LT_IMP_NE:

$\forall (x::real)\ y::real.\ x < y \longrightarrow x \neq y$

thm REAL_LE_ADDR:

$\forall (x::real)\ y::real.\ (x \leq x + y) = ((0::real) \leq y)$

thm REAL_LE_ADDL:

$\forall (x::real)\ y::real.\ (y \leq x + y) = ((0::real) \leq x)$

thm REAL_LT_ADDR:

$\forall (x::real)\ y::real.\ (x < x + y) = ((0::real) < y)$

thm REAL_LT_ADDL:

$\forall (x::real)\ y::real.\ (y < x + y) = ((0::real) < x)$

thm REAL_SUB_SUB:

$\forall (x::real)\ y::real.\ x - y - x = -\ y$

thm REAL_LT_ADD_SUB:

$\forall (x::real)\ (y::real)\ z::real.\ (x + y < z) = (x < z - y)$

thm REAL_LT_SUB_RADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x - y < z) = (x < z + y)$

thm REAL_LT_SUB_LADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x < y - z) = (x + z < y)$

thm REAL_LE_SUB_LADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x \leq y - z) = (x + z \leq y)$

thm REAL_LE_SUB_RADD:

$\forall (x::real)\ (y::real)\ z::real.\ (x - y \leq z) = (x \leq z + y)$

thm REAL_LT_NEG2:

$\forall (x::real)\ y::real.\ (-\ x < -\ y) = (y < x)$

thm REAL_ADD2_SUB2:

$\forall (a::real)\ (b::real)\ (c::real)\ d::real.\ a + b - (c + d) = a - c + (b - d)$

thm REAL_SUB_LZERO:

$\forall x::real.\ (0::real) - x = -\ x$

thm REAL_SUB_RZERO:

$\forall x{::}real.\ x - (0{::}real) = x$

thm REAL_LET_ADD2:

$\forall (w{::}real)\ (x{::}real)\ (y{::}real)\ z{::}real.\ w \le x \wedge y < z \longrightarrow w + y < x + z$

thm REAL_LTE_ADD2:

$\forall (w{::}real)\ (x{::}real)\ (y{::}real)\ z{::}real.\ w < x \wedge y \le z \longrightarrow w + y < x + z$

thm REAL_SUB_LNEG:

$\forall (x{::}real)\ y{::}real.\ - x - y = - (x + y)$

thm REAL_SUB_RNEG:

$\forall (x{::}real)\ y{::}real.\ x - - y = x + y$

thm REAL_SUB_NEG2:

$\forall (x{::}real)\ y{::}real.\ - x - - y = y - x$

thm REAL_SUB_TRIANGLE:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ a - b + (b - c) = a - c$

thm REAL_EQ_SUB_LADD:

$\forall (x{::}real)\ (y{::}real)\ z{::}real.\ (x = y - z) = (x + z = y)$

thm REAL_EQ_SUB_RADD:

$\forall (x{::}real)\ (y{::}real)\ z{::}real.\ (x - y = z) = (x = z + y)$

thm REAL_SUB_SUB2:

$\forall (x{::}real)\ y{::}real.\ x - (x - y) = y$

thm REAL_ADD_SUB2:

$\forall (x{::}real)\ y{::}real.\ x - (x + y) = - y$

thm REAL_EQ_IMP_LE:

$\forall (x{::}real)\ y{::}real.\ x = y \longrightarrow x \le y$

thm REAL_POS_NZ:

$\forall x{>}0{::}real.\ x \neq (0{::}real)$

thm REAL_DIFFSQ:

$\forall (x{::}real)\ y{::}real.\ (x + y) * (x - y) = x * x - y * y$

thm REAL_EQ_NEG2:

$\forall (x{::}real)\ y{::}real.\ (- x = - y) = (x = y)$

thm REAL_SUB_LDISTRIB:

$\forall (x{::}real)\ (y{::}real)\ z{::}real.\ x * (y - z) = x * y - x * z$

thm REAL_SUB_RDISTRIB:

$\forall (x::real)\ (y::real)\ z::real.\ (x - y) * z = x * z - y * z$

thm REAL_ABS_ZERO:

$\forall x::real.\ (|x| = (0::real)) = (x = (0::real))$

thm REAL_ABS_0:

$|0::real| = (0::real)$

thm REAL_ABS_1:

$|1::real| = (1::real)$

thm REAL_ABS_TRIANGLE:

$\forall (x::real)\ y::real.\ |x + y| \leq |x| + |y|$

thm REAL_ABS_TRIANGLE_LE:

$\forall (x::real)\ (y::real)\ z::real.\ |x| + |y - x| \leq z \longrightarrow |y| \leq z$

thm REAL_ABS_TRIANGLE_LT:

$\forall (x::real)\ (y::real)\ z::real.\ |x| + |y - x| < z \longrightarrow |y| < z$

thm REAL_ABS_POS:

$\forall x::real.\ (0::real) \leq |x|$

thm REAL_ABS_SUB:

$\forall (x::real)\ y::real.\ |x - y| = |y - x|$

thm REAL_ABS_NZ:

$\forall x::real.\ (x \neq (0::real)) = ((0::real) < |x|)$

thm REAL_ABS_ABS:

$\forall x::real.\ ||x|| = |x|$

thm REAL_ABS_LE:

$\forall x::real.\ x \leq |x|$

thm REAL_ABS_REFL:

$\forall x::real.\ (|x| = x) = ((0::real) \leq x)$

thm REAL_ABS_BETWEEN:

$\forall (x::real)\ (y::real)\ d::real.\ ((0::real) < d \land x - d < y \land y < x + d) = (|y - x| < d)$

thm REAL_ABS_BOUND:

$\forall (x::real)\ (y::real)\ d::real.\ |x - y| < d \longrightarrow y < x + d$

thm REAL_ABS_STILLNZ:

$\forall (x::real)\ y::real.\ |x - y| < |y| \longrightarrow x \neq (0::real)$

thm REAL_ABS_CASES:

$\forall x::real.\ x = (0::real) \vee (0::real) < |x|$

thm REAL_ABS_BETWEEN1:

$\forall (x::real)\ (y::real)\ z::real.\ x < z \wedge |y - x| < z - x \longrightarrow y < z$

thm REAL_ABS_SIGN2:

$\forall (x::real)\ y::real.\ |x - y| < - y \longrightarrow x < (0::real)$

thm REAL_ABS_CIRCLE:

$\forall (x::real)\ (y::real)\ h::real.\ |h| < |y| - |x| \longrightarrow |x + h| < |y|$

thm REAL_SUB_ABS:

$\forall (x::real)\ y::real.\ |x| - |y| \leq |x - y|$

thm REAL_ABS_SUB_ABS:

$\forall (x::real)\ y::real.\ ||x| - |y|| \leq |x - y|$

thm REAL_ABS_BETWEEN2:

$\forall (x0::real)\ (x::real)\ (y0::real)\ y::real.\ x0 < y0 \wedge real\_of\_nat\ (2::nat) * |x - x0| < y0 - x0 \wedge real\_of\_nat\ (2::nat) * |y - y0| < y0 - x0 \longrightarrow x < y$

thm REAL_ABS_BOUNDS:

$\forall (x::real)\ k::real.\ (|x| \leq k) = (- k \leq x \wedge x \leq k)$

thm REAL_BOUNDS_LE:

$\forall (x::real)\ k::real.\ (- k \leq x \wedge x \leq k) = (|x| \leq k)$

thm REAL_BOUNDS_LT:

$\forall (x::real)\ k::real.\ (- k < x \wedge x < k) = (|x| < k)$

thm REAL_MIN_MAX:

$\forall (x::real)\ y::real.\ min\ x\ y = - max\ (- x)\ (- y)$

thm REAL_MAX_MIN:

$\forall (x::real)\ y::real.\ max\ x\ y = - min\ (- x)\ (- y)$

thm REAL_MAX_MAX:

$\forall (x::real)\ y::real.\ x \leq max\ x\ y \wedge y \leq max\ x\ y$

thm REAL_MIN_MIN:

$\forall (x::real)\ y::real.\ min\ x\ y \leq x \wedge min\ x\ y \leq y$

thm REAL_MAX_SYM:

$\forall (x::real)\ y::real.\ max\ x\ y = max\ y\ x$

thm REAL_MIN_SYM:

$\forall (x::real)\ y::real.\ min\ x\ y = min\ y\ x$

thm REAL_LE_MAX:

$\forall (x::real)\ (y::real)\ z::real.\ (z \leq max\ x\ y) = (z \leq x \vee z \leq y)$

thm REAL_LE_MIN:

$\forall (x::real)\ (y::real)\ z::real.\ (z \leq min\ x\ y) = (z \leq x \wedge z \leq y)$

thm REAL_LT_MAX:

$\forall (x::real)\ (y::real)\ z::real.\ (z < max\ x\ y) = (z < x \vee z < y)$

thm REAL_LT_MIN:

$\forall (x::real)\ (y::real)\ z::real.\ (z < min\ x\ y) = (z < x \wedge z < y)$

thm REAL_MAX_LE:

$\forall (x::real)\ (y::real)\ z::real.\ (max\ x\ y \leq z) = (x \leq z \wedge y \leq z)$

thm REAL_MIN_LE:

$\forall (x::real)\ (y::real)\ z::real.\ (min\ x\ y \leq z) = (x \leq z \vee y \leq z)$

thm REAL_MAX_LT:

$\forall (x::real)\ (y::real)\ z::real.\ (max\ x\ y < z) = (x < z \wedge y < z)$

thm REAL_MIN_LT:

$\forall (x::real)\ (y::real)\ z::real.\ (min\ x\ y < z) = (x < z \vee y < z)$

thm REAL_MAX_ASSOC:

$\forall (x::real)\ (y::real)\ z::real.\ max\ x\ (max\ y\ z) = max\ (max\ x\ y)\ z$

thm REAL_MIN_ASSOC:

$\forall (x::real)\ (y::real)\ z::real.\ min\ x\ (min\ y\ z) = min\ (min\ x\ y)\ z$

thm REAL_MAX_ACI:

$max\ (?x::real)\ (?y::real) = max\ ?y\ ?x \wedge max\ (max\ ?x\ ?y)\ (?z::real) = max\ ?x\ (max\ ?y\ ?z) \wedge max\ ?x\ (max\ ?y\ ?z) = max\ ?y\ (max\ ?x\ ?z) \wedge max\ ?x\ ?x = ?x \wedge max\ ?x\ (max\ ?x\ ?y) = max\ ?x\ ?y$

thm REAL_MIN_ACI:

$min\ (?x::real)\ (?y::real) = min\ ?y\ ?x \wedge min\ (min\ ?x\ ?y)\ (?z::real) = min\ ?x\ (min\ ?y\ ?z) \wedge min\ ?x\ (min\ ?y\ ?z) = min\ ?y\ (min\ ?x\ ?z) \wedge min\ ?x\ ?x = ?x \wedge min\ ?x\ (min\ ?x\ ?y) = min\ ?x\ ?y$

thm REAL_ABS_MUL:

$\forall (x::real)\ y::real.\ |x * y| = |x| * |y|$

thm REAL_POW_LE:

$\forall (x::real)\ n::nat.\ (0::real) \leq x \longrightarrow (0::real) \leq x^n$

thm REAL_POW_LT:

$\forall (x::real)\ n::nat.\ (0::real) < x \longrightarrow (0::real) < x^n$

thm REAL_ABS_POW:

$\forall (x::real)\ n::nat.\ |x^n| = |x|^n$

thm REAL_LE_LMUL:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge y \leq z \longrightarrow x * y \leq x * z$

thm REAL_LE_RMUL:

$\forall (x::real)\ (y::real)\ z::real.\ x \leq y \wedge (0::real) \leq z \longrightarrow x * z \leq y * z$

thm REAL_LT_LMUL:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \wedge y < z \longrightarrow x * y < x * z$

thm REAL_LT_RMUL:

$\forall (x::real)\ (y::real)\ z::real.\ x < y \wedge (0::real) < z \longrightarrow x * z < y * z$

thm REAL_EQ_MUL_LCANCEL:

$\forall (x::real)\ (y::real)\ z::real.\ (x * y = x * z) = (x = (0::real) \vee y = z)$

thm REAL_EQ_MUL_RCANCEL:

$\forall (x::real)\ (y::real)\ z::real.\ (x * z = y * z) = (x = y \vee z = (0::real))$

thm REAL_MUL_LINV_UNIQ:

$\forall (x::real)\ y::real.\ x * y = (1::real) \longrightarrow inverse\ y = x$

thm REAL_MUL_RINV_UNIQ:

$\forall (x::real)\ y::real.\ x * y = (1::real) \longrightarrow inverse\ x = y$

thm REAL_INV_INV:

$\forall x::real.\ inverse\ (inverse\ x) = x$

thm REAL_EQ_INV2:

$\forall (x::real)\ y::real.\ (inverse\ x = inverse\ y) = (x = y)$

thm REAL_INV_EQ_0:

$\forall x::real.\ (inverse\ x = (0::real)) = (x = (0::real))$

thm REAL_LT_INV:

$\forall x>0::real.\ (0::real) < inverse\ x$

thm REAL_LT_INV_EQ:

$\forall x::real.\ ((0::real) < inverse\ x) = ((0::real) < x)$

thm REAL_INV_NEG:

$\forall x::real.\ inverse\ (-\ x) = -\ inverse\ x$

thm REAL_LE_INV_EQ:

$\forall x::real.\ ((0::real) \leq inverse\ x) = ((0::real) \leq x)$

thm REAL_LE_INV:

$\forall x\geq 0::real.\ (0::real) \leq inverse\ x$

thm REAL_MUL_RINV:

$\forall x::real.\ x \neq (0::real) \longrightarrow x * inverse\ x = (1::real)$

thm REAL_INV_1:

$inverse\ (1::real) = (1::real)$

thm REAL_INV_EQ_1:

$\forall x::real.\ (inverse\ x = (1::real)) = (x = (1::real))$

thm REAL_DIV_1:

$\forall x::real.\ x\ /\ (1::real) = x$

thm REAL_DIV_REFL:

$\forall x::real.\ x \neq (0::real) \longrightarrow x\ /\ x = (1::real)$

thm REAL_DIV_RMUL:

$\forall (x::real)\ y::real.\ y \neq (0::real) \longrightarrow x\ /\ y * y = x$

thm REAL_DIV_LMUL:

$\forall (x::real)\ y::real.\ y \neq (0::real) \longrightarrow y * (x\ /\ y) = x$

thm REAL_ABS_INV:

$\forall x::real.\ |inverse\ x| = inverse\ |x|$

thm REAL_ABS_DIV:

$\forall (x::real)\ y::real.\ |x\ /\ y| = |x|\ /\ |y|$

thm REAL_MUL_AC_conjunct2:

$(?m::real) * ((?n::real) * (?p::real)) = ?n * (?m * ?p)$

thm REAL_MUL_AC_conjunct1:

$(?m::real) * (?n::real) * (?p::real) = ?m * (?n * ?p)$

thm REAL_INV_MUL:

$\forall (x::real)\ y::real.\ inverse\ (x * y) = inverse\ x * inverse\ y$

thm REAL_INV_DIV:

$\forall (x::real)\ y::real.\ inverse\ (x\ /\ y) = y\ /\ x$

thm REAL_POW_MUL:

$\forall (x::real)\ (y::real)\ n::nat.\ (x * y)^n = x^n * y^n$

thm REAL_POW_INV:

$\forall (x::real)\ n::nat.\ (inverse\ x)^n = inverse\ x^n$

thm REAL_INV_POW:

$\forall (x::real)\ n::nat.\ inverse\ x^n = (inverse\ x)^n$

thm REAL_POW_DIV:

$\forall (x::real)\ (y::real)\ n::nat.\ (x\ /\ y)^n = x^n\ /\ y^n$

thm REAL_POW_ADD:

$\forall (x::real)\ (m::nat)\ n::nat.\ x^{m\ +\ n} = x^m * x^n$

thm REAL_POW_NZ:

$\forall (x::real)\ n::nat.\ x \neq (0::real) \longrightarrow x^n \neq (0::real)$

thm REAL_POW_SUB:

$\forall (x::real)\ (m::nat)\ n::nat.\ x \neq (0::real) \wedge m \leq n \longrightarrow x^{n\ -\ m} = x^n\ /\ x^m$

thm REAL_LT_LCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \wedge x * y < x * z \longrightarrow y < z$

thm REAL_LT_RCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \wedge x * z < y * z \longrightarrow x < y$

thm REAL_LE_LCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \wedge x * y \leq x * z \longrightarrow y \leq z$

thm REAL_LE_RCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \wedge x * z \leq y * z \longrightarrow x \leq y$

thm REAL_LE_RMUL_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x * z \leq y * z) = (x \leq y)$

thm REAL_LE_LMUL_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (z * x \leq z * y) = (x \leq y)$

thm REAL_LT_RMUL_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x * z < y * z) = (x < y)$

thm REAL_LT_LMUL_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (z * x < z * y) = (x < y)$

thm REAL_LE_MUL_EQ:

$(\forall (x::real)\ y::real.\ (0::real) < x \longrightarrow ((0::real) \leq x * y) = ((0::real) \leq y)) \wedge (\forall (x::real)\ y::real.\ (0::real) < y \longrightarrow ((0::real) \leq x * y) = ((0::real) \leq x))$

thm REAL_LT_MUL_EQ:

$(\forall (x::real)\ y::real.\ (0::real) < x \longrightarrow ((0::real) < x * y) = ((0::real) < y)) \wedge (\forall (x::real)\ y::real.\ (0::real) < y \longrightarrow ((0::real) < x * y) = ((0::real) < x))$

thm REAL_MUL_POS_LT:

$\forall (x::real)\ y::real.\ ((0::real) < x * y) = ((0::real) < x \wedge (0::real) < y \vee x < (0::real) \wedge y < (0::real))$

thm REAL_MUL_POS_LE:

$\forall (x::real)\ y::real.\ ((0::real) \leq x * y) = (x = (0::real) \vee y = (0::real) \vee (0::real) < x \wedge (0::real) < y \vee x < (0::real) \wedge y < (0::real))$

thm REAL_LE_RDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x \leq y\ /\ z) = (x * z \leq y)$

thm REAL_LE_LDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x\ /\ z \leq y) = (x \leq y * z)$

thm REAL_LT_RDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x < y\ /\ z) = (x * z < y)$

thm REAL_LT_LDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x\ /\ z < y) = (x < y * z)$

thm REAL_EQ_RDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x = y\ /\ z) = (x * z = y)$

thm REAL_EQ_LDIV_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x\ /\ z = y) = (x = y * z)$

thm REAL_LT_DIV2_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x\ /\ z < y\ /\ z) = (x < y)$

thm REAL_LE_DIV2_EQ:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < z \longrightarrow (x\ /\ z \leq y\ /\ z) = (x \leq y)$

thm REAL_MUL_2:

$\forall x::real.\ real\_of\_nat\ (2::nat) * x = x + x$

thm REAL_POW_EQ_0:

$\forall (x::real)\ n::nat.\ (x^n = (0::real)) = (x = (0::real) \wedge n \neq (0::nat))$

thm REAL_LE_MUL2:

$\forall (w::real)\ (x::real)\ (y::real)\ z::real.\ (0::real) \leq w \wedge w \leq x \wedge (0::real) \leq y \wedge y \leq z \longrightarrow w * y \leq x * z$

thm REAL_LT_MUL2:

$\forall (w::real)\ (x::real)\ (y::real)\ z::real.\ (0::real) \leq w \wedge w < x \wedge (0::real) \leq y \wedge y < z \longrightarrow w * y < x * z$

thm REAL_LT_SQUARE:

$\forall x::real.\ ((0::real) < x * x) = (x \neq (0::real))$

thm REAL_POW_1:

$\forall x::real.\ x^{1::nat} = x$

thm REAL_POW_ONE:

$\forall n::nat.\ (1::real)^n = (1::real)$

thm REAL_LT_INV2:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x < y \longrightarrow inverse\ y < inverse\ x$

thm REAL_LE_INV2:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x \leq y \longrightarrow inverse\ y \leq inverse\ x$

thm REAL_LT_LINV:

$\forall (x::real)\ y::real.\ (0::real) < y \wedge inverse\ y < x \longrightarrow inverse\ x < y$

thm REAL_LT_RINV:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x < inverse\ y \longrightarrow y < inverse\ x$

thm REAL_LE_LINV:

$\forall (x::real)\ y::real.\ (0::real) < y \wedge inverse\ y \leq x \longrightarrow inverse\ x \leq y$

thm REAL_LE_RINV:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x \leq inverse\ y \longrightarrow y \leq inverse\ x$

thm REAL_INV_LE_1:

$\forall x \geq 1::real.\ inverse\ x \leq (1::real)$

thm REAL_INV_1_LE:

$\forall x::real.\ (0::real) < x \wedge x \leq (1::real) \longrightarrow (1::real) \leq inverse\ x$

thm REAL_INV_LT_1:

$\forall x{>}1{::}real.\ inverse\ x < (1{::}real)$

thm REAL_INV_1_LT:

$\forall x{::}real.\ (0{::}real) < x \wedge x < (1{::}real) \longrightarrow (1{::}real) < inverse\ x$

thm REAL_SUB_INV:

$\forall (x{::}real)\ y{::}real.\ x \neq (0{::}real) \wedge y \neq (0{::}real) \longrightarrow inverse\ x - inverse\ y = (y - x) / (x * y)$

thm Real_ext.REAL_INV2_conjunct0:

$inverse\ (real\_of\_nat\ (2{::}nat)) * real\_of\_nat\ (2{::}nat) = (1{::}real)$

thm REAL_DOWN:

$\forall d{>}0{::}real.\ \exists e{>}0{::}real.\ e < d$

thm REAL_DOWN2:

$\forall (d1{::}real)\ d2{::}real.\ (0{::}real) < d1 \wedge (0{::}real) < d2 \longrightarrow (\exists e{>}0{::}real.\ e < d1 \wedge e < d2)$

thm REAL_POW_LE2:

$\forall (n{::}nat)\ (x{::}real)\ y{::}real.\ (0{::}real) \leq x \wedge x \leq y \longrightarrow x^n \leq y^n$

thm REAL_POW_LE_1:

$\forall (n{::}nat)\ x{::}real.\ (1{::}real) \leq x \longrightarrow (1{::}real) \leq x^n$

thm REAL_POW_1_LE:

$\forall (n{::}nat)\ x{::}real.\ (0{::}real) \leq x \wedge x \leq (1{::}real) \longrightarrow x^n \leq (1{::}real)$

thm REAL_POW_MONO:

$\forall (m{::}nat)\ (n{::}nat)\ x{::}real.\ (1{::}real) \leq x \wedge m \leq n \longrightarrow x^m \leq x^n$

thm REAL_POW_LT2:

$\forall (n{::}nat)\ (x{::}real)\ y{::}real.\ n \neq (0{::}nat) \wedge (0{::}real) \leq x \wedge x < y \longrightarrow x^n < y^n$

thm REAL_POW_LT_1:

$\forall (n{::}nat)\ x{::}real.\ n \neq (0{::}nat) \wedge (1{::}real) < x \longrightarrow (1{::}real) < x^n$

thm REAL_POW_1_LT:

$\forall (n{::}nat)\ x{::}real.\ n \neq (0{::}nat) \wedge (0{::}real) \leq x \wedge x < (1{::}real) \longrightarrow x^n < (1{::}real)$

thm REAL_POW_MONO_LT:

$\forall (m{::}nat)\ (n{::}nat)\ x{::}real.\ (1{::}real) < x \wedge m < n \longrightarrow x^m < x^n$

thm REAL_POW_POW:

$\forall (x{::}real)\ (m{::}nat)\ n{::}nat.\ x^{mn} = x^{m\ *\ n}$

thm REAL_EQ_RCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ z \neq (0::real) \wedge x * z = y * z \longrightarrow x = y$

thm REAL_EQ_LCANCEL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ z \neq (0::real) \wedge z * x = z * y \longrightarrow x = y$

thm REAL_LT_DIV:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow (0::real) < x\ /\ y$

thm REAL_LE_DIV:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (0::real) \leq x\ /\ y$

thm REAL_DIV_POW2:

$\forall (x::real)\ (m::nat)\ n::nat.\ x \neq (0::real) \longrightarrow x^m\ /\ x^n = (if\ n \leq m\ then\ x^{m\ -\ n}\ else\ inverse\ x^{n\ -\ m})$

thm REAL_DIV_POW2_ALT:

$\forall (x::real)\ (m::nat)\ n::nat.\ x \neq (0::real) \longrightarrow x^m\ /\ x^n = (if\ n < m\ then\ x^{m\ -\ n}\ else\ inverse\ x^{n\ -\ m})$

thm REAL_LT_POW2:

$\forall n::nat.\ (0::real) < (real\_of\_nat\ (2::nat))^n$

thm REAL_LE_POW2:

$\forall n::nat.\ (1::real) \leq (real\_of\_nat\ (2::nat))^n$

thm REAL_POW2_ABS:

$\forall x::real.\ |x|^2 = x^2$

thm REAL_LE_SQUARE_ABS:

$\forall (x::real)\ y::real.\ (|x| \leq |y|) = (x^2 \leq y^2)$

thm REAL_LT_SQUARE_ABS:

$\forall (x::real)\ y::real.\ (|x| < |y|) = (x^2 < y^2)$

thm REAL_EQ_SQUARE_ABS:

$\forall (x::real)\ y::real.\ (|x| = |y|) = (x^2 = y^2)$

thm REAL_LE_POW_2:

$\forall x::real.\ (0::real) \leq x^2$

thm REAL_SOS_EQ_0:

$\forall (x::real)\ y::real.\ (x^2 + y^2 = (0::real)) = (x = (0::real) \wedge y = (0::real))$

thm REAL_POW_ZERO:

$\forall n::nat.\ (0::real)^n = (if\ n = (0::nat)\ then\ 1::real\ else\ (0::real))$

thm REAL_POW_MONO_INV:

$\forall (m::nat)\ (n::nat)\ x::real.\ (0::real) \leq x \wedge x \leq (1::real) \wedge n \leq m \longrightarrow x^m \leq x^n$

thm REAL_POW_LE2_REV:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq y \wedge x^n \leq y^n \longrightarrow x \leq y$

thm REAL_POW_LT2_REV:

$\forall (n::nat)\ (x::real)\ y::real.\ (0::real) \leq y \wedge x^n < y^n \longrightarrow x < y$

thm REAL_POW_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \wedge x^n = y^n \longrightarrow x = y$

thm REAL_POW_EQ_ABS:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge x^n = y^n \longrightarrow |x| = |y|$

thm REAL_POW_EQ_1_IMP:

$\forall (x::real)\ n::nat.\ n \neq (0::nat) \wedge x^n = (1::real) \longrightarrow |x| = (1::real)$

thm REAL_POW_EQ_1:

$\forall (x::real)\ n::nat.\ (x^n = (1::real)) = (|x| = (1::real) \wedge (x < (0::real) \longrightarrow even\ n) \vee n = (0::nat))$

thm REAL_POW_LT2_ODD:

$\forall (n::nat)\ (x::real)\ y::real.\ x < y \wedge ODD\ n \longrightarrow x^n < y^n$

thm REAL_POW_LE2_ODD:

$\forall (n::nat)\ (x::real)\ y::real.\ x \leq y \wedge ODD\ n \longrightarrow x^n \leq y^n$

thm REAL_POW_LT2_ODD_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ ODD\ n \longrightarrow (x^n < y^n) = (x < y)$

thm REAL_POW_LE2_ODD_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ ODD\ n \longrightarrow (x^n \leq y^n) = (x \leq y)$

thm REAL_POW_EQ_ODD_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ ODD\ n \longrightarrow (x^n = y^n) = (x = y)$

thm REAL_POW_EQ_ODD:

$\forall (n::nat)\ (x::real)\ y::real.\ ODD\ n \wedge x^n = y^n \longrightarrow x = y$

thm REAL_POW_EQ_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ (x^n = y^n) = (if\ even\ n\ then\ n = (0::nat) \vee |x| = |y|\ else\ x = y)$

thm REAL_ARCH_SIMPLE:

$\forall x::real.\ \exists n::nat.\ x \leq real\_of\_nat\ n$

thm REAL_ARCH_LT:

$\forall x::real.\ \exists n::nat.\ x < real\_of\_nat\ n$

thm REAL_ARCH:

$\forall x>0::real.\ \forall y::real.\ \exists n::nat.\ y < real\_of\_nat\ n * x$

thm real_sgn:

$\forall x::real.\ sgn\ x = (if\ (0::real) < x\ then\ 1::real\ else\ if\ x < (0::real)\ then\ -\ (1::real)\ else\ (0::real))$

thm REAL_SGN_0:

$sgn\ (0::real) = (0::real)$

thm REAL_SGN_NEG:

$\forall x::real.\ sgn\ (-\ x) = -\ sgn\ x$

thm REAL_SGN_ABS:

$\forall x::real.\ sgn\ x * |x| = x$

thm REAL_ABS_SGN:

$\forall x::real.\ |sgn\ x| = sgn\ |x|$

thm REAL_SGN:

$\forall x::real.\ sgn\ x = x\ /\ |x|$

thm REAL_SGN_MUL:

$\forall (x::real)\ y::real.\ sgn\ (x * y) = sgn\ x * sgn\ y$

thm REAL_SGN_INV:

$\forall x::real.\ sgn\ (inverse\ x) = sgn\ x$

thm REAL_SGN_DIV:

$\forall (x::real)\ y::real.\ sgn\ (x\ /\ y) = sgn\ x\ /\ sgn\ y$

thm REAL_SGN_EQ:

$(\forall x::real.\ (sgn\ x = (0::real)) = (x = (0::real))) \land (\forall x::real.\ (sgn\ x = (1::real)) = ((0::real) < x)) \land (\forall x::real.\ (sgn\ x = -\ (1::real)) = (x < (0::real)))$

thm REAL_SGN_CASES:

$\forall x::real.\ sgn\ x = (0::real) \lor sgn\ x = (1::real) \lor sgn\ x = -\ (1::real)$

thm REAL_WLOG_LE:

$(\forall (x::real)\ y::real.\ (?P::real \Rightarrow real \Rightarrow bool)\ x\ y = ?P\ y\ x) \wedge (\forall (x::real)\ y::real.\ x \leq y \longrightarrow ?P\ x\ y) \longrightarrow (\forall (x::real)\ y::real.\ ?P\ x\ y)$

thm REAL_WLOG_LT:

$(\forall x::real.\ (?P::real \Rightarrow real \Rightarrow bool)\ x\ x) \wedge (\forall (x::real)\ y::real.\ ?P\ x\ y = ?P\ y\ x) \wedge (\forall (x::real)\ y::real.\ x < y \longrightarrow ?P\ x\ y) \longrightarrow (\forall (x::real)\ y::real.\ ?P\ x\ y)$

thm DEF_DECIMAL:

$DECIMAL = (\lambda(\_15534::nat)\ \_15535::nat.\ real\_of\_nat\ \_15534\ /\ real\_of\_nat\ \_15535)$

thm DECIMAL:

$\forall (x::nat)\ y::nat.\ DECIMAL\ x\ y = real\_of\_nat\ x\ /\ real\_of\_nat\ y$

thm Collect_geom.PER_MUL3_conjunct0:

$(?a::real) * ((?b::real) * (?c::real)) = ?b * (?a * ?c)$

thm RAT_LEMMA1:

$(?y1.0::real) \neq (0::real) \wedge (?y2.0::real) \neq (0::real) \longrightarrow (?x1.0::real)\ /\ ?y1.0 + (?x2.0::real)\ /\ ?y2.0 = (?x1.0 * ?y2.0 + ?x2.0 * ?y1.0) * (inverse\ ?y1.0 * inverse\ ?y2.0)$

thm RAT_LEMMA2:

$(0::real) < (?y1.0::real) \wedge (0::real) < (?y2.0::real) \longrightarrow (?x1.0::real)\ /\ ?y1.0 + (?x2.0::real)\ /\ ?y2.0 = (?x1.0 * ?y2.0 + ?x2.0 * ?y1.0) * (inverse\ ?y1.0 * inverse\ ?y2.0)$

thm RAT_LEMMA3:

$(0::real) < (?y1.0::real) \wedge (0::real) < (?y2.0::real) \longrightarrow (?x1.0::real)\ /\ ?y1.0 - (?x2.0::real)\ /\ ?y2.0 = (?x1.0 * ?y2.0 - ?x2.0 * ?y1.0) * (inverse\ ?y1.0 * inverse\ ?y2.0)$

thm RAT_LEMMA4:

$(0::real) < (?y1.0::real) \wedge (0::real) < (?y2.0::real) \longrightarrow ((?x1.0::real)\ /\ ?y1.0 \leq (?x2.0::real)\ /\ ?y2.0) = (?x1.0 * ?y2.0 \leq ?x2.0 * ?y1.0)$

thm RAT_LEMMA5:

$(0::real) < (?y1.0::real) \wedge (0::real) < (?y2.0::real) \longrightarrow ((?x1.0::real)\ /\ ?y1.0 = (?x2.0::real)\ /\ ?y2.0) = (?x1.0 * ?y2.0 = ?x2.0 * ?y1.0)$

thm integer:

$\forall x::real.\ integer\ x = (\exists n::nat.\ |x| = real\_of\_nat\ n)$

thm is_int:

$integer\ (?x::real) = (\exists n::nat.\ ?x = real\_of\_nat\ n \vee ?x = -\ real\_of\_nat\ n)$

thm int_abstr:

$\lfloor real\_of\_int\ (?a::int) \rfloor = ?a$

thm int_rep:

$integer\ (?r::real) = (real\_of\_int\ \lfloor ?r \rfloor = ?r)$

thm int_tybij_conjunct1:

$\forall r::real.\ integer\ r = (real\_of\_int\ \lfloor r \rfloor = r)$

thm int_tybij_conjunct0:

$\forall a::int.\ \lfloor real\_of\_int\ a \rfloor = a$

thm int_tybij:

$(\forall a::int.\ \lfloor real\_of\_int\ a \rfloor = a) \wedge (\forall r::real.\ integer\ r = (real\_of\_int\ \lfloor r \rfloor = r))$

thm int_eq:

$\forall (x::int)\ y::int.\ (x = y) = (real\_of\_int\ x = real\_of\_int\ y)$

thm int_le:

$\forall (x::int)\ y::int.\ (x \leq y) = (real\_of\_int\ x \leq real\_of\_int\ y)$

thm int_lt:

$\forall (x::int)\ y::int.\ (x < y) = (real\_of\_int\ x < real\_of\_int\ y)$

thm int_ge:

$\forall (x::int)\ y::int.\ (y \leq x) = (real\_of\_int\ y \leq real\_of\_int\ x)$

thm int_gt:

$\forall (x::int)\ y::int.\ (y < x) = (real\_of\_int\ y < real\_of\_int\ x)$

thm int_of_num:

$\forall n::nat.\ int\ n = \lfloor real\_of\_nat\ n \rfloor$

thm int_of_num_th:

$\forall n::nat.\ real\_of\_int\ (int\ n) = real\_of\_nat\ n$

thm DEF_int_neg:

$int\_neg = (\lambda\_15683::int.\ \lfloor - real\_of\_int\ \_15683 \rfloor)$

thm int_neg_th:

$\forall x::int.\ real\_of\_int\ (- x) = - real\_of\_int\ x$

thm int_add_th:

$\forall (x::int)\ y::int.\ real\_of\_int\ (x + y) = real\_of\_int\ x + real\_of\_int\ y$

thm int_sub_th:

$\forall (x::int)\ y::int.\ real\_of\_int\ (x - y) = real\_of\_int\ x - real\_of\_int\ y$

thm int_mul_th:

$\forall (x::int)\ y::int.\ real\_of\_int\ (x * y) = real\_of\_int\ x * real\_of\_int\ y$

thm int_abs_th:

$\forall x::int.\ real\_of\_int\ |x| = |real\_of\_int\ x|$

thm int_sgn_th:

$\forall x::int.\ real\_of\_int\ (sgn\ x) = sgn\ (real\_of\_int\ x)$

thm int_max_th:

$\forall (x::int)\ y::int.\ real\_of\_int\ (max\ x\ y) = max\ (real\_of\_int\ x)\ (real\_of\_int\ y)$

thm int_min_th:

$\forall (x::int)\ y::int.\ real\_of\_int\ (min\ x\ y) = min\ (real\_of\_int\ x)\ (real\_of\_int\ y)$

thm int_pow_th:

$\forall (x::int)\ n::nat.\ real\_of\_int\ x^n = (real\_of\_int\ x)^n$

thm INT_IMAGE:

$\forall x::int.\ (\exists n::nat.\ x = int\ n) \lor (\exists n::nat.\ x = -\ int\ n)$

thm INT_LT_DISCRETE:

$\forall (x::int)\ y::int.\ (x < y) = (x + int\ (1::nat) \le y)$

thm INT_GT_DISCRETE:

$\forall (x::int)\ y::int.\ (y < x) = (y + int\ (1::nat) \le x)$

thm INT_ABS_0:

$|int\ (0::nat)| = int\ (0::nat)$

thm INT_ABS_1:

$|int\ (1::nat)| = int\ (1::nat)$

thm INT_ABS_ABS:

$\forall x::int.\ ||x|| = |x|$

thm INT_ABS_BETWEEN:

$\forall (x::int)\ (y::int)\ d::int.\ (int\ (0::nat) < d \land x - d < y \land y < x + d) = (|y - x| < d)$

thm INT_ABS_BETWEEN1:

$\forall (x::int)\ (y::int)\ z::int.\ x < z \land |y - x| < z - x \longrightarrow y < z$

thm INT_ABS_BETWEEN2:

$\forall (x0::int)\ (x::int)\ (y0::int)\ y::int.\ x0 < y0 \land int\ (2::nat) * |x - x0| < y0 - x0 \land int\ (2::nat) * |y - y0| < y0 - x0 \longrightarrow x < y$

thm INT_ABS_BOUND:

$\forall (x::int)\ (y::int)\ d::int.\ |x - y| < d \longrightarrow y < x + d$

thm INT_ABS_CASES:

$\forall x::int.\ x = int\ (0::nat) \lor int\ (0::nat) < |x|$

thm INT_ABS_CIRCLE:

$\forall (x::int)\ (y::int)\ h::int.\ |h| < |y| - |x| \longrightarrow |x + h| < |y|$

thm INT_ABS_LE:

$\forall x::int.\ x \leq |x|$

thm INT_ABS_MUL:

$\forall (x::int)\ y::int.\ |x * y| = |x| * |y|$

thm INT_ABS_NEG:

$\forall x::int.\ |- x| = |x|$

thm INT_ABS_NUM:

$\forall n::nat.\ |int\ n| = int\ n$

thm INT_ABS_NZ:

$\forall x::int.\ (x \neq int\ (0::nat)) = (int\ (0::nat) < |x|)$

thm INT_ABS_POS:

$\forall x::int.\ int\ (0::nat) \leq |x|$

thm INT_ABS_POW:

$\forall (x::int)\ n::nat.\ |x^n| = |x|^n$

thm INT_ABS_REFL:

$\forall x::int.\ (|x| = x) = (int\ (0::nat) \leq x)$

thm INT_ABS_SGN:

$\forall x::int.\ |sgn\ x| = sgn\ |x|$

thm INT_ABS_SIGN:

$\forall (x::int)\ y::int.\ |x - y| < y \longrightarrow int\ (0::nat) < x$

thm INT_ABS_SIGN2:

$\forall (x::int)\ y::int.\ |x - y| < - y \longrightarrow x < int\ (0::nat)$

thm INT_ABS_STILLNZ:

$\forall (x::int)\ y::int.\ |x - y| < |y| \longrightarrow x \neq int\ (0::nat)$

thm INT_ABS_SUB:

$\forall\,(x::int)\ y::int.\ |x\ -\ y|\ =\ |y\ -\ x|$

thm INT_ABS_SUB_ABS:

$\forall\,(x::int)\ y::int.\ ||x|\ -\ |y||\ \leq\ |x\ -\ y|$

thm INT_ABS_TRIANGLE:

$\forall\,(x::int)\ y::int.\ |x\ +\ y|\ \leq\ |x|\ +\ |y|$

thm INT_ABS_ZERO:

$\forall\,x::int.\ (|x|\ =\ int\ (0::nat))\ =\ (x\ =\ int\ (0::nat))$

thm INT_ADD2_SUB2:

$\forall\,(a::int)\ (b::int)\ (c::int)\ d::int.\ a\ +\ b\ -\ (c\ +\ d)\ =\ a\ -\ c\ +\ (b\ -\ d)$

thm INT_ADD_AC_conjunct0:

$(?m::int)\ +\ (?n::int)\ =\ ?n\ +\ ?m$

thm INT_ADD_AC:

$(?m::int)\ +\ (?n::int)\ =\ ?n\ +\ ?m\ \wedge\ ?m\ +\ ?n\ +\ (?p::int)\ =\ ?m\ +\ (?n\ +\ ?p)$
$\wedge\ ?m\ +\ (?n\ +\ ?p)\ =\ ?n\ +\ (?m\ +\ ?p)$

thm INT_ADD_ASSOC:

$\forall\,(x::int)\ (y::int)\ z::int.\ x\ +\ (y\ +\ z)\ =\ x\ +\ y\ +\ z$

thm INT_ADD_LDISTRIB:

$\forall\,(x::int)\ (y::int)\ z::int.\ x\ *\ (y\ +\ z)\ =\ x\ *\ y\ +\ x\ *\ z$

thm INT_ADD_LID:

$\forall\,x::int.\ int\ (0::nat)\ +\ x\ =\ x$

thm INT_ADD_LINV:

$\forall\,x::int.\ -\ x\ +\ x\ =\ int\ (0::nat)$

thm INT_ADD_RDISTRIB:

$\forall\,(x::int)\ (y::int)\ z::int.\ (x\ +\ y)\ *\ z\ =\ x\ *\ z\ +\ y\ *\ z$

thm INT_ADD_RID:

$\forall\,x::int.\ x\ +\ int\ (0::nat)\ =\ x$

thm INT_ADD_RINV:

$\forall\,x::int.\ x\ +\ -\ x\ =\ int\ (0::nat)$

thm INT_ADD_SUB:

$\forall\,(x::int)\ y::int.\ x\ +\ y\ -\ x\ =\ y$

thm INT_ADD_SUB2:

$\forall (x{::}int)\ y{::}int.\ x - (x + y) = - y$

thm INT_ADD_SYM:

$\forall (x{::}int)\ y{::}int.\ x + y = y + x$

thm INT_BOUNDS_LE:

$\forall (x{::}int)\ k{::}int.\ (- k \le x \wedge x \le k) = (|x| \le k)$

thm INT_BOUNDS_LT:

$\forall (x{::}int)\ k{::}int.\ (- k < x \wedge x < k) = (|x| < k)$

thm INT_DIFFSQ:

$\forall (x{::}int)\ y{::}int.\ (x + y) * (x - y) = x * x - y * y$

thm INT_ENTIRE:

$\forall (x{::}int)\ y{::}int.\ (x * y = int\ (0{::}nat)) = (x = int\ (0{::}nat) \vee y = int\ (0{::}nat))$

thm INT_EQ_ADD_LCANCEL:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x + y = x + z) = (y = z)$

thm INT_EQ_ADD_LCANCEL_0:

$\forall (x{::}int)\ y{::}int.\ (x + y = x) = (y = int\ (0{::}nat))$

thm INT_EQ_ADD_RCANCEL:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x + z = y + z) = (x = y)$

thm INT_EQ_ADD_RCANCEL_0:

$\forall (x{::}int)\ y{::}int.\ (x + y = y) = (x = int\ (0{::}nat))$

thm INT_EQ_IMP_LE:

$\forall (x{::}int)\ y{::}int.\ x = y \longrightarrow x \le y$

thm INT_EQ_MUL_LCANCEL:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x * y = x * z) = (x = int\ (0{::}nat) \vee y = z)$

thm INT_EQ_MUL_RCANCEL:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x * z = y * z) = (x = y \vee z = int\ (0{::}nat))$

thm INT_EQ_NEG2:

$\forall (x{::}int)\ y{::}int.\ (- x = - y) = (x = y)$

thm INT_EQ_SQUARE_ABS:

$\forall (x{::}int)\ y{::}int.\ (|x| = |y|) = (x^2 = y^2)$

thm INT_EQ_SUB_LADD:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x = y - z) = (x + z = y)$

thm INT_EQ_SUB_RADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x - y = z) = (x = z + y)$

thm INT_LET_ADD:

$\forall (x::int)\ y::int.\ int\ (0::nat) \leq x \wedge int\ (0::nat) < y \longrightarrow int\ (0::nat) < x + y$

thm INT_LET_ADD2:

$\forall (w::int)\ (x::int)\ (y::int)\ z::int.\ w \leq x \wedge y < z \longrightarrow w + y < x + z$

thm INT_LET_ANTISYM:

$\forall (x::int)\ y::int.\ \neg\ (x \leq y \wedge y < x)$

thm INT_LET_TOTAL:

$\forall (x::int)\ y::int.\ x \leq y \vee y < x$

thm INT_LET_TRANS:

$\forall (x::int)\ (y::int)\ z::int.\ x \leq y \wedge y < z \longrightarrow x < z$

thm INT_LE_01:

$int\ (0::nat) \leq int\ (1::nat)$

thm INT_LE_ADD:

$\forall (x::int)\ y::int.\ int\ (0::nat) \leq x \wedge int\ (0::nat) \leq y \longrightarrow int\ (0::nat) \leq x + y$

thm INT_LE_ADD2:

$\forall (w::int)\ (x::int)\ (y::int)\ z::int.\ w \leq x \wedge y \leq z \longrightarrow w + y \leq x + z$

thm INT_LE_ADDL:

$\forall (x::int)\ y::int.\ (y \leq x + y) = (int\ (0::nat) \leq x)$

thm INT_LE_ADDR:

$\forall (x::int)\ y::int.\ (x \leq x + y) = (int\ (0::nat) \leq y)$

thm INT_LE_ANTISYM:

$\forall (x::int)\ y::int.\ (x \leq y \wedge y \leq x) = (x = y)$

thm INT_LE_DOUBLE:

$\forall x::int.\ (int\ (0::nat) \leq x + x) = (int\ (0::nat) \leq x)$

thm INT_LE_LADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x + y \leq x + z) = (y \leq z)$

thm INT_LE_LADD_IMP:

$\forall (x::int)\ (y::int)\ z::int.\ y \leq z \longrightarrow x + y \leq x + z$

thm INT_LE_LMUL:

$\forall (x::int)\ (y::int)\ z::int.\ int\ (0::nat) \leq x \wedge y \leq z \longrightarrow x * y \leq x * z$

thm INT_LE_LNEG:

$\forall (x::int)\ y::int.\ (-\ x \leq y) = (int\ (0::nat) \leq x + y)$

thm INT_LE_LT:

$\forall (x::int)\ y::int.\ (x \leq y) = (x < y \vee x = y)$

thm INT_LE_MAX:

$\forall (x::int)\ (y::int)\ z::int.\ (z \leq max\ x\ y) = (z \leq x \vee z \leq y)$

thm INT_LE_MIN:

$\forall (x::int)\ (y::int)\ z::int.\ (z \leq min\ x\ y) = (z \leq x \wedge z \leq y)$

thm INT_LE_MUL:

$\forall (x::int)\ y::int.\ int\ (0::nat) \leq x \wedge int\ (0::nat) \leq y \longrightarrow int\ (0::nat) \leq x * y$

thm REAL_LE_MUL_EQ_conjunct1:

$\forall (x::real)\ y::real.\ (0::real) < y \longrightarrow ((0::real) \leq x * y) = ((0::real) \leq x)$

thm INT_LE_MUL_EQ_conjunct1:

$\forall (x::int)\ y::int.\ int\ (0::nat) < y \longrightarrow (int\ (0::nat) \leq x * y) = (int\ (0::nat) \leq x)$

thm REAL_LE_MUL_EQ_conjunct0:

$\forall (x::real)\ y::real.\ (0::real) < x \longrightarrow ((0::real) \leq x * y) = ((0::real) \leq y)$

thm INT_LE_MUL_EQ_conjunct0:

$\forall (x::int)\ y::int.\ int\ (0::nat) < x \longrightarrow (int\ (0::nat) \leq x * y) = (int\ (0::nat) \leq y)$

thm INT_LE_MUL_EQ:

$(\forall (x::int)\ y::int.\ int\ (0::nat) < x \longrightarrow (int\ (0::nat) \leq x * y) = (int\ (0::nat) \leq y)) \wedge (\forall (x::int)\ y::int.\ int\ (0::nat) < y \longrightarrow (int\ (0::nat) \leq x * y) = (int\ (0::nat) \leq x))$

thm INT_LE_NEG:

$\forall (x::int)\ y::int.\ (-\ x \leq -\ y) = (y \leq x)$

thm INT_LE_NEGL:

$\forall x::int.\ (-\ x \leq x) = (int\ (0::nat) \leq x)$

thm INT_LE_NEGR:

$\forall x::int.\ (x \leq -\ x) = (x \leq int\ (0::nat))$

thm INT_LE_NEGTOTAL:

$\forall x::int.\ int\ (0::nat) \leq x \vee int\ (0::nat) \leq -\ x$

thm INT_LE_POW2:

$\forall n::nat.\ int\ (1::nat) \leq (int\ (2::nat))^n$

thm INT_LE_RADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x + z \leq y + z) = (x \leq y)$

thm INT_LE_REFL:

$\forall x::int.\ x \leq x$

thm INT_LE_RMUL:

$\forall (x::int)\ (y::int)\ z::int.\ x \leq y \wedge int\ (0::nat) \leq z \longrightarrow x * z \leq y * z$

thm INT_LE_RNEG:

$\forall (x::int)\ y::int.\ (x \leq -\ y) = (x + y \leq int\ (0::nat))$

thm INT_LE_SQUARE:

$\forall x::int.\ int\ (0::nat) \leq x * x$

thm INT_LE_SQUARE_ABS:

$\forall (x::int)\ y::int.\ (|x| \leq |y|) = (x^2 \leq y^2)$

thm INT_LE_SUB_LADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x \leq y - z) = (x + z \leq y)$

thm INT_LE_SUB_RADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x - y \leq z) = (x \leq z + y)$

thm INT_LE_TOTAL:

$\forall (x::int)\ y::int.\ x \leq y \vee y \leq x$

thm INT_LE_TRANS:

$\forall (x::int)\ (y::int)\ z::int.\ x \leq y \wedge y \leq z \longrightarrow x \leq z$

thm INT_LNEG_UNIQ:

$\forall (x::int)\ y::int.\ (x + y = int\ (0::nat)) = (x = -\ y)$

thm INT_LTE_ADD:

$\forall (x::int)\ y::int.\ int\ (0::nat) < x \wedge int\ (0::nat) \leq y \longrightarrow int\ (0::nat) < x + y$

thm INT_LTE_ADD2:

$\forall (w::int)\ (x::int)\ (y::int)\ z::int.\ w < x \wedge y \leq z \longrightarrow w + y < x + z$

thm INT_LTE_ANTISYM:

$\forall (x::int)\ y::int.\ \neg\ (x < y \wedge y \leq x)$

thm INT_LTE_TOTAL:

$\forall (x::int)\ y::int.\ x < y \lor y \leq x$

thm INT_LTE_TRANS:

$\forall (x::int)\ (y::int)\ z::int.\ x < y \land y \leq z \longrightarrow x < z$

thm INT_LT_01:

$int\ (0::nat) < int\ (1::nat)$

thm INT_LT_ADD:

$\forall (x::int)\ y::int.\ int\ (0::nat) < x \land int\ (0::nat) < y \longrightarrow int\ (0::nat) < x + y$

thm INT_LT_ADD1:

$\forall (x::int)\ y::int.\ x \leq y \longrightarrow x < y + int\ (1::nat)$

thm INT_LT_ADD2:

$\forall (w::int)\ (x::int)\ (y::int)\ z::int.\ w < x \land y < z \longrightarrow w + y < x + z$

thm INT_LT_ADDL:

$\forall (x::int)\ y::int.\ (y < x + y) = (int\ (0::nat) < x)$

thm INT_LT_ADDNEG:

$\forall (x::int)\ (y::int)\ z::int.\ (y < x + - z) = (y + z < x)$

thm INT_LT_ADDNEG2:

$\forall (x::int)\ (y::int)\ z::int.\ (x + - y < z) = (x < z + y)$

thm INT_LT_ADDR:

$\forall (x::int)\ y::int.\ (x < x + y) = (int\ (0::nat) < y)$

thm INT_LT_ADD_SUB:

$\forall (x::int)\ (y::int)\ z::int.\ (x + y < z) = (x < z - y)$

thm INT_LT_ANTISYM:

$\forall (x::int)\ y::int.\ \neg\ (x < y \land y < x)$

thm INT_LT_GT:

$\forall (x::int)\ y::int.\ x < y \longrightarrow \neg\ y < x$

thm INT_LT_IMP_LE:

$\forall (x::int)\ y::int.\ x < y \longrightarrow x \leq y$

thm INT_LT_IMP_NE:

$\forall (x::int)\ y::int.\ x < y \longrightarrow x \neq y$

thm INT_LT_LADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x + y < x + z) = (y < z)$

thm INT_LT_LE:

$\forall (x::int)\ y::int.\ (x < y) = (x \leq y \wedge x \neq y)$

thm INT_LT_LMUL_EQ:

$\forall (x::int)\ (y::int)\ z::int.\ int\ (0::nat) < z \longrightarrow (z * x < z * y) = (x < y)$

thm INT_LT_MAX:

$\forall (x::int)\ (y::int)\ z::int.\ (z < max\ x\ y) = (z < x \vee z < y)$

thm INT_LT_MIN:

$\forall (x::int)\ (y::int)\ z::int.\ (z < min\ x\ y) = (z < x \wedge z < y)$

thm INT_LT_MUL:

$\forall (x::int)\ y::int.\ int\ (0::nat) < x \wedge int\ (0::nat) < y \longrightarrow int\ (0::nat) < x * y$

thm REAL_LT_MUL_EQ_conjunct1:

$\forall (x::real)\ y::real.\ (0::real) < y \longrightarrow ((0::real) < x * y) = ((0::real) < x)$

thm INT_LT_MUL_EQ_conjunct1:

$\forall (x::int)\ y::int.\ int\ (0::nat) < y \longrightarrow (int\ (0::nat) < x * y) = (int\ (0::nat) < x)$

thm REAL_LT_MUL_EQ_conjunct0:

$\forall (x::real)\ y::real.\ (0::real) < x \longrightarrow ((0::real) < x * y) = ((0::real) < y)$

thm INT_LT_MUL_EQ_conjunct0:

$\forall (x::int)\ y::int.\ int\ (0::nat) < x \longrightarrow (int\ (0::nat) < x * y) = (int\ (0::nat) < y)$

thm INT_LT_MUL_EQ:

$(\forall (x::int)\ y::int.\ int\ (0::nat) < x \longrightarrow (int\ (0::nat) < x * y) = (int\ (0::nat) < y)) \wedge (\forall (x::int)\ y::int.\ int\ (0::nat) < y \longrightarrow (int\ (0::nat) < x * y) = (int\ (0::nat) < x))$

thm INT_LT_NEG:

$\forall (x::int)\ y::int.\ (-\ x < -\ y) = (y < x)$

thm INT_LT_NEGTOTAL:

$\forall x::int.\ x = int\ (0::nat) \vee int\ (0::nat) < x \vee int\ (0::nat) < -\ x$

thm INT_LT_POW2:

$\forall n::nat.\ int\ (0::nat) < (int\ (2::nat))^n$

thm INT_LT_RADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x + z < y + z) = (x < y)$

thm INT_LT_REFL:

$\forall x::int.\ \neg\ x < x$

thm INT_LT_RMUL_EQ:

$\forall (x::int)\ (y::int)\ z::int.\ int\ (0::nat) < z \longrightarrow (x * z < y * z) = (x < y)$

thm INT_LT_SQUARE_ABS:

$\forall (x::int)\ y::int.\ (|x| < |y|) = (x^2 < y^2)$

thm INT_LT_SUB_LADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x < y - z) = (x + z < y)$

thm INT_LT_SUB_RADD:

$\forall (x::int)\ (y::int)\ z::int.\ (x - y < z) = (x < z + y)$

thm INT_LT_TOTAL:

$\forall (x::int)\ y::int.\ x = y \lor x < y \lor y < x$

thm INT_LT_TRANS:

$\forall (x::int)\ (y::int)\ z::int.\ x < y \land y < z \longrightarrow x < z$

thm REAL_MAX_ACI_conjunct0:

$max\ (?x::real)\ (?y::real) = max\ ?y\ ?x$

thm INT_MAX_ACI_conjunct0:

$max\ (?x::int)\ (?y::int) = max\ ?y\ ?x$

thm INT_MAX_ACI:

$max\ (?x::int)\ (?y::int) = max\ ?y\ ?x \land max\ (max\ ?x\ ?y)\ (?z::int) = max\ ?x\ (max\ ?y\ ?z) \land max\ ?x\ (max\ ?y\ ?z) = max\ ?y\ (max\ ?x\ ?z) \land max\ ?x\ ?x = ?x \land max\ ?x\ (max\ ?x\ ?y) = max\ ?x\ ?y$

thm INT_MAX_ASSOC:

$\forall (x::int)\ (y::int)\ z::int.\ max\ x\ (max\ y\ z) = max\ (max\ x\ y)\ z$

thm INT_MAX_LE:

$\forall (x::int)\ (y::int)\ z::int.\ (max\ x\ y \le z) = (x \le z \land y \le z)$

thm INT_MAX_LT:

$\forall (x::int)\ (y::int)\ z::int.\ (max\ x\ y < z) = (x < z \land y < z)$

thm INT_MAX_MAX:

$\forall (x::int)\ y::int.\ x \le max\ x\ y \land y \le max\ x\ y$

thm INT_MAX_MIN:

$\forall (x::int)\ y::int.\ max\ x\ y = -\ min\ (-\ x)\ (-\ y)$

thm INT_MAX_SYM:

$\forall (x::int)\ y::int.\ max\ x\ y = max\ y\ x$

thm REAL_MIN_ACI_conjunct0:

$min\ (?x::real)\ (?y::real) = min\ ?y\ ?x$

thm INT_MIN_ACI_conjunct0:

$min\ (?x::int)\ (?y::int) = min\ ?y\ ?x$

thm INT_MIN_ACI:

$min\ (?x::int)\ (?y::int) = min\ ?y\ ?x \wedge min\ (min\ ?x\ ?y)\ (?z::int) = min\ ?x\ (min\ ?y\ ?z) \wedge min\ ?x\ (min\ ?y\ ?z) = min\ ?y\ (min\ ?x\ ?z) \wedge min\ ?x\ ?x = ?x \wedge min\ ?x\ (min\ ?x\ ?y) = min\ ?x\ ?y$

thm INT_MIN_ASSOC:

$\forall (x::int)\ (y::int)\ z::int.\ min\ x\ (min\ y\ z) = min\ (min\ x\ y)\ z$

thm INT_MIN_LE:

$\forall (x::int)\ (y::int)\ z::int.\ (min\ x\ y \leq z) = (x \leq z \vee y \leq z)$

thm INT_MIN_LT:

$\forall (x::int)\ (y::int)\ z::int.\ (min\ x\ y < z) = (x < z \vee y < z)$

thm INT_MIN_MAX:

$\forall (x::int)\ y::int.\ min\ x\ y = -\ max\ (-\ x)\ (-\ y)$

thm INT_MIN_MIN:

$\forall (x::int)\ y::int.\ min\ x\ y \leq x \wedge min\ x\ y \leq y$

thm INT_MIN_SYM:

$\forall (x::int)\ y::int.\ min\ x\ y = min\ y\ x$

thm INT_MUL_AC_conjunct0:

$(?m::int) * (?n::int) = ?n * ?m$

thm INT_MUL_AC:

$(?m::int) * (?n::int) = ?n * ?m \wedge ?m * ?n * (?p::int) = ?m * (?n * ?p) \wedge ?m * (?n * ?p) = ?n * (?m * ?p)$

thm INT_MUL_ASSOC:

$\forall (x::int)\ (y::int)\ z::int.\ x * (y * z) = x * y * z$

thm INT_MUL_LID:

$\forall x::int.\ int\ (1::nat) * x = x$

thm INT_MUL_LNEG:

$\forall (x::int)\ y::int.\ -\ x * y = -\ (x * y)$

thm INT_MUL_LZERO:

$\forall x::int.\ int\ (0::nat) * x = int\ (0::nat)$

thm INT_MUL_POS_LE:

$\forall (x::int)\ y::int.\ (int\ (0::nat) \leq x * y) = (x = int\ (0::nat) \vee y = int\ (0::nat) \vee int\ (0::nat) < x \wedge int\ (0::nat) < y \vee x < int\ (0::nat) \wedge y < int\ (0::nat))$

thm INT_MUL_POS_LT:

$\forall (x::int)\ y::int.\ (int\ (0::nat) < x * y) = (int\ (0::nat) < x \wedge int\ (0::nat) < y \vee x < int\ (0::nat) \wedge y < int\ (0::nat))$

thm INT_MUL_RID:

$\forall x::int.\ x * int\ (1::nat) = x$

thm INT_MUL_RNEG:

$\forall (x::int)\ y::int.\ x * -\ y = -\ (x * y)$

thm INT_MUL_RZERO:

$\forall x::int.\ x * int\ (0::nat) = int\ (0::nat)$

thm INT_MUL_SYM:

$\forall (x::int)\ y::int.\ x * y = y * x$

thm INT_NEG_NEG:

$\forall x::int.\ -\ (-\ x) = x$

thm INT_NEG_0:

$-\ int\ (0::nat) = int\ (0::nat)$

thm INT_NEG_ADD:

$\forall (x::int)\ y::int.\ -\ (x + y) = -\ x + -\ y$

thm INT_NEG_EQ:

$\forall (x::int)\ y::int.\ (-\ x = y) = (x = -\ y)$

thm INT_NEG_EQ_0:

$\forall x::int.\ (-\ x = int\ (0::nat)) = (x = int\ (0::nat))$

thm INT_NEG_GE0:

$\forall x::int.\ (int\ (0::nat) \leq -\ x) = (x \leq int\ (0::nat))$

thm INT_NEG_GT0:

$\forall x::int.\ (int\ (0::nat) < -\ x) = (x < int\ (0::nat))$

thm INT_NEG_LE0:

$\forall x{::}int.\ (-\ x \leq int\ (0{::}nat)) = (int\ (0{::}nat) \leq x)$

thm INT_NEG_LMUL:

$\forall (x{::}int)\ y{::}int.\ -\ (x * y) = -\ x * y$

thm INT_NEG_LT0:

$\forall x{::}int.\ (-\ x < int\ (0{::}nat)) = (int\ (0{::}nat) < x)$

thm INT_NEG_MINUS1:

$\forall x{::}int.\ -\ x = -\ int\ (1{::}nat) * x$

thm INT_NEG_MUL2:

$\forall (x{::}int)\ y{::}int.\ -\ x * -\ y = x * y$

thm INT_NEG_RMUL:

$\forall (x{::}int)\ y{::}int.\ -\ (x * y) = x * -\ y$

thm INT_NEG_SUB:

$\forall (x{::}int)\ y{::}int.\ -\ (x - y) = y - x$

thm INT_NOT_EQ:

$\forall (x{::}int)\ y{::}int.\ (x \neq y) = (x < y \vee y < x)$

thm INT_NOT_LE:

$\forall (x{::}int)\ y{::}int.\ (\neg\ x \leq y) = (y < x)$

thm INT_NOT_LT:

$\forall (x{::}int)\ y{::}int.\ (\neg\ x < y) = (y \leq x)$

thm INT_OF_NUM_ADD:

$\forall (m{::}nat)\ n{::}nat.\ int\ m + int\ n = int\ (m + n)$

thm INT_OF_NUM_EQ:

$\forall (m{::}nat)\ n{::}nat.\ (int\ m = int\ n) = (m = n)$

thm INT_OF_NUM_GE:

$\forall (m{::}nat)\ n{::}nat.\ (int\ n \leq int\ m) = (n \leq m)$

thm INT_OF_NUM_GT:

$\forall (m{::}nat)\ n{::}nat.\ (int\ n < int\ m) = (n < m)$

thm INT_OF_NUM_LE:

$\forall (m{::}nat)\ n{::}nat.\ (int\ m \leq int\ n) = (m \leq n)$

thm INT_OF_NUM_LT:

$\forall (m::nat)\ n::nat.\ (int\ m < int\ n) = (m < n)$

thm INT_OF_NUM_MAX:

$\forall (m::nat)\ n::nat.\ max\ (int\ m)\ (int\ n) = int\ (max\ m\ n)$

thm INT_OF_NUM_MIN:

$\forall (m::nat)\ n::nat.\ min\ (int\ m)\ (int\ n) = int\ (min\ m\ n)$

thm INT_OF_NUM_MUL:

$\forall (m::nat)\ n::nat.\ int\ m * int\ n = int\ (m * n)$

thm INT_OF_NUM_POW:

$\forall (x::nat)\ n::nat.\ (int\ x)^n = int\ x^n$

thm INT_OF_NUM_SUB:

$\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow int\ n - int\ m = int\ (n - m)$

thm INT_OF_NUM_SUC:

$\forall n::nat.\ int\ n + int\ (1::nat) = int\ (Suc\ n)$

thm INT_POS:

$\forall n::nat.\ int\ (0::nat) \leq int\ n$

thm INT_POS_NZ:

$\forall x{>}int\ (0::nat).\ x \neq int\ (0::nat)$

thm INT_POW2_ABS:

$\forall x::int.\ |x|^2 = x^2$

thm INT_POW_1:

$\forall x::int.\ x^{1::nat} = x$

thm INT_POW_1_LE:

$\forall (n::nat)\ x::int.\ int\ (0::nat) \leq x \wedge x \leq int\ (1::nat) \longrightarrow x^n \leq int\ (1::nat)$

thm INT_POW_1_LT:

$\forall (n::nat)\ x::int.\ n \neq (0::nat) \wedge int\ (0::nat) \leq x \wedge x < int\ (1::nat) \longrightarrow x^n < int\ (1::nat)$

thm INT_POW_2:

$\forall x::int.\ x^2 = x * x$

thm INT_POW_ADD:

$\forall (x::int)\ (m::nat)\ n::nat.\ x^{m + n} = x^m * x^n$

thm INT_POW_EQ:

$\forall (n::nat)\ (x::int)\ y::int.\ n \neq (0::nat) \wedge int\ (0::nat) \leq x \wedge int\ (0::nat) \leq y \wedge x^n = y^n \longrightarrow x = y$

thm INT_POW_EQ_0:

$\forall (x::int)\ n::nat.\ (x^n = int\ (0::nat)) = (x = int\ (0::nat) \wedge n \neq (0::nat))$

thm INT_POW_EQ_ABS:

$\forall (n::nat)\ (x::int)\ y::int.\ n \neq (0::nat) \wedge x^n = y^n \longrightarrow |x| = |y|$

thm INT_POW_LE:

$\forall (x::int)\ n::nat.\ int\ (0::nat) \leq x \longrightarrow int\ (0::nat) \leq x^n$

thm INT_POW_LE2:

$\forall (n::nat)\ (x::int)\ y::int.\ int\ (0::nat) \leq x \wedge x \leq y \longrightarrow x^n \leq y^n$

thm INT_POW_LE2_ODD:

$\forall (n::nat)\ (x::int)\ y::int.\ x \leq y \wedge ODD\ n \longrightarrow x^n \leq y^n$

thm INT_POW_LE2_REV:

$\forall (n::nat)\ (x::int)\ y::int.\ n \neq (0::nat) \wedge int\ (0::nat) \leq y \wedge x^n \leq y^n \longrightarrow x \leq y$

thm INT_POW_LE_1:

$\forall (n::nat)\ x::int.\ int\ (1::nat) \leq x \longrightarrow int\ (1::nat) \leq x^n$

thm INT_POW_LT:

$\forall (x::int)\ n::nat.\ int\ (0::nat) < x \longrightarrow int\ (0::nat) < x^n$

thm INT_POW_LT2:

$\forall (n::nat)\ (x::int)\ y::int.\ n \neq (0::nat) \wedge int\ (0::nat) \leq x \wedge x < y \longrightarrow x^n < y^n$

thm INT_POW_LT2_REV:

$\forall (n::nat)\ (x::int)\ y::int.\ int\ (0::nat) \leq y \wedge x^n < y^n \longrightarrow x < y$

thm INT_POW_LT_1:

$\forall (n::nat)\ x::int.\ n \neq (0::nat) \wedge int\ (1::nat) < x \longrightarrow int\ (1::nat) < x^n$

thm INT_POW_MONO:

$\forall (m::nat)\ (n::nat)\ x::int.\ int\ (1::nat) \leq x \wedge m \leq n \longrightarrow x^m \leq x^n$

thm INT_POW_MONO_LT:

$\forall (m::nat)\ (n::nat)\ x::int.\ int\ (1::nat) < x \wedge m < n \longrightarrow x^m < x^n$

thm INT_POW_MUL:

$\forall (x::int)\ (y::int)\ n::nat.\ (x * y)^n = x^n * y^n$

thm INT_POW_NEG:

$\forall (x::int)\ n::nat.\ (-\ x)^n = (if\ even\ n\ then\ x^n\ else\ -\ x^n)$

thm INT_POW_NZ:

$\forall (x::int)\ n::nat.\ x \neq int\ (0::nat) \longrightarrow x^n \neq int\ (0::nat)$

thm INT_POW_ONE:

$\forall n::nat.\ (int\ (1::nat))^n = int\ (1::nat)$

thm INT_POW_POW:

$\forall (x::int)\ (m::nat)\ n::nat.\ x^{mn} = x^{m\ *\ n}$

thm INT_POW_ZERO:

$\forall n::nat.\ (int\ (0::nat))^n = (if\ n = (0::nat)\ then\ int\ (1::nat)\ else\ int\ (0::nat))$

thm INT_RNEG_UNIQ:

$\forall (x::int)\ y::int.\ (x + y = int\ (0::nat)) = (y = -\ x)$

thm INT_SGN:

$\forall x::int.\ sgn\ x = (if\ int\ (0::nat) < x\ then\ int\ (1::nat)\ else\ if\ x < int\ (0::nat)\ then\ -\ int\ (1::nat)\ else\ int\ (0::nat))$

thm INT_SGN_0:

$sgn\ (int\ (0::nat)) = int\ (0::nat)$

thm INT_SGN_ABS:

$\forall x::int.\ sgn\ x * |x| = x$

thm INT_SGN_CASES:

$\forall x::int.\ sgn\ x = int\ (0::nat) \lor sgn\ x = int\ (1::nat) \lor sgn\ x = -\ int\ (1::nat)$

thm REAL_SGN_EQ_conjunct0:

$\forall x::real.\ (sgn\ x = (0::real)) = (x = (0::real))$

thm INT_SGN_EQ_conjunct0:

$\forall x::int.\ (sgn\ x = int\ (0::nat)) = (x = int\ (0::nat))$

thm INT_SGN_EQ:

$(\forall x::int.\ (sgn\ x = int\ (0::nat)) = (x = int\ (0::nat))) \land (\forall x::real.\ (sgn\ x = real\_of\_int\ (int\ (1::nat))) = (real\_of\_int\ (int\ (0::nat)) < x)) \land (\forall x::real.\ (sgn\ x = real\_of\_int\ (-\ int\ (1::nat))) = (x < real\_of\_int\ (int\ (0::nat))))$

thm INT_SGN_MUL:

$\forall (x::int)\ y::int.\ sgn\ (x * y) = sgn\ x * sgn\ y$

thm INT_SGN_NEG:

$\forall x{::}int.\ sgn\ (-\ x) = -\ sgn\ x$

thm INT_SOS_EQ_0:

$\forall (x{::}int)\ y{::}int.\ (x^2 + y^2 = int\ (0{::}nat)) = (x = int\ (0{::}nat) \wedge y = int\ (0{::}nat))$

thm INT_SUB_0:

$\forall (x{::}int)\ y{::}int.\ (x - y = int\ (0{::}nat)) = (x = y)$

thm INT_SUB_ABS:

$\forall (x{::}int)\ y{::}int.\ |x| - |y| \leq |x - y|$

thm INT_SUB_ADD:

$\forall (x{::}int)\ y{::}int.\ x - y + y = x$

thm INT_SUB_ADD2:

$\forall (x{::}int)\ y{::}int.\ y + (x - y) = x$

thm INT_SUB_LDISTRIB:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ x * (y - z) = x * y - x * z$

thm INT_SUB_LE:

$\forall (x{::}int)\ y{::}int.\ (int\ (0{::}nat) \leq x - y) = (y \leq x)$

thm INT_SUB_LNEG:

$\forall (x{::}int)\ y{::}int.\ -\ x - y = -\ (x + y)$

thm INT_SUB_LT:

$\forall (x{::}int)\ y{::}int.\ (int\ (0{::}nat) < x - y) = (y < x)$

thm INT_SUB_LZERO:

$\forall x{::}int.\ int\ (0{::}nat) - x = -\ x$

thm INT_SUB_NEG2:

$\forall (x{::}int)\ y{::}int.\ -\ x - -\ y = y - x$

thm INT_SUB_RDISTRIB:

$\forall (x{::}int)\ (y{::}int)\ z{::}int.\ (x - y) * z = x * z - y * z$

thm INT_SUB_REFL:

$\forall x{::}int.\ x - x = int\ (0{::}nat)$

thm INT_SUB_RNEG:

$\forall (x{::}int)\ y{::}int.\ x - -\ y = x + y$

thm INT_SUB_RZERO:

$\forall\, x{::}int.\ x - int\ (0{::}nat) = x$

thm INT_SUB_SUB:

$\forall\, (x{::}int)\ y{::}int.\ x - y - x = - y$

thm INT_SUB_SUB2:

$\forall\, (x{::}int)\ y{::}int.\ x - (x - y) = y$

thm INT_SUB_TRIANGLE:

$\forall\, (a{::}int)\ (b{::}int)\ c{::}int.\ a - b + (b - c) = a - c$

thm INT_FORALL_POS:

$\forall\, P{::}int \Rightarrow bool.\ (\forall\, n{::}nat.\ P\ (int\ n)) = (\forall\, i{\geq}int\ (0{::}nat).\ P\ i)$

thm INT_EXISTS_POS:

$\forall\, P{::}int \Rightarrow bool.\ (\exists\, n{::}nat.\ P\ (int\ n)) = (\exists\, i{\geq}int\ (0{::}nat).\ P\ i)$

thm INT_FORALL_ABS:

$\forall\, P{::}int \Rightarrow bool.\ (\forall\, n{::}nat.\ P\ (int\ n)) = (\forall\, x{::}int.\ P\ |x|)$

thm INT_EXISTS_ABS:

$\forall\, P{::}int \Rightarrow bool.\ (\exists\, n{::}nat.\ P\ (int\ n)) = (\exists\, x{::}int.\ P\ |x|)$

thm INT_ABS_MUL_1:

$\forall\, (x{::}int)\ y{::}int.\ (|x * y| = int\ (1{::}nat)) = (|x| = int\ (1{::}nat) \wedge |y| = int\ (1{::}nat))$

thm INT_WOP:

$(\exists\, x{\geq}int\ (0{::}nat).\ (?P{::}int \Rightarrow bool)\ x) = (\exists\, x{\geq}int\ (0{::}nat).\ ?P\ x \wedge (\forall\, y{::}int.\ int\ (0{::}nat) \leq y \wedge ?P\ y \longrightarrow x \leq y))$

thm INT_POW_conjunct0:

$(?x{::}int)^{0{::}nat} = int\ (1{::}nat)$

thm INT_POW_conjunct1:

$\forall\, n{::}nat.\ (?x{::}int)^{Suc\ n} = ?x * ?x^{n}$

thm INT_POW:

$(?x{::}int)^{0{::}nat} = int\ (1{::}nat) \wedge (\forall\, n{::}nat.\ ?x^{Suc\ n} = ?x * ?x^{n})$

thm INT_ABS:

$\forall\, x{::}int.\ |x| = (if\ int\ (0{::}nat) \leq x\ then\ x\ else\ - x)$

thm INT_GE:

$\forall\, (x{::}int)\ y{::}int.\ (y \leq x) = (y \leq x)$

thm INT_GT:

$\forall (x::int)\ y::int.\ (y < x) = (y < x)$

thm INT_LT:

$\forall (x::int)\ y::int.\ (x < y) = (\neg\ y \le x)$

thm INT_SUB:

$\forall (x::int)\ y::int.\ x - y = x + - y$

thm INT_MAX:

$\forall (x::int)\ y::int.\ max\ x\ y = (if\ x \le y\ then\ y\ else\ x)$

thm INT_MIN:

$\forall (x::int)\ y::int.\ min\ x\ y = (if\ x \le y\ then\ x\ else\ y)$

thm INT_ARCH:

$\forall (x::int)\ d::int.\ d \ne int\ (0::nat) \longrightarrow (\exists\, c::int.\ x < c * d)$

thm INT_DIVMOD_EXIST_0:

$\forall (m::int)\ n::int.\ \exists (q::int)\ r::int.\ if\ n = int\ (0::nat)\ then\ q = int\ (0::nat) \wedge r = m\ else\ int\ (0::nat) \le r \wedge r < |n| \wedge m = q * n + r$

thm DEF_div:

$HOL\_Light\_Import.div = (SOME\ q::nat \Rightarrow int \Rightarrow int \Rightarrow int.\ \forall\, \_16111::nat.\ \exists\, r::int \Rightarrow int \Rightarrow int.\ \forall (m::int)\ n::int.\ if\ n = int\ (0::nat)\ then\ q\ \_16111\ m\ n = int\ (0::nat) \wedge r\ m\ n = m\ else\ int\ (0::nat) \le r\ m\ n \wedge r\ m\ n < |n| \wedge m = q\ \_16111\ m\ n * n + r\ m\ n)\ (41::nat)$

thm DEF_rem:

$rem = (SOME\ r::nat \Rightarrow int \Rightarrow int \Rightarrow int.\ \forall (\_16112::nat)\ (m::int)\ n::int.\ if\ n = int\ (0::nat)\ then\ HOL\_Light\_Import.div\ m\ n = int\ (0::nat) \wedge r\ \_16112\ m\ n = m\ else\ int\ (0::nat) \le r\ \_16112\ m\ n \wedge r\ \_16112\ m\ n < |n| \wedge m = HOL\_Light\_Import.div\ m\ n * n + r\ \_16112\ m\ n)\ (42::nat)$

thm INT_DIVISION_0:

$\forall (m::int)\ n::int.\ if\ n = int\ (0::nat)\ then\ HOL\_Light\_Import.div\ m\ n = int\ (0::nat) \wedge rem\ m\ n = m\ else\ int\ (0::nat) \le rem\ m\ n \wedge rem\ m\ n < |n| \wedge m = HOL\_Light\_Import.div\ m\ n * n + rem\ m\ n$

thm INT_DIVISION:

$\forall (m::int)\ n::int.\ n \ne int\ (0::nat) \longrightarrow m = HOL\_Light\_Import.div\ m\ n * n + rem\ m\ n \wedge int\ (0::nat) \le rem\ m\ n \wedge rem\ m\ n < |n|$

thm INT_DIVMOD_UNIQ:

$\forall (m::int)\ (n::int)\ (q::int)\ r::int.\ m = q * n + r \wedge int\ (0::nat) \le r \wedge r < |n| \longrightarrow HOL\_Light\_Import.div\ m\ n = q \wedge rem\ m\ n = r$

thm DEF_==:

$HOL\_Light\_Import.== = (\lambda(\_16389::?'a::type)\ (\_16390::?'a::type)\ \_16391::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ \_16391\ \_16389\ \_16390)$

thm cong:

$\forall (rel::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ HOL\_Light\_Import.== x\ y\ rel = rel\ x\ y$

thm DEF_real_mod:

$real\_mod = (\lambda(\_16410::real)\ (\_16411::real)\ \_16412::real.\ \exists q::real.\ integer\ q \wedge \_16411 - \_16412 = q * \_16410)$

thm real_mod:

$\forall (x::real)\ (y::real)\ n::real.\ real\_mod\ n\ x\ y = (\exists q::real.\ integer\ q \wedge x - y = q * n)$

thm DEF_int_divides:

$int\_divides = (\lambda(\_16431::int)\ \_16432::int.\ \exists x::int.\ \_16432 = \_16431 * x)$

thm int_divides:

$\forall (b::int)\ a::int.\ int\_divides\ a\ b = (\exists x::int.\ b = a * x)$

thm DEF_int_mod:

$int\_mod = (\lambda(\_16443::int)\ (\_16444::int)\ \_16445::int.\ int\_divides\ \_16443\ (\_16444 - \_16445))$

thm int_mod:

$\forall (n::int)\ (x::int)\ y::int.\ int\_mod\ n\ x\ y = int\_divides\ n\ (x - y)$

thm int_congruent:

$\forall (x::int)\ (y::int)\ n::int.\ HOL\_Light\_Import.== x\ y\ (int\_mod\ n) = (\exists d::int.\ x - y = n * d)$

thm DEF_int_coprime:

$int\_coprime = (\lambda\_16466::int \times int.\ \exists (x::int)\ y::int.\ fst\ \_16466 * x + snd\ \_16466 * y = int\ (1::nat))$

thm int_coprime:

$\forall (a::int)\ b::int.\ int\_coprime\ (a,\ b) = (\exists (x::int)\ y::int.\ a * x + b * y = int\ (1::nat))$

thm WF_INT_MEASURE:

$\forall (P::?'a::type \Rightarrow bool)\ m::?'a::type \Rightarrow int.\ (\forall x::?'a::type.\ int\ (0::nat) \leq m\ x) \wedge (\forall x::?'a::type.\ (\forall y::?'a::type.\ m\ y < m\ x \longrightarrow P\ y) \longrightarrow P\ x) \longrightarrow (\forall x::?'a::type.\ P\ x)$

thm WF_INT_MEASURE_2:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ m::?'b::type \Rightarrow ?'a::type \Rightarrow int.\ (\forall (x::?'b::type)\ y::?'a::type.\ int\ (0::nat) \leq m\ x\ y) \wedge (\forall (x::?'b::type)\ y::?'a::type.\ (\forall (x'::?'b::type)\ y'::?'a::type.\ m\ x'\ y' < m\ x\ y \longrightarrow P\ x'\ y') \longrightarrow P\ x\ y) \longrightarrow (\forall (x::?'b::type)\ y::?'a::type.\ P\ x\ y)$

thm INT_GCD_EXISTS:

$\forall (a::int)\ b::int.\ \exists\, d::int.\ int\_divides\ d\ a \wedge int\_divides\ d\ b \wedge (\exists (x::int)\ y::int.\ d = a * x + b * y)$

thm INT_GCD_EXISTS_POS:

$\forall (a::int)\ b::int.\ \exists\, d \geq int\ (0::nat).\ int\_divides\ d\ a \wedge int\_divides\ d\ b \wedge (\exists (x::int)\ y::int.\ d = a * x + b * y)$

thm DEF_int_gcd:

$int\_gcd = (SOME\ d::nat \Rightarrow int \times int \Rightarrow int.\ \forall (\_16775::nat)\ (a::int)\ b::int.\ int\ (0::nat) \leq d\ \_16775\ (a,\ b) \wedge int\_divides\ (d\ \_16775\ (a,\ b))\ a \wedge int\_divides\ (d\ \_16775\ (a,\ b))\ b \wedge (\exists (x::int)\ y::int.\ d\ \_16775\ (a,\ b) = a * x + b * y))\ (43::nat)$

thm int_gcd:

$\forall (a::int)\ b::int.\ int\ (0::nat) \leq int\_gcd\ (a,\ b) \wedge int\_divides\ (int\_gcd\ (a,\ b))\ a \wedge int\_divides\ (int\_gcd\ (a,\ b))\ b \wedge (\exists (x::int)\ y::int.\ int\_gcd\ (a,\ b) = a * x + b * y)$

thm DEF_num_of_int:

$num\_of\_int = (\lambda\_16776::int.\ SOME\ n::nat.\ int\ n = \_16776)$

thm num_of_int:

$\forall x::int.\ num\_of\_int\ x = (SOME\ n::nat.\ int\ n = x)$

thm NUM_OF_INT_OF_NUM:

$\forall n::nat.\ num\_of\_int\ (int\ n) = n$

thm INT_OF_NUM_OF_INT:

$\forall x \geq int\ (0::nat).\ int\ (num\_of\_int\ x) = x$

thm NUM_OF_INT:

$\forall x::int.\ (int\ (0::nat) \leq x) = (int\ (num\_of\_int\ x) = x)$

thm DEF_num_divides:

$num\_divides = (\lambda(\_16808::nat)\ \_16809::nat.\ int\_divides\ (int\ \_16808)\ (int\ \_16809))$

thm num_divides:

$\forall (a::nat)\ b::nat.\ num\_divides\ a\ b = int\_divides\ (int\ a)\ (int\ b)$

thm DEF_num_mod:

$num\_mod = (\lambda(\_16820{::}nat)\ (\_16821{::}nat)\ \_16822{::}nat.\ int\_mod\ (int\ \_16820)\ (int\ \_16821)\ (int\ \_16822))$

thm num_mod:

$\forall (n{::}nat)\ (x{::}nat)\ y{::}nat.\ num\_mod\ n\ x\ y = int\_mod\ (int\ n)\ (int\ x)\ (int\ y)$

thm num_congruent:

$\forall (x{::}nat)\ (y{::}nat)\ n{::}nat.\ HOL\_Light\_Import.{==}\ x\ y\ (num\_mod\ n) = HOL\_Light\_Import.{==}\ (int\ x)\ (int\ y)\ (int\_mod\ (int\ n))$

thm DEF_num_coprime:

$num\_coprime = (\lambda\_16841{::}nat \times nat.\ int\_coprime\ (int\ (fst\ \_16841),\ int\ (snd\ \_16841)))$

thm num_coprime:

$\forall (a{::}nat)\ b{::}nat.\ num\_coprime\ (a,\ b) = int\_coprime\ (int\ a,\ int\ b)$

thm DEF_num_gcd:

$num\_gcd = (\lambda\_16850{::}nat \times nat.\ num\_of\_int\ (int\_gcd\ (int\ (fst\ \_16850),\ int\ (snd\ \_16850))))$

thm num_gcd:

$\forall (a{::}nat)\ b{::}nat.\ num\_gcd\ (a,\ b) = num\_of\_int\ (int\_gcd\ (int\ a,\ int\ b))$

thm NUM_GCD:

$\forall (a{::}nat)\ b{::}nat.\ int\ (num\_gcd\ (a,\ b)) = int\_gcd\ (int\ a,\ int\ b)$

thm DEF_IN:

$IN = (\lambda(\_16869{::}?'a{::}type)\ \_16870{::}?'a{::}type \Rightarrow bool.\ \_16870\ \_16869)$

thm IN:

$\forall (P{::}?'a{::}type \Rightarrow bool)\ x{::}?'a{::}type.\ IN\ x\ P = P\ x$

thm EXTENSION:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ (s = t) = (\forall x{::}?'a{::}type.\ IN\ x\ s = IN\ x\ t)$

thm DEF_GSPEC:

$GSPEC = (\lambda\_16881{::}?'a{::}type \Rightarrow bool.\ \_16881)$

thm GSPEC:

$\forall p{::}?'a{::}type \Rightarrow bool.\ GSPEC\ p = p$

thm DEF_SETSPEC:

$SETSPEC = (\lambda(\_16886{::}?'a{::}type)\ (\_16887{::}bool)\ \_16888{::}?'a{::}type.\ \_16887 \wedge \_16886 = \_16888)$

thm SETSPEC:

$\forall (P::bool)\ (v::?'a::type)\ t::?'a::type.\ SETSPEC\ v\ P\ t = (P \wedge v = t)$

thm IN_ELIM_THM_conjunct4:

$\forall (p::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ p = p\ x$

thm IN_ELIM_THM_conjunct3:

$\forall (p::?'a::type \Rightarrow bool)\ x::?'a::type.\ GSPEC\ (\lambda v::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ v\ (p\ y)\ y)\ x = p\ x$

thm IN_ELIM_THM_conjunct2:

$\forall (P::(bool \Rightarrow ?'a::type \Rightarrow bool) \Rightarrow bool)\ x::?'a::type.\ GSPEC\ (\lambda v::?'a::type.\ P\ (SETSPEC\ v))\ x = P\ (\lambda (p::bool)\ t::?'a::type.\ p \wedge x = t)$

thm IN_ELIM_THM_conjunct1:

$\forall (p::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (GSPEC\ (\lambda v::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ v\ (p\ y)\ y)) = p\ x$

thm IN_ELIM_THM_conjunct0:

$\forall (P::(bool \Rightarrow ?'a::type \Rightarrow bool) \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (GSPEC\ (\lambda v::?'a::type.\ P\ (SETSPEC\ v))) = P\ (\lambda (p::bool)\ t::?'a::type.\ p \wedge x = t)$

thm IN_ELIM_THM:

$(\forall (P::(bool \Rightarrow ?'e::type \Rightarrow bool) \Rightarrow bool)\ x::?'e::type.\ IN\ x\ (GSPEC\ (\lambda v::?'e::type.\ P\ (SETSPEC\ v))) = P\ (\lambda (p::bool)\ t::?'e::type.\ p \wedge x = t)) \wedge (\forall (p::?'d::type \Rightarrow bool)\ x::?'d::type.\ IN\ x\ (GSPEC\ (\lambda v::?'d::type.\ \exists y::?'d::type.\ SETSPEC\ v\ (p\ y)\ y)) = p\ x) \wedge (\forall (P::(bool \Rightarrow ?'c::type \Rightarrow bool) \Rightarrow bool)\ x::?'c::type.\ GSPEC\ (\lambda v::?'c::type.\ P\ (SETSPEC\ v))\ x = P\ (\lambda (p::bool)\ t::?'c::type.\ p \wedge x = t)) \wedge (\forall (p::?'b::type \Rightarrow bool)\ x::?'b::type.\ GSPEC\ (\lambda v::?'b::type.\ \exists y::?'b::type.\ SETSPEC\ v\ (p\ y)\ y)\ x = p\ x) \wedge (\forall (p::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ p = p\ x)$

thm EMPTY:

$EMPTY = (\lambda x::?'a::type.\ False)$

thm DEF_INSERT:

$INSERT = (\lambda (\_16925::?'a::type)\ (\_16926::?'a::type \Rightarrow bool)\ y::?'a::type.\ IN\ y\ \_16926 \vee y = \_16925)$

thm INSERT_DEF:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ INSERT\ x\ s = (\lambda y::?'a::type.\ IN\ y\ s \vee y = x)$

thm UNIV:

$HOL\_Light\_Import.UNIV = (\lambda x::?'a::type.\ True)$

thm DEF_UNION:

$HOL\_Light\_Import.UNION = (\lambda(\_16937::?'a::type \Rightarrow bool)\ \_16938::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%0::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%0\ (IN\ x\ \_16937 \lor IN\ x\ \_16938)\ x))$

thm UNION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ s\ t = GSPEC\ (\lambda GEN\%PVAR\%0::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%0\ (IN\ x\ s \lor IN\ x\ t)\ x)$

thm DEF_UNIONS:

$UNIONS = (\lambda\_16949::(?'a::type \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%1\ (\exists u::?'a::type \Rightarrow bool.\ IN\ u\ \_16949 \land IN\ x\ u)\ x))$

thm UNIONS:

$\forall s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ UNIONS\ s = GSPEC\ (\lambda GEN\%PVAR\%1::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%1\ (\exists u::?'a::type \Rightarrow bool.\ IN\ u\ s \land IN\ x\ u)\ x)$

thm DEF_INTER:

$HOL\_Light\_Import.INTER = (\lambda(\_16954::?'a::type \Rightarrow bool)\ \_16955::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%2\ (IN\ x\ \_16954 \land IN\ x\ \_16955)\ x))$

thm INTER:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ t = GSPEC\ (\lambda GEN\%PVAR\%2::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%2\ (IN\ x\ s \land IN\ x\ t)\ x)$

thm DEF_INTERS:

$INTERS = (\lambda\_16966::(?'a::type \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%3::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%3\ (\forall u::?'a::type \Rightarrow bool.\ IN\ u\ \_16966 \longrightarrow IN\ x\ u)\ x))$

thm INTERS:

$\forall s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ INTERS\ s = GSPEC\ (\lambda GEN\%PVAR\%3::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%3\ (\forall u::?'a::type \Rightarrow bool.\ IN\ u\ s \longrightarrow IN\ x\ u)\ x)$

thm DEF_DIFF:

$DIFF = (\lambda(\_16971::?'a::type \Rightarrow bool)\ \_16972::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%4::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%4\ (IN\ x\ \_16971 \land \neg\ IN\ x\ \_16972)\ x))$

thm DIFF:

$\forall\,(s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ DIFF\ s\ t = GSPEC\ (\lambda GEN\%PVAR\%4{::}?'a{::}type.$
$\exists\,x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%4\ \ (IN\ x\ s\ \wedge \neg\ IN\ x\ t)\ x)$

thm INSERT:

$INSERT\ (?x{::}?'a{::}type)\ (?s{::}?'a{::}type \Rightarrow bool) = GSPEC\ (\lambda GEN\%PVAR\%5{::}?'a{::}type.$
$\exists\,y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%5\ \ (IN\ y\ ?s\ \vee\ y\ =\ ?x)\ y)$

thm DEF_DELETE:

$DELETE = (\lambda(\_16983{::}?'a{::}type \Rightarrow bool)\ \_16984{::}?'a{::}type.\ GSPEC\ (\lambda GEN\%PVAR\%6{::}?'a{::}type.$
$\exists\,y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%6\ \ (IN\ y\ \_16983\ \wedge\ y \neq \_16984)\ y))$

thm DELETE:

$\forall\,(s{::}?'a{::}type \Rightarrow bool)\ x{::}?'a{::}type.\ DELETE\ s\ x = GSPEC\ (\lambda GEN\%PVAR\%6{::}?'a{::}type.$
$\exists\,y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%6\ \ (IN\ y\ s\ \wedge\ y \neq x)\ y)$

thm DEF_SUBSET:

$SUBSET = (\lambda(\_16995{::}?'a{::}type \Rightarrow bool)\ \_16996{::}?'a{::}type \Rightarrow bool.\ \forall\,x{::}?'a{::}type.$
$IN\ x\ \_16995\ \longrightarrow IN\ x\ \_16996)$

thm SUBSET:

$\forall\,(s{::}?'a{::}type\ \Rightarrow\ bool)\ \ t{::}?'a{::}type\ \Rightarrow\ bool.\ \ SUBSET\ \ s\ \ t\ =\ (\forall\,x{::}?'a{::}type.\ \ IN$
$x\ s\ \longrightarrow\ IN\ x\ t)$

thm DEF_PSUBSET:

$PSUBSET = (\lambda(\_17007{::}?'a{::}type \Rightarrow bool)\ \_17008{::}?'a{::}type \Rightarrow bool.\ SUBSET$
$\_17007\ \_17008\ \wedge\ \_17007 \neq \_17008)$

thm PSUBSET:

$\forall\,(s{::}?'a{::}type\ \Rightarrow\ bool)\ \ t{::}?'a{::}type\ \Rightarrow\ bool.\ \ PSUBSET\ s\ t\ =\ (SUBSET\ \ s\ \ t\ \wedge\ \ s$
$\neq\ t)$

thm DEF_DISJOINT:

$DISJOINT = (\lambda(\_17019{::}?'a{::}type \Rightarrow bool)\ \_17020{::}?'a{::}type \Rightarrow bool.\ HOL\_Light\_Import.INTER$
$\_17019\ \_17020\ =\ EMPTY)$

thm DISJOINT:

$\forall\,(s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ DISJOINT\ s\ t = (HOL\_Light\_Import.INTER$
$s\ t\ =\ EMPTY)$

thm DEF_SING:

$SING = (\lambda\_17031{::}?'a{::}type \Rightarrow bool.\ \exists\,x{::}?'a{::}type.\ \_17031 = INSERT\ x\ EMPTY)$

thm SING:

$\forall\,s{::}?'a{::}type\ \Rightarrow\ bool.\ \ SING\ \ s\ =\ (\exists\,x{::}?'a{::}type.\ \ s\ =\ INSERT\ \ x\ \ EMPTY)$

thm DEF_FINITE:

$FINITE = (\lambda a{::}?'a{::}type \Rightarrow bool.\ \forall FINITE'{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall a{::}?'a{::}type \Rightarrow bool.\ a = EMPTY \vee (\exists (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ a = INSERT\ x\ s \wedge FINITE'\ s) \longrightarrow FINITE'\ a) \longrightarrow FINITE'\ a)$

thm FINITE_RULES:

$FINITE\ EMPTY \wedge (\forall (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (INSERT\ x\ s))$

thm FINITE_CASES:

$\forall a{::}?'a{::}type \Rightarrow bool.\ FINITE\ a = (a = EMPTY \vee (\exists (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ a = INSERT\ x\ s \wedge FINITE\ s))$

thm FINITE_INDUCT:

$\forall FINITE'{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ FINITE'\ EMPTY \wedge (\forall (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE'\ s \longrightarrow FINITE'\ (INSERT\ x\ s)) \longrightarrow (\forall a{::}?'a{::}type \Rightarrow bool.\ FINITE\ a \longrightarrow FINITE'\ a)$

thm DEF_INFINITE:

$INFINITE = (\lambda\_17040{::}?'a{::}type \Rightarrow bool.\ \neg\ FINITE\ \_17040)$

thm INFINITE:

$\forall s{::}?'a{::}type \Rightarrow bool.\ INFINITE\ s = (\neg\ FINITE\ s)$

thm DEF_IMAGE:

$IMAGE = (\lambda(\_17045{::}?'b{::}type \Rightarrow ?'a{::}type)\ \_17046{::}?'b{::}type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%7{::}?'a{::}type.\ \exists y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%7\ (\exists x{::}?'b{::}type.\ IN\ x\ \_17046 \wedge y = \_17045\ x)\ y))$

thm IMAGE:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ f{::}?'b{::}type \Rightarrow ?'a{::}type.\ IMAGE\ f\ s = GSPEC\ (\lambda GEN\%PVAR\%7{::}?'a{::}type.\ \exists y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%7\ (\exists x{::}?'b{::}type.\ IN\ x\ s \wedge y = f\ x)\ y)$

thm DEF_INJ:

$INJ = (\lambda(\_17057{::}?'b{::}type \Rightarrow ?'a{::}type)\ (\_17058{::}?'b{::}type \Rightarrow bool)\ \_17059{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'b{::}type.\ IN\ x\ \_17058 \longrightarrow IN\ (\_17057\ x)\ \_17059) \wedge (\forall (x{::}?'b{::}type)\ y{::}?'b{::}type.\ IN\ x\ \_17058 \wedge IN\ y\ \_17058 \wedge \_17057\ x = \_17057\ y \longrightarrow x = y))$

thm INJ:

$\forall (t{::}?'b{::}type \Rightarrow bool)\ (s{::}?'a{::}type \Rightarrow bool)\ f{::}?'a{::}type \Rightarrow ?'b{::}type.\ INJ\ f\ s\ t = ((\forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall (x{::}?'a{::}type)\ y{::}?'a{::}type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y))$

thm DEF_SURJ:

$SURJ = (\lambda(\_17078{::}?'b{::}type \Rightarrow ?'a{::}type)\ (\_17079{::}?'b{::}type \Rightarrow bool)\ \_17080{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'b{::}type.\ IN\ x\ \_17079 \longrightarrow IN\ (\_17078\ x)\ \_17080) \wedge (\forall x{::}?'a{::}type.\ IN\ x\ \_17080 \longrightarrow (\exists y{::}?'b{::}type.\ IN\ y\ \_17079 \wedge \_17078\ y = x)))$

thm SURJ:

$\forall (t::?'b::type \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'b::type.\ SURJ\ f\ s\ t = ((\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall x::?'b::type.\ IN\ x\ t \longrightarrow (\exists y::?'a::type.\ IN\ y\ s \wedge f\ y = x)))$

thm DEF_BIJ:

$BIJ = (\lambda(\_17099::?'b::type \Rightarrow ?'a::type)\ (\_17100::?'b::type \Rightarrow bool)\ \_17101::?'a::type \Rightarrow bool.\ INJ\ \_17099\ \_17100\ \_17101 \wedge SURJ\ \_17099\ \_17100\ \_17101)$

thm BIJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ BIJ\ f\ s\ t = (INJ\ f\ s\ t \wedge SURJ\ f\ s\ t)$

thm DEF_CHOICE:

$CHOICE = (\lambda\_17120::?'a::type \Rightarrow bool.\ SOME\ x::?'a::type.\ IN\ x\ \_17120)$

thm CHOICE:

$\forall s::?'a::type \Rightarrow bool.\ CHOICE\ s = (SOME\ x::?'a::type.\ IN\ x\ s)$

thm DEF_REST:

$REST = (\lambda\_17125::?'a::type \Rightarrow bool.\ DELETE\ \_17125\ (CHOICE\ \_17125))$

thm REST:

$\forall s::?'a::type \Rightarrow bool.\ REST\ s = DELETE\ s\ (CHOICE\ s)$

thm NOT_IN_EMPTY:

$\forall x::?'a::type.\ \neg\ IN\ x\ EMPTY$

thm IN_UNIV:

$\forall x::?'a::type.\ IN\ x\ HOL\_Light\_Import.UNIV$

thm IN_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (HOL\_Light\_Import.UNION\ s\ t) = (IN\ x\ s \vee IN\ x\ t)$

thm IN_UNIONS:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (UNIONS\ s) = (\exists t::?'a::type \Rightarrow bool.\ IN\ t\ s \wedge IN\ x\ t)$

thm IN_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ t) = (IN\ x\ s \wedge IN\ x\ t)$

thm IN_INTERS:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (INTERS\ s) = (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow IN\ x\ t)$

thm IN_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (DIFF\ s\ t) = (IN\ x\ s \wedge \neg\ IN\ x\ t)$

thm IN_INSERT:

$\forall (x::?'a::type)\ (y::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ x\ (INSERT\ y\ s) = (x = y \vee IN\ x\ s)$

thm IN_DELETE:

$\forall (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (DELETE\ s\ y) = (IN\ x\ s \wedge x \neq y)$

thm IN_SING:

$\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ (INSERT\ y\ EMPTY) = (x = y)$

thm IN_IMAGE:

$\forall (y::?'b::type)\ (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'b::type.\ IN\ y\ (IMAGE\ f\ s) = (\exists x::?'a::type.\ y = f\ x \wedge IN\ x\ s)$

thm IN_REST:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ x\ (REST\ s) = (IN\ x\ s \wedge x \neq CHOICE\ s)$

thm FORALL_IN_INSERT:

$\forall (P::?'a::type \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ (INSERT\ a\ s) \longrightarrow P\ x) = (P\ a \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow P\ x))$

thm EXISTS_IN_INSERT:

$\forall (P::?'a::type \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ IN\ x\ (INSERT\ a\ s) \wedge P\ x) = (P\ a \vee (\exists x::?'a::type.\ IN\ x\ s \wedge P\ x))$

thm CHOICE_DEF:

$\forall s::?'a::type \Rightarrow bool.\ s \neq EMPTY \longrightarrow IN\ (CHOICE\ s)\ s$

thm NOT_EQUAL_SETS:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (s \neq t) = (\exists x::?'a::type.\ IN\ x\ t = (\neg\ IN\ x\ s))$

thm MEMBER_NOT_EMPTY:

$\forall s::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ IN\ x\ s) = (s \neq EMPTY)$

thm UNIV_NOT_EMPTY:

$HOL\_Light\_Import.UNIV \neq EMPTY$

thm EMPTY_NOT_UNIV:

$EMPTY \neq HOL\_Light\_Import.UNIV$

thm EQ_UNIV:

$(\forall x::?'a::type.\ IN\ x\ (?s::?'a::type \Rightarrow bool)) = (?s = HOL\_Light\_Import.UNIV)$

thm SUBSET_TRANS:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \land SUBSET\ t\ u \longrightarrow SUBSET\ s\ u$

thm SUBSET_REFL:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ s\ s$

thm SUBSET_ANTISYM:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \land SUBSET\ t\ s \longrightarrow s = t$

thm SUBSET_ANTISYM_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (SUBSET\ s\ t \land SUBSET\ t\ s) = (s = t)$

thm EMPTY_SUBSET:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ EMPTY\ s$

thm SUBSET_EMPTY:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ s\ EMPTY = (s = EMPTY)$

thm SUBSET_UNIV:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ s\ HOL\_Light\_Import.UNIV$

thm UNIV_SUBSET:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ HOL\_Light\_Import.UNIV\ s = (s = HOL\_Light\_Import.UNIV)$

thm SING_SUBSET:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ SUBSET\ (INSERT\ x\ EMPTY)\ s = IN\ x\ s$

thm PSUBSET_TRANS:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ PSUBSET\ s\ t \land PSUBSET\ t\ u \longrightarrow PSUBSET\ s\ u$

thm PSUBSET_SUBSET_TRANS:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ PSUBSET\ s\ t \land SUBSET\ t\ u \longrightarrow PSUBSET\ s\ u$

thm SUBSET_PSUBSET_TRANS:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \land PSUBSET\ t\ u \longrightarrow PSUBSET\ s\ u$

thm PSUBSET_IRREFL:

$\forall s::?'a::type \Rightarrow bool. \neg PSUBSET\ s\ s$

thm NOT_PSUBSET_EMPTY:

$\forall s::?'a::type \Rightarrow bool. \neg PSUBSET\ s\ EMPTY$

thm NOT_UNIV_PSUBSET:

$\forall s::?'a::type \Rightarrow bool. \neg PSUBSET\ HOL\_Light\_Import.UNIV\ s$

thm PSUBSET_UNIV:

$\forall s::?'a::type \Rightarrow bool. PSUBSET\ s\ HOL\_Light\_Import.UNIV = (\exists x::?'a::type. \neg IN\ x\ s)$

thm PSUBSET_ALT:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool. PSUBSET\ s\ t = (SUBSET\ s\ t \wedge (\exists a::?'a::type. IN\ a\ t \wedge \neg IN\ a\ s))$

thm UNION_ASSOC:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ (HOL\_Light\_Import.UNION\ s\ t)\ u = HOL\_Light\_Import.UNION\ s\ (HOL\_Light\_Import.UNION\ t\ u)$

thm UNION_IDEMPOT:

$\forall s::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ s\ s = s$

thm UNION_COMM:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ s\ t = HOL\_Light\_Import.UNION\ t\ s$

thm SUBSET_UNION:

$(\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool. SUBSET\ s\ (HOL\_Light\_Import.UNION\ s\ t)) \wedge (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool. SUBSET\ s\ (HOL\_Light\_Import.UNION\ t\ s))$

thm SUBSET_UNION_ABSORPTION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool. SUBSET\ s\ t = (HOL\_Light\_Import.UNION\ s\ t = t)$

thm UNION_EMPTY:

$(\forall s::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ EMPTY\ s = s) \wedge (\forall s::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ s\ EMPTY = s)$

thm UNION_UNIV:

$(\forall s::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ HOL\_Light\_Import.UNIV\ s = HOL\_Light\_Import.UNIV) \wedge (\forall s::?'a::type \Rightarrow bool. HOL\_Light\_Import.UNION\ s\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV)$

thm EMPTY_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (HOL\_Light\_Import.UNION\ s\ t = EMPTY) = (s = EMPTY \wedge t = EMPTY)$

thm UNION_SUBSET:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.UNION\ s\ t)\ u = (SUBSET\ s\ u \wedge SUBSET\ t\ u)$

thm INTER_ASSOC:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (HOL\_Light\_Import.INTER\ s\ t)\ u = HOL\_Light\_Import.INTER\ s\ (HOL\_Light\_Import.INTER\ t\ u)$

thm INTER_IDEMPOT:

$\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ s = s$

thm INTER_COMM:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ t = HOL\_Light\_Import.INTER\ t\ s$

thm INTER_SUBSET:

$(\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.INTER\ s\ t)\ s) \wedge (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.INTER\ t\ s)\ s)$

thm SUBSET_INTER_ABSORPTION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t = (HOL\_Light\_Import.INTER\ s\ t = s)$

thm INTER_EMPTY:

$(\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ EMPTY\ s = EMPTY) \wedge (\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ EMPTY = EMPTY)$

thm INTER_UNIV:

$(\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ HOL\_Light\_Import.UNIV\ s = s) \wedge (\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ HOL\_Light\_Import.UNIV = s)$

thm SUBSET_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SUBSET\ s\ (HOL\_Light\_Import.INTER\ t\ u) = (SUBSET\ s\ t \wedge SUBSET\ s\ u)$

thm UNION_OVER_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ (HOL\_Light\_Import.UNION\ t\ u) = HOL\_Light\_Import.UNION\ (HOL\_Light\_Import.INTER\ s\ t)\ (HOL\_Light\_Import.INTER\ s\ u)$

thm INTER_OVER_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ s\ (HOL\_Light\_Import.INTER\ t\ u) = HOL\_Light\_Import.INTER\ (HOL\_Light\_Import.UNION\ s\ t)\ (HOL\_Light\_Import.UNION\ s\ u)$

thm IN_DISJOINT:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ DISJOINT\ s\ t = (\neg (\exists x::?'a::type.\ IN\ x\ s \wedge IN\ x\ t))$

thm DISJOINT_SYM:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ DISJOINT\ s\ t = DISJOINT\ t\ s$

thm DISJOINT_EMPTY:

$\forall s::?'a::type \Rightarrow bool.\ DISJOINT\ EMPTY\ s \wedge DISJOINT\ s\ EMPTY$

thm DISJOINT_EMPTY_REFL:

$\forall s::?'a::type \Rightarrow bool.\ (s = EMPTY) = DISJOINT\ s\ s$

thm DISJOINT_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ DISJOINT\ (HOL\_Light\_Import.UNION\ s\ t)\ u = (DISJOINT\ s\ u \wedge DISJOINT\ t\ u)$

thm DIFF_EMPTY:

$\forall s::?'a::type \Rightarrow bool.\ DIFF\ s\ EMPTY = s$

thm EMPTY_DIFF:

$\forall s::?'a::type \Rightarrow bool.\ DIFF\ EMPTY\ s = EMPTY$

thm DIFF_UNIV:

$\forall s::?'a::type \Rightarrow bool.\ DIFF\ s\ HOL\_Light\_Import.UNIV = EMPTY$

thm DIFF_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ DIFF\ (DIFF\ s\ t)\ t = DIFF\ s\ t$

thm DIFF_EQ_EMPTY:

$\forall s::?'a::type \Rightarrow bool.\ DIFF\ s\ s = EMPTY$

thm SUBSET_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (DIFF\ s\ t)\ s$

thm COMPONENT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ x\ (INSERT\ x\ s)$

thm DECOMPOSITION:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ s = (\exists t::?'a::type \Rightarrow bool.\ s = INSERT\ x\ t \wedge \neg IN\ x\ t)$

thm SET_CASES:

$\forall s::?'a::type \Rightarrow bool.\ s = EMPTY \vee (\exists (x::?'a::type)\ t::?'a::type \Rightarrow bool.\ s = INSERT\ x\ t \wedge \neg\ IN\ x\ t)$

thm ABSORPTION:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ x\ s = (INSERT\ x\ s = s)$

thm INSERT_INSERT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ INSERT\ x\ (INSERT\ x\ s) = INSERT\ x\ s$

thm INSERT_COMM:

$\forall (x::?'a::type)\ (y::?'a::type)\ s::?'a::type \Rightarrow bool.\ INSERT\ x\ (INSERT\ y\ s) = INSERT\ y\ (INSERT\ x\ s)$

thm INSERT_UNIV:

$\forall x::?'a::type.\ INSERT\ x\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm NOT_INSERT_EMPTY:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ INSERT\ x\ s \neq EMPTY$

thm NOT_EMPTY_INSERT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ EMPTY \neq INSERT\ x\ s$

thm INSERT_UNION:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (INSERT\ x\ s)\ t = (if\ IN\ x\ t\ then\ HOL\_Light\_Import.UNION\ s\ t\ else\ INSERT\ x\ (HOL\_Light\_Import.UNION\ s\ t))$

thm INSERT_UNION_EQ:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (INSERT\ x\ s)\ t = INSERT\ x\ (HOL\_Light\_Import.UNION\ s\ t)$

thm INSERT_INTER:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (INSERT\ x\ s)\ t = (if\ IN\ x\ t\ then\ INSERT\ x\ (HOL\_Light\_Import.INTER\ s\ t)\ else\ HOL\_Light\_Import.INTER\ s\ t)$

thm DISJOINT_INSERT:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ DISJOINT\ (INSERT\ x\ s)\ t = (DISJOINT\ s\ t \wedge \neg\ IN\ x\ t)$

thm INSERT_SUBSET:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (INSERT\ x\ s)\ t = (IN\ x\ t \wedge SUBSET\ s\ t)$

thm SUBSET_INSERT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ \neg\ IN\ x\ s \longrightarrow (\forall\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (INSERT\ x\ t) = SUBSET\ s\ t)$

thm INSERT_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ DIFF\ (INSERT\ x\ s)\ t = (if\ IN\ x\ t\ then\ DIFF\ s\ t\ else\ INSERT\ x\ (DIFF\ s\ t))$

thm INSERT_AC_conjunct0:

$INSERT\ (?x::?'a::type)\ (INSERT\ (?y::?'a::type)\ (?s::?'a::type \Rightarrow bool)) = INSERT\ ?y\ (INSERT\ ?x\ ?s)$

thm INSERT_AC_conjunct1:

$INSERT\ (?x::?'a::type)\ (INSERT\ ?x\ (?s::?'a::type \Rightarrow bool)) = INSERT\ ?x\ ?s$

thm INSERT_AC:

$INSERT\ (?x::?'a::type)\ (INSERT\ (?y::?'a::type)\ (?s::?'a::type \Rightarrow bool)) = INSERT\ ?y\ (INSERT\ ?x\ ?s) \wedge INSERT\ ?x\ (INSERT\ ?x\ ?s) = INSERT\ ?x\ ?s$

thm INTER_ACI:

$HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ (?q::?'a::type \Rightarrow bool) = HOL\_Light\_Import.INTER\ ?q\ ?p \wedge HOL\_Light\_Import.INTER\ (HOL\_Light\_Import.INTER\ ?p\ ?q)\ (?r::?'a::type \Rightarrow bool) = HOL\_Light\_Import.INTER\ ?p\ (HOL\_Light\_Import.INTER\ ?q\ ?r) \wedge HOL\_Light\_Import.INTER\ ?p\ (HOL\_Light\_Import.INTER\ ?q\ ?r) = HOL\_Light\_Import.INTER\ ?q\ (HOL\_Light\_Import.INTER\ ?p\ ?r) \wedge HOL\_Light\_Import.INTER\ ?p\ ?p = ?p \wedge HOL\_Light\_Import.INTER\ ?p\ (HOL\_Light\_Import.INTER\ ?p\ ?q) = HOL\_Light\_Import.INTER\ ?p\ ?q$

thm UNION_ACI:

$HOL\_Light\_Import.UNION\ (?p::?'a::type \Rightarrow bool)\ (?q::?'a::type \Rightarrow bool) = HOL\_Light\_Import.UNION\ ?q\ ?p \wedge HOL\_Light\_Import.UNION\ (HOL\_Light\_Import.UNION\ ?p\ ?q)\ (?r::?'a::type \Rightarrow bool) = HOL\_Light\_Import.UNION\ ?p\ (HOL\_Light\_Import.UNION\ ?q\ ?r) \wedge HOL\_Light\_Import.UNION\ ?p\ (HOL\_Light\_Import.UNION\ ?q\ ?r) = HOL\_Light\_Import.UNION\ ?q\ (HOL\_Light\_Import.UNION\ ?p\ ?r) \wedge HOL\_Light\_Import.UNION\ ?p\ ?p = ?p \wedge HOL\_Light\_Import.UNION\ ?p\ (HOL\_Light\_Import.UNION\ ?p\ ?q) = HOL\_Light\_Import.UNION\ ?p\ ?q$

thm DELETE_NON_ELEMENT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ (\neg\ IN\ x\ s) = (DELETE\ s\ x = s)$

thm IN_DELETE_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ x'::?'a::type.\ (IN\ x\ s = IN\ x'\ s) = (IN\ x\ (DELETE\ s\ x') = IN\ x'\ (DELETE\ s\ x))$

thm EMPTY_DELETE:

$\forall\ x::?'a::type.\ DELETE\ EMPTY\ x = EMPTY$

thm DELETE_DELETE:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ DELETE\ (DELETE\ s\ x)\ x = DELETE\ s\ x$

thm DELETE_COMM:

$\forall (x::?'a::type)\ (y::?'a::type)\ s::?'a::type \Rightarrow bool.\ DELETE\ (DELETE\ s\ x)\ y = DELETE\ (DELETE\ s\ y)\ x$

thm DELETE_SUBSET:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ SUBSET\ (DELETE\ s\ x)\ s$

thm SUBSET_DELETE:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (DELETE\ t\ x) = (\neg\ IN\ x\ s \wedge SUBSET\ s\ t)$

thm SUBSET_INSERT_DELETE:

$\forall (x::?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (INSERT\ x\ t) = SUBSET\ (DELETE\ s\ x)\ t$

thm DIFF_INSERT:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ DIFF\ s\ (INSERT\ x\ t) = DIFF\ (DELETE\ s\ x)\ t$

thm PSUBSET_INSERT_SUBSET:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ PSUBSET\ s\ t = (\exists x::?'a::type.\ \neg\ IN\ x\ s \wedge SUBSET\ (INSERT\ x\ s)\ t)$

thm PSUBSET_MEMBER:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ PSUBSET\ s\ t = (SUBSET\ s\ t \wedge (\exists y::?'a::type.\ IN\ y\ t \wedge \neg\ IN\ y\ s))$

thm DELETE_INSERT:

$\forall (x::?'a::type)\ (y::?'a::type)\ s::?'a::type \Rightarrow bool.\ DELETE\ (INSERT\ x\ s)\ y = (if\ x = y\ then\ DELETE\ s\ y\ else\ INSERT\ x\ (DELETE\ s\ y))$

thm INSERT_DELETE:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ x\ s \longrightarrow INSERT\ x\ (DELETE\ s\ x) = s$

thm DELETE_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ HOL\_Light\_Import.INTER\ (DELETE\ s\ x)\ t = DELETE\ (HOL\_Light\_Import.INTER\ s\ t)\ x$

thm DISJOINT_DELETE_SYM:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ DISJOINT\ (DELETE\ s\ x)\ t = DISJOINT\ (DELETE\ t\ x)\ s$

thm UNIONS_0:

UNIONS EMPTY = *EMPTY*

thm UNIONS_1:

UNIONS (*INSERT* (*?s*::*?'a*::*type* ⇒ *bool*) *EMPTY*) = *?s*

thm UNIONS_2:

UNIONS (*INSERT* (*?s*::*?'a*::*type* ⇒ *bool*) (*INSERT* (*?t*::*?'a*::*type* ⇒ *bool*) *EMPTY*)) = *HOL_Light_Import*.*UNION* *?s* *?t*

thm UNIONS_INSERT:

UNIONS (*INSERT* (*?s*::*?'a*::*type* ⇒ *bool*) (*?u*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) = *HOL_Light_Import*.*UNION* *?s* (*UNIONS* *?u*)

thm FORALL_IN_UNIONS:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. (∀ *x*::*?'a*::*type*. *IN* *x* (*UNIONS* *s*) ⟶ *P* *x*) = (∀ (*t*::*?'a*::*type* ⇒ *bool*) *x*::*?'a*::*type*. *IN* *t* *s* ∧ *IN* *x* *t* ⟶ *P* *x*)

thm EXISTS_IN_UNIONS:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. (∃ *x*::*?'a*::*type*. *IN* *x* (*UNIONS* *s*) ∧ *P* *x*) = (∃ (*t*::*?'a*::*type* ⇒ *bool*) *x*::*?'a*::*type*. *IN* *t* *s* ∧ *IN* *x* *t* ∧ *P* *x*)

thm EMPTY_UNIONS:

∀ *s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. (*UNIONS* *s* = *EMPTY*) = (∀ *t*::*?'a*::*type* ⇒ *bool*. *IN* *t* *s* ⟶ *t* = *EMPTY*)

thm INTER_UNIONS:

(∀ (*s*::(*?'b*::*type* ⇒ *bool*) ⇒ *bool*) *t*::*?'b*::*type* ⇒ *bool*. *HOL_Light_Import*.*INTER* (*UNIONS* *s*) *t* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%8*::*?'b*::*type* ⇒ *bool*. ∃ *x*::*?'b*::*type* ⇒ *bool*. *SETSPEC* *GEN%PVAR%8* (*IN* *x* *s*) (*HOL_Light_Import*.*INTER* *x* *t*)))) ∧ (∀ (*s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *HOL_Light_Import*.*INTER* *t* (*UNIONS* *s*) = *UNIONS* (*GSPEC* (λ*GEN%PVAR%9*::*?'a*::*type* ⇒ *bool*. ∃ *x*::*?'a*::*type* ⇒ *bool*. *SETSPEC* *GEN%PVAR%9* (*IN* *x* *s*) (*HOL_Light_Import*.*INTER* *t* *x*))))

thm UNIONS_SUBSET:

∀ (*f*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *SUBSET* (*UNIONS* *f*) *t* = (∀ *s*::*?'a*::*type* ⇒ *bool*. *IN* *s* *f* ⟶ *SUBSET* *s* *t*)

thm SUBSET_UNIONS:

∀ (*f*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *g*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *SUBSET* *f* *g* ⟶ *SUBSET* (*UNIONS* *f*) (*UNIONS* *g*)

thm UNIONS_UNION:

∀ (*s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *t*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *UNIONS* (*HOL_Light_Import*.*UNION* *s* *t*) = *HOL_Light_Import*.*UNION* (*UNIONS* *s*) (*UNIONS* *t*)

thm INTERS_UNION:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::(?'a::type \Rightarrow bool) \Rightarrow bool.\ INTERS\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.INTER\ (INTERS\ s)\ (INTERS\ t)$

thm INTERS_0:

$INTERS\ EMPTY = HOL\_Light\_Import.UNIV$

thm INTERS_1:

$INTERS\ (INSERT\ (?s::?'a::type \Rightarrow bool)\ EMPTY) = ?s$

thm INTERS_2:

$INTERS\ (INSERT\ (?s::?'a::type \Rightarrow bool)\ (INSERT\ (?t::?'a::type \Rightarrow bool)\ EMPTY)) = HOL\_Light\_Import.INTER\ ?s\ ?t$

thm INTERS_INSERT:

$INTERS\ (INSERT\ (?s::?'a::type \Rightarrow bool)\ (?u::(?'a::type \Rightarrow bool) \Rightarrow bool)) = HOL\_Light\_Import.INTER\ ?s\ (INTERS\ ?u)$

thm IMAGE_CLAUSES:

$IMAGE\ (?f::?'a::type \Rightarrow ?'b::type)\ EMPTY = EMPTY \wedge IMAGE\ ?f\ (INSERT\ (?x::?'a::type)\ (?s::?'a::type \Rightarrow bool)) = INSERT\ (?f\ ?x)\ (IMAGE\ ?f\ ?s)$

thm IMAGE_UNION:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ IMAGE\ f\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.UNION\ (IMAGE\ f\ s)\ (IMAGE\ f\ t)$

thm IMAGE_ID:

$\forall s::?'a::type \Rightarrow bool.\ IMAGE\ (\lambda x::?'a::type.\ x)\ s = s$

thm IMAGE_I:

$\forall s::?'a::type \Rightarrow bool.\ IMAGE\ id\ s = s$

thm IMAGE_o:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'a::type \Rightarrow ?'c::type)\ s::?'a::type \Rightarrow bool.\ IMAGE\ (f \circ g)\ s = IMAGE\ f\ (IMAGE\ g\ s)$

thm IMAGE_SUBSET:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (IMAGE\ f\ s)\ (IMAGE\ f\ t)$

thm IMAGE_INTER_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow IMAGE\ f\ (HOL\_Light\_Import.INTER\ s\ t) = HOL\_Light\_Import.INTER\ (IMAGE\ f\ s)\ (IMAGE\ f\ t)$

thm IMAGE_DIFF_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow IMAGE\ f\ (DIFF\ s\ t) = DIFF\ (IMAGE\ f\ s)\ (IMAGE\ f\ t)$

thm IMAGE_DELETE_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ a::?'b::type.\ (\forall x::?'b::type.\ f\ x = f\ a \longrightarrow x = a) \longrightarrow IMAGE\ f\ (DELETE\ s\ a) = DELETE\ (IMAGE\ f\ s)\ (f\ a)$

thm IMAGE_EQ_EMPTY:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (IMAGE\ f\ s = EMPTY) = (s = EMPTY)$

thm FORALL_IN_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (\forall y::?'a::type.\ IN\ y\ (IMAGE\ f\ s) \longrightarrow (?P::?'a::type \Rightarrow bool)\ y) = (\forall x::?'b::type.\ IN\ x\ s \longrightarrow ?P\ (f\ x))$

thm EXISTS_IN_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (\exists y::?'a::type.\ IN\ y\ (IMAGE\ f\ s) \land (?P::?'a::type \Rightarrow bool)\ y) = (\exists x::?'b::type.\ IN\ x\ s \land ?P\ (f\ x))$

thm SUBSET_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ SUBSET\ s\ (IMAGE\ f\ t) = (\exists u::?'b::type \Rightarrow bool.\ SUBSET\ u\ t \land s = IMAGE\ f\ u)$

thm FORALL_SUBSET_IMAGE:

$\forall (P::(?'b::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ (\forall t::?'b::type \Rightarrow bool.\ SUBSET\ t\ (IMAGE\ f\ s) \longrightarrow P\ t) = (\forall t::?'a::type \Rightarrow bool.\ SUBSET\ t\ s \longrightarrow P\ (IMAGE\ f\ t))$

thm EXISTS_SUBSET_IMAGE:

$\forall (P::(?'b::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ (\exists t::?'b::type \Rightarrow bool.\ SUBSET\ t\ (IMAGE\ f\ s) \land P\ t) = (\exists t::?'a::type \Rightarrow bool.\ SUBSET\ t\ s \land P\ (IMAGE\ f\ t))$

thm IMAGE_CLAUSES_conjunct1:

$IMAGE\ (?f::?'a::type \Rightarrow ?'b::type)\ (INSERT\ (?x::?'a::type)\ (?s::?'a::type \Rightarrow bool)) = INSERT\ (?f\ ?x)\ (IMAGE\ ?f\ ?s)$

thm IMAGE_CLAUSES_conjunct0:

$IMAGE\ (?f::?'a::type \Rightarrow ?'b::type)\ EMPTY = EMPTY$

thm IMAGE_CONST:

$\forall (s::?'b::type \Rightarrow bool)\ c::?'a::type.\ IMAGE\ (\lambda x::?'b::type.\ c)\ s = (if\ s = EMPTY\ then\ EMPTY\ else\ INSERT\ c\ EMPTY)$

thm SIMPLE_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%11::?'a::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%11\ (IN\ x\ s)\ (f\ x)) = IMAGE\ f\ s$

thm SIMPLE_IMAGE_GEN:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ P::?'b::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%12::?'a::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%12\ (P\ x)\ (f\ x)) = IMAGE\ f\ (GSPEC\ (\lambda GEN\%PVAR\%13::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%13\ (P\ x)\ x))$

thm IMAGE_UNIONS:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::(?'b::type \Rightarrow bool) \Rightarrow bool.\ IMAGE\ f\ (UNIONS\ s) = UNIONS\ (IMAGE\ (IMAGE\ f)\ s)$

thm FUN_IN_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ (IMAGE\ f\ s)$

thm SURJECTIVE_IMAGE_EQ:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists x::?'b::type.\ (?f::?'b::type \Rightarrow ?'a::type)\ x = y)) \wedge (\forall x::?'b::type.\ IN\ (?f\ x)\ t = IN\ x\ s) \longrightarrow IMAGE\ ?f\ s = t$

thm EMPTY_GSPEC:

$GSPEC\ (\lambda GEN\%PVAR\%14::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%14\ False\ x) = EMPTY$

thm UNIV_GSPEC:

$GSPEC\ (\lambda GEN\%PVAR\%15::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%15\ True\ x) = HOL\_Light\_Import.UNIV$

thm SING_GSPEC:

$(\forall a::?'b::type.\ GSPEC\ (\lambda GEN\%PVAR\%16::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%16\ (x = a)\ x) = INSERT\ a\ EMPTY) \wedge (\forall a::?'a::type.\ GSPEC\ (\lambda GEN\%PVAR\%17::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%17\ (a = x)\ x) = INSERT\ a\ EMPTY)$

thm IN_ELIM_PAIR_THM:

$\forall (P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (a::?'b::type)\ b::?'a::type.\ IN\ (a,\ b)\ (GSPEC\ (\lambda GEN\%PVAR\%18::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%18\ (P\ x\ y)\ (x,\ y))) = P\ a\ b$

thm SET_PAIR_THM:

$\forall P::?'b::type \times ?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%19::?'b::type \times ?'a::type.\ \exists p::?'b::type \times ?'a::type.\ SETSPEC\ GEN\%PVAR\%19\ (P\ p)\ p) = GSPEC\ (\lambda GEN\%PVAR\%20::?'b::type \times ?'a::type.\ \exists (a::?'b::type)\ b::?'a::type.\ SETSPEC\ GEN\%PVAR\%20\ (P\ (a,\ b))\ (a,\ b))$

thm FORALL_IN_GSPEC:

$(\forall (P::?'g::type \Rightarrow bool)\ f::?'g::type \Rightarrow ?'f::type.\ (\forall z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%21::?'f::type.\ \exists x::?'g::type.\ SETSPEC\ GEN\%PVAR\%21\ (P\ x)\ (f\ x))) \longrightarrow (?Q::?'f::type \Rightarrow bool)\ z) = (\forall x::?'g::type.\ P\ x \longrightarrow ?Q\ (f\ x))) \wedge (\forall (P::?'e::type \Rightarrow ?'d::type \Rightarrow bool)\ f::?'e::type \Rightarrow ?'d::type \Rightarrow ?'f::type.\ (\forall z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%22::?'f::type.\ \exists (x::?'e::type)\ y::?'d::type.\ SETSPEC\ GEN\%PVAR\%22\ (P\ x\ y)\ (f\ x\ y))) \longrightarrow ?Q\ z) = (\forall (x::?'e::type)\ y::?'d::type.\ P\ x\ y \longrightarrow ?Q\ (f\ x\ y))) \wedge (\forall (P::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'f::type.\ (\forall z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%23::?'f::type.\ \exists (w::?'c::type)\ (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%23\ (P\ w\ x\ y)\ (f\ w\ x\ y))) \longrightarrow ?Q\ z) = (\forall (w::?'c::type)\ (x::?'b::type)\ y::?'a::type.\ P\ w\ x\ y \longrightarrow ?Q\ (f\ w\ x\ y)))$

thm EXISTS_IN_GSPEC:

$(\forall (P::?'g::type \Rightarrow bool)\ f::?'g::type \Rightarrow ?'f::type.\ (\exists z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%24::?'f::type.\ \exists x::?'g::type.\ SETSPEC\ GEN\%PVAR\%24\ (P\ x)\ (f\ x))) \wedge (?Q::?'f::type \Rightarrow bool)\ z) = (\exists x::?'g::type.\ P\ x \wedge ?Q\ (f\ x))) \wedge (\forall (P::?'e::type \Rightarrow ?'d::type \Rightarrow bool)\ f::?'e::type \Rightarrow ?'d::type \Rightarrow ?'f::type.\ (\exists z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%25::?'f::type.\ \exists (x::?'e::type)\ y::?'d::type.\ SETSPEC\ GEN\%PVAR\%25\ (P\ x\ y)\ (f\ x\ y))) \wedge ?Q\ z) = (\exists (x::?'e::type)\ y::?'d::type.\ P\ x\ y \wedge ?Q\ (f\ x\ y))) \wedge (\forall (P::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'f::type.\ (\exists z::?'f::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%26::?'f::type.\ \exists (w::?'c::type)\ (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%26\ (P\ w\ x\ y)\ (f\ w\ x\ y))) \wedge ?Q\ z) = (\exists (w::?'c::type)\ (x::?'b::type)\ y::?'a::type.\ P\ w\ x\ y \wedge ?Q\ (f\ w\ x\ y)))$

thm SET_PROVE_CASES:

$\forall P::(?'a::type \Rightarrow bool) \Rightarrow bool.\ P\ EMPTY \wedge (\forall (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ \neg\ IN\ a\ s \longrightarrow P\ (INSERT\ a\ s)) \longrightarrow (\forall s::?'a::type \Rightarrow bool.\ P\ s)$

thm UNIONS_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ UNIONS\ (IMAGE\ f\ s) = GSPEC\ (\lambda GEN\%PVAR\%27::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%27\ (\exists x::?'b::type.\ IN\ x\ s \wedge IN\ y\ (f\ x))\ y)$

thm INTERS_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ INTERS\ (IMAGE\ f\ s) = GSPEC\ (\lambda GEN\%PVAR\%28::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%28\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ y\ (f\ x))\ y)$

thm UNIONS_GSPEC_conjunct2:

$\forall (P::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow bool)\ f::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%33::?'a::type \Rightarrow bool.\ \exists (x::?'d::type)\ (y::?'c::type)\ z::?'b::type.\ SETSPEC\ GEN\%PVAR\%33$

(P x y z) (f x y z))) = GSPEC (λGEN%PVAR%34::?'a::type. ∃ a::?'a::type. SETSPEC GEN%PVAR%34 (∃ (x::?'d::type) (y::?'c::type) z::?'b::type. P x y z ∧ IN a (f x y z)) a)

thm UNIONS_GSPEC_conjunct1:

∀ (P::?'c::type ⇒ ?'b::type ⇒ bool) f::?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. UNIONS (GSPEC (λGEN%PVAR%31::?'a::type ⇒ bool. ∃ (x::?'c::type) y::?'b::type. SETSPEC GEN%PVAR%31 (P x y) (f x y))) = GSPEC (λGEN%PVAR%32::?'a::type. ∃ a::?'a::type. SETSPEC GEN%PVAR%32 (∃ (x::?'c::type) y::?'b::type. P x y ∧ IN a (f x y)) a)

thm UNIONS_GSPEC_conjunct0:

∀ (P::?'b::type ⇒ bool) f::?'b::type ⇒ ?'a::type ⇒ bool. UNIONS (GSPEC (λGEN%PVAR%29::?'a::type ⇒ bool. ∃ x::?'b::type. SETSPEC GEN%PVAR%29 (P x) (f x))) = GSPEC (λGEN%PVAR%30::?'a::type. ∃ a::?'a::type. SET-SPEC GEN%PVAR%30 (∃ x::?'b::type. P x ∧ IN a (f x)) a)

thm UNIONS_GSPEC:

(∀ (P::?'i::type ⇒ bool) f::?'i::type ⇒ ?'h::type ⇒ bool. UNIONS (GSPEC (λGEN%PVAR%29::?'h::type ⇒ bool. ∃ x::?'i::type. SETSPEC GEN%PVAR%29 (P x) (f x))) = GSPEC (λGEN%PVAR%30::?'h::type. ∃ a::?'h::type. SET-SPEC GEN%PVAR%30 (∃ x::?'i::type. P x ∧ IN a (f x)) a)) ∧ (∀ (P::?'g::type ⇒ ?'f::type ⇒ bool) f::?'g::type ⇒ ?'f::type ⇒ ?'e::type ⇒ bool. UNIONS (GSPEC (λGEN%PVAR%31::?'e::type ⇒ bool. ∃ (x::?'g::type) y::?'f::type. SETSPEC GEN%PVAR%31 (P x y) (f x y))) = GSPEC (λGEN%PVAR%32::?'e::type. ∃ a::?'e::type. SETSPEC GEN%PVAR%32 (∃ (x::?'g::type) y::?'f::type. P x y ∧ IN a (f x y)) a)) ∧ (∀ (P::?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ bool) f::?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. UNIONS (GSPEC (λGEN%PVAR%33::?'a::type ⇒ bool. ∃ (x::?'d::type) (y::?'c::type) z::?'b::type. SETSPEC GEN%PVAR%33 (P x y z) (f x y z))) = GSPEC (λGEN%PVAR%34::?'a::type. ∃ a::?'a::type. SETSPEC GEN%PVAR%34 (∃ (x::?'d::type) (y::?'c::type) z::?'b::type. P x y z ∧ IN a (f x y z)) a))

thm INTERS_GSPEC_conjunct2:

∀ (P::?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ bool) f::?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. INTERS (GSPEC (λGEN%PVAR%39::?'a::type ⇒ bool. ∃ (x::?'d::type) (y::?'c::type) z::?'b::type. SETSPEC GEN%PVAR%39 (P x y z) (f x y z))) = GSPEC (λGEN%PVAR%40::?'a::type. ∃ a::?'a::type. SETSPEC GEN%PVAR%40 (∀ (x::?'d::type) (y::?'c::type) z::?'b::type. P x y z ⟶ IN a (f x y z)) a)

thm INTERS_GSPEC_conjunct1:

∀ (P::?'c::type ⇒ ?'b::type ⇒ bool) f::?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. INTERS (GSPEC (λGEN%PVAR%37::?'a::type ⇒ bool. ∃ (x::?'c::type) y::?'b::type. SETSPEC GEN%PVAR%37 (P x y) (f x y))) = GSPEC (λGEN%PVAR%38::?'a::type.

$\exists\, a{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%38\ (\forall\, (x{::}?'c{::}type)\ y{::}?'b{::}type.\ P\ x\ y \longrightarrow IN\ a\ (f\ x\ y))\ a)$

thm INTERS_GSPEC_conjunct0:

$\forall\, (P{::}?'b{::}type \Rightarrow bool)\ f{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%35{::}?'a{::}type \Rightarrow bool.\ \exists\, x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%35\ (P\ x)\ (f\ x))) = GSPEC\ (\lambda GEN\%PVAR\%36{::}?'a{::}type.\ \exists\, a{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%36\ (\forall\, x{::}?'b{::}type.\ P\ x \longrightarrow IN\ a\ (f\ x))\ a)$

thm INTERS_GSPEC:

$(\forall\, (P{::}?'i{::}type \Rightarrow bool)\ f{::}?'i{::}type \Rightarrow ?'h{::}type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%35{::}?'h{::}type \Rightarrow bool.\ \exists\, x{::}?'i{::}type.\ SETSPEC\ GEN\%PVAR\%35\ (P\ x)\ (f\ x))) = GSPEC\ (\lambda GEN\%PVAR\%36{::}?'h{::}type.\ \exists\, a{::}?'h{::}type.\ SETSPEC\ GEN\%PVAR\%36\ (\forall\, x{::}?'i{::}type.\ P\ x \longrightarrow IN\ a\ (f\ x))\ a)) \wedge (\forall\, (P{::}?'g{::}type \Rightarrow ?'f{::}type \Rightarrow bool)\ f{::}?'g{::}type \Rightarrow ?'f{::}type \Rightarrow ?'e{::}type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%37{::}?'e{::}type \Rightarrow bool.\ \exists\, (x{::}?'g{::}type)\ y{::}?'f{::}type.\ SETSPEC\ GEN\%PVAR\%37\ (P\ x\ y)\ (f\ x\ y))) = GSPEC\ (\lambda GEN\%PVAR\%38{::}?'e{::}type.\ \exists\, a{::}?'e{::}type.\ SETSPEC\ GEN\%PVAR\%38\ (\forall\, (x{::}?'g{::}type)\ y{::}?'f{::}type.\ P\ x\ y \longrightarrow IN\ a\ (f\ x\ y))\ a)) \wedge (\forall\, (P{::}?'d{::}type \Rightarrow ?'c{::}type \Rightarrow ?'b{::}type \Rightarrow bool)\ f{::}?'d{::}type \Rightarrow ?'c{::}type \Rightarrow ?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%39{::}?'a{::}type \Rightarrow bool.\ \exists\, (x{::}?'d{::}type)\ (y{::}?'c{::}type)\ z{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%39\ (P\ x\ y\ z)\ (f\ x\ y\ z))) = GSPEC\ (\lambda GEN\%PVAR\%40{::}?'a{::}type.\ \exists\, a{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%40\ (\forall\, (x{::}?'d{::}type)\ (y{::}?'c{::}type)\ z{::}?'b{::}type.\ P\ x\ y\ z \longrightarrow IN\ a\ (f\ x\ y\ z))\ a))$

thm DIFF_INTERS:

$\forall\, (u{::}?'a{::}type \Rightarrow bool)\ s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ DIFF\ u\ (INTERS\ s) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%41{::}?'a{::}type \Rightarrow bool.\ \exists\, t{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%41\ (IN\ t\ s)\ (DIFF\ u\ t)))$

thm INTERS_UNIONS:

$\forall\, s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ INTERS\ s = DIFF\ HOL\_Light\_Import.UNIV\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%42{::}?'a{::}type \Rightarrow bool.\ \exists\, t{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%42\ (IN\ t\ s)\ (DIFF\ HOL\_Light\_Import.UNIV\ t))))$

thm UNIONS_INTERS:

$\forall\, s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ UNIONS\ s = DIFF\ HOL\_Light\_Import.UNIV\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%43{::}?'a{::}type \Rightarrow bool.\ \exists\, t{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%43\ (IN\ t\ s)\ (DIFF\ HOL\_Light\_Import.UNIV\ t))))$

thm DIFF_UNIONS:

$\forall\, (s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ DIFF\ (UNIONS\ s)\ t = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%44{::}?'a{::}type \Rightarrow bool.\ \exists\, x{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%44\ (IN\ x\ s)\ (DIFF\ x\ t)))$

thm INTERS_OVER_UNIONS:

∀ (f::?'b::type ⇒ (?'a::type ⇒ bool) ⇒ bool) s::?'b::type ⇒ bool. INTERS (GSPEC (λGEN%PVAR%45::?'a::type ⇒ bool. ∃ x::?'b::type. SETSPEC GEN%PVAR%45 (IN x s) (UNIONS (f x)))) = UNIONS (GSPEC (λGEN%PVAR%49::?'a::type ⇒ bool. ∃ g::?'b::type ⇒ ?'a::type ⇒ bool. SETSPEC GEN%PVAR%49 (∀ x::?'b::type. IN x s ⟶ IN (g x) (f x)) (INTERS (GSPEC (λGEN%PVAR%48::?'a::type ⇒ bool. ∃ x::?'b::type. SETSPEC GEN%PVAR%48 (IN x s) (g x))))))

thm FINITE_RULES_conjunct1:

∀ (x::?'a::type) s::?'a::type ⇒ bool. FINITE s ⟶ FINITE (INSERT x s)

thm FINITE_RULES_conjunct0:

FINITE EMPTY

thm FINITE_INDUCT_STRONG:

∀ P::(?'a::type ⇒ bool) ⇒ bool. P EMPTY ∧ (∀ (x::?'a::type) s::?'a::type ⇒ bool. P s ∧ ¬ IN x s ∧ FINITE s ⟶ P (INSERT x s)) ⟶ (∀ s::?'a::type ⇒ bool. FINITE s ⟶ P s)

thm SURJECTIVE_ON_RIGHT_INVERSE:

∀ (f::?'b::type ⇒ ?'a::type) t::?'a::type ⇒ bool. (∀ y::?'a::type. IN y t ⟶ (∃ x::?'b::type. IN x (?s::?'b::type ⇒ bool) ∧ f x = y)) = (∃ g::?'a::type ⇒ ?'b::type. ∀ y::?'a::type. IN y t ⟶ IN (g y) ?s ∧ f (g y) = y)

thm INJECTIVE_ON_LEFT_INVERSE:

∀ (f::?'b::type ⇒ ?'a::type) s::?'b::type ⇒ bool. (∀ (x::?'b::type) y::?'b::type. IN x s ∧ IN y s ∧ f x = f y ⟶ x = y) = (∃ g::?'a::type ⇒ ?'b::type. ∀ x::?'b::type. IN x s ⟶ g (f x) = x)

thm BIJECTIVE_ON_LEFT_RIGHT_INVERSE:

∀ (f::?'b::type ⇒ ?'a::type) (s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. (∀ x::?'b::type. IN x s ⟶ IN (f x) t) ⟶ ((∀ (x::?'b::type) y::?'b::type. IN x s ∧ IN y s ∧ f x = f y ⟶ x = y) ∧ (∀ y::?'a::type. IN y t ⟶ (∃ x::?'b::type. IN x s ∧ f x = y))) = (∃ g::?'a::type ⇒ ?'b::type. (∀ y::?'a::type. IN y t ⟶ IN (g y) s) ∧ (∀ y::?'a::type. IN y t ⟶ f (g y) = y) ∧ (∀ x::?'b::type. IN x s ⟶ g (f x) = x))

thm SURJECTIVE_RIGHT_INVERSE:

(∀ y::?'b::type. ∃ x::?'a::type. (?f::?'a::type ⇒ ?'b::type) x = y) = (∃ g::?'b::type ⇒ ?'a::type. ∀ y::?'b::type. ?f (g y) = y)

thm INJECTIVE_LEFT_INVERSE:

(∀ (x::?'b::type) y::?'b::type. (?f::?'b::type ⇒ ?'a::type) x = ?f y ⟶ x = y) = (∃ g::?'a::type ⇒ ?'b::type. ∀ x::?'b::type. g (?f x) = x)

thm BIJECTIVE_LEFT_RIGHT_INVERSE:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ ((\forall (x{::}?'b{::}type)\ y{::}?'b{::}type.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\ x = y)) = (\exists g{::}?'a{::}type \Rightarrow ?'b{::}type.\ (\forall y{::}?'a{::}type.\ f\ (g\ y) = y) \wedge (\forall x{::}?'b{::}type.\ g\ (f\ x) = x))$

thm FUNCTION_FACTORS_LEFT_GEN:

$\forall (P{::}?'c{::}type \Rightarrow bool)\ (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ g{::}?'c{::}type \Rightarrow ?'a{::}type.\ (\forall (x{::}?'c{::}type)\ y{::}?'c{::}type.\ P\ x \wedge P\ y \wedge g\ x = g\ y \longrightarrow f\ x = f\ y) = (\exists h{::}?'a{::}type \Rightarrow ?'b{::}type.\ \forall x{::}?'c{::}type.\ P\ x \longrightarrow f\ x = h\ (g\ x))$

thm FUNCTION_FACTORS_LEFT:

$\forall (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ g{::}?'c{::}type \Rightarrow ?'a{::}type.\ (\forall (x{::}?'c{::}type)\ y{::}?'c{::}type.\ g\ x = g\ y \longrightarrow f\ x = f\ y) = (\exists h{::}?'a{::}type \Rightarrow ?'b{::}type.\ f = h \circ g)$

thm FUNCTION_FACTORS_RIGHT_GEN:

$\forall (P{::}?'c{::}type \Rightarrow bool)\ (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ g{::}?'a{::}type \Rightarrow ?'b{::}type.\ (\forall x{::}?'c{::}type.\ P\ x \longrightarrow (\exists y{::}?'a{::}type.\ g\ y = f\ x)) = (\exists h{::}?'c{::}type \Rightarrow ?'a{::}type.\ \forall x{::}?'c{::}type.\ P\ x \longrightarrow f\ x = g\ (h\ x))$

thm FUNCTION_FACTORS_RIGHT:

$\forall (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ g{::}?'a{::}type \Rightarrow ?'b{::}type.\ (\forall x{::}?'c{::}type.\ \exists y{::}?'a{::}type.\ g\ y = f\ x) = (\exists h{::}?'c{::}type \Rightarrow ?'a{::}type.\ f = g \circ h)$

thm SURJECTIVE_FORALL_THM:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\ x = y) = (\forall P{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'b{::}type.\ P\ (f\ x)) = (\forall y{::}?'a{::}type.\ P\ y))$

thm SURJECTIVE_EXISTS_THM:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\ x = y) = (\forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'b{::}type.\ P\ (f\ x)) = (\exists y{::}?'a{::}type.\ P\ y))$

thm SURJECTIVE_IMAGE_THM:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\ x = y) = (\forall P{::}?'a{::}type \Rightarrow bool.\ IMAGE\ f\ (GSPEC\ (\lambda GEN\%PVAR\%50{::}?'b{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%50\ (P\ (f\ x))\ x)) = GSPEC\ (\lambda GEN\%PVAR\%51{::}?'a{::}type.\ \exists x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%51\ (P\ x)\ x))$

thm IMAGE_INJECTIVE_IMAGE_OF_SUBSET:

$\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ s{::}?'b{::}type \Rightarrow bool.\ \exists t{::}?'b{::}type \Rightarrow bool.\ SUBSET\ t\ s \wedge IMAGE\ f\ s = IMAGE\ f\ t \wedge (\forall (x{::}?'b{::}type)\ y{::}?'b{::}type.\ IN\ x\ t \wedge IN\ y\ t \wedge f\ x = f\ y \longrightarrow x = y)$

thm FINITE_EMPTY:

$FINITE\ EMPTY$

thm FINITE_SUBSET:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ s\ t \longrightarrow FINITE\ s$

thm UNION_EMPTY_conjunct1:

$\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ s\ EMPTY = s$

thm UNION_EMPTY_conjunct0:

$\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ EMPTY\ s = s$

thm FINITE_UNION_IMP:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (HOL\_Light\_Import.UNION\ s\ t)$

thm FINITE_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.UNION\ s\ t) = (FINITE\ s \wedge FINITE\ t)$

thm FINITE_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \vee FINITE\ t \longrightarrow FINITE\ (HOL\_Light\_Import.INTER\ s\ t)$

thm FINITE_INSERT:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ FINITE\ (INSERT\ x\ s) = FINITE\ s$

thm FINITE_SING:

$\forall a::?'a::type.\ FINITE\ (INSERT\ a\ EMPTY)$

thm FINITE_DELETE_IMP:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ FINITE\ s \longrightarrow FINITE\ (DELETE\ s\ x)$

thm FINITE_DELETE:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ FINITE\ (DELETE\ s\ x) = FINITE\ s$

thm FINITE_FINITE_UNIONS:

$\forall s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (UNIONS\ s) = (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow FINITE\ t)$

thm FINITE_IMAGE_EXPAND:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%54::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%54\ (\exists x::?'b::type.\ IN\ x\ s \wedge y = f\ x)\ y))$

thm FINITE_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (IMAGE\ f\ s)$

thm FINITE_IMAGE_INJ_GENERAL:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (A::?'a::type \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge FINITE\ A \longrightarrow FINITE$

$(GSPEC\ (\lambda GEN\%PVAR\%55::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%55\ (IN\ x\ s \land IN\ (f\ x)\ A)\ x))$

thm FINITE_FINITE_PREIMAGE_GENERAL:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \land (\forall y::?'a::type.\ IN\ y\ t \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%58::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%58\ (IN\ x\ s \land f\ x = y)\ x))) \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%59::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%59\ (IN\ x\ s \land IN\ (f\ x)\ t)\ x))$

thm FINITE_FINITE_PREIMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \land (\forall y::?'a::type.\ IN\ y\ t \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%60::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%60\ (f\ x = y)\ x))) \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%61::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%61\ (IN\ (f\ x)\ t)\ x))$

thm FINITE_IMAGE_INJ_EQ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \land IN\ y\ s \land f\ x = f\ y \longrightarrow x = y) \longrightarrow FINITE\ (IMAGE\ f\ s) = FINITE\ s$

thm FINITE_IMAGE_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ A::?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y) \land FINITE\ A \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%62::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%62\ (IN\ (f\ x)\ A)\ x))$

thm INFINITE_IMAGE_INJ:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\forall s::?'b::type \Rightarrow bool.\ INFINITE\ s \longrightarrow INFINITE\ (IMAGE\ f\ s))$

thm INFINITE_NONEMPTY:

$\forall s::?'a::type \Rightarrow bool.\ INFINITE\ s \longrightarrow s \neq EMPTY$

thm INFINITE_DIFF_FINITE:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ s \land FINITE\ t \longrightarrow INFINITE\ (DIFF\ s\ t)$

thm FINITE_SUBSET_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (FINITE\ t \land SUBSET\ t\ (IMAGE\ f\ s)) = (\exists s'::?'b::type \Rightarrow bool.\ FINITE\ s' \land SUBSET\ s'\ s \land t = IMAGE\ f\ s')$

thm EXISTS_FINITE_SUBSET_IMAGE:

$\forall (P::(?'b::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ (\exists t::?'b::type \Rightarrow bool.\ FINITE\ t \land SUBSET\ t\ (IMAGE\ f\ s) \land P\ t) = (\exists t::?'a::type \Rightarrow bool.\ FINITE\ t \land SUBSET\ t\ s \land P\ (IMAGE\ f\ t))$

thm FORALL_FINITE_SUBSET_IMAGE:

$\forall (P::(?'b::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.$ $(\forall t::?'b::type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ t\ (IMAGE\ f\ s) \longrightarrow P\ t) =$ $(\forall t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ t\ s \longrightarrow P\ (IMAGE\ f\ t))$

thm FINITE_SUBSET_IMAGE_IMP:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t$ $\wedge\ SUBSET\ t\ (IMAGE\ f\ s) \longrightarrow (\exists s'::?'b::type \Rightarrow bool.\ FINITE\ s' \wedge SUBSET$ $s'\ s \wedge SUBSET\ t\ (IMAGE\ f\ s'))$

thm FINITE_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (DIFF\ s\ t)$

thm INFINITE_SUPERSET:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ s \wedge SUBSET\ s\ t \longrightarrow$ $INFINITE\ t$

thm DEF_FINREC:

$FINREC = (SOME\ FINREC::nat \Rightarrow (?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type) \Rightarrow$ $?'a::type \Rightarrow (?'b::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow nat \Rightarrow bool.\ \forall\ \_21953::nat.$ $(\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ (a::?'a::type)$ $b::?'a::type.\ FINREC\ \_21953\ f\ b\ s\ a\ (0::nat) = (s = EMPTY \wedge a = b))$ $\wedge\ (\forall (b::?'a::type)\ (s::?'b::type \Rightarrow bool)\ (n::nat)\ (a::?'a::type)\ f::?'b::type \Rightarrow$ $?'a::type \Rightarrow ?'a::type.\ FINREC\ \_21953\ f\ b\ s\ a\ (Suc\ n) = (\exists (x::?'b::type)$ $c::?'a::type.\ IN\ x\ s \wedge FINREC\ \_21953\ f\ b\ (DELETE\ s\ x)\ c\ n \wedge a = f\ x\ c)))$ $(44::nat)$

thm FINREC_conjunct0:

$FINREC\ (?f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (?b::?'a::type)\ (?s::?'b::type$ $\Rightarrow bool)\ (?a::?'a::type)\ (0::nat) = (?s = EMPTY \wedge ?a = ?b)$

thm FINREC_conjunct1:

$FINREC\ (?f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (?b::?'a::type)\ (?s::?'b::type$ $\Rightarrow bool)\ (?a::?'a::type)\ (Suc\ (?n::nat)) = (\exists (x::?'b::type)\ c::?'a::type.\ IN\ x\ ?s$ $\wedge\ FINREC\ ?f\ ?b\ (DELETE\ ?s\ x)\ c\ ?n \wedge ?a = ?f\ x\ c)$

thm FINREC:

$FINREC\ (?f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (?b::?'a::type)\ (?s::?'b::type$ $\Rightarrow bool)\ (?a::?'a::type)\ (0::nat) = (?s = EMPTY \wedge ?a = ?b) \wedge FINREC\ ?f$ $?b\ ?s\ ?a\ (Suc\ (?n::nat)) = (\exists (x::?'b::type)\ c::?'a::type.\ IN\ x\ ?s \wedge FINREC\ ?f$ $?b\ (DELETE\ ?s\ x)\ c\ ?n \wedge ?a = ?f\ x\ c)$

thm FINREC_1_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (b::?'a::type)\ (s::?'b::type \Rightarrow bool)$ $a::?'a::type.\ FINREC\ f\ b\ s\ a\ (Suc\ (0::nat)) = (\exists x::?'b::type.\ s = INSERT$ $x\ EMPTY \wedge a = f\ x\ b)$

thm FINREC_SUC_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow (\forall (n::nat)\ (s::?'b::type \Rightarrow bool)\ z::?'a::type.\ FINREC\ f\ b\ s\ z\ (Suc\ n) \longrightarrow (\forall x::?'b::type.\ IN\ x\ s \longrightarrow (\exists w::?'a::type.\ FINREC\ f\ b\ (DELETE\ s\ x)\ w\ n \land z = f\ x\ w)))$

thm FINREC_UNIQUE_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow (\forall (n1::nat)\ (n2::nat)\ (s::?'b::type \Rightarrow bool)\ (a1::?'a::type)\ a2::?'a::type.\ FINREC\ f\ b\ s\ a1\ n1 \land FINREC\ f\ b\ s\ a2\ n2 \longrightarrow a1 = a2 \land n1 = n2)$

thm FINREC_EXISTS_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (b::?'a::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists (a::?'a::type)\ n::nat.\ FINREC\ f\ b\ s\ a\ n)$

thm FINREC_FUN_LEMMA:

$\forall (P::?'c::type \Rightarrow bool)\ R::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall s::?'c::type.\ P\ s \longrightarrow (\exists (a::?'b::type)\ n::?'a::type.\ R\ s\ a\ n)) \land (\forall (n1::?'a::type)\ (n2::?'a::type)\ (s::?'c::type)\ (a1::?'b::type)\ a2::?'b::type.\ R\ s\ a1\ n1 \land R\ s\ a2\ n2 \longrightarrow a1 = a2 \land n1 = n2) \longrightarrow (\exists f::?'c::type \Rightarrow ?'b::type.\ \forall (s::?'c::type)\ a::?'b::type.\ P\ s \longrightarrow (\exists n::?'a::type.\ R\ s\ a\ n) = (f\ s = a))$

thm FINREC_FUN:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow (\exists g::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ g\ EMPTY = b \land (\forall (s::?'b::type \Rightarrow bool)\ x::?'b::type.\ FINITE\ s \land IN\ x\ s \longrightarrow g\ s = f\ x\ (g\ (DELETE\ s\ x))))$

thm SET_RECURSION_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow (\exists g::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ g\ EMPTY = b \land (\forall (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow g\ (INSERT\ x\ s) = (if\ IN\ x\ s\ then\ g\ s\ else\ f\ x\ (g\ s))))$

thm DEF_ITSET:

$ITSET = (\lambda(\_22803::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (\_22804::?'b::type \Rightarrow bool)\ \_22805::?'a::type.\ (SOME\ g::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ g\ EMPTY = \_22805 \land (\forall (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow g\ (INSERT\ x\ s) = (if\ IN\ x\ s\ then\ g\ s\ else\ \_22803\ x\ (g\ s))))\ \_22804)$

thm ITSET:

$\forall (b::?'b::type)\ (f::?'a::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ ITSET\ f\ s\ b = (SOME\ g::(?'a::type \Rightarrow bool) \Rightarrow ?'b::type.\ g\ EMPTY = b \land (\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow g\ (INSERT\ x\ s) = (if\ IN\ x\ s\ then\ g\ s\ else\ f\ x\ (g\ s))))\ s$

thm FINITE_RECURSION:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow ITSET\ f\ EMPTY\ b = b \wedge (\forall (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow ITSET\ f\ (INSERT\ x\ s)\ b = (if\ IN\ x\ s\ then\ ITSET\ f\ s\ b\ else\ f\ x\ (ITSET\ f\ s\ b)))$

thm FINITE_RECURSION_DELETE:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \longrightarrow ITSET\ f\ EMPTY\ b = b \wedge (\forall (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow ITSET\ f\ s\ b = (if\ IN\ x\ s\ then\ f\ x\ (ITSET\ f\ (DELETE\ s\ x)\ b)\ else\ ITSET\ f\ (DELETE\ s\ x)\ b))$

thm ITSET_EQ:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ b::?'a::type.\ FINITE\ s \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow f\ x\ (f\ y\ s) = f\ y\ (f\ x\ s)) \wedge (\forall (x::?'b::type)\ (y::?'b::type)\ s::?'a::type.\ x \neq y \longrightarrow g\ x\ (g\ y\ s) = g\ y\ (g\ x\ s)) \longrightarrow ITSET\ f\ s\ b = ITSET\ g\ s\ b$

thm SUBSET_RESTRICT:

$\forall (s::?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow bool.\ SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%64::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%64\ (IN\ x\ s \wedge P\ x)\ x))\ s$

thm FINITE_RESTRICT:

$\forall (s::?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%65::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%65\ (IN\ x\ s \wedge P\ x)\ x))$

thm DEF_CARD:

$CARD = (\lambda\_23030::?'a::type \Rightarrow bool.\ ITSET\ (\lambda x::?'a::type.\ Suc)\ \_23030\ (0::nat))$

thm CARD:

$\forall s::?'a::type \Rightarrow bool.\ CARD\ s = ITSET\ (\lambda x::?'a::type.\ Suc)\ s\ (0::nat)$

thm CARD_CLAUSES:

$CARD\ EMPTY = (0::nat) \wedge (\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (INSERT\ x\ s) = (if\ IN\ x\ s\ then\ CARD\ s\ else\ Suc\ (CARD\ s)))$

thm CARD_CLAUSES_conjunct1:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (INSERT\ x\ s) = (if\ IN\ x\ s\ then\ CARD\ s\ else\ Suc\ (CARD\ s))$

thm INTER_EMPTY_conjunct1:

$\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ EMPTY = EMPTY$

thm INTER_EMPTY_conjunct0:

$\forall s{::}?'a{::}type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ EMPTY\ s = EMPTY$

thm CARD_CLAUSES_conjunct0:

$CARD\ EMPTY = (0{::}nat)$

thm CARD_UNION:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \land FINITE\ t \land HOL\_Light\_Import.INTER\ s\ t = EMPTY \longrightarrow CARD\ (HOL\_Light\_Import.UNION\ s\ t) = CARD\ s + CARD\ t$

thm CARD_DELETE:

$\forall (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (DELETE\ s\ x) = (if\ IN\ x\ s\ then\ CARD\ s - (1{::}nat)\ else\ CARD\ s)$

thm CARD_UNION_EQ:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ (t{::}?'a{::}type \Rightarrow bool)\ u{::}?'a{::}type \Rightarrow bool.\ FINITE\ u \land HOL\_Light\_Import.INTER\ s\ t = EMPTY \land HOL\_Light\_Import.UNION\ s\ t = u \longrightarrow CARD\ s + CARD\ t = CARD\ u$

thm CARD_DIFF:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \land SUBSET\ t\ s \longrightarrow CARD\ (DIFF\ s\ t) = CARD\ s - CARD\ t$

thm CARD_EQ_0:

$\forall s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow (CARD\ s = (0{::}nat)) = (s = EMPTY)$

thm FINITE_INDUCT_DELETE:

$\forall P{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ P\ EMPTY \land (\forall s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \land s \neq EMPTY \longrightarrow (\exists x{::}?'a{::}type.\ IN\ x\ s \land (P\ (DELETE\ s\ x) \longrightarrow P\ s))) \longrightarrow (\forall s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow P\ s)$

thm DEF_HAS_SIZE:

$HAS\_SIZE = (\lambda(\_23245{::}?'a{::}type \Rightarrow bool)\ \_23246{::}nat.\ FINITE\ \_23245 \land CARD\ \_23245 = \_23246)$

thm HAS_SIZE:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ n{::}nat.\ HAS\_SIZE\ s\ n = (FINITE\ s \land CARD\ s = n)$

thm HAS_SIZE_CARD:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ n{::}nat.\ HAS\_SIZE\ s\ n \longrightarrow CARD\ s = n$

thm HAS_SIZE_0:

$\forall s{::}?'a{::}type \Rightarrow bool.\ HAS\_SIZE\ s\ (0{::}nat) = (s = EMPTY)$

thm HAS_SIZE_SUC:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ n{::}nat.\ HAS\_SIZE\ s\ (Suc\ n) = (s \neq EMPTY \land (\forall a{::}?'a{::}type.\ IN\ a\ s \longrightarrow HAS\_SIZE\ (DELETE\ s\ a)\ n))$

thm HAS_SIZE_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (m::nat)\ n::nat.\ HAS\_SIZE\ s\ m \wedge HAS\_SIZE\ t\ n \wedge DISJOINT\ s\ t \longrightarrow HAS\_SIZE\ (HOL\_Light\_Import.UNION\ s\ t)\ (m + n)$

thm HAS_SIZE_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (m::nat)\ n::nat.\ HAS\_SIZE\ s\ m \wedge HAS\_SIZE\ t\ n \wedge SUBSET\ t\ s \longrightarrow HAS\_SIZE\ (DIFF\ s\ t)\ (m - n)$

thm HAS_SIZE_UNIONS:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (m::nat)\ n::nat.\ HAS\_SIZE\ s\ m \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow HAS\_SIZE\ (t\ x)\ n) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (t\ x)\ (t\ y)) \longrightarrow HAS\_SIZE\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%68::?'a::type \Rightarrow bool.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%68\ (IN\ x\ s)\ (t\ x))))\ (m * n)$

thm FINITE_HAS_SIZE:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s = HAS\_SIZE\ s\ (CARD\ s)$

thm HAS_SIZE_CLAUSES_conjunct0:

$HAS\_SIZE\ (?s::?'a::type \Rightarrow bool)\ (0::nat) = (?s = EMPTY)$

thm HAS_SIZE_CLAUSES_conjunct1:

$HAS\_SIZE\ (?s::?'a::type \Rightarrow bool)\ (Suc\ (?n::nat)) = (\exists (a::?'a::type)\ t::?'a::type \Rightarrow bool.\ HAS\_SIZE\ t\ ?n \wedge \neg\ IN\ a\ t \wedge ?s = INSERT\ a\ t)$

thm HAS_SIZE_CLAUSES:

$HAS\_SIZE\ (?s::?'a::type \Rightarrow bool)\ (0::nat) = (?s = EMPTY) \wedge HAS\_SIZE\ ?s\ (Suc\ (?n::nat)) = (\exists (a::?'a::type)\ t::?'a::type \Rightarrow bool.\ HAS\_SIZE\ t\ ?n \wedge \neg\ IN\ a\ t \wedge ?s = INSERT\ a\ t)$

thm CARD_SUBSET_EQ:

$\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ FINITE\ b \wedge SUBSET\ a\ b \wedge CARD\ a = CARD\ b \longrightarrow a = b$

thm CARD_SUBSET:

$\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ SUBSET\ a\ b \wedge FINITE\ b \longrightarrow CARD\ a \leq CARD\ b$

thm CARD_SUBSET_LE:

$\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ FINITE\ b \wedge SUBSET\ a\ b \wedge CARD\ b \leq CARD\ a \longrightarrow a = b$

thm SUBSET_CARD_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ s\ t \longrightarrow (CARD\ s = CARD\ t) = (s = t)$

thm CARD_PSUBSET:

$\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ PSUBSET\ a\ b \wedge FINITE\ b \longrightarrow CARD\ a < CARD\ b$

thm CARD_UNION_LE:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow CARD\ (HOL\_Light\_Import.UNION\ s\ t) \leq CARD\ s + CARD\ t$

thm CARD_UNIONS_LE:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (m::nat)\ n::nat.\ HAS\_SIZE\ s\ m \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow FINITE\ (t\ x) \wedge CARD\ (t\ x) \leq n) \longrightarrow CARD\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%74::?'a::type \Rightarrow bool.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%74\ (IN\ x\ s)\ (t\ x)))) \leq m * n$

thm INTER_SUBSET_conjunct1:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.INTER\ t\ s)\ s$

thm INTER_SUBSET_conjunct0:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.INTER\ s\ t)\ s$

thm CARD_UNION_GEN:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow CARD\ (HOL\_Light\_Import.UNION\ s\ t) = CARD\ s + CARD\ t - CARD\ (HOL\_Light\_Import.INTER\ s\ t)$

thm CARD_UNION_OVERLAP_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow (CARD\ (HOL\_Light\_Import.UNION\ s\ t) = CARD\ s + CARD\ t) = (HOL\_Light\_Import.INTER\ s\ t = EMPTY)$

thm CARD_UNION_OVERLAP:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge CARD\ (HOL\_Light\_Import.UNION\ s\ t) < CARD\ s + CARD\ t \longrightarrow HOL\_Light\_Import.INTER\ s\ t \neq EMPTY$

thm CARD_IMAGE_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge FINITE\ s \longrightarrow CARD\ (IMAGE\ f\ s) = CARD\ s$

thm HAS_SIZE_IMAGE_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ n::nat.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge HAS\_SIZE\ s\ n \longrightarrow HAS\_SIZE\ (IMAGE\ f\ s)\ n$

thm CARD_IMAGE_LE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (IMAGE\ f\ s) \leq CARD\ s$

thm CARD_IMAGE_INJ_EQ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \land (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \land (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists !x::?'b::type.\ IN\ x\ s \land f\ x = y)) \longrightarrow CARD\ t = CARD\ s$

thm CARD_SUBSET_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ FINITE\ t \land SUBSET\ s\ (IMAGE\ f\ t) \longrightarrow CARD\ s \leq CARD\ t$

thm HAS_SIZE_IMAGE_INJ_EQ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ n::nat.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \land IN\ y\ s \land f\ x = f\ y \longrightarrow x = y) \longrightarrow HAS\_SIZE\ (IMAGE\ f\ s)\ n = HAS\_SIZE\ s\ n$

thm CARD_IMAGE_EQ_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow (CARD\ (IMAGE\ f\ s) = CARD\ s) = (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \land IN\ y\ s \land f\ x = f\ y \longrightarrow x = y)$

thm CHOOSE_SUBSET_STRONG:

$\forall (n::nat)\ s::?'a::type \Rightarrow bool.\ (FINITE\ s \longrightarrow n \leq CARD\ s) \longrightarrow (\exists t::?'a::type \Rightarrow bool.\ SUBSET\ t\ s \land HAS\_SIZE\ t\ n)$

thm CHOOSE_SUBSET:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow (\forall n \leq CARD\ s.\ \exists t::?'a::type \Rightarrow bool.\ SUBSET\ t\ s \land HAS\_SIZE\ t\ n)$

thm CHOOSE_SUBSET_BETWEEN:

$\forall (n::nat)\ (s::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SUBSET\ s\ u \land FINITE\ s \land CARD\ s \leq n \land (FINITE\ u \longrightarrow n \leq CARD\ u) \longrightarrow (\exists t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \land SUBSET\ t\ u \land HAS\_SIZE\ t\ n)$

thm HAS_SIZE_PRODUCT_DEPENDENT:

$\forall (s::?'b::type \Rightarrow bool)\ (m::nat)\ (t::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ n::nat.\ HAS\_SIZE\ s\ m \land (\forall x::?'b::type.\ IN\ x\ s \longrightarrow HAS\_SIZE\ (t\ x)\ n) \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%77::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%77\ (IN\ x\ s \land IN\ y\ (t\ x))\ (x,\ y)))\ (m * n)$

thm FORALL_IN_GSPEC_conjunct2:

$\forall (P::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow bool)\ f::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\forall z::?'a::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%23::?'a::type.\ \exists (w::?'d::type)\ (x::?'c::type)\ y::?'b::type.\ SETSPEC\ GEN\%PVAR\%23\ (P\ w$

$x\ y)\ (f\ w\ x\ y))) \longrightarrow (?Q::?'a::type \Rightarrow bool)\ z) = (\forall (w::?'d::type)\ (x::?'c::type)\ y::?'b::type.\ P\ w\ x\ y \longrightarrow ?Q\ (f\ w\ x\ y))$

thm FORALL_IN_GSPEC_conjunct1:

$\forall (P::?'c::type \Rightarrow ?'b::type \Rightarrow bool)\ f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\forall z::?'a::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%22::?'a::type.\ \exists (x::?'c::type)\ y::?'b::type.\ SETSPEC\ GEN\%PVAR\%22\ (P\ x\ y)\ (f\ x\ y))) \longrightarrow (?Q::?'a::type \Rightarrow bool)\ z) = (\forall (x::?'c::type)\ y::?'b::type.\ P\ x\ y \longrightarrow ?Q\ (f\ x\ y))$

thm FORALL_IN_GSPEC_conjunct0:

$\forall (P::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ (\forall z::?'a::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%21::?'a::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%21\ (P\ x)\ (f\ x))) \longrightarrow (?Q::?'a::type \Rightarrow bool)\ z) = (\forall x::?'b::type.\ P\ x \longrightarrow ?Q\ (f\ x))$

thm FINITE_PRODUCT_DEPENDENT:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ t::?'c::type \Rightarrow ?'b::type \Rightarrow bool.\ FINITE\ s \land (\forall x::?'c::type.\ IN\ x\ s \longrightarrow FINITE\ (t\ x)) \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%82::?'a::type.\ \exists (x::?'c::type)\ y::?'b::type.\ SETSPEC\ GEN\%PVAR\%82\ (IN\ x\ s \land IN\ y\ (t\ x))\ (f\ x\ y)))$

thm FINITE_PRODUCT:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \land FINITE\ t \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%83::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%83\ (IN\ x\ s \land IN\ y\ t)\ (x,\ y)))$

thm CARD_PRODUCT:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \land FINITE\ t \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%84::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%84\ (IN\ x\ s \land IN\ y\ t)\ (x,\ y))) = CARD\ s * CARD\ t$

thm HAS_SIZE_PRODUCT:

$\forall (s::?'b::type \Rightarrow bool)\ (m::nat)\ (t::?'a::type \Rightarrow bool)\ n::nat.\ HAS\_SIZE\ s\ m \land HAS\_SIZE\ t\ n \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%85::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%85\ (IN\ x\ s \land IN\ y\ t)\ (x,\ y)))\ (m * n)$

thm DEF_CROSS:

$CROSS = (\lambda (\_25953::?'b::type \Rightarrow bool)\ \_25954::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%86::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%86\ (IN\ x\ \_25953 \land IN\ y\ \_25954)\ (x,\ y)))$

thm CROSS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ CROSS\ s\ t = GSPEC\ (\lambda GEN\%PVAR\%86::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%86\ (IN\ x\ s \land IN\ y\ t)\ (x,\ y))$

thm IN_CROSS:

$\forall (x::?'b::type)\ (y::?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ IN\ (x, y)\ (CROSS\ s\ t) = (IN\ x\ s \wedge IN\ y\ t)$

thm HAS_SIZE_CROSS:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (m::nat)\ n::nat.\ HAS\_SIZE\ s\ m \wedge HAS\_SIZE\ t\ n \longrightarrow HAS\_SIZE\ (CROSS\ s\ t)\ (m * n)$

thm FINITE_CROSS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (CROSS\ s\ t)$

thm CARD_CROSS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow CARD\ (CROSS\ s\ t) = CARD\ s * CARD\ t$

thm CROSS_EQ_EMPTY:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (CROSS\ s\ t = EMPTY) = (s = EMPTY \vee t = EMPTY)$

thm HAS_SIZE_FUNSPACE:

$\forall (d::?'b::type)\ (n::nat)\ (t::?'b::type \Rightarrow bool)\ (m::nat)\ s::?'a::type \Rightarrow bool.\ HAS\_SIZE\ s\ m \wedge HAS\_SIZE\ t\ n \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%90::?'a::type \Rightarrow ?'b::type.\ \exists f::?'a::type \Rightarrow ?'b::type.\ SETSPEC\ GEN\%PVAR\%90\ ((\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall x::?'a::type.\ \neg\ IN\ x\ s \longrightarrow f\ x = d))\ f))\ n^m$

thm CARD_FUNSPACE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%91::?'b::type \Rightarrow ?'a::type.\ \exists f::?'b::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%91\ ((\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall x::?'b::type.\ \neg\ IN\ x\ s \longrightarrow f\ x = (?d::?'a::type)))\ f)) = (CARD\ t)^{CARD\ s}$

thm FINITE_FUNSPACE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%92::?'b::type \Rightarrow ?'a::type.\ \exists f::?'b::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%92\ ((\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall x::?'b::type.\ \neg\ IN\ x\ s \longrightarrow f\ x = (?d::?'a::type)))\ f))$

thm HAS_SIZE_FUNSPACE_UNIV:

$\forall (m::nat)\ n::nat.\ HAS\_SIZE\ HOL\_Light\_Import.UNIV\ m \wedge HAS\_SIZE\ HOL\_Light\_Import.UNIV\ n \longrightarrow HAS\_SIZE\ HOL\_Light\_Import.UNIV\ n^m$

thm CARD_FUNSPACE_UNIV:

$FINITE\ HOL\_Light\_Import.UNIV \wedge FINITE\ HOL\_Light\_Import.UNIV \longrightarrow CARD\ HOL\_Light\_Import.UNIV = (CARD\ HOL\_Light\_Import.UNIV)^{CARD\ HOL\_Light\_Import.UNIV}$

thm FINITE_FUNSPACE_UNIV:

FINITE HOL_Light_Import.UNIV ∧ *FINITE HOL_Light_Import.UNIV* ⟶ *FINITE HOL_Light_Import.UNIV*

thm HAS_SIZE_BOOL:

HAS_SIZE HOL_Light_Import.UNIV (*2*::*nat*)

thm CARD_BOOL:

CARD HOL_Light_Import.UNIV = (*2*::*nat*)

thm FINITE_BOOL:

FINITE HOL_Light_Import.UNIV

thm HAS_SIZE_POWERSET:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *n*::*nat*. *HAS_SIZE s n* ⟶ *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%95*::*?'a*::*type* ⇒ *bool*. ∃ *t*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%95* (*SUBSET t s*) *t*)) $(2::nat)^{n}$

thm CARD_POWERSET:

∀ *s*::*?'a*::*type* ⇒ *bool*. *FINITE s* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%96*::*?'a*::*type* ⇒ *bool*. ∃ *t*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%96* (*SUBSET t s*) *t*)) = $(2::nat)^{CARD\ s}$

thm FINITE_POWERSET:

∀ *s*::*?'a*::*type* ⇒ *bool*. *FINITE s* ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%97*::*?'a*::*type* ⇒ *bool*. ∃ *t*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%97* (*SUBSET t s*) *t*))

thm FINITE_UNIONS:

∀ *s*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *FINITE* (*UNIONS s*) = (*FINITE s* ∧ (∀ *t*::*?'a*::*type* ⇒ *bool*. *IN t s* ⟶ *FINITE t*))

thm SING_GSPEC_conjunct1:

∀ *a*::*?'a*::*type*. *GSPEC* (λ*GEN%PVAR%17*::*?'a*::*type*. ∃ *x*::*?'a*::*type*. *SETSPEC GEN%PVAR%17* (*a* = *x*) *x*) = *INSERT a EMPTY*

thm SING_GSPEC_conjunct0:

∀ *a*::*?'a*::*type*. *GSPEC* (λ*GEN%PVAR%16*::*?'a*::*type*. ∃ *x*::*?'a*::*type*. *SETSPEC GEN%PVAR%16* (*x* = *a*) *x*) = *INSERT a EMPTY*

thm POWERSET_CLAUSES_conjunct0:

GSPEC (λ*GEN%PVAR%98*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%98* (*SUBSET s EMPTY*) *s*) = *INSERT EMPTY EMPTY*

thm POWERSET_CLAUSES:

GSPEC (λ*GEN%PVAR%98*::*?'b*::*type* ⇒ *bool*. ∃ *s*::*?'b*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%98* (*SUBSET s EMPTY*) *s*) = *INSERT EMPTY EMPTY* ∧ (∀ (*a*::*?'a*::*type*) *t*::*?'a*::*type* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%99*::*?'a*::*type*

$\Rightarrow$ *bool*. $\exists$ *s*::*?'a*::*type* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%99* (*SUBSET s* (*INSERT a t*)) *s*) = *HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%100*::*?'a*::*type* $\Rightarrow$ *bool*. $\exists$ *s*::*?'a*::*type* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%100* (*SUBSET s t*) *s*)) (*IMAGE* (*INSERT a*) (*GSPEC* (λ*GEN%PVAR%101*::*?'a*::*type* $\Rightarrow$ *bool*. $\exists$ *s*::*?'a*::*type* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%101* (*SUBSET s t*) *s*))))

thm HAS_SIZE_NUMSEG_LT:

$\forall$ *n*::*nat*. *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%105*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%105* ($m < n$) *m*)) *n*

thm CARD_NUMSEG_LT:

$\forall$ *n*::*nat*. *CARD* (*GSPEC* (λ*GEN%PVAR%106*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%106* ($m < n$) *m*)) = *n*

thm FINITE_NUMSEG_LT:

$\forall$ *n*::*nat*. *FINITE* (*GSPEC* (λ*GEN%PVAR%107*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%107* ($m < n$) *m*))

thm HAS_SIZE_NUMSEG_LE:

$\forall$ *n*::*nat*. *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%108*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%108* ($m \leq n$) *m*)) (n + (*1*::*nat*))

thm FINITE_NUMSEG_LE:

$\forall$ *n*::*nat*. *FINITE* (*GSPEC* (λ*GEN%PVAR%109*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%109* ($m \leq n$) *m*))

thm CARD_NUMSEG_LE:

$\forall$ *n*::*nat*. *CARD* (*GSPEC* (λ*GEN%PVAR%110*::*nat*. $\exists$ *m*::*nat*. *SETSPEC GEN%PVAR%110* ($m \leq n$) *m*)) = n + (*1*::*nat*)

thm num_FINITE:

$\forall$ *s*::*nat* $\Rightarrow$ *bool*. *FINITE s* = ($\exists$ *a*::*nat*. $\forall$ *x*::*nat*. *IN x s* $\longrightarrow$ $x \leq a$)

thm num_FINITE_AVOID:

$\forall$ *s*::*nat* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ ($\exists$ *a*::*nat*. $\neg$ *IN a s*)

thm Misc_defs_and_lemmas.num_infinite:

$\neg$ *FINITE HOL_Light_Import*.*UNIV*

thm num_INFINITE:

INFINITE HOL_Light_Import.*UNIV*

thm string_INFINITE:

INFINITE HOL_Light_Import.*UNIV*

thm FINITE_REAL_INTERVAL:

$(\forall\, a{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%120{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%120\ (a < x)\ x))) \wedge (\forall\, a{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%121{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%121\ (a \leq x)\ x))) \wedge (\forall\, b{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%122{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%122\ (x < b)\ x))) \wedge (\forall\, b{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%123{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%123\ (x \leq b)\ x))) \wedge (\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%124{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%124\ (a < x \wedge x < b)\ x)) = (b \leq a)) \wedge (\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%125{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%125\ (a \leq x \wedge x < b)\ x)) = (b \leq a)) \wedge (\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%126{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%126\ (a < x \wedge x \leq b)\ x)) = (b \leq a)) \wedge (\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%127{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%127\ (a \leq x \wedge x \leq b)\ x)) = (b \leq a))$

thm FINITE_REAL_INTERVAL_conjunct7:

$\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%127{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%127\ (a \leq x \wedge x \leq b)\ x)) = (b \leq a)$

thm FINITE_REAL_INTERVAL_conjunct6:

$\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%126{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%126\ (a < x \wedge x \leq b)\ x)) = (b \leq a)$

thm FINITE_REAL_INTERVAL_conjunct5:

$\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%125{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%125\ (a \leq x \wedge x < b)\ x)) = (b \leq a)$

thm FINITE_REAL_INTERVAL_conjunct4:

$\forall\, (a{::}real)\ b{::}real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%124{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%124\ (a < x \wedge x < b)\ x)) = (b \leq a)$

thm FINITE_REAL_INTERVAL_conjunct3:

$\forall\, b{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%123{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%123\ (x \leq b)\ x))$

thm FINITE_REAL_INTERVAL_conjunct2:

$\forall\, b{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%122{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%122\ (x < b)\ x))$

thm FINITE_REAL_INTERVAL_conjunct1:

$\forall\, a{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%121{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%121\ (a \leq x)\ x))$

thm FINITE_REAL_INTERVAL_conjunct0:

$\forall\, a{::}real.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%120{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%120\ (a < x)\ x))$

thm real_INFINITE:

INFINITE HOL_Light_Import.UNIV

thm HAS_SIZE_INDEX:

$\forall (s::?'a::type \Rightarrow bool)\ n::nat.\ HAS\_SIZE\ s\ n \longrightarrow (\exists f::nat \Rightarrow ?'a::type.\ (\forall m<n.\ IN\ (f\ m)\ s) \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (\exists !m::nat.\ m < n \wedge f\ m = x)))$

thm DEF_set_of_list:

$set\_of\_list = (SOME\ set\_of\_list::nat \Rightarrow ?'a::type\ list \Rightarrow ?'a::type \Rightarrow bool.\ \forall \_31730::nat.\ set\_of\_list\ \_31730\ [] = EMPTY \wedge (\forall (h::?'a::type)\ t::?'a::type\ list.\ set\_of\_list\ \_31730\ (h\ \#\ t) = INSERT\ h\ (set\_of\_list\ \_31730\ t)))\ (45::nat)$

thm set_of_list_conjunct0:

$set\_of\_list\ [] = EMPTY$

thm set_of_list_conjunct1:

$set\_of\_list\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = INSERT\ ?h\ (set\_of\_list\ ?t)$

thm set_of_list:

$set\_of\_list\ [] = EMPTY \wedge set\_of\_list\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = INSERT\ ?h\ (set\_of\_list\ ?t)$

thm DEF_list_of_set:

$list\_of\_set = (\lambda\_31731::?'a::type \Rightarrow bool.\ SOME\ l::?'a::type\ list.\ set\_of\_list\ l = \_31731 \wedge length\ l = CARD\ \_31731)$

thm list_of_set:

$\forall s::?'a::type \Rightarrow bool.\ list\_of\_set\ s = (SOME\ l::?'a::type\ list.\ set\_of\_list\ l = s \wedge length\ l = CARD\ s)$

thm LIST_OF_SET_PROPERTIES:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow set\_of\_list\ (list\_of\_set\ s) = s \wedge length\ (list\_of\_set\ s) = CARD\ s$

thm SET_OF_LIST_OF_SET:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow set\_of\_list\ (list\_of\_set\ s) = s$

thm LENGTH_LIST_OF_SET:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow length\ (list\_of\_set\ s) = CARD\ s$

thm MEM_LIST_OF_SET:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow (\forall x::?'a::type.\ MEM\ x\ (list\_of\_set\ s) = IN\ x\ s)$

thm FINITE_SET_OF_LIST:

$\forall l::?'a::type\ list.\ FINITE\ (set\_of\_list\ l)$

thm IN_SET_OF_LIST:

$\forall (x::?'a::type)\ l::?'a::type\ list.\ IN\ x\ (set\_of\_list\ l) = MEM\ x\ l$

thm SET_OF_LIST_APPEND:

$\forall (l1::?'a::type\ list)\ l2::?'a::type\ list.\ set\_of\_list\ (l1\ @\ l2) = HOL\_Light\_Import.UNION\ (set\_of\_list\ l1)\ (set\_of\_list\ l2)$

thm SET_OF_LIST_MAP:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ l::?'b::type\ list.\ set\_of\_list\ (map\ f\ l) = IMAGE\ f\ (set\_of\_list\ l)$

thm SET_OF_LIST_EQ_EMPTY:

$\forall l::?'a::type\ list.\ (set\_of\_list\ l = EMPTY) = (l = [])$

thm DEF_pairwise:

$pairwise = (\lambda(\_31890::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ \_31891::?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ \_31891 \wedge IN\ y\ \_31891 \wedge x \neq y \longrightarrow \_31890\ x\ y)$

thm pairwise:

$\forall (s::?'a::type \Rightarrow bool)\ r::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ pairwise\ r\ s = (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow r\ x\ y)$

thm DEF_PAIRWISE:

$PAIRWISE = (SOME\ PAIRWISE::nat \Rightarrow (?'a::type \Rightarrow ?'a::type \Rightarrow bool) \Rightarrow ?'a::type\ list \Rightarrow bool.\ \forall \_31908::nat.\ (\forall r::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ PAIRWISE\ \_31908\ r\ [] = True) \wedge (\forall (h::?'a::type)\ (r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ t::?'a::type\ list.\ PAIRWISE\ \_31908\ r\ (h\ \#\ t) = (list\_all\ (r\ h)\ t \wedge PAIRWISE\ \_31908\ r\ t)))\ (46::nat)$

thm PAIRWISE_conjunct0:

$PAIRWISE\ (?r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ [] = True$

thm PAIRWISE_conjunct1:

$PAIRWISE\ (?r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (list\_all\ (?r\ ?h)\ ?t \wedge PAIRWISE\ ?r\ ?t)$

thm PAIRWISE:

$PAIRWISE\ (?r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ [] = True \wedge PAIRWISE\ ?r\ ((?h::?'a::type)\ \#\ (?t::?'a::type\ list)) = (list\_all\ (?r\ ?h)\ ?t \wedge PAIRWISE\ ?r\ ?t)$

thm PAIRWISE_EMPTY:

$\forall r::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ pairwise\ r\ EMPTY = True$

thm PAIRWISE_SING:

$\forall (r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ pairwise\ r\ (INSERT\ x\ EMPTY) = True$

thm PAIRWISE_MONO:

$\forall (r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.$ $pairwise\ r\ s \wedge SUBSET\ t\ s \longrightarrow pairwise\ r\ t$

thm PAIRWISE_INSERT:

$\forall (r::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ pairwise\ r\ (INSERT\ x\ s) = ((\forall y::?'a::type.\ IN\ y\ s \wedge y \neq x \longrightarrow r\ x\ y \wedge r\ y\ x) \wedge pairwise\ r\ s)$

thm PAIRWISE_IMAGE:

$\forall (r::?'b::type \Rightarrow ?'b::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'b::type.\ pairwise\ r\ (IMAGE\ f\ (?s::?'a::type \Rightarrow bool)) = pairwise\ (\lambda(x::?'a::type)\ y::?'a::type.\ f\ x \neq f\ y \longrightarrow r\ (f\ x)\ (f\ y))\ ?s$

thm CARD_SET_OF_LIST_LE:

$\forall l::?'a::type\ list.\ CARD\ (set\_of\_list\ l) \leq length\ l$

thm HAS_SIZE_SET_OF_LIST:

$\forall l::?'a::type\ list.\ HAS\_SIZE\ (set\_of\_list\ l)\ (length\ l) = PAIRWISE\ op \neq l$

thm SURJECTIVE_IFF_INJECTIVE_GEN:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ s \wedge FINITE\ t \wedge CARD\ s = CARD\ t \wedge SUBSET\ (IMAGE\ f\ s)\ t \longrightarrow (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists x::?'b::type.\ IN\ x\ s \wedge f\ x = y)) = (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y)$

thm SURJECTIVE_IFF_INJECTIVE:

$\forall (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge SUBSET\ (IMAGE\ f\ s)\ s \longrightarrow (\forall y::?'a::type.\ IN\ y\ s \longrightarrow (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x = y)) = (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y)$

thm IMAGE_IMP_INJECTIVE_GEN:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ s \wedge CARD\ s = CARD\ t \wedge IMAGE\ f\ s = t \longrightarrow (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y)$

thm IMAGE_IMP_INJECTIVE:

$\forall (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge IMAGE\ f\ s = s \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y)$

thm CARD_LE_INJ:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge CARD\ s \leq CARD\ t \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ SUBSET\ (IMAGE\ f\ s)\ t \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y))$

thm FORALL_IN_CLAUSES:

$(\forall P::?'b::type \Rightarrow bool.\ (\forall x::?'b::type.\ IN\ x\ EMPTY \longrightarrow P\ x) = True) \wedge (\forall (P::?'a::type \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ (INSERT\ a\ s) \longrightarrow P\ x) = (P\ a \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow P\ x)))$

thm EXISTS_IN_CLAUSES:

$(\forall P::?'b::type \Rightarrow bool.\ (\exists x::?'b::type.\ IN\ x\ EMPTY \wedge P\ x) = False) \wedge (\forall (P::?'a::type \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ (\exists x::?'a::type.\ IN\ x\ (INSERT\ a\ s) \wedge P\ x) = (P\ a \vee (\exists x::?'a::type.\ IN\ x\ s \wedge P\ x)))$

thm INJECTIVE_ON_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ u::?'b::type \Rightarrow bool.\ (\forall (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ SUBSET\ s\ u \wedge SUBSET\ t\ u \wedge IMAGE\ f\ s = IMAGE\ f\ t \longrightarrow s = t) = (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ u \wedge IN\ y\ u \wedge f\ x = f\ y \longrightarrow x = y)$

thm INJECTIVE_IMAGE:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ IMAGE\ f\ s = IMAGE\ f\ t \longrightarrow s = t) = (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y)$

thm SURJECTIVE_ON_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (u::?'b::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ (\forall t::?'a::type \Rightarrow bool.\ SUBSET\ t\ v \longrightarrow (\exists s::?'b::type \Rightarrow bool.\ SUBSET\ s\ u \wedge IMAGE\ f\ s = t)) = (\forall y::?'a::type.\ IN\ y\ v \longrightarrow (\exists x::?'b::type.\ IN\ x\ u \wedge f\ x = y))$

thm SURJECTIVE_IMAGE:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall t::?'a::type \Rightarrow bool.\ \exists s::?'b::type \Rightarrow bool.\ IMAGE\ f\ s = t) = (\forall y::?'a::type.\ \exists x::?'b::type.\ f\ x = y)$

thm CARD_EQ_BIJECTION:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge CARD\ s = CARD\ t \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \wedge (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists x::?'b::type.\ IN\ x\ s \wedge f\ x = y)) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y))$

thm CARD_EQ_BIJECTIONS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge CARD\ s = CARD\ t \longrightarrow (\exists (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::?'a::type.\ IN\ y\ t \longrightarrow IN\ (g\ y)\ s \wedge f\ (g\ y) = y))$

thm BIJECTIONS_HAS_SIZE:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::?'a::type.\ IN\ y\ t \longrightarrow IN\ (g\ y)\ s \wedge f\ (g\ y) = y) \wedge HAS\_SIZE\ s\ (?n::nat) \longrightarrow HAS\_SIZE\ t\ ?n$

thm BIJECTIONS_HAS_SIZE_EQ:

$$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'b::type.\ \mathit{IN}\ x\ s \longrightarrow \mathit{IN}\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::?'a::type.\ \mathit{IN}\ y\ t \longrightarrow \mathit{IN}\ (g\ y)\ s \wedge f\ (g\ y) = y) \longrightarrow (\forall n::nat.\ \mathit{HAS\_SIZE}\ s\ n = \mathit{HAS\_SIZE}\ t\ n)$$

thm BIJECTIONS_CARD_EQ:

$$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (\mathit{FINITE}\ s \vee \mathit{FINITE}\ t) \wedge (\forall x::?'b::type.\ \mathit{IN}\ x\ s \longrightarrow \mathit{IN}\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::?'a::type.\ \mathit{IN}\ y\ t \longrightarrow \mathit{IN}\ (g\ y)\ s \wedge f\ (g\ y) = y) \longrightarrow \mathit{CARD}\ s = \mathit{CARD}\ t$$

thm WF_FINITE:

$$\forall <<::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ \neg << x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ << x\ y \wedge << y\ z \longrightarrow << x\ z) \wedge (\forall x::?'a::type.\ \mathit{FINITE}\ (\mathit{GSPEC}\ (\lambda \mathit{GEN\%PVAR\%131}::?'a::type.\ \exists y::?'a::type.\ \mathit{SETSPEC}\ \mathit{GEN\%PVAR\%131}\ (<< y\ x)\ y))) \longrightarrow \mathit{WF}\ <<$$

thm DEF_<=_c:

$$<=\_c = (\lambda(\_36891::?'b::type \Rightarrow bool)\ \_36892::?'a::type \Rightarrow bool.\ \exists f::?'b::type \Rightarrow ?'a::type.\ (\forall x::?'b::type.\ \mathit{IN}\ x\ \_36891 \longrightarrow \mathit{IN}\ (f\ x)\ \_36892) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ \mathit{IN}\ x\ \_36891 \wedge \mathit{IN}\ y\ \_36891 \wedge f\ x = f\ y \longrightarrow x = y))$$

thm le_c:

$$\forall (t::?'b::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ <=\_c\ s\ t = (\exists f::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'a::type.\ \mathit{IN}\ x\ s \longrightarrow \mathit{IN}\ (f\ x)\ t) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ \mathit{IN}\ x\ s \wedge \mathit{IN}\ y\ s \wedge f\ x = f\ y \longrightarrow x = y))$$

thm DEF_<_c:

$$<\_c = (\lambda(\_36903::?'b::type \Rightarrow bool)\ \_36904::?'a::type \Rightarrow bool.\ <=\_c\ \_36903\ \_36904 \wedge \neg <=\_c\ \_36904\ \_36903)$$

thm lt_c:

$$\forall (t::?'b::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ <\_c\ s\ t = (<=\_c\ s\ t \wedge \neg <=\_c\ t\ s)$$

thm DEF_=_c:

$$=\_c = (\lambda(\_36915::?'b::type \Rightarrow bool)\ \_36916::?'a::type \Rightarrow bool.\ \exists f::?'b::type \Rightarrow ?'a::type.\ (\forall x::?'b::type.\ \mathit{IN}\ x\ \_36915 \longrightarrow \mathit{IN}\ (f\ x)\ \_36916) \wedge (\forall y::?'a::type.\ \mathit{IN}\ y\ \_36916 \longrightarrow (\exists! x::?'b::type.\ \mathit{IN}\ x\ \_36915 \wedge f\ x = y)))$$

thm eq_c:

$$\forall (t::?'b::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ =\_c\ s\ t = (\exists f::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'a::type.\ \mathit{IN}\ x\ s \longrightarrow \mathit{IN}\ (f\ x)\ t) \wedge (\forall y::?'b::type.\ \mathit{IN}\ y\ t \longrightarrow (\exists! x::?'a::type.\ \mathit{IN}\ x\ s \wedge f\ x = y)))$$

thm DEF_>=_c:

$>=\_c = (\lambda(\_36927::?'b::type \Rightarrow bool)\ \_36928::?'a::type \Rightarrow bool.\ <=\_c\ \_36928\ \_36927)$

thm ge_c:

$\forall (t::?'b::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ >=\_c\ s\ t = <=\_c\ t\ s$

thm DEF_>_c:

$>\_c = (\lambda(\_36939::?'b::type \Rightarrow bool)\ \_36940::?'a::type \Rightarrow bool.\ <\_c\ \_36940\ \_36939)$

thm gt_c:

$\forall (t::?'b::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ >\_c\ s\ t = <\_c\ t\ s$

thm LE_C:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t = (\exists g::?'a::type \Rightarrow ?'b::type.\ \forall x::?'b::type.\ IN\ x\ s \longrightarrow (\exists y::?'a::type.\ IN\ y\ t \wedge g\ y = x))$

thm GE_C:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ >=\_c\ s\ t = (\exists f::?'b::type \Rightarrow ?'a::type.\ \forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists x::?'b::type.\ IN\ x\ s \wedge y = f\ x))$

thm DEF_COUNTABLE:

$COUNTABLE = >=\_c\ HOL\_Light\_Import.UNIV$

thm COUNTABLE:

$\forall t::?'a::type \Rightarrow bool.\ COUNTABLE\ t = >=\_c\ HOL\_Light\_Import.UNIV\ t$

thm DEF_sup:

$HOL\_Light\_Import.sup = (\lambda\_37095::real \Rightarrow bool.\ SOME\ a::real.\ (\forall x::real.\ IN\ x\ \_37095 \longrightarrow x \le a) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ \_37095 \longrightarrow x \le b) \longrightarrow a \le b))$

thm sup:

$\forall s::real \Rightarrow bool.\ HOL\_Light\_Import.sup\ s = (SOME\ a::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow x \le a) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow x \le b) \longrightarrow a \le b))$

thm SUP_EQ:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow x \le b) = (\forall x::real.\ IN\ x\ t \longrightarrow x \le b)) \longrightarrow HOL\_Light\_Import.sup\ s = HOL\_Light\_Import.sup\ t$

thm SUP:

$\forall s::real \Rightarrow bool.\ s \neq EMPTY \wedge (\exists b::real.\ \forall x::real.\ IN\ x\ s \longrightarrow x \le b) \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow x \le HOL\_Light\_Import.sup\ s) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow x \le b) \longrightarrow HOL\_Light\_Import.sup\ s \le b)$

thm SUP_FINITE_LEMMA:

$\forall\, s{::}real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (\exists\, b{::}real.\ IN\ b\ s \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow x \leq b))$

thm SUP_FINITE:

$\forall\, s{::}real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow IN\ (HOL\_Light\_Import.sup\ s)\ s \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow x \leq HOL\_Light\_Import.sup\ s)$

thm REAL_LE_SUP_FINITE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (a \leq HOL\_Light\_Import.sup\ s) = (\exists\, x{::}real.\ IN\ x\ s \wedge a \leq x)$

thm REAL_SUP_LE_FINITE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.sup\ s \leq a) = (\forall\, x{::}real.\ IN\ x\ s \longrightarrow x \leq a)$

thm REAL_LT_SUP_FINITE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (a < HOL\_Light\_Import.sup\ s) = (\exists\, x{::}real.\ IN\ x\ s \wedge a < x)$

thm REAL_SUP_LT_FINITE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.sup\ s < a) = (\forall\, x{::}real.\ IN\ x\ s \longrightarrow x < a)$

thm REAL_SUP_UNIQUE:

$\forall\, (s{::}real \Rightarrow bool)\ b{::}real.\ (\forall\, x{::}real.\ IN\ x\ s \longrightarrow x \leq b) \wedge (\forall\, b'{<}b.\ \exists\, x{::}real.\ IN\ x\ s \wedge b' < x) \longrightarrow HOL\_Light\_Import.sup\ s = b$

thm REAL_SUP_LE:

$\forall\, b{::}real.\ (?s{::}real \Rightarrow bool) \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ ?s \longrightarrow x \leq b) \longrightarrow HOL\_Light\_Import.sup\ ?s \leq b$

thm REAL_SUP_LE_SUBSET:

$\forall\, (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ s \neq EMPTY \wedge SUBSET\ s\ t \wedge (\exists\, b{::}real.\ \forall\, x{::}real.\ IN\ x\ t \longrightarrow x \leq b) \longrightarrow HOL\_Light\_Import.sup\ s \leq HOL\_Light\_Import.sup\ t$

thm REAL_SUP_BOUNDS:

$\forall\, (s{::}real \Rightarrow bool)\ (a{::}real)\ b{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow a \leq x \wedge x \leq b) \longrightarrow a \leq HOL\_Light\_Import.sup\ s \wedge HOL\_Light\_Import.sup\ s \leq b$

thm REAL_ABS_SUP_LE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow |x| \leq a) \longrightarrow |HOL\_Light\_Import.sup\ s| \leq a$

thm REAL_SUP_ASCLOSE:

$\forall\, (s{::}real \Rightarrow bool)\ (l{::}real)\ e{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow |x - l| \leq e) \longrightarrow |HOL\_Light\_Import.sup\ s - l| \leq e$

thm DEF_inf:

$HOL\_Light\_Import.inf = (\lambda\_37591::real \Rightarrow bool.\ SOME\ a::real.\ (\forall x::real.\ IN\ x\ \_37591 \longrightarrow a \leq x) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ \_37591 \longrightarrow b \leq x) \longrightarrow b \leq a))$

thm inf:

$\forall s::real \Rightarrow bool.\ HOL\_Light\_Import.inf\ s = (SOME\ a::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow a \leq x) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow b \leq x) \longrightarrow b \leq a))$

thm INF_EQ:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ (\forall a::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow a \leq x) = (\forall x::real.\ IN\ x\ t \longrightarrow a \leq x)) \longrightarrow HOL\_Light\_Import.inf\ s = HOL\_Light\_Import.inf\ t$

thm INF:

$\forall s::real \Rightarrow bool.\ s \neq EMPTY \wedge (\exists b::real.\ \forall x::real.\ IN\ x\ s \longrightarrow b \leq x) \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.inf\ s \leq x) \wedge (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow b \leq x) \longrightarrow b \leq HOL\_Light\_Import.inf\ s)$

thm INF_FINITE_LEMMA:

$\forall s::real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (\exists b::real.\ IN\ b\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow b \leq x))$

thm INF_FINITE:

$\forall s::real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow IN\ (HOL\_Light\_Import.inf\ s)\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.inf\ s \leq x)$

thm REAL_LE_INF_FINITE:

$\forall (s::real \Rightarrow bool)\ a::real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (a \leq HOL\_Light\_Import.inf\ s) = (\forall x::real.\ IN\ x\ s \longrightarrow a \leq x)$

thm REAL_INF_LE_FINITE:

$\forall (s::real \Rightarrow bool)\ a::real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.inf\ s \leq a) = (\exists x::real.\ IN\ x\ s \wedge x \leq a)$

thm REAL_LT_INF_FINITE:

$\forall (s::real \Rightarrow bool)\ a::real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (a < HOL\_Light\_Import.inf\ s) = (\forall x::real.\ IN\ x\ s \longrightarrow a < x)$

thm REAL_INF_LT_FINITE:

$\forall (s::real \Rightarrow bool)\ a::real.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.inf\ s < a) = (\exists x::real.\ IN\ x\ s \wedge x < a)$

thm REAL_INF_UNIQUE:

$\forall (s::real \Rightarrow bool)\ b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow b \leq x) \wedge (\forall b'>b.\ \exists x::real.\ IN\ x\ s \wedge x < b') \longrightarrow HOL\_Light\_Import.inf\ s = b$

thm REAL_LE_INF:

$\forall\, b{::}real.\ (?s{::}real \Rightarrow bool) \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ ?s \longrightarrow b \leq x) \longrightarrow b \leq HOL\_Light\_Import.inf\ ?s$

thm REAL_LE_INF_SUBSET:

$\forall\, (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ t \neq EMPTY \wedge SUBSET\ t\ s \wedge (\exists\, b{::}real.\ \forall\, x{::}real.\ IN\ x\ s \longrightarrow b \leq x) \longrightarrow HOL\_Light\_Import.inf\ s \leq HOL\_Light\_Import.inf\ t$

thm REAL_INF_BOUNDS:

$\forall\, (s{::}real \Rightarrow bool)\ (a{::}real)\ b{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow a \leq x \wedge x \leq b) \longrightarrow a \leq HOL\_Light\_Import.inf\ s \wedge HOL\_Light\_Import.inf\ s \leq b$

thm REAL_ABS_INF_LE:

$\forall\, (s{::}real \Rightarrow bool)\ a{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow |x| \leq a) \longrightarrow |HOL\_Light\_Import.inf\ s| \leq a$

thm REAL_INF_ASCLOSE:

$\forall\, (s{::}real \Rightarrow bool)\ (l{::}real)\ e{::}real.\ s \neq EMPTY \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow |x - l| \leq e) \longrightarrow |HOL\_Light\_Import.inf\ s - l| \leq e$

thm SUP_UNIQUE_FINITE:

$\forall\, s{::}real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.sup\ s = (?a{::}real)) = (IN\ ?a\ s \wedge (\forall\, y{::}real.\ IN\ y\ s \longrightarrow y \leq ?a))$

thm INF_UNIQUE_FINITE:

$\forall\, s{::}real \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \longrightarrow (HOL\_Light\_Import.inf\ s = (?a{::}real)) = (IN\ ?a\ s \wedge (\forall\, y{::}real.\ IN\ y\ s \longrightarrow ?a \leq y))$

thm SUP_INSERT_FINITE:

$\forall\, (x{::}real)\ s{::}real \Rightarrow bool.\ FINITE\ s \longrightarrow HOL\_Light\_Import.sup\ (INSERT\ x\ s) = (if\ s = EMPTY\ then\ x\ else\ max\ x\ (HOL\_Light\_Import.sup\ s))$

thm SUP_SING:

$\forall\, a{::}real.\ HOL\_Light\_Import.sup\ (INSERT\ a\ EMPTY) = a$

thm INF_INSERT_FINITE:

$\forall\, (x{::}real)\ s{::}real \Rightarrow bool.\ FINITE\ s \longrightarrow HOL\_Light\_Import.inf\ (INSERT\ x\ s) = (if\ s = EMPTY\ then\ x\ else\ min\ x\ (HOL\_Light\_Import.inf\ s))$

thm INF_SING:

$\forall\, a{::}real.\ HOL\_Light\_Import.inf\ (INSERT\ a\ EMPTY) = a$

thm REAL_SUP_EQ_INF:

$\forall\, s{::}real \Rightarrow bool.\ s \neq EMPTY \wedge (\exists\, B{::}real.\ \forall\, x{::}real.\ IN\ x\ s \longrightarrow |x| \leq B) \longrightarrow (HOL\_Light\_Import.sup\ s = HOL\_Light\_Import.inf\ s) = (\exists\, a{::}real.\ s = INSERT\ a\ EMPTY)$

thm DEF_..:

$dotdot = (\lambda(\_38741::nat)\ \_38742::nat.\ GSPEC\ (\lambda GEN\%PVAR\%132::nat.\ \exists\, x::nat.\ SETSPEC\ GEN\%PVAR\%132\ (\_38741 \le x \land x \le \_38742)\ x))$

thm numseg:

$\forall\,(m::nat)\ n::nat.\ dotdot\ m\ n = GSPEC\ (\lambda GEN\%PVAR\%132::nat.\ \exists\, x::nat.\ SETSPEC\ GEN\%PVAR\%132\ (m \le x \land x \le n)\ x)$

thm FINITE_NUMSEG:

$\forall\,(m::nat)\ n::nat.\ FINITE\ (dotdot\ m\ n)$

thm NUMSEG_COMBINE_R:

$\forall\,(m::nat)\ (p::nat)\ n::nat.\ m \le p + (1::nat) \land p \le n \longrightarrow HOL\_Light\_Import.UNION\ (dotdot\ m\ p)\ (dotdot\ (p + (1::nat))\ n) = dotdot\ m\ n$

thm NUMSEG_COMBINE_L:

$\forall\,(m::nat)\ (p::nat)\ n::nat.\ m \le p \land p \le n + (1::nat) \longrightarrow HOL\_Light\_Import.UNION\ (dotdot\ m\ (p - (1::nat)))\ (dotdot\ p\ n) = dotdot\ m\ n$

thm NUMSEG_LREC:

$\forall\,(m::nat)\ n::nat.\ m \le n \longrightarrow INSERT\ m\ (dotdot\ (m + (1::nat))\ n) = dotdot\ m\ n$

thm NUMSEG_RREC:

$\forall\,(m::nat)\ n::nat.\ m \le n \longrightarrow INSERT\ n\ (dotdot\ m\ (n - (1::nat))) = dotdot\ m\ n$

thm NUMSEG_REC:

$\forall\,(m::nat)\ n::nat.\ m \le Suc\ n \longrightarrow dotdot\ m\ (Suc\ n) = INSERT\ (Suc\ n)\ (dotdot\ m\ n)$

thm IN_NUMSEG:

$\forall\,(m::nat)\ (n::nat)\ p::nat.\ IN\ p\ (dotdot\ m\ n) = (m \le p \land p \le n)$

thm IN_NUMSEG_0:

$\forall\,(m::nat)\ n::nat.\ IN\ m\ (dotdot\ (0::nat)\ n) = (m \le n)$

thm NUMSEG_SING:

$\forall\, n::nat.\ dotdot\ n\ n = INSERT\ n\ EMPTY$

thm NUMSEG_EMPTY:

$\forall\,(m::nat)\ n::nat.\ (dotdot\ m\ n = EMPTY) = (n < m)$

thm CARD_NUMSEG_LEMMA:

$\forall\,(m::nat)\ d::nat.\ CARD\ (dotdot\ m\ (m + d)) = d + (1::nat)$

thm CARD_NUMSEG:

$\forall (m::nat)\ n::nat.\ CARD\ (dotdot\ m\ n) = n + (1::nat) - m$

thm HAS_SIZE_NUMSEG:

$\forall (m::nat)\ n::nat.\ HAS\_SIZE\ (dotdot\ m\ n)\ (n + (1::nat) - m)$

thm CARD_NUMSEG_1:

$\forall n::nat.\ CARD\ (dotdot\ (1::nat)\ n) = n$

thm HAS_SIZE_NUMSEG_1:

$\forall n::nat.\ HAS\_SIZE\ (dotdot\ (1::nat)\ n)\ n$

thm NUMSEG_CLAUSES_conjunct1:

$\forall (m::nat)\ n::nat.\ dotdot\ m\ (Suc\ n) = (if\ m \leq Suc\ n\ then\ INSERT\ (Suc\ n)\ (dotdot\ m\ n)\ else\ dotdot\ m\ n)$

thm NUMSEG_CLAUSES_conjunct0:

$\forall m::nat.\ dotdot\ m\ (0::nat) = (if\ m = (0::nat)\ then\ INSERT\ (0::nat)\ EMPTY\ else\ EMPTY)$

thm NUMSEG_CLAUSES:

$(\forall m::nat.\ dotdot\ m\ (0::nat) = (if\ m = (0::nat)\ then\ INSERT\ (0::nat)\ EMPTY\ else\ EMPTY)) \wedge (\forall (m::nat)\ n::nat.\ dotdot\ m\ (Suc\ n) = (if\ m \leq Suc\ n\ then\ INSERT\ (Suc\ n)\ (dotdot\ m\ n)\ else\ dotdot\ m\ n))$

thm FINITE_INDEX_NUMSEG:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s = (\exists f::nat \Rightarrow ?'a::type.\ (\forall (i::nat)\ j::nat.\ IN\ i\ (dotdot\ (1::nat)\ (CARD\ s)) \wedge IN\ j\ (dotdot\ (1::nat)\ (CARD\ s)) \wedge f\ i = f\ j \longrightarrow i = j) \wedge s = IMAGE\ f\ (dotdot\ (1::nat)\ (CARD\ s)))$

thm FINITE_INDEX_NUMBERS:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s = (\exists (k::nat \Rightarrow bool)\ f::nat \Rightarrow ?'a::type.\ (\forall (i::nat)\ j::nat.\ IN\ i\ k \wedge IN\ j\ k \wedge f\ i = f\ j \longrightarrow i = j) \wedge FINITE\ k \wedge s = IMAGE\ f\ k)$

thm DISJOINT_NUMSEG:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ DISJOINT\ (dotdot\ m\ n)\ (dotdot\ p\ q) = (n < p \vee q < m \vee n < m \vee q < p)$

thm NUMSEG_ADD_SPLIT:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m \leq n + (1::nat) \longrightarrow dotdot\ m\ (n + p) = HOL\_Light\_Import.UNION\ (dotdot\ m\ n)\ (dotdot\ (n + (1::nat))\ (n + p))$

thm NUMSEG_OFFSET_IMAGE:

$\forall (m::nat)\ (n::nat)\ p::nat.\ dotdot\ (m + p)\ (n + p) = IMAGE\ (\lambda i::nat.\ i + p)\ (dotdot\ m\ n)$

thm SUBSET_NUMSEG:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ SUBSET\ (dotdot\ m\ n)\ (dotdot\ p\ q) = (n < m \lor p \leq m \land n \leq q)$

thm NUMSEG_LE:

$\forall n::nat.\ GSPEC\ (\lambda GEN\%PVAR\%134::nat.\ \exists x::nat.\ SETSPEC\ GEN\%PVAR\%134\ (x \leq n)\ x) = dotdot\ (0::nat)\ n$

thm NUMSEG_LT:

$\forall n::nat.\ GSPEC\ (\lambda GEN\%PVAR\%135::nat.\ \exists x::nat.\ SETSPEC\ GEN\%PVAR\%135\ (x < n)\ x) = (if\ n = (0::nat)\ then\ EMPTY\ else\ dotdot\ (0::nat)\ (n - (1::nat)))$

thm TOPOLOGICAL_SORT:

$\forall <<::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ <<\ x\ y \land <<\ y\ x \longrightarrow x = y) \land (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ <<\ x\ y \land <<\ y\ z \longrightarrow <<\ x\ z) \longrightarrow (\forall (n::nat)\ s::?'a::type \Rightarrow bool.\ HAS\_SIZE\ s\ n \longrightarrow (\exists f::nat \Rightarrow ?'a::type.\ s = IMAGE\ f\ (dotdot\ (1::nat)\ n) \land (\forall (j::nat)\ k::nat.\ IN\ j\ (dotdot\ (1::nat)\ n) \land IN\ k\ (dotdot\ (1::nat)\ n) \land j < k \longrightarrow \neg <<\ (f\ k)\ (f\ j))))$

thm DEF_neutral:

$neutral = (\lambda\_40728::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ SOME\ x::?'a::type.\ \forall y::?'a::type.\ \_40728\ x\ y = y \land \_40728\ y\ x = y)$

thm neutral:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ neutral\ op = (SOME\ x::?'a::type.\ \forall y::?'a::type.\ op\ x\ y = y \land op\ y\ x = y)$

thm DEF_monoidal:

$monoidal = (\lambda\_40733::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ (\forall (x::?'a::type)\ y::?'a::type.\ \_40733\ x\ y = \_40733\ y\ x) \land (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ \_40733\ x\ (\_40733\ y\ z) = \_40733\ (\_40733\ x\ y)\ z) \land (\forall x::?'a::type.\ \_40733\ (neutral\ \_40733)\ x = x))$

thm monoidal:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op = ((\forall (x::?'a::type)\ y::?'a::type.\ op\ x\ y = op\ y\ x) \land (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ op\ x\ (op\ y\ z) = op\ (op\ x\ y)\ z) \land (\forall x::?'a::type.\ op\ (neutral\ op)\ x = x))$

thm MONOIDAL_AC:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow (\forall a::?'a::type.\ op\ (neutral\ op)\ a = a) \land (\forall a::?'a::type.\ op\ a\ (neutral\ op) = a) \land (\forall (a::?'a::type)\ b::?'a::type.\ op\ a\ b = op\ b\ a) \land (\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ op\ (op\ a\ b)\ c = op\ a\ (op\ b\ c)) \land (\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ op\ a\ (op\ b\ c) = op\ b\ (op\ a\ c))$

thm DEF_support:

$support = (\lambda(\_40818{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (\_40819{::}?'a{::}type \Rightarrow ?'b{::}type)\ \_40820{::}?'a{::}type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%140{::}?'a{::}type.\ \exists x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%140\ (IN\ x\ \_40820 \wedge \_40819\ x \neq neutral\ \_40818)\ x))$

thm support:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ op{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow ?'a{::}type.\ support\ op\ f\ s = GSPEC\ (\lambda GEN\%PVAR\%140{::}?'b{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%140\ (IN\ x\ s \wedge f\ x \neq neutral\ op)\ x)$

thm DEF_iterate:

$iterate = (\lambda(\_40839{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (\_40840{::}?'a{::}type \Rightarrow bool)\ \_40841{::}?'a{::}type \Rightarrow ?'b{::}type.\ if\ FINITE\ (support\ \_40839\ \_40841\ \_40840)\ then\ ITSET\ (\lambda x{::}?'a{::}type.\ \_40839\ (\_40841\ x))\ (support\ \_40839\ \_40841\ \_40840)\ (neutral\ \_40839)\ else\ neutral\ \_40839)$

thm iterate:

$\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ (s{::}?'b{::}type \Rightarrow bool)\ op{::}?'a{::}type \Rightarrow ?'a{::}type \Rightarrow ?'a{::}type.\ iterate\ op\ s\ f = (if\ FINITE\ (support\ op\ f\ s)\ then\ ITSET\ (\lambda x{::}?'b{::}type.\ op\ (f\ x))\ (support\ op\ f\ s)\ (neutral\ op)\ else\ neutral\ op)$

thm IN_SUPPORT:

$\forall (op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ (x{::}?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ IN\ x\ (support\ op\ f\ s) = (IN\ x\ s \wedge f\ x \neq neutral\ op)$

thm SUPPORT_SUPPORT:

$\forall (op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ support\ op\ f\ (support\ op\ f\ s) = support\ op\ f\ s$

thm SUPPORT_EMPTY:

$\forall (op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow f\ x = neutral\ op) = (support\ op\ f\ s = EMPTY)$

thm SUPPORT_SUBSET:

$\forall (op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ SUBSET\ (support\ op\ f\ s)\ s$

thm FINITE_SUPPORT:

$\forall (op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type)\ (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (support\ op\ f\ s)$

thm SUPPORT_CLAUSES:

$(\forall f{::}?'i{::}type \Rightarrow ?'h{::}type.\ support\ (?op{::}?'h{::}type \Rightarrow ?'h{::}type \Rightarrow ?'h{::}type)\ f\ EMPTY = EMPTY) \wedge (\forall (f{::}?'g{::}type \Rightarrow ?'h{::}type)\ (x{::}?'g{::}type)\ s{::}?'g{::}type$

$\Rightarrow$ *bool. support ?op f (INSERT x s) = (if f x = neutral ?op then support ?op f s else INSERT x (support ?op f s)))* $\wedge$ $(\forall (f::?'f::type \Rightarrow ?'h::type)\ (x::?'f::type)\ s::?'f::type \Rightarrow bool.$ *support ?op f (DELETE s x) = DELETE (support ?op f s) x)* $\wedge$ $(\forall (f::?'e::type \Rightarrow ?'h::type)\ (s::?'e::type \Rightarrow bool)\ t::?'e::type \Rightarrow bool.$ *support ?op f (HOL_Light_Import.UNION s t) = HOL_Light_Import.UNION (support ?op f s) (support ?op f t))* $\wedge$ $(\forall (f::?'d::type \Rightarrow ?'h::type)\ (s::?'d::type \Rightarrow bool)\ t::?'d::type \Rightarrow bool.$ *support ?op f (HOL_Light_Import.INTER s t) = HOL_Light_Import.INTER (support ?op f s) (support ?op f t))* $\wedge$ $(\forall (f::?'c::type \Rightarrow ?'h::type)\ (s::?'c::type \Rightarrow bool)\ t::?'c::type \Rightarrow bool.$ *support ?op f (DIFF s t) = DIFF (support ?op f s) (support ?op f t))* $\wedge$ $(\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow ?'h::type)\ s::?'b::type \Rightarrow bool.$ *support ?op g (IMAGE f s) = IMAGE f (support ?op (g* $\circ$ *f) s))*

thm SUPPORT_DELTA:

$\forall (op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ a::?'a::type.$ *support op* $(\lambda x::?'a::type.$ *if x = a then f x else neutral op) s = (if IN a s then support op f (INSERT a EMPTY) else EMPTY)*

thm FINITE_SUPPORT_DELTA:

$\forall (op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ (f::?'a::type \Rightarrow ?'b::type)\ a::?'a::type.$ *FINITE (support op* $(\lambda x::?'a::type.$ *if x = a then f x else neutral op)* $(?s::?'a::type \Rightarrow bool))$

thm ITERATE_SUPPORT:

$\forall (op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.$ *iterate op (support op f s) f = iterate op s f*

thm ITERATE_EXPAND_CASES:

$\forall (op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.$ *iterate op s f = (if FINITE (support op f s) then iterate op (support op f s) f else neutral op)*

thm SUPPORT_CLAUSES_conjunct6:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ s::?'c::type \Rightarrow bool.$ *support* $(?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)$ *g (IMAGE f s) = IMAGE f (support ?op (g* $\circ$ *f) s)*

thm SUPPORT_CLAUSES_conjunct5:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.$ *support* $(?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)$ *f (DIFF s t) = DIFF (support ?op f s) (support ?op f t)*

thm SUPPORT_CLAUSES_conjunct4:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.$ *support* $(?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)$ *f (HOL_Light_Import.INTER s t) = HOL_Light_Import.INTER (support ?op f s) (support ?op f t)*

thm SUPPORT_CLAUSES_conjunct3:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ support\ (?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ f\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.UNION\ (support\ ?op\ f\ s)\ (support\ ?op\ f\ t)$

thm SUPPORT_CLAUSES_conjunct2:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ support\ (?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ f\ (DELETE\ s\ x) = DELETE\ (support\ ?op\ f\ s)\ x$

thm SUPPORT_CLAUSES_conjunct1:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (x::?'b::type)\ s::?'b::type \Rightarrow bool.\ support\ (?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ f\ (INSERT\ x\ s) = (if\ f\ x = neutral\ ?op\ then\ support\ ?op\ f\ s\ else\ INSERT\ x\ (support\ ?op\ f\ s))$

thm SUPPORT_CLAUSES_conjunct0:

$\forall f::?'b::type \Rightarrow ?'a::type.\ support\ (?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type)\ f\ EMPTY = EMPTY$

thm ITERATE_CLAUSES_GEN:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall f::?'a::type \Rightarrow ?'b::type.\ iterate\ op\ EMPTY\ f = neutral\ op) \wedge (\forall (f::?'a::type \Rightarrow ?'b::type)\ (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ monoidal\ op \wedge FINITE\ (support\ op\ f\ s) \longrightarrow iterate\ op\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ iterate\ op\ s\ f\ else\ op\ (f\ x)\ (iterate\ op\ s\ f)))$

thm ITERATE_CLAUSES:

$\forall op::?'c::type \Rightarrow ?'c::type \Rightarrow ?'c::type.\ monoidal\ op \longrightarrow (\forall f::?'b::type \Rightarrow ?'c::type.\ iterate\ op\ EMPTY\ f = neutral\ op) \wedge (\forall (f::?'a::type \Rightarrow ?'c::type)\ (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow iterate\ op\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ iterate\ op\ s\ f\ else\ op\ (f\ x)\ (iterate\ op\ s\ f)))$

thm ITERATE_UNION:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge DISJOINT\ s\ t \longrightarrow iterate\ op\ (HOL\_Light\_Import.UNION\ s\ t)\ f = op\ (iterate\ op\ s\ f)\ (iterate\ op\ t\ f))$

thm ITERATE_UNION_GEN:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ (support\ op\ f\ s) \wedge FINITE\ (support\ op\ f\ t) \wedge DISJOINT\ (support\ op\ f\ s)\ (support\ op\ f\ t) \longrightarrow iterate\ op\ (HOL\_Light\_Import.UNION\ s\ t)\ f = op\ (iterate\ op\ s\ f)\ (iterate\ op\ t\ f))$

thm ITERATE_DIFF:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ t\ s \longrightarrow op\ (iterate\ op\ (DIFF\ s\ t)\ f)\ (iterate\ op\ t\ f) = iterate\ op\ s\ f)$

thm ITERATE_DIFF_GEN:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ (support\ op\ f\ s) \wedge SUBSET\ (support\ op\ f\ t)\ (support\ op\ f\ s) \longrightarrow op\ (iterate\ op\ (DIFF\ s\ t)\ f)\ (iterate\ op\ t\ f) = iterate\ op\ s\ f)$

thm ITERATE_INCL_EXCL:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (s{::}?'a{::}type \Rightarrow bool)\ (t{::}?'a{::}type \Rightarrow bool)\ f{::}?'a{::}type \Rightarrow ?'b{::}type.\ FINITE\ s \wedge FINITE\ t \longrightarrow op\ (iterate\ op\ s\ f)\ (iterate\ op\ t\ f) = op\ (iterate\ op\ (HOL\_Light\_Import.UNION\ s\ t)\ f)\ (iterate\ op\ (HOL\_Light\_Import.INTER\ s\ t)\ f))$

thm ITERATE_CLOSED:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, P{::}?'b{::}type \Rightarrow bool.\ P\ (neutral\ op) \wedge (\forall\, (x{::}?'b{::}type)\ y{::}?'b{::}type.\ P\ x \wedge P\ y \longrightarrow P\ (op\ x\ y)) \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ (\forall\, x{::}?'a{::}type.\ IN\ x\ s \wedge f\, x \neq neutral\ op \longrightarrow P\ (f\, x)) \longrightarrow P\ (iterate\ op\ s\ f)))$

thm ITERATE_RELATED:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, R{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow bool.\ R\ (neutral\ op)\ (neutral\ op) \wedge (\forall\, (x1{::}?'b{::}type)\ (y1{::}?'b{::}type)\ (x2{::}?'b{::}type)\ y2{::}?'b{::}type.\ R\ x1\ x2 \wedge R\ y1\ y2 \longrightarrow R\ (op\ x1\ y1)\ (op\ x2\ y2)) \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ (g{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge (\forall\, x{::}?'a{::}type.\ IN\ x\ s \longrightarrow R\ (f\ x)\ (g\ x)) \longrightarrow R\ (iterate\ op\ s\ f)\ (iterate\ op\ s\ g)))$

thm ITERATE_EQ_NEUTRAL:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ (\forall\, x{::}?'a{::}type.\ IN\ x\ s \longrightarrow f\, x = neutral\ op) \longrightarrow iterate\ op\ s\ f = neutral\ op)$

thm ITERATE_SING:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ x{::}?'a{::}type.\ iterate\ op\ (INSERT\ x\ EMPTY)\ f = f\, x)$

thm ITERATE_DELETE:

$\forall\, op{::}?'b{::}type \Rightarrow ?'b{::}type \Rightarrow ?'b{::}type.\ monoidal\ op \longrightarrow (\forall\, (f{::}?'a{::}type \Rightarrow ?'b{::}type)\ (s{::}?'a{::}type \Rightarrow bool)\ a{::}?'a{::}type.\ FINITE\ s \wedge IN\ a\ s \longrightarrow op\ (f\ a)\ (iterate\ op\ (DELETE\ s\ a)\ f) = iterate\ op\ s\ f)$

thm ITERATE_DELTA:

$\forall$ *op::?'b::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ *?'b::type. monoidal op* $\longrightarrow$ ($\forall$ (*f::?'a::type* $\Rightarrow$ *?'b::type*) (*a::?'a::type*) *s::?'a::type* $\Rightarrow$ *bool. iterate op s* (λ*x::?'a::type. if x* = *a then f x else neutral op*) = (*if IN a s then f a else neutral op*))

thm ITERATE_IMAGE:

$\forall$ *op::?'c::type* $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'c::type. monoidal op* $\longrightarrow$ ($\forall$ (*f::?'b::type* $\Rightarrow$ *?'a::type*) (*g::?'a::type* $\Rightarrow$ *?'c::type*) *s::?'b::type* $\Rightarrow$ *bool*. ($\forall$ (*x::?'b::type*) *y::?'b::type. IN x s* $\wedge$ *IN y s* $\wedge$ *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *iterate op* (*IMAGE f s*) *g* = *iterate op s* (*g* $\circ$ *f*))

thm ITERATE_BIJECTION:

$\forall$ *op::?'b::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ *?'b::type. monoidal op* $\longrightarrow$ ($\forall$ (*f::?'a::type* $\Rightarrow$ *?'b::type*) (*p::?'a::type* $\Rightarrow$ *?'a::type*) *s::?'a::type* $\Rightarrow$ *bool*. ($\forall$ *x::?'a::type. IN x s* $\longrightarrow$ *IN* (*p x*) *s*) $\wedge$ ($\forall$ *y::?'a::type. IN y s* $\longrightarrow$ ($\exists$!*x::?'a::type. IN x s* $\wedge$ *p x* = *y*)) $\longrightarrow$ *iterate op s f* = *iterate op s* (*f* $\circ$ *p*))

thm ITERATE_ITERATE_PRODUCT:

$\forall$ *op::?'c::type* $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'c::type. monoidal op* $\longrightarrow$ ($\forall$ (*s::?'b::type* $\Rightarrow$ *bool*) (*t::?'b::type* $\Rightarrow$ *?'a::type* $\Rightarrow$ *bool*) *x::?'b::type* $\Rightarrow$ *?'a::type* $\Rightarrow$ *?'c::type. FINITE s* $\wedge$ ($\forall$ *i::?'b::type. IN i s* $\longrightarrow$ *FINITE* (*t i*)) $\longrightarrow$ *iterate op s* (λ*i::?'b::type. iterate op* (*t i*) (*x i*)) = *iterate op* (*GSPEC* (λ*GEN%PVAR%144::?'b::type* $\times$ *?'a::type*. $\exists$ (*i::?'b::type*) *j::?'a::type. SETSPEC GEN%PVAR%144* (*IN i s* $\wedge$ *IN j* (*t i*)) (*i*, *j*))) (*GABS* (λ*f::?'b::type* $\times$ *?'a::type* $\Rightarrow$ *?'c::type*. $\forall$ (*i::?'b::type*) *j::?'a::type. GEQ* (*f* (*i*, *j*)) (*x i j*))))

thm ITERATE_EQ:

$\forall$ *op::?'b::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ *?'b::type. monoidal op* $\longrightarrow$ ($\forall$ (*f::?'a::type* $\Rightarrow$ *?'b::type*) (*g::?'a::type* $\Rightarrow$ *?'b::type*) *s::?'a::type* $\Rightarrow$ *bool*. ($\forall$ *x::?'a::type. IN x s* $\longrightarrow$ *f x* = *g x*) $\longrightarrow$ *iterate op s f* = *iterate op s g*)

thm ITERATE_EQ_GENERAL:

$\forall$ *op::?'c::type* $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'c::type. monoidal op* $\longrightarrow$ ($\forall$ (*s::?'b::type* $\Rightarrow$ *bool*) (*t::?'a::type* $\Rightarrow$ *bool*) (*f::?'b::type* $\Rightarrow$ *?'c::type*) (*g::?'a::type* $\Rightarrow$ *?'c::type*) *h::?'b::type* $\Rightarrow$ *?'a::type*. ($\forall$ *y::?'a::type. IN y t* $\longrightarrow$ ($\exists$!*x::?'b::type. IN x s* $\wedge$ *h x* = *y*)) $\wedge$ ($\forall$ *x::?'b::type. IN x s* $\longrightarrow$ *IN* (*h x*) *t* $\wedge$ *g* (*h x*) = *f x*) $\longrightarrow$ *iterate op s f* = *iterate op t g*)

thm ITERATE_EQ_GENERAL_INVERSES:

$\forall$ *op::?'c::type* $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'c::type. monoidal op* $\longrightarrow$ ($\forall$ (*s::?'b::type* $\Rightarrow$ *bool*) (*t::?'a::type* $\Rightarrow$ *bool*) (*f::?'b::type* $\Rightarrow$ *?'c::type*) (*g::?'a::type* $\Rightarrow$ *?'c::type*) (*h::?'b::type* $\Rightarrow$ *?'a::type*) *k::?'a::type* $\Rightarrow$ *?'b::type*. ($\forall$ *y::?'a::type. IN y t* $\longrightarrow$ *IN* (*k y*) *s* $\wedge$ *h* (*k y*) = *y*) $\wedge$ ($\forall$ *x::?'b::type. IN x s* $\longrightarrow$ *IN* (*h x*) *t* $\wedge$ *k* (*h x*) = *x* $\wedge$ *g* (*h x*) = *f x*) $\longrightarrow$ *iterate op s f* = *iterate op t g*)

thm ITERATE_INJECTION:

$\forall$ *op::?'b::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ *?'b::type. monoidal op* $\longrightarrow$ ($\forall$ (*f::?'a::type* $\Rightarrow$ *?'b::type*) (*p::?'a::type* $\Rightarrow$ *?'a::type*) *s::?'a::type* $\Rightarrow$ *bool. FINITE s* $\wedge$ ($\forall$ *x::?'a::type*.

$IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge p\ x = p\ y \longrightarrow x = y) \longrightarrow iterate\ op\ s\ (f \circ p) = iterate\ op\ s\ f)$

thm ITERATE_UNION_NONZERO:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge (\forall x::?'a::type.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow f\ x = neutral\ op) \longrightarrow iterate\ op\ (HOL\_Light\_Import.UNION\ s\ t)\ f = op\ (iterate\ op\ s\ f)\ (iterate\ op\ t\ f))$

thm ITERATE_OP:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (g::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow iterate\ op\ s\ (\lambda x::?'a::type.\ op\ (f\ x)\ (g\ x)) = op\ (iterate\ op\ s\ f)\ (iterate\ op\ s\ g))$

thm ITERATE_SUPERSET:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ SUBSET\ u\ v \wedge (\forall x::?'a::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = neutral\ op) \longrightarrow iterate\ op\ v\ f = iterate\ op\ u\ f)$

thm ITERATE_IMAGE_NONZERO:

$\forall op::?'c::type \Rightarrow ?'c::type \Rightarrow ?'c::type.\ monoidal\ op \longrightarrow (\forall (g::?'b::type \Rightarrow ?'c::type)\ (f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \wedge f\ x = f\ y \longrightarrow g\ (f\ x) = neutral\ op) \longrightarrow iterate\ op\ (IMAGE\ f\ s)\ g = iterate\ op\ s\ (g \circ f))$

thm ITERATE_CASES:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (s::?'a::type \Rightarrow bool)\ (P::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'b::type)\ g::?'a::type \Rightarrow ?'b::type.\ FINITE\ s \longrightarrow iterate\ op\ s\ (\lambda x::?'a::type.\ if\ P\ x\ then\ f\ x\ else\ g\ x) = op\ (iterate\ op\ (GSPEC\ (\lambda GEN\%PVAR\%147::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%147\ (IN\ x\ s \wedge P\ x)\ x))\ f)\ (iterate\ op\ (GSPEC\ (\lambda GEN\%PVAR\%148::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%148\ (IN\ x\ s \wedge \neg\ P\ x)\ x))\ g))$

thm ITERATE_OP_GEN:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (g::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ (support\ op\ f\ s) \wedge FINITE\ (support\ op\ g\ s) \longrightarrow iterate\ op\ s\ (\lambda x::?'a::type.\ op\ (f\ x)\ (g\ x)) = op\ (iterate\ op\ s\ f)\ (iterate\ op\ s\ g))$

thm ITERATE_CLAUSES_NUMSEG:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow (\forall m::nat.\ iterate\ op\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow ?'a::type) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ neutral\ op)) \wedge (\forall (m::nat)\ n::nat.\ iterate\ op\ (dotdot\ m\ (Suc\ n))\ ?f = (if\ m \leq Suc\ n\ then\ op\ (iterate\ op\ (dotdot\ m\ n)\ ?f)\ (?f\ (Suc\ n))\ else\ iterate\ op\ (dotdot\ m\ n)\ ?f))$

thm ITERATE_PAIR:

$\forall\, op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow (\forall\, (f::nat \Rightarrow ?'a::type)\ (m::nat)\ n::nat.\ iterate\ op\ (dotdot\ ((2::nat) * m)\ ((2::nat) * n + (1::nat)))\ f = iterate\ op\ (dotdot\ m\ n)\ (\lambda i::nat.\ op\ (f\ ((2::nat) * i))\ (f\ ((2::nat) * i + (1::nat)))))$

thm nsum:

$nsum = iterate\ op\ +$

thm NEUTRAL_ADD:

$neutral\ op\ + = (0::nat)$

thm NEUTRAL_MUL:

$neutral\ op\ * = (1::nat)$

thm MONOIDAL_ADD:

$monoidal\ op\ +$

thm MONOIDAL_MUL:

$monoidal\ op\ *$

thm NSUM_CLAUSES:

$(\forall\, f::?'b::type \Rightarrow nat.\ nsum\ EMPTY\ f = (0::nat)) \land (\forall\, (x::?'a::type)\ (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow nsum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ nsum\ s\ f\ else\ f\ x + nsum\ s\ f))$

thm NSUM_UNION:

$\forall\, (f::?'a::type \Rightarrow nat)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \land FINITE\ t \land DISJOINT\ s\ t \longrightarrow nsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = nsum\ s\ f + nsum\ t\ f$

thm NSUM_DIFF:

$\forall\, (f::?'a::type \Rightarrow nat)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \land SUBSET\ t\ s \longrightarrow nsum\ (DIFF\ s\ t)\ f = nsum\ s\ f - nsum\ t\ f$

thm NSUM_INCL_EXCL:

$\forall\, (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow nat.\ FINITE\ s \land FINITE\ t \longrightarrow nsum\ s\ f + nsum\ t\ f = nsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f + nsum\ (HOL\_Light\_Import.INTER\ s\ t)\ f$

thm NSUM_SUPPORT:

$\forall\, (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ nsum\ (support\ op\ +\ f\ s)\ f = nsum\ s\ f$

thm NSUM_ADD:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow nsum\ s\ (\lambda x::?'a::type.\ f\ x + g\ x) = nsum\ s\ f + nsum\ s\ g$

thm NSUM_ADD_GEN:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%153::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%153\ (IN\ x\ s \wedge f\ x \neq (0::nat))\ x)) \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%154::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%154\ (IN\ x\ s \wedge g\ x \neq (0::nat))\ x)) \longrightarrow nsum\ s\ (\lambda x::?'a::type.\ f\ x + g\ x) = nsum\ s\ f + nsum\ s\ g$

thm NSUM_EQ_0:

$\forall (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ s\ f = (0::nat)$

thm NSUM_0:

$\forall s::?'a::type \Rightarrow bool.\ nsum\ s\ (\lambda n::?'a::type.\ 0::nat) = (0::nat)$

thm NSUM_CLAUSES_conjunct1:

$\forall (x::?'a::type)\ (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow nsum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ nsum\ s\ f\ else\ f\ x + nsum\ s\ f)$

thm NSUM_CLAUSES_conjunct0:

$\forall f::?'a::type \Rightarrow nat.\ nsum\ EMPTY\ f = (0::nat)$

thm NSUM_LMUL:

$\forall (f::?'a::type \Rightarrow nat)\ (c::nat)\ s::?'a::type \Rightarrow bool.\ nsum\ s\ (\lambda x::?'a::type.\ c * f\ x) = c * nsum\ s\ f$

thm NSUM_RMUL:

$\forall (f::?'a::type \Rightarrow nat)\ (c::nat)\ s::?'a::type \Rightarrow bool.\ nsum\ s\ (\lambda x::?'a::type.\ f\ x * c) = nsum\ s\ f * c$

thm NSUM_LE:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \longrightarrow nsum\ s\ f \leq nsum\ s\ g$

thm NSUM_LT:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \wedge (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x < g\ x) \longrightarrow nsum\ s\ f < nsum\ s\ g$

thm NSUM_LT_ALL:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < g\ x) \longrightarrow nsum\ s\ f < nsum\ s\ g$

thm NSUM_EQ:

$\forall (f::?'a::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow nsum\ s\ f = nsum\ s\ g$

thm NSUM_CONST:

$\forall (c::nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow nsum\ s\ (\lambda n::?'a::type.\ c) = CARD\ s * c$

thm NSUM_POS_BOUND:

$\forall (f::?'a::type \Rightarrow nat)\ (b::nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge nsum\ s\ f \leq b \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b)$

thm NSUM_EQ_0_IFF:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow (nsum\ s\ (?f::?'a::type \Rightarrow nat) = (0::nat)) = (\forall x::?'a::type.\ IN\ x\ s \longrightarrow ?f\ x = (0::nat))$

thm NSUM_DELETE:

$\forall (f::?'a::type \Rightarrow nat)\ (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ FINITE\ s \wedge IN\ a\ s \longrightarrow f\ a + nsum\ (DELETE\ s\ a)\ f = nsum\ s\ f$

thm NSUM_SING:

$\forall (f::?'a::type \Rightarrow nat)\ x::?'a::type.\ nsum\ (INSERT\ x\ EMPTY)\ f = f\ x$

thm NSUM_DELTA:

$\forall (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ nsum\ s\ (\lambda x::?'a::type.\ if\ x = a\ then\ ?b::nat\ else\ (0::nat)) = (if\ IN\ a\ s\ then\ ?b\ else\ (0::nat))$

thm NSUM_SWAP:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow nat)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow nsum\ s\ (\lambda i::?'b::type.\ nsum\ t\ (f\ i)) = nsum\ t\ (\lambda j::?'a::type.\ nsum\ s\ (\lambda i::?'b::type.\ f\ i\ j))$

thm NSUM_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow nat)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow nsum\ (IMAGE\ f\ s)\ g = nsum\ s\ (g \circ f)$

thm NSUM_SUPERSET:

$\forall (f::?'a::type \Rightarrow nat)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ SUBSET\ u\ v \wedge (\forall x::?'a::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ v\ f = nsum\ u\ f$

thm NSUM_UNION_RZERO:

$\forall (f::?'a::type \Rightarrow nat)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ FINITE\ u \wedge (\forall x::?'a::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = nsum\ u\ f$

thm NSUM_UNION_LZERO:

$\forall (f::?'a::type \Rightarrow nat)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ FINITE\ v \wedge (\forall x::?'a::type.\ IN\ x\ u \wedge \neg\ IN\ x\ v \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = nsum\ v\ f$

thm NSUM_RESTRICT:

$\forall (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow nsum\ s\ (\lambda x::?'a::type.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::nat)) = nsum\ s\ f$

thm NSUM_BOUND:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ b::nat.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b) \longrightarrow nsum\ s\ f \leq CARD\ s * b$

thm NSUM_BOUND_GEN:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ b::nat.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b\ div\ CARD\ s) \longrightarrow nsum\ s\ f \leq b$

thm NSUM_BOUND_LT:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ b::nat.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b) \wedge (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x < b) \longrightarrow nsum\ s\ f < CARD\ s * b$

thm NSUM_BOUND_LT_ALL:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ b::nat.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < b) \longrightarrow nsum\ s\ f < CARD\ s * b$

thm NSUM_BOUND_LT_GEN:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ b::nat.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < b\ div\ CARD\ s) \longrightarrow nsum\ s\ f < b$

thm NSUM_UNION_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ FINITE\ u \wedge HOL\_Light\_Import.INTER\ s\ t = EMPTY \wedge HOL\_Light\_Import.UNION\ s\ t = u \longrightarrow nsum\ s\ (?f::?'a::type \Rightarrow nat) + nsum\ t\ ?f = nsum\ u\ ?f$

thm NSUM_EQ_SUPERSET:

$\forall (f::?'a::type \Rightarrow nat)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ t\ s \wedge (\forall x::?'a::type.\ IN\ x\ t \longrightarrow f\ x = (?g::?'a::type \Rightarrow nat)\ x) \wedge (\forall x::?'a::type.\ IN\ x\ s \wedge \neg\ IN\ x\ t \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ s\ f = nsum\ t\ ?g$

thm NSUM_RESTRICT_SET:

$\forall (P::?'a::type \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow nat.\ nsum\ (GSPEC\ (\lambda GEN\%PVAR\%155::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%155\ (IN\ x\ s \wedge P\ x)\ x))\ f = nsum\ s\ (\lambda x::?'a::type.\ if\ P\ x\ then\ f\ x\ else\ (0::nat))$

thm NSUM_NSUM_RESTRICT:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool) (f::?'b::type \Rightarrow ?'a::type \Rightarrow nat) (s::?'b::type \Rightarrow bool) t::?'a::type \Rightarrow bool. FINITE s \wedge FINITE t \longrightarrow nsum s (\lambda x::?'b::type. nsum (GSPEC (\lambda GEN\%PVAR\%156::?'a::type. \exists y::?'a::type. SETSPEC GEN\%PVAR\%156 (IN y t \wedge R x y) y)) (f x)) = nsum t (\lambda y::?'a::type. nsum (GSPEC (\lambda GEN\%PVAR\%157::?'b::type. \exists x::?'b::type. SETSPEC GEN\%PVAR\%157 (IN x s \wedge R x y) x)) (\lambda x::?'b::type. f x y))$

thm CARD_EQ_NSUM:

$\forall s::?'a::type \Rightarrow bool. FINITE s \longrightarrow CARD s = nsum s (\lambda x::?'a::type. 1::nat)$

thm NSUM_MULTICOUNT_GEN:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool) (s::?'b::type \Rightarrow bool) (t::?'a::type \Rightarrow bool) k::?'a::type \Rightarrow nat. FINITE s \wedge FINITE t \wedge (\forall j::?'a::type. IN j t \longrightarrow CARD (GSPEC (\lambda GEN\%PVAR\%159::?'b::type. \exists i::?'b::type. SETSPEC GEN\%PVAR\%159 (IN i s \wedge R i j) i)) = k j) \longrightarrow nsum s (\lambda i::?'b::type. CARD (GSPEC (\lambda GEN\%PVAR\%160::?'a::type. \exists j::?'a::type. SETSPEC GEN\%PVAR\%160 (IN j t \wedge R i j) j))) = nsum t k$

thm NSUM_MULTICOUNT:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool) (s::?'b::type \Rightarrow bool) (t::?'a::type \Rightarrow bool) k::nat. FINITE s \wedge FINITE t \wedge (\forall j::?'a::type. IN j t \longrightarrow CARD (GSPEC (\lambda GEN\%PVAR\%161::?'b::type. \exists i::?'b::type. SETSPEC GEN\%PVAR\%161 (IN i s \wedge R i j) i)) = k) \longrightarrow nsum s (\lambda i::?'b::type. CARD (GSPEC (\lambda GEN\%PVAR\%162::?'a::type. \exists j::?'a::type. SETSPEC GEN\%PVAR\%162 (IN j t \wedge R i j) j))) = k * CARD t$

thm NSUM_IMAGE_GEN:

$\forall (f::?'b::type \Rightarrow ?'a::type) (g::?'b::type \Rightarrow nat) s::?'b::type \Rightarrow bool. FINITE s \longrightarrow nsum s g = nsum (IMAGE f s) (\lambda y::?'a::type. nsum (GSPEC (\lambda GEN\%PVAR\%165::?'b::type. \exists x::?'b::type. SETSPEC GEN\%PVAR\%165 (IN x s \wedge f x = y) x)) g)$

thm NSUM_GROUP:

$\forall (f::?'b::type \Rightarrow ?'a::type) (g::?'b::type \Rightarrow nat) (s::?'b::type \Rightarrow bool) t::?'a::type \Rightarrow bool. FINITE s \wedge SUBSET (IMAGE f s) t \longrightarrow nsum t (\lambda y::?'a::type. nsum (GSPEC (\lambda GEN\%PVAR\%166::?'b::type. \exists x::?'b::type. SETSPEC GEN\%PVAR\%166 (IN x s \wedge f x = y) x)) g) = nsum s g$

thm NSUM_SUBSET:

$\forall (u::?'a::type \Rightarrow bool) (v::?'a::type \Rightarrow bool) f::?'a::type \Rightarrow nat. FINITE u \wedge FINITE v \wedge (\forall x::?'a::type. IN x (DIFF u v) \longrightarrow f x = (0::nat)) \longrightarrow nsum u f \leq nsum v f$

thm NSUM_SUBSET_SIMPLE:

$\forall (u::?'a::type \Rightarrow bool) (v::?'a::type \Rightarrow bool) f::?'a::type \Rightarrow nat. FINITE v \wedge SUBSET u v \longrightarrow nsum u f \leq nsum v f$

thm NSUM_IMAGE_NONZERO:

$\forall (d::?'b::type \Rightarrow nat)\ (i::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \wedge i\ x = i\ y \longrightarrow d\ (i\ x) = (0::nat)) \longrightarrow nsum\ (IMAGE\ i\ s)\ d = nsum\ s\ (d \circ i)$

thm NSUM_BIJECTION:

$\forall (f::?'a::type \Rightarrow nat)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall y::?'a::type.\ IN\ y\ s \longrightarrow (\exists !x::?'a::type.\ IN\ x\ s \wedge p\ x = y)) \longrightarrow nsum\ s\ f = nsum\ s\ (f \circ p)$

thm NSUM_NSUM_PRODUCT:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'b::type \Rightarrow ?'a::type \Rightarrow nat.\ FINITE\ s \wedge (\forall i::?'b::type.\ IN\ i\ s \longrightarrow FINITE\ (t\ i)) \longrightarrow nsum\ s\ (\lambda i::?'b::type.\ nsum\ (t\ i)\ (x\ i)) = nsum\ (GSPEC\ (\lambda GEN\%PVAR\%167::?'b::type \times ?'a::type.\ \exists (i::?'b::type)\ j::?'a::type.\ SETSPEC\ GEN\%PVAR\%167\ (IN\ i\ s \wedge IN\ j\ (t\ i))\ (i, j)))\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow nat.\ \forall (i::?'b::type)\ j::?'a::type.\ GEQ\ (f\ (i, j))\ (x\ i\ j)))$

thm NSUM_EQ_GENERAL:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ h::?'b::type \Rightarrow ?'a::type.\ (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists !x::?'b::type.\ IN\ x\ s \wedge h\ x = y)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge g\ (h\ x) = f\ x) \longrightarrow nsum\ s\ f = nsum\ t\ g$

thm NSUM_EQ_GENERAL_INVERSES:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow nat)\ (g::?'a::type \Rightarrow nat)\ (h::?'b::type \Rightarrow ?'a::type)\ k::?'a::type \Rightarrow ?'b::type.\ (\forall y::?'a::type.\ IN\ y\ t \longrightarrow IN\ (k\ y)\ s \wedge h\ (k\ y) = y) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge k\ (h\ x) = x \wedge g\ (h\ x) = f\ x) \longrightarrow nsum\ s\ f = nsum\ t\ g$

thm NSUM_INJECTION:

$\forall (f::?'a::type \Rightarrow nat)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge p\ x = p\ y \longrightarrow x = y) \longrightarrow nsum\ s\ (f \circ p) = nsum\ s\ f$

thm NSUM_UNION_NONZERO:

$\forall (f::?'a::type \Rightarrow nat)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge (\forall x::?'a::type.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = nsum\ s\ f + nsum\ t\ f$

thm NSUM_UNIONS_NONZERO:

$\forall (f::?'a::type \Rightarrow nat)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow FINITE\ t) \wedge (\forall (t1::?'a::type \Rightarrow bool)\ (t2::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ t1\ s \wedge IN\ t2\ s \wedge t1 \neq t2 \wedge IN\ x\ t1 \wedge IN\ x\ t2 \longrightarrow f\ x = (0::nat)) \longrightarrow nsum\ (UNIONS\ s)\ f = nsum\ s\ (\lambda t::?'a::type \Rightarrow bool.\ nsum\ t\ f)$

thm NSUM_CASES:

$\forall (s::?'a::type \Rightarrow bool)\ (P::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ g::?'a::type \Rightarrow nat.\ FINITE\ s \longrightarrow nsum\ s\ (\lambda x::?'a::type.\ if\ P\ x\ then\ f\ x\ else\ g\ x) = nsum\ (GSPEC\ (\lambda GEN\%PVAR\%168::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%168\ (IN\ x\ s \wedge P\ x)\ x))\ f + nsum\ (GSPEC\ (\lambda GEN\%PVAR\%169::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%169\ (IN\ x\ s \wedge \neg\ P\ x)\ x))\ g$

thm NSUM_CLOSED:

$\forall (P::nat \Rightarrow bool)\ (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ P\ (0::nat) \wedge (\forall (x::nat)\ y::nat.\ P\ x \wedge P\ y \longrightarrow P\ (x + y)) \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow P\ (f\ a)) \longrightarrow P\ (nsum\ s\ f)$

thm NSUM_ADD_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (g::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ nsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ i + g\ i) = nsum\ (dotdot\ m\ n)\ f + nsum\ (dotdot\ m\ n)\ g$

thm NSUM_LE_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (g::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i \leq g\ i) \longrightarrow nsum\ (dotdot\ m\ n)\ f \leq nsum\ (dotdot\ m\ n)\ g$

thm NSUM_EQ_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (g::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i = g\ i) \longrightarrow nsum\ (dotdot\ m\ n)\ f = nsum\ (dotdot\ m\ n)\ g$

thm NSUM_CONST_NUMSEG:

$\forall (c::nat)\ (m::nat)\ n::nat.\ nsum\ (dotdot\ m\ n)\ (\lambda n::nat.\ c) = (n + (1::nat) - m) * c$

thm NSUM_EQ_0_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i = (0::nat)) \longrightarrow nsum\ (dotdot\ m\ n)\ f = (0::nat)$

thm NSUM_EQ_0_IFF_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ (nsum\ (dotdot\ m\ n)\ f = (0::nat)) = (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i = (0::nat))$

thm NSUM_TRIV_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ n < m \longrightarrow nsum\ (dotdot\ m\ n)\ f = (0::nat)$

thm NSUM_SING_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ n::nat.\ nsum\ (dotdot\ n\ n)\ f = f\ n$

thm NSUM_CLAUSES_NUMSEG:

$(\forall m::nat.\ nsum\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow nat) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ (0::nat))) \wedge (\forall (m::nat)\ n::nat.\ nsum\ (dotdot\ m\ (Suc\ n))\ ?f = (if\ m \leq Suc\ n\ then\ nsum\ (dotdot\ m\ n)\ ?f + ?f\ (Suc\ n)\ else\ nsum\ (dotdot\ m\ n)\ ?f))$

thm NSUM_SWAP_NUMSEG:

$\forall (a::nat)\ (b::nat)\ (c::nat)\ (d::nat)\ f::nat \Rightarrow nat \Rightarrow nat.\ nsum\ (dotdot\ a\ b)\ (\lambda i::nat.\ nsum\ (dotdot\ c\ d)\ (f\ i)) = nsum\ (dotdot\ c\ d)\ (\lambda j::nat.\ nsum\ (dotdot\ a\ b)\ (\lambda i::nat.\ f\ i\ j))$

thm NSUM_ADD_SPLIT:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ (n::nat)\ p::nat.\ m \leq n + (1::nat) \longrightarrow nsum\ (dotdot\ m\ (n + p))\ f = nsum\ (dotdot\ m\ n)\ f + nsum\ (dotdot\ (n + (1::nat))\ (n + p))\ f$

thm NSUM_OFFSET:

$\forall (p::nat)\ (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ nsum\ (dotdot\ (m + p)\ (n + p))\ f = nsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ (i + p))$

thm NSUM_OFFSET_0:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow nsum\ (dotdot\ m\ n)\ f = nsum\ (dotdot\ (0::nat)\ (n - m))\ (\lambda i::nat.\ f\ (i + m))$

thm NSUM_CLAUSES_LEFT:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow nsum\ (dotdot\ m\ n)\ f = f\ m + nsum\ (dotdot\ (m + (1::nat))\ n)\ f$

thm NSUM_CLAUSES_NUMSEG_conjunct1:

$\forall (m::nat)\ n::nat.\ nsum\ (dotdot\ m\ (Suc\ n))\ (?f::nat \Rightarrow nat) = (if\ m \leq Suc\ n\ then\ nsum\ (dotdot\ m\ n)\ ?f + ?f\ (Suc\ n)\ else\ nsum\ (dotdot\ m\ n)\ ?f)$

thm NSUM_CLAUSES_NUMSEG_conjunct0:

$\forall m::nat.\ nsum\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow nat) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ (0::nat))$

thm NSUM_CLAUSES_RIGHT:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ (0::nat) < n \land m \leq n \longrightarrow nsum\ (dotdot\ m\ n)\ f = nsum\ (dotdot\ m\ (n - (1::nat)))\ f + f\ n$

thm NSUM_PAIR:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ nsum\ (dotdot\ ((2::nat) * m)\ ((2::nat) * n + (1::nat)))\ f = nsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ ((2::nat) * i) + f\ ((2::nat) * i + (1::nat)))$

thm MOD_NSUM_MOD:

$\forall (f::?'a::type \Rightarrow nat)\ (n::nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \land n \neq (0::nat) \longrightarrow nsum\ s\ f\ mod\ n = nsum\ s\ (\lambda i::?'a::type.\ f\ i\ mod\ n)\ mod\ n$

thm MOD_NSUM_MOD_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (a::nat)\ (b::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow nsum\ (dotdot\ a\ b)\ f\ mod\ n = nsum\ (dotdot\ a\ b)\ (\lambda i::nat.\ f\ i\ mod\ n)\ mod\ n$

thm INTER_UNIONS_conjunct1:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ t\ (UNIONS\ s) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%9::?'a::type \Rightarrow bool.\ \exists x::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%9\ (IN\ x\ s)\ (HOL\_Light\_Import.INTER\ t\ x)))$

thm INTER_UNIONS_conjunct0:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (UNIONS\ s)\ t = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%8::?'a::type \Rightarrow bool.\ \exists x::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%8\ (IN\ x\ s)\ (HOL\_Light\_Import.INTER\ x\ t)))$

thm CARD_UNIONS:

$\forall s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow FINITE\ t) \wedge (\forall (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ IN\ t\ s \wedge IN\ u\ s \wedge t \neq u \longrightarrow HOL\_Light\_Import.INTER\ t\ u = EMPTY) \longrightarrow CARD\ (UNIONS\ s) = nsum\ s\ CARD$

thm sum:

$sum = iterate\ op\ +$

thm NEUTRAL_REAL_ADD:

$neutral\ op\ + = (0::real)$

thm NEUTRAL_REAL_MUL:

$neutral\ op\ * = (1::real)$

thm MONOIDAL_REAL_ADD:

$monoidal\ op\ +$

thm MONOIDAL_REAL_MUL:

$monoidal\ op\ *$

thm SUM_CLAUSES:

$(\forall f::?'b::type \Rightarrow real.\ sum\ EMPTY\ f = (0::real)) \wedge (\forall (x::?'a::type)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ sum\ s\ f\ else\ f\ x + sum\ s\ f))$

thm SUM_UNION:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge DISJOINT\ s\ t \longrightarrow sum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = sum\ s\ f + sum\ t\ f$

thm SUM_DIFF:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ t\ s \longrightarrow sum\ (DIFF\ s\ t)\ f = sum\ s\ f - sum\ t\ f$

thm SUM_INCL_EXCL:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ FINITE\ s \wedge FINITE\ t \longrightarrow sum\ s\ f + sum\ t\ f = sum\ (HOL\_Light\_Import.UNION\ s\ t)\ f + sum\ (HOL\_Light\_Import.INTER\ s\ t)\ f$

thm SUM_SUPPORT:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ sum\ (support\ op\ +\ f\ s)\ f = sum\ s\ f$

thm SUM_ADD:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ (\lambda x::?'a::type.\ f\ x + g\ x) = sum\ s\ f + sum\ s\ g$

thm SUM_ADD_GEN:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%172::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%172\ (IN\ x\ s \wedge f\ x \neq (0::real))\ x)) \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%173::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%173\ (IN\ x\ s \wedge g\ x \neq (0::real))\ x)) \longrightarrow sum\ s\ (\lambda x::?'a::type.\ f\ x + g\ x) = sum\ s\ f + sum\ s\ g$

thm SUM_EQ_0:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = (0::real)) \longrightarrow sum\ s\ f = (0::real)$

thm SUM_0:

$\forall s::?'a::type \Rightarrow bool.\ sum\ s\ (\lambda n::?'a::type.\ 0::real) = (0::real)$

thm SUM_CLAUSES_conjunct1:

$\forall (x::?'a::type)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ sum\ s\ f\ else\ f\ x + sum\ s\ f)$

thm SUM_CLAUSES_conjunct0:

$\forall f::?'a::type \Rightarrow real.\ sum\ EMPTY\ f = (0::real)$

thm SUM_LMUL:

$\forall (f::?'a::type \Rightarrow real)\ (c::real)\ s::?'a::type \Rightarrow bool.\ sum\ s\ (\lambda x::?'a::type.\ c * f\ x) = c * sum\ s\ f$

thm SUM_RMUL:

$\forall (f::?'a::type \Rightarrow real)\ (c::real)\ s::?'a::type \Rightarrow bool.\ sum\ s\ (\lambda x::?'a::type.\ f\ x * c) = sum\ s\ f * c$

thm SUM_NEG:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ sum\ s\ (\lambda x::?'a::type.\ -\ f\ x) = -\ sum\ s\ f$

thm SUM_SUB:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ (\lambda x::?'a::type.\ f\ x\ -\ g\ x) = sum\ s\ f\ -\ sum\ s\ g$

thm SUM_LE:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s\ \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \longrightarrow sum\ s\ f \leq sum\ s\ g$

thm SUM_LT:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s\ \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \wedge (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x < g\ x) \longrightarrow sum\ s\ f < sum\ s\ g$

thm SUM_LT_ALL:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s\ \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < g\ x) \longrightarrow sum\ s\ f < sum\ s\ g$

thm SUM_EQ:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow sum\ s\ f = sum\ s\ g$

thm SUM_ABS:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow |sum\ s\ f| \leq sum\ s\ (\lambda x::?'a::type.\ |f\ x|)$

thm SUM_ABS_LE:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s\ \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow |f\ x| \leq g\ x) \longrightarrow |sum\ s\ f| \leq sum\ s\ g$

thm SUM_CONST:

$\forall (c::real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ (\lambda n::?'a::type.\ c) = real\_of\_nat\ (CARD\ s) * c$

thm SUM_POS_LE:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow (0::real) \leq sum\ s\ f$

thm SUM_POS_BOUND:

$\forall (f::?'a::type \Rightarrow real)\ (b::real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \wedge sum\ s\ f \leq b \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b)$

thm SUM_POS_EQ_0:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \wedge sum\ s\ f = (0::real) \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = (0::real))$

thm SUM_ZERO_EXISTS:

$\forall (u::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge sum\ s\ u = (0::real) \longrightarrow (\forall i::?'a::type.\ IN\ i\ s \longrightarrow u\ i = (0::real)) \vee (\exists (j::?'a::type)\ k::?'a::type.\ IN\ j\ s \wedge u\ j < (0::real) \wedge IN\ k\ s \wedge (0::real) < u\ k)$

thm SUM_DELETE:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ FINITE\ s \wedge IN\ a\ s \longrightarrow sum\ (DELETE\ s\ a)\ f = sum\ s\ f - f\ a$

thm SUM_DELETE_CASES:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ FINITE\ s \longrightarrow sum\ (DELETE\ s\ a)\ f = (if\ IN\ a\ s\ then\ sum\ s\ f - f\ a\ else\ sum\ s\ f)$

thm SUM_SING:

$\forall (f::?'a::type \Rightarrow real)\ x::?'a::type.\ sum\ (INSERT\ x\ EMPTY)\ f = f\ x$

thm SUM_DELTA:

$\forall (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ sum\ s\ (\lambda x::?'a::type.\ if\ x = a\ then\ ?b::real\ else\ (0::real)) = (if\ IN\ a\ s\ then\ ?b\ else\ (0::real))$

thm SUM_SWAP:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow sum\ s\ (\lambda i::?'b::type.\ sum\ t\ (f\ i)) = sum\ t\ (\lambda j::?'a::type.\ sum\ s\ (\lambda i::?'b::type.\ f\ i\ j))$

thm SUM_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow real)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow sum\ (IMAGE\ f\ s)\ g = sum\ s\ (g \circ f)$

thm SUM_SUPERSET:

$\forall (f::?'a::type \Rightarrow real)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ SUBSET\ u\ v \wedge (\forall x::?'a::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = (0::real)) \longrightarrow sum\ v\ f = sum\ u\ f$

thm SUM_UNION_RZERO:

$\forall (f::?'a::type \Rightarrow real)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ FINITE\ u \wedge (\forall x::?'a::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = (0::real)) \longrightarrow sum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = sum\ u\ f$

thm SUM_UNION_LZERO:

$\forall (f::?'a::type \Rightarrow real)\ (u::?'a::type \Rightarrow bool)\ v::?'a::type \Rightarrow bool.\ FINITE\ v \wedge (\forall x::?'a::type.\ IN\ x\ u \wedge \neg\ IN\ x\ v \longrightarrow f\ x = (0::real)) \longrightarrow sum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = sum\ v\ f$

thm SUM_RESTRICT:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ (\lambda x::?'a::type.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::real)) = sum\ s\ f$

thm SUM_BOUND:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b) \longrightarrow sum\ s\ f \leq real\_of\_nat\ (CARD\ s) * b$

thm SUM_BOUND_GEN:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b\ /\ real\_of\_nat\ (CARD\ s)) \longrightarrow sum\ s\ f \leq b$

thm SUM_ABS_BOUND:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow |f\ x| \leq b) \longrightarrow |sum\ s\ f| \leq real\_of\_nat\ (CARD\ s) * b$

thm SUM_BOUND_LT:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b) \wedge (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x < b) \longrightarrow sum\ s\ f < real\_of\_nat\ (CARD\ s) * b$

thm SUM_BOUND_LT_ALL:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < b) \longrightarrow sum\ s\ f < real\_of\_nat\ (CARD\ s) * b$

thm SUM_BOUND_LT_GEN:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ b::real.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x < b\ /\ real\_of\_nat\ (CARD\ s)) \longrightarrow sum\ s\ f < b$

thm SUM_UNION_EQ:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ FINITE\ u \wedge HOL\_Light\_Import.INTER\ s\ t = EMPTY \wedge HOL\_Light\_Import.UNION\ s\ t = u \longrightarrow sum\ s\ (?f::?'a::type \Rightarrow real) + sum\ t\ ?f = sum\ u\ ?f$

thm SUM_EQ_SUPERSET:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge SUBSET\ t\ s \wedge (\forall x::?'a::type.\ IN\ x\ t \longrightarrow f\ x = (?g::?'a::type \Rightarrow real)\ x) \wedge (\forall x::?'a::type.\ IN\ x\ s \wedge \neg\ IN\ x\ t \longrightarrow f\ x = (0::real)) \longrightarrow sum\ s\ f = sum\ t\ ?g$

thm SUM_RESTRICT_SET:

$\forall (P::?'a::type \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%174::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%174\ (IN\ x\ s \wedge P\ x)\ x))\ f = sum\ s\ (\lambda x::?'a::type.\ if\ P\ x\ then\ f\ x\ else\ (0::real))$

thm SUM_SUM_RESTRICT:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow sum\ s\ (\lambda x::?'b::type.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%175::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%175\ (IN\ y\ t \wedge R\ x\ y)\ y))\ (f\ x)) = sum\ t\ (\lambda y::?'a::type.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%176::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%176\ (IN\ x\ s \wedge R\ x\ y)\ x))\ (\lambda x::?'b::type.\ f\ x\ y))$

thm CARD_EQ_SUM:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow real\_of\_nat\ (CARD\ s) = sum\ s\ (\lambda x::?'a::type.\ 1::real)$

thm SUM_MULTICOUNT_GEN:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ k::?'a::type \Rightarrow nat.\ FINITE\ s \wedge FINITE\ t \wedge (\forall j::?'a::type.\ IN\ j\ t \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%178::?'b::type.\ \exists i::?'b::type.\ SETSPEC\ GEN\%PVAR\%178\ (IN\ i\ s \wedge R\ i\ j)\ i)) = k\ j) \longrightarrow sum\ s\ (\lambda i::?'b::type.\ real\_of\_nat\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%179::?'a::type.\ \exists j::?'a::type.\ SETSPEC\ GEN\%PVAR\%179\ (IN\ j\ t \wedge R\ i\ j)\ j)))) = sum\ t\ (\lambda i::?'a::type.\ real\_of\_nat\ (k\ i))$

thm SUM_MULTICOUNT:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ k::nat.\ FINITE\ s \wedge FINITE\ t \wedge (\forall j::?'a::type.\ IN\ j\ t \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%180::?'b::type.\ \exists i::?'b::type.\ SETSPEC\ GEN\%PVAR\%180\ (IN\ i\ s \wedge R\ i\ j)\ i)) = k) \longrightarrow sum\ s\ (\lambda i::?'b::type.\ real\_of\_nat\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%181::?'a::type.\ \exists j::?'a::type.\ SETSPEC\ GEN\%PVAR\%181\ (IN\ j\ t \wedge R\ i\ j)\ j)))) = real\_of\_nat\ (k * CARD\ t)$

thm SUM_IMAGE_GEN:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow real)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ g = sum\ (IMAGE\ f\ s)\ (\lambda y::?'a::type.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%184::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%184\ (IN\ x\ s \wedge f\ x = y)\ x))\ g)$

thm SUM_GROUP:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow real)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ (IMAGE\ f\ s)\ t \longrightarrow sum\ t\ (\lambda y::?'a::type.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%185::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%185\ (IN\ x\ s \wedge f\ x = y)\ x))\ g) = sum\ s\ g$

thm REAL_OF_NUM_SUM:

$\forall (f::?'a::type \Rightarrow nat)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow real\_of\_nat\ (nsum\ s\ f) = sum\ s\ (\lambda x::?'a::type.\ real\_of\_nat\ (f\ x))$

thm SUM_SUBSET:

$\forall (u::?'a::type \Rightarrow bool)\ (v::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ FINITE\ u \wedge FINITE\ v \wedge (\forall x::?'a::type.\ IN\ x\ (DIFF\ u\ v) \longrightarrow f\ x \leq (0::real)) \wedge (\forall x::?'a::type.\ IN\ x\ (DIFF\ v\ u) \longrightarrow (0::real) \leq f\ x) \longrightarrow sum\ u\ f \leq sum\ v\ f$

thm SUM_SUBSET_SIMPLE:

$\forall (u::?'a::type \Rightarrow bool)\ (v::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ FINITE\ v \wedge SUBSET\ u\ v \wedge (\forall x::?'a::type.\ IN\ x\ (DIFF\ v\ u) \longrightarrow (0::real) \leq f\ x) \longrightarrow sum\ u\ f \leq sum\ v\ f$

thm SUM_IMAGE_NONZERO:

$\forall (d::?'b::type \Rightarrow real)\ (i::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \wedge i\ x = i\ y \longrightarrow d\ (i\ x) = (0::real)) \longrightarrow sum\ (IMAGE\ i\ s)\ d = sum\ s\ (d \circ i)$

thm SUM_BIJECTION:

$\forall (f::?'a::type \Rightarrow real)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall y::?'a::type.\ IN\ y\ s \longrightarrow (\exists!x::?'a::type.\ IN\ x\ s \wedge p\ x = y)) \longrightarrow sum\ s\ f = sum\ s\ (f \circ p)$

thm SUM_SUM_PRODUCT:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'b::type \Rightarrow ?'a::type \Rightarrow real.\ FINITE\ s \wedge (\forall i::?'b::type.\ IN\ i\ s \longrightarrow FINITE\ (t\ i)) \longrightarrow sum\ s\ (\lambda i::?'b::type.\ sum\ (t\ i)\ (x\ i)) = sum\ (GSPEC\ (\lambda GEN\%PVAR\%186::?'b::type \times ?'a::type.\ \exists (i::?'b::type)\ j::?'a::type.\ SETSPEC\ GEN\%PVAR\%186\ (IN\ i\ s \wedge IN\ j\ (t\ i))\ (i, j)))\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow real.\ \forall (i::?'b::type)\ j::?'a::type.\ GEQ\ (f\ (i, j))\ (x\ i\ j)))$

thm SUM_EQ_GENERAL:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ h::?'b::type \Rightarrow ?'a::type.\ (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (\exists!x::?'b::type.\ IN\ x\ s \wedge h\ x = y)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge g\ (h\ x) = f\ x) \longrightarrow sum\ s\ f = sum\ t\ g$

thm SUM_EQ_GENERAL_INVERSES:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (h::?'b::type \Rightarrow ?'a::type)\ k::?'a::type \Rightarrow ?'b::type.\ (\forall y::?'a::type.\ IN\ y\ t \longrightarrow IN\ (k\ y)\ s \wedge h\ (k\ y) = y) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge k\ (h\ x) = x \wedge g\ (h\ x) = f\ x) \longrightarrow sum\ s\ f = sum\ t\ g$

thm SUM_INJECTION:

$\forall (f::?'a::type \Rightarrow real)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge p\ x = p\ y \longrightarrow x = y) \longrightarrow sum\ s\ (f \circ p) = sum\ s\ f$

thm SUM_UNION_NONZERO:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge (\forall x::?'a::type.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow f\ x = (0::real)) \longrightarrow sum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = sum\ s\ f + sum\ t\ f$

thm SUM_UNIONS_NONZERO:

$\forall (f::?'a::type \Rightarrow real)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow FINITE\ t) \wedge (\forall (t1::?'a::type \Rightarrow bool)\ (t2::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ t1\ s \wedge IN\ t2\ s \wedge t1 \neq t2 \wedge IN\ x\ t1 \wedge IN\ x\ t2 \longrightarrow f\ x = (0::real)) \longrightarrow sum\ (UNIONS\ s)\ f = sum\ s\ (\lambda t::?'a::type \Rightarrow bool.\ sum\ t\ f)$

thm SUM_CASES:

$\forall (s::?'a::type \Rightarrow bool)\ (P::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ FINITE\ s \longrightarrow sum\ s\ (\lambda x::?'a::type.\ if\ P\ x\ then\ f\ x\ else\ g\ x) = sum\ (GSPEC\ (\lambda GEN\%PVAR\%187::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%187\ (IN\ x\ s \wedge P\ x)\ x))\ f + sum\ (GSPEC\ (\lambda GEN\%PVAR\%188::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%188\ (IN\ x\ s \wedge \neg\ P\ x)\ x))\ g$

thm SUM_CASES_1:

$\forall (s::?'a::type \Rightarrow bool)\ a::?'a::type.\ FINITE\ s \wedge IN\ a\ s \longrightarrow sum\ s\ (\lambda x::?'a::type.\ if\ x = a\ then\ ?y::real\ else\ (?f::?'a::type \Rightarrow real)\ x) = sum\ s\ ?f + (?y - ?f\ a)$

thm SUM_LE_INCLUDED:

$\forall (f::?'b::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ i::?'a::type \Rightarrow ?'b::type.\ FINITE\ s \wedge FINITE\ t \wedge (\forall y::?'a::type.\ IN\ y\ t \longrightarrow (0::real) \leq g\ y) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow (\exists y::?'a::type.\ IN\ y\ t \wedge i\ y = x \wedge f\ x \leq g\ y)) \longrightarrow sum\ s\ f \leq sum\ t\ g$

thm SUM_IMAGE_LE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow real)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow (0::real) \leq g\ (f\ x)) \longrightarrow sum\ (IMAGE\ f\ s)\ g \leq sum\ s\ (g \circ f)$

thm SUM_CLOSED:

$\forall (P::real \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ P\ (0::real) \wedge (\forall (x::real)\ y::real.\ P\ x \wedge P\ y \longrightarrow P\ (x + y)) \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow P\ (f\ a)) \longrightarrow P\ (sum\ s\ f)$

thm SUM_ADD_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ i + g\ i) = sum\ (dotdot\ m\ n)\ f + sum\ (dotdot\ m\ n)\ g$

thm SUM_SUB_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ i - g\ i) = sum\ (dotdot\ m\ n)\ f - sum\ (dotdot\ m\ n)\ g$

thm SUM_LE_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i \leq g\ i) \longrightarrow sum\ (dotdot\ m\ n)\ f \leq sum\ (dotdot\ m\ n)\ g$

thm SUM_EQ_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i = g\ i) \longrightarrow sum\ (dotdot\ m\ n)\ f = sum\ (dotdot\ m\ n)\ g$

thm SUM_ABS_NUMSEG:

$\forall (f::nat \Rightarrow real)$ $(m::nat)$ $n::nat.$ $|sum\ (dotdot\ m\ n)\ f| \le sum\ (dotdot\ m\ n)$ $(\lambda i::nat.\ |f\ i|)$

thm SUM_CONST_NUMSEG:

$\forall (c::real)$ $(m::nat)$ $n::nat.$ $sum\ (dotdot\ m\ n)\ (\lambda n::nat.\ c) = real\_of\_nat\ (n +$ $(1::nat) - m) * c$

thm SUM_EQ_0_NUMSEG:

$\forall (f::nat \Rightarrow real)$ $(m::nat)$ $n::nat.$ $(\forall i::nat.\ m \le i \wedge i \le n \longrightarrow f\ i = (0::real))$ $\longrightarrow sum\ (dotdot\ m\ n)\ f = (0::real)$

thm SUM_TRIV_NUMSEG:

$\forall (f::nat \Rightarrow real)$ $(m::nat)$ $n::nat.$ $n < m \longrightarrow sum\ (dotdot\ m\ n)\ f = (0::real)$

thm SUM_POS_LE_NUMSEG:

$\forall (m::nat)$ $(n::nat)$ $f::nat \Rightarrow real.$ $(\forall p::nat.\ m \le p \wedge p \le n \longrightarrow (0::real) \le f$ $p) \longrightarrow (0::real) \le sum\ (dotdot\ m\ n)\ f$

thm SUM_POS_EQ_0_NUMSEG:

$\forall (f::nat \Rightarrow real)$ $(m::nat)$ $n::nat.$ $(\forall p::nat.\ m \le p \wedge p \le n \longrightarrow (0::real) \le f$ $p) \wedge sum\ (dotdot\ m\ n)\ f = (0::real) \longrightarrow (\forall p::nat.\ m \le p \wedge p \le n \longrightarrow f\ p =$ $(0::real))$

thm SUM_SING_NUMSEG:

$\forall (f::nat \Rightarrow real)$ $n::nat.$ $sum\ (dotdot\ n\ n)\ f = f\ n$

thm SUM_CLAUSES_NUMSEG:

$(\forall m::nat.\ sum\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow real) = (if\ m = (0::nat)\ then$ $?f\ (0::nat)\ else\ (0::real))) \wedge (\forall (m::nat)\ n::nat.\ sum\ (dotdot\ m\ (Suc\ n))\ ?f =$ $(if\ m \le Suc\ n\ then\ sum\ (dotdot\ m\ n)\ ?f + ?f\ (Suc\ n)\ else\ sum\ (dotdot\ m\ n)$ $?f))$

thm SUM_SWAP_NUMSEG:

$\forall (a::nat)$ $(b::nat)$ $(c::nat)$ $(d::nat)$ $f::nat \Rightarrow nat \Rightarrow real.$ $sum\ (dotdot\ a\ b)$ $(\lambda i::nat.\ sum\ (dotdot\ c\ d)\ (f\ i)) = sum\ (dotdot\ c\ d)\ (\lambda j::nat.\ sum\ (dotdot\ a\ b)$ $(\lambda i::nat.\ f\ i\ j))$

thm SUM_ADD_SPLIT:

$\forall (f::nat \Rightarrow real)$ $(m::nat)$ $(n::nat)$ $p::nat.$ $m \le n + (1::nat) \longrightarrow sum\ (dotdot$ $m\ (n + p))\ f = sum\ (dotdot\ m\ n)\ f + sum\ (dotdot\ (n + (1::nat))\ (n + p))\ f$

thm SUM_OFFSET:

$\forall (p::nat)$ $(f::nat \Rightarrow real)$ $(m::nat)$ $n::nat.$ $sum\ (dotdot\ (m + p)\ (n + p))\ f =$ $sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ (i + p))$

thm SUM_OFFSET_0:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow sum\ (dotdot\ m\ n)\ f = sum\ (dotdot\ (0::nat)\ (n - m))\ (\lambda i::nat.\ f\ (i + m))$

thm SUM_CLAUSES_LEFT:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow sum\ (dotdot\ m\ n)\ f = f\ m + sum\ (dotdot\ (m + (1::nat))\ n)\ f$

thm SUM_CLAUSES_NUMSEG_conjunct1:

$\forall (m::nat)\ n::nat.\ sum\ (dotdot\ m\ (Suc\ n))\ (?f::nat \Rightarrow real) = (if\ m \leq Suc\ n\ then\ sum\ (dotdot\ m\ n)\ ?f + ?f\ (Suc\ n)\ else\ sum\ (dotdot\ m\ n)\ ?f)$

thm SUM_CLAUSES_NUMSEG_conjunct0:

$\forall m::nat.\ sum\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow real) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ (0::real))$

thm SUM_CLAUSES_RIGHT:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (0::nat) < n \wedge m \leq n \longrightarrow sum\ (dotdot\ m\ n)\ f = sum\ (dotdot\ m\ (n - (1::nat)))\ f + f\ n$

thm SUM_PAIR:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (dotdot\ ((2::nat) * m)\ ((2::nat) * n + (1::nat)))\ f = sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ f\ ((2::nat) * i) + f\ ((2::nat) * i + (1::nat)))$

thm REAL_OF_NUM_SUM_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ real\_of\_nat\ (nsum\ (dotdot\ m\ n)\ f) = sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ real\_of\_nat\ (f\ i))$

thm SUM_PARTIAL_SUC:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ k * (g\ (k + (1::nat)) - g\ k)) = (if\ m \leq n\ then\ f\ (n + (1::nat)) * g\ (n + (1::nat)) - f\ m * g\ m - sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ g\ (k + (1::nat)) * (f\ (k + (1::nat)) - f\ k))\ else\ (0::real))$

thm SUM_PARTIAL_PRE:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ k * (g\ k - g\ (k - (1::nat)))) = (if\ m \leq n\ then\ f\ (n + (1::nat)) * g\ n - f\ m * g\ (m - (1::nat)) - sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ g\ k * (f\ (k + (1::nat)) - f\ k))\ else\ (0::real))$

thm SUM_DIFFS:

$\forall (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ (?f::nat \Rightarrow real)\ k - ?f\ (k + (1::nat))) = (if\ m \leq n\ then\ ?f\ m - ?f\ (n + (1::nat))\ else\ (0::real))$

thm SUM_DIFFS_ALT:

$\forall (m::nat)\ n::nat.\ sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ (?f::nat \Rightarrow real)\ (k + (1::nat)) - ?f\ k) = (if\ m \leq n\ then\ ?f\ (n + (1::nat)) - ?f\ m\ else\ (0::real))$

thm SUM_COMBINE_R:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ (n::nat)\ p::nat.\ m \leq n + (1::nat) \wedge n \leq p \longrightarrow sum\ (dotdot\ m\ n)\ f + sum\ (dotdot\ (n + (1::nat))\ p)\ f = sum\ (dotdot\ m\ p)\ f$

thm SUM_COMBINE_L:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ (n::nat)\ p::nat.\ (0::nat) < n \wedge m \leq n \wedge n \leq p + (1::nat) \longrightarrow sum\ (dotdot\ m\ (n - (1::nat)))\ f + sum\ (dotdot\ n\ p)\ f = sum\ (dotdot\ m\ p)\ f$

thm REAL_SUB_POW:

$\forall (x::real)\ (y::real)\ n::nat.\ (1::nat) \leq n \longrightarrow x^n - y^n = (x - y) * sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ x^i * y^{n - (1::nat) - i})$

thm REAL_SUB_POW_R1:

$\forall (x::real)\ n::nat.\ (1::nat) \leq n \longrightarrow x^n - (1::real) = (x - (1::real)) * sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (op\ \hat{}\ x)$

thm REAL_SUB_POW_L1:

$\forall (x::real)\ n::nat.\ (1::nat) \leq n \longrightarrow (1::real) - x^n = ((1::real) - x) * sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (op\ \hat{}\ x)$

thm REAL_SUB_POLYFUN:

$\forall (a::nat \Rightarrow real)\ (x::real)\ (y::real)\ n::nat.\ (1::nat) \leq n \longrightarrow sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ a\ i * x^i) - sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ a\ i * y^i) = (x - y) * sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda j::nat.\ sum\ (dotdot\ (j + (1::nat))\ n)\ (\lambda i::nat.\ a\ i * y^{i - j - (1::nat)}) * x^j)$

thm REAL_SUB_POLYFUN_ALT:

$\forall (a::nat \Rightarrow real)\ (x::real)\ (y::real)\ n::nat.\ (1::nat) \leq n \longrightarrow sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ a\ i * x^i) - sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ a\ i * y^i) = (x - y) * sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda j::nat.\ sum\ (dotdot\ (0::nat)\ (n - j - (1::nat)))\ (\lambda k::nat.\ a\ (j + (k + (1::nat))) * y^k) * x^j)$

thm REAL_POLYFUN_ROOTBOUND:

$\forall (n::nat)\ c::nat \Rightarrow real.\ \neg (\forall i::nat.\ IN\ i\ (dotdot\ (0::nat)\ n) \longrightarrow c\ i = (0::real)) \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%197::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%197\ (sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ c\ i * x^i) = (0::real))\ x)) \wedge CARD\ (GSPEC\ (\lambda GEN\%PVAR\%198::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%198\ (sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ c\ i * x^i) = (0::real))\ x)) \leq n$

thm REAL_POLYFUN_FINITE_ROOTS:

$\forall (n::nat)\ c::nat \Rightarrow real.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%200::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%200\ (sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ c\ i * x^i) = (0::real))\ x)) = (\exists i::nat.\ IN\ i\ (dotdot\ (0::nat)\ n) \wedge c\ i \neq (0::real))$

thm REAL_POLYFUN_EQ_0:

$\forall (n::nat)\ c::nat \Rightarrow real.\ (\forall x::real.\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ c\ i * x^{i}) = (0::real)) = (\forall i::nat.\ IN\ i\ (dotdot\ (0::nat)\ n) \longrightarrow c\ i = (0::real))$

thm REAL_POLYFUN_EQ_CONST:

$\forall (n::nat)\ (c::nat \Rightarrow real)\ k::real.\ (\forall x::real.\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ c\ i * x^{i}) = k) = (c\ (0::nat) = k \land (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow c\ i = (0::real)))$

thm DEF_dimindex:

$dimindex = (\lambda\_52367::?'a::type \Rightarrow bool.\ if\ FINITE\ HOL\_Light\_Import.UNIV\ then\ CARD\ HOL\_Light\_Import.UNIV\ else\ (1::nat))$

thm dimindex:

$\forall s::?'a::type \Rightarrow bool.\ dimindex\ s = (if\ FINITE\ HOL\_Light\_Import.UNIV\ then\ CARD\ HOL\_Light\_Import.UNIV\ else\ (1::nat))$

thm DIMINDEX_NONZERO:

$\forall s::?'a::type \Rightarrow bool.\ dimindex\ s \neq (0::nat)$

thm DIMINDEX_GE_1:

$\forall s::?'a::type \Rightarrow bool.\ (1::nat) \leq dimindex\ s$

thm DIMINDEX_UNIV:

$\forall s::?'a::type \Rightarrow bool.\ dimindex\ s = dimindex\ HOL\_Light\_Import.UNIV$

thm DIMINDEX_UNIQUE:

$HAS\_SIZE\ HOL\_Light\_Import.UNIV\ (?n::nat) \longrightarrow dimindex\ HOL\_Light\_Import.UNIV = ?n$

thm TYDEF_finite_image:

$finite\_index\ (dest\_finite\_image\ (?a::?'a::type\ finite\_image)) = ?a \land IN\ (?r::nat)\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_image\ (finite\_index\ ?r) = ?r)$

thm finite_image_tybij_conjunct1:

$\forall r::nat.\ IN\ r\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_image\ (finite\_index\ r) = r)$

thm finite_image_tybij_conjunct0:

$\forall a::?'a::type\ finite\_image.\ finite\_index\ (dest\_finite\_image\ a) = a$

thm finite_image_tybij:

$(\forall a::?'a::type\ finite\_image.\ finite\_index\ (dest\_finite\_image\ a) = a) \land (\forall r::nat.\ IN\ r\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_image\ (finite\_index\ r) = r))$

thm FINITE_IMAGE_IMAGE:

HOL_Light_Import.UNIV = IMAGE finite_index (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV))

thm HAS_SIZE_FINITE_IMAGE:

∀ *s::?'a::type ⇒ bool. HAS_SIZE HOL_Light_Import.UNIV (dimindex s)*

thm CARD_FINITE_IMAGE:

∀ *s::?'a::type ⇒ bool. CARD HOL_Light_Import.UNIV = dimindex s*

thm FINITE_FINITE_IMAGE:

FINITE HOL_Light_Import.UNIV

thm DIMINDEX_FINITE_IMAGE:

∀ *(s::?'a::type finite_image ⇒ bool) t::?'a::type ⇒ bool. dimindex s = dimindex t*

thm FINITE_INDEX_WORKS:

∀ *i::?'a::type finite_image.* ∃!*n::nat. (1::nat) ≤ n ∧ n ≤ dimindex HOL_Light_Import.UNIV ∧ finite_index n = i*

thm FINITE_INDEX_INJ:

∀ *(i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ⟶ (finite_index i = finite_index j) = (i = j)*

thm FORALL_FINITE_INDEX:

(∀ k::?'a::type finite_image. (?P::?'a::type finite_image ⇒ bool) k) = (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ ?P (finite_index i))

thm TYDEF_cart:

mk_cart (dest_cart (?a::(?'b::type, ?'a::type) cart)) = ?a ∧ True = (dest_cart (mk_cart (?r::?'a::type finite_image ⇒ ?'b::type)) = ?r)

thm cart_tybij_conjunct1:

∀ *r::?'b::type finite_image ⇒ ?'a::type. True = (dest_cart (mk_cart r) = r)*

thm cart_tybij_conjunct0:

∀ *a::(?'b::type, ?'a::type) cart. mk_cart (dest_cart a) = a*

thm cart_tybij:

(∀ a::(?'b::type, ?'a::type) cart. mk_cart (dest_cart a) = a) ∧ (∀ r::?'a::type finite_image ⇒ ?'b::type. True = (dest_cart (mk_cart r) = r))

thm DEF_$:

$\$ = (\lambda(\_52701::(?'b::type, ?'a::type)\ cart)\ \_52702::nat.\ dest\_cart\ \_52701\ (finite\_index\ \_52702))$

thm finite_index:

$\forall (x::(?'b::type, ?'a::type)\ cart)\ i::nat.\ \$\ x\ i = dest\_cart\ x\ (finite\_index\ i)$

thm CART_EQ:

$\forall (x::(?'b::type, ?'a::type)\ cart)\ y::(?'b::type, ?'a::type)\ cart.\ (x = y) = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ i = \$\ y\ i)$

thm DEF_lambda:

$lambda = (\lambda\_52737::nat \Rightarrow ?'b::type.\ SOME\ f::(?'b::type, ?'a::type)\ cart.\ \forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ f\ i = \_52737\ i)$

thm lambda:

$\forall g::nat \Rightarrow ?'b::type.\ lambda\ g = (SOME\ f::(?'b::type, ?'a::type)\ cart.\ \forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ f\ i = g\ i)$

thm LAMBDA_BETA:

$\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (lambda\ (?g::nat \Rightarrow ?'a::type))\ i = ?g\ i$

thm LAMBDA_UNIQUE:

$\forall (f::(?'b::type, ?'a::type)\ cart)\ g::nat \Rightarrow ?'b::type.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ f\ i = g\ i) = (lambda\ g = f)$

thm LAMBDA_ETA:

$\forall g::(?'b::type, ?'a::type)\ cart.\ lambda\ (\$\ g) = g$

thm FINITE_INDEX_INRANGE:

$\forall i::nat.\ \exists k \geq 1::nat.\ k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(?'a::type, ?'b::type)\ cart.\ \$\ x\ i = \$\ x\ k)$

thm FINITE_INDEX_INRANGE_2:

$\forall i::nat.\ \exists k \geq 1::nat.\ k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(?'b::type, ?'c::type)\ cart.\ \$\ x\ i = \$\ x\ k) \wedge (\forall y::(?'a::type, ?'c::type)\ cart.\ \$\ y\ i = \$\ y\ k)$

thm CART_EQ_FULL:

$\forall (x::(?'b::type, ?'a::type)\ cart)\ y::(?'b::type, ?'a::type)\ cart.\ (x = y) = (\forall i::nat.\ \$\ x\ i = \$\ y\ i)$

thm TYDEF_finite_sum:

$mk\_finite\_sum\ (dest\_finite\_sum\ (?a::(?'b::type, ?'a::type)\ finite\_sum)) = ?a \wedge IN\ (?r::nat)\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV + dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_sum\ (mk\_finite\_sum\ ?r) = ?r)$

thm finite_sum_tybij_conjunct1:

$\forall r::nat.\ IN\ r\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV + dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_sum\ (mk\_finite\_sum\ r) = r)$

thm finite_sum_tybij_conjunct0:

$\forall a::(?'b::type,\ ?'a::type)\ finite\_sum.\ mk\_finite\_sum\ (dest\_finite\_sum\ a) = a$

thm finite_sum_tybij:

$(\forall a::(?'b::type,\ ?'a::type)\ finite\_sum.\ mk\_finite\_sum\ (dest\_finite\_sum\ a) = a) \wedge (\forall r::nat.\ IN\ r\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV + dimindex\ HOL\_Light\_Import.UNIV)) = (dest\_finite\_sum\ (mk\_finite\_sum\ r) = r))$

thm DEF_pastecart:

$pastecart = (\lambda(\_53028::(?'c::type,\ ?'b::type)\ cart)\ \_53029::(?'c::type,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ if\ i \leq dimindex\ HOL\_Light\_Import.UNIV\ then\ \$\ \_53028\ i\ else\ \$\ \_53029\ (i - dimindex\ HOL\_Light\_Import.UNIV)))$

thm pastecart:

$\forall (f::(?'c::type,\ ?'b::type)\ cart)\ g::(?'c::type,\ ?'a::type)\ cart.\ pastecart\ f\ g = lambda\ (\lambda i::nat.\ if\ i \leq dimindex\ HOL\_Light\_Import.UNIV\ then\ \$\ f\ i\ else\ \$\ g\ (i - dimindex\ HOL\_Light\_Import.UNIV))$

thm DEF_fstcart:

$fstcart = (\lambda\_53040::(?'c::type,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ lambda\ (\$\ \_53040))$

thm fstcart:

$\forall f::(?'c::type,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ fstcart\ f = lambda\ (\$\ f)$

thm DEF_sndcart:

$sndcart = (\lambda\_53045::(?'c::type,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ lambda\ (\lambda i::nat.\ \$\ \_53045\ (i + dimindex\ HOL\_Light\_Import.UNIV)))$

thm sndcart:

$\forall f::(?'c::type,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ sndcart\ f = lambda\ (\lambda i::nat.\ \$\ f\ (i + dimindex\ HOL\_Light\_Import.UNIV))$

thm FINITE_SUM_IMAGE:

$HOL\_Light\_Import.UNIV = IMAGE\ mk\_finite\_sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV + dimindex\ HOL\_Light\_Import.UNIV))$

thm DIMINDEX_HAS_SIZE_FINITE_SUM:

$HAS\_SIZE\ HOL\_Light\_Import.UNIV\ (dimindex\ HOL\_Light\_Import.UNIV + dimindex\ HOL\_Light\_Import.UNIV)$

thm DIMINDEX_FINITE_SUM:

dimindex HOL_Light_Import.UNIV = dimindex HOL_Light_Import.UNIV + dimindex HOL_Light_Import.UNIV

thm FSTCART_PASTECART:

∀ (x::(?'c::type, ?'b::type) cart) y::(?'c::type, ?'a::type) cart. fstcart (pastecart x y) = x

thm SNDCART_PASTECART:

∀ (x::(?'c::type, ?'b::type) cart) y::(?'c::type, ?'a::type) cart. sndcart (pastecart x y) = y

thm PASTECART_FST_SND:

∀ z::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart. pastecart (fstcart z) (sndcart z) = z

thm PASTECART_EQ:

∀ (x::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart) y::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart. (x = y) = (fstcart x = fstcart y ∧ sndcart x = sndcart y)

thm FORALL_PASTECART:

(∀ p::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart. (?P::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart ⇒ bool) p) = (∀ (x::(?'c::type, ?'b::type) cart) y::(?'c::type, ?'a::type) cart. ?P (pastecart x y))

thm EXISTS_PASTECART:

(∃ p::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart. (?P::(?'c::type, (?'b::type, ?'a::type) finite_sum) cart ⇒ bool) p) = (∃ (x::(?'c::type, ?'b::type) cart) y::(?'c::type, ?'a::type) cart. ?P (pastecart x y))

thm HAS_SIZE_1:

HAS_SIZE HOL_Light_Import.UNIV (1::nat)

thm TYDEF_2:

mk_auto_define_finite_type_2 (dest_auto_define_finite_type_2 (?a::2)) = ?a ∧ IN (?r::nat) (dotdot (1::nat) (2::nat)) = (dest_auto_define_finite_type_2 (mk_auto_define_finite_type_2 ?r) = ?r)

thm HAS_SIZE_2:

HAS_SIZE HOL_Light_Import.UNIV (2::nat)

thm TYDEF_3:

mk_auto_define_finite_type_3 (dest_auto_define_finite_type_3 (?a::3)) = ?a ∧ IN (?r::nat) (dotdot (1::nat) (3::nat)) = (dest_auto_define_finite_type_3 (mk_auto_define_finite_type_3 ?r) = ?r)

thm HAS_SIZE_3:

HAS_SIZE HOL_Light_Import.UNIV (*3*::*nat*)

thm DIMINDEX_1:

dimindex HOL_Light_Import.UNIV = (*1*::*nat*)

thm DIMINDEX_2:

dimindex HOL_Light_Import.UNIV = (*2*::*nat*)

thm DIMINDEX_3:

dimindex HOL_Light_Import.UNIV = (*3*::*nat*)

thm Hypermap.GE_1:

∀ *n*::*nat*. (*1*::*nat*) ≤ *Suc n*

thm FINITE_CART:

∀ *P*::*nat* ⇒ *?'b*::*type* ⇒ *bool*. (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%207*::*?'b*::*type*. ∃ *x*::*?'b*::*type*. *SETSPEC* *GEN%PVAR%207* (*P i x*) *x*))) ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%208*::(*?'b*::*type*, *?'a*::*type*) *cart*. ∃ *v*::(*?'b*::*type*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%208* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *P i* ($ *v* *i*)) *v*))

thm HAS_SIZE_CART_UNIV:

∀ *m*::*nat*. *HAS_SIZE HOL_Light_Import.UNIV m* ⟶ *HAS_SIZE HOL_Light_Import.UNIV* $m^{dimindex\ HOL\_Light\_Import.UNIV}$

thm CARD_CART_UNIV:

FINITE HOL_Light_Import.UNIV ⟶ *CARD HOL_Light_Import.UNIV* = $(CARD\ HOL\_Light\_Import.UNIV)^{dimindex\ HOL\_Light\_Import.UNIV}$

thm FINITE_CART_UNIV:

FINITE HOL_Light_Import.UNIV ⟶ *FINITE HOL_Light_Import.UNIV*

thm DEF_vector:

vector = (λ*_54977*::*?'b*::*type list*. *lambda* (λ*i*::*nat*. *EL* (*i* − (*1*::*nat*)) *_54977*))

thm vector:

∀ *l*::*?'b*::*type list*. *vector l* = *lambda* (λ*i*::*nat*. *EL* (*i* − (*1*::*nat*)) *l*)

thm IN_ELIM_PASTECART_THM:

∀ (*P*::(*?'c*::*type*, *?'b*::*type*) *cart* ⇒ (*?'c*::*type*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*?'c*::*type*, *?'b*::*type*) *cart*) *b*::(*?'c*::*type*, *?'a*::*type*) *cart*. *IN* (*pastecart a b*) (*GSPEC* (λ*GEN%PVAR%209*::(*?'c*::*type*, (*?'b*::*type*, *?'a*::*type*) *finite_sum*) *cart*. ∃ (*x*::(*?'c*::*type*, *?'b*::*type*) *cart*) *y*::(*?'c*::*type*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%209* (*P x y*) (*pastecart x y*))) = *P a* *b*

thm DEF_CASEWISE:

CASEWISE = (SOME CASEWISE::nat ⇒ ((?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type)) list ⇒ ?'b::type ⇒ ?'c::type ⇒ ?'a::type. ∀ _55007::nat. (∀ (f::?'b::type) x::?'c::type. CASEWISE _55007 [] f x = (SOME y::?'a::type. True)) ∧ (∀ (h::(?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type)) (t::((?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type)) list) (f::?'b::type) x::?'c::type. CASEWISE _55007 (h # t) f x = (if ∃ y::?'d::type. fst h y = x then snd h f (SOME y::?'d::type. fst h y = x) else CASEWISE _55007 t f x))) (47::nat)

thm CASEWISE_DEF_conjunct0:

CASEWISE [] (?f::?'a::type) (?x::?'b::type) = (SOME y::?'d::type. True)

thm CASEWISE_DEF_conjunct1:

CASEWISE ((?h::(?'c::type ⇒ ?'b::type) × (?'a::type ⇒ ?'c::type ⇒ ?'d::type)) # (?t::((?'c::type ⇒ ?'b::type) × (?'a::type ⇒ ?'c::type ⇒ ?'d::type)) list)) (?f::?'a::type) (?x::?'b::type) = (if ∃ y::?'c::type. fst ?h y = ?x then snd ?h ?f (SOME y::?'c::type. fst ?h y = ?x) else CASEWISE ?t ?f ?x)

thm CASEWISE_DEF:

CASEWISE [] (?f::?'a::type) (?x::?'b::type) = (SOME y::?'d::type. True) ∧ CASEWISE ((?h::(?'c::type ⇒ ?'b::type) × (?'a::type ⇒ ?'c::type ⇒ ?'d::type)) # (?t::((?'c::type ⇒ ?'b::type) × (?'a::type ⇒ ?'c::type ⇒ ?'d::type)) list)) ?f ?x = (if ∃ y::?'c::type. fst ?h y = ?x then snd ?h ?f (SOME y::?'c::type. fst ?h y = ?x) else CASEWISE ?t ?f ?x)

thm CASEWISE_conjunct0:

CASEWISE [] (?f::?'a::type) (?x::?'b::type) = (SOME y::?'d::type. True)

thm CASEWISE_conjunct1:

CASEWISE ((?s::?'c::type ⇒ ?'b::type, ?t::?'a::type ⇒ ?'c::type ⇒ ?'d::type) # (?clauses::((?'c::type ⇒ ?'b::type) × (?'a::type ⇒ ?'c::type ⇒ ?'d::type)) list)) (?f::?'a::type) (?x::?'b::type) = (if ∃ y::?'c::type. ?s y = ?x then ?t ?f (SOME y::?'c::type. ?s y = ?x) else CASEWISE ?clauses ?f ?x)

thm CASEWISE:

CASEWISE [] (?f::?'c::type) (?x::?'d::type) = (SOME y::?'f::type. True) ∧ CASEWISE ((?s::?'a::type ⇒ ?'d::type, ?t::?'c::type ⇒ ?'a::type ⇒ ?'b::type) # (?clauses::((?'a::type ⇒ ?'d::type) × (?'c::type ⇒ ?'a::type ⇒ ?'b::type)) list)) ?f ?x = (if ∃ y::?'a::type. ?s y = ?x then ?t ?f (SOME y::?'a::type. ?s y = ?x) else CASEWISE ?clauses ?f ?x)

thm CASEWISE_CASES:

∀ (clauses::((?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type)) list) (c::?'b::type) x::?'c::type. (∃ (s::?'d::type ⇒ ?'c::type) (t::?'b::type ⇒ ?'d::type ⇒ ?'a::type) a::?'d::type. MEM (s, t) clauses ∧ s a = x ∧ CASEWISE clauses c x = t c a) ∨ ¬ (∃ (s::?'d::type ⇒ ?'c::type) (t::?'b::type ⇒ ?'d::type ⇒

?'a::type) a::?'d::type. MEM (s, t) clauses ∧ s a = x) ∧ CASEWISE clauses c x = (SOME y::?'a::type. True)

thm CASEWISE_WORKS:

∀ (clauses::((?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type)) list) c::?'b::type. (∀ (s::?'d::type ⇒ ?'c::type) (t::?'b::type ⇒ ?'d::type ⇒ ?'a::type) (s'::?'d::type ⇒ ?'c::type) (t'::?'b::type ⇒ ?'d::type ⇒ ?'a::type) (x::?'d::type) y::?'d::type. MEM (s, t) clauses ∧ MEM (s', t') clauses ∧ s x = s' y ⟶ t c x = t' c y) ⟶ list_all (GABS (λf::(?'d::type ⇒ ?'c::type) × (?'b::type ⇒ ?'d::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'d::type ⇒ ?'c::type) t::?'b::type ⇒ ?'d::type ⇒ ?'a::type. GEQ (f (s, t)) (∀ x::?'d::type. CASEWISE clauses c (s x) = t c x))) clauses

thm DEF_admissible:

admissible = (λ(_56074::?'e::type ⇒ ?'d::type ⇒ bool) (_56075::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ bool) (_56076::?'b::type ⇒ ?'d::type) _56077::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ ?'a::type. ∀ (f::?'e::type ⇒ ?'c::type) (g::?'e::type ⇒ ?'c::type) a::?'b::type. _56075 f a ∧ _56075 g a ∧ (∀ z::?'e::type. _56074 z (_56076 a) ⟶ f z = g z) ⟶ _56077 f a = _56077 g a)

thm admissible:

∀ (p::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (<<::?'e::type ⇒ ?'b::type ⇒ bool) (s::?'c::type ⇒ ?'b::type) t::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'a::type. admissible << p s t = (∀ (f::?'e::type ⇒ ?'d::type) (g::?'e::type ⇒ ?'d::type) a::?'c::type. p f a ∧ p g a ∧ (∀ z::?'e::type. << z (s a) ⟶ f z = g z) ⟶ t f a = t g a)

thm DEF_tailadmissible:

tailadmissible = (λ(_56106::?'c::type ⇒ ?'c::type ⇒ bool) (_56107::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (_56108::?'a::type ⇒ ?'c::type) _56109::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type. ∃ (P::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (G::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'c::type) H::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type. (∀ (f::?'c::type ⇒ ?'b::type) (a::?'a::type) y::?'c::type. P f a ∧ _56106 y (G f a) ⟶ _56106 y (_56108 a)) ∧ (∀ (f::?'c::type ⇒ ?'b::type) (g::?'c::type ⇒ ?'b::type) a::?'a::type. (∀ z::?'c::type. _56106 z (_56108 a) ⟶ f z = g z) ⟶ P f a = P g a ∧ G f a = G g a ∧ H f a = H g a) ∧ (∀ (f::?'c::type ⇒ ?'b::type) a::?'a::type. _56107 f a ⟶ _56109 f a = (if P f a then f (G f a) else H f a)))

thm tailadmissible:

∀ (<<::?'c::type ⇒ ?'c::type ⇒ bool) (s::?'b::type ⇒ ?'c::type) (p::(?'c::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ bool) t::(?'c::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ ?'a::type. tailadmissible << p s t = (∃ (P::(?'c::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ bool) (G::(?'c::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ ?'c::type) H::(?'c::type ⇒ ?'a::type) ⇒ ?'b::type ⇒ ?'a::type. (∀ (f::?'c::type ⇒ ?'a::type) (a::?'b::type) y::?'c::type. P f a ∧ << y (G f a) ⟶ << y (s a)) ∧ (∀ (f::?'c::type ⇒

?'a::type) (g::?'c::type ⇒ ?'a::type) a::?'b::type. (∀ z::?'c::type. << z (s a) ⟶ f z = g z) ⟶ P f a = P g a ∧ G f a = G g a ∧ H f a = H g a) ∧ (∀ (f::?'c::type ⇒ ?'a::type) a::?'b::type. p f a ⟶ t f a = (if P f a then f (G f a) else H f a)))

thm DEF_superadmissible:

superadmissible = (λ(_56138::?'c::type ⇒ ?'c::type ⇒ bool) (_56139::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (_56140::?'a::type ⇒ ?'c::type) _56141::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type. admissible _56138 (λ(f::?'c::type ⇒ ?'b::type) a::?'a::type. True) _56140 _56139 ⟶ tailadmissible _56138 _56139 _56140 _56141)

thm superadmissible:

∀ (<<::?'c::type ⇒ ?'c::type ⇒ bool) (p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) t::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type. superadmissible << p s t = (admissible << (λ(f::?'c::type ⇒ ?'b::type) a::?'a::type. True) s p ⟶ tailadmissible << p s t)

thm MATCH_SEQPATTERN:

_MATCH (?x::?'a::type) (_SEQPATTERN (?r::?'a::type ⇒ ?'b::type ⇒ bool) (?s::?'a::type ⇒ ?'b::type ⇒ bool)) = (if ∃ y::?'b::type. ?r ?x y then _MATCH ?x ?r else _MATCH ?x ?s)

thm ADMISSIBLE_CONST:

∀ (p::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'b::type) c::?'c::type ⇒ ?'a::type. admissible (?<<::?'e::type ⇒ ?'b::type ⇒ bool) p s (λf::?'e::type ⇒ ?'d::type. c)

thm ADMISSIBLE_BASE:

∀ (<<::?'c::type ⇒ ?'c::type ⇒ bool) (p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) t::?'a::type ⇒ ?'c::type. (∀ (f::?'c::type ⇒ ?'b::type) a::?'a::type. p f a ⟶ << (t a) (s a)) ⟶ admissible << p s (λ(f::?'c::type ⇒ ?'b::type) x::?'a::type. f (t x))

thm ADMISSIBLE_COMB:

∀ (<<::?'e::type ⇒ ?'e::type ⇒ bool) (p::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (g::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type) y::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type. admissible << p s g ∧ admissible << p s y ⟶ admissible << p s (λ(f::?'e::type ⇒ ?'d::type) x::?'c::type. g f x (y f x))

thm ADMISSIBLE_RAND:

∀ (<<::?'e::type ⇒ ?'e::type ⇒ bool) (p::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (g::?'c::type ⇒ ?'b::type ⇒ ?'a::type) y::(?'e::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type. admissible << p s y ⟶ admissible << p s (λ(f::?'e::type ⇒ ?'d::type) x::?'c::type. g x (y f x))

thm ADMISSIBLE_LAMBDA:

∀ (<<::?'d::type ⇒ ?'d::type ⇒ bool) (p::(?'d::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ bool) (s::?'b::type ⇒ ?'d::type) t::(?'d::type ⇒ ?'c::type) ⇒ ?'a::type ⇒ ?'b::type ⇒ bool. admissible << (λf::?'d::type ⇒ ?'c::type. GABS (λfa::?'a::type × ?'b::type ⇒ bool. ∀ (u::?'a::type) x::?'b::type. GEQ (fa (u, x)) (p f x))) (GABS (λf::?'a::type × ?'b::type ⇒ ?'d::type. ∀ (u::?'a::type) x::?'b::type. GEQ (f (u, x)) (s x))) (λf::?'d::type ⇒ ?'c::type. GABS (λfa::?'a::type × ?'b::type ⇒ bool. ∀ (u::?'a::type) x::?'b::type. GEQ (fa (u, x)) (t f u x))) ⟶ admissible << p s (λ(f::?'d::type ⇒ ?'c::type) (x::?'b::type) u::?'a::type. t f u x)

thm ADMISSIBLE_NEST:

∀ (<<::?'c::type ⇒ ?'c::type ⇒ bool) (p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) t::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'c::type. admissible << p s t ∧ (∀ (f::?'c::type ⇒ ?'b::type) a::?'a::type. p f a ⟶ << (t f a) (s a)) ⟶ admissible << p s (λ(f::?'c::type ⇒ ?'b::type) x::?'a::type. f (t f x))

thm ADMISSIBLE_COND:

∀ (<<::?'e::type ⇒ ?'d::type ⇒ bool) (p::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ bool) (P::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ bool) (s::?'b::type ⇒ ?'d::type) (h::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ ?'a::type) k::(?'e::type ⇒ ?'c::type) ⇒ ?'b::type ⇒ ?'a::type. admissible << p s P ∧ admissible << (λ(f::?'e::type ⇒ ?'c::type) x::?'b::type. p f x ∧ P f x) s h ∧ admissible << (λ(f::?'e::type ⇒ ?'c::type) x::?'b::type. p f x ∧ ¬ P f x) s k ⟶ admissible << p s (λ(f::?'e::type ⇒ ?'c::type) x::?'b::type. if P f x then h f x else k f x)

thm ADMISSIBLE_MATCH:

∀ (<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (e::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type) c::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. admissible << p s e ∧ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. c f x (e f x)) ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _MATCH (e f x) (c f x))

thm ADMISSIBLE_SEQPATTERN:

∀ (<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (c1::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool) (c2::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool) e::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type. admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. ∃ y::?'a::type. c1 f x (e f x) y) ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ (∃ y::?'a::type. c1 f x (e f x) y)) s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. c1 f x (e f x)) ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ ¬ (∃ y::?'a::type. c1 f x (e f x) y)) s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type.

c2 f x (e f x)) ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _SEQPATTERN (c1 f x) (c2 f x) (e f x))

thm ADMISSIBLE_UNGUARDED_PATTERN:

∀(<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (pat::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type) (e::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type) (t::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'a::type) y::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'a::type. admissible << p s pat ∧ admissible << p s e ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ pat f x = e f x) s t ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ pat f x = e f x) s y ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _UNGUARDED_PATTERN (GEQ (pat f x) (e f x)) (GEQ (t f x) (y f x)))

thm ADMISSIBLE_GUARDED_PATTERN:

∀(<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (pat::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type) (q::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (e::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type) (t::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'a::type) y::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'a::type. admissible << p s pat ∧ admissible << p s e ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ pat f x = e f x ∧ q f x) s t ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ pat f x = e f x) s q ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ pat f x = e f x ∧ q f x) s y ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _GUARDED_PATTERN (GEQ (pat f x) (e f x)) (q f x) (GEQ (t f x) (y f x)))

thm ADMISSIBLE_NSUM:

∀(<<::?'d::type ⇒ ?'c::type ⇒ bool) (p::(?'d::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) (h::(?'d::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ nat ⇒ nat) (a::?'a::type ⇒ nat) b::?'a::type ⇒ nat. admissible << (λf::?'d::type ⇒ ?'b::type. GABS (λfa::nat × ?'a::type ⇒ bool. ∀(k::nat) x::?'a::type. GEQ (fa (k, x)) (a x ≤ k ∧ k ≤ b x ∧ p f x))) (GABS (λf::nat × ?'a::type ⇒ ?'c::type. ∀(k::nat) x::?'a::type. GEQ (f (k, x)) (s x))) (λf::?'d::type ⇒ ?'b::type. GABS (λfa::nat × ?'a::type ⇒ nat. ∀(k::nat) x::?'a::type. GEQ (fa (k, x)) (h f x k))) ⟶ admissible << p s (λ(f::?'d::type ⇒ ?'b::type) x::?'a::type. nsum (dotdot (a x) (b x)) (h f x))

thm ADMISSIBLE_SUM:

∀(<<::?'d::type ⇒ ?'c::type ⇒ bool) (p::(?'d::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) (h::(?'d::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ nat ⇒ real) (a::?'a::type ⇒ nat) b::?'a::type ⇒ nat. admissible << (λf::?'d::type ⇒ ?'b::type. GABS (λfa::nat × ?'a::type ⇒ bool. ∀(k::nat) x::?'a::type. GEQ (fa (k, x)) (a x ≤ k ∧ k ≤ b x ∧ p f x))) (GABS (λf::nat × ?'a::type ⇒ ?'c::type. ∀(k::nat) x::?'a::type. GEQ (f (k, x)) (s x))) (λf::?'d::type ⇒ ?'b::type. GABS (λfa::nat × ?'a::type ⇒ real. ∀(k::nat) x::?'a::type. GEQ (fa (k, x)) (h f x k))) ⟶ admissible << p s (λ(f::?'d::type ⇒ ?'b::type) x::?'a::type. sum (dotdot (a x) (b x)) (h f x))

thm ADMISSIBLE_MAP:

∀(<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (h::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type) l::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type list. admissible << p s l ∧ admissible << (λf::?'f::type ⇒ ?'d::type. GABS (λfa::?'b::type × ?'c::type ⇒ bool. ∀(y::?'b::type) x::?'c::type. GEQ (fa (y, x)) (p f x ∧ MEM y (l f x)))) (GABS (λf::?'b::type × ?'c::type ⇒ ?'e::type. ∀(y::?'b::type) x::?'c::type. GEQ (f (y, x)) (s x))) (λf::?'f::type ⇒ ?'d::type. GABS (λfa::?'b::type × ?'c::type ⇒ ?'a::type. ∀(y::?'b::type) x::?'c::type. GEQ (fa (y, x)) (h f x y))) ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. map (h f x) (l f x))

thm ADMISSIBLE_MATCH_SEQPATTERN:

∀(<<::?'f::type ⇒ ?'e::type ⇒ bool) (p::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ bool) (s::?'c::type ⇒ ?'e::type) (c1::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool) (c2::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool) e::(?'f::type ⇒ ?'d::type) ⇒ ?'c::type ⇒ ?'b::type. admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. ∃y::?'a::type. c1 f x (e f x) y) ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ (∃y::?'a::type. c1 f x (e f x) y)) s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _MATCH (e f x) (c1 f x)) ∧ admissible << (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. p f x ∧ ¬ (∃y::?'a::type. c1 f x (e f x) y)) s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _MATCH (e f x) (c2 f x)) ⟶ admissible << p s (λ(f::?'f::type ⇒ ?'d::type) x::?'c::type. _MATCH (e f x) (_SEQPATTERN (c1 f x) (c2 f x)))

thm ADMISSIBLE_IMP_SUPERADMISSIBLE:

∀(<<::?'c::type ⇒ ?'c::type ⇒ bool) (p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) t::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type. admissible << p s t ⟶ superadmissible << p s t

thm SUPERADMISSIBLE_CONST:

∀(p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) c::?'a::type ⇒ ?'b::type. superadmissible (?<<::?'c::type ⇒ ?'c::type ⇒ bool) p s (λf::?'c::type ⇒ ?'b::type. c)

thm SUPERADMISSIBLE_TAIL:

∀(<<::?'c::type ⇒ ?'c::type ⇒ bool) (p::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ bool) (s::?'a::type ⇒ ?'c::type) t::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'c::type. admissible << p s t ∧ (∀(f::?'c::type ⇒ ?'b::type) a::?'a::type. p f a ⟶ (∀y::?'c::type. << y (t f a) ⟶ << y (s a))) ⟶ superadmissible << p s (λ(f::?'c::type ⇒ ?'b::type) x::?'a::type. f (t f x))

thm SUPERADMISSIBLE_COND:

$\forall (<<::?'c::type \Rightarrow ?'c::type \Rightarrow bool)\ (p::(?'c::type \Rightarrow ?'b::type) \Rightarrow ?'a::type \Rightarrow bool)\ (P::(?'c::type \Rightarrow ?'b::type) \Rightarrow ?'a::type \Rightarrow bool)\ (s::?'a::type \Rightarrow ?'c::type)\ (h::(?'c::type \Rightarrow ?'b::type) \Rightarrow ?'a::type \Rightarrow ?'b::type)\ k::(?'c::type \Rightarrow ?'b::type) \Rightarrow ?'a::type \Rightarrow ?'b::type.\ admissible << p\ s\ P \wedge superadmissible << (\lambda(f::?'c::type \Rightarrow ?'b::type)\ x::?'a::type.\ p\ f\ x \wedge P\ f\ x)\ s\ h \wedge superadmissible << (\lambda(f::?'c::type \Rightarrow ?'b::type)\ x::?'a::type.\ p\ f\ x \wedge \neg\ P\ f\ x)\ s\ k \longrightarrow superadmissible << p\ s\ (\lambda(f::?'c::type \Rightarrow ?'b::type)\ x::?'a::type.\ if\ P\ f\ x\ then\ h\ f\ x\ else\ k\ f\ x)$

thm SUPERADMISSIBLE_MATCH_SEQPATTERN:

$\forall (<<::?'d::type \Rightarrow ?'d::type \Rightarrow bool)\ (p::(?'d::type \Rightarrow ?'c::type) \Rightarrow ?'b::type \Rightarrow bool)\ (s::?'b::type \Rightarrow ?'d::type)\ (c1::(?'d::type \Rightarrow ?'c::type) \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'c::type \Rightarrow bool)\ (c2::(?'d::type \Rightarrow ?'c::type) \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'c::type \Rightarrow bool)\ e::(?'d::type \Rightarrow ?'c::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ admissible << p\ s\ (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ \exists\, y::?'c::type.\ c1\ f\ x\ (e\ f\ x)\ y) \wedge superadmissible << (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ p\ f\ x \wedge (\exists\, y::?'c::type.\ c1\ f\ x\ (e\ f\ x)\ y))\ s\ (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ \_MATCH\ (e\ f\ x)\ (c1\ f\ x)) \wedge superadmissible << (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ p\ f\ x \wedge \neg\ (\exists\, y::?'c::type.\ c1\ f\ x\ (e\ f\ x)\ y))\ s\ (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ \_MATCH\ (e\ f\ x)\ (c2\ f\ x)) \longrightarrow superadmissible << p\ s\ (\lambda(f::?'d::type \Rightarrow ?'c::type)\ x::?'b::type.\ \_MATCH\ (e\ f\ x)\ (\_SEQPATTERN\ (c1\ f\ x)\ (c2\ f\ x)))$

thm SUPERADMISSIBLE_MATCH_UNGUARDED_PATTERN:

$\forall (<<::?'e::type \Rightarrow ?'e::type \Rightarrow bool)\ (p::(?'e::type \Rightarrow ?'d::type) \Rightarrow ?'c::type \Rightarrow bool)\ (s::?'c::type \Rightarrow ?'e::type)\ (e::?'c::type \Rightarrow ?'b::type)\ (pat::?'a::type \Rightarrow ?'b::type)\ arg::?'c::type \Rightarrow ?'a::type \Rightarrow ?'e::type.\ (\forall (f::?'e::type \Rightarrow ?'d::type)\ (a::?'c::type)\ (t::?'a::type)\ u::?'a::type.\ p\ f\ a \wedge pat\ t = e\ a \wedge pat\ u = e\ a \longrightarrow arg\ a\ t = arg\ a\ u) \wedge (\forall (f::?'e::type \Rightarrow ?'d::type)\ (a::?'c::type)\ t::?'a::type.\ p\ f\ a \wedge pat\ t = e\ a \longrightarrow (\forall\, y::?'e::type.\ << y\ (arg\ a\ t) \longrightarrow << y\ (s\ a))) \longrightarrow superadmissible << p\ s\ (\lambda(f::?'e::type \Rightarrow ?'d::type)\ x::?'c::type.\ \_MATCH\ (e\ x)\ (\lambda(u::?'b::type)\ v::?'d::type.\ \exists\, t::?'a::type.\ \_UNGUARDED\_PATTERN\ (GEQ\ (pat\ t)\ u)\ (GEQ\ (f\ (arg\ x\ t))\ v)))$

thm SUPERADMISSIBLE_MATCH_GUARDED_PATTERN:

$\forall (<<::?'e::type \Rightarrow ?'e::type \Rightarrow bool)\ (p::(?'e::type \Rightarrow ?'d::type) \Rightarrow ?'c::type \Rightarrow bool)\ (s::?'c::type \Rightarrow ?'e::type)\ (e::?'c::type \Rightarrow ?'b::type)\ (pat::?'a::type \Rightarrow ?'b::type)\ (q::?'c::type \Rightarrow ?'a::type \Rightarrow bool)\ arg::?'c::type \Rightarrow ?'a::type \Rightarrow ?'e::type.\ (\forall (f::?'e::type \Rightarrow ?'d::type)\ (a::?'c::type)\ (t::?'a::type)\ u::?'a::type.\ p\ f\ a \wedge pat\ t = e\ a \wedge q\ a\ t \wedge pat\ u = e\ a \wedge q\ a\ u \longrightarrow arg\ a\ t = arg\ a\ u) \wedge (\forall (f::?'e::type \Rightarrow ?'d::type)\ (a::?'c::type)\ t::?'a::type.\ p\ f\ a \wedge q\ a\ t \wedge pat\ t = e\ a \longrightarrow (\forall\, y::?'e::type.\ << y\ (arg\ a\ t) \longrightarrow << y\ (s\ a))) \longrightarrow superadmissible << p\ s\ (\lambda(f::?'e::type \Rightarrow ?'d::type)\ x::?'c::type.\ \_MATCH\ (e\ x)\ (\lambda(u::?'b::type)\ v::?'d::type.\ \exists\, t::?'a::type.\ \_GUARDED\_PATTERN\ (GEQ\ (pat\ t)\ u)\ (q\ x\ t)\ (GEQ\ (f\ (arg\ x\ t))\ v)))$

thm WF_REC_TAIL_GENERAL′:

$\forall (P::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow bool)\ (G::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'b::type)\ (H::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type)\ H'::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ WF\ (?{<<}::?'b::type \Rightarrow ?'b::type \Rightarrow bool) \wedge (\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ (\forall z::?'b::type.\ ?{<<}\ z\ x \longrightarrow f\ z = g\ z) \longrightarrow P\ f\ x = P\ g\ x \wedge G\ f\ x = G\ g\ x \wedge H'\ f\ x = H'\ g\ x) \wedge (\forall (f::?'b::type \Rightarrow ?'a::type)\ (x::?'b::type)\ y::?'b::type.\ P\ f\ x \wedge ?{<<}\ y\ (G\ f\ x) \longrightarrow ?{<<}\ y\ x) \wedge (\forall (f::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ H\ f\ x = (if\ P\ f\ x\ then\ f\ (G\ f\ x)\ else\ H'\ f\ x)) \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ \forall x::?'b::type.\ f\ x = H\ f\ x)$

thm WF_REC_CASES:

$\forall ({<<}::?'c::type \Rightarrow ?'c::type \Rightarrow bool)\ clauses::((?'b::type \Rightarrow ?'c::type) \times ((?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type))\ list.\ WF\ {<<} \wedge list\_all\ (GABS\ (\lambda f::(?'b::type \Rightarrow ?'c::type) \times ((?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type) \Rightarrow bool.\ \forall (s::?'b::type \Rightarrow ?'c::type)\ t::(?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ GEQ\ (f\ (s,\ t))\ (\exists (P::(?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow bool)\ (G::(?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'c::type)\ H::(?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\forall (f::?'c::type \Rightarrow ?'a::type)\ (a::?'b::type)\ y::?'c::type.\ P\ f\ a \wedge {<<}\ y\ (G\ f\ a) \longrightarrow {<<}\ y\ (s\ a)) \wedge (\forall (f::?'c::type \Rightarrow ?'a::type)\ (g::?'c::type \Rightarrow ?'a::type)\ a::?'b::type.\ (\forall z::?'c::type.\ {<<}\ z\ (s\ a) \longrightarrow f\ z = g\ z) \longrightarrow P\ f\ a = P\ g\ a \wedge G\ f\ a = G\ g\ a \wedge H\ f\ a = H\ g\ a) \wedge (\forall (f::?'c::type \Rightarrow ?'a::type)\ a::?'b::type.\ t\ f\ a = (if\ P\ f\ a\ then\ f\ (G\ f\ a)\ else\ H\ f\ a)))))\ clauses \longrightarrow (\exists f::?'c::type \Rightarrow ?'a::type.\ \forall x::?'c::type.\ f\ x = CASEWISE\ clauses\ f\ x)$

thm WF_REC_CASES':

$\forall ({<<}::?'c::type \Rightarrow ?'c::type \Rightarrow bool)\ clauses::((?'b::type \Rightarrow ?'c::type) \times ((?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type))\ list.\ WF\ {<<} \wedge list\_all\ (GABS\ (\lambda f::(?'b::type \Rightarrow ?'c::type) \times ((?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type) \Rightarrow bool.\ \forall (s::?'b::type \Rightarrow ?'c::type)\ t::(?'c::type \Rightarrow ?'a::type) \Rightarrow ?'b::type \Rightarrow ?'a::type.\ GEQ\ (f\ (s,\ t))\ (tailadmissible\ {<<}\ (\lambda (f::?'c::type \Rightarrow ?'a::type)\ a::?'b::type.\ True)\ s\ t)))\ clauses \longrightarrow (\exists f::?'c::type \Rightarrow ?'a::type.\ \forall x::?'c::type.\ f\ x = CASEWISE\ clauses\ f\ x)$

thm RECURSION_CASEWISE:

$\forall clauses::((?'c::type \Rightarrow ?'b::type) \times ((?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type))\ list.\ (\exists {<<}::?'b::type \Rightarrow ?'b::type \Rightarrow bool.\ WF\ {<<} \wedge list\_all\ (GABS\ (\lambda f::(?'c::type \Rightarrow ?'b::type) \times ((?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type) \Rightarrow bool.\ \forall (s::?'c::type \Rightarrow ?'b::type)\ t::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type.\ GEQ\ (f\ (s,\ t))\ (tailadmissible\ {<<}\ (\lambda (f::?'b::type \Rightarrow ?'a::type)\ a::?'c::type.\ True)\ s\ t)))\ clauses) \wedge (\forall (s::?'c::type \Rightarrow ?'b::type)\ (t::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type)\ (s'::?'c::type \Rightarrow ?'b::type)\ (t'::(?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type)\ (f::?'b::type \Rightarrow ?'a::type)\ (x::?'c::type)\ y::?'c::type.\ MEM\ (s,\ t)\ clauses \wedge MEM\ (s',\ t')\ clauses \longrightarrow s\ x = s'\ y \longrightarrow t\ f\ x = t'\ f\ y) \longrightarrow (\exists f::?'b::type \Rightarrow ?'a::type.\ list\_all\ (GABS\ (\lambda fa::(?'c::type \Rightarrow ?'b::type) \times ((?'b::type \Rightarrow ?'a::type) \Rightarrow ?'c::type \Rightarrow ?'a::type) \Rightarrow bool.\ \forall (s::?'c::type \Rightarrow$

?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (fa (s, t)) (∀ x::?'c::type. f (s x) = t f x))) clauses)

thm RECURSION_CASEWISE_PAIRWISE:

∀ clauses::((?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type)) list. (∃ <<::?'b::type ⇒ ?'b::type ⇒ bool. WF << ∧ list_all (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (tailadmissible << (λ(f::?'b::type ⇒ ?'a::type) a::?'c::type. True) s t))) clauses) ∧ list_all (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (∀ (f::?'b::type ⇒ ?'a::type) (x::?'c::type) y::?'c::type. s x = s y ⟶ t f x = t f y))) clauses ∧ PAIRWISE (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ (?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s'::?'c::type ⇒ ?'b::type) t'::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s', t')) (∀ (f::?'b::type ⇒ ?'a::type) (x::?'c::type) y::?'c::type. s x = s' y ⟶ t f x = t' f y))))) clauses ⟶ (∃ f::?'b::type ⇒ ?'a::type. list_all (GABS (λfa::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (fa (s, t)) (∀ x::?'c::type. f (s x) = t f x))) clauses)

thm SUPERADMISSIBLE_T:

superadmissible (?<<::?'c::type ⇒ ?'c::type ⇒ bool) (λ(f::?'c::type ⇒ ?'b::type) x::?'a::type. True) (?s::?'a::type ⇒ ?'c::type) (?t::(?'c::type ⇒ ?'b::type) ⇒ ?'a::type ⇒ ?'b::type) = tailadmissible ?<< (λ(f::?'c::type ⇒ ?'b::type) x::?'a::type. True) ?s ?t

thm RECURSION_SUPERADMISSIBLE:

∀ clauses::((?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type)) list. (∃ <<::?'b::type ⇒ ?'b::type ⇒ bool. WF << ∧ list_all (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (superadmissible << (λ(f::?'b::type ⇒ ?'a::type) a::?'c::type. True) s t))) clauses) ∧ list_all (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (∀ (f::?'b::type ⇒ ?'a::type) (x::?'c::type) y::?'c::type. s x = s y ⟶ t f x = t f y))) clauses ∧ PAIRWISE (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ (?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type) ⇒ bool. ∀ (s::?'c::type ⇒ ?'b::type) t::(?'b::type ⇒ ?'a::type) ⇒ ?'c::type ⇒ ?'a::type. GEQ (f (s, t)) (GABS (λf::(?'c::type ⇒ ?'b::type) × ((?'b::type ⇒ ?'a::type) ⇒ ?'c::type

$\Rightarrow$ *?'a::type*) $\Rightarrow$ *bool*. $\forall$ (*s'::?'c::type* $\Rightarrow$ *?'b::type*) *t'*::(*?'b::type* $\Rightarrow$ *?'a::type*) $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'a::type*. *GEQ* (*f* (*s'*, *t'*)) ($\forall$ (*f::?'b::type* $\Rightarrow$ *?'a::type*) (*x::?'c::type*) *y::?'c::type*. *s x* = *s' y* $\longrightarrow$ *t f x* = *t' f y*))))) *clauses* $\longrightarrow$ ($\exists$ *f::?'b::type* $\Rightarrow$ *?'a::type*. *list_all* (*GABS* (λ*fa*::(*?'c::type* $\Rightarrow$ *?'b::type*) $\times$ ((*?'b::type* $\Rightarrow$ *?'a::type*) $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'a::type*) $\Rightarrow$ *bool*. $\forall$ (*s::?'c::type* $\Rightarrow$ *?'b::type*) *t*::(*?'b::type* $\Rightarrow$ *?'a::type*) $\Rightarrow$ *?'c::type* $\Rightarrow$ *?'a::type*. *GEQ* (*fa* (*s*, *t*)) ($\forall$ *x::?'c::type*. *f* (*s x*) = *t f x*))) *clauses*)

thm SUBSET_PRED:

$\forall$ (*P::?'a::type* $\Rightarrow$ *bool*) *Q::?'a::type* $\Rightarrow$ *bool*. *SUBSET P Q* = ($\forall$ *x::?'a::type*. *P x* $\longrightarrow$ *Q x*)

thm UNIONS_PRED:

UNIONS (*?P*::(*?'a::type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) = (λ*x::?'a::type*. $\exists$ *p::?'a::type* $\Rightarrow$ *bool*. *?P p* $\wedge$ *p x*)

thm DEF_less:

HOL_Light_Import.less = (λ(*_66062::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*) *_66063::?'a::type* $\times$ *?'a::type*. *_66062* (*fst _66063*, *snd _66063*) $\wedge$ *fst _66063* $\neq$ *snd _66063*)

thm less:

$\forall$ (*l::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*) (*x::?'a::type*) *y::?'a::type*. *HOL_Light_Import.less* *l* (*x*, *y*) = (*l* (*x*, *y*) $\wedge$ *x* $\neq$ *y*)

thm DEF_fl:

fl = (λ(*_66079::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*) *_66080::?'a::type*. $\exists$ *y::?'a::type*. *_66079* (*_66080*, *y*) $\vee$ *_66079* (*y*, *_66080*))

thm fl:

$\forall$ (*l::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*) *x::?'a::type*. *fl l x* = ($\exists$ *y::?'a::type*. *l* (*x*, *y*) $\vee$ *l* (*y*, *x*))

thm DEF_poset:

poset = (λ*_66091::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*. ($\forall$ *x::?'a::type*. *fl _66091 x* $\longrightarrow$ *_66091* (*x*, *x*)) $\wedge$ ($\forall$ (*x::?'a::type*) (*y::?'a::type*) *z::?'a::type*. *_66091* (*x*, *y*) $\wedge$ *_66091* (*y*, *z*) $\longrightarrow$ *_66091* (*x*, *z*)) $\wedge$ ($\forall$ (*x::?'a::type*) *y::?'a::type*. *_66091* (*x*, *y*) $\wedge$ *_66091* (*y*, *x*) $\longrightarrow$ *x* = *y*))

thm poset:

$\forall$ *l::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*. *poset l* = (($\forall$ *x::?'a::type*. *fl l x* $\longrightarrow$ *l* (*x*, *x*)) $\wedge$ ($\forall$ (*x::?'a::type*) (*y::?'a::type*) *z::?'a::type*. *l* (*x*, *y*) $\wedge$ *l* (*y*, *z*) $\longrightarrow$ *l* (*x*, *z*)) $\wedge$ ($\forall$ (*x::?'a::type*) *y::?'a::type*. *l* (*x*, *y*) $\wedge$ *l* (*y*, *x*) $\longrightarrow$ *x* = *y*))

thm DEF_chain:

chain = (λ(*_66096::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*) *_66097::?'a::type* $\Rightarrow$ *bool*. $\forall$ (*x::?'a::type*) *y::?'a::type*. *_66097 x* $\wedge$ *_66097 y* $\longrightarrow$ *_66096* (*x*, *y*) $\vee$ *_66096* (*y*, *x*))

thm chain:

$\forall (P::?'a::type \Rightarrow bool)\ l::?'a::type \times ?'a::type \Rightarrow bool.\ chain\ l\ P = (\forall (x::?'a::type)\ y::?'a::type.\ P\ x \wedge P\ y \longrightarrow l\ (x,\ y) \vee l\ (y,\ x))$

thm DEF_woset:

$woset = (\lambda\_66108::?'a::type \times ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ fl\ \_66108\ x \longrightarrow \_66108\ (x,\ x)) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ \_66108\ (x,\ y) \wedge \_66108\ (y,\ z) \longrightarrow \_66108\ (x,\ z)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ \_66108\ (x,\ y) \wedge \_66108\ (y,\ x) \longrightarrow x = y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ fl\ \_66108\ x \wedge fl\ \_66108\ y \longrightarrow \_66108\ (x,\ y) \vee \_66108\ (y,\ x)) \wedge (\forall P::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow fl\ \_66108\ x) \wedge (\exists x::?'a::type.\ P\ x) \longrightarrow (\exists y::?'a::type.\ P\ y \wedge (\forall z::?'a::type.\ P\ z \longrightarrow \_66108\ (y,\ z)))))$

thm woset:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l = ((\forall x::?'a::type.\ fl\ l\ x \longrightarrow l\ (x,\ x)) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ z) \longrightarrow l\ (x,\ z)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ x) \longrightarrow x = y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ fl\ l\ x \wedge fl\ l\ y \longrightarrow l\ (x,\ y) \vee l\ (y,\ x)) \wedge (\forall P::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow fl\ l\ x) \wedge (\exists x::?'a::type.\ P\ x) \longrightarrow (\exists y::?'a::type.\ P\ y \wedge (\forall z::?'a::type.\ P\ z \longrightarrow l\ (y,\ z)))))$

thm DEF_inseg:

$inseg = (\lambda(\_66113::?'a::type \times ?'a::type \Rightarrow bool)\ \_66114::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ \_66113\ (x,\ y) = (\_66114\ (x,\ y) \wedge fl\ \_66113\ y))$

thm inseg:

$\forall (m::?'a::type \times ?'a::type \Rightarrow bool)\ l::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ l\ m = (\forall (x::?'a::type)\ y::?'a::type.\ l\ (x,\ y) = (m\ (x,\ y) \wedge fl\ l\ y))$

thm DEF_linseg:

$linseg = (\lambda(\_66125::?'a::type \times ?'a::type \Rightarrow bool)\ \_66126::?'a::type.\ GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (\_66125\ (x,\ y) \wedge HOL\_Light\_Import.less\ \_66125\ (y,\ \_66126))))$

thm linseg:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ linseg\ l\ a = GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (l\ (x,\ y) \wedge HOL\_Light\_Import.less\ l\ (y,\ a)))$

thm DEF_ordinal:

$ordinal = (\lambda\_66137::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ \_66137 \wedge (\forall x::?'a::type.\ fl\ \_66137\ x \longrightarrow x = (SOME\ y::?'a::type.\ \neg\ HOL\_Light\_Import.less\ \_66137\ (y,\ x))))$

thm ordinal:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l = (woset\ l \wedge (\forall\, x::?'a::type.\ fl\ l\ x \longrightarrow x = (SOME\ y::?'a::type.\ \neg\ HOL\_Light\_Import.less\ l\ (y,\ x))))$

thm POSET_REFL:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ poset\ l \longrightarrow (\forall\, x::?'a::type.\ fl\ l\ x \longrightarrow l\ (x,\ x))$

thm POSET_TRANS:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ poset\ l \longrightarrow (\forall\, (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ z) \longrightarrow l\ (x,\ z))$

thm POSET_ANTISYM:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ poset\ l \longrightarrow (\forall\, (x::?'a::type)\ y::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ x) \longrightarrow x = y)$

thm POSET_FLEQ:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ poset\ l \longrightarrow (\forall\, x::?'a::type.\ fl\ l\ x = l\ (x,\ x))$

thm CHAIN_SUBSET:

$\forall\, (l::?'a::type \times ?'a::type \Rightarrow bool)\ (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ chain\ l\ P \wedge SUBSET\ Q\ P \longrightarrow chain\ l\ Q$

thm WOSET_REFL:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall\, x::?'a::type.\ fl\ l\ x \longrightarrow l\ (x,\ x))$

thm WOSET_TRANS:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall\, (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ z) \longrightarrow l\ (x,\ z))$

thm WOSET_ANTISYM:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall\, (x::?'a::type)\ y::?'a::type.\ l\ (x,\ y) \wedge l\ (y,\ x) \longrightarrow x = y)$

thm WOSET_TOTAL:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall\, (x::?'a::type)\ y::?'a::type.\ fl\ l\ x \wedge fl\ l\ y \longrightarrow l\ (x,\ y) \vee l\ (y,\ x))$

thm WOSET_WELL:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall\, P::?'a::type \Rightarrow bool.\ (\forall\, x::?'a::type.\ P\ x \longrightarrow fl\ l\ x) \wedge (\exists\, x::?'a::type.\ P\ x) \longrightarrow (\exists\, y::?'a::type.\ P\ y \wedge (\forall\, z::?'a::type.\ P\ z \longrightarrow l\ (y,\ z))))$

thm WOSET_POSET:

$\forall\, l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow poset\ l$

thm WOSET_FLEQ:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall x::?'a::type.\ fl\ l\ x = l\ (x, x))$

thm WOSET_TRANS_LESS:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ HOL\_Light\_Import.less\ l\ (x, y) \wedge l\ (y, z) \longrightarrow HOL\_Light\_Import.less\ l\ (x, z))$

thm WOSET:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l = ((\forall (x::?'a::type)\ y::?'a::type.\ l\ (x, y) \wedge l\ (y, x) \longrightarrow x = y) \wedge (\forall P::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow fl\ l\ x) \wedge (\exists x::?'a::type.\ P\ x) \longrightarrow (\exists y::?'a::type.\ P\ y \wedge (\forall z::?'a::type.\ P\ z \longrightarrow l\ (y, z)))))$

thm PAIRED_EXT:

$\forall (l::?'c::type \times ?'b::type \Rightarrow ?'a::type)\ m::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ (\forall (x::?'c::type)\ y::?'b::type.\ l\ (x, y) = m\ (x, y)) = (l = m)$

thm WOSET_TRANS_LE:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ l\ (x, y) \wedge HOL\_Light\_Import.less\ l\ (y, z) \longrightarrow HOL\_Light\_Import.less\ l\ (x, z))$

thm WOSET_WELL_CONTRAPOS:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall P::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ P\ x \longrightarrow fl\ l\ x) \wedge (\exists x::?'a::type.\ P\ x) \longrightarrow (\exists y::?'a::type.\ P\ y \wedge (\forall z::?'a::type.\ HOL\_Light\_Import.less\ l\ (z, y) \longrightarrow \neg\ P\ z)))$

thm WOSET_TOTAL_LE:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ fl\ l\ x \wedge fl\ l\ y \longrightarrow l\ (x, y) \vee HOL\_Light\_Import.less\ l\ (y, x))$

thm WOSET_TOTAL_LT:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ fl\ l\ x \wedge fl\ l\ y \longrightarrow x = y \vee HOL\_Light\_Import.less\ l\ (x, y) \vee HOL\_Light\_Import.less\ l\ (y, x))$

thm UNION_FL:

$\forall (P::(?'b::type \times ?'b::type \Rightarrow bool) \Rightarrow bool)\ l::?'a::type \times ?'a::type \Rightarrow bool.\ fl\ (UNIONS\ P)\ (?x::?'b::type) = (\exists l::?'b::type \times ?'b::type \Rightarrow bool.\ P\ l \wedge fl\ l\ ?x)$

thm UNION_INSEG:

$\forall (P::(?'a::type \times ?'a::type \Rightarrow bool) \Rightarrow bool)\ l::?'a::type \times ?'a::type \Rightarrow bool.\ (\forall m::?'a::type \times ?'a::type \Rightarrow bool.\ P\ m \longrightarrow inseg\ m\ l) \longrightarrow inseg\ (UNIONS\ P)\ l$

thm INSEG_SUBSET:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ m\ l \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ m\ (x,\ y) \longrightarrow l\ (x,\ y))$

thm INSEG_SUBSET_FL:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ m\ l \longrightarrow (\forall x::?'a::type.\ fl\ m\ x \longrightarrow fl\ l\ x)$

thm INSEG_WOSET:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ m\ l \wedge woset\ l \longrightarrow woset\ m$

thm LINSEG_INSEG:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ woset\ l \longrightarrow inseg\ (linseg\ l\ a)\ l$

thm LINSEG_WOSET:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ woset\ l \longrightarrow woset\ (linseg\ l\ a)$

thm LINSEG_FL:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ (a::?'a::type)\ x::?'a::type.\ woset\ l \longrightarrow fl\ (linseg\ l\ a)\ x = HOL\_Light\_Import.less\ l\ (x,\ a)$

thm INSEG_PROPER_SUBSET:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ m\ l \wedge l \neq m \longrightarrow (\exists (x::?'a::type)\ y::?'a::type.\ l\ (x,\ y) \wedge \neg\ m\ (x,\ y))$

thm INSEG_PROPER_SUBSET_FL:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ inseg\ m\ l \wedge l \neq m \longrightarrow (\exists a::?'a::type.\ fl\ l\ a \wedge \neg\ fl\ m\ a)$

thm INSEG_LINSEG:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow inseg\ m\ l = (m = l \vee (\exists a::?'a::type.\ fl\ l\ a \wedge m = linseg\ l\ a))$

thm EXTEND_FL:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ x::?'a::type.\ woset\ l \longrightarrow fl\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (l\ (x,\ y) \wedge l\ (y,\ ?a::?'a::type))))\ x = l\ (x,\ ?a)$

thm EXTEND_INSEG:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ woset\ l \wedge fl\ l\ a \longrightarrow inseg\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (l\ (x,\ y) \wedge l\ (y,\ a))))\ l$

thm EXTEND_LINSEG:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ woset\ l \wedge fl\ l\ a \longrightarrow inseg\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (linseg\ l\ a\ (x,\ y) \vee y = a \wedge (fl\ (linseg\ l\ a)\ x \vee x = a))))\ l$

thm ORDINAL_CHAINED_LEMMA:

$\forall (k::?'a::type \times ?'a::type \Rightarrow bool)\ (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l \wedge ordinal\ m \longrightarrow inseg\ k\ l \wedge inseg\ k\ m \longrightarrow k = l \vee k = m \vee (\exists a::?'a::type.\ fl\ l\ a \wedge fl\ m\ a \wedge k = linseg\ l\ a \wedge k = linseg\ m\ a)$

thm ORDINAL_CHAINED:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ m::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l \wedge ordinal\ m \longrightarrow inseg\ m\ l \vee inseg\ l\ m$

thm FL_SUC:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ a::?'a::type.\ fl\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (l\ (x,\ y) \vee y = a \wedge (fl\ l\ x \vee x = a))))\ (?x::?'a::type) = (fl\ l\ ?x \vee ?x = a)$

thm ORDINAL_SUC:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l \wedge (\exists x::?'a::type.\ \neg\ fl\ l\ x) \longrightarrow ordinal\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (l\ (x,\ y) \vee y = (SOME\ y::?'a::type.\ \neg\ fl\ l\ y) \wedge (fl\ l\ x \vee x = (SOME\ y::?'a::type.\ \neg\ fl\ l\ y)))))$

thm ORDINAL_UNION:

$\forall P::(?'a::type \times ?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ P\ l \longrightarrow ordinal\ l) \longrightarrow ordinal\ (UNIONS\ P)$

thm ORDINAL_UNION_LEMMA:

$\forall (l::?'a::type \times ?'a::type \Rightarrow bool)\ x::?'a::type.\ ordinal\ l \longrightarrow fl\ l\ x \longrightarrow fl\ (UNIONS\ ordinal)\ x$

thm ORDINAL_UP:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l \longrightarrow (\forall x::?'a::type.\ fl\ l\ x) \vee (\exists (m::?'a::type \times ?'a::type \Rightarrow bool)\ x::?'a::type.\ ordinal\ m \wedge fl\ m\ x \wedge \neg\ fl\ l\ x)$

thm LEMMA:

$\exists l::?'a::type \times ?'a::type \Rightarrow bool.\ ordinal\ l \wedge (\forall x::?'a::type.\ fl\ l\ x)$

thm FL_RESTRICT:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \longrightarrow (\forall P::?'a::type \Rightarrow bool.\ fl\ (GABS\ (\lambda f::?'a::type \times ?'a::type \Rightarrow bool.\ \forall (x::?'a::type)\ y::?'a::type.\ GEQ\ (f\ (x,\ y))\ (P\ x \wedge P\ y \wedge l\ (x,\ y))))\ (?x::?'a::type) = (P\ ?x \wedge fl\ l\ ?x))$

thm WO:

$\forall P::?'a::type \Rightarrow bool.\ \exists l::?'a::type \times ?'a::type \Rightarrow bool.\ woset\ l \wedge fl\ l = P$

thm HP:

$\forall l::?'a::type \times ?'a::type \Rightarrow bool.\ poset\ l \longrightarrow (\exists P::?'a::type \Rightarrow bool.\ chain\ l\ P \wedge (\forall Q::?'a::type \Rightarrow bool.\ chain\ l\ Q \wedge SUBSET\ P\ Q \longrightarrow Q = P))$

thm ZL:

$\forall\, l{::}?'a{::}type \times ?'a{::}type \Rightarrow bool.\ poset\ l \wedge (\forall\, P{::}?'a{::}type \Rightarrow bool.\ chain\ l\ P \longrightarrow (\exists\, y{::}?'a{::}type.\ fl\ l\ y \wedge (\forall\, x{::}?'a{::}type.\ P\ x \longrightarrow l\ (x,\ y)))) \longrightarrow (\exists\, y{::}?'a{::}type.\ fl\ l\ y \wedge (\forall\, x{::}?'a{::}type.\ l\ (y,\ x) \longrightarrow y = x))$

thm KL_POSET_LEMMA:

$poset\ (GABS\ (\lambda f{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow bool) \Rightarrow bool.\ \forall\, (c1{::}?'a{::}type \Rightarrow bool)\ c2{::}?'a{::}type \Rightarrow bool.\ GEQ\ (f\ (c1,\ c2))\ (SUBSET\ (?C{::}?'a{::}type \Rightarrow bool)\ c1 \wedge SUBSET\ c1\ c2 \wedge chain\ (?l{::}?'a{::}type \times ?'a{::}type \Rightarrow bool)\ c2)))$

thm KL:

$\forall\, l{::}?'a{::}type \times ?'a{::}type \Rightarrow bool.\ poset\ l \longrightarrow (\forall\, C{::}?'a{::}type \Rightarrow bool.\ chain\ l\ C \longrightarrow (\exists\, P{::}?'a{::}type \Rightarrow bool.\ (chain\ l\ P \wedge SUBSET\ C\ P) \wedge (\forall\, R{::}?'a{::}type \Rightarrow bool.\ chain\ l\ R \wedge SUBSET\ P\ R \longrightarrow R = P)))$

thm sum_CASES:

$\forall\, x{::}?'b{::}type + ?'a{::}type.\ (\exists\, a{::}?'b{::}type.\ x = Inl\ a) \vee (\exists\, a{::}?'a{::}type.\ x = Inr\ a)$

thm FORALL_SUM_THM:

$(\forall\, z{::}?'b{::}type + ?'a{::}type.\ (?P{::}?'b{::}type + ?'a{::}type \Rightarrow bool)\ z) = ((\forall\, x{::}?'b{::}type.\ ?P\ (Inl\ x)) \wedge (\forall\, x{::}?'a{::}type.\ ?P\ (Inr\ x)))$

thm EXISTS_SUM_THM:

$(\exists\, z{::}?'b{::}type + ?'a{::}type.\ (?P{::}?'b{::}type + ?'a{::}type \Rightarrow bool)\ z) = ((\exists\, x{::}?'b{::}type.\ ?P\ (Inl\ x)) \vee (\exists\, x{::}?'a{::}type.\ ?P\ (Inr\ x)))$

thm POSET_RESTRICTED_SUBSET:

$\forall\, P{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ poset\ (GABS\ (\lambda f{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow bool) \Rightarrow bool.\ \forall\, (x{::}?'a{::}type \Rightarrow bool)\ y{::}?'a{::}type \Rightarrow bool.\ GEQ\ (f\ (x,\ y))\ (P\ x \wedge P\ y \wedge SUBSET\ x\ y)))$

thm FL_RESTRICTED_SUBSET:

$\forall\, P{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ fl\ (GABS\ (\lambda f{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow bool) \Rightarrow bool.\ \forall\, (x{::}?'a{::}type \Rightarrow bool)\ y{::}?'a{::}type \Rightarrow bool.\ GEQ\ (f\ (x,\ y))\ (P\ x \wedge P\ y \wedge SUBSET\ x\ y))) = P$

thm ZL_SUBSETS:

$\forall\, P{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall\, c{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall\, x{::}?'a{::}type \Rightarrow bool.\ IN\ x\ c \longrightarrow P\ x) \wedge (\forall\, (x{::}?'a{::}type \Rightarrow bool)\ y{::}?'a{::}type \Rightarrow bool.\ IN\ x\ c \wedge IN\ y\ c \longrightarrow SUBSET\ x\ y \vee SUBSET\ y\ x) \longrightarrow (\exists\, z{::}?'a{::}type \Rightarrow bool.\ P\ z \wedge (\forall\, x{::}?'a{::}type \Rightarrow bool.\ IN\ x\ c \longrightarrow SUBSET\ x\ z))) \longrightarrow (\exists\, a{::}?'a{::}type \Rightarrow bool.\ P\ a \wedge (\forall\, x{::}?'a{::}type \Rightarrow bool.\ P\ x \wedge SUBSET\ a\ x \longrightarrow a = x))$

thm ZL_SUBSETS_UNIONS:

$\forall P::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall c::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall x::?'a::type \Rightarrow bool.\ IN\ x\ c \longrightarrow P\ x) \wedge (\forall (x::?'a::type \Rightarrow bool)\ y::?'a::type \Rightarrow bool.\ IN\ x\ c \wedge IN\ y\ c \longrightarrow SUBSET\ x\ y \vee SUBSET\ y\ x) \longrightarrow P\ (UNIONS\ c)) \longrightarrow (\exists a::?'a::type \Rightarrow bool.\ P\ a \wedge (\forall x::?'a::type \Rightarrow bool.\ P\ x \wedge SUBSET\ a\ x \longrightarrow a = x))$

thm ZL_SUBSETS_UNIONS_NONEMPTY:

$\forall P::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\exists x::?'a::type \Rightarrow bool.\ P\ x) \wedge (\forall c::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\exists x::?'a::type \Rightarrow bool.\ IN\ x\ c) \wedge (\forall x::?'a::type \Rightarrow bool.\ IN\ x\ c \longrightarrow P\ x) \wedge (\forall (x::?'a::type \Rightarrow bool)\ y::?'a::type \Rightarrow bool.\ IN\ x\ c \wedge IN\ y\ c \longrightarrow SUBSET\ x\ y \vee SUBSET\ y\ x) \longrightarrow P\ (UNIONS\ c)) \longrightarrow (\exists a::?'a::type \Rightarrow bool.\ P\ a \wedge (\forall x::?'a::type \Rightarrow bool.\ P\ x \wedge SUBSET\ a\ x \longrightarrow a = x))$

thm FLATTEN_LEMMA:

$(\forall x::?'b::type.\ IN\ x\ (?s::?'b::type \Rightarrow bool) \longrightarrow (?g::?'a::type \Rightarrow ?'b::type)\ ((?f::?'b::type \Rightarrow ?'a::type)\ x) = x) = (\forall (y::?'a::type)\ x::?'b::type.\ IN\ x\ ?s \wedge y = ?f\ x \longrightarrow ?g\ y = x)$

thm TARSKI_SET:

$\forall f::(?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (f\ s)\ (f\ t)) \longrightarrow (\exists s::?'a::type \Rightarrow bool.\ f\ s = s)$

thm INJECTIVE_LEFT_INVERSE_NONEMPTY:

$(\exists x::?'b::type.\ IN\ x\ (?s::?'b::type \Rightarrow bool)) \longrightarrow (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ ?s \wedge IN\ y\ ?s \wedge (?f::?'b::type \Rightarrow ?'a::type)\ x = ?f\ y \longrightarrow x = y) = (\exists g::?'a::type \Rightarrow ?'b::type.\ (\forall y::?'a::type.\ IN\ y\ (?t::?'a::type \Rightarrow bool) \longrightarrow IN\ (g\ y)\ ?s) \wedge (\forall x::?'b::type.\ IN\ x\ ?s \longrightarrow g\ (?f\ x) = x))$

thm BIJECTIVE_INJECTIVE_SURJECTIVE:

$((\forall x::?'b::type.\ IN\ x\ (?s::?'b::type \Rightarrow bool) \longrightarrow IN\ ((?f::?'b::type \Rightarrow ?'a::type)\ x)\ (?t::?'a::type \Rightarrow bool)) \wedge (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow (\exists! x::?'b::type.\ IN\ x\ ?s \wedge ?f\ x = y))) = ((\forall x::?'b::type.\ IN\ x\ ?s \longrightarrow IN\ (?f\ x)\ ?t) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ ?s \wedge IN\ y\ ?s \wedge ?f\ x = ?f\ y \longrightarrow x = y) \wedge (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow (\exists x::?'b::type.\ IN\ x\ ?s \wedge ?f\ x = y)))$

thm BIJECTIVE_INVERSES:

$((\forall x::?'b::type.\ IN\ x\ (?s::?'b::type \Rightarrow bool) \longrightarrow IN\ ((?f::?'b::type \Rightarrow ?'a::type)\ x)\ (?t::?'a::type \Rightarrow bool)) \wedge (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow (\exists! x::?'b::type.\ IN\ x\ ?s \wedge ?f\ x = y))) = ((\forall x::?'b::type.\ IN\ x\ ?s \longrightarrow IN\ (?f\ x)\ ?t) \wedge (\exists g::?'a::type \Rightarrow ?'b::type.\ (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow IN\ (g\ y)\ ?s) \wedge (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow ?f\ (g\ y) = y) \wedge (\forall x::?'b::type.\ IN\ x\ ?s \longrightarrow g\ (?f\ x) = x)))$

thm EQ_C_BIJECTIONS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t = (\exists (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::?'a::type.\ IN\ y\ t \longrightarrow IN\ (g\ y)\ s \wedge f\ (g\ y) = y))$

thm EQ_C:

$=\_c\ (?s::?'b::type \Rightarrow bool)\ (?t::?'a::type \Rightarrow bool) = (\exists R::?'b::type \times ?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'a::type.\ R\ (x,\ y) \longrightarrow IN\ x\ ?s \wedge IN\ y\ ?t) \wedge (\forall x::?'b::type.\ IN\ x\ ?s \longrightarrow (\exists !y::?'a::type.\ IN\ y\ ?t \wedge R\ (x,\ y))) \wedge (\forall y::?'a::type.\ IN\ y\ ?t \longrightarrow (\exists !x::?'b::type.\ IN\ x\ ?s \wedge R\ (x,\ y))))$

thm CARD_LE_REFL:

$\forall s::?'a::type \Rightarrow bool.\ <=\_c\ s\ s$

thm CARD_LE_TRANS:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \wedge <=\_c\ t\ u \longrightarrow <=\_c\ s\ u$

thm CARD_LT_REFL:

$\forall s::?'a::type \Rightarrow bool.\ \neg <\_c\ s\ s$

thm CARD_LET_TRANS:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \wedge <\_c\ t\ u \longrightarrow <\_c\ s\ u$

thm CARD_LTE_TRANS:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <\_c\ s\ t \wedge <=\_c\ t\ u \longrightarrow <\_c\ s\ u$

thm CARD_LT_TRANS:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <\_c\ s\ t \wedge <\_c\ t\ u \longrightarrow <\_c\ s\ u$

thm CARD_EQ_REFL:

$\forall s::?'a::type \Rightarrow bool.\ =\_c\ s\ s$

thm CARD_EQ_SYM:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t = =\_c\ t\ s$

thm CARD_EQ_IMP_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t \longrightarrow <=\_c\ s\ t$

thm CARD_LT_IMP_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <\_c\ s\ t \longrightarrow <=\_c\ s\ t$

thm CARD_LE_RELATIONAL:

$\forall R::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ (y::?'a::type)\ y'::?'a::type.\ IN\ x\ (?s::?'b::type \Rightarrow bool) \wedge R\ x\ y \wedge R\ x\ y' \longrightarrow y = y') \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%211::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%211\ (\exists x::?'b::type.\ IN\ x\ ?s \wedge R\ x\ y)\ y))\ ?s$

thm CARD_LE_EMPTY:

$\forall s::?'b::type \Rightarrow bool.\ <=\_c\ s\ EMPTY = (s = EMPTY)$

thm CARD_EQ_EMPTY:

$\forall s::?'b::type \Rightarrow bool.\ =\_c\ s\ EMPTY = (s = EMPTY)$

thm CARD_LE_ANTISYM:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (<=\_c\ s\ t \wedge <=\_c\ t\ s) = =\_c\ s\ t$

thm CARD_LE_TOTAL:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \vee <=\_c\ t\ s$

thm CARD_LET_TOTAL:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \vee <\_c\ t\ s$

thm CARD_LTE_TOTAL:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <\_c\ s\ t \vee <=\_c\ t\ s$

thm CARD_LT_TOTAL:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t \vee <\_c\ s\ t \vee <\_c\ t\ s$

thm CARD_NOT_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\neg <=\_c\ s\ t) = <\_c\ t\ s$

thm CARD_NOT_LT:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\neg <\_c\ s\ t) = <=\_c\ t\ s$

thm CARD_LT_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <\_c\ s\ t = (<=\_c\ s\ t \wedge \neg =\_c\ s\ t)$

thm CARD_LE_LT:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t = (<\_c\ s\ t \vee =\_c\ s\ t)$

thm CARD_LE_CONG:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ =\_c\ s\ s' \wedge =\_c\ t\ t' \longrightarrow <=\_c\ s\ t = <=\_c\ s'\ t'$

thm CARD_LT_CONG:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ =\_c\ s\ s' \wedge =\_c\ t\ t' \longrightarrow <\_c\ s\ t = <\_c\ s'\ t'$

thm CARD_EQ_TRANS:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ =\_c\ s\ t \wedge =\_c\ t\ u \longrightarrow =\_c\ s\ u$

thm CARD_EQ_CONG:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ =\_c\ s\ s' \wedge =\_c\ t\ t' \longrightarrow =\_c\ s\ t = =\_c\ s'\ t'$

thm INFINITE_CARD_LE:

$\forall s::?'a::type \Rightarrow bool.\ INFINITE\ s = <=\_c\ HOL\_Light\_Import.UNIV\ s$

thm FINITE_CARD_LT:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s = <\_c\ s\ HOL\_Light\_Import.UNIV$

thm CARD_LE_SUBSET:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow <=\_c\ s\ t$

thm CARD_LE_UNIV:

$\forall s::?'a::type \Rightarrow bool.\ <=\_c\ s\ HOL\_Light\_Import.UNIV$

thm CARD_LE_EQ_SUBSET:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t = (\exists u::?'a::type \Rightarrow bool.\ SUBSET\ u\ t \wedge =\_c\ s\ u)$

thm CARD_INFINITE_CONG:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t \longrightarrow INFINITE\ s = INFINITE\ t$

thm CARD_FINITE_CONG:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t \longrightarrow FINITE\ s = FINITE\ t$

thm CARD_LE_FINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge <=\_c\ s\ t \longrightarrow FINITE\ s$

thm CARD_EQ_FINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge =\_c\ s\ t \longrightarrow FINITE\ s$

thm CARD_LE_INFINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ s \wedge <=\_c\ s\ t \longrightarrow INFINITE\ t$

thm CARD_LT_FINITE_INFINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge INFINITE\ t \longrightarrow <\_c\ s\ t$

thm CARD_LE_CARD_IMP:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge <=\_c\ s\ t \longrightarrow CARD\ s \leq CARD\ t$

thm CARD_EQ_CARD_IMP:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ t \wedge =\_c\ s\ t \longrightarrow CARD\ s = CARD\ t$

thm CARD_LE_CARD:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow <=\_c\ s\ t = (CARD\ s \leq CARD\ t)$

thm CARD_EQ_CARD:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow =\_c\ s\ t = (CARD\ s = CARD\ t)$

thm CARD_LT_CARD:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow <\_c\ s\ t = (CARD\ s < CARD\ t)$

thm CARD_HAS_SIZE_CONG:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ (t{::}?'a{::}type \Rightarrow bool)\ n{::}nat.\ HAS\_SIZE\ s\ n \wedge =\_c\ s\ t \longrightarrow HAS\_SIZE\ t\ n$

thm CARD_LE_IMAGE:

$\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ s{::}?'b{::}type \Rightarrow bool.\ <=\_c\ (IMAGE\ f\ s)\ s$

thm CARD_LE_IMAGE_GEN:

$\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ SUBSET\ t\ (IMAGE\ f\ s) \longrightarrow <=\_c\ t\ s$

thm CARD_EQ_IMAGE:

$\forall (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ s{::}?'b{::}type \Rightarrow bool.\ (\forall (x{::}?'b{::}type)\ y{::}?'b{::}type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow =\_c\ (IMAGE\ f\ s)\ s$

thm DEF_+_c:

$+\_c = (\lambda(\_76592{::}?'b{::}type \Rightarrow bool)\ \_76593{::}?'a{::}type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (GSPEC\ (\lambda GEN\%PVAR\%214{::}?'b{::}type + ?'a{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%214\ (IN\ x\ \_76592)\ (Inl\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%215{::}?'b{::}type + ?'a{::}type.\ \exists y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%215\ (IN\ y\ \_76593)\ (Inr\ y))))$

thm add_c:

$\forall (s{::}?'b{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ +\_c\ s\ t = HOL\_Light\_Import.UNION\ (GSPEC\ (\lambda GEN\%PVAR\%214{::}?'b{::}type + ?'a{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%214\ (IN\ x\ s)\ (Inl\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%215{::}?'b{::}type + ?'a{::}type.\ \exists y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%215\ (IN\ y\ t)\ (Inr\ y)))$

thm DEF_*_c:

$*\_c = (\lambda(\_76604{::}?'b{::}type \Rightarrow bool)\ \_76605{::}?'a{::}type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%216{::}?'b{::}type \times ?'a{::}type.\ \exists (x{::}?'b{::}type)\ y{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%216\ (IN\ x\ \_76604 \wedge IN\ y\ \_76605)\ (x,\ y)))$

thm mul_c:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ *\_c\ s\ t = GSPEC\ (\lambda GEN\%PVAR\%216::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%216\ (IN\ x\ s \wedge IN\ y\ t)\ (x,\ y))$

thm CARD_LE_ADD:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ <=\_c\ s\ s' \wedge <=\_c\ t\ t' \longrightarrow <=\_c\ (+\_c\ s\ t)\ (+\_c\ s'\ t')$

thm CARD_LE_MUL:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ <=\_c\ s\ s' \wedge <=\_c\ t\ t' \longrightarrow <=\_c\ (*\_c\ s\ t)\ (*\_c\ s'\ t')$

thm CARD_FUNSPACE_LE:

$<=\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV \wedge <=\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV \longrightarrow <=\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV$

thm CARD_ADD_CONG:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ =\_c\ s\ s' \wedge =\_c\ t\ t' \longrightarrow =\_c\ (+\_c\ s\ t)\ (+\_c\ s'\ t')$

thm CARD_MUL_CONG:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ =\_c\ s\ s' \wedge =\_c\ t\ t' \longrightarrow =\_c\ (*\_c\ s\ t)\ (*\_c\ s'\ t')$

thm CARD_FUNSPACE_CONG:

$=\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV \wedge =\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV \longrightarrow =\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV$

thm MUL_C_UNIV:

$*\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm CARD_FUNSPACE_CURRY:

$=\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV$

thm IN_CARD_ADD:

$(\forall x::?'b::type.\ IN\ (Inl\ x)\ (+\_c\ (?s::?'b::type \Rightarrow bool)\ (?t::?'a::type \Rightarrow bool)) = IN\ x\ ?s) \wedge (\forall y::?'a::type.\ IN\ (Inr\ y)\ (+\_c\ ?s\ ?t) = IN\ y\ ?t)$

thm IN_CARD_MUL:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (x::?'b::type)\ y::?'a::type.\ IN\ (x,\ y)\ (*\_c\ s\ t) = (IN\ x\ s \wedge IN\ y\ t)$

thm CARD_LE_SQUARE:

$\forall s::?'a::type \Rightarrow bool.\ <=\_c\ s\ (*\_c\ s\ s)$

thm CARD_SQUARE_NUM:

=_c (∗_c HOL_Light_Import.UNIV HOL_Light_Import.UNIV) HOL_Light_Import.UNIV

thm UNION_LE_ADD_C:

∀(s::?'a::type ⇒ bool) t::?'a::type ⇒ bool. <=_c (HOL_Light_Import.UNION s t) (+_c s t)

thm CARD_ADD_C:

∀(s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. FINITE s ∧ FINITE t ⟶ CARD (+_c s t) = CARD s + CARD t

thm IN_CARD_ADD_conjunct1:

∀y::?'b::type. IN (Inr y) (+_c (?s::?'a::type ⇒ bool) (?t::?'b::type ⇒ bool)) = IN y ?t

thm IN_CARD_ADD_conjunct0:

∀x::?'b::type. IN (Inl x) (+_c (?s::?'b::type ⇒ bool) (?t::?'a::type ⇒ bool)) = IN x ?s

thm CARD_ADD_SYM:

∀(s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. =_c (+_c s t) (+_c t s)

thm CARD_ADD_ASSOC:

∀(s::?'c::type ⇒ bool) (t::?'b::type ⇒ bool) u::?'a::type ⇒ bool. =_c (+_c s (+_c t u)) (+_c (+_c s t) u)

thm CARD_MUL_SYM:

∀(s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. =_c (∗_c s t) (∗_c t s)

thm CARD_MUL_ASSOC:

∀(s::?'c::type ⇒ bool) (t::?'b::type ⇒ bool) u::?'a::type ⇒ bool. =_c (∗_c s (∗_c t u)) (∗_c (∗_c s t) u)

thm CARD_LDISTRIB:

∀(s::?'c::type ⇒ bool) (t::?'b::type ⇒ bool) u::?'a::type ⇒ bool. =_c (∗_c s (+_c t u)) (+_c (∗_c s t) (∗_c s u))

thm CARD_RDISTRIB:

∀(s::?'c::type ⇒ bool) (t::?'b::type ⇒ bool) u::?'a::type ⇒ bool. =_c (∗_c (+_c s t) u) (+_c (∗_c s u) (∗_c t u))

thm CARD_LE_ADDR:

∀(s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. <=_c s (+_c s t)

thm CARD_LE_ADDL:

∀(s::?'b::type ⇒ bool) t::?'a::type ⇒ bool. <=_c t (+_c s t)

thm CARD_ADD_LE_MUL_INFINITE:

$\forall s::?'a::type \Rightarrow bool.\ INFINITE\ s \longrightarrow <=\_c\ (+\_c\ s\ s)\ (*\_c\ s\ s)$

thm CARD_DISJOINT_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ t = EMPTY \longrightarrow =\_c\ (HOL\_Light\_Import.UNION\ s\ t)\ (+\_c\ s\ t)$

thm CARD_SQUARE_INFINITE:

$\forall k::?'a::type \Rightarrow bool.\ INFINITE\ k \longrightarrow =\_c\ (*\_c\ k\ k)\ k$

thm CARD_ADD_FINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (+\_c\ s\ t)$

thm CARD_ADD_FINITE_EQ:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ (+\_c\ s\ t) = (FINITE\ s \wedge FINITE\ t)$

thm CARD_MUL_FINITE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (*\_c\ s\ t)$

thm CARD_MUL_ABSORB_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ t \wedge <=\_c\ s\ t \longrightarrow <=\_c\ (*\_c\ s\ t)\ t$

thm CARD_MUL2_ABSORB_LE:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ INFINITE\ u \wedge <=\_c\ s\ u \wedge <=\_c\ t\ u \longrightarrow <=\_c\ (*\_c\ s\ t)\ u$

thm CARD_ADD_ABSORB_LE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ t \wedge <=\_c\ s\ t \longrightarrow <=\_c\ (+\_c\ s\ t)\ t$

thm CARD_ADD2_ABSORB_LE:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ INFINITE\ u \wedge <=\_c\ s\ u \wedge <=\_c\ t\ u \longrightarrow <=\_c\ (+\_c\ s\ t)\ u$

thm CARD_MUL_ABSORB:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ t \wedge s \neq EMPTY \wedge <=\_c\ s\ t \longrightarrow =\_c\ (*\_c\ s\ t)\ t$

thm CARD_ADD_ABSORB:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ INFINITE\ t \wedge <=\_c\ s\ t \longrightarrow =\_c\ (+\_c\ s\ t)\ t$

thm CARD_ADD2_ABSORB_LT:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ INFINITE\ u \wedge <\_c\ s\ u \wedge <\_c\ t\ u \longrightarrow <\_c\ (+\_c\ s\ t)\ u$

thm CARD_LT_ADD:

$\forall (s::?'d::type \Rightarrow bool)\ (s'::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ t'::?'a::type \Rightarrow bool.\ <\_c\ s\ s' \wedge <\_c\ t\ t' \longrightarrow <\_c\ (+\_c\ s\ t)\ (+\_c\ s'\ t')$

thm CARD_MUL_LT_LEMMA:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \wedge <\_c\ t\ u \wedge INFINITE\ u \longrightarrow <\_c\ (*\_c\ s\ t)\ u$

thm CARD_MUL_LT_INFINITE:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ <\_c\ s\ u \wedge <\_c\ t\ u \wedge INFINITE\ u \longrightarrow <\_c\ (*\_c\ s\ t)\ u$

thm CANTOR_THM:

$\forall s::?'a::type \Rightarrow bool.\ <\_c\ s\ (GSPEC\ (\lambda GEN\%PVAR\%223::?'a::type \Rightarrow bool.\ \exists t::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%223\ (SUBSET\ t\ s)\ t))$

thm CANTOR_THM_UNIV:

$<\_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV$

thm NUM_COUNTABLE:

$COUNTABLE\ HOL\_Light\_Import.UNIV$

thm COUNTABLE_ALT:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s = <=\_c\ s\ HOL\_Light\_Import.UNIV$

thm COUNTABLE_CASES:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s = (FINITE\ s \vee =\_c\ s\ HOL\_Light\_Import.UNIV)$

thm CARD_LE_COUNTABLE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ t \wedge <=\_c\ s\ t \longrightarrow COUNTABLE\ s$

thm CARD_EQ_COUNTABLE:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ t \wedge =\_c\ s\ t \longrightarrow COUNTABLE\ s$

thm CARD_COUNTABLE_CONG:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ =\_c\ s\ t \longrightarrow COUNTABLE\ s = COUNTABLE\ t$

thm COUNTABLE_SUBSET:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ t \wedge SUBSET\ s\ t \longrightarrow COUNTABLE\ s$

thm COUNTABLE_RESTRICT:

$\forall (s::?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow bool.\ COUNTABLE\ s \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%224::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%224\ (IN\ x\ s \wedge P\ x)\ x))$

thm FINITE_IMP_COUNTABLE:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow COUNTABLE\ s$

thm COUNTABLE_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ COUNTABLE\ s \longrightarrow COUNTABLE\ (IMAGE\ f\ s)$

thm COUNTABLE_IMAGE_INJ_GENERAL:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (A::?'a::type \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge COUNTABLE\ A \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%225::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%225\ (IN\ x\ s \wedge IN\ (f\ x)\ A)\ x))$

thm COUNTABLE_IMAGE_INJ_EQ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow COUNTABLE\ (IMAGE\ f\ s) = COUNTABLE\ s$

thm COUNTABLE_IMAGE_INJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ A::?'a::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ f\ x = f\ y \longrightarrow x = y) \wedge COUNTABLE\ A \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%226::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%226\ (IN\ (f\ x)\ A)\ x))$

thm COUNTABLE_EMPTY:

$COUNTABLE\ EMPTY$

thm COUNTABLE_INTER:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ s \vee COUNTABLE\ t \longrightarrow COUNTABLE\ (HOL\_Light\_Import.INTER\ s\ t)$

thm COUNTABLE_UNION_IMP:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ s \wedge COUNTABLE\ t \longrightarrow COUNTABLE\ (HOL\_Light\_Import.UNION\ s\ t)$

thm COUNTABLE_UNION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ (HOL\_Light\_Import.UNION\ s\ t) = (COUNTABLE\ s \wedge COUNTABLE\ t)$

thm COUNTABLE_SING:

$\forall x::?'a::type.\ COUNTABLE\ (INSERT\ x\ EMPTY)$

thm COUNTABLE_INSERT:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ COUNTABLE\ (INSERT\ x\ s) = COUNTABLE\ s$

thm COUNTABLE_DELETE:

$\forall (x::?'a::type)\ s::?'a::type \Rightarrow bool.\ COUNTABLE\ (DELETE\ s\ x) = COUNTABLE\ s$

thm COUNTABLE_DIFF_FINITE:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow COUNTABLE\ (DIFF\ t\ s) = COUNTABLE\ t$

thm COUNTABLE_CROSS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ COUNTABLE\ s \wedge COUNTABLE\ t \longrightarrow COUNTABLE\ (CROSS\ s\ t)$

thm COUNTABLE_AS_IMAGE_SUBSET:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s \longrightarrow (\exists f::nat \Rightarrow ?'a::type.\ SUBSET\ s\ (IMAGE\ f\ HOL\_Light\_Import.UNIV))$

thm COUNTABLE_AS_IMAGE_SUBSET_EQ:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s = (\exists f::nat \Rightarrow ?'a::type.\ SUBSET\ s\ (IMAGE\ f\ HOL\_Light\_Import.UNIV))$

thm COUNTABLE_AS_IMAGE:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s \wedge s \neq EMPTY \longrightarrow (\exists f::nat \Rightarrow ?'a::type.\ s = IMAGE\ f\ HOL\_Light\_Import.UNIV)$

thm FORALL_COUNTABLE_AS_IMAGE:

$(\forall d::?'a::type \Rightarrow bool.\ COUNTABLE\ d \longrightarrow (?P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ d) = (?P\ EMPTY \wedge (\forall f::nat \Rightarrow ?'a::type.\ ?P\ (IMAGE\ f\ HOL\_Light\_Import.UNIV)))$

thm COUNTABLE_AS_INJECTIVE_IMAGE:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s \wedge INFINITE\ s \longrightarrow (\exists f::nat \Rightarrow ?'a::type.\ s = IMAGE\ f\ HOL\_Light\_Import.UNIV \wedge (\forall (m::nat)\ n::nat.\ f\ m = f\ n \longrightarrow m = n))$

thm COUNTABLE_UNIONS:

$\forall A::(?'a::type \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ A \wedge (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ A \longrightarrow COUNTABLE\ s) \longrightarrow COUNTABLE\ (UNIONS\ A)$

thm COUNTABLE_PRODUCT_DEPENDENT:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ t::?'c::type \Rightarrow ?'b::type \Rightarrow bool.\ COUNTABLE\ s \wedge (\forall x::?'c::type.\ IN\ x\ s \longrightarrow COUNTABLE$

$(t\,x)) \longrightarrow$ *COUNTABLE* (*GSPEC* (λ*GEN%PVAR%229*::?'*a*::*type*. $\exists$ (*x*::?'*c*::*type*) *y*::?'*b*::*type*. *SETSPEC GEN%PVAR%229* (*IN x s* $\wedge$ *IN y* (*t x*)) (*f x y*)))

thm COUNTABLE_CART:

$\forall$ *P*::*nat* $\Rightarrow$?'*b*::*type* $\Rightarrow$ *bool*. ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *COUNTABLE* (*GSPEC* (λ*GEN%PVAR%235*::?'*b*::*type*. $\exists$ *x*::?'*b*::*type*. *SET-SPEC GEN%PVAR%235* (*P i x*) *x*))) $\longrightarrow$ *COUNTABLE* (*GSPEC* (λ*GEN%PVAR%236*::(?'*b*::*type*, ?'*a*::*type*) *cart*. $\exists$ *v*::(?'*b*::*type*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%236* ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *P i* ($ *v* *i*)) *v*))

thm CARD_EQ_LIST_GEN:

$\forall$ *s*::?'*a*::*type* $\Rightarrow$ *bool*. *INFINITE s* $\longrightarrow$ =_*c* (*GSPEC* (λ*GEN%PVAR%237*::?'*a*::*type list*. $\exists$ *l*::?'*a*::*type list*. *SETSPEC GEN%PVAR%237* ($\forall$ *x*::?'*a*::*type*. *MEM x l* $\longrightarrow$ *IN x s*) *l*)) *s*

thm CARD_EQ_LIST:

INFINITE HOL_Light_Import.UNIV $\longrightarrow$ =_*c HOL_Light_Import.UNIV HOL_Light_Import.UNIV*

thm CARD_EQ_CART:

INFINITE HOL_Light_Import.UNIV $\longrightarrow$ =_*c HOL_Light_Import.UNIV HOL_Light_Import.UNIV*

thm LE_1_conjunct5:

$\forall$ *n*$\geq$*1*::*nat*. *n* $\neq$ (*0*::*nat*)

thm LE_1_conjunct4:

$\forall$ *n*$\geq$*1*::*nat*. (*0*::*nat*) < *n*

thm LE_1_conjunct3:

$\forall$ *n*>*0*::*nat*. (*1*::*nat*) $\leq$ *n*

thm LE_1_conjunct2:

$\forall$ *n*>*0*::*nat*. *n* $\neq$ (*0*::*nat*)

thm LE_1_conjunct1:

$\forall$ *n*::*nat*. *n* $\neq$ (*0*::*nat*) $\longrightarrow$ (*1*::*nat*) $\leq$ *n*

thm CARD_EQ_REAL:

=_*c HOL_Light_Import.UNIV HOL_Light_Import.UNIV*

thm UNCOUNTABLE_REAL:

$\neg$ *COUNTABLE HOL_Light_Import.UNIV*

thm CARD_EQ_REAL_IMP_UNCOUNTABLE:

$\forall$ *s*::?'*a*::*type* $\Rightarrow$ *bool*. =_*c s HOL_Light_Import.UNIV* $\longrightarrow$ $\neg$ *COUNTABLE s*

thm CARD_EQ_FINITE_SUBSETS:

$\forall s::?'a::type \Rightarrow bool.\ INFINITE\ s \longrightarrow =\_c\ (GSPEC\ (\lambda GEN\%PVAR\%243::?'a::type \Rightarrow bool.\ \exists t::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%243\ (SUBSET\ t\ s \wedge FINITE\ t)\ t))\ s$

thm CARD_LE_LIST:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%244::?'b::type\ list.\ \exists l::?'b::type\ list.\ SETSPEC\ GEN\%PVAR\%244\ (\forall x::?'b::type.\ MEM\ x\ l \longrightarrow IN\ x\ s)\ l))\ (GSPEC\ (\lambda GEN\%PVAR\%245::?'a::type\ list.\ \exists l::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%245\ (\forall x::?'a::type.\ MEM\ x\ l \longrightarrow IN\ x\ t)\ l))$

thm CARD_LE_SUBPOWERSET:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \wedge (\forall (f::?'b::type \Rightarrow ?'a::type)\ s::?'b::type \Rightarrow bool.\ (?P::(?'b::type \Rightarrow bool) \Rightarrow bool)\ s \longrightarrow (?Q::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (IMAGE\ f\ s)) \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%246::?'b::type \Rightarrow bool.\ \exists u::?'b::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%246\ (SUBSET\ u\ s \wedge ?P\ u)\ u))\ (GSPEC\ (\lambda GEN\%PVAR\%247::?'a::type \Rightarrow bool.\ \exists v::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%247\ (SUBSET\ v\ t \wedge ?Q\ v)\ v))$

thm CARD_LE_FINITE_SUBSETS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%248::?'b::type \Rightarrow bool.\ \exists u::?'b::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%248\ (SUBSET\ u\ s \wedge FINITE\ u)\ u))\ (GSPEC\ (\lambda GEN\%PVAR\%249::?'a::type \Rightarrow bool.\ \exists v::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%249\ (SUBSET\ v\ t \wedge FINITE\ v)\ v))$

thm CARD_LE_COUNTABLE_SUBSETS:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%250::?'b::type \Rightarrow bool.\ \exists u::?'b::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%250\ (SUBSET\ u\ s \wedge COUNTABLE\ u)\ u))\ (GSPEC\ (\lambda GEN\%PVAR\%251::?'a::type \Rightarrow bool.\ \exists v::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%251\ (SUBSET\ v\ t \wedge COUNTABLE\ v)\ v))$

thm CARD_LE_POWERSET:

$\forall (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ <=\_c\ s\ t \longrightarrow <=\_c\ (GSPEC\ (\lambda GEN\%PVAR\%254::?'b::type \Rightarrow bool.\ \exists u::?'b::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%254\ (SUBSET\ u\ s)\ u))\ (GSPEC\ (\lambda GEN\%PVAR\%255::?'a::type \Rightarrow bool.\ \exists v::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%255\ (SUBSET\ v\ t)\ v))$

thm COUNTABLE_LIST_GEN:

$\forall s::?'a::type \Rightarrow bool.\ COUNTABLE\ s \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%257::?'a::type\ list.\ \exists l::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%257\ (\forall x::?'a::type.\ MEM\ x\ l \longrightarrow IN\ x\ s)\ l))$

thm COUNTABLE_LIST:

$COUNTABLE\ HOL\_Light\_Import.UNIV \longrightarrow COUNTABLE\ HOL\_Light\_Import.UNIV$

thm COUNTABLE_FINITE_SUBSETS:

$\forall s{::}?'a{::}type \Rightarrow bool.\ COUNTABLE\ s \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%260{::}?'a{::}type \Rightarrow bool.\ \exists t{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%260\ (SUBSET\ t\ s \wedge FINITE\ t)\ t))$

thm CARD_EQ_REAL_SEQUENCES:

$=_c\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV$

thm CARD_EQ_COUNTABLE_SUBSETS_REAL:

$=_c\ (GSPEC\ (\lambda GEN\%PVAR\%262{::}real \Rightarrow bool.\ \exists s{::}real \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%262\ (COUNTABLE\ s)\ s))\ HOL\_Light\_Import.UNIV$

thm EXISTS_DIFF:

$(\exists s{::}?'a{::}type \Rightarrow bool.\ (?P{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool)\ (DIFF\ HOL\_Light\_Import.UNIV\ s)) = (\exists s{::}?'a{::}type \Rightarrow bool.\ ?P\ s)$

thm GE_REFL:

$\forall n{::}nat.\ n \leq n$

thm FORALL_SUC:

$(\forall n{::}nat.\ n \neq (0{::}nat) \longrightarrow (?P{::}nat \Rightarrow bool)\ n) = (\forall n{::}nat.\ ?P\ (Suc\ n))$

thm SEQ_MONO_LEMMA:

$\forall (d{::}nat \Rightarrow real)\ e{::}nat \Rightarrow real.\ (\forall n{\geq}?m{::}nat.\ d\ n < e\ n) \wedge (\forall n{\geq}?m.\ e\ n \leq e\ ?m) \longrightarrow (\forall n{\geq}?m.\ d\ n < e\ ?m)$

thm REAL_HALF:

$(\forall e{::}real.\ ((0{::}real) < e\ /\ real\_of\_nat\ (2{::}nat)) = ((0{::}real) < e)) \wedge (\forall e{::}real.\ e\ /\ real\_of\_nat\ (2{::}nat) + e\ /\ real\_of\_nat\ (2{::}nat) = e) \wedge (\forall e{::}real.\ real\_of\_nat\ (2{::}nat) * (e\ /\ real\_of\_nat\ (2{::}nat)) = e)$

thm UPPER_BOUND_FINITE_SET:

$\forall (f{::}?'a{::}type \Rightarrow nat)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists a{::}nat.\ \forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow f\ x \leq a)$

thm UPPER_BOUND_FINITE_SET_REAL:

$\forall (f{::}?'a{::}type \Rightarrow real)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists a{::}real.\ \forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow f\ x \leq a)$

thm LOWER_BOUND_FINITE_SET:

$\forall (f{::}?'a{::}type \Rightarrow nat)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists a{::}nat.\ \forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow a \leq f\ x)$

thm LOWER_BOUND_FINITE_SET_REAL:

$\forall (f{::}?'a{::}type \Rightarrow real)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists a{::}real.\ \forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow a \leq f\ x)$

thm REAL_CONVEX_BOUND2_LT:

$\forall (x::real)\ (y::real)\ (a::real)\ (u::real)\ v::real.\ x < a \wedge y < (?b::real) \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow u * x + v * y < u * a + v * ?b$

thm REAL_CONVEX_BOUND_LT:

$\forall (x::real)\ (y::real)\ (a::real)\ (u::real)\ v::real.\ x < a \wedge y < a \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow u * x + v * y < a$

thm REAL_CONVEX_BOUND_LE:

$\forall (x::real)\ (y::real)\ (a::real)\ (u::real)\ v::real.\ x \leq a \wedge y \leq a \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow u * x + v * y \leq a$

thm INFINITE_ENUMERATE:

$\forall s::nat \Rightarrow bool.\ INFINITE\ s \longrightarrow (\exists r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge (\forall n::nat.\ IN\ (r\ n)\ s))$

thm APPROACHABLE_LT_LE:

$\forall (P::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ (\exists d>0::real.\ \forall x::?'a::type.\ f\ x < d \longrightarrow P\ x) = (\exists d>0::real.\ \forall x::?'a::type.\ f\ x \leq d \longrightarrow P\ x)$

thm REAL_LE_BETWEEN:

$\forall (a::real)\ b::real.\ (a \leq b) = (\exists x \geq a.\ x \leq b)$

thm REAL_LT_BETWEEN:

$\forall (a::real)\ b::real.\ (a < b) = (\exists x > a.\ x < b)$

thm TRIANGLE_LEMMA:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge (0::real) \leq z \wedge x^2 \leq y^2 + z^2 \longrightarrow x \leq y + z$

thm LAMBDA_SKOLEM:

$(\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists x::?'a::type.\ (?P::nat \Rightarrow ?'a::type \Rightarrow bool)\ i\ x)) = (\exists x::(?'a::type,\ ?'b::type)\ cart.\ \forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow ?P\ i\ (\$\ x\ i))$

thm LAMBDA_PAIR:

$GABS\ (\lambda f::?'c::type \times ?'b::type \Rightarrow ?'a::type.\ \forall (x::?'c::type)\ y::?'b::type.\ GEQ\ (f\ (x,\ y))\ ((?P::?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ x\ y)) = (\lambda p::?'c::type \times ?'b::type.\ ?P\ (fst\ p)\ (snd\ p))$

thm EPSILON_DELTA_MINIMAL:

$\forall (P::real \Rightarrow ?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%267::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%267\ (Q\ x)\ x)) \wedge (\forall (d::real)\ (e::real)\ x::?'a::type.\ Q\ x \wedge (0::real) < e \wedge e < d \longrightarrow P\ d\ x \longrightarrow P\ e\ x) \wedge (\forall x::?'a::type.\ Q\ x \longrightarrow (\exists d>0::real.\ P\ d\ x)) \longrightarrow (\exists d>0::real.\ \forall x::?'a::type.\ Q\ x \longrightarrow P\ d\ x)$

thm DEF_hull:

$hull = (\lambda(\_105971::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_105972::?'a::type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%268::?'a::type \Rightarrow bool.\ \exists t::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%268\ (\_105971\ t \wedge SUBSET\ \_105972\ t)\ t)))$

thm hull:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ hull\ P\ s = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%268::?'a::type \Rightarrow bool.\ \exists t::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%268\ (P\ t \wedge SUBSET\ s\ t)\ t))$

thm HULL_P:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ P\ s \longrightarrow hull\ P\ s = s$

thm P_HULL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow P\ s) \longrightarrow P\ (INTERS\ f)) \longrightarrow P\ (hull\ P\ s)$

thm HULL_EQ:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow P\ s) \longrightarrow P\ (INTERS\ f)) \longrightarrow (hull\ P\ s = s) = P\ s$

thm HULL_HULL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ hull\ P\ (hull\ P\ s) = hull\ P\ s$

thm HULL_SUBSET:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ SUBSET\ s\ (hull\ P\ s)$

thm HULL_MONO:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (hull\ P\ s)\ (hull\ P\ t)$

thm HULL_ANTIMONO:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (Q::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ SUBSET\ P\ Q \longrightarrow SUBSET\ (hull\ Q\ s)\ (hull\ P\ s)$

thm HULL_MINIMAL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \wedge P\ t \longrightarrow SUBSET\ (hull\ P\ s)\ t$

thm SUBSET_HULL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ P\ t \longrightarrow SUBSET\ (hull\ P\ s)\ t = SUBSET\ s\ t$

thm HULL_UNIQUE:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ t \wedge P\ t \wedge (\forall t'::?'a::type \Rightarrow bool.\ SUBSET\ s\ t' \wedge P\ t' \longrightarrow SUBSET\ t\ t') \longrightarrow hull\ P\ s = t$

thm SUBSET_UNION_conjunct1:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (HOL\_Light\_Import.UNION\ t\ s)$

thm SUBSET_UNION_conjunct0:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (HOL\_Light\_Import.UNION\ s\ t)$

thm HULL_UNION_SUBSET:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.UNION\ (hull\ P\ s)\ (hull\ P\ t))\ (hull\ P\ (HOL\_Light\_Import.UNION\ s\ t))$

thm HULL_UNION:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ hull\ P\ (HOL\_Light\_Import.UNION\ s\ t) = hull\ P\ (HOL\_Light\_Import.UNION\ (hull\ P\ s)\ (hull\ P\ t))$

thm HULL_UNION_LEFT:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ hull\ P\ (HOL\_Light\_Import.UNION\ s\ t) = hull\ P\ (HOL\_Light\_Import.UNION\ (hull\ P\ s)\ t)$

thm HULL_UNION_RIGHT:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ hull\ P\ (HOL\_Light\_Import.UNION\ s\ t) = hull\ P\ (HOL\_Light\_Import.UNION\ s\ (hull\ P\ t))$

thm HULL_REDUNDANT_EQ:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ a\ (hull\ P\ s) = (hull\ P\ (INSERT\ a\ s) = hull\ P\ s)$

thm HULL_REDUNDANT:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ a\ (hull\ P\ s) \longrightarrow hull\ P\ (INSERT\ a\ s) = hull\ P\ s$

thm HULL_INDUCT:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (p::?'a::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow p\ x) \wedge P\ (GSPEC\ (\lambda GEN\%PVAR\%270::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%270\ (p\ x)\ x)) \longrightarrow (\forall x::?'a::type.\ IN\ x\ (hull\ P\ s) \longrightarrow p\ x)$

thm HULL_INC:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ s \longrightarrow IN\ x\ (hull\ P\ s)$

thm HULL_IMAGE_SUBSET:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ P\ (hull\ P\ s) \wedge (\forall s::?'a::type \Rightarrow bool.\ P\ s \longrightarrow P\ (IMAGE\ f\ s)) \longrightarrow SUBSET\ (hull\ P\ (IMAGE\ f\ s))\ (IMAGE\ f\ (hull\ P\ s))$

thm HULL_IMAGE_GALOIS:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ P\ (hull\ P\ s)) \wedge (\forall s::?'a::type \Rightarrow bool.\ P\ s \longrightarrow P\ (IMAGE\ f\ s)) \wedge (\forall s::?'a::type \Rightarrow bool.\ P\ s \longrightarrow P\ (IMAGE\ g\ s)) \wedge (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ SUBSET\ s\ (IMAGE\ g\ t) = SUBSET\ (IMAGE\ f\ s)\ t) \longrightarrow hull\ P\ (IMAGE\ f\ s) = IMAGE\ f\ (hull\ P\ s)$

thm HULL_IMAGE:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ P\ (hull\ P\ s)) \wedge (\forall s::?'a::type \Rightarrow bool.\ P\ (IMAGE\ f\ s) = P\ s) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::?'a::type.\ \exists x::?'a::type.\ f\ x = y) \longrightarrow hull\ P\ (IMAGE\ f\ s) = IMAGE\ f\ (hull\ P\ s)$

thm IS_HULL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow P\ s) \longrightarrow P\ (INTERS\ f)) \longrightarrow P\ s = (\exists t::?'a::type \Rightarrow bool.\ s = hull\ P\ t)$

thm HULLS_EQ:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow P\ s) \longrightarrow P\ (INTERS\ f)) \wedge SUBSET\ s\ (hull\ P\ t) \wedge SUBSET\ t\ (hull\ P\ s) \longrightarrow hull\ P\ s = hull\ P\ t$

thm HULL_P_AND_Q:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ Q::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow P\ s) \longrightarrow P\ (INTERS\ f)) \wedge (\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ f \longrightarrow Q\ s) \longrightarrow Q\ (INTERS\ f)) \wedge (\forall s::?'a::type \Rightarrow bool.\ Q\ s \longrightarrow Q\ (hull\ P\ s)) \longrightarrow hull\ (\lambda x::?'a::type \Rightarrow bool.\ P\ x \wedge Q\ x)\ (?s::?'a::type \Rightarrow bool) = hull\ P\ (hull\ Q\ ?s)$

thm REAL_ARCH_INV:

$\forall e::real.\ ((0::real) < e) = (\exists n::nat.\ n \neq (0::nat) \wedge (0::real) < inverse\ (real\_of\_nat\ n) \wedge inverse\ (real\_of\_nat\ n) < e)$

thm REAL_POW_LBOUND:

$\forall (x::real)\ n::nat.\ (0::real) \leq x \longrightarrow (1::real) + real\_of\_nat\ n * x \leq ((1::real) + x)^n$

thm REAL_ARCH_POW:

$\forall (x::real)\ y::real.\ (1::real) < x \longrightarrow (\exists\, n::nat.\ y < x^n)$

thm REAL_ARCH_POW2:

$\forall\, x::real.\ \exists\, n::nat.\ x < (real\_of\_nat\ (2::nat))^n$

thm REAL_ARCH_POW_INV:

$\forall (x::real)\ y::real.\ (0::real) < y \wedge x < (1::real) \longrightarrow (\exists\, n::nat.\ x^n < y)$

thm FORALL_POS_MONO:

$\forall\, P::real \Rightarrow bool.\ (\forall (d::real)\ e::real.\ d < e \wedge P\ d \longrightarrow P\ e) \wedge (\forall\, n::nat.\ n \neq (0::nat) \longrightarrow P\ (inverse\ (real\_of\_nat\ n))) \longrightarrow (\forall\, e{>}0::real.\ P\ e)$

thm FORALL_POS_MONO_1:

$\forall\, P::real \Rightarrow bool.\ (\forall (d::real)\ e::real.\ d < e \wedge P\ d \longrightarrow P\ e) \wedge (\forall\, n::nat.\ P\ (inverse\ (real\_of\_nat\ n + (1::real)))) \longrightarrow (\forall\, e{>}0::real.\ P\ e)$

thm REAL_ARCH_RDIV_EQ_0:

$\forall (x::real)\ c::real.\ (0::real) \le x \wedge (0::real) \le c \wedge (\forall\, m{>}0::nat.\ real\_of\_nat\ m * x \le c) \longrightarrow x = (0::real)$

thm REAL_MAX_SUP:

$\forall (x::real)\ y::real.\ max\ x\ y = HOL\_Light\_Import.sup\ (INSERT\ x\ (INSERT\ y\ EMPTY))$

thm REAL_MIN_INF:

$\forall (x::real)\ y::real.\ min\ x\ y = HOL\_Light\_Import.inf\ (INSERT\ x\ (INSERT\ y\ EMPTY))$

thm DEF_sqrt:

$sqrt = (\lambda\_106843::real.\ SOME\ y::real.\ (0::real) \le y \wedge y^2 = \_106843)$

thm sqrt:

$\forall\, x::real.\ sqrt\ x = (SOME\ y::real.\ (0::real) \le y \wedge y^2 = x)$

thm SQRT_UNIQUE:

$\forall (x::real)\ y::real.\ (0::real) \le y \wedge y^2 = x \longrightarrow sqrt\ x = y$

thm POW_2_SQRT:

$\forall\, x{\ge}0::real.\ sqrt\ (x^2) = x$

thm SQRT_0:

$sqrt\ (0::real) = (0::real)$

thm SQRT_1:

$sqrt\ (1::real) = (1::real)$

thm POW_2_SQRT_ABS:

$\forall\, x{::}real.\ sqrt\ (x^2) = |x|$

thm SUM_GP_BASIC:

$\forall\, (x{::}real)\ n{::}nat.\ ((1{::}real) - x) * sum\ (dotdot\ (0{::}nat)\ n)\ (op\ \hat{}\ x) = (1{::}real) - x^{Suc\ n}$

thm SUM_GP_MULTIPLIED:

$\forall\, (x{::}real)\ (m{::}nat)\ n{::}nat.\ m \leq n \longrightarrow ((1{::}real) - x) * sum\ (dotdot\ m\ n)\ (op\ \hat{}\ x) = x^m - x^{Suc\ n}$

thm SUM_GP:

$\forall\, (x{::}real)\ (m{::}nat)\ n{::}nat.\ sum\ (dotdot\ m\ n)\ (op\ \hat{}\ x) = (if\ n < m\ then\ 0{::}real\ else\ if\ x = (1{::}real)\ then\ real\_of\_nat\ (n + (1{::}nat) - m)\ else\ (x^m - x^{Suc\ n})\ /\ ((1{::}real) - x))$

thm SUM_GP_OFFSET:

$\forall\, (x{::}real)\ (m{::}nat)\ n{::}nat.\ sum\ (dotdot\ m\ (m + n))\ (op\ \hat{}\ x) = (if\ x = (1{::}real)\ then\ real\_of\_nat\ n + (1{::}real)\ else\ x^m * (((1{::}real) - x^{Suc\ n})\ /\ ((1{::}real) - x)))$

thm FORALL_1:

$(\forall\, i{::}nat.\ (1{::}nat) \leq i \wedge i \leq (1{::}nat) \longrightarrow (?P{::}nat \Rightarrow bool)\ i) = ?P\ (1{::}nat)$

thm FORALL_2:

$\forall\, P{::}nat \Rightarrow bool.\ (\forall\, i{::}nat.\ (1{::}nat) \leq i \wedge i \leq (2{::}nat) \longrightarrow P\ i) = (P\ (1{::}nat) \wedge P\ (2{::}nat))$

thm FORALL_3:

$\forall\, P{::}nat \Rightarrow bool.\ (\forall\, i{::}nat.\ (1{::}nat) \leq i \wedge i \leq (3{::}nat) \longrightarrow P\ i) = (P\ (1{::}nat) \wedge P\ (2{::}nat) \wedge P\ (3{::}nat))$

thm SUM_1:

$sum\ (dotdot\ (1{::}nat)\ (1{::}nat))\ (?f{::}nat \Rightarrow real) = ?f\ (1{::}nat)$

thm SUM_2:

$\forall\, t{::}nat \Rightarrow real.\ sum\ (dotdot\ (1{::}nat)\ (2{::}nat))\ t = t\ (1{::}nat) + t\ (2{::}nat)$

thm Hypermap.THREE:

$(3{::}nat) = Suc\ (2{::}nat)$

thm SUM_3:

$\forall\, t{::}nat \Rightarrow real.\ sum\ (dotdot\ (1{::}nat)\ (3{::}nat))\ t = t\ (1{::}nat) + (t\ (2{::}nat) + t\ (3{::}nat))$

thm DEF_vector_add:

$vector\_add = (\lambda(\_107096::(real, ?'a::type)\ cart)\ \_107097::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ \$\ \_107096\ i + \$\ \_107097\ i))$

thm vector_add:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ vector\_add\ x\ y = lambda\ (\lambda i::nat.\ \$\ x\ i + \$\ y\ i)$

thm DEF_vector_sub:

$vector\_sub = (\lambda(\_107108::(real, ?'a::type)\ cart)\ \_107109::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ \$\ \_107108\ i - \$\ \_107109\ i))$

thm vector_sub:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ vector\_sub\ x\ y = lambda\ (\lambda i::nat.\ \$\ x\ i - \$\ y\ i)$

thm DEF_vector_neg:

$vector\_neg = (\lambda\_107120::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ - \$\ \_107120\ i))$

thm vector_neg:

$\forall x::(real, ?'a::type)\ cart.\ vector\_neg\ x = lambda\ (\lambda i::nat.\ - \$\ x\ i)$

thm DEF_%:

$\% = (\lambda(\_107125::real)\ \_107126::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ \_107125 * \$\ \_107126\ i))$

thm vector_mul:

$\forall (c::real)\ x::(real, ?'a::type)\ cart.\ \%\ c\ x = lambda\ (\lambda i::nat.\ c * \$\ x\ i)$

thm DEF_vec:

$vec = (\lambda\_107137::nat.\ lambda\ (\lambda i::nat.\ real\_of\_nat\ \_107137))$

thm vec:

$\forall n::nat.\ vec\ n = lambda\ (\lambda i::nat.\ real\_of\_nat\ n)$

thm DEF_dot:

$dot = (\lambda(\_107142::(real, ?'a::type)\ cart)\ \_107143::(real, ?'a::type)\ cart.\ sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ \_107142\ i * \$\ \_107143\ i))$

thm dot:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ dot\ x\ y = sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ x\ i * \$\ y\ i)$

thm DOT_1:

$dot\ (?x::(real, unit)\ cart)\ (?y::(real, unit)\ cart) = \$\ ?x\ (1::nat) * \$\ ?y\ (1::nat)$

thm DOT_2:

dot (*?x*::(*real*, *2*) *cart*) (*?y*::(*real*, *2*) *cart*) = \$ *?x* (*1*::*nat*) ∗ \$ *?y* (*1*::*nat*) + \$ *?x* (*2*::*nat*) ∗ \$ *?y* (*2*::*nat*)

thm DOT_3:

dot (*?x*::(*real*, *3*) *cart*) (*?y*::(*real*, *3*) *cart*) = \$ *?x* (*1*::*nat*) ∗ \$ *?y* (*1*::*nat*) + (\$ *?x* (*2*::*nat*) ∗ \$ *?y* (*2*::*nat*) + \$ *?x* (*3*::*nat*) ∗ \$ *?y* (*3*::*nat*))

thm VEC_COMPONENT:

∀ (*k*::*nat*) *i*::*nat*. \$ (*vec k*) *i* = *real_of_nat k*

thm VECTOR_ADD_COMPONENT:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *i*::*nat*. \$ (*vector_add x y*) *i* = \$ *x i* + \$ *y i*

thm VECTOR_SUB_COMPONENT:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *i*::*nat*. \$ (*vector_sub x y*) *i* = \$ *x i* − \$ *y i*

thm VECTOR_NEG_COMPONENT:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *i*::*nat*. \$ (*vector_neg x*) *i* = − \$ *x i*

thm VECTOR_MUL_COMPONENT:

∀ (*c*::*real*) (*x*::(*real*, *?'a*::*type*) *cart*) *i*::*nat*. \$ (% *c x*) *i* = *c* ∗ \$ *x i*

thm COND_COMPONENT:

\$ (*if ?b*::*bool then ?x*::(*?'b*::*type*, *?'a*::*type*) *cart else* (*?y*::(*?'b*::*type*, *?'a*::*type*) *cart*)) (*?i*::*nat*) = (*if ?b then* \$ *?x ?i else* \$ *?y ?i*)

thm VECTOR_ADD_SYM:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *vector_add x y* = *vector_add y x*

thm VECTOR_ADD_LID:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *vector_add* (*vec* (*0*::*nat*)) *x* = *x*

thm VECTOR_ADD_RID:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *vector_add x* (*vec* (*0*::*nat*)) = *x*

thm VECTOR_SUB_REFL:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *vector_sub x x* = *vec* (*0*::*nat*)

thm VECTOR_ADD_LINV:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *vector_add* (*vector_neg x*) *x* = *vec* (*0*::*nat*)

thm VECTOR_ADD_RINV:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_add\ x\ (vector\_neg\ x) = vec\ (0::nat)$

thm VECTOR_SUB_RADD:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_sub\ x\ (vector\_add\ x\ y) = vector\_neg\ y$

thm VECTOR_NEG_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_neg\ (vector\_sub\ x\ y) = vector\_sub\ y\ x$

thm VECTOR_SUB_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (vector\_sub\ x\ y = vec\ (0::nat)) = (x = y)$

thm VECTOR_MUL_ASSOC:

$\forall (a::real)\ (b::real)\ x::(real,\ ?'a::type)\ cart.\ \%\ a\ (\%\ b\ x) = \%\ (a * b)\ x$

thm VECTOR_MUL_LID:

$\forall x::(real,\ ?'a::type)\ cart.\ \%\ (1::real)\ x = x$

thm VECTOR_MUL_LZERO:

$\forall x::(real,\ ?'a::type)\ cart.\ \%\ (0::real)\ x = vec\ (0::nat)$

thm VECTOR_SUB_ADD:

$vector\_add\ (vector\_sub\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart))\ ?y = ?x$

thm VECTOR_SUB_ADD2:

$vector\_add\ (?y::(real,\ ?'a::type)\ cart)\ (vector\_sub\ (?x::(real,\ ?'a::type)\ cart)\ ?y) = ?x$

thm VECTOR_ADD_LDISTRIB:

$\%\ (?c::real)\ (vector\_add\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)) = vector\_add\ (\%\ ?c\ ?x)\ (\%\ ?c\ ?y)$

thm VECTOR_SUB_LDISTRIB:

$\%\ (?c::real)\ (vector\_sub\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)) = vector\_sub\ (\%\ ?c\ ?x)\ (\%\ ?c\ ?y)$

thm VECTOR_ADD_RDISTRIB:

$\%\ ((?a::real) + (?b::real))\ (?x::(real,\ ?'a::type)\ cart) = vector\_add\ (\%\ ?a\ ?x)\ (\%\ ?b\ ?x)$

thm VECTOR_SUB_RDISTRIB:

$\%\ ((?a::real) - (?b::real))\ (?x::(real,\ ?'a::type)\ cart) = vector\_sub\ (\%\ ?a\ ?x)\ (\%\ ?b\ ?x)$

thm VECTOR_ADD_SUB:

vector_sub (vector_add (?x::(real, ?'a::type) cart) (?y::(real, ?'a::type) cart)) ?x = ?y

thm VECTOR_EQ_ADDR:

(vector_add (?x::(real, ?'a::type) cart) (?y::(real, ?'a::type) cart) = ?x) = (?y = vec (0::nat))

thm VECTOR_SUB:

vector_sub (?x::(real, ?'a::type) cart) (?y::(real, ?'a::type) cart) = vector_add ?x (vector_neg ?y)

thm VECTOR_SUB_RZERO:

vector_sub (?x::(real, ?'a::type) cart) (vec (0::nat)) = ?x

thm VECTOR_MUL_RZERO:

% *(?c::real) (vec (0::nat)) = vec (0::nat)*

thm VECTOR_NEG_MINUS1:

vector_neg (?x::(real, ?'a::type) cart) = % *(− (1::real)) ?x*

thm VECTOR_ADD_ASSOC:

vector_add (?x::(real, ?'a::type) cart) (vector_add (?y::(real, ?'a::type) cart) (?z::(real, ?'a::type) cart)) = vector_add (vector_add ?x ?y) ?z

thm VECTOR_SUB_LZERO:

vector_sub (vec (0::nat)) (?x::(real, ?'a::type) cart) = vector_neg ?x

thm VECTOR_NEG_NEG:

vector_neg (vector_neg (?x::(real, ?'a::type) cart)) = ?x

thm VECTOR_MUL_LNEG:

% *(− (?c::real)) (?x::(real, ?'a::type) cart) = vector_neg (% ?c ?x)*

thm VECTOR_MUL_RNEG:

% *(?c::real) (vector_neg (?x::(real, ?'a::type) cart)) = vector_neg (% ?c ?x)*

thm VECTOR_NEG_0:

vector_neg (vec (0::nat)) = vec (0::nat)

thm VECTOR_NEG_EQ_0:

(vector_neg (?x::(real, ?'a::type) cart) = vec (0::nat)) = (?x = vec (0::nat))

thm VECTOR_ADD_AC_conjunct2:

vector_add (?m::(real, ?'a::type) cart) (vector_add (?n::(real, ?'a::type) cart) (?p::(real, ?'a::type) cart)) = vector_add ?n (vector_add ?m ?p)

thm VECTOR_ADD_AC_conjunct1:

vector_add (vector_add (?m::(real, ?'a::type) cart) (?n::(real, ?'a::type) cart)) (?p::(real, ?'a::type) cart) = vector_add ?m (vector_add ?n ?p)

thm VECTOR_ADD_AC_conjunct0:

vector_add (?m::(real, ?'a::type) cart) (?n::(real, ?'a::type) cart) = vector_add ?n ?m

thm VECTOR_ADD_AC:

vector_add (?m::(real, ?'a::type) cart) (?n::(real, ?'a::type) cart) = vector_add ?n ?m ∧ vector_add (vector_add ?m ?n) (?p::(real, ?'a::type) cart) = vector_add ?m (vector_add ?n ?p) ∧ vector_add ?m (vector_add ?n ?p) = vector_add ?n (vector_add ?m ?p)

thm VEC_EQ:

∀ *(m::nat) n::nat. (vec m = vec n) = (m = n)*

thm EUCLIDEAN_SPACE_INFINITE:

INFINITE HOL_Light_Import.UNIV

thm DOT_SYM:

∀ *(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. dot x y = dot y x*

thm DOT_LADD:

∀ *(x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. dot (vector_add x y) z = dot x z + dot y z*

thm DOT_RADD:

∀ *(x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. dot x (vector_add y z) = dot x y + dot x z*

thm DOT_LSUB:

∀ *(x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. dot (vector_sub x y) z = dot x z − dot y z*

thm DOT_RSUB:

∀ *(x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. dot x (vector_sub y z) = dot x y − dot x z*

thm DOT_LMUL:

∀ *(c::real) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. dot (% c x) y = c ∗ dot x y*

thm DOT_RMUL:

∀ *(c::real) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. dot x (% c y) = c ∗ dot x y*

thm DOT_LNEG:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ (vector\_neg\ x)\ y = -\ dot\ x\ y$

thm DOT_RNEG:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ x\ (vector\_neg\ y) = -\ dot\ x\ y$

thm DOT_LZERO:

$\forall x::(real,\ ?'a::type)\ cart.\ dot\ (vec\ (0::nat))\ x = (0::real)$

thm DOT_RZERO:

$\forall x::(real,\ ?'a::type)\ cart.\ dot\ x\ (vec\ (0::nat)) = (0::real)$

thm DOT_POS_LE:

$\forall x::(real,\ ?'a::type)\ cart.\ (0::real) \leq dot\ x\ x$

thm DOT_EQ_0:

$\forall x::(real,\ ?'a::type)\ cart.\ (dot\ x\ x = (0::real)) = (x = vec\ (0::nat))$

thm DOT_POS_LT:

$\forall x::(real,\ ?'a::type)\ cart.\ ((0::real) < dot\ x\ x) = (x \neq vec\ (0::nat))$

thm FORALL_DOT_EQ_0:

$(\forall y::(real,\ ?'b::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ dot\ x\ y = (0::real)) = (y = vec\ (0::nat))) \land (\forall x::(real,\ ?'a::type)\ cart.\ (\forall y::(real,\ ?'a::type)\ cart.\ dot\ x\ y = (0::real)) = (x = vec\ (0::nat)))$

thm DEF_vector_norm:

$vector\_norm = (\lambda\_107604::(real,\ ?'a::type)\ cart.\ sqrt\ (dot\ \_107604\ \_107604))$

thm vector_norm:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ x = sqrt\ (dot\ x\ x)$

thm FORALL_DIMINDEX_1:

$(\forall i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (?P::nat \Rightarrow bool)\ i) = ?P\ (1::nat)$

thm VECTOR_ONE:

$\forall x::(real,\ unit)\ cart.\ x = lambda\ (\lambda i::nat.\ \$\ x\ (1::nat))$

thm FORALL_REAL_ONE:

$(\forall x::(real,\ unit)\ cart.\ (?P::(real,\ unit)\ cart \Rightarrow bool)\ x) = (\forall x::real.\ ?P\ (lambda\ (\lambda i::nat.\ x)))$

thm NORM_REAL:

$\forall x$::(*real*, *unit*) *cart*. *vector_norm* x = |\$ x (*1*::*nat*)|

thm DEF_distance:

distance = (λ_*107660*::(*real*, *?'a*::*type*) *cart* $\times$ (*real*, *?'a*::*type*) *cart*. *vector_norm* (*vector_sub* (*fst* _*107660*) (*snd* _*107660*)))

thm dist:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. *distance* (x, y) = *vector_norm* (*vector_sub* x y)

thm DIST_REAL:

$\forall$ (x::(*real*, *unit*) *cart*) y::(*real*, *unit*) *cart*. *distance* (x, y) = |\$ x (*1*::*nat*) $-$ \$ y (*1*::*nat*)|

thm CONNECTED_REAL_LEMMA:

$\forall$ (f::*real* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (a::*real*) (b::*real*) (*e1*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *e2*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $a \leq b \wedge$ *IN* (f a) *e1* $\wedge$ *IN* (f b) *e2* $\wedge$ ($\forall$ (e::*real*) x::*real*. $a \leq x \wedge x \leq b \wedge$ (*0*::*real*) $< e \longrightarrow$ ($\exists$ d>*0*::*real*. $\forall$ y::*real*. $|y - x| < d \longrightarrow$ *distance* (f y, f x) $< e$)) $\wedge$ ($\forall$ y::(*real*, *?'a*::*type*) *cart*. *IN* y *e1* $\longrightarrow$ ($\exists$ e>*0*::*real*. $\forall$ y'::(*real*, *?'a*::*type*) *cart*. *distance* (y', y) $< e \longrightarrow$ *IN* y' *e1*)) $\wedge$ ($\forall$ y::(*real*, *?'a*::*type*) *cart*. *IN* y *e2* $\longrightarrow$ ($\exists$ e>*0*::*real*. $\forall$ y'::(*real*, *?'a*::*type*) *cart*. *distance* (y', y) $< e \longrightarrow$ *IN* y' *e2*)) $\wedge \neg$ ($\exists$ $x \geq a$. $x \leq b \wedge$ *IN* (f x) *e1* $\wedge$ *IN* (f x) *e2*) $\longrightarrow$ ($\exists$ $x \geq a$. $x \leq b \wedge \neg$ *IN* (f x) *e1* $\wedge \neg$ *IN* (f x) *e2*)

thm SQUARE_BOUND_LEMMA:

$\forall$ x::*real*. $x <$ ((*1*::*real*) + x) $*$ ((*1*::*real*) + x)

thm SQUARE_CONTINUOUS:

$\forall$ (x::*real*) e::*real*. (*0*::*real*) $< e \longrightarrow$ ($\exists$ d>*0*::*real*. $\forall$ y::*real*. $|y - x| < d \longrightarrow |y * y - x * x| < e$)

thm SQRT_WORKS:

$\forall$ $x \geq$*0*::*real*. (*0*::*real*) $\leq$ *sqrt* x $\wedge$ (*sqrt* x)^2 = x

thm SQRT_POS_LE:

$\forall$ $x \geq$*0*::*real*. (*0*::*real*) $\leq$ *sqrt* x

thm SQRT_POW_2:

$\forall$ $x \geq$*0*::*real*. (*sqrt* x)^2 = x

thm SQRT_MUL:

$\forall$ (x::*real*) y::*real*. (*0*::*real*) $\leq x \wedge$ (*0*::*real*) $\leq y \longrightarrow$ *sqrt* ($x * y$) = *sqrt* x $*$ *sqrt* y

thm SQRT_INV:

$\forall$ $x \geq$*0*::*real*. *sqrt* (*inverse* x) = *inverse* (*sqrt* x)

thm SQRT_DIV:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow sqrt\ (x\ /\ y) = sqrt\ x\ /\ sqrt\ y$

thm SQRT_POW2:

$\forall x::real.\ ((sqrt\ x)^2 = x) = ((0::real) \leq x)$

thm SQRT_MONO_LT:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge x < y \longrightarrow sqrt\ x < sqrt\ y$

thm SQRT_MONO_LE:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge x \leq y \longrightarrow sqrt\ x \leq sqrt\ y$

thm SQRT_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (sqrt\ x < sqrt\ y) = (x < y)$

thm SQRT_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (sqrt\ x \leq sqrt\ y) = (x \leq y)$

thm SQRT_INJ:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (sqrt\ x = sqrt\ y) = (x = y)$

thm SQRT_LT_0:

$\forall x \geq 0::real.\ ((0::real) < sqrt\ x) = ((0::real) < x)$

thm SQRT_EQ_0:

$\forall x \geq 0::real.\ (sqrt\ x = (0::real)) = (x = (0::real))$

thm SQRT_POS_LT:

$\forall x > 0::real.\ (0::real) < sqrt\ x$

thm REAL_LE_LSQRT:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x \leq y^2 \longrightarrow sqrt\ x \leq y$

thm REAL_LE_RSQRT:

$\forall (x::real)\ y::real.\ x^2 \leq y \longrightarrow x \leq sqrt\ y$

thm REAL_LT_RSQRT:

$\forall (x::real)\ y::real.\ x^2 < y \longrightarrow x < sqrt\ y$

thm SQRT_EVEN_POW2:

$\forall n::nat.\ even\ n \longrightarrow sqrt\ (real\_of\_nat\ (2::nat))^n = (real\_of\_nat\ (2::nat))^{n\ div\ (2::nat)}$

thm REAL_DIV_SQRT:

$\forall x \geq 0::real.\ x\ /\ sqrt\ x = sqrt\ x$

thm REAL_RSQRT_LE:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x \leq sqrt\ y \longrightarrow x^2 \leq y$

thm REAL_LSQRT_LE:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge sqrt\ x \leq y \longrightarrow x \leq y^2$

thm NORM_0:

$vector\_norm\ (vec\ (0::nat)) = (0::real)$

thm NORM_POS_LE:

$\forall x::(real,\ ?'a::type)\ cart.\ (0::real) \leq vector\_norm\ x$

thm NORM_NEG:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ (vector\_neg\ x) = vector\_norm\ x$

thm NORM_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ (vector\_sub\ x\ y) = vector\_norm\ (vector\_sub\ y\ x)$

thm NORM_MUL:

$\forall (a::real)\ x::(real,\ ?'a::type)\ cart.\ vector\_norm\ (\%\ a\ x) = |a| * vector\_norm\ x$

thm NORM_EQ_0_DOT:

$\forall x::(real,\ ?'a::type)\ cart.\ (vector\_norm\ x = (0::real)) = (dot\ x\ x = (0::real))$

thm NORM_EQ_0:

$\forall x::(real,\ ?'a::type)\ cart.\ (vector\_norm\ x = (0::real)) = (x = vec\ (0::nat))$

thm NORM_POS_LT:

$\forall x::(real,\ ?'a::type)\ cart.\ ((0::real) < vector\_norm\ x) = (x \neq vec\ (0::nat))$

thm NORM_POW_2:

$\forall x::(real,\ ?'a::type)\ cart.\ (vector\_norm\ x)^2 = dot\ x\ x$

thm NORM_EQ_0_IMP:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ x = (0::real) \longrightarrow x = vec\ (0::nat)$

thm NORM_LE_0:

$\forall x::(real,\ ?'a::type)\ cart.\ (vector\_norm\ x \leq (0::real)) = (x = vec\ (0::nat))$

thm VECTOR_MUL_EQ_0:

$\forall (a::real)\ x::(real,\ ?'a::type)\ cart.\ (\%\ a\ x = vec\ (0::nat)) = (a = (0::real) \lor x = vec\ (0::nat))$

thm VECTOR_MUL_LCANCEL:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (\%\ a\ x = \%\ a\ y) = (a = (0::real) \lor x = y)$

thm VECTOR_MUL_RCANCEL:

$\forall (a::real)\ (b::real)\ x::(real,\ ?'a::type)\ cart.\ (\%\ a\ x = \%\ b\ x) = (a = b \lor x = vec\ (0::nat))$

thm VECTOR_MUL_LCANCEL_IMP:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \land \%\ a\ x = \%\ a\ y \longrightarrow x = y$

thm VECTOR_MUL_RCANCEL_IMP:

$\forall (a::real)\ (b::real)\ x::(real,\ ?'a::type)\ cart.\ x \neq vec\ (0::nat) \land \%\ a\ x = \%\ b\ x \longrightarrow a = b$

thm NORM_CAUCHY_SCHWARZ:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ x\ y \leq vector\_norm\ x * vector\_norm\ y$

thm NORM_CAUCHY_SCHWARZ_ABS:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ |dot\ x\ y| \leq vector\_norm\ x * vector\_norm\ y$

thm REAL_ABS_NORM:

$\forall x::(real,\ ?'a::type)\ cart.\ |vector\_norm\ x| = vector\_norm\ x$

thm NORM_CAUCHY_SCHWARZ_DIV:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ |dot\ x\ y\ /\ (vector\_norm\ x * vector\_norm\ y)| \leq (1::real)$

thm NORM_TRIANGLE:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ (vector\_add\ x\ y) \leq vector\_norm\ x + vector\_norm\ y$

thm NORM_TRIANGLE_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ x \leq vector\_norm\ y + vector\_norm\ (vector\_sub\ x\ y)$

thm NORM_TRIANGLE_LE:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ x + vector\_norm\ y \leq (?e::real) \longrightarrow vector\_norm\ (vector\_add\ x\ y) \leq ?e$

thm NORM_TRIANGLE_LT:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ vector\_norm\ x + vector\_norm\ y < (?e::real) \longrightarrow vector\_norm\ (vector\_add\ x\ y) < ?e$

thm COMPONENT_LE_NORM:

$\forall (x::(real, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow |\$\ x\ i| \leq vector\_norm\ x$

thm NORM_BOUND_COMPONENT_LE:

$\forall (x::(real, ?'a::type)\ cart)\ e::real.\ vector\_norm\ x \leq e \longrightarrow (\forall i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow |\$\ x\ i| \leq e)$

thm NORM_BOUND_COMPONENT_LT:

$\forall (x::(real, ?'a::type)\ cart)\ e::real.\ vector\_norm\ x < e \longrightarrow (\forall i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow |\$\ x\ i| < e)$

thm NORM_LE_L1:

$\forall x::(real, ?'a::type)\ cart.\ vector\_norm\ x \leq sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ |\$\ x\ i|)$

thm REAL_ABS_SUB_NORM:

$|vector\_norm\ (?x::(real, ?'a::type)\ cart) - vector\_norm\ (?y::(real, ?'a::type)\ cart)| \leq vector\_norm\ (vector\_sub\ ?x\ ?y)$

thm NORM_LE:

$\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (vector\_norm\ x \leq vector\_norm\ y) = (dot\ x\ x \leq dot\ y\ y)$

thm NORM_LT:

$\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (vector\_norm\ x < vector\_norm\ y) = (dot\ x\ x < dot\ y\ y)$

thm NORM_EQ:

$\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (vector\_norm\ x = vector\_norm\ y) = (dot\ x\ x = dot\ y\ y)$

thm NORM_EQ_1:

$\forall x::(real, ?'a::type)\ cart.\ (vector\_norm\ x = (1::real)) = (dot\ x\ x = (1::real))$

thm NORM_LE_COMPONENTWISE:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow |\$\ x\ i| \leq |\$\ y\ i|) \longrightarrow vector\_norm\ x \leq vector\_norm\ y$

thm DOT_SQUARE_NORM:

$\forall x::(real, ?'a::type)\ cart.\ dot\ x\ x = (vector\_norm\ x)^2$

thm NORM_EQ_SQUARE:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. (*vector_norm* x = (*?a*::*real*)) = ((*0*::*real*) ≤ *?a* ∧ *dot* x x = *?a*^{2})

thm NORM_LE_SQUARE:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. (*vector_norm* x ≤ (*?a*::*real*)) = ((*0*::*real*) ≤ *?a* ∧ *dot* x x ≤ *?a*^{2})

thm NORM_GE_SQUARE:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. ((*?a*::*real*) ≤ *vector_norm* x) = (*?a* ≤ (*0*::*real*) ∨ *?a*^{2} ≤ *dot* x x)

thm NORM_LT_SQUARE:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. (*vector_norm* x < (*?a*::*real*)) = ((*0*::*real*) < *?a* ∧ *dot* x x < *?a*^{2})

thm NORM_GT_SQUARE:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. ((*?a*::*real*) < *vector_norm* x) = (*?a* < (*0*::*real*) ∨ *?a*^{2} < *dot* x x)

thm NORM_LT_SQUARE_ALT:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. (*vector_norm* x < (*?a*::*real*)) = ((*0*::*real*) ≤ *?a* ∧ *dot* x x < *?a*^{2})

thm DOT_NORM:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. *dot* x y = ((*vector_norm* (*vector_add* x y))^2 − (*vector_norm* x)^2 − (*vector_norm* y)^2) / *real_of_nat* (*2*::*nat*)

thm DOT_NORM_NEG:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. *dot* x y = ((*vector_norm* x)^2 + (*vector_norm* y)^2 − (*vector_norm* (*vector_sub* x y))^2) / *real_of_nat* (*2*::*nat*)

thm DOT_NORM_SUB:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. *dot* x y = ((*vector_norm* x)^2 + (*vector_norm* y)^2 − (*vector_norm* (*vector_sub* x y))^2) / *real_of_nat* (*2*::*nat*)

thm VECTOR_EQ:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. (x = y) = (*dot* x x = *dot* x y ∧ *dot* y y = *dot* x x)

thm DIST_REFL:

$\forall\, x$::(*real*, *?'a*::*type*) *cart*. *distance* (x, x) = (*0*::*real*)

thm DIST_SYM:

$\forall$ (x::(*real*, *?'a*::*type*) *cart*) y::(*real*, *?'a*::*type*) *cart*. *distance* (x, y) = *distance* (y, x)

thm DIST_POS_LE:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (0::real) \leq distance\ (x, y)$

thm DIST_TRIANGLE:

$\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ distance\ (x, z) \leq distance\ (x, y) + distance\ (y, z)$

thm DIST_TRIANGLE_ALT:

$\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ distance\ (y, z) \leq distance\ (x, y) + distance\ (x, z)$

thm DIST_EQ_0:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (distance\ (x, y) = (0::real)) = (x = y)$

thm DIST_POS_LT:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ x \neq y \longrightarrow (0::real) < distance\ (x, y)$

thm DIST_NZ:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (x \neq y) = ((0::real) < distance\ (x, y))$

thm DIST_TRIANGLE_LE:

$\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ (z::(real, ?'a::type)\ cart)\ e::real.\ distance\ (x, z) + distance\ (y, z) \leq e \longrightarrow distance\ (x, y) \leq e$

thm DIST_TRIANGLE_LT:

$\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ (z::(real, ?'a::type)\ cart)\ e::real.\ distance\ (x, z) + distance\ (y, z) < e \longrightarrow distance\ (x, y) < e$

thm DIST_TRIANGLE_HALF_L:

$\forall (x1::(real, ?'a::type)\ cart)\ (x2::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ distance\ (x1, y) < (?e::real)\ /\ real\_of\_nat\ (2::nat) \wedge distance\ (x2, y) < ?e\ /\ real\_of\_nat\ (2::nat) \longrightarrow distance\ (x1, x2) < ?e$

thm DIST_TRIANGLE_HALF_R:

$\forall (x1::(real, ?'a::type)\ cart)\ (x2::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ distance\ (y, x1) < (?e::real)\ /\ real\_of\_nat\ (2::nat) \wedge distance\ (y, x2) < ?e\ /\ real\_of\_nat\ (2::nat) \longrightarrow distance\ (x1, x2) < ?e$

thm DIST_TRIANGLE_ADD:

$\forall (x::(real, ?'a::type)\ cart)\ (x'::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ y'::(real, ?'a::type)\ cart.\ distance\ (vector\_add\ x\ y, vector\_add\ x'\ y') \leq distance\ (x, x') + distance\ (y, y')$

thm DIST_MUL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ c::real.\ distance\ (\%\ c\ x,\ \%\ c\ y) = |c| * distance\ (x,\ y)$

thm DIST_TRIANGLE_ADD_HALF:

$\forall (x::(real,\ ?'a::type)\ cart)\ (x'::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ y'::(real,\ ?'a::type)\ cart.\ distance\ (x,\ x') < (?e::real)\ /\ real\_of\_nat\ (2::nat) \land distance\ (y,\ y') < ?e\ /\ real\_of\_nat\ (2::nat) \longrightarrow distance\ (vector\_add\ x\ y,\ vector\_add\ x'\ y') < ?e$

thm DIST_LE_0:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (distance\ (x,\ y) \leq (0::real)) = (x = y)$

thm DIST_EQ:

$\forall (w::(real,\ ?'b::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ (distance\ (w,\ x) = distance\ (y,\ z)) = ((distance\ (w,\ x))^2 = (distance\ (y,\ z))^2)$

thm DIST_0:

$\forall x::(real,\ ?'a::type)\ cart.\ distance\ (x,\ vec\ (0::nat)) = vector\_norm\ x \land distance\ (vec\ (0::nat),\ x) = vector\_norm\ x$

thm NEUTRAL_VECTOR_ADD:

$neutral\ vector\_add = vec\ (0::nat)$

thm MONOIDAL_VECTOR_ADD:

$monoidal\ vector\_add$

thm DEF_vsum:

$vsum = (\lambda(\_109748::?'b::type \Rightarrow bool)\ \_109749::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ sum\ \_109748\ (\lambda x::?'b::type.\ \$\ (\_109749\ x)\ i)))$

thm vsum:

$\forall (s::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ vsum\ s\ f = lambda\ (\lambda i::nat.\ sum\ s\ (\lambda x::?'b::type.\ \$\ (f\ x)\ i))$

thm VSUM_CLAUSES:

$(\forall f::?'d::type \Rightarrow (real,\ ?'c::type)\ cart.\ vsum\ EMPTY\ f = vec\ (0::nat)) \land (\forall (x::?'b::type)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ vsum\ s\ f\ else\ vector\_add\ (f\ x)\ (vsum\ s\ f)))$

thm VSUM_CLAUSES_conjunct1:

$\forall (x::?'b::type)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ vsum\ s\ f\ else\ vector\_add\ (f\ x)\ (vsum\ s\ f))$

thm VSUM_CLAUSES_conjunct0:

$\forall f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ vsum\ EMPTY\ f = vec\ (0::nat)$

thm VSUM:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ f = iterate\ vector\_add\ s\ f$

thm VSUM_EQ_0:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ s\ f = vec\ (0::nat)$

thm VSUM_0:

$vsum\ (?s::?'a::type \Rightarrow bool)\ (\lambda x::?'a::type.\ vec\ (0::nat)) = vec\ (0::nat)$

thm VSUM_LMUL:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (c::real)\ s::?'b::type \Rightarrow bool.\ vsum\ s\ (\lambda x::?'b::type.\ \%\ c\ (f\ x)) = \%\ c\ (vsum\ s\ f)$

thm VSUM_RMUL:

$\forall (c::?'b::type \Rightarrow real)\ (s::?'b::type \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ vsum\ s\ (\lambda x::?'b::type.\ \%\ (c\ x)\ v) = \%\ (sum\ s\ c)\ v$

thm VSUM_ADD:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ vector\_add\ (f\ x)\ (g\ x)) = vector\_add\ (vsum\ s\ f)\ (vsum\ s\ g)$

thm VSUM_SUB:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ vector\_sub\ (f\ x)\ (g\ x)) = vector\_sub\ (vsum\ s\ f)\ (vsum\ s\ g)$

thm VSUM_CONST:

$\forall (c::(real,\ ?'b::type)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda n::?'a::type.\ c) = \%\ (real\_of\_nat\ (CARD\ s))\ c$

thm VSUM_COMPONENT:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (vsum\ s\ f)\ i = sum\ s\ (\lambda x::?'b::type.\ \$\ (f\ x)\ i)$

thm VSUM_IMAGE:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'c::type \Rightarrow bool.\ FINITE\ s \land (\forall (x::?'c::type)\ y::?'c::type.\ IN\ x\ s \land IN\ y\ s \land f\ x = f\ y \longrightarrow x = y) \longrightarrow vsum\ (IMAGE\ f\ s)\ g = vsum\ s\ (g \circ f)$

thm VSUM_UNION:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.$ $FINITE\ s \wedge FINITE\ t \wedge DISJOINT\ s\ t \longrightarrow vsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = vector\_add\ (vsum\ s\ f)\ (vsum\ t\ f)$

thm VSUM_DIFF:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.$ $FINITE\ s \wedge SUBSET\ t\ s \longrightarrow vsum\ (DIFF\ s\ t)\ f = vector\_sub\ (vsum\ s\ f)\ (vsum\ t\ f)$

thm VSUM_DELETE:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ a::?'b::type.$ $FINITE\ s \wedge IN\ a\ s \longrightarrow vsum\ (DELETE\ s\ a)\ f = vector\_sub\ (vsum\ s\ f)\ (f\ a)$

thm VSUM_INCL_EXCL:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow (real, ?'a::type)\ cart.$ $FINITE\ s \wedge FINITE\ t \longrightarrow vector\_add\ (vsum\ s\ f)\ (vsum\ t\ f) = vector\_add\ (vsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f)\ (vsum\ (HOL\_Light\_Import.INTER\ s\ t)\ f)$

thm VSUM_NEG:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ vsum\ s\ (\lambda x::?'b::type.$ $vector\_neg\ (f\ x)) = vector\_neg\ (vsum\ s\ f)$

thm VSUM_EQ:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real, ?'a::type)\ cart)$ $s::?'b::type \Rightarrow bool.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow vsum\ s\ f = vsum\ s\ g$

thm VSUM_SUPERSET:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (u::?'b::type \Rightarrow bool)\ v::?'b::type \Rightarrow bool.\ SUBSET\ u\ v \wedge (\forall x::?'b::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = vec\ (0::nat))$ $\longrightarrow vsum\ v\ f = vsum\ u\ f$

thm VSUM_EQ_SUPERSET:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.$ $FINITE\ t \wedge SUBSET\ t\ s \wedge (\forall x::?'b::type.\ IN\ x\ t \longrightarrow f\ x = (?g::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ x) \wedge (\forall x::?'b::type.\ IN\ x\ s \wedge \neg\ IN\ x\ t \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ s\ f = vsum\ t\ ?g$

thm VSUM_UNION_RZERO:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (u::?'b::type \Rightarrow bool)\ v::?'b::type \Rightarrow bool.\ FINITE\ u \wedge (\forall x::?'b::type.\ IN\ x\ v \wedge \neg\ IN\ x\ u \longrightarrow f\ x = vec\ (0::nat))$ $\longrightarrow vsum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = vsum\ u\ f$

thm VSUM_UNION_LZERO:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (u::?'b::type \Rightarrow bool)\ v::?'b::type \Rightarrow bool.\ FINITE\ v \wedge (\forall x::?'b::type.\ IN\ x\ u \wedge \neg\ IN\ x\ v \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ (HOL\_Light\_Import.UNION\ u\ v)\ f = vsum\ v\ f$

thm VSUM_RESTRICT:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ if\ IN\ x\ s\ then\ f\ x\ else\ vec\ (0::nat)) = vsum\ s\ f$

thm VSUM_RESTRICT_SET:

$\forall (P::?'b::type \Rightarrow bool)\ (s::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ vsum\ (GSPEC\ (\lambda GEN\%PVAR\%274::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%274\ (IN\ x\ s \wedge P\ x)\ x))\ f = vsum\ s\ (\lambda x::?'b::type.\ if\ P\ x\ then\ f\ x\ else\ vec\ (0::nat))$

thm VSUM_CASES:

$\forall (s::?'b::type \Rightarrow bool)\ (P::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ g::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ if\ P\ x\ then\ f\ x\ else\ g\ x) = vector\_add\ (vsum\ (GSPEC\ (\lambda GEN\%PVAR\%275::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%275\ (IN\ x\ s \wedge P\ x)\ x))\ f)\ (vsum\ (GSPEC\ (\lambda GEN\%PVAR\%276::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%276\ (IN\ x\ s \wedge \neg\ P\ x)\ x))\ g)$

thm VSUM_SING:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ x::?'b::type.\ vsum\ (INSERT\ x\ EMPTY)\ f = f\ x$

thm VSUM_NORM:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vector\_norm\ (vsum\ s\ f) \le sum\ s\ (\lambda x::?'b::type.\ vector\_norm\ (f\ x))$

thm VSUM_NORM_LE:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ g::?'b::type \Rightarrow real.\ FINITE\ s \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) \le g\ x) \longrightarrow vector\_norm\ (vsum\ s\ f) \le sum\ s\ g$

thm VSUM_NORM_TRIANGLE:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ b::real.\ FINITE\ s \wedge sum\ s\ (\lambda a::?'b::type.\ vector\_norm\ (f\ a)) \le b \longrightarrow vector\_norm\ (vsum\ s\ f) \le b$

thm VSUM_NORM_BOUND:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ b::real.\ FINITE\ s \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) \le b) \longrightarrow vector\_norm\ (vsum\ s\ f) \le real\_of\_nat\ (CARD\ s) * b$

thm VSUM_CLAUSES_NUMSEG:

(∀ m::nat. vsum (dotdot m (0::nat)) (?f::nat ⇒ (real, ?'a::type) cart) = (if m = (0::nat) then ?f (0::nat) else vec (0::nat))) ∧ (∀ (m::nat) n::nat. vsum (dotdot m (Suc n)) ?f = (if m ≤ Suc n then vector_add (vsum (dotdot m n) ?f) (?f (Suc n)) else vsum (dotdot m n) ?f))

thm VSUM_CLAUSES_NUMSEG_conjunct1:

∀ (m::nat) n::nat. vsum (dotdot m (Suc n)) (?f::nat ⇒ (real, ?'a::type) cart) = (if m ≤ Suc n then vector_add (vsum (dotdot m n) ?f) (?f (Suc n)) else vsum (dotdot m n) ?f)

thm VSUM_CLAUSES_NUMSEG_conjunct0:

∀ m::nat. vsum (dotdot m (0::nat)) (?f::nat ⇒ (real, ?'a::type) cart) = (if m = (0::nat) then ?f (0::nat) else vec (0::nat))

thm VSUM_CLAUSES_RIGHT:

∀ (f::nat ⇒ (real, ?'a::type) cart) (m::nat) n::nat. (0::nat) < n ∧ m ≤ n ⟶ vsum (dotdot m n) f = vector_add (vsum (dotdot m (n − (1::nat))) f) (f n)

thm VSUM_CMUL_NUMSEG:

∀ (f::nat ⇒ (real, ?'a::type) cart) (c::real) (m::nat) n::nat. vsum (dotdot m n) (λx::nat. % c (f x)) = % c (vsum (dotdot m n) f)

thm VSUM_EQ_NUMSEG:

∀ (f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, ?'a::type) cart) (m::nat) n::nat. (∀ x::nat. m ≤ x ∧ x ≤ n ⟶ f x = g x) ⟶ vsum (dotdot m n) f = vsum (dotdot m n) g

thm VSUM_IMAGE_GEN:

∀ (f::?'c::type ⇒ ?'b::type) (g::?'c::type ⇒ (real, ?'a::type) cart) s::?'c::type ⇒ bool. FINITE s ⟶ vsum s g = vsum (IMAGE f s) (λy::?'b::type. vsum (GSPEC (λGEN%PVAR%277::?'c::type. ∃ x::?'c::type. SETSPEC GEN%PVAR%277 (IN x s ∧ f x = y) x)) g)

thm VSUM_GROUP:

∀ (f::?'c::type ⇒ ?'b::type) (g::?'c::type ⇒ (real, ?'a::type) cart) (s::?'c::type ⇒ bool) t::?'b::type ⇒ bool. FINITE s ∧ SUBSET (IMAGE f s) t ⟶ vsum t (λy::?'b::type. vsum (GSPEC (λGEN%PVAR%278::?'c::type. ∃ x::?'c::type. SETSPEC GEN%PVAR%278 (IN x s ∧ f x = y) x)) g) = vsum s g

thm VSUM_VMUL:

∀ (f::?'b::type ⇒ real) (v::(real, ?'a::type) cart) s::?'b::type ⇒ bool. FINITE s ⟶ % (sum s f) v = vsum s (λx::?'b::type. % (f x) v)

thm VSUM_DELTA:

∀ (s::?'b::type ⇒ bool) a::?'b::type. vsum s (λx::?'b::type. if x = a then ?b::(real, ?'a::type) cart else vec (0::nat)) = (if IN a s then ?b else vec (0::nat))

thm VSUM_ADD_NUMSEG:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ vector\_add\ (f\ i)\ (g\ i)) = vector\_add\ (vsum\ (dotdot\ m\ n)\ f)\ (vsum\ (dotdot\ m\ n)\ g)$

thm VSUM_SUB_NUMSEG:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ vector\_sub\ (f\ i)\ (g\ i)) = vector\_sub\ (vsum\ (dotdot\ m\ n)\ f)\ (vsum\ (dotdot\ m\ n)\ g)$

thm VSUM_ADD_SPLIT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ (n::nat)\ p::nat.\ m \leq n + (1::nat) \longrightarrow vsum\ (dotdot\ m\ (n + p))\ f = vector\_add\ (vsum\ (dotdot\ m\ n)\ f)\ (vsum\ (dotdot\ (n + (1::nat))\ (n + p))\ f)$

thm VSUM_VSUM_PRODUCT:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'c::type \Rightarrow ?'b::type \Rightarrow bool)\ x::?'c::type \Rightarrow ?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ FINITE\ s \wedge (\forall i::?'c::type.\ IN\ i\ s \longrightarrow FINITE\ (t\ i)) \longrightarrow vsum\ s\ (\lambda i::?'c::type.\ vsum\ (t\ i)\ (x\ i)) = vsum\ (GSPEC\ (\lambda GEN\%PVAR\%279::?'c::type \times ?'b::type.\ \exists (i::?'c::type)\ j::?'b::type.\ SETSPEC\ GEN\%PVAR\%279\ (IN\ i\ s \wedge IN\ j\ (t\ i))\ (i,\ j)))\ (GABS\ (\lambda f::?'c::type \times ?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (i::?'c::type)\ j::?'b::type.\ GEQ\ (f\ (i,\ j))\ (x\ i\ j)))$

thm VSUM_IMAGE_NONZERO:

$\forall (d::?'c::type \Rightarrow (real,\ ?'b::type)\ cart)\ (i::?'a::type \Rightarrow ?'c::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \wedge i\ x = i\ y \longrightarrow d\ (i\ x) = vec\ (0::nat)) \longrightarrow vsum\ (IMAGE\ i\ s)\ d = vsum\ s\ (d \circ i)$

thm VSUM_UNION_NONZERO:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge (\forall x::?'b::type.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = vector\_add\ (vsum\ s\ f)\ (vsum\ t\ f)$

thm VSUM_UNIONS_NONZERO:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::(?'b::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::?'b::type \Rightarrow bool.\ IN\ t\ s \longrightarrow FINITE\ t) \wedge (\forall (t1::?'b::type \Rightarrow bool)\ (t2::?'b::type \Rightarrow bool)\ x::?'b::type.\ IN\ t1\ s \wedge IN\ t2\ s \wedge t1 \neq t2 \wedge IN\ x\ t1 \wedge IN\ x\ t2 \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ (UNIONS\ s)\ f = vsum\ s\ (\lambda t::?'b::type \Rightarrow bool.\ vsum\ t\ f)$

thm VSUM_CLAUSES_LEFT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow vsum\ (dotdot\ m\ n)\ f = vector\_add\ (f\ m)\ (vsum\ (dotdot\ (m + (1::nat))\ n)\ f)$

thm VSUM_DIFFS:

$\forall (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ vector\_sub\ ((?f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k)\ (?f\ (k + (1::nat)))) = (if\ m \leq n\ then\ vector\_sub\ (?f\ m)\ (?f\ (n + (1::nat)))\ else\ vec\ (0::nat))$

thm VSUM_DIFFS_ALT:

$\forall (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ vector\_sub\ ((?f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (k + (1::nat)))\ (?f\ k)) = (if\ m \leq n\ then\ vector\_sub\ (?f\ (n + (1::nat)))\ (?f\ m)\ else\ vec\ (0::nat))$

thm VSUM_DELETE_CASES:

$\forall (x::?'b::type)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ (DELETE\ s\ x)\ f = (if\ IN\ x\ s\ then\ vector\_sub\ (vsum\ s\ f)\ (f\ x)\ else\ vsum\ s\ f)$

thm VSUM_EQ_GENERAL:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ (f::?'c::type \Rightarrow (real,\ ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ h::?'c::type \Rightarrow ?'b::type.\ (\forall y::?'b::type.\ IN\ y\ t \longrightarrow (\exists !x::?'c::type.\ IN\ x\ s \wedge h\ x = y)) \wedge (\forall x::?'c::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge g\ (h\ x) = f\ x) \longrightarrow vsum\ s\ f = vsum\ t\ g$

thm VSUM_EQ_GENERAL_INVERSES:

$\forall (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ (f::?'c::type \Rightarrow (real,\ ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (h::?'c::type \Rightarrow ?'b::type)\ k::?'b::type \Rightarrow ?'c::type.\ (\forall y::?'b::type.\ IN\ y\ t \longrightarrow IN\ (k\ y)\ s \wedge h\ (k\ y) = y) \wedge (\forall x::?'c::type.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t \wedge k\ (h\ x) = x \wedge g\ (h\ x) = f\ x) \longrightarrow vsum\ s\ f = vsum\ t\ g$

thm VSUM_NORM_ALLSUBSETS_BOUND:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (p::?'b::type \Rightarrow bool)\ e::real.\ FINITE\ p \wedge (\forall q::?'b::type \Rightarrow bool.\ SUBSET\ q\ p \longrightarrow vector\_norm\ (vsum\ q\ f) \leq e) \longrightarrow sum\ p\ (\lambda x::?'b::type.\ vector\_norm\ (f\ x)) \leq real\_of\_nat\ (2::nat) * (real\_of\_nat\ (dimindex\ HOL\_Light\_Import.UNIV) * e)$

thm DOT_LSUM:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ FINITE\ s \longrightarrow dot\ (vsum\ s\ f)\ y = sum\ s\ (\lambda x::?'b::type.\ dot\ (f\ x)\ y)$

thm DOT_RSUM:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ FINITE\ s \longrightarrow dot\ x\ (vsum\ s\ f) = sum\ s\ (\lambda y::?'b::type.\ dot\ x\ (f\ y))$

thm VSUM_OFFSET:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ p::nat.\ vsum\ (dotdot\ (m + p)\ ((?n::nat) + p))\ f = vsum\ (dotdot\ m\ ?n)\ (\lambda i::nat.\ f\ (i + p))$

thm VSUM_OFFSET_0:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow vsum\ (dotdot\ m\ n)\ f = vsum\ (dotdot\ (0::nat)\ (n - m))\ (\lambda i::nat.\ f\ (i + m))$

thm VSUM_TRIV_NUMSEG:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ n < m \longrightarrow vsum\ (dotdot\ m\ n)\ f = vec\ (0::nat)$

thm VSUM_CONST_NUMSEG:

$\forall (c::(real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda n::nat.\ c) = \%\ (real\_of\_nat\ (n + (1::nat) - m))\ c$

thm VSUM_SUC:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ (Suc\ n)\ (Suc\ m))\ f = vsum\ (dotdot\ n\ m)\ (f \circ Suc)$

thm VSUM_BIJECTION:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (p::?'b::type \Rightarrow ?'b::type)\ s::?'b::type \Rightarrow bool.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \land (\forall y::?'b::type.\ IN\ y\ s \longrightarrow (\exists !x::?'b::type.\ IN\ x\ s \land p\ x = y)) \longrightarrow vsum\ s\ f = vsum\ s\ (f \circ p)$

thm VSUM_PARTIAL_SUC:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ \%\ (f\ k)\ (vector\_sub\ (g\ (k + (1::nat)))\ (g\ k))) = (if\ m \leq n\ then\ vector\_sub\ (vector\_sub\ (\%\ (f\ (n + (1::nat)))\ (g\ (n + (1::nat))))\ (\%\ (f\ m)\ (g\ m)))\ (vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ \%\ (f\ (k + (1::nat)) - f\ k)\ (g\ (k + (1::nat)))))\ else\ vec\ (0::nat))$

thm VSUM_PARTIAL_PRE:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ \%\ (f\ k)\ (vector\_sub\ (g\ k)\ (g\ (k - (1::nat))))) = (if\ m \leq n\ then\ vector\_sub\ (vector\_sub\ (\%\ (f\ (n + (1::nat)))\ (g\ n))\ (\%\ (f\ m)\ (g\ (m - (1::nat)))))\ (vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ \%\ (f\ (k + (1::nat)) - f\ k)\ (g\ k)))\ else\ vec\ (0::nat))$

thm VSUM_COMBINE_L:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ (n::nat)\ p::nat.\ (0::nat) < n \land m \leq n \land n \leq p + (1::nat) \longrightarrow vector\_add\ (vsum\ (dotdot\ m\ (n - (1::nat)))\ f)\ (vsum\ (dotdot\ n\ p)\ f) = vsum\ (dotdot\ m\ p)\ f$

thm VSUM_COMBINE_R:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ (n::nat)\ p::nat.\ m \leq n + (1::nat) \land n \leq p \longrightarrow vector\_add\ (vsum\ (dotdot\ m\ n)\ f)\ (vsum\ (dotdot\ (n + (1::nat))\ p)\ f) = vsum\ (dotdot\ m\ p)\ f$

thm VSUM_INJECTION:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (p::?'b::type \Rightarrow ?'b::type)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge p\ x = p\ y \longrightarrow x = y) \longrightarrow vsum\ s\ (f \circ p) = vsum\ s\ f$

thm VSUM_SWAP:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'c::type \Rightarrow bool)\ t::?'b::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow vsum\ s\ (\lambda i::?'c::type.\ vsum\ t\ (f\ i)) = vsum\ t\ (\lambda j::?'b::type.\ vsum\ s\ (\lambda i::?'c::type.\ f\ i\ j))$

thm VSUM_SWAP_NUMSEG:

$\forall (a::nat)\ (b::nat)\ (c::nat)\ (d::nat)\ f::nat \Rightarrow nat \Rightarrow (real, ?'a::type)\ cart.\ vsum\ (dotdot\ a\ b)\ (\lambda i::nat.\ vsum\ (dotdot\ c\ d)\ (f\ i)) = vsum\ (dotdot\ c\ d)\ (\lambda j::nat.\ vsum\ (dotdot\ a\ b)\ (\lambda i::nat.\ f\ i\ j))$

thm VSUM_ADD_GEN:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%282::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%282\ (IN\ x\ s \wedge f\ x \neq vec\ (0::nat))\ x)) \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%283::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%283\ (IN\ x\ s \wedge g\ x \neq vec\ (0::nat))\ x)) \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ vector\_add\ (f\ x)\ (g\ x)) = vector\_add\ (vsum\ s\ f)\ (vsum\ s\ g)$

thm VSUM_CASES_1:

$\forall (s::?'b::type \Rightarrow bool)\ a::?'b::type.\ FINITE\ s \wedge IN\ a\ s \longrightarrow vsum\ s\ (\lambda x::?'b::type.\ if\ x = a\ then\ ?y::(real, ?'a::type)\ cart\ else\ (?f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ x) = vector\_add\ (vsum\ s\ ?f)\ (vector\_sub\ ?y\ (?f\ a))$

thm VSUM_SING_NUMSEG:

$vsum\ (dotdot\ (?n::nat)\ ?n)\ (?f::nat \Rightarrow (real, ?'a::type)\ cart) = ?f\ ?n$

thm VSUM_1:

$vsum\ (dotdot\ (1::nat)\ (1::nat))\ (?f::nat \Rightarrow (real, ?'a::type)\ cart) = ?f\ (1::nat)$

thm VSUM_2:

$\forall t::nat \Rightarrow (real, ?'a::type)\ cart.\ vsum\ (dotdot\ (1::nat)\ (2::nat))\ t = vector\_add\ (t\ (1::nat))\ (t\ (2::nat))$

thm VSUM_3:

$\forall t::nat \Rightarrow (real, ?'a::type)\ cart.\ vsum\ (dotdot\ (1::nat)\ (3::nat))\ t = vector\_add\ (t\ (1::nat))\ (vector\_add\ (t\ (2::nat))\ (t\ (3::nat)))$

thm VSUM_PAIR:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ ((2::nat) * m)\ ((2::nat) * n + (1::nat)))\ f = vsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ vector\_add\ (f\ ((2::nat) * i))\ (f\ ((2::nat) * i + (1::nat))))$

thm VSUM_PAIR_0:

$\forall (f::nat \Rightarrow (real, ?'a::type)\ cart)\ n::nat.\ vsum\ (dotdot\ (0::nat)\ ((2::nat) * n + (1::nat)))\ f = vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ vector\_add\ (f\ ((2::nat) * i))\ (f\ ((2::nat) * i + (1::nat))))$

thm DEF_basis:

$basis = (\lambda\_111344::nat.\ lambda\ (\lambda i::nat.\ if\ i = \_111344\ then\ 1::real\ else\ (0::real)))$

thm basis:

$\forall k::nat.\ basis\ k = lambda\ (\lambda i::nat.\ if\ i = k\ then\ 1::real\ else\ (0::real))$

thm NORM_BASIS:

$\forall k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow vector\_norm\ (basis\ k) = (1::real)$

thm NORM_BASIS_1:

$vector\_norm\ (basis\ (1::nat)) = (1::real)$

thm VECTOR_CHOOSE_SIZE:

$\forall c \geq 0::real.\ \exists x::(real, ?'a::type)\ cart.\ vector\_norm\ x = c$

thm VECTOR_CHOOSE_DIST:

$\forall (x::(real, ?'a::type)\ cart)\ e::real.\ (0::real) \leq e \longrightarrow (\exists y::(real, ?'a::type)\ cart.\ distance\ (x, y) = e)$

thm BASIS_INJ:

$\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge basis\ i = basis\ j \longrightarrow i = j$

thm BASIS_NE:

$\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge i \neq j \longrightarrow basis\ i \neq basis\ j$

thm BASIS_COMPONENT:

$\forall (k::nat)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (basis\ k)\ i = (if\ i = k\ then\ 1::real\ else\ (0::real))$

thm BASIS_EXPANSION:

$\forall x::(real, ?'a::type)\ cart.\ vsum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \%\ (\$\ x\ i)\ (basis\ i)) = x$

thm BASIS_EXPANSION_UNIQUE:

$\forall (f::nat \Rightarrow real)\ x::(real, ?'a::type)\ cart.\ (vsum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \%\ (f\ i)\ (basis\ i)) = x) = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow f\ i = \$\ x\ i)$

thm DOT_BASIS:

$\forall (x::(real, ?'a::type) \ cart) \ i::nat. \ (1::nat) \leq i \wedge i \leq dimindex \ HOL\_Light\_Import.UNIV \longrightarrow dot \ (basis \ i) \ x = \$ \ x \ i \wedge dot \ x \ (basis \ i) = \$ \ x \ i$

thm DOT_BASIS_BASIS:

$\forall (i::nat) \ j::nat. \ (1::nat) \leq i \wedge i \leq dimindex \ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex \ HOL\_Light\_Import.UNIV \longrightarrow dot \ (basis \ i) \ (basis \ j) = (if \ i = j \ then \ 1::real \ else \ (0::real))$

thm DOT_BASIS_BASIS_UNEQUAL:

$\forall (i::nat) \ j::nat. \ i \neq j \longrightarrow dot \ (basis \ i) \ (basis \ j) = (0::real)$

thm BASIS_EQ_0:

$\forall i::nat. \ (basis \ i = vec \ (0::nat)) = (\neg \ IN \ i \ (dotdot \ (1::nat) \ (dimindex \ HOL\_Light\_Import.UNIV)))$

thm BASIS_NONZERO:

$\forall k::nat. \ (1::nat) \leq k \wedge k \leq dimindex \ HOL\_Light\_Import.UNIV \longrightarrow basis \ k \neq vec \ (0::nat)$

thm VECTOR_EQ_LDOT:

$\forall (y::(real, ?'a::type) \ cart) \ z::(real, ?'a::type) \ cart. \ (\forall x::(real, ?'a::type) \ cart. \ dot \ x \ y = dot \ x \ z) = (y = z)$

thm VECTOR_EQ_RDOT:

$\forall (x::(real, ?'a::type) \ cart) \ y::(real, ?'a::type) \ cart. \ (\forall z::(real, ?'a::type) \ cart. \ dot \ x \ z = dot \ y \ z) = (x = y)$

thm DEF_orthogonal:

$orthogonal = (\lambda(\_111508::(real, ?'a::type) \ cart) \ \_111509::(real, ?'a::type) \ cart. \ dot \ \_111508 \ \_111509 = (0::real))$

thm orthogonal:

$\forall (x::(real, ?'a::type) \ cart) \ y::(real, ?'a::type) \ cart. \ orthogonal \ x \ y = (dot \ x \ y = (0::real))$

thm ORTHOGONAL_0:

$\forall x::(real, ?'a::type) \ cart. \ orthogonal \ (vec \ (0::nat)) \ x \wedge orthogonal \ x \ (vec \ (0::nat))$

thm ORTHOGONAL_REFL:

$\forall x::(real, ?'a::type) \ cart. \ orthogonal \ x \ x = (x = vec \ (0::nat))$

thm ORTHOGONAL_SYM:

$\forall (x::(real, ?'a::type) \ cart) \ y::(real, ?'a::type) \ cart. \ orthogonal \ x \ y = orthogonal \ y \ x$

thm ORTHOGONAL_LNEG:

$\forall (x::(real, ?'a::type) cart)\ y::(real, ?'a::type)\ cart.\ orthogonal\ (vector\_neg\ x)\ y = orthogonal\ x\ y$

thm ORTHOGONAL_RNEG:

$\forall (x::(real, ?'a::type) cart)\ y::(real, ?'a::type)\ cart.\ orthogonal\ x\ (vector\_neg\ y) = orthogonal\ x\ y$

thm ORTHOGONAL_BASIS:

$\forall (x::(real, ?'a::type) cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow orthogonal\ (basis\ i)\ x = (\$\ x\ i = (0::real))$

thm ORTHOGONAL_BASIS_BASIS:

$\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow orthogonal\ (basis\ i)\ (basis\ j) = (i \neq j)$

thm ORTHOGONAL_CLAUSES:

$(\forall a::(real, ?'j::type)\ cart.\ orthogonal\ a\ (vec\ (0::nat))) \wedge (\forall (a::(real, ?'i::type)\ cart)\ (x::(real, ?'i::type)\ cart)\ c::real.\ orthogonal\ a\ x \longrightarrow orthogonal\ a\ (\%\ c\ x)) \wedge (\forall (a::(real, ?'h::type)\ cart)\ x::(real, ?'h::type)\ cart.\ orthogonal\ a\ x \longrightarrow orthogonal\ a\ (vector\_neg\ x)) \wedge (\forall (a::(real, ?'g::type)\ cart)\ (x::(real, ?'g::type)\ cart)\ y::(real, ?'g::type)\ cart.\ orthogonal\ a\ x \wedge orthogonal\ a\ y \longrightarrow orthogonal\ a\ (vector\_add\ x\ y)) \wedge (\forall (a::(real, ?'f::type)\ cart)\ (x::(real, ?'f::type)\ cart)\ y::(real, ?'f::type)\ cart.\ orthogonal\ a\ x \wedge orthogonal\ a\ y \longrightarrow orthogonal\ a\ (vector\_sub\ x\ y)) \wedge (\forall a::(real, ?'e::type)\ cart.\ orthogonal\ (vec\ (0::nat))\ a) \wedge (\forall (a::(real, ?'d::type)\ cart)\ (x::(real, ?'d::type)\ cart)\ c::real.\ orthogonal\ x\ a \longrightarrow orthogonal\ (\%\ c\ x)\ a) \wedge (\forall (a::(real, ?'c::type)\ cart)\ x::(real, ?'c::type)\ cart.\ orthogonal\ x\ a \longrightarrow orthogonal\ (vector\_neg\ x)\ a) \wedge (\forall (a::(real, ?'b::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ orthogonal\ x\ a \wedge orthogonal\ y\ a \longrightarrow orthogonal\ (vector\_add\ x\ y)\ a) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ orthogonal\ x\ a \wedge orthogonal\ y\ a \longrightarrow orthogonal\ (vector\_sub\ x\ y)\ a)$

thm VECTOR_1:

$\$\ (vector\ [?x::?'a::type])\ (1::nat) = ?x$

thm VECTOR_2:

$\$\ (vector\ [?x::?'a::type,\ ?y::?'a::type])\ (1::nat) = ?x \wedge \$\ (vector\ [?x,\ ?y])\ (2::nat) = ?y$

thm VECTOR_3:

$\$\ (vector\ [?x::?'a::type,\ ?y::?'a::type,\ ?z::?'a::type])\ (1::nat) = ?x \wedge \$\ (vector\ [?x,\ ?y,\ ?z])\ (2::nat) = ?y \wedge \$\ (vector\ [?x,\ ?y,\ ?z])\ (3::nat) = ?z$

thm FORALL_VECTOR_1:

$(\forall v::(?'a::type, unit)\ cart.\ (?P::(?'a::type, unit)\ cart \Rightarrow bool)\ v) = (\forall x::?'a::type.\ ?P\ (vector\ [x]))$

thm VECTOR_2_conjunct1:

$\$ (vector\ [?x::?'a::type,\ ?y::?'a::type])\ (2::nat) = ?y$

thm VECTOR_2_conjunct0:

$\$ (vector\ [?x::?'a::type,\ ?y::?'a::type])\ (1::nat) = ?x$

thm FORALL_VECTOR_2:

$(\forall v::(?'a::type,\ 2)\ cart.\ (?P::(?'a::type,\ 2)\ cart \Rightarrow bool)\ v) = (\forall (x::?'a::type)\ y::?'a::type.\ ?P\ (vector\ [x,\ y]))$

thm VECTOR_3_conjunct2:

$\$ (vector\ [?x::?'a::type,\ ?y::?'a::type,\ ?z::?'a::type])\ (3::nat) = ?z$

thm VECTOR_3_conjunct1:

$\$ (vector\ [?x::?'a::type,\ ?y::?'a::type,\ ?z::?'a::type])\ (2::nat) = ?y$

thm VECTOR_3_conjunct0:

$\$ (vector\ [?x::?'a::type,\ ?y::?'a::type,\ ?z::?'a::type])\ (1::nat) = ?x$

thm FORALL_VECTOR_3:

$(\forall v::(?'a::type,\ 3)\ cart.\ (?P::(?'a::type,\ 3)\ cart \Rightarrow bool)\ v) = (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ ?P\ (vector\ [x,\ y,\ z]))$

thm EXISTS_VECTOR_1:

$(\exists v::(?'a::type,\ unit)\ cart.\ (?P::(?'a::type,\ unit)\ cart \Rightarrow bool)\ v) = (\exists x::?'a::type.\ ?P\ (vector\ [x]))$

thm EXISTS_VECTOR_2:

$(\exists v::(?'a::type,\ 2)\ cart.\ (?P::(?'a::type,\ 2)\ cart \Rightarrow bool)\ v) = (\exists (x::?'a::type)\ y::?'a::type.\ ?P\ (vector\ [x,\ y]))$

thm EXISTS_VECTOR_3:

$(\exists v::(?'a::type,\ 3)\ cart.\ (?P::(?'a::type,\ 3)\ cart \Rightarrow bool)\ v) = (\exists (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ ?P\ (vector\ [x,\ y,\ z]))$

thm DEF_linear:

$linear = (\lambda\_111552::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ \_111552\ (vector\_add\ x\ y) = vector\_add\ (\_111552\ x)\ (\_111552\ y)) \wedge (\forall (c::real)\ x::(real,\ ?'b::type)\ cart.\ \_111552\ (\%\ c\ x) = \%\ c\ (\_111552\ x)))$

thm linear:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f = ((\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ (vector\_add\ x\ y) = vector\_add\ (f\ x)\ (f\ y)) \wedge (\forall (c::real)\ x::(real,\ ?'b::type)\ cart.\ f\ (\%\ c\ x) = \%\ c\ (f\ x)))$

thm LINEAR_COMPOSE_CMUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ c::real.\ linear\ f \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ c\ (f\ x))$

thm LINEAR_COMPOSE_NEG:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f\ x))$

thm LINEAR_COMPOSE_ADD:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x))$

thm LINEAR_COMPOSE_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))$

thm LINEAR_COMPOSE:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow linear\ (g \circ f)$

thm LINEAR_ID:

$linear\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)$

thm LINEAR_I:

$linear\ id$

thm LINEAR_ZERO:

$linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ vec\ (0::nat))$

thm LINEAR_NEGATION:

$linear\ vector\_neg$

thm LINEAR_COMPOSE_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'c::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'c::type.\ IN\ a\ s \longrightarrow linear\ (f\ a)) \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ s\ (\lambda a::?'c::type.\ f\ a\ x))$

thm LINEAR_VMUL_COMPONENT:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ k::nat.\ linear\ f \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow linear\ (\lambda x::(real,\ ?'c::type)\ cart.\ \%\ (\$\ (f\ x)\ k)\ v)$

thm LINEAR_0:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow f\ (vec\ (0::nat)) = vec\ (0::nat)$

thm LINEAR_CMUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::real)\ x::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow f\ (\%\ c\ x) = \%\ c\ (f\ x)$

thm LINEAR_NEG:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow f\ (vector\_neg\ x) = vector\_neg\ (f\ x)$

thm LINEAR_ADD:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow f\ (vector\_add\ x\ y) = vector\_add\ (f\ x)\ (f\ y)$

thm LINEAR_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow f\ (vector\_sub\ x\ y) = vector\_sub\ (f\ x)\ (f\ y)$

thm LINEAR_VSUM:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::?'a::type \Rightarrow (real,\ ?'c::type)\ cart)\ s::?'a::type \Rightarrow bool.\ linear\ f \wedge FINITE\ s \longrightarrow f\ (vsum\ s\ g) = vsum\ s\ (f \circ g)$

thm LINEAR_VSUM_MUL:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::?'a::type \Rightarrow bool)\ (c::?'a::type \Rightarrow real)\ v::?'a::type \Rightarrow (real,\ ?'c::type)\ cart.\ linear\ f \wedge FINITE\ s \longrightarrow f\ (vsum\ s\ (\lambda i::?'a::type.\ \%\ (c\ i)\ (v\ i))) = vsum\ s\ (\lambda i::?'a::type.\ \%\ (c\ i)\ (f\ (v\ i)))$

thm LINEAR_INJECTIVE_0:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) = (\forall x::(real,\ ?'b::type)\ cart.\ f\ x = vec\ (0::nat) \longrightarrow x = vec\ (0::nat))$

thm LINEAR_BOUNDED:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\exists B::real.\ \forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) \leq B * vector\_norm\ x)$

thm LINEAR_BOUNDED_POS:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\exists B>0::real.\ \forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) \leq B * vector\_norm\ x)$

thm SYMMETRIC_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (vector\_neg\ x)\ s) \wedge linear\ f \longrightarrow$

$(\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (IMAGE\ f\ s) \longrightarrow IN\ (vector\_neg\ x)\ (IMAGE\ f\ s))$

thm DEF_bilinear:

$bilinear = (\lambda\_111862::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'c::type)\ cart.\ linear\ (\_111862\ x)) \wedge (\forall y::(real,\ ?'b::type)\ cart.\ linear\ (\lambda x::(real,\ ?'c::type)\ cart.\ \_111862\ x\ y)))$

thm bilinear:

$\forall f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ bilinear\ f = ((\forall x::(real,\ ?'c::type)\ cart.\ linear\ (f\ x)) \wedge (\forall y::(real,\ ?'b::type)\ cart.\ linear\ (\lambda x::(real,\ ?'c::type)\ cart.\ f\ x\ y)))$

thm BILINEAR_LADD:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ (y::(real,\ ?'c::type)\ cart)\ z::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ (vector\_add\ x\ y)\ z = vector\_add\ (h\ x\ z)\ (h\ y\ z)$

thm BILINEAR_RADD:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ (y::(real,\ ?'b::type)\ cart)\ z::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ x\ (vector\_add\ y\ z) = vector\_add\ (h\ x\ y)\ (h\ x\ z)$

thm BILINEAR_LMUL:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::real)\ (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ (\%\ c\ x)\ y = \%\ c\ (h\ x\ y)$

thm BILINEAR_RMUL:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::real)\ (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ x\ (\%\ c\ y) = \%\ c\ (h\ x\ y)$

thm BILINEAR_LNEG:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ (vector\_neg\ x)\ y = vector\_neg\ (h\ x\ y)$

thm BILINEAR_RNEG:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ x\ (vector\_neg\ y) = vector\_neg\ (h\ x\ y)$

thm BILINEAR_LZERO:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ (vec\ (0::nat))\ x = vec\ (0::nat)$

thm BILINEAR_RZERO:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)$ $x::(real,\ ?'c::type)\ cart.\ bilinear\ h \longrightarrow h\ x\ (vec\ (0::nat)) = vec\ (0::nat)$

thm BILINEAR_LSUB:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)$ $(x::(real,\ ?'c::type)\ cart)\ (y::(real,\ ?'c::type)\ cart)\ z::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ (vector\_sub\ x\ y)\ z = vector\_sub\ (h\ x\ z)\ (h\ y\ z)$

thm BILINEAR_RSUB:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)$ $(x::(real,\ ?'c::type)\ cart)\ (y::(real,\ ?'b::type)\ cart)\ z::(real,\ ?'b::type)\ cart.\ bilinear\ h \longrightarrow h\ x\ (vector\_sub\ y\ z) = vector\_sub\ (h\ x\ y)\ (h\ x\ z)$

thm BILINEAR_VSUM:

$\forall h::(real,\ ?'e::type)\ cart \Rightarrow (real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'c::type)\ cart.\ bilinear\ h \wedge FINITE\ (?s::?'b::type \Rightarrow bool) \wedge FINITE\ (?t::?'a::type \Rightarrow bool) \longrightarrow h\ (vsum\ ?s\ (?f::?'b::type \Rightarrow (real,\ ?'e::type)\ cart))\ (vsum\ ?t\ (?g::?'a::type \Rightarrow (real,\ ?'d::type)\ cart)) = vsum\ (CROSS\ ?s\ ?t)\ (GABS\ (\lambda f::?'b::type \times ?'a::type \Rightarrow (real,\ ?'c::type)\ cart.\ \forall (i::?'b::type)\ j::?'a::type.\ GEQ\ (f\ (i,\ j))\ (h\ (?f\ i)\ (?g\ j))))$

thm BILINEAR_BOUNDED:

$\forall h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ bilinear\ h \longrightarrow (\exists B::real.\ \forall (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ vector\_norm\ (h\ x\ y) \leq B * (vector\_norm\ x * vector\_norm\ y))$

thm BILINEAR_BOUNDED_POS:

$\forall h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ bilinear\ h \longrightarrow (\exists B>0::real.\ \forall (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ vector\_norm\ (h\ x\ y) \leq B * (vector\_norm\ x * vector\_norm\ y))$

thm BILINEAR_VSUM_PARTIAL_SUC:

$\forall (f::nat \Rightarrow (real,\ ?'c::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ bilinear\ h \longrightarrow vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ h\ (f\ k)\ (vector\_sub\ (g\ (k + (1::nat)))\ (g\ k))) = (if\ m \leq n\ then\ vector\_sub\ (vector\_sub\ (h\ (f\ (n + (1::nat)))\ (g\ (n + (1::nat))))\ (h\ (f\ m)\ (g\ m)))\ (vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ h\ (vector\_sub\ (f\ (k + (1::nat)))\ (f\ k))\ (g\ (k + (1::nat))))) \ else\ vec\ (0::nat))$

thm BILINEAR_VSUM_PARTIAL_PRE:

$\forall (f::nat \Rightarrow (real,\ ?'c::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ bilinear\ h \longrightarrow vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ h\ (f\ k)\ (vector\_sub\ (g\ k)\ (g\ (k - (1::nat))))) = (if\ m \leq n\ then\ vector\_sub\ (vector\_sub\ (h\ (f\ (n + (1::nat)))\ (g$

$n))$ $(h\ (f\ m)\ (g\ (m - (1::nat)))))$ $(vsum\ (dotdot\ m\ n)\ (\lambda k::nat.\ h\ (vector\_sub\ (f\ (k + (1::nat)))\ (f\ k))\ (g\ k)))\ else\ vec\ (0::nat))$

thm DEF_adjoint:

$adjoint = (\lambda\_111933::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ SOME\ f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ (\_111933\ x)\ y = dot\ x\ (f'\ y))$

thm adjoint:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ adjoint\ f = (SOME\ f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ (f\ x)\ y = dot\ x\ (f'\ y))$

thm ADJOINT_WORKS:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ (f\ x)\ y = dot\ x\ (adjoint\ f\ y))$

thm ADJOINT_LINEAR:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow linear\ (adjoint\ f)$

thm ADJOINT_CLAUSES:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ x\ (adjoint\ f\ y) = dot\ (f\ x)\ y) \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ (adjoint\ f\ y)\ x = dot\ y\ (f\ x))$

thm ADJOINT_ADJOINT:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow adjoint\ (adjoint\ f) = f$

thm ADJOINT_UNIQUE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ dot\ (f'\ x)\ y = dot\ x\ (f\ y)) \longrightarrow f' = adjoint\ f$

thm DEF_%%:

$\%\% = (\lambda(\_112100::real)\ \_112101::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ \_112100 * \$\ (\$\ \_112101\ i)\ j)))$

thm matrix_cmul:

$\forall (c::real)\ A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ \%\%\ c\ A = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ c * \$\ (\$\ A\ i)\ j))$

thm DEF_matrix_neg:

$matrix\_neg = (\lambda\_112112::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ - \$\ (\$\ \_112112\ i)\ j)))$

thm matrix_neg:

$\forall A$::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*. *matrix_neg* *A* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. − \$ (\$ *A* *i*) *j*))

thm DEF_matrix_add:

matrix_add = (λ(*_112117*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *_112118*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (\$ *_112117* *i*) *j* + \$ (\$ *_112118* *i*) *j*)))

thm matrix_add:

$\forall$ (*A*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *B*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*. *matrix_add* *A* *B* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (\$ *A* *i*) *j* + \$ (\$ *B* *i*) *j*))

thm DEF_matrix_sub:

matrix_sub = (λ(*_112129*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *_112130*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (\$ *_112129* *i*) *j* − \$ (\$ *_112130* *i*) *j*)))

thm matrix_sub:

$\forall$ (*A*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *B*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*. *matrix_sub* *A* *B* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (\$ *A* *i*) *j* − \$ (\$ *B* *i*) *j*))

thm DEF_matrix_mul:

matrix_mul = (λ(*_112141*::((*real*, ?'*c*::*type*) *cart*, ?'*b*::*type*) *cart*) *_112142*::((*real*, ?'*a*::*type*) *cart*, ?'*c*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex* *HOL_Light_Import*.*UNIV*)) (λ*k*::*nat*. \$ (\$ *_112141* *i*) *k* ∗ \$ (\$ *_112142* *k*) *j*))))

thm matrix_mul:

$\forall$ (*A*::((*real*, ?'*c*::*type*) *cart*, ?'*b*::*type*) *cart*) *B*::((*real*, ?'*a*::*type*) *cart*, ?'*c*::*type*) *cart*. *matrix_mul* *A* *B* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex* *HOL_Light_Import*.*UNIV*)) (λ*k*::*nat*. \$ (\$ *A* *i*) *k* ∗ \$ (\$ *B* *k*) *j*)))

thm DEF_matrix_vector_mul:

matrix_vector_mul = (λ(*_112153*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *_112154*::(*real*, ?'*b*::*type*) *cart*. *lambda* (λ*i*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex* *HOL_Light_Import*.*UNIV*)) (λ*j*::*nat*. \$ (\$ *_112153* *i*) *j* ∗ \$ *_112154* *j*)))

thm matrix_vector_mul:

$\forall$ (*A*::((*real*, ?'*b*::*type*) *cart*, ?'*a*::*type*) *cart*) *x*::(*real*, ?'*b*::*type*) *cart*. *matrix_vector_mul* *A* *x* = *lambda* (λ*i*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex* *HOL_Light_Import*.*UNIV*)) (λ*j*::*nat*. \$ (\$ *A* *i*) *j* ∗ \$ *x* *j*))

thm DEF_vector_matrix_mul:

vector_matrix_mul = (λ(*_112165*::(*real*, *?'b*::*type*) *cart*) *_112166*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *lambda* (λ*j*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. $ ($ *_112166 i*) *j* ∗ $ *_112165 i*)))

thm vector_matrix_mul:

∀ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *vector_matrix_mul x A* = *lambda* (λ*j*::*nat*. *sum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. $ ($ *A i*) *j* ∗ $ *x i*))

thm DEF_mat:

mat = (λ*_112177*::*nat*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if i* = *j then real_of_nat _112177 else* (*0*::*real*))))

thm mat:

∀ *k*::*nat*. *mat k* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if i* = *j then real_of_nat k else* (*0*::*real*)))

thm DEF_transp:

HOL_Light_Import.transp = (λ*_112182*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. $ ($ *_112182 j*) *i*)))

thm transp:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *HOL_Light_Import.transp A* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. $ ($ *A j*) *i*))

thm DEF_row:

row = (λ(*_112187*::*nat*) *_112188*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *lambda* ($ ($ *_112188 _112187*)))

thm row:

∀ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *i*::*nat*. *row i A* = *lambda* ($ ($ *A i*))

thm DEF_column:

column = (λ(*_112199*::*nat*) *_112200*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *lambda* (λ*i*::*nat*. $ ($ *_112200 i*) *_112199*))

thm column:

∀ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *j*::*nat*. *column j A* = *lambda* (λ*i*::*nat*. $ ($ *A i*) *j*)

thm DEF_rows:

rows = (λ*_112211*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *GSPEC* (λ*GEN%PVAR%287*::(*real*, *?'b*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%287* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row i _112211*)))

thm rows:

$\forall A$::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *rows A* = *GSPEC* (λ*GEN%PVAR%287*::(*real*, *?'b*::*type*) *cart*. $\exists$ *i*::*nat*. *SETSPEC GEN%PVAR%287* ((*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV*) (*row i A*))

thm DEF_columns:

columns = (λ*_112216*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *GSPEC* (λ*GEN%PVAR%288*::(*real*, *?'a*::*type*) *cart*. $\exists$ *i*::*nat*. *SETSPEC GEN%PVAR%288* ((*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV*) (*column i _112216*)))

thm columns:

$\forall A$::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *columns A* = *GSPEC* (λ*GEN%PVAR%288*::(*real*, *?'a*::*type*) *cart*. $\exists$ *i*::*nat*. *SETSPEC GEN%PVAR%288* ((*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV*) (*column i A*))

thm MATRIX_CMUL_COMPONENT:

$\forall$ (*c*::*real*) (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *i*::*nat*. \$ (\$ (%% *c A*) *i*) (*?j*::*nat*) = *c* * \$ (\$ *A i*) *?j*

thm MATRIX_ADD_COMPONENT:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. \$ (\$ (*matrix_add A B*) *i*) *j* = \$ (\$ *A i*) *j* + \$ (\$ *B i*) *j*

thm MATRIX_SUB_COMPONENT:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. \$ (\$ (*matrix_sub A B*) *i*) *j* = \$ (\$ *A i*) *j* − \$ (\$ *B i*) *j*

thm MATRIX_NEG_COMPONENT:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. \$ (\$ (*matrix_neg A*) *i*) *j* = − \$ (\$ *A i*) *j*

thm TRANSP_COMPONENT:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. \$ (\$ (*HOL_Light_Import.transp A*) *i*) *j* = \$ (\$ *A j*) *i*

thm MAT_COMPONENT:

$\forall$ (*n*::*nat*) (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ \$ (\$ (*mat n*) *i*) *j* = (*if i* = *j then real_of_nat n else* (*0*::*real*))

thm MATRIX_CMUL_ASSOC:

$\forall$ (*a*::*real*) (*b*::*real*) *X*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. %% *a* (%% *b X*) = %% (*a* * *b*) *X*

thm MATRIX_CMUL_LID:

$\forall X$::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. %% (*1*::*real*) *X* = *X*

thm MATRIX_ADD_SYM:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ A\ B = matrix\_add\ B\ A$

thm MATRIX_ADD_ASSOC:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ (B::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ C::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ A\ (matrix\_add\ B\ C) = matrix\_add\ (matrix\_add\ A\ B)\ C$

thm MATRIX_ADD_LID:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ (mat\ (0::nat))\ A = A$

thm MATRIX_ADD_RID:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ A\ (mat\ (0::nat)) = A$

thm MATRIX_ADD_LNEG:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ (matrix\_neg\ A)\ A = mat\ (0::nat)$

thm MATRIX_ADD_RNEG:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_add\ A\ (matrix\_neg\ A) = mat\ (0::nat)$

thm MATRIX_SUB:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_sub\ A\ B = matrix\_add\ A\ (matrix\_neg\ B)$

thm MATRIX_SUB_REFL:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_sub\ A\ A = mat\ (0::nat)$

thm MATRIX_ADD_LDISTRIB:

$\forall (A::((real, ?'c::type)\ cart, ?'b::type)\ cart)\ (B::((real, ?'a::type)\ cart, ?'c::type)\ cart)\ C::((real, ?'a::type)\ cart, ?'c::type)\ cart.\ matrix\_mul\ A\ (matrix\_add\ B\ C) = matrix\_add\ (matrix\_mul\ A\ B)\ (matrix\_mul\ A\ C)$

thm MATRIX_MUL_LID:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_mul\ (mat\ (1::nat))\ A = A$

thm MATRIX_MUL_RID:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ matrix\_mul\ A\ (mat\ (1::nat)) = A$

thm MATRIX_MUL_ASSOC:

$\forall (A::((real, ?'d::type)\ cart, ?'c::type)\ cart)\ (B::((real, ?'b::type)\ cart, ?'d::type)\ cart)\ C::((real, ?'a::type)\ cart, ?'b::type)\ cart.\ matrix\_mul\ A\ (matrix\_mul\ B\ C) = matrix\_mul\ (matrix\_mul\ A\ B)\ C$

thm MATRIX_MUL_LZERO:

$\forall$ *A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul* (*mat* (*0*::*nat*)) *A* = *mat* (*0*::*nat*)

thm MATRIX_MUL_RZERO:

$\forall$ *A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A* (*mat* (*0*::*nat*)) = *mat* (*0*::*nat*)

thm MATRIX_ADD_RDISTRIB:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) *C*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*. *matrix_mul* (*matrix_add A B*) *C* = *matrix_add* (*matrix_mul A C*) (*matrix_mul B C*)

thm MATRIX_SUB_LDISTRIB:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*) *C*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*. *matrix_mul A* (*matrix_sub B C*) = *matrix_sub* (*matrix_mul A B*) (*matrix_mul A C*)

thm MATRIX_SUB_RDISTRIB:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) *C*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*. *matrix_mul* (*matrix_sub A B*) *C* = *matrix_sub* (*matrix_mul A C*) (*matrix_mul B C*)

thm MATRIX_MUL_LMUL:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*) *c*::*real*. *matrix_mul* (%% *c A*) *B* = %% *c* (*matrix_mul A B*)

thm MATRIX_MUL_RMUL:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*) *c*::*real*. *matrix_mul A* (%% *c B*) = %% *c* (*matrix_mul A B*)

thm MATRIX_CMUL_ADD_LDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *c*::*real*. %% *c* (*matrix_add A B*) = *matrix_add* (%% *c A*) (%% *c B*)

thm MATRIX_CMUL_SUB_LDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *c*::*real*. %% *c* (*matrix_sub A B*) = *matrix_sub* (%% *c A*) (%% *c B*)

thm MATRIX_CMUL_ADD_RDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::*real*. %% (*b* + *c*) *A* = *matrix_add* (%% *b A*) (%% *c A*)

thm MATRIX_CMUL_SUB_RDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::*real*. %% (*b* − *c*) *A* = *matrix_sub* (%% *b A*) (%% *c A*)

thm MATRIX_CMUL_RZERO:

$\forall$ *c*::*real*. %% *c* (*mat* (*0*::*nat*)) = *mat* (*0*::*nat*)

thm MATRIX_CMUL_LZERO:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. %% (*0*::*real*) *A* = *mat* (*0*::*nat*)

thm MATRIX_NEG_MINUS1:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_neg* *A* = %% (− (*1*::*real*)) *A*

thm MATRIX_ADD_AC:

matrix_add (*?A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*?B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) = *matrix_add* *?B* *?A* ∧ *matrix_add* (*matrix_add* *?A* *?B*) (*?C*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) = *matrix_add* *?A* (*matrix_add* *?B* *?C*) ∧ *matrix_add* *?A* (*matrix_add* *?B* *?C*) = *matrix_add* *?B* (*matrix_add* *?A* *?C*)

thm MATRIX_NEG_ADD:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_neg* (*matrix_add* *A* *B*) = *matrix_add* (*matrix_neg* *A*) (*matrix_neg* *B*)

thm MATRIX_NEG_SUB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_neg* (*matrix_sub* *A* *B*) = *matrix_sub* *B* *A*

thm MATRIX_NEG_0:

matrix_neg (*mat* (*0*::*nat*)) = *mat* (*0*::*nat*)

thm MATRIX_SUB_RZERO:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_sub* *A* (*mat* (*0*::*nat*)) = *A*

thm MATRIX_SUB_LZERO:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_sub* (*mat* (*0*::*nat*)) *A* = *matrix_neg* *A*

thm MATRIX_NEG_EQ_0:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (*matrix_neg* *A* = *mat* (*0*::*nat*)) = (*A* = *mat* (*0*::*nat*))

thm MATRIX_VECTOR_MUL_ASSOC:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) (*B*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *matrix_vector_mul* *A* (*matrix_vector_mul* *B* *x*) = *matrix_vector_mul* (*matrix_mul* *A* *B*) *x*

thm MATRIX_VECTOR_MUL_LID:

$\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *matrix_vector_mul* (*mat* (*1*::*nat*)) *x* = *x*

thm MATRIX_VECTOR_MUL_LZERO:

$\forall x$::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul* (*mat* (*0*::*nat*)) *x* = *vec* (*0*::*nat*)

thm MATRIX_VECTOR_MUL_RZERO:

$\forall A$::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_vector_mul A* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*)

thm MATRIX_VECTOR_MUL_ADD_LDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A* (*vector_add x y*) = *vector_add* (*matrix_vector_mul A x*) (*matrix_vector_mul A y*)

thm MATRIX_VECTOR_MUL_SUB_LDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A* (*vector_sub x y*) = *vector_sub* (*matrix_vector_mul A x*) (*matrix_vector_mul A y*)

thm MATRIX_VECTOR_MUL_ADD_RDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *x*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul* (*matrix_add A B*) *x* = *vector_add* (*matrix_vector_mul A x*) (*matrix_vector_mul B x*)

thm MATRIX_VECTOR_MUL_SUB_RDISTRIB:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *x*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul* (*matrix_sub A B*) *x* = *vector_sub* (*matrix_vector_mul A x*) (*matrix_vector_mul B x*)

thm MATRIX_VECTOR_MUL_RMUL:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *c*::*real*. *matrix_vector_mul A* (% *c x*) = % *c* (*matrix_vector_mul A x*)

thm MATRIX_TRANSP_MUL:

$\forall$ (*A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*) *B*::((*real*, *?'a*::*type*) *cart*, *?'c*::*type*) *cart*. *HOL_Light_Import.transp* (*matrix_mul A B*) = *matrix_mul* (*HOL_Light_Import.transp B*) (*HOL_Light_Import.transp A*)

thm MATRIX_EQ:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (*A* = *B*) = ($\forall x$::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A x* = *matrix_vector_mul B x*)

thm MATRIX_VECTOR_MUL_COMPONENT:

$\forall$ (*A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *k*::*nat*. (*1*::*nat*) $\leq k \wedge k \leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ \$ (*matrix_vector_mul A x*) *k* = *dot* (\$ *A k*) *x*

thm DOT_LMUL_MATRIX:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'b::type)\ cart.\ dot\ (vector\_matrix\_mul\ x\ A)\ y = dot\ x\ (matrix\_vector\_mul\ A\ y)$

thm TRANSP_MATRIX_CMUL:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ c::real.\ HOL\_Light\_Import.transp\ (\%\%\ c\ A) = \%\%\ c\ (HOL\_Light\_Import.transp\ A)$

thm TRANSP_MATRIX_ADD:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (matrix\_add\ A\ B) = matrix\_add\ (HOL\_Light\_Import.transp\ A)\ (HOL\_Light\_Import.transp\ B)$

thm TRANSP_MATRIX_SUB:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (matrix\_sub\ A\ B) = matrix\_sub\ (HOL\_Light\_Import.transp\ A)\ (HOL\_Light\_Import.transp\ B)$

thm TRANSP_MATRIX_NEG:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (matrix\_neg\ A) = matrix\_neg\ (HOL\_Light\_Import.transp\ A)$

thm TRANSP_MAT:

$\forall n::nat.\ HOL\_Light\_Import.transp\ (mat\ n) = mat\ n$

thm TRANSP_TRANSP:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (HOL\_Light\_Import.transp\ A) = A$

thm TRANSP_EQ:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ (HOL\_Light\_Import.transp\ A = HOL\_Light\_Import.transp\ B) = (A = B)$

thm ROW_TRANSP:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow row\ i\ (HOL\_Light\_Import.transp\ A) = column\ i\ A$

thm COLUMN_TRANSP:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow column\ i\ (HOL\_Light\_Import.transp\ A) = row\ i\ A$

thm ROWS_TRANSP:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rows\ (HOL\_Light\_Import.transp\ A) = columns\ A$

thm COLUMNS_TRANSP:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ columns\ (HOL\_Light\_Import.transp\ A) = rows\ A$

thm VECTOR_MATRIX_MUL_TRANSP:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ vector\_matrix\_mul\ x\ A = matrix\_vector\_mul\ (HOL\_Light\_Import.transp\ A)\ x$

thm MATRIX_VECTOR_MUL_TRANSP:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = vector\_matrix\_mul\ x\ (HOL\_Light\_Import.transp\ A)$

thm FINITE_ROWS:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ FINITE\ (rows\ A)$

thm FINITE_COLUMNS:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ FINITE\ (columns\ A)$

thm MATRIX_EQUAL_ROWS:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ (A = B) = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow row\ i\ A = row\ i\ B)$

thm MATRIX_EQUAL_COLUMNS:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ (A = B) = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow column\ i\ A = column\ i\ B)$

thm MATRIX_MUL_DOT:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = lambda\ (\lambda i::nat.\ dot\ (\$\ A\ i)\ x)$

thm MATRIX_MUL_VSUM:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = vsum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \%\ (\$\ x\ i)\ (column\ i\ A))$

thm VECTOR_COMPONENTWISE:

$\forall x::(real, ?'a::type)\ cart.\ x = lambda\ (\lambda j::nat.\ sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ x\ i * \$\ (basis\ i)\ j))$

thm LINEAR_COMPONENTWISE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall (x::(real, ?'b::type)\ cart)\ j::nat.\ (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV$

$\longrightarrow$ \$ (*f x*) *j* = *sum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. \$ *x i* ∗ \$ (*f* (*basis i*)) *j*))

thm DEF_invertible:

invertible = (λ*_112665*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. ∃ *A'*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul _112665 A'* = *mat* (*1*::*nat*) ∧ *matrix_mul A' _112665* = *mat* (*1*::*nat*))

thm invertible:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *invertible A* = (∃ *A'*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A A'* = *mat* (*1*::*nat*) ∧ *matrix_mul A' A* = *mat* (*1*::*nat*))

thm DEF_matrix_inv:

matrix_inv = (λ*_112670*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *SOME A'*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul _112670 A'* = *mat* (*1*::*nat*) ∧ *matrix_mul A' _112670* = *mat* (*1*::*nat*))

thm matrix_inv:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_inv A* = (*SOME A'*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A A'* = *mat* (*1*::*nat*) ∧ *matrix_mul A' A* = *mat* (*1*::*nat*))

thm MATRIX_INV:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *invertible A* $\longrightarrow$ *matrix_mul A* (*matrix_inv A*) = *mat* (*1*::*nat*) ∧ *matrix_mul* (*matrix_inv A*) *A* = *mat* (*1*::*nat*)

thm DEF_matrix:

matrix = (λ*_112675*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (*_112675* (*basis j*)) *i*)))

thm matrix:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *matrix f* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. \$ (*f* (*basis j*)) *i*))

thm MATRIX_VECTOR_MUL_LINEAR:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *linear* (*matrix_vector_mul A*)

thm MATRIX_WORKS:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* $\longrightarrow$ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul* (*matrix f*) *x* = *f x*)

thm MATRIX_VECTOR_MUL:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* $\longrightarrow$ *f* = *matrix_vector_mul* (*matrix f*)

thm MATRIX_OF_MATRIX_VECTOR_MUL:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ matrix\ (matrix\_vector\_mul\ A) = A$

thm MATRIX_COMPOSE:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow matrix\ (g \circ f) = matrix\_mul\ (matrix\ g)\ (matrix\ f)$

thm MATRIX_VECTOR_COLUMN:

$\forall (A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = vsum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \%\ (\$\ x\ i)\ (\$\ (HOL\_Light\_Import.transp\ A)\ i))$

thm MATRIX_MUL_COMPONENT:

$\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (matrix\_mul\ (?A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ (?B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart))\ i = matrix\_vector\_mul\ (HOL\_Light\_Import.transp\ ?B)\ (\$\ ?A\ i)$

thm ADJOINT_MATRIX:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ adjoint\ (matrix\_vector\_mul\ A) = matrix\_vector\_mul\ (HOL\_Light\_Import.transp\ A)$

thm MATRIX_ADJOINT:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow matrix\ (adjoint\ f) = HOL\_Light\_Import.transp\ (matrix\ f)$

thm MATRIX_ID:

$matrix\ (\lambda x::(real,\ ?'a::type)\ cart.\ x) = mat\ (1::nat)$

thm MATRIX_I:

$matrix\ id = mat\ (1::nat)$

thm LINEAR_EQ_MATRIX:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \wedge matrix\ f = matrix\ g \longrightarrow f = g$

thm MATRIX_SELF_ADJOINT:

$\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (adjoint\ f = f) = (HOL\_Light\_Import.transp\ (matrix\ f) = matrix\ f)$

thm LINEAR_MATRIX_EXISTS:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f = (\exists A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ f = matrix\_vector\_mul\ A)$

thm LINEAR_1:

$\forall f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart.\ linear\ f = (\exists c::real.\ f = \%\ c)$

thm DEF_onorm:

$onorm = (\lambda\_112722::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%289::real.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%289\ (vector\_norm\ x = (1::real))\ (vector\_norm\ (\_112722\ x)))))$

thm onorm:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ onorm\ f = HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%289::real.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%289\ (vector\_norm\ x = (1::real))\ (vector\_norm\ (f\ x))))$

thm NORM_BOUND_GENERALIZE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ b::real.\ linear\ f \longrightarrow (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ x = (1::real) \longrightarrow vector\_norm\ (f\ x) \leq b) = (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ (f\ x) \leq b * vector\_norm\ x)$

thm ONORM:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ (f\ x) \leq onorm\ f * vector\_norm\ x) \wedge (\forall b::real.\ (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ (f\ x) \leq b * vector\_norm\ x) \longrightarrow onorm\ f \leq b)$

thm ONORM_POS_LE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow (0::real) \leq onorm\ f$

thm ONORM_EQ_0:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow (onorm\ f = (0::real)) = (\forall x::(real, ?'b::type)\ cart.\ f\ x = vec\ (0::nat))$

thm ONORM_CONST:

$\forall y::(real, ?'b::type)\ cart.\ onorm\ (\lambda x::(real, ?'a::type)\ cart.\ y) = vector\_norm\ y$

thm ONORM_POS_LT:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow ((0::real) < onorm\ f) = (\neg\ (\forall x::(real, ?'b::type)\ cart.\ f\ x = vec\ (0::nat)))$

thm ONORM_COMPOSE:

$\forall (f::(real, ?'c::type)\ cart \Rightarrow (real, ?'b::type)\ cart)\ g::(real, ?'a::type)\ cart \Rightarrow (real, ?'c::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow onorm\ (f \circ g) \leq onorm\ f * onorm\ g$

thm ONORM_NEG_LEMMA:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow onorm\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_neg\ (f\ x)) \leq onorm\ f$

thm ONORM_NEG:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \longrightarrow onorm\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_neg\ (f\ x)) = onorm\ f$

thm ONORM_TRIANGLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow onorm\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x)) \leq onorm\ f + onorm\ g$

thm ONORM_TRIANGLE_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \wedge onorm\ f + onorm\ g \leq (?e::real) \longrightarrow onorm\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x)) \leq ?e$

thm ONORM_TRIANGLE_LT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \wedge onorm\ f + onorm\ g < (?e::real) \longrightarrow onorm\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x)) < ?e$

thm DEF_lift:

$lift = (\lambda\_114324::real.\ lambda\ (\lambda i::nat.\ \_114324))$

thm lift:

$\forall x::real.\ lift\ x = lambda\ (\lambda i::nat.\ x)$

thm DEF_drop:

$HOL\_Light\_Import.drop = (\lambda\_114329::(real,\ unit)\ cart.\ \$\ \_114329\ (1::nat))$

thm drop:

$\forall x::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ x = \$\ x\ (1::nat)$

thm LIFT_COMPONENT:

$\forall x::real.\ \$\ (lift\ x)\ (1::nat) = x$

thm LIFT_DROP:

$(\forall x::(real,\ unit)\ cart.\ lift\ (HOL\_Light\_Import.drop\ x) = x) \wedge (\forall x::real.\ HOL\_Light\_Import.drop\ (lift\ x) = x)$

thm LIFT_DROP_conjunct1:

$\forall x::real.\ HOL\_Light\_Import.drop\ (lift\ x) = x$

thm LIFT_DROP_conjunct0:

$\forall x::(real,\ unit)\ cart.\ lift\ (HOL\_Light\_Import.drop\ x) = x$

thm IMAGE_LIFT_DROP:

$(\forall s::(real,\ unit)\ cart \Rightarrow bool.\ IMAGE\ (lift \circ HOL\_Light\_Import.drop)\ s = s) \wedge (\forall s::real \Rightarrow bool.\ IMAGE\ (HOL\_Light\_Import.drop \circ lift)\ s = s)$

thm IN_IMAGE_LIFT_DROP:

$(\forall (x::(real, unit)\ cart)\ s::real \Rightarrow bool.\ IN\ x\ (IMAGE\ lift\ s) = IN\ (HOL\_Light\_Import.drop\ x)\ s) \wedge (\forall (x::real)\ s::(real, unit)\ cart \Rightarrow bool.\ IN\ x\ (IMAGE\ HOL\_Light\_Import.drop\ s) = IN\ (lift\ x)\ s)$

thm FORALL_LIFT:

$(\forall x::(real, unit)\ cart.\ (?P::(real, unit)\ cart \Rightarrow bool)\ x) = (\forall x::real.\ ?P\ (lift\ x))$

thm EXISTS_LIFT:

$(\exists x::(real, unit)\ cart.\ (?P::(real, unit)\ cart \Rightarrow bool)\ x) = (\exists x::real.\ ?P\ (lift\ x))$

thm FORALL_DROP:

$(\forall x::real.\ (?P::real \Rightarrow bool)\ x) = (\forall x::(real, unit)\ cart.\ ?P\ (HOL\_Light\_Import.drop\ x))$

thm EXISTS_DROP:

$(\exists x::real.\ (?P::real \Rightarrow bool)\ x) = (\exists x::(real, unit)\ cart.\ ?P\ (HOL\_Light\_Import.drop\ x))$

thm FORALL_LIFT_FUN:

$\forall P::(?'a::type \Rightarrow (real, unit)\ cart) \Rightarrow bool.\ (\forall f::?'a::type \Rightarrow (real, unit)\ cart.\ P\ f) = (\forall f::?'a::type \Rightarrow real.\ P\ (lift \circ f))$

thm FORALL_DROP_FUN:

$\forall P::(?'a::type \Rightarrow real) \Rightarrow bool.\ (\forall f::?'a::type \Rightarrow real.\ P\ f) = (\forall f::?'a::type \Rightarrow (real, unit)\ cart.\ P\ (HOL\_Light\_Import.drop \circ f))$

thm EXISTS_LIFT_FUN:

$\forall P::(?'a::type \Rightarrow (real, unit)\ cart) \Rightarrow bool.\ (\exists f::?'a::type \Rightarrow (real, unit)\ cart.\ P\ f) = (\exists f::?'a::type \Rightarrow real.\ P\ (lift \circ f))$

thm EXISTS_DROP_FUN:

$\forall P::(?'a::type \Rightarrow real) \Rightarrow bool.\ (\exists f::?'a::type \Rightarrow real.\ P\ f) = (\exists f::?'a::type \Rightarrow (real, unit)\ cart.\ P\ (HOL\_Light\_Import.drop \circ f))$

thm LIFT_EQ:

$\forall (x::real)\ y::real.\ (lift\ x = lift\ y) = (x = y)$

thm DROP_EQ:

$\forall (x::(real, unit)\ cart)\ y::(real, unit)\ cart.\ (HOL\_Light\_Import.drop\ x = HOL\_Light\_Import.drop\ y) = (x = y)$

thm LIFT_IN_IMAGE_LIFT:

$\forall (x::real)\ s::real \Rightarrow bool.\ IN\ (lift\ x)\ (IMAGE\ lift\ s) = IN\ x\ s$

thm LIFT_NUM:

$\forall\, n{::}nat.\ lift\ (real\_of\_nat\ n) = vec\ n$

thm LIFT_ADD:

$\forall\, (x{::}real)\ y{::}real.\ lift\ (x + y) = vector\_add\ (lift\ x)\ (lift\ y)$

thm LIFT_SUB:

$\forall\, (x{::}real)\ y{::}real.\ lift\ (x - y) = vector\_sub\ (lift\ x)\ (lift\ y)$

thm LIFT_CMUL:

$\forall\, (x{::}real)\ c{::}real.\ lift\ (c * x) = \%\ c\ (lift\ x)$

thm LIFT_NEG:

$\forall\, x{::}real.\ lift\ (-\ x) = vector\_neg\ (lift\ x)$

thm LIFT_EQ_CMUL:

$\forall\, x{::}real.\ lift\ x = \%\ x\ (vec\ (1{::}nat))$

thm LIFT_SUM:

$\forall\, (k{::}?'a{::}type \Rightarrow bool)\ x{::}?'a{::}type \Rightarrow real.\ FINITE\ k \longrightarrow lift\ (sum\ k\ x) = vsum\ k\ (lift \circ x)$

thm DROP_LAMBDA:

$\forall\, x{::}nat \Rightarrow real.\ HOL\_Light\_Import.drop\ (lambda\ x) = x\ (1{::}nat)$

thm DROP_VEC:

$\forall\, n{::}nat.\ HOL\_Light\_Import.drop\ (vec\ n) = real\_of\_nat\ n$

thm DROP_ADD:

$\forall\, (x{::}(real, unit)\ cart)\ y{::}(real, unit)\ cart.\ HOL\_Light\_Import.drop\ (vector\_add\ x\ y) = HOL\_Light\_Import.drop\ x + HOL\_Light\_Import.drop\ y$

thm DROP_SUB:

$\forall\, (x{::}(real, unit)\ cart)\ y{::}(real, unit)\ cart.\ HOL\_Light\_Import.drop\ (vector\_sub\ x\ y) = HOL\_Light\_Import.drop\ x - HOL\_Light\_Import.drop\ y$

thm DROP_CMUL:

$\forall\, (x{::}(real, unit)\ cart)\ c{::}real.\ HOL\_Light\_Import.drop\ (\%\ c\ x) = c * HOL\_Light\_Import.drop\ x$

thm DROP_NEG:

$\forall\, x{::}(real, unit)\ cart.\ HOL\_Light\_Import.drop\ (vector\_neg\ x) = -\ HOL\_Light\_Import.drop\ x$

thm DROP_VSUM:

$\forall\, (k{::}?'a{::}type \Rightarrow bool)\ x{::}?'a{::}type \Rightarrow (real, unit)\ cart.\ FINITE\ k \longrightarrow HOL\_Light\_Import.drop\ (vsum\ k\ x) = sum\ k\ (HOL\_Light\_Import.drop \circ x)$

thm ABS_DROP:

$\forall x::(real,\ unit)\ cart.\ vector\_norm\ x = |HOL\_Light\_Import.drop\ x|$

thm NORM_1_POS:

$\forall x::(real,\ unit)\ cart.\ (0::real) \leq HOL\_Light\_Import.drop\ x \longrightarrow vector\_norm\ x = HOL\_Light\_Import.drop\ x$

thm NORM_LIFT:

$\forall x::real.\ vector\_norm\ (lift\ x) = |x|$

thm DIST_LIFT:

$\forall (x::real)\ y::real.\ distance\ (lift\ x,\ lift\ y) = |x - y|$

thm LINEAR_VMUL_DROP:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ unit)\ cart)\ v::(real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow linear\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ (HOL\_Light\_Import.drop\ (f\ x))\ v)$

thm LINEAR_FROM_REALS:

$\forall f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow f = (\lambda x::(real,\ unit)\ cart.\ \%\ (HOL\_Light\_Import.drop\ x)\ (column\ (1::nat)\ (matrix\ f)))$

thm LINEAR_TO_REALS:

$\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart.\ linear\ f \longrightarrow f = (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (dot\ (row\ (1::nat)\ (matrix\ f))\ x))$

thm DROP_EQ_0:

$\forall x::(real,\ unit)\ cart.\ (HOL\_Light\_Import.drop\ x = (0::real)) = (x = vec\ (0::nat))$

thm VSUM_REAL:

$\forall (f::?'a::type \Rightarrow (real,\ unit)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ f = lift\ (sum\ s\ (HOL\_Light\_Import.drop \circ f))$

thm DROP_WLOG_LE:

$(\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ (?P::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow bool)\ x\ y = ?P\ y\ x) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow ?P\ x\ y) \longrightarrow (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ ?P\ x\ y)$

thm IMAGE_LIFT_UNIV:

$IMAGE\ lift\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm IMAGE_DROP_UNIV:

$IMAGE\ HOL\_Light\_Import.drop\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm SUM_VSUM:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ s\ f = HOL\_Light\_Import.drop\ (vsum\ s\ (lift \circ f))$

thm LINEAR_LIFT_DOT:

$\forall a::(real,\ ?'a::type)\ cart.\ linear\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (dot\ a\ x))$

thm LINEAR_LIFT_COMPONENT:

$\forall k::nat.\ linear\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (\$\ x\ k))$

thm LINEAR_FSTCART:

$linear\ fstcart$

thm LINEAR_SNDCART:

$linear\ sndcart$

thm FSTCART_VEC:

$\forall n::nat.\ fstcart\ (vec\ n) = vec\ n$

thm FSTCART_ADD:

$\forall (x::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart)\ y::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ fstcart\ (vector\_add\ x\ y) = vector\_add\ (fstcart\ x)\ (fstcart\ y)$

thm FSTCART_CMUL:

$\forall (x::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart)\ c::real.\ fstcart\ (\%\ c\ x) = \%\ c\ (fstcart\ x)$

thm FSTCART_NEG:

$\forall x::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ vector\_neg\ (fstcart\ x) = fstcart\ (vector\_neg\ x)$

thm FSTCART_SUB:

$\forall (x::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart)\ y::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ fstcart\ (vector\_sub\ x\ y) = vector\_sub\ (fstcart\ x)\ (fstcart\ y)$

thm FSTCART_VSUM:

$\forall (k::?'c::type \Rightarrow bool)\ x::?'c::type \Rightarrow (real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ FINITE\ k \longrightarrow fstcart\ (vsum\ k\ x) = vsum\ k\ (\lambda i::?'c::type.\ fstcart\ (x\ i))$

thm SNDCART_VEC:

$\forall n::nat.\ sndcart\ (vec\ n) = vec\ n$

thm SNDCART_ADD:

$\forall (x::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart)\ y::(real, (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ sndcart\ (vector\_add\ x\ y) = vector\_add\ (sndcart\ x)\ (sndcart\ y)$

thm SNDCART_CMUL:

$\forall (x::(real, (?'b::type, ?'a::type) finite\_sum) cart) c::real. sndcart (\% c x) = \% c (sndcart x)$

thm SNDCART_NEG:

$\forall x::(real, (?'b::type, ?'a::type) finite\_sum) cart. vector\_neg (sndcart x) = sndcart (vector\_neg x)$

thm SNDCART_SUB:

$\forall (x::(real, (?'b::type, ?'a::type) finite\_sum) cart) y::(real, (?'b::type, ?'a::type) finite\_sum) cart. sndcart (vector\_sub x y) = vector\_sub (sndcart x) (sndcart y)$

thm SNDCART_VSUM:

$\forall (k::?'c::type \Rightarrow bool) x::?'c::type \Rightarrow (real, (?'b::type, ?'a::type) finite\_sum) cart. FINITE k \longrightarrow sndcart (vsum k x) = vsum k (\lambda i::?'c::type. sndcart (x i))$

thm PASTECART_VEC:

$\forall n::nat. pastecart (vec n) (vec n) = vec n$

thm PASTECART_ADD:

$\forall (x1::(real, ?'b::type) cart) (y1::(real, ?'a::type) cart) (x2::(real, ?'b::type) cart) y2::(real, ?'a::type) cart. vector\_add (pastecart x1 y1) (pastecart x2 y2) = pastecart (vector\_add x1 x2) (vector\_add y1 y2)$

thm PASTECART_CMUL:

$\forall (x1::(real, ?'b::type) cart) (y1::(real, ?'a::type) cart) c::real. pastecart (\% c x1) (\% c y1) = \% c (pastecart x1 y1)$

thm PASTECART_NEG:

$\forall (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. pastecart (vector\_neg x) (vector\_neg y) = vector\_neg (pastecart x y)$

thm PASTECART_SUB:

$\forall (x1::(real, ?'b::type) cart) (y1::(real, ?'a::type) cart) (x2::(real, ?'b::type) cart) y2::(real, ?'a::type) cart. vector\_sub (pastecart x1 y1) (pastecart x2 y2) = pastecart (vector\_sub x1 x2) (vector\_sub y1 y2)$

thm PASTECART_VSUM:

$\forall (k::?'c::type \Rightarrow bool) (x::?'c::type \Rightarrow (real, ?'b::type) cart) y::?'c::type \Rightarrow (real, ?'a::type) cart. FINITE k \longrightarrow pastecart (vsum k x) (vsum k y) = vsum k (\lambda i::?'c::type. pastecart (x i) (y i))$

thm PASTECART_EQ_VEC:

$\forall (x::(real, ?'b::type) cart) (y::(real, ?'a::type) cart) n::nat. (pastecart x y = vec n) = (x = vec n \wedge y = vec n)$

thm NORM_FSTCART:

$\forall x::(real, (?'b::type, ?'a::type) finite\_sum) cart. vector\_norm (fstcart x) \le vector\_norm x$

thm DIST_FSTCART:

$\forall (x::(real, (?'b::type, ?'a::type) finite\_sum) cart) y::(real, (?'b::type, ?'a::type) finite\_sum) cart. distance (fstcart x, fstcart y) \le distance (x, y)$

thm NORM_SNDCART:

$\forall x::(real, (?'b::type, ?'a::type) finite\_sum) cart. vector\_norm (sndcart x) \le vector\_norm x$

thm DIST_SNDCART:

$\forall (x::(real, (?'b::type, ?'a::type) finite\_sum) cart) y::(real, (?'b::type, ?'a::type) finite\_sum) cart. distance (sndcart x, sndcart y) \le distance (x, y)$

thm DOT_PASTECART:

$\forall (x1::(real, ?'b::type) cart) (x2::(real, ?'a::type) cart) (y1::(real, ?'b::type) cart) y2::(real, ?'a::type) cart. dot (pastecart x1 x2) (pastecart y1 y2) = dot x1 y1 + dot x2 y2$

thm NORM_PASTECART:

$\forall (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. vector\_norm (pastecart x y) = sqrt ((vector\_norm x)^2 + (vector\_norm y)^2)$

thm NORM_PASTECART_LE:

$\forall (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. vector\_norm (pastecart x y) \le vector\_norm x + vector\_norm y$

thm NORM_LE_PASTECART:

$\forall (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector\_norm x \le vector\_norm (pastecart x y) \wedge vector\_norm y \le vector\_norm (pastecart x y)$

thm NORM_PASTECART_0:

$(\forall x::(real, ?'d::type) cart. vector\_norm (pastecart x (vec (0::nat))) = vector\_norm x) \wedge (\forall y::(real, ?'b::type) cart. vector\_norm (pastecart (vec (0::nat)) y) = vector\_norm y)$

thm NORM_PASTECART_0_conjunct1:

$\forall y::(real, ?'b::type) cart. vector\_norm (pastecart (vec (0::nat)) y) = vector\_norm y$

thm NORM_PASTECART_0_conjunct0:

$\forall x::(real, ?'b::type) cart. vector\_norm (pastecart x (vec (0::nat))) = vector\_norm x$

thm DIST_PASTECART_CANCEL:

$(\forall\,(x::(real,\ ?'d::type)\ cart)\ (x'::(real,\ ?'d::type)\ cart)\ y::(real,\ ?'c::type)\ cart.\ distance\ (pastecart\ x\ y,\ pastecart\ x'\ y) = distance\ (x,\ x')) \wedge (\forall\,(x::(real,\ ?'b::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ y'::(real,\ ?'a::type)\ cart.\ distance\ (pastecart\ x\ y,\ pastecart\ x\ y') = distance\ (y,\ y'))$

thm DEF_subspace:

$subspace = (\lambda\_114726::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ (vec\ (0::nat))\ \_114726 \wedge (\forall\,(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ \_114726 \wedge IN\ y\ \_114726 \longrightarrow IN\ (vector\_add\ x\ y)\ \_114726) \wedge (\forall\,(c::real)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_114726 \longrightarrow IN\ (\%\ c\ x)\ \_114726))$

thm subspace:

$\forall\,s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s = (IN\ (vec\ (0::nat))\ s \wedge (\forall\,(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow IN\ (vector\_add\ x\ y)\ s) \wedge (\forall\,(c::real)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (\%\ c\ x)\ s))$

thm DEF_span:

$span = hull\ subspace$

thm span:

$\forall\,s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ span\ s = hull\ subspace\ s$

thm DEF_dependent:

$dependent = (\lambda\_114736::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\,a::(real,\ ?'a::type)\ cart.\ IN\ a\ \_114736 \wedge IN\ a\ (span\ (DELETE\ \_114736\ a)))$

thm dependent:

$\forall\,s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dependent\ s = (\exists\,a::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge IN\ a\ (span\ (DELETE\ s\ a)))$

thm DEF_independent:

$independent = (\lambda\_114741::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ dependent\ \_114741)$

thm independent:

$\forall\,s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ independent\ s = (\neg\ dependent\ s)$

thm SUBSPACE_UNIV:

$subspace\ HOL\_Light\_Import.UNIV$

thm SUBSPACE_IMP_NONEMPTY:

$\forall\,s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow s \neq EMPTY$

thm SUBSPACE_0:

$subspace\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) \longrightarrow IN\ (vec\ (0::nat))\ ?s$

thm SUBSPACE_ADD:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge IN\ x\ s \wedge IN\ y\ s \longrightarrow IN\ (vector\_add\ x\ y)\ s$

thm SUBSPACE_MUL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge IN\ x\ s \longrightarrow IN\ (\%\ c\ x)\ s$

thm SUBSPACE_NEG:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge IN\ x\ s \longrightarrow IN\ (vector\_neg\ x)\ s$

thm SUBSPACE_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge IN\ x\ s \wedge IN\ y\ s \longrightarrow IN\ (vector\_sub\ x\ y)\ s$

thm SUBSPACE_VSUM:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::?'a::type \Rightarrow (real,\ ?'b::type)\ cart)\ t::?'a::type \Rightarrow bool.\ subspace\ s \wedge FINITE\ t \wedge (\forall x::?'a::type.\ IN\ x\ t \longrightarrow IN\ (f\ x)\ s) \longrightarrow IN\ (vsum\ t\ f)\ s$

thm SUBSPACE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge subspace\ s \longrightarrow subspace\ (IMAGE\ f\ s)$

thm SUBSPACE_LINEAR_PREIMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge subspace\ s \longrightarrow subspace\ (GSPEC\ (\lambda GEN\%PVAR\%292::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%292\ (IN\ (f\ x)\ s)\ x))$

thm SUBSPACE_TRIVIAL:

$subspace\ (INSERT\ (vec\ (0::nat))\ EMPTY)$

thm SUBSPACE_INTER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge subspace\ t \longrightarrow subspace\ (HOL\_Light\_Import.INTER\ s\ t)$

thm SUBSPACE_INTERS:

$\forall f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow subspace\ s) \longrightarrow subspace\ (INTERS\ f)$

thm LINEAR_INJECTIVE_0_SUBSPACE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge subspace\ s \longrightarrow (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = vec\ (0::nat) \longrightarrow x = vec\ (0::nat))$

thm SUBSPACE_UNION_CHAIN:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; subspace \; s \wedge subspace \; t \wedge subspace \; (HOL\_Light\_Import.UNION \; s \; t) \longrightarrow SUBSET \; s \; t \vee SUBSET \; t \; s$

thm SPAN_SPAN:

$\forall s::(real, ?'a::type) \; cart \Rightarrow bool. \; span \; (span \; s) = span \; s$

thm SPAN_MONO:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; SUBSET \; s \; t \longrightarrow SUBSET \; (span \; s) \; (span \; t)$

thm SUBSPACE_SPAN:

$\forall s::(real, ?'a::type) \; cart \Rightarrow bool. \; subspace \; (span \; s)$

thm SPAN_CLAUSES:

$(\forall (a::(real, ?'d::type) \; cart) \; s::(real, ?'d::type) \; cart \Rightarrow bool. \; IN \; a \; s \longrightarrow IN \; a \; (span \; s)) \wedge IN \; (vec \; (0::nat)) \; (span \; (?s::(real, ?'c::type) \; cart \Rightarrow bool)) \wedge (\forall (x::(real, ?'b::type) \; cart) \; (y::(real, ?'b::type) \; cart) \; s::(real, ?'b::type) \; cart \Rightarrow bool. \; IN \; x \; (span \; s) \wedge IN \; y \; (span \; s) \longrightarrow IN \; (vector\_add \; x \; y) \; (span \; s)) \wedge (\forall (x::(real, ?'a::type) \; cart) \; (c::real) \; s::(real, ?'a::type) \; cart \Rightarrow bool. \; IN \; x \; (span \; s) \longrightarrow IN \; (\% \; c \; x) \; (span \; s))$

thm SPAN_INDUCT:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; h::(real, ?'a::type) \; cart \Rightarrow bool. \; (\forall x::(real, ?'a::type) \; cart. \; IN \; x \; s \longrightarrow IN \; x \; h) \wedge subspace \; h \longrightarrow (\forall x::(real, ?'a::type) \; cart. \; IN \; x \; (span \; s) \longrightarrow h \; x)$

thm SPAN_EMPTY:

$span \; EMPTY = INSERT \; (vec \; (0::nat)) \; EMPTY$

thm INDEPENDENT_EMPTY:

$independent \; EMPTY$

thm INDEPENDENT_NONZERO:

$\forall s::(real, ?'a::type) \; cart \Rightarrow bool. \; independent \; s \longrightarrow \neg \; IN \; (vec \; (0::nat)) \; s$

thm INDEPENDENT_MONO:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; independent \; t \wedge SUBSET \; s \; t \longrightarrow independent \; s$

thm DEPENDENT_MONO:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; dependent \; s \wedge SUBSET \; s \; t \longrightarrow dependent \; t$

thm SPAN_SUBSPACE:

$\forall (b::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ b\ s \wedge SUBSET\ s\ (span\ b) \wedge subspace\ s \longrightarrow span\ b = s$

thm SPAN_INDUCT_ALT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ h\ (vec\ (0::nat)) \wedge (\forall (c::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge h\ y \longrightarrow h\ (vector\_add\ (\%\ c\ x)\ y)) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (span\ s) \longrightarrow h\ x)$

thm SPAN_SUPERSET:

$\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) \longrightarrow IN\ x\ (span\ ?s)$

thm SPAN_INC:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ (span\ s)$

thm SPAN_UNION_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (HOL\_Light\_Import.UNION\ (span\ s)\ (span\ t))\ (span\ (HOL\_Light\_Import.UNION\ s\ t))$

thm SPAN_UNIV:

$span\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm SPAN_0:

$IN\ (vec\ (0::nat))\ (span\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool))$

thm SPAN_ADD:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \wedge IN\ y\ (span\ s) \longrightarrow IN\ (vector\_add\ x\ y)\ (span\ s)$

thm SPAN_MUL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \longrightarrow IN\ (\%\ c\ x)\ (span\ s)$

thm SPAN_MUL_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ c \neq (0::real) \longrightarrow IN\ (\%\ c\ x)\ (span\ s) = IN\ x\ (span\ s)$

thm SPAN_NEG:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \longrightarrow IN\ (vector\_neg\ x)\ (span\ s)$

thm SPAN_NEG_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ (vector\_neg\ x)\ (span\ s) = IN\ x\ (span\ s)$

thm SPAN_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \wedge IN\ y\ (span\ s) \longrightarrow IN\ (vector\_sub\ x\ y)\ (span\ s)$

thm SPAN_VSUM:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::?'a::type \Rightarrow (real,\ ?'b::type)\ cart)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge (\forall x::?'a::type.\ IN\ x\ t \longrightarrow IN\ (f\ x)\ (span\ s)) \longrightarrow IN\ (vsum\ t\ f)\ (span\ s)$

thm SPAN_ADD_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ (span\ s) \longrightarrow IN\ (vector\_add\ x\ y)\ (span\ s) = IN\ y\ (span\ s)$

thm SPAN_EQ_SELF:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (span\ s = s) = subspace\ s$

thm SPAN_SUBSET_SUBSPACE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge subspace\ t \longrightarrow SUBSET\ (span\ s)\ t$

thm SPAN_CLAUSES_conjunct3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \longrightarrow IN\ (\%\ c\ x)\ (span\ s)$

thm SPAN_CLAUSES_conjunct2:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \wedge IN\ y\ (span\ s) \longrightarrow IN\ (vector\_add\ x\ y)\ (span\ s)$

thm SPAN_CLAUSES_conjunct1:

$IN\ (vec\ (0::nat))\ (span\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool))$

thm SPAN_CLAUSES_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ s \longrightarrow IN\ a\ (span\ s)$

thm SUBSPACE_TRANSLATION_SELF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ subspace\ s \wedge IN\ a\ s \longrightarrow IMAGE\ (vector\_add\ a)\ s = s$

thm SUBSPACE_TRANSLATION_SELF_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ subspace\ s \longrightarrow (IMAGE\ (vector\_add\ a)\ s = s) = IN\ a\ s$

thm SUBSPACE_SUMS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge subspace\ t \longrightarrow subspace\ (GSPEC\ (\lambda GEN\%PVAR\%293::(real,\ ?'a::type)\ cart.$

$\exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%293$ $(IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm SPAN_UNION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ span\ (HOL\_Light\_Import.UNION\ s\ t) = GSPEC\ (\lambda GEN\%PVAR\%294::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%294\ (IN\ x\ (span\ s) \wedge IN\ y\ (span\ t))\ (vector\_add\ x\ y))$

thm SPAN_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \longrightarrow span\ (IMAGE\ f\ s) = IMAGE\ f\ (span\ s)$

thm DEPENDENT_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow dependent\ (IMAGE\ f\ s) = dependent\ s$

thm DEPENDENT_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge dependent\ s \longrightarrow dependent\ (IMAGE\ f\ s)$

thm INDEPENDENT_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow independent\ (IMAGE\ f\ s) = independent\ s$

thm SPAN_BREAKDOWN:

$\forall (b::(real, ?'a::type)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ a::(real, ?'a::type)\ cart.\ IN\ b\ s \wedge IN\ a\ (span\ s) \longrightarrow (\exists k::real.\ IN\ (vector\_sub\ a\ (\%\ k\ b))\ (span\ (DELETE\ s\ b)))$

thm SPAN_BREAKDOWN_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ (?x::(real, ?'a::type)\ cart)\ (span\ (INSERT\ a\ s)) = (\exists k::real.\ IN\ (vector\_sub\ ?x\ (\%\ k\ a))\ (span\ s))$

thm SPAN_INSERT_0:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ span\ (INSERT\ (vec\ (0::nat))\ s) = span\ s$

thm SPAN_SING:

$\forall a::(real, ?'a::type)\ cart.\ span\ (INSERT\ a\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%296::(real, ?'a::type)\ cart.\ \exists u::real.\ SETSPEC\ GEN\%PVAR\%296\ (IN\ u\ HOL\_Light\_Import.UNIV)\ (\%\ u\ a))$

thm SPAN_2:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ span\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%297::(real,\ ?'a::type)\ cart.\ \exists (u::real)\ v::real.\ SETSPEC\ GEN\%PVAR\%297\ (IN\ u\ HOL\_Light\_Import.UNIV \wedge IN\ v\ HOL\_Light\_Import.UNIV)\ (vector\_add\ (\%\ u\ a)\ (\%\ v\ b)))$

thm SPAN_3:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ span\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%298::(real,\ ?'a::type)\ cart.\ \exists (u::real)\ (v::real)\ w::real.\ SETSPEC\ GEN\%PVAR\%298\ (IN\ u\ HOL\_Light\_Import.UNIV \wedge IN\ v\ HOL\_Light\_Import.UNIV \wedge IN\ w\ HOL\_Light\_Import.UNIV)\ (vector\_add\ (\%\ u\ a)\ (vector\_add\ (\%\ v\ b)\ (\%\ w\ c))))$

thm IN_SPAN_INSERT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ (span\ (INSERT\ b\ s)) \wedge \neg\ IN\ a\ (span\ s) \longrightarrow IN\ b\ (span\ (INSERT\ a\ s))$

thm IN_SPAN_DELETE:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ (span\ s) \wedge \neg\ IN\ a\ (span\ (DELETE\ s\ b)) \longrightarrow IN\ b\ (span\ (INSERT\ a\ (DELETE\ s\ b)))$

thm EQ_SPAN_INSERT_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ (vector\_sub\ x\ y)\ (span\ s) \longrightarrow span\ (INSERT\ x\ s) = span\ (INSERT\ y\ s)$

thm SPAN_TRANS:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (span\ s) \wedge IN\ y\ (span\ (INSERT\ x\ s)) \longrightarrow IN\ y\ (span\ s)$

thm SPAN_EXPLICIT:

$\forall p::(real,\ ?'a::type)\ cart \Rightarrow bool.\ span\ p = GSPEC\ (\lambda GEN\%PVAR\%299::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%299\ (\exists (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow real.\ FINITE\ s \wedge SUBSET\ s\ p \wedge vsum\ s\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = y)\ y)$

thm DEPENDENT_EXPLICIT:

$\forall p::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dependent\ p = (\exists (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow real.\ FINITE\ s \wedge SUBSET\ s\ p \wedge (\exists v::(real,\ ?'a::type)\ cart.\ IN\ v\ s \wedge u\ v \neq (0::real)) \wedge vsum\ s\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = vec\ (0::nat))$

thm DEPENDENT_FINITE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow dependent\ s = (\exists u::(real,\ ?'a::type)\ cart \Rightarrow real.\ (\exists v::(real,\ ?'a::type)\ cart.\ IN\ v\ s \wedge u\ v \neq (0::real)) \wedge vsum\ s\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = vec\ (0::nat))$

thm SPAN_FINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *span s* = *GSPEC* (λ*GEN%PVAR%302*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%302* (∃ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *vsum s* (λ*v*::(*real*, *?'a*::*type*) *cart*. % (*u v*) *v*) = *y*) *y*)

thm SPAN_STDBASIS:

span (*GSPEC* (λ*GEN%PVAR%303*::(*real*, *?'a*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%303* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*basis i*))) = *HOL_Light_Import.UNIV*

thm HAS_SIZE_STDBASIS:

HAS_SIZE (*GSPEC* (λ*GEN%PVAR%306*::(*real*, *?'a*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%306* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*basis i*))) (*dimindex HOL_Light_Import.UNIV*)

thm FINITE_STDBASIS:

FINITE (*GSPEC* (λ*GEN%PVAR%307*::(*real*, *?'a*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%307* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*basis i*)))

thm CARD_STDBASIS:

CARD (*GSPEC* (λ*GEN%PVAR%308*::(*real*, *?'a*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%308* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*basis i*))) = *dimindex HOL_Light_Import.UNIV*

thm IN_SPAN_IMAGE_BASIS:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *IN x* (*span* (*IMAGE basis s*)) = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ ¬ *IN i s* ⟶ $ *x i* = (*0*::*real*))

thm INDEPENDENT_STDBASIS:

independent (*GSPEC* (λ*GEN%PVAR%313*::(*real*, *?'a*::*type*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%313* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*basis i*)))

thm INDEPENDENT_INSERT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *independent* (*INSERT a s*) = (*if IN a s then independent s else independent s* ∧ ¬ *IN a* (*span s*))

thm SPANNING_SUBSET_INDEPENDENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *independent s* ∧ *SUBSET s* (*span t*) ⟶ *s* = *t*

thm EXCHANGE_LEMMA:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE t* ∧ *independent s* ∧ *SUBSET s* (*span t*) ⟶ (∃ *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*.

$HAS\_SIZE\ t'\ (CARD\ t) \wedge SUBSET\ s\ t' \wedge SUBSET\ t'\ (HOL\_Light\_Import.UNION\ s\ t) \wedge SUBSET\ s\ (span\ t'))$

thm INDEPENDENT_SPAN_BOUND:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ t \wedge independent\ s \wedge SUBSET\ s\ (span\ t) \longrightarrow FINITE\ s \wedge CARD\ s \leq CARD\ t$

thm INDEPENDENT_BOUND:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ independent\ s \longrightarrow FINITE\ s \wedge CARD\ s \leq dimindex\ HOL\_Light\_Import.UNIV$

thm DEPENDENT_BIGGERSET:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (FINITE\ s \longrightarrow dimindex\ HOL\_Light\_Import.UNIV < CARD\ s) \longrightarrow dependent\ s$

thm INDEPENDENT_IMP_FINITE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ independent\ s \longrightarrow FINITE\ s$

thm INDEPENDENT_EXPLICIT:

$\forall b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ independent\ b = (FINITE\ b \wedge (\forall c::(real,\ ?'a::type)\ cart \Rightarrow real.\ vsum\ b\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (c\ v)\ v) = vec\ (0::nat) \longrightarrow (\forall v::(real,\ ?'a::type)\ cart.\ IN\ v\ b \longrightarrow c\ v = (0::real))))$

thm INDEPENDENT_2:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (x::real)\ y::real.\ independent\ (INSERT\ a\ (INSERT\ b\ EMPTY)) \wedge a \neq b \longrightarrow (vector\_add\ (\%\ x\ a)\ (\%\ y\ b) = vec\ (0::nat)) = (x = (0::real) \wedge y = (0::real))$

thm INDEPENDENT_3:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ (x::real)\ (y::real)\ z::real.\ independent\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \wedge a \neq b \wedge a \neq c \wedge b \neq c \longrightarrow (vector\_add\ (\%\ x\ a)\ (vector\_add\ (\%\ y\ b)\ (\%\ z\ c)) = vec\ (0::nat)) = (x = (0::real) \wedge y = (0::real) \wedge z = (0::real))$

thm MAXIMAL_INDEPENDENT_SUBSET_EXTEND:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ v \wedge independent\ s \longrightarrow (\exists b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ b \wedge SUBSET\ b\ v \wedge independent\ b \wedge SUBSET\ v\ (span\ b))$

thm MAXIMAL_INDEPENDENT_SUBSET:

$\forall v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge independent\ b \wedge SUBSET\ v\ (span\ b)$

thm DEF_dim:

$dim = (\lambda\_124538::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SOME\ n::nat.\ \exists b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ b\ \_124538 \wedge independent\ b \wedge SUBSET\ \_124538\ (span\ b) \wedge HAS\_SIZE\ b\ n)$

thm dim:

$\forall\, v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ dim\ v = (SOME\ n{::}nat.\ \exists\, b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge independent\ b \wedge SUBSET\ v\ (span\ b) \wedge HAS\_SIZE\ b\ n)$

thm BASIS_EXISTS:

$\forall\, v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ \exists\, b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge independent\ b \wedge SUBSET\ v\ (span\ b) \wedge HAS\_SIZE\ b\ (dim\ v)$

thm BASIS_EXISTS_FINITE:

$\forall\, v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ \exists\, b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ FINITE\ b \wedge SUBSET\ b\ v \wedge independent\ b \wedge SUBSET\ v\ (span\ b) \wedge HAS\_SIZE\ b\ (dim\ v)$

thm BASIS_SUBSPACE_EXISTS:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow (\exists\, b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ FINITE\ b \wedge SUBSET\ b\ s \wedge independent\ b \wedge span\ b = s \wedge HAS\_SIZE\ b\ (dim\ s))$

thm INDEPENDENT_CARD_LE_DIM:

$\forall\, (v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge independent\ b \longrightarrow FINITE\ b \wedge CARD\ b \leq dim\ v$

thm SPAN_CARD_GE_DIM:

$\forall\, (v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ v\ (span\ b) \wedge FINITE\ b \longrightarrow dim\ v \leq CARD\ b$

thm BASIS_CARD_EQ_DIM:

$\forall\, (v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge SUBSET\ v\ (span\ b) \wedge independent\ b \longrightarrow FINITE\ b \wedge CARD\ b = dim\ v$

thm BASIS_HAS_SIZE_DIM:

$\forall\, (v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ independent\ b \wedge span\ b = v \longrightarrow HAS\_SIZE\ b\ (dim\ v)$

thm DIM_UNIQUE:

$\forall\, (v{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ b{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SUBSET\ b\ v \wedge SUBSET\ v\ (span\ b) \wedge independent\ b \wedge HAS\_SIZE\ b\ (?n{::}nat) \longrightarrow dim\ v = ?n$

thm DIM_LE_CARD:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow dim\ s \leq CARD\ s$

thm DIM_UNIV:

$dim\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV$

thm DIM_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET s t* $\longrightarrow$ *dim s* $\leq$ *dim t*

thm DIM_SUBSET_UNIV:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *dim s* $\leq$ *dimindex HOL_Light_Import.UNIV*

thm BASIS_HAS_SIZE_UNIV:

$\forall$ *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *independent b* $\wedge$ *span b* = *HOL_Light_Import.UNIV* $\longrightarrow$ *HAS_SIZE b* (*dimindex HOL_Light_Import.UNIV*)

thm CARD_GE_DIM_INDEPENDENT:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET b v* $\wedge$ *independent b* $\wedge$ *dim v* $\leq$ *CARD b* $\longrightarrow$ *SUBSET v* (*span b*)

thm CARD_LE_DIM_SPANNING:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET v* (*span b*) $\wedge$ *FINITE b* $\wedge$ *CARD b* $\leq$ *dim v* $\longrightarrow$ *independent b*

thm CARD_EQ_DIM:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET b v* $\wedge$ *HAS_SIZE b* (*dim v*) $\longrightarrow$ *independent b* = *SUBSET v* (*span b*)

thm INDEPENDENT_BOUND_GENERAL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *independent s* $\longrightarrow$ *FINITE s* $\wedge$ *CARD s* $\leq$ *dim s*

thm DEPENDENT_BIGGERSET_GENERAL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*FINITE s* $\longrightarrow$ *dim s* < *CARD s*) $\longrightarrow$ *dependent s*

thm DIM_SPAN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *dim* (*span s*) = *dim s*

thm DIM_INSERT_0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *dim* (*INSERT* (*vec* (*0*::*nat*)) *s*) = *dim s*

thm DIM_EQ_CARD:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *independent s* $\longrightarrow$ *dim s* = *CARD s*

thm SUBSET_LE_DIM:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET s* (*span t*) $\longrightarrow$ *dim s* $\leq$ *dim t*

thm SPAN_EQ_DIM:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *span s* = *span t* $\longrightarrow$ *dim s* = *dim t*

thm SPANS_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ *SUBSET v* (*span b*) $\longrightarrow$ *SUBSET* (*IMAGE f v*) (*span* (*IMAGE f b*))

thm DIM_LINEAR_IMAGE_LE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\longrightarrow$ *dim* (*IMAGE f s*) $\leq$ *dim s*

thm DIM_EMPTY:

dim EMPTY = (*0*::*nat*)

thm DIM_INSERT:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *dim* (*INSERT x s*) = (*if IN x* (*span s*) *then dim s else dim s* + (*1*::*nat*))

thm DIM_SING:

$\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *dim* (*INSERT x EMPTY*) = (*if x* = *vec* (*0*::*nat*) *then 0*::*nat else* (*1*::*nat*))

thm DIM_EQ_0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*dim s* = (*0*::*nat*)) = *SUBSET s* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*)

thm SPANNING_SURJECTIVE_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET HOL_Light_Import.UNIV* (*span s*) $\wedge$ *linear f* $\wedge$ ($\forall$ *y*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'b*::*type*) *cart*. *f x* = *y*) $\longrightarrow$ *SUBSET HOL_Light_Import.UNIV* (*span* (*IMAGE f s*))

thm INDEPENDENT_INJECTIVE_IMAGE_GEN:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *independent s* $\wedge$ *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x* (*span s*) $\wedge$ *IN y* (*span s*) $\wedge$ *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *independent* (*IMAGE f s*)

thm INDEPENDENT_INJECTIVE_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *independent s* $\wedge$ *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *independent* (*IMAGE f s*)

thm VECTOR_SUB_PROJECT_ORTHOGONAL:

$\forall$ (*b*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *dot b* (*vector_sub x* (% (*dot b x* / *dot b b*) *b*)) = (*0*::*real*)

thm BASIS_ORTHOGONAL:

$\forall\, b$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE b* $\longrightarrow$ ($\exists\, c$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE c* $\wedge$ *CARD c* $\leq$ *CARD b* $\wedge$ *span c* = *span b* $\wedge$ *pairwise orthogonal c*)

thm ORTHOGONAL_BASIS_EXISTS:

$\forall\, v$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists\, b$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *independent b* $\wedge$ *SUBSET b* (*span v*) $\wedge$ *SUBSET v* (*span b*) $\wedge$ *HAS_SIZE b* (*dim v*) $\wedge$ *pairwise orthogonal b*

thm SPAN_EQ:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*span s* = *span t*) = (*SUBSET s* (*span t*) $\wedge$ *SUBSET t* (*span s*))

thm LINEAR_INDEP_IMAGE_LEMMA:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ *FINITE b* $\wedge$ *independent* (*IMAGE f b*) $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x b* $\wedge$ *IN y b* $\wedge$ *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ ($\forall\, x$::(*real*, *?'b*::*type*) *cart*. *IN x* (*span b*) $\longrightarrow$ *f x* = *vec* (*0*::*nat*) $\longrightarrow$ *x* = *vec* (*0*::*nat*))

thm LINEAR_INDEPENDENT_EXTEND_LEMMA:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE b* $\longrightarrow$ *independent b* $\longrightarrow$ ($\exists\, g$::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x* (*span b*) $\wedge$ *IN y* (*span b*) $\longrightarrow$ *g* (*vector_add x y*) = *vector_add* (*g x*) (*g y*)) $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *c*::*real*. *IN x* (*span b*) $\longrightarrow$ *g* (% *c x*) = % *c* (*g x*)) $\wedge$ ($\forall\, x$::(*real*, *?'b*::*type*) *cart*. *IN x b* $\longrightarrow$ *g x* = *f x*))

thm LINEAR_INDEPENDENT_EXTEND:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *independent b* $\longrightarrow$ ($\exists\, g$::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *linear g* $\wedge$ ($\forall\, x$::(*real*, *?'b*::*type*) *cart*. *IN x b* $\longrightarrow$ *g x* = *f x*))

thm SUBSPACE_KERNEL:

$\forall\, f$::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *linear f* $\longrightarrow$ *subspace* (*GSPEC* (λ*GEN%PVAR%315*::(*real*, *?'b*::*type*) *cart*. $\exists\, x$::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%315* (*f x* = *vec* (*0*::*nat*)) *x*))

thm LINEAR_EQ_0_SPAN:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall\, x$::(*real*, *?'b*::*type*) *cart*. *IN x b* $\longrightarrow$ *f x* = *vec* (*0*::*nat*)) $\longrightarrow$ ($\forall\, x$::(*real*, *?'b*::*type*) *cart*. *IN x* (*span b*) $\longrightarrow$ *f x* = *vec* (*0*::*nat*))

thm LINEAR_EQ_0:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (b::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge SUBSET\ s\ (span\ b) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ b \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = vec\ (0::nat))$

thm LINEAR_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (b::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge linear\ g \wedge SUBSET\ s\ (span\ b) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ b \longrightarrow f\ x = g\ x) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x)$

thm LINEAR_EQ_STDBASIS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow f\ (basis\ i) = g\ (basis\ i)) \longrightarrow f = g$

thm BILINEAR_EQ:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (b::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (c::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ bilinear\ f \wedge bilinear\ g \wedge SUBSET\ s\ (span\ b) \wedge SUBSET\ (?t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (span\ c) \wedge (\forall (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ b \wedge IN\ y\ c \longrightarrow f\ x\ y = g\ x\ y) \longrightarrow (\forall (x::(real,\ ?'c::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ ?t \longrightarrow f\ x\ y = g\ x\ y)$

thm BILINEAR_EQ_STDBASIS:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ bilinear\ f \wedge bilinear\ g \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow f\ (basis\ i)\ (basis\ j) = g\ (basis\ i)\ (basis\ j)) \longrightarrow f = g$

thm LEFT_INVERTIBLE_TRANSP:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ (\exists B::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ matrix\_mul\ B\ (HOL\_Light\_Import.transp\ A) = mat\ (1::nat)) = (\exists B::((real,\ ?'a::type)\ cart,\ ?'b::type)\ cart.\ matrix\_mul\ A\ B = mat\ (1::nat))$

thm RIGHT_INVERTIBLE_TRANSP:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ (\exists B::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ matrix\_mul\ (HOL\_Light\_Import.transp\ A)\ B = mat\ (1::nat)) = (\exists B::((real,\ ?'a::type)\ cart,\ ?'b::type)\ cart.\ matrix\_mul\ B\ A = mat\ (1::nat))$

thm LINEAR_INJECTIVE_LEFT_INVERSE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ g \wedge g \circ f = id)$

thm LINEAR_SURJECTIVE_RIGHT_INVERSE:

$\forall f$::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ∧ (∀ *y*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *f x* = *y*) ⟶ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *linear g* ∧ *f* ∘ *g* = *id*)

thm MATRIX_LEFT_INVERTIBLE_INJECTIVE:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul B A* = *mat* (*1*::*nat*)) = (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A x* = *matrix_vector_mul A y* ⟶ *x* = *y*)

thm MATRIX_LEFT_INVERTIBLE_KER:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul B A* = *mat* (*1*::*nat*)) = (∀ *x*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A x* = *vec* (*0*::*nat*) ⟶ *x* = *vec* (*0*::*nat*))

thm MATRIX_RIGHT_INVERTIBLE_SURJECTIVE:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A B* = *mat* (*1*::*nat*)) = (∀ *y*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *matrix_vector_mul A x* = *y*)

thm MATRIX_LEFT_INVERTIBLE_INDEPENDENT_COLUMNS:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul B A* = *mat* (*1*::*nat*)) = (∀ *c*::*nat* ⇒ *real*. *vsum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. % (*c i*) (*column i A*)) = *vec* (*0*::*nat*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *c i* = (*0*::*real*)))

thm MATRIX_RIGHT_INVERTIBLE_INDEPENDENT_ROWS:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A B* = *mat* (*1*::*nat*)) = (∀ *c*::*nat* ⇒ *real*. *vsum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. % (*c i*) (*row i A*)) = *vec* (*0*::*nat*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *c i* = (*0*::*real*)))

thm MATRIX_RIGHT_INVERTIBLE_SPAN_COLUMNS:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul A B* = *mat* (*1*::*nat*)) = (*span* (*columns A*) = *HOL_Light_Import.UNIV*)

thm MATRIX_LEFT_INVERTIBLE_SPAN_ROWS:

∀ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *B*::((*real*, *?'a*::*type*) *cart*, *?'b*::*type*) *cart*. *matrix_mul B A* = *mat* (*1*::*nat*)) = (*span* (*rows A*) = *HOL_Light_Import.UNIV*)

thm LINEAR_INJECTIVE_IMP_SURJECTIVE:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ∧ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ (∀ *y*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *f x* = *y*)

thm LINEAR_SURJECTIVE_IMP_INJECTIVE:

$\forall f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ f\, x = y) \longrightarrow (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ f\, x = f\, y \longrightarrow x = y)$

thm LINEAR_SURJECTIVE_IFF_INJECTIVE:

$\forall f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \longrightarrow (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ f\, x = y) = (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ f\, x = f\, y \longrightarrow x = y)$

thm LEFT_RIGHT_INVERSE_EQ:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow ?'a::type)\ h::?'a::type \Rightarrow ?'a::type.\ f \circ g = id \wedge g \circ h = id \longrightarrow f = h$

thm ISOMORPHISM_EXPAND:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ (f \circ g = id \wedge g \circ f = id) = ((\forall x::?'a::type.\ f\ (g\ x) = x) \wedge (\forall x::?'b::type.\ g\ (f\, x) = x))$

thm LINEAR_INJECTIVE_ISOMORPHISM:

$\forall f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ f\, x = f\, y \longrightarrow x = y) \longrightarrow (\exists f'::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f' \wedge (\forall x::(real, ?'a::type)\ cart.\ f'\ (f\, x) = x) \wedge (\forall x::(real, ?'a::type)\ cart.\ f\ (f'\ x) = x))$

thm LINEAR_SURJECTIVE_ISOMORPHISM:

$\forall f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ f\, x = y) \longrightarrow (\exists f'::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f' \wedge (\forall x::(real, ?'a::type)\ cart.\ f'\ (f\, x) = x) \wedge (\forall x::(real, ?'a::type)\ cart.\ f\ (f'\ x) = x))$

thm LINEAR_INVERSE_LEFT:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ f'::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge linear\, f' \longrightarrow (f \circ f' = id) = (f' \circ f = id)$

thm LEFT_INVERSE_LINEAR:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ g::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge g \circ f = id \longrightarrow linear\ g$

thm RIGHT_INVERSE_LINEAR:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ g::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\, f \wedge f \circ g = id \longrightarrow linear\ g$

thm LEFT_RIGHT_INVERSE_LINEAR:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ g::(real, ?'a::type)\ cart \Rightarrow (real, ?'b::type)\ cart.\ linear\, f \wedge g \circ f = id \wedge f \circ g = id \longrightarrow linear\ g$

thm LINEAR_BIJECTIVE_LEFT_RIGHT_INVERSE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \longrightarrow (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ g \wedge (\forall x::(real,\ ?'b::type)\ cart.\ g\ (f\ x) = x) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ f\ (g\ y) = y))$

thm MATRIX_LEFT_RIGHT_INVERSE:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ A'::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ (matrix\_mul\ A\ A' = mat\ (1::nat)) = (matrix\_mul\ A'\ A = mat\ (1::nat))$

thm MATRIX_LEFT_INVERTIBLE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\exists B::((real,\ ?'a::type)\ cart,\ ?'b::type)\ cart.\ matrix\_mul\ B\ (matrix\ f) = mat\ (1::nat)) = (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ g \wedge g \circ f = id)$

thm MATRIX_RIGHT_INVERTIBLE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\exists B::((real,\ ?'a::type)\ cart,\ ?'b::type)\ cart.\ matrix\_mul\ (matrix\ f)\ B = mat\ (1::nat)) = (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ g \wedge f \circ g = id)$

thm INVERTIBLE_LEFT_INVERSE:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ invertible\ A = (\exists B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ matrix\_mul\ B\ A = mat\ (1::nat))$

thm INVERTIBLE_RIGHT_INVERSE:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ invertible\ A = (\exists B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ matrix\_mul\ A\ B = mat\ (1::nat))$

thm MATRIX_INVERTIBLE:

$\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow invertible\ (matrix\ f) = (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ g \wedge f \circ g = id \wedge g \circ f = id)$

thm LINEAR_INVERTIBLE_BOUNDED_BELOW_POS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge linear\ g \wedge g \circ f = id \longrightarrow (\exists B>0::real.\ \forall x::(real,\ ?'b::type)\ cart.\ B * vector\_norm\ x \le vector\_norm\ (f\ x))$

thm LINEAR_INVERTIBLE_BOUNDED_BELOW:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge linear\ g \wedge g \circ f = id \longrightarrow (\exists B::real.\ \forall x::(real,\ ?'b::type)\ cart.\ B * vector\_norm\ x \le vector\_norm\ (f\ x))$

thm LINEAR_INJECTIVE_BOUNDED_BELOW_POS:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\,f \land (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\,x = f\,y \longrightarrow x = y) \longrightarrow (\exists B>0::real.\ \forall x::(real, ?'b::type)\ cart.\ vector\_norm\ x * B \leq vector\_norm\ (f\,x))$

thm DIM_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \land (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow dim\ (IMAGE\ f\ s) = dim\ s$

thm DEF_rowvector:

$rowvector = (\lambda\_134811::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ lambda\ (\$\ \_134811)))$

thm rowvector:

$\forall v::(real, ?'a::type)\ cart.\ rowvector\ v = lambda\ (\lambda i::nat.\ lambda\ (\$\ v))$

thm DEF_columnvector:

$columnvector = (\lambda\_134816::(real, ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ \$\ \_134816\ i)))$

thm columnvector:

$\forall v::(real, ?'a::type)\ cart.\ columnvector\ v = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ \$\ v\ i))$

thm TRANSP_COLUMNVECTOR:

$\forall v::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (columnvector\ v) = rowvector\ v$

thm TRANSP_ROWVECTOR:

$\forall v::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.transp\ (rowvector\ v) = columnvector\ v$

thm DOT_ROWVECTOR_COLUMNVECTOR:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ v::(real, ?'b::type)\ cart.\ columnvector\ (matrix\_vector\_mul\ A\ v) = matrix\_mul\ A\ (columnvector\ v)$

thm DOT_MATRIX_PRODUCT:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ dot\ x\ y = \$\ (\$\ (matrix\_mul\ (rowvector\ x)\ (columnvector\ y))\ (1::nat))\ (1::nat)$

thm DOT_MATRIX_VECTOR_MUL:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ (B::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ dot\ (matrix\_vector\_mul\ A\ x)\ (matrix\_vector\_mul\ B\ y) = \$\ (\$\ (matrix\_mul\ (rowvector\ x)\ (matrix\_mul\ (matrix\_mul\ (HOL\_Light\_Import.transp\ A)\ B)\ (columnvector\ y)))\ (1::nat))\ (1::nat)$

thm MATRIX_VECTOR_MUL_IN_COLUMNSPACE:

∀ (A::((real, ?'b::type) cart, ?'a::type) cart) x::(real, ?'b::type) cart. IN (matrix_vector_mul A x) (span (columns A))

thm ORTHOGONAL_CLAUSES_conjunct9:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. orthogonal x a ∧ orthogonal y a ⟶ orthogonal (vector_sub x y) a

thm ORTHOGONAL_CLAUSES_conjunct8:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. orthogonal x a ∧ orthogonal y a ⟶ orthogonal (vector_add x y) a

thm ORTHOGONAL_CLAUSES_conjunct7:

∀ (a::(real, ?'a::type) cart) x::(real, ?'a::type) cart. orthogonal x a ⟶ orthogonal (vector_neg x) a

thm ORTHOGONAL_CLAUSES_conjunct6:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) c::real. orthogonal x a ⟶ orthogonal (% c x) a

thm ORTHOGONAL_CLAUSES_conjunct5:

∀ a::(real, ?'a::type) cart. orthogonal (vec (0::nat)) a

thm ORTHOGONAL_CLAUSES_conjunct4:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. orthogonal a x ∧ orthogonal a y ⟶ orthogonal a (vector_sub x y)

thm ORTHOGONAL_CLAUSES_conjunct3:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. orthogonal a x ∧ orthogonal a y ⟶ orthogonal a (vector_add x y)

thm ORTHOGONAL_CLAUSES_conjunct2:

∀ (a::(real, ?'a::type) cart) x::(real, ?'a::type) cart. orthogonal a x ⟶ orthogonal a (vector_neg x)

thm ORTHOGONAL_CLAUSES_conjunct1:

∀ (a::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) c::real. orthogonal a x ⟶ orthogonal a (% c x)

thm ORTHOGONAL_CLAUSES_conjunct0:

∀ a::(real, ?'a::type) cart. orthogonal a (vec (0::nat))

thm SUBSPACE_ORTHOGONAL_TO_VECTOR:

∀ x::(real, ?'a::type) cart. subspace (GSPEC (λGEN%PVAR%321::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%321 (orthogonal x y) y))

thm SUBSPACE_ORTHOGONAL_TO_VECTORS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ subspace\ (GSPEC\ (\lambda GEN\%PVAR\%322::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%322\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow orthogonal\ x\ y)\ y))$

thm ORTHOGONAL_TO_SPAN:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ s \longrightarrow orthogonal\ x\ y) \longrightarrow (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ (span\ s) \longrightarrow orthogonal\ x\ y)$

thm ORTHOGONAL_TO_SPAN_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ (span\ s) \longrightarrow orthogonal\ x\ y) = (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ s \longrightarrow orthogonal\ x\ y)$

thm ORTHOGONAL_TO_SPANS_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ (span\ s) \land IN\ y\ (span\ t) \longrightarrow orthogonal\ x\ y) = (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ s \land IN\ y\ t \longrightarrow orthogonal\ x\ y)$

thm ORTHOGONAL_NULLSPACE_ROWSPACE:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = vec\ (0::nat) \land IN\ y\ (span\ (rows\ A)) \longrightarrow orthogonal\ x\ y$

thm NULLSPACE_INTER_ROWSPACE:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ (matrix\_vector\_mul\ A\ x = vec\ (0::nat) \land IN\ x\ (span\ (rows\ A))) = (x = vec\ (0::nat))$

thm MATRIX_VECTOR_MUL_INJECTIVE_ON_ROWSPACE:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ IN\ x\ (span\ (rows\ A)) \land IN\ y\ (span\ (rows\ A)) \land matrix\_vector\_mul\ A\ x = matrix\_vector\_mul\ A\ y \longrightarrow x = y$

thm DIM_ROWS_LE_DIM_COLUMNS:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ dim\ (rows\ A) \leq dim\ (columns\ A)$

thm DEF_rank:

$rank = (\lambda\_135042::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ dim\ (columns\ \_135042))$

thm rank:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ A = dim\ (columns\ A)$

thm RANK_ROW:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ A = dim\ (rows\ A)$

thm RANK_TRANSP:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ (HOL\_Light\_Import.transp\ A) = rank\ A$

thm MATRIX_VECTOR_MUL_BASIS:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow matrix\_vector\_mul\ A\ (basis\ k) = column\ k\ A$

thm COLUMNS_IMAGE_BASIS:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ columns\ A = IMAGE\ (matrix\_vector\_mul\ A)\ (GSPEC\ (\lambda GEN\%PVAR\%325::(real, ?'b::type)\ cart.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%325\ ((1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV)\ (basis\ i)))$

thm RANK_DIM_IM:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ A = dim\ (IMAGE\ (matrix\_vector\_mul\ A)\ HOL\_Light\_Import.UNIV)$

thm DIM_EQ_SPAN:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge dim\ t \leq dim\ s \longrightarrow span\ s = span\ t$

thm DIM_EQ_FULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (dim\ s = dimindex\ HOL\_Light\_Import.UNIV) = (span\ s = HOL\_Light\_Import.UNIV)$

thm DIM_PSUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ PSUBSET\ (span\ s)\ (span\ t) \longrightarrow dim\ s < dim\ t$

thm RANK_BOUND:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ A \leq min\ (dimindex\ HOL\_Light\_Import.UNIV)\ (dimindex\ HOL\_Light\_Import.UNIV)$

thm FULL_RANK_INJECTIVE:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ (rank\ A = dimindex\ HOL\_Light\_Import.UNIV) = (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = matrix\_vector\_mul\ A\ y \longrightarrow x = y)$

thm FULL_RANK_SURJECTIVE:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ (rank\ A = dimindex\ HOL\_Light\_Import.UNIV) = (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = y)$

thm MATRIX_FULL_LINEAR_EQUATIONS:

$\forall (A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ rank\ A = dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists x::(real,\ ?'b::type)\ cart.\ matrix\_vector\_mul\ A\ x = b)$

thm MATRIX_NONFULL_LINEAR_EQUATIONS_EQ:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ (\exists x::(real,\ ?'b::type)\ cart.\ x \neq vec\ (0::nat) \wedge matrix\_vector\_mul\ A\ x = vec\ (0::nat)) = (rank\ A \neq dimindex\ HOL\_Light\_Import.UNIV)$

thm MATRIX_NONFULL_LINEAR_EQUATIONS:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ rank\ A \neq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists x::(real,\ ?'b::type)\ cart.\ x \neq vec\ (0::nat) \wedge matrix\_vector\_mul\ A\ x = vec\ (0::nat))$

thm MATRIX_TRIVIAL_LINEAR_EQUATIONS:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV < dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists x::(real,\ ?'b::type)\ cart.\ x \neq vec\ (0::nat) \wedge matrix\_vector\_mul\ A\ x = vec\ (0::nat))$

thm FORALL_DOT_EQ_0_conjunct1:

$\forall x::(real,\ ?'a::type)\ cart.\ (\forall y::(real,\ ?'a::type)\ cart.\ dot\ x\ y = (0::real)) = (x = vec\ (0::nat))$

thm FORALL_DOT_EQ_0_conjunct0:

$\forall y::(real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'a::type)\ cart.\ dot\ x\ y = (0::real)) = (y = vec\ (0::nat))$

thm RANK_EQ_0:

$\forall A::((real,\ ?'b::type)\ cart,\ ?'a::type)\ cart.\ (rank\ A = (0::nat)) = (A = mat\ (0::nat))$

thm RANK_0:

$rank\ (mat\ (0::nat)) = (0::nat)$

thm RANK_MUL_LE_RIGHT:

$\forall (A::((real,\ ?'c::type)\ cart,\ ?'b::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'c::type)\ cart.\ rank\ (matrix\_mul\ A\ B) \leq rank\ B$

thm RANK_MUL_LE_LEFT:

$\forall (A::((real,\ ?'c::type)\ cart,\ ?'b::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'c::type)\ cart.\ rank\ (matrix\_mul\ A\ B) \leq rank\ A$

thm ADJOINT_INJECTIVE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ adjoint\ f\ x = adjoint\ f\ y \longrightarrow x = y) = (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y)$

thm ADJOINT_SURJECTIVE:

$\forall f$::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ⟶ (∀ *y*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *adjoint f x* = *y*) = (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*)

thm ADJOINT_INJECTIVE_INJECTIVE:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ⟶ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *adjoint f x* = *adjoint f y* ⟶ *x* = *y*) = (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*)

thm ADJOINT_INJECTIVE_INJECTIVE_0:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ⟶ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *adjoint f x* = *vec* (*0*::*nat*) ⟶ *x* = *vec* (*0*::*nat*)) = (∀ *x*::(*real*, *?'a*::*type*) *cart*. *f x* = *vec* (*0*::*nat*) ⟶ *x* = *vec* (*0*::*nat*))

thm LINEAR_SINGULAR_INTO_HYPERPLANE:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ⟶ (¬ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*)) = (∃ *a*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *dot a* (*f x*) = (*0*::*real*)))

thm LINEAR_SINGULAR_IMAGE_HYPERPLANE:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ∧ ¬ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*IMAGE f s*) (*GSPEC* (λ*GEN%PVAR%326*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%326* (*dot a x* = (*0*::*real*)) *x*))))

thm LOWDIM_EXPAND_DIMENSION:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *n*::*nat*. *dim s* ≤ *n* ∧ *n* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ (∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dim t* = *n* ∧ *SUBSET* (*span s*) (*span t*))

thm LOWDIM_EXPAND_BASIS:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *n*::*nat*. *dim s* ≤ *n* ∧ *n* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ (∃ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HAS_SIZE b n* ∧ *independent b* ∧ *SUBSET* (*span s*) (*span b*))

thm SPAN_DELETE_0:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *span* (*DELETE s* (*vec* (*0*::*nat*))) = *span s*

thm SPAN_IMAGE_SCALE:

∀ (*c*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *c x* ≠ (*0*::*real*)) ⟶ *span* (*IMAGE* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*c x*) *x*) *s*) = *span s*

thm PAIRWISE_ORTHOGONAL_INDEPENDENT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ s \land \neg\ IN\ (vec\ (0::nat))\ s \longrightarrow independent\ s$

thm PAIRWISE_ORTHOGONAL_IMP_FINITE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ s \longrightarrow FINITE\ s$

thm GRAM_SCHMIDT_STEP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ pairwise\ orthogonal\ s \land IN\ x\ (span\ s) \longrightarrow orthogonal\ x\ (vector\_sub\ a\ (vsum\ s\ (\lambda b::(real,\ ?'a::type)\ cart.\ \%\ (dot\ b\ a\ /\ dot\ b\ b)\ b)))$

thm ORTHOGONAL_EXTENSION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ s \longrightarrow (\exists u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ (HOL\_Light\_Import.UNION\ s\ u) \land span\ (HOL\_Light\_Import.UNION\ s\ u) = span\ (HOL\_Light\_Import.UNION\ s\ t))$

thm ORTHOGONAL_EXTENSION_STRONG:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ s \longrightarrow (\exists u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ DISJOINT\ u\ (INSERT\ (vec\ (0::nat))\ s) \land pairwise\ orthogonal\ (HOL\_Light\_Import.UNION\ s\ u) \land span\ (HOL\_Light\_Import.UNION\ s\ u) = span\ (HOL\_Light\_Import.UNION\ s\ t))$

thm ORTHONORMAL_EXTENSION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ pairwise\ orthogonal\ s \land (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ x = (1::real)) \longrightarrow (\exists u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ DISJOINT\ u\ s \land pairwise\ orthogonal\ (HOL\_Light\_Import.UNION\ s\ u) \land (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ u \longrightarrow vector\_norm\ x = (1::real)) \land span\ (HOL\_Light\_Import.UNION\ s\ u) = span\ (HOL\_Light\_Import.UNION\ s\ t))$

thm VECTOR_IN_ORTHOGONAL_SPANNINGSET:

$\forall a::(real,\ ?'a::type)\ cart.\ \exists s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ s \land pairwise\ orthogonal\ s \land span\ s = HOL\_Light\_Import.UNIV$

thm VECTOR_IN_ORTHOGONAL_BASIS:

$\forall a::(real,\ ?'a::type)\ cart.\ a \neq vec\ (0::nat) \longrightarrow (\exists s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ s \land \neg\ IN\ (vec\ (0::nat))\ s \land pairwise\ orthogonal\ s \land independent\ s \land HAS\_SIZE\ s\ (dimindex\ HOL\_Light\_Import.UNIV) \land span\ s = HOL\_Light\_Import.UNIV)$

thm VECTOR_IN_ORTHONORMAL_BASIS:

$\forall a::(real,\ ?'a::type)\ cart.\ vector\_norm\ a = (1::real) \longrightarrow (\exists s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ s \land pairwise\ orthogonal\ s \land (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x$

s ⟶ *vector_norm x* = (*1*::*real*)) ∧ *independent s* ∧ *HAS_SIZE s* (*dimindex HOL_Light_Import.UNIV*) ∧ *span s* = *HOL_Light_Import.UNIV*)

thm ORTHOGONAL_SPANNINGSET_SUBSPACE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *subspace s* ⟶ (∃ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET b s* ∧ *pairwise orthogonal b* ∧ *span b* = *s*)

thm ORTHOGONAL_BASIS_SUBSPACE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *subspace s* ⟶ (∃ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *IN* (*vec* (*0*::*nat*)) *b* ∧ *SUBSET b s* ∧ *pairwise orthogonal b* ∧ *independent b* ∧ *HAS_SIZE b* (*dim s*) ∧ *span b* = *s*)

thm ORTHONORMAL_BASIS_SUBSPACE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *subspace s* ⟶ (∃ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET b* (*span s*) ∧ *pairwise orthogonal b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x b* ⟶ *vector_norm x* = (*1*::*real*)) ∧ *independent b* ∧ *HAS_SIZE b* (*dim s*) ∧ *span b* = *s*)

thm ORTHOGONAL_TO_SUBSPACE_EXISTS_GEN:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *PSUBSET* (*span s*) (*span t*) ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *x* ≠ *vec* (*0*::*nat*) ∧ *IN x* (*span t*) ∧ (∀ *y*::(*real*, *?'a*::*type*) *cart*. *IN y* (*span s*) ⟶ *orthogonal x y*))

thm ORTHOGONAL_TO_SUBSPACE_EXISTS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dim s* < *dimindex HOL_Light_Import.UNIV* ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *x* ≠ *vec* (*0*::*nat*) ∧ (∀ *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* ⟶ *orthogonal x y*))

thm ORTHOGONAL_TO_VECTOR_EXISTS:

∀ *x*::(*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ⟶ (∃ *y*::(*real*, *?'a*::*type*) *cart*. *y* ≠ *vec* (*0*::*nat*) ∧ *orthogonal x y*)

thm SPAN_NOT_UNIV_ORTHOGONAL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *span s* ≠ *HOL_Light_Import.UNIV* ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*span s*) ⟶ *dot a x* = (*0*::*real*)))

thm SPAN_NOT_UNIV_SUBSET_HYPERPLANE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *span s* ≠ *HOL_Light_Import.UNIV* ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *vec* (*0*::*nat*) ∧ *SUBSET* (*span s*) (*GSPEC* (λ*GEN%PVAR%329*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%329* (*dot a x* = (*0*::*real*)) *x*)))

thm LOWDIM_SUBSET_HYPERPLANE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dim s* < *dimindex HOL_Light_Import.UNIV* ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *vec* (*0*::*nat*) ∧ *SUBSET* (*span s*) (*GSPEC*

$(\lambda GEN\%PVAR\%330::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%330\ (dot\ a\ x = (0::real))\ x)))$

thm ORTHOGONAL_SUBSPACE_DECOMP_UNIQUE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (x'::(real,\ ?'a::type)\ cart)\ y'::(real,\ ?'a::type)\ cart.\ (\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ a\ s \land IN\ b\ t \longrightarrow orthogonal\ a\ b) \land IN\ x\ (span\ s) \land IN\ x'\ (span\ s) \land IN\ y\ (span\ t) \land IN\ y'\ (span\ t) \land vector\_add\ x\ y = vector\_add\ x'\ y' \longrightarrow x = x' \land y = y'$

thm ORTHOGONAL_SUBSPACE_DECOMP_EXISTS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ \exists (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ IN\ y\ (span\ s) \land (\forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ (span\ s) \longrightarrow orthogonal\ z\ w) \land x = vector\_add\ y\ z$

thm ORTHOGONAL_SUBSPACE_DECOMP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ Ex1\ (GABS\ (\lambda f::(real,\ ?'a::type)\ cart \times (real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ GEQ\ (f\ (y,\ z))\ (IN\ y\ (span\ s) \land IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%332::(real,\ ?'a::type)\ cart.\ \exists z::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%332\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (span\ s) \longrightarrow orthogonal\ z\ x)\ z)) \land x = vector\_add\ y\ z)))$

thm ISOMETRY_SUBSPACES:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \land subspace\ t \land dim\ s = dim\ t \longrightarrow (\exists f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \land IMAGE\ f\ s = t \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) = vector\_norm\ x))$

thm ISOMETRY_UNIV_SUBSPACE:

$\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ subspace\ s \land dimindex\ HOL\_Light\_Import.UNIV = dim\ s \longrightarrow (\exists f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \land IMAGE\ f\ HOL\_Light\_Import.UNIV = s \land (\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x))$

thm ISOMETRY_UNIV_SUPERSET_SUBSPACE:

$\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ subspace\ s \land dim\ s \le dimindex\ HOL\_Light\_Import.UNIV \land dimindex\ HOL\_Light\_Import.UNIV \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \land SUBSET\ s\ (IMAGE\ f\ HOL\_Light\_Import.UNIV) \land (\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x))$

thm ISOMETRY_UNIV_UNIV:

$dimindex\ HOL\_Light\_Import.UNIV \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \land (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x))$

thm SUBSPACE_ISOMORPHISM:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge subspace\ t \wedge dim\ s = dim\ t \longrightarrow (\exists f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge IMAGE\ f\ s = t \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y))$

thm ISOMORPHISMS_UNIV_UNIV:

$dimindex\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge linear\ g \wedge (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ vector\_norm\ (g\ y) = vector\_norm\ y) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ g\ (f\ x) = x) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ f\ (g\ y) = y))$

thm SUBSPACE_HYPERPLANE:

$\forall a::(real,\ ?'a::type)\ cart.\ subspace\ (GSPEC\ (\lambda GEN\%PVAR\%334::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%334\ (dot\ a\ x = (0::real))\ x))$

thm SUBSPACE_SPECIAL_HYPERPLANE:

$\forall k::nat.\ subspace\ (GSPEC\ (\lambda GEN\%PVAR\%335::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%335\ (\$\ x\ k = (0::real))\ x))$

thm SPECIAL_HYPERPLANE_SPAN:

$\forall k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%336::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%336\ (\$\ x\ k = (0::real))\ x) = span\ (IMAGE\ basis\ (DELETE\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ k))$

thm DIM_SPECIAL_HYPERPLANE:

$\forall k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow dim\ (GSPEC\ (\lambda GEN\%PVAR\%338::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%338\ (\$\ x\ k = (0::real))\ x)) = dimindex\ HOL\_Light\_Import.UNIV - (1::nat)$

thm DIM_IMAGE_KERNEL_GEN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge subspace\ s \longrightarrow dim\ (IMAGE\ f\ s) + dim\ (GSPEC\ (\lambda GEN\%PVAR\%347::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%347\ (IN\ x\ s \wedge f\ x = vec\ (0::nat))\ x)) = dim\ s$

thm DIM_IMAGE_KERNEL:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow dim\ (IMAGE\ f\ HOL\_Light\_Import.UNIV) + dim\ (GSPEC\ (\lambda GEN\%PVAR\%348::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%348\ (f\ x = vec\ (0::nat))\ x)) = dimindex\ HOL\_Light\_Import.UNIV$

thm DIM_SUMS_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge subspace\ t \longrightarrow dim\ (GSPEC\ (\lambda GEN\%PVAR\%351::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%351\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y))) + dim\ (HOL\_Light\_Import.INTER\ s\ t) = dim\ s + dim\ t$

thm DIM_KERNEL_COMPOSE:

$\forall (f::(real, ?'c::type)\ cart \Rightarrow (real, ?'b::type)\ cart)\ g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \wedge linear\ g \longrightarrow dim\ (GSPEC\ (\lambda GEN\%PVAR\%358::(real, ?'c::type)\ cart.\ \exists x::(real, ?'c::type)\ cart.\ SETSPEC\ GEN\%PVAR\%358\ ((g \circ f)\ x = vec\ (0::nat))\ x)) \leq dim\ (GSPEC\ (\lambda GEN\%PVAR\%359::(real, ?'c::type)\ cart.\ \exists x::(real, ?'c::type)\ cart.\ SETSPEC\ GEN\%PVAR\%359\ (f\ x = vec\ (0::nat))\ x)) + dim\ (GSPEC\ (\lambda GEN\%PVAR\%360::(real, ?'b::type)\ cart.\ \exists y::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%360\ (g\ y = vec\ (0::nat))\ y))$

thm DIM_ORTHOGONAL_SUM:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ t \longrightarrow dot\ x\ y = (0::real)) \longrightarrow dim\ (HOL\_Light\_Import.UNION\ s\ t) = dim\ s + dim\ t$

thm DIM_SUBSPACE_ORTHOGONAL_TO_VECTORS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \wedge subspace\ t \wedge SUBSET\ s\ t \longrightarrow dim\ (GSPEC\ (\lambda GEN\%PVAR\%361::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%361\ (IN\ y\ t \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow orthogonal\ x\ y))\ y)) + dim\ s = dim\ t$

thm RANK_NULLSPACE:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ A + dim\ (GSPEC\ (\lambda GEN\%PVAR\%362::(real, ?'b::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%362\ (matrix\_vector\_mul\ A\ x = vec\ (0::nat))\ x)) = dimindex\ HOL\_Light\_Import.UNIV$

thm RANK_SYLVESTER:

$\forall (A::((real, ?'c::type)\ cart, ?'b::type)\ cart)\ B::((real, ?'a::type)\ cart, ?'c::type)\ cart.\ rank\ A + rank\ B \leq rank\ (matrix\_mul\ A\ B) + dimindex\ HOL\_Light\_Import.UNIV$

thm RANK_GRAM:

$\forall A::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ (matrix\_mul\ (HOL\_Light\_Import.transp\ A)\ A) = rank\ A$

thm RANK_TRIANGLE:

$\forall (A::((real, ?'b::type)\ cart, ?'a::type)\ cart)\ B::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ rank\ (matrix\_add\ A\ B) \leq rank\ A + rank\ B$

thm DEF_infnorm:

infnorm = (*SOME infnorm*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *real*. ∀ (*_146457*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *infnorm _146457 x* = *HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%368*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%368* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ *x i*|))) (*48*::*nat*)

thm infnorm:

infnorm (*?x*::(*real*, *?'a*::*type*) *cart*) = *HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%368*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%368* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ *?x i*|))

thm NUMSEG_DIMINDEX_NONEMPTY:

∃ *i*::*nat*. *IN i* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*))

thm INFNORM_SET_IMAGE:

GSPEC (λ*GEN%PVAR%369*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%369* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ (*?x*::(*real*, *?'a*::*type*) *cart*) *i*|) = *IMAGE* (λ*i*::*nat*. |$ *?x i*|) (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*))

thm INFNORM_SET_LEMMA:

FINITE (*GSPEC* (λ*GEN%PVAR%370*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%370* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ (*?x*::(*real*, *?'a*::*type*) *cart*) *i*|)) ∧ *GSPEC* (λ*GEN%PVAR%371*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%371* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ *?x i*|) ≠ *EMPTY*

thm INFNORM_SET_LEMMA_conjunct1:

GSPEC (λ*GEN%PVAR%371*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%371* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ (*?x*::(*real*, *?'a*::*type*) *cart*) *i*|) ≠ *EMPTY*

thm INFNORM_SET_LEMMA_conjunct0:

FINITE (*GSPEC* (λ*GEN%PVAR%370*::*real*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%370* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) |$ (*?x*::(*real*, *?'a*::*type*) *cart*) *i*|))

thm INFNORM_POS_LE:

∀ *x*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) ≤ *infnorm x*

thm INFNORM_TRIANGLE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *infnorm* (*vector_add x y*) ≤ *infnorm x* + *infnorm y*

thm INFNORM_EQ_0:

∀ *x*::(*real*, *?'a*::*type*) *cart*. (*infnorm x* = (*0*::*real*)) = (*x* = *vec* (*0*::*nat*))

thm INFNORM_0:

infnorm (*vec* (*0*::*nat*)) = (*0*::*real*)

thm INFNORM_NEG:

$\forall x::(real,\ ?'a::type)\ cart.\ infnorm\ (vector\_neg\ x) = infnorm\ x$

thm INFNORM_SUB:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ infnorm\ (vector\_sub\ x\ y) = infnorm\ (vector\_sub\ y\ x)$

thm REAL_ABS_SUB_INFNORM:

$|infnorm\ (?x::(real,\ ?'a::type)\ cart) - infnorm\ (?y::(real,\ ?'a::type)\ cart)| \leq infnorm\ (vector\_sub\ ?x\ ?y)$

thm REAL_ABS_INFNORM:

$\forall x::(real,\ ?'a::type)\ cart.\ |infnorm\ x| = infnorm\ x$

thm COMPONENT_LE_INFNORM:

$\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow |\$\ x\ i| \leq infnorm\ x$

thm INFNORM_MUL_LEMMA:

$\forall (a::real)\ x::(real,\ ?'a::type)\ cart.\ infnorm\ (\%\ a\ x) \leq |a| * infnorm\ x$

thm INFNORM_MUL:

$\forall (a::real)\ x::(real,\ ?'a::type)\ cart.\ infnorm\ (\%\ a\ x) = |a| * infnorm\ x$

thm INFNORM_POS_LT:

$\forall x::(real,\ ?'a::type)\ cart.\ ((0::real) < infnorm\ x) = (x \neq vec\ (0::nat))$

thm INFNORM_LE_NORM:

$\forall x::(real,\ ?'a::type)\ cart.\ infnorm\ x \leq vector\_norm\ x$

thm NORM_LE_INFNORM:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ x \leq sqrt\ (real\_of\_nat\ (dimindex\ HOL\_Light\_Import.UNIV)) * infnorm\ x$

thm NORM_CAUCHY_SCHWARZ_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (dot\ x\ y = vector\_norm\ x * vector\_norm\ y) = (\%\ (vector\_norm\ x)\ y = \%\ (vector\_norm\ y)\ x)$

thm NORM_CAUCHY_SCHWARZ_ABS_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (|dot\ x\ y| = vector\_norm\ x * vector\_norm\ y) = (\%\ (vector\_norm\ x)\ y = \%\ (vector\_norm\ y)\ x \vee \%\ (vector\_norm\ x)\ y = \%\ (-\ vector\_norm\ y)\ x)$

thm NORM_TRIANGLE_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (vector\_norm\ (vector\_add\ x\ y) = vector\_norm\ x + vector\_norm\ y) = (\%\ (vector\_norm\ x)\ y = \%\ (vector\_norm\ y)\ x)$

thm DIST_TRIANGLE_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.$ $(distance\ (x,\ z) = distance\ (x,\ y) + distance\ (y,\ z)) = (\%\ (vector\_norm\ (vector\_sub\ x\ y))\ (vector\_sub\ y\ z) = \%\ (vector\_norm\ (vector\_sub\ y\ z))\ (vector\_sub\ x\ y))$

thm NORM_CROSS_MULTIPLY:

$\forall (a::real)\ (b::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ \%\ a\ x = \%\ b\ y \wedge (0::real) < a \wedge (0::real) < b \longrightarrow \%\ (vector\_norm\ y)\ x = \%\ (vector\_norm\ x)\ y$

thm DEF_collinear:

$collinear = (\lambda\_146885::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, u::(real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ \_146885 \wedge IN\ y\ \_146885 \longrightarrow (\exists\, c::real.\ vector\_sub\ x\ y = \%\ c\ u))$

thm collinear:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ s = (\exists\, u::(real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow (\exists\, c::real.\ vector\_sub\ x\ y = \%\ c\ u))$

thm COLLINEAR_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ t \wedge SUBSET\ s\ t \longrightarrow collinear\ s$

thm COLLINEAR_EMPTY:

$collinear\ EMPTY$

thm COLLINEAR_SING:

$\forall\, x::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ x\ EMPTY)$

thm COLLINEAR_2:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ EMPTY))$

thm COLLINEAR_SMALL:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \wedge CARD\ s \leq (2::nat) \longrightarrow collinear\ s$

thm COLLINEAR_3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ x\ y)\ (INSERT\ (vector\_sub\ z\ y)\ EMPTY)))$

thm COLLINEAR_LEMMA:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY))) = (x = vec\ (0::nat) \lor y = vec\ (0::nat) \lor (\exists c::real.\ y = \%\ c\ x))$

thm COLLINEAR_LEMMA_ALT:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY))) = (x = vec\ (0::nat) \lor (\exists c::real.\ y = \%\ c\ x))$

thm NORM_CAUCHY_SCHWARZ_EQUAL:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (|dot\ x\ y| = vector\_norm\ x * vector\_norm\ y) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY)))$

thm DOT_CAUCHY_SCHWARZ_EQUAL:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ ((dot\ x\ y)^2 = dot\ x\ x * dot\ y\ y) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY)))$

thm COLLINEAR_3_EXPAND:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) = (a = c \lor (\exists u::real.\ b = vector\_add\ (\%\ u\ a)\ (\%\ ((1::real) - u)\ c)))$

thm COLLINEAR_TRIPLES:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ a \neq b \longrightarrow collinear\ (INSERT\ a\ (INSERT\ b\ s)) = (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ EMPTY))))$

thm COLLINEAR_4_3:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ a \neq b \longrightarrow collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) = (collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \land collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ d\ EMPTY))))$

thm COLLINEAR_3_TRANS:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \land collinear\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))) \land b \neq c \longrightarrow collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ d\ EMPTY)))$

thm ORTHOGONAL_TO_ORTHOGONAL_2D:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ x \neq vec\ (0::nat) \land orthogonal\ x\ y \land orthogonal\ x\ z \longrightarrow collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ y\ (INSERT\ z\ EMPTY)))$

thm COLLINEAR_3_2D:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = ((\$\ z\ (1::nat) - \$\ x\ (1::nat)) * (\$\ y\ (2::nat) - \$\ x\ (2::nat)) = (\$\ y\ (1::nat) - \$\ x\ (1::nat)) * (\$\ z\ (2::nat) - \$\ x\ (2::nat)))$

thm DEF_between:

$between = (\lambda(\_148177::(real, ?'a::type)\ cart)\ \_148178::(real, ?'a::type)\ cart \times (real, ?'a::type)\ cart.\ distance\ (fst\ \_148178, snd\ \_148178) = distance\ (fst\ \_148178, \_148177) + distance\ (\_148177, snd\ \_148178))$

thm between:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ between\ x\ (a, b) = (distance\ (a, b) = distance\ (a, x) + distance\ (x, b))$

thm BETWEEN_REFL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ between\ a\ (a, b) \wedge between\ b\ (a, b) \wedge between\ a\ (a, a)$

thm BETWEEN_REFL_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ between\ x\ (a, a) = (x = a)$

thm BETWEEN_SYM:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ between\ x\ (a, b) = between\ x\ (b, a)$

thm BETWEEN_ANTISYM:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ between\ a\ (b, c) \wedge between\ b\ (a, c) \longrightarrow a = b$

thm BETWEEN_TRANS:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::(real, ?'a::type)\ cart)\ d::(real, ?'a::type)\ cart.\ between\ a\ (b, c) \wedge between\ d\ (a, c) \longrightarrow between\ d\ (b, c)$

thm BETWEEN_TRANS_2:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::(real, ?'a::type)\ cart)\ d::(real, ?'a::type)\ cart.\ between\ a\ (b, c) \wedge between\ d\ (a, b) \longrightarrow between\ a\ (c, d)$

thm BETWEEN_NORM:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ between\ x\ (a, b) = (\%\ (vector\_norm\ (vector\_sub\ x\ a))\ (vector\_sub\ b\ x) = \%\ (vector\_norm\ (vector\_sub\ b\ x))\ (vector\_sub\ x\ a))$

thm BETWEEN_DOT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $between\ x\ (a,\ b) = (dot\ (vector\_sub\ x\ a)\ (vector\_sub\ b\ x) = vector\_norm\ (vector\_sub\ x\ a) * vector\_norm\ (vector\_sub\ b\ x))$

thm BETWEEN_IMP_COLLINEAR:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $between\ x\ (a,\ b) \longrightarrow collinear\ (INSERT\ a\ (INSERT\ x\ (INSERT\ b\ EMPTY)))$

thm COLLINEAR_BETWEEN_CASES:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.$ $collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) = (between\ a\ (b,\ c) \lor between\ b\ (c,\ a) \lor between\ c\ (a,\ b))$

thm COLLINEAR_DIST_BETWEEN:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $collinear\ (INSERT\ x\ (INSERT\ a\ (INSERT\ b\ EMPTY))) \land distance\ (x,\ a) \leq distance\ (a,\ b) \land distance\ (x,\ b) \leq distance\ (a,\ b) \longrightarrow between\ x\ (a,\ b)$

thm COLLINEAR_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ collinear\ s$

thm DEF_midpoint:

$midpoint = (\lambda\_148579::(real,\ ?'a::type)\ cart \times (real,\ ?'a::type)\ cart.\ \%\ (inverse\ (real\_of\_nat\ (2::nat)))\ (vector\_add\ (fst\ \_148579)\ (snd\ \_148579)))$

thm midpoint:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ midpoint\ (a,\ b) = \%\ (inverse\ (real\_of\_nat\ (2::nat)))\ (vector\_add\ a\ b)$

thm MIDPOINT_REFL:

$\forall x::(real,\ ?'a::type)\ cart.\ midpoint\ (x,\ x) = x$

thm MIDPOINT_SYM:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ midpoint\ (a,\ b) = midpoint\ (b,\ a)$

thm DIST_MIDPOINT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ distance\ (a,\ midpoint\ (a,\ b)) = distance\ (a,\ b)\ /\ real\_of\_nat\ (2::nat) \land distance\ (b,\ midpoint\ (a,\ b)) = distance\ (a,\ b)\ /\ real\_of\_nat\ (2::nat) \land distance\ (midpoint\ (a,\ b),\ a) = distance\ (a,\ b)\ /\ real\_of\_nat\ (2::nat) \land distance\ (midpoint\ (a,\ b),\ b) = distance\ (a,\ b)\ /\ real\_of\_nat\ (2::nat)$

thm MIDPOINT_EQ_ENDPOINT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (midpoint\ (a,\ b) = a) = (a = b) \land (midpoint\ (a,\ b) = b) = (a = b) \land (a = midpoint\ (a,\ b)) = (a = b) \land (b = midpoint\ (a,\ b)) = (a = b)$

thm BETWEEN_MIDPOINT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ between\ (midpoint\ (a,\ b))\ (a,\ b) \wedge between\ (midpoint\ (a,\ b))\ (b,\ a)$

thm MIDPOINT_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow midpoint\ (f\ a,\ f\ b) = f\ (midpoint\ (a,\ b))$

thm COLLINEAR_MIDPOINT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ a\ (INSERT\ (midpoint\ (a,\ b))\ (INSERT\ b\ EMPTY)))$

thm MIDPOINT_COLLINEAR:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ a \neq c \longrightarrow (b = midpoint\ (a,\ c)) = (collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \wedge distance\ (a,\ b) = distance\ (b,\ c))$

thm WLOG_LINEAR_INJECTIVE_IMAGE_2:

$\forall (P::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ Q::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ P\ s \wedge linear\ f \longrightarrow Q\ (IMAGE\ f\ s)) \wedge (\forall (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ Q\ t \wedge linear\ g \longrightarrow P\ (IMAGE\ g\ t)) \longrightarrow (\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ Q\ (IMAGE\ f\ s) = P\ s))$

thm WLOG_LINEAR_INJECTIVE_IMAGE_2_ALT:

$\forall (P::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (Q::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ u::(real,\ ?'b::type)\ cart \Rightarrow bool.\ P\ u \wedge linear\ h \longrightarrow Q\ (IMAGE\ h\ u)) \wedge (\forall (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ Q\ t \wedge linear\ g \longrightarrow P\ (IMAGE\ g\ t)) \wedge linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow Q\ (IMAGE\ f\ s) = P\ s$

thm WLOG_LINEAR_INJECTIVE_IMAGE:

$\forall P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ P\ s \wedge linear\ f \longrightarrow P\ (IMAGE\ f\ s)) \longrightarrow (\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ P\ (IMAGE\ f\ s) = P\ s))$

thm WLOG_LINEAR_INJECTIVE_IMAGE_ALT:

$\forall (P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall (g::(real,\ ?'a::type)\ cart \Rightarrow$

$(real,\ ?'a::type)\ cart)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ P\ t \land linear\ g \longrightarrow P\ (IMAGE\ g\ t)) \land linear\ f \land (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow P\ (IMAGE\ f\ s) = P\ s$

thm SUBSPACE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \land (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow subspace\ (IMAGE\ f\ s) = subspace\ s$

thm LINEAR_SCALING:

$\forall c::real.\ linear\ (\%\ c)$

thm INJECTIVE_SCALING:

$\forall c::real.\ (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ \%\ c\ x = \%\ c\ y \longrightarrow x = y) = (c \neq (0::real))$

thm SURJECTIVE_SCALING:

$\forall c::real.\ (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ \%\ c\ x = y) = (c \neq (0::real))$

thm PRESERVES_NORM_PRESERVES_DOT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ linear\ f \land (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow dot\ (f\ x)\ (f\ y) = dot\ x\ y$

thm PRESERVES_NORM_INJECTIVE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \land (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y)$

thm ORTHOGONAL_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ linear\ f \land (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow orthogonal\ (f\ x)\ (f\ y) = orthogonal\ x\ y$

thm MEM_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart\ list.\ MEM\ (vector\_add\ a\ x)\ (map\ (vector\_add\ a)\ l) = MEM\ x\ l$

thm MEM_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ l::(real,\ ?'b::type)\ cart\ list.\ linear\ f \land (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow MEM\ (f\ x)\ (map\ f\ l) = MEM\ x\ l$

thm QUANTIFY_SURJECTION_THM:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\,x = y) \longrightarrow ((\forall P{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ P\ x) = (\forall x{::}?'b{::}type.\ P\ (f\ x))) \wedge (\forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'a{::}type.\ P\,x) = (\exists x{::}?'b{::}type.\ P\ (f\,x))) \wedge (\forall Q{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall s{::}?'a{::}type \Rightarrow bool.\ Q\ s) = (\forall s{::}?'b{::}type \Rightarrow bool.\ Q\ (IMAGE\,f\,s))) \wedge (\forall Q{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\exists s{::}?'a{::}type \Rightarrow bool.\ Q\ s) = (\exists s{::}?'b{::}type \Rightarrow bool.\ Q\ (IMAGE\,f\,s)))) \wedge (\forall P{::}?'a{::}type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%373{::}?'a{::}type.\ \exists x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%373\ (P\ x)\ x) = IMAGE\ f\ (GSPEC\ (\lambda GEN\%PVAR\%374{::}?'b{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%374\ (P\ (f\ x))\ x)))$

thm QUANTIFY_SURJECTION_HIGHER_THM:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall y{::}?'a{::}type.\ \exists x{::}?'b{::}type.\ f\,x = y) \longrightarrow ((\forall P{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ P\ x) = (\forall x{::}?'b{::}type.\ P\ (f\ x))) \wedge (\forall P{::}?'a{::}type \Rightarrow bool.\ (\exists x{::}?'a{::}type.\ P\,x) = (\exists x{::}?'b{::}type.\ P\ (f\,x))) \wedge (\forall Q{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\forall s{::}?'a{::}type \Rightarrow bool.\ Q\ s) = (\forall s{::}?'b{::}type \Rightarrow bool.\ Q\ (IMAGE\,f\,s))) \wedge (\forall Q{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ (\exists s{::}?'a{::}type \Rightarrow bool.\ Q\ s) = (\exists s{::}?'b{::}type \Rightarrow bool.\ Q\ (IMAGE\,f\,s))) \wedge (\forall Q{::}((?'a{::}type \Rightarrow bool) \Rightarrow bool) \Rightarrow bool.\ (\forall s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ Q\ s) = (\forall s{::}(?'b{::}type \Rightarrow bool) \Rightarrow bool.\ Q\ (IMAGE\ (IMAGE\ f)\ s))) \wedge (\forall Q{::}((?'a{::}type \Rightarrow bool) \Rightarrow bool) \Rightarrow bool.\ (\exists s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ Q\ s) = (\exists s{::}(?'b{::}type \Rightarrow bool) \Rightarrow bool.\ Q\ (IMAGE\ (IMAGE\,f)\ s))) \wedge (\forall P{::}((real,\ unit)\ cart \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\forall g{::}(real,\ unit)\ cart \Rightarrow ?'a{::}type.\ P\ g) = (\forall g{::}(real,\ unit)\ cart \Rightarrow ?'b{::}type.\ P\ (f \circ g))) \wedge (\forall P{::}((real,\ unit)\ cart \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\exists g{::}(real,\ unit)\ cart \Rightarrow ?'a{::}type.\ P\,g) = (\exists g{::}(real,\ unit)\ cart \Rightarrow ?'b{::}type.\ P\ (f \circ g))) \wedge (\forall P{::}(nat \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\forall g{::}nat \Rightarrow ?'a{::}type.\ P\ g) = (\forall g{::}nat \Rightarrow ?'b{::}type.\ P\ (f \circ g))) \wedge (\forall P{::}(nat \Rightarrow ?'a{::}type) \Rightarrow bool.\ (\exists g{::}nat \Rightarrow ?'a{::}type.\ P\ g) = (\exists g{::}nat \Rightarrow ?'b{::}type.\ P\ (f \circ g))) \wedge (\forall Q{::}?'a{::}type\ list \Rightarrow bool.\ (\forall l{::}?'a{::}type\ list.\ Q\ l) = (\forall l{::}?'b{::}type\ list.\ Q\ (map\ f\,l))) \wedge (\forall Q{::}?'a{::}type\ list \Rightarrow bool.\ (\exists l{::}?'a{::}type\ list.\ Q\ l) = (\exists l{::}?'b{::}type\ list.\ Q\ (map\,f\,l)))) \wedge (\forall P{::}?'a{::}type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%375{::}?'a{::}type.\ \exists x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%375\ (P\ x)\ x) = IMAGE\ f\ (GSPEC\ (\lambda GEN\%PVAR\%376{::}?'b{::}type.\ \exists x{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%376\ (P\ (f\,x))\ x))) \wedge (\forall Q{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%377{::}?'a{::}type \Rightarrow bool.\ \exists s{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%377\ (Q\ s)\ s) = IMAGE\ (IMAGE\,f)\ (GSPEC\ (\lambda GEN\%PVAR\%378{::}?'b{::}type \Rightarrow bool.\ \exists s{::}?'b{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%378\ (Q\ (IMAGE\,f\,s))\ s))) \wedge (\forall R{::}?'a{::}type\ list \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%379{::}?'a{::}type\ list.\ \exists l{::}?'a{::}type\ list.\ SETSPEC\ GEN\%PVAR\%379\ (R\ l)\ l) = IMAGE\ (map\,f)\ (GSPEC\ (\lambda GEN\%PVAR\%380{::}?'b{::}type\ list.\ \exists l{::}?'b{::}type\ list.\ SETSPEC\ GEN\%PVAR\%380\ (R\ (map\ f\ l))\ l)))$

thm DEF_permutes:

$permutes = (\lambda(\_152830{::}?'a{::}type \Rightarrow ?'a{::}type)\ \_152831{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ \neg\ IN\,x\ \_152831 \longrightarrow \_152830\,x = x) \wedge (\forall y{::}?'a{::}type.\ \exists!x{::}?'a{::}type.\ \_152830\ x = y))$

thm permutes:

$\forall (s{::}?'a{::}type \Rightarrow bool)\ p{::}?'a{::}type \Rightarrow ?'a{::}type.\ permutes\ p\ s = ((\forall x{::}?'a{::}type.\ \neg\ IN\ x\ s \longrightarrow p\ x = x) \wedge (\forall y{::}?'a{::}type.\ \exists!x{::}?'a{::}type.\ p\ x = y))$

thm DEF_inverse:

$HOL\_Light\_Import.inverse = (\lambda(\_152842::?'b::type \Rightarrow ?'a::type)\ y::?'a::type.\ SOME\ x::?'b::type.\ \_152842\ x = y)$

thm inverse:

$\forall f::?'b::type \Rightarrow ?'a::type.\ HOL\_Light\_Import.inverse\ f = (\lambda y::?'a::type.\ SOME\ x::?'b::type.\ f\ x = y)$

thm SURJECTIVE_INVERSE:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall y::?'a::type.\ \exists x::?'b::type.\ f\ x = y) = (\forall y::?'a::type.\ f\ (HOL\_Light\_Import.inverse\ f\ y) = y)$

thm SURJECTIVE_INVERSE_o:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall y::?'a::type.\ \exists x::?'b::type.\ f\ x = y) = (f \circ HOL\_Light\_Import.inverse\ f = id)$

thm INJECTIVE_INVERSE:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall (x::?'b::type)\ x'::?'b::type.\ f\ x = f\ x' \longrightarrow x = x') = (\forall x::?'b::type.\ HOL\_Light\_Import.inverse\ f\ (f\ x) = x)$

thm INJECTIVE_INVERSE_o:

$\forall f::?'b::type \Rightarrow ?'a::type.\ (\forall (x::?'b::type)\ x'::?'b::type.\ f\ x = f\ x' \longrightarrow x = x') = (HOL\_Light\_Import.inverse\ f \circ f = id)$

thm INVERSE_UNIQUE_o:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ g::?'a::type \Rightarrow ?'b::type.\ f \circ g = id \wedge g \circ f = id \longrightarrow HOL\_Light\_Import.inverse\ f = g$

thm INVERSE_I:

$HOL\_Light\_Import.inverse\ id = id$

thm DEF_swap:

$swap = (\lambda(\_153256::?'a::type \times ?'a::type)\ \_153257::?'a::type.\ if\ \_153257 = fst\ \_153256\ then\ snd\ \_153256\ else\ if\ \_153257 = snd\ \_153256\ then\ fst\ \_153256\ else\ \_153257)$

thm swap:

$\forall (j::?'a::type)\ (i::?'a::type)\ k::?'a::type.\ swap\ (i,\ j)\ k = (if\ k = i\ then\ j\ else\ if\ k = j\ then\ i\ else\ k)$

thm SWAP_REFL:

$\forall a::?'a::type.\ swap\ (a,\ a) = id$

thm SWAP_SYM:

$\forall (a::?'a::type)\ b::?'a::type.\ swap\ (a,\ b) = swap\ (b,\ a)$

thm SWAP_IDEMPOTENT:

$\forall (a::?'a::type)\ b::?'a::type.\ swap\ (a,\ b) \circ swap\ (a,\ b) = id$

thm INVERSE_SWAP:

$\forall (a::?'a::type)\ b::?'a::type.\ HOL\_Light\_Import.inverse\ (swap\ (a,\ b)) = swap\ (a,\ b)$

thm SWAP_GALOIS:

$\forall (a::?'a::type)\ (b::?'a::type)\ (x::?'a::type)\ y::?'a::type.\ (x = swap\ (a,\ b)\ y) = (y = swap\ (a,\ b)\ x)$

thm PERMUTES_IN_IMAGE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ permutes\ p\ s \longrightarrow IN\ (p\ x)\ s = IN\ x\ s$

thm PERMUTES_IMAGE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow IMAGE\ p\ s = s$

thm PERMUTES_INJECTIVE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ (p\ x = p\ y) = (x = y))$

thm PERMUTES_SURJECTIVE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow (\forall y::?'a::type.\ \exists x::?'a::type.\ p\ x = y)$

thm PERMUTES_INVERSES_o:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow p \circ HOL\_Light\_Import.inverse\ p = id \wedge HOL\_Light\_Import.inverse\ p \circ p = id$

thm PERMUTES_INVERSES:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow (\forall x::?'a::type.\ p\ (HOL\_Light\_Import.inverse\ p\ x) = x) \wedge (\forall x::?'a::type.\ HOL\_Light\_Import.inverse\ p\ (p\ x) = x)$

thm PERMUTES_SUBSET:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ permutes\ p\ s \wedge SUBSET\ s\ t \longrightarrow permutes\ p\ t$

thm PERMUTES_EMPTY:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ EMPTY = (p = id)$

thm PERMUTES_SING:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ a::?'a::type.\ permutes\ p\ (INSERT\ a\ EMPTY) = (p = id)$

thm PERMUTES_UNIV:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ HOL\_Light\_Import.UNIV = (\forall y::?'a::type.\ \exists!x::?'a::type.\ p\ x = y)$

thm PERMUTES_INVERSE_EQ:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ (HOL\_Light\_Import.inverse\ p\ y = x) = (p\ x = y))$

thm PERMUTES_SWAP:

$\forall (a::?'a::type)\ (b::?'a::type)\ s::?'a::type \Rightarrow bool.\ IN\ a\ s \wedge IN\ b\ s \longrightarrow permutes\ (swap\ (a,\ b))\ s$

thm PERMUTES_SUPERSET:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ permutes\ p\ s \wedge (\forall x::?'a::type.\ IN\ x\ (DIFF\ s\ t) \longrightarrow p\ x = x) \longrightarrow permutes\ p\ t$

thm PERMUTES_I:

$\forall s::?'a::type \Rightarrow bool.\ permutes\ id\ s$

thm PERMUTES_COMPOSE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (q::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \wedge permutes\ q\ s \longrightarrow permutes\ (q \circ p)\ s$

thm PERMUTES_INVERSE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow permutes\ (HOL\_Light\_Import.inverse\ p)\ s$

thm PERMUTES_INVERSE_INVERSE:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ (?s::?'a::type \Rightarrow bool) \longrightarrow HOL\_Light\_Import.inverse\ (HOL\_Light\_Import.inverse\ p) = p$

thm PERMUTES_INSERT_LEMMA:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ (INSERT\ a\ s) \longrightarrow permutes\ (swap\ (a,\ p\ a) \circ p)\ s$

thm PERMUTES_INSERT:

$GSPEC\ (\lambda GEN\%PVAR\%381::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%381\ (permutes\ p\ (INSERT\ (?a::?'a::type)\ (?s::?'a::type \Rightarrow bool)))\ p) = IMAGE\ (GABS\ (\lambda f::?'a::type \times (?'a::type \Rightarrow ?'a::type) \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall (b::?'a::type)\ p::?'a::type \Rightarrow ?'a::type.\ GEQ\ (f\ (b,\ p))\ (swap\ (?a,\ b) \circ p)))\ (GSPEC\ (\lambda GEN\%PVAR\%383::?'a::type \times (?'a::type \Rightarrow ?'a::type).\ \exists (b::?'a::type)\ p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%383\ (IN\ b\ (INSERT\ ?a\ ?s) \wedge IN\ p\ (GSPEC\ (\lambda GEN\%PVAR\%382::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%382\ (permutes\ p\ ?s)\ p)))\ (b,\ p)))$

thm HAS_SIZE_PERMUTATIONS:

$\forall (s::?'a::type \Rightarrow bool)\ n::nat.\ HAS\_SIZE\ s\ n \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%385::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%385\ (permutes\ p\ s)\ p))\ (fact\ n)$

thm FINITE_PERMUTATIONS:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%386::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%386\ (permutes\ p\ s)\ p))$

thm CARD_PERMUTATIONS:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%387::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%387\ (permutes\ p\ s)\ p)) = fact\ (CARD\ s)$

thm PERMUTES_FINITE_INJECTIVE:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \longrightarrow permutes\ p\ s = ((\forall x::?'a::type.\ \neg\ IN\ x\ s \longrightarrow p\ x = x) \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge p\ x = p\ y \longrightarrow x = y))$

thm PERMUTES_FINITE_SURJECTIVE:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \longrightarrow permutes\ p\ s = ((\forall x::?'a::type.\ \neg\ IN\ x\ s \longrightarrow p\ x = x) \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (p\ x)\ s) \wedge (\forall y::?'a::type.\ IN\ y\ s \longrightarrow (\exists x::?'a::type.\ IN\ x\ s \wedge p\ x = y)))$

thm ITERATE_PERMUTE:

$\forall op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall (f::?'a::type \Rightarrow ?'b::type)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow iterate\ op\ s\ f = iterate\ op\ s\ (f \circ p))$

thm NSUM_PERMUTE:

$\forall (f::?'a::type \Rightarrow nat)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow nsum\ s\ f = nsum\ s\ (f \circ p)$

thm NSUM_PERMUTE_NUMSEG:

$\forall (f::nat \Rightarrow nat)\ (p::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ permutes\ p\ (dotdot\ m\ n) \longrightarrow nsum\ (dotdot\ m\ n)\ f = nsum\ (dotdot\ m\ n)\ (f \circ p)$

thm SUM_PERMUTE:

$\forall (f::?'a::type \Rightarrow real)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow sum\ s\ f = sum\ s\ (f \circ p)$

thm SUM_PERMUTE_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (p::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ permutes\ p\ (dotdot\ m\ n) \longrightarrow sum\ (dotdot\ m\ n)\ f = sum\ (dotdot\ m\ n)\ (f \circ p)$

thm SWAP_COMMON:

$\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ a \neq c \wedge b \neq c \longrightarrow swap\ (a,\ b) \circ swap\ (a,\ c) = swap\ (b,\ c) \circ swap\ (a,\ b)$

thm SWAP_COMMON′:

$\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ a \neq b \wedge a \neq c \longrightarrow swap\ (a,\ c) \circ swap\ (b,\ c) = swap\ (b,\ c) \circ swap\ (a,\ b)$

thm SWAP_INDEPENDENT:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ a \neq c \wedge a \neq d \wedge b \neq c \wedge b \neq d \longrightarrow swap\ (a,\ b) \circ swap\ (c,\ d) = swap\ (c,\ d) \circ swap\ (a,\ b)$

thm DEF_swapseq:

$swapseq = (\lambda(a0::nat)\ a1::?'a::type \Rightarrow ?'a::type.\ \forall swapseq'::nat \Rightarrow (?'a::type \Rightarrow ?'a::type) \Rightarrow bool.\ (\forall (a0::nat)\ a1::?'a::type \Rightarrow ?'a::type.\ a0 = (0::nat) \wedge a1 = id \vee (\exists (a::?'a::type)\ (b::?'a::type)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ a0 = Suc\ n \wedge a1 = swap\ (a,\ b) \circ p \wedge swapseq'\ n\ p \wedge a \neq b) \longrightarrow swapseq'\ a0\ a1) \longrightarrow swapseq'\ a0\ a1)$

thm swapseq_RULES:

$swapseq\ (0::nat)\ id \wedge (\forall (a::?'a::type)\ (b::?'a::type)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ swapseq\ n\ p \wedge a \neq b \longrightarrow swapseq\ (Suc\ n)\ (swap\ (a,\ b) \circ p))$

thm swapseq_CASES:

$\forall (a0::nat)\ a1::?'a::type \Rightarrow ?'a::type.\ swapseq\ a0\ a1 = (a0 = (0::nat) \wedge a1 = id \vee (\exists (a::?'a::type)\ (b::?'a::type)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ a0 = Suc\ n \wedge a1 = swap\ (a,\ b) \circ p \wedge swapseq\ n\ p \wedge a \neq b))$

thm swapseq_INDUCT:

$\forall swapseq'::nat \Rightarrow (?'a::type \Rightarrow ?'a::type) \Rightarrow bool.\ swapseq'\ (0::nat)\ id \wedge (\forall (a::?'a::type)\ (b::?'a::type)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ swapseq'\ n\ p \wedge a \neq b \longrightarrow swapseq'\ (Suc\ n)\ (swap\ (a,\ b) \circ p)) \longrightarrow (\forall (a0::nat)\ a1::?'a::type \Rightarrow ?'a::type.\ swapseq\ a0\ a1 \longrightarrow swapseq'\ a0\ a1)$

thm DEF_permutation:

$permutation = (\lambda\_157261::?'a::type \Rightarrow ?'a::type.\ \exists n::nat.\ swapseq\ n\ \_157261)$

thm permutation:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p = (\exists n::nat.\ swapseq\ n\ p)$

thm SWAPSEQ_I:

$swapseq\ (0::nat)\ id$

thm PERMUTATION_I:

$permutation\ id$

thm SWAPSEQ_SWAP:

$\forall (a::?'a::type)\ b::?'a::type.\ swapseq\ (if\ a = b\ then\ 0::nat\ else\ (1::nat))\ (swap\ (a,\ b))$

thm PERMUTATION_SWAP:

$\forall (a::?'a::type)\ b::?'a::type.\ permutation\ (swap\ (a,\ b))$

thm SWAPSEQ_COMPOSE:

$\forall (n::nat)\ (p::?'a::type \Rightarrow ?'a::type)\ (m::nat)\ q::?'a::type \Rightarrow ?'a::type.\ swapseq\ n\ p \wedge swapseq\ m\ q \longrightarrow swapseq\ (n + m)\ (p \circ q)$

thm PERMUTATION_COMPOSE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ p \wedge permutation\ q \longrightarrow permutation\ (p \circ q)$

thm SWAPSEQ_ENDSWAP:

$\forall (n::nat)\ (p::?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ b::?'a::type.\ swapseq\ n\ p \wedge a \neq b \longrightarrow swapseq\ (Suc\ n)\ (p \circ swap\ (a,\ b))$

thm SWAPSEQ_INVERSE_EXISTS:

$\forall (n::nat)\ p::?'a::type \Rightarrow ?'a::type.\ swapseq\ n\ p \longrightarrow (\exists\, q::?'a::type \Rightarrow ?'a::type.\ swapseq\ n\ q \wedge p \circ q = id \wedge q \circ p = id)$

thm SWAPSEQ_INVERSE:

$\forall (n::nat)\ p::?'a::type \Rightarrow ?'a::type.\ swapseq\ n\ p \longrightarrow swapseq\ n\ (HOL\_Light\_Import.inverse\ p)$

thm PERMUTATION_INVERSE:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow permutation\ (HOL\_Light\_Import.inverse\ p)$

thm SYMMETRY_LEMMA:

$(\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ (?P::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ a\ b\ c\ d \longrightarrow ?P\ a\ b\ d\ c) \wedge (\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ a \neq b \wedge c \neq d \wedge (a = c \wedge b = d \vee a = c \wedge b \neq d \vee a \neq c \wedge b = d \vee a \neq c \wedge a \neq d \wedge b \neq c \wedge b \neq d) \longrightarrow ?P\ a\ b\ c\ d) \longrightarrow (\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ a \neq b \wedge c \neq d \longrightarrow ?P\ a\ b\ c\ d)$

thm SWAP_GENERAL:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ a \neq b \wedge c \neq d \longrightarrow swap\ (a,\ b) \circ swap\ (c,\ d) = id \vee (\exists (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ x \neq a \wedge y \neq a \wedge z \neq a \wedge x \neq y \wedge swap\ (a,\ b) \circ swap\ (c,\ d) = swap\ (x,\ y) \circ swap\ (a,\ z))$

thm FIXING_SWAPSEQ_DECREASE:

$\forall (n::nat)\ (p::?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ b::?'a::type.\ swapseq\ n\ p \wedge a \neq b \wedge (swap\ (a,\ b) \circ p)\ a = a \longrightarrow n \neq (0::nat) \wedge swapseq\ (n - (1::nat))\ (swap\ (a,\ b) \circ p)$

thm SWAPSEQ_IDENTITY_EVEN:

$\forall n::nat.\ swapseq\ n\ id \longrightarrow even\ n$

thm DEF_evenperm:

$evenperm = (\lambda\_160198::?'a::type \Rightarrow ?'a::type.\ even\ (SOME\ n::nat.\ swapseq\ n\ \_160198))$

thm evenperm:

$\forall p::?'a::type \Rightarrow ?'a::type.\ evenperm\ p = even\ (SOME\ n::nat.\ swapseq\ n\ p)$

thm SWAPSEQ_EVEN_EVEN:

$\forall (m::nat)\ (n::nat)\ p::?'a::type \Rightarrow ?'a::type.\ swapseq\ m\ p \wedge swapseq\ n\ p \longrightarrow even\ m = even\ n$

thm EVENPERM_UNIQUE:

$\forall (n::nat)\ (p::?'a::type \Rightarrow ?'a::type)\ b::bool.\ swapseq\ n\ p \wedge even\ n = b \longrightarrow evenperm\ p = b$

thm EVENPERM_I:

$evenperm\ id = True$

thm EVENPERM_SWAP:

$\forall (a::?'a::type)\ b::?'a::type.\ evenperm\ (swap\ (a,\ b)) = (a = b)$

thm EVENPERM_COMPOSE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ p \wedge permutation\ q \longrightarrow evenperm\ (p \circ q) = (evenperm\ p = evenperm\ q)$

thm EVENPERM_INVERSE:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow evenperm\ (HOL\_Light\_Import.inverse\ p) = evenperm\ p$

thm PERMUTATION_BIJECTIVE:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow (\forall y::?'a::type.\ \exists!x::?'a::type.\ p\ x = y)$

thm PERMUTATION_FINITE_SUPPORT:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%390::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%390\ (p\ x \neq x)\ x))$

thm PERMUTATION_LEMMA:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge (\forall y::?'a::type.\ \exists !x::?'a::type.\ p\ x = y) \wedge (\forall x::?'a::type.\ \neg\ IN\ x\ s \longrightarrow p\ x = x) \longrightarrow permutation\ p$

thm PERMUTATION:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p = ((\forall y::?'a::type.\ \exists !x::?'a::type.\ p\ x = y) \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%392::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%392\ (p\ x \neq x)\ x)))$

thm PERMUTATION_INVERSE_WORKS:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow HOL\_Light\_Import.inverse\ p \circ p = id \wedge p \circ HOL\_Light\_Import.inverse\ p = id$

thm PERMUTATION_INVERSE_COMPOSE:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ p \wedge permutation\ q \longrightarrow HOL\_Light\_Import.inverse\ (p \circ q) = HOL\_Light\_Import.inverse\ q \circ HOL\_Light\_Import.inverse\ p$

thm PERMUTATION_COMPOSE_EQ_conjunct1:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ q \longrightarrow permutation\ (p \circ q) = permutation\ p$

thm PERMUTATION_COMPOSE_EQ_conjunct0:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow permutation\ (p \circ q) = permutation\ q$

thm PERMUTATION_COMPOSE_EQ:

$(\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ p \longrightarrow permutation\ (p \circ q) = permutation\ q) \wedge (\forall (p::?'a::type \Rightarrow ?'a::type)\ q::?'a::type \Rightarrow ?'a::type.\ permutation\ q \longrightarrow permutation\ (p \circ q) = permutation\ p)$

thm PERMUTATION_COMPOSE_SWAP:

$(\forall (p::?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ b::?'a::type.\ permutation\ (swap\ (a, b) \circ p) = permutation\ p) \wedge (\forall (p::?'a::type \Rightarrow ?'a::type)\ (a::?'a::type)\ b::?'a::type.\ permutation\ (p \circ swap\ (a, b)) = permutation\ p)$

thm PERMUTATION_PERMUTES:

$\forall p::?'a::type \Rightarrow ?'a::type.\ permutation\ p = (\exists s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge permutes\ p\ s)$

thm PERMUTES_INDUCT:

$\forall (P::(?'a::type \Rightarrow ?'a::type) \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge P\ id \wedge (\forall (a::?'a::type)\ (b::?'a::type)\ p::?'a::type \Rightarrow ?'a::type.\ IN\ a\ s \wedge IN\ b\ s \wedge P\ p \wedge permutation\ p \longrightarrow P\ (swap\ (a, b) \circ p)) \longrightarrow (\forall p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ s \longrightarrow P\ p)$

thm DEF_sign:

sign = (λ*_162333*::*?'a*::*type* ⇒ *?'a*::*type*. *if evenperm _162333 then 1*::*real else* − (*1*::*real*))

thm sign:

∀ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *sign p* = (*if evenperm p then 1*::*real else* − (*1*::*real*))

thm SIGN_NZ:

∀ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *sign p* ≠ (*0*::*real*)

thm SIGN_I:

sign id = (*1*::*real*)

thm SIGN_INVERSE:

∀ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *permutation p* ⟶ *sign* (*HOL_Light_Import.inverse p*) = *sign p*

thm SIGN_COMPOSE:

∀ (*p*::*?'a*::*type* ⇒ *?'a*::*type*) *q*::*?'a*::*type* ⇒ *?'a*::*type*. *permutation p* ∧ *permutation q* ⟶ *sign* (*p* ∘ *q*) = *sign p* ∗ *sign q*

thm SIGN_SWAP:

∀ (*a*::*?'a*::*type*) *b*::*?'a*::*type*. *sign* (*swap* (*a*, *b*)) = (*if a* = *b then 1*::*real else* − (*1*::*real*))

thm SIGN_IDEMPOTENT:

∀ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *sign p* ∗ *sign p* = (*1*::*real*)

thm REAL_ABS_SIGN:

∀ *p*::*?'a*::*type* ⇒ *?'a*::*type*. |*sign p*| = (*1*::*real*)

thm PERMUTES_NUMSET_LE:

∀ (*p*::*nat* ⇒ *nat*) *s*::*nat* ⇒ *bool*. *permutes p s* ∧ (∀ *i*::*nat*. *IN i s* ⟶ *p i* ≤ *i*) ⟶ *p* = *id*

thm PERMUTES_NUMSET_GE:

∀ (*p*::*nat* ⇒ *nat*) *s*::*nat* ⇒ *bool*. *permutes p s* ∧ (∀ *i*::*nat*. *IN i s* ⟶ *i* ≤ *p i*) ⟶ *p* = *id*

thm IMAGE_INVERSE_PERMUTATIONS:

∀ *s*::*?'a*::*type* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%394*::*?'a*::*type* ⇒ *?'a*::*type*. ∃ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *SETSPEC GEN%PVAR%394* (*permutes p s*) (*HOL_Light_Import.inverse p*)) = *GSPEC* (λ*GEN%PVAR%395*::*?'a*::*type* ⇒ *?'a*::*type*. ∃ *p*::*?'a*::*type* ⇒ *?'a*::*type*. *SETSPEC GEN%PVAR%395* (*permutes p s*) *p*)

thm IMAGE_COMPOSE_PERMUTATIONS_L:

$\forall (s::?'a::type \Rightarrow bool)\ q::?'a::type \Rightarrow ?'a::type.\ permutes\ q\ s \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%396::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%396\ (permutes\ p\ s)\ (q \circ p)) = GSPEC\ (\lambda GEN\%PVAR\%397::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%397\ (permutes\ p\ s)\ p)$

thm IMAGE_COMPOSE_PERMUTATIONS_R:

$\forall (s::?'a::type \Rightarrow bool)\ q::?'a::type \Rightarrow ?'a::type.\ permutes\ q\ s \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%398::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%398\ (permutes\ p\ s)\ (p \circ q)) = GSPEC\ (\lambda GEN\%PVAR\%399::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%399\ (permutes\ p\ s)\ p)$

thm PERMUTES_IN_NUMSEG:

$\forall (p::nat \Rightarrow nat)\ (n::nat)\ i::nat.\ permutes\ p\ (dotdot\ (1::nat)\ n) \wedge IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow (1::nat) \leq p\ i \wedge p\ i \leq n$

thm SUM_PERMUTATIONS_INVERSE:

$\forall (f::(nat \Rightarrow nat) \Rightarrow real)\ (m::nat)\ n::nat.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%402::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%402\ (permutes\ p\ (dotdot\ m\ n))\ p))\ f = sum\ (GSPEC\ (\lambda GEN\%PVAR\%403::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%403\ (permutes\ p\ (dotdot\ m\ n))\ p))\ (\lambda p::nat \Rightarrow nat.\ f\ (HOL\_Light\_Import.inverse\ p))$

thm SUM_PERMUTATIONS_COMPOSE_L:

$\forall (f::(nat \Rightarrow nat) \Rightarrow real)\ (m::nat)\ (n::nat)\ q::nat \Rightarrow nat.\ permutes\ q\ (dotdot\ m\ n) \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%406::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%406\ (permutes\ p\ (dotdot\ m\ n))\ p))\ f = sum\ (GSPEC\ (\lambda GEN\%PVAR\%407::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%407\ (permutes\ p\ (dotdot\ m\ n))\ p))\ (\lambda p::nat \Rightarrow nat.\ f\ (q \circ p))$

thm SUM_PERMUTATIONS_COMPOSE_R:

$\forall (f::(nat \Rightarrow nat) \Rightarrow real)\ (m::nat)\ (n::nat)\ q::nat \Rightarrow nat.\ permutes\ q\ (dotdot\ m\ n) \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%410::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%410\ (permutes\ p\ (dotdot\ m\ n))\ p))\ f = sum\ (GSPEC\ (\lambda GEN\%PVAR\%411::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%411\ (permutes\ p\ (dotdot\ m\ n))\ p))\ (\lambda p::nat \Rightarrow nat.\ f\ (p \circ q))$

thm SUM_OVER_PERMUTATIONS_INSERT:

$\forall (f::(?'a::type \Rightarrow ?'a::type) \Rightarrow real)\ (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge \neg\ IN\ a\ s \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%416::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%416\ (permutes\ p\ (INSERT\ a\ s))\ p))\ f = sum\ (INSERT\ a\ s)\ (\lambda b::?'a::type.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%417::?'a::type \Rightarrow ?'a::type.\ \exists p::?'a::type \Rightarrow ?'a::type.\ SETSPEC\ GEN\%PVAR\%417\ (permutes\ p\ s)\ p))\ (\lambda q::?'a::type \Rightarrow ?'a::type.\ f\ (swap\ (a,\ b) \circ q)))$

thm SUM_OVER_PERMUTATIONS_NUMSEG:

$\forall (f::(nat \Rightarrow nat) \Rightarrow real)\ (m::nat)\ n::nat.\ m \leq n \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%418::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%418\ (permutes\ p\ (dotdot\ m\ n))\ p))\ f = sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%419::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%419\ (permutes\ p\ (dotdot\ (m + (1::nat))\ n))\ p))\ (\lambda q::nat \Rightarrow nat.\ f\ (swap\ (m,\ i) \circ q)))$

thm INTEGER_CASES:

$integer\ (?x::real) = ((\exists n::nat.\ ?x = real\_of\_nat\ n) \vee (\exists n::nat.\ ?x = -\ real\_of\_nat\ n))$

thm REAL_ABS_INTEGER_LEMMA:

$\forall x::real.\ integer\ x \wedge x \neq (0::real) \longrightarrow (1::real) \leq |x|$

thm INTEGER_CLOSED:

$(\forall n::nat.\ integer\ (real\_of\_nat\ n)) \wedge (\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x + y)) \wedge (\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x - y)) \wedge (\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x * y)) \wedge (\forall (x::real)\ r::nat.\ integer\ x \longrightarrow integer\ x^r) \wedge (\forall x::real.\ integer\ x \longrightarrow integer\ (-\ x)) \wedge (\forall x::real.\ integer\ x \longrightarrow integer\ |x|)$

thm INTEGER_CLOSED_conjunct6:

$\forall x::real.\ integer\ x \longrightarrow integer\ |x|$

thm INTEGER_CLOSED_conjunct5:

$\forall x::real.\ integer\ x \longrightarrow integer\ (-\ x)$

thm INTEGER_CLOSED_conjunct4:

$\forall (x::real)\ r::nat.\ integer\ x \longrightarrow integer\ x^r$

thm INTEGER_MUL:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x * y)$

thm INTEGER_SUB:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x - y)$

thm INTEGER_ADD:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow integer\ (x + y)$

thm INTEGER_CLOSED_conjunct0:

$\forall n::nat.\ integer\ (real\_of\_nat\ n)$

thm INTEGER_POW:

$\forall (x::real)\ n::nat.\ integer\ x \longrightarrow integer\ x^n$

thm REAL_LE_INTEGERS:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow (x \le y) = (x = y \vee x + (1::real) \le y)$

thm REAL_LE_CASES_INTEGERS:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow x \le y \vee y + (1::real) \le x$

thm REAL_LE_REVERSE_INTEGERS:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \wedge \neg\ y + (1::real) \le x \longrightarrow x \le y$

thm REAL_LT_INTEGERS:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow (x < y) = (x + (1::real) \le y)$

thm REAL_EQ_INTEGERS:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow (x = y) = (|x - y| < (1::real))$

thm REAL_EQ_INTEGERS_IMP:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \wedge |x - y| < (1::real) \longrightarrow x = y$

thm INTEGER_NEG:

$\forall x::real.\ integer\ (-\ x) = integer\ x$

thm INTEGER_ABS:

$\forall x::real.\ integer\ |x| = integer\ x$

thm INTEGER_POS:

$\forall x \ge 0::real.\ integer\ x = (\exists n::nat.\ x = real\_of\_nat\ n)$

thm INTEGER_ADD_EQ:

$(\forall (x::real)\ y::real.\ integer\ x \longrightarrow integer\ (x + y) = integer\ y) \wedge (\forall (x::real)\ y::real.\ integer\ y \longrightarrow integer\ (x + y) = integer\ x)$

thm INTEGER_SUB_EQ:

$(\forall (x::real)\ y::real.\ integer\ x \longrightarrow integer\ (x - y) = integer\ y) \wedge (\forall (x::real)\ y::real.\ integer\ y \longrightarrow integer\ (x - y) = integer\ x)$

thm FORALL_INTEGER:

$\forall P::real \Rightarrow bool.\ (\forall n::nat.\ P\ (real\_of\_nat\ n)) \wedge (\forall x::real.\ P\ x \longrightarrow P\ (-\ x)) \longrightarrow (\forall x::real.\ integer\ x \longrightarrow P\ x)$

thm INTEGER_SUM:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow integer\ (f\,x)) \longrightarrow integer\ (sum\ s\ f)$

thm INTEGER_ABS_MUL_EQ_1:

$\forall (x::real)\ y::real.\ integer\ x \wedge integer\ y \longrightarrow (|x * y| = (1::real)) = (|x| = (1::real) \wedge |y| = (1::real))$

thm DEF_rational:

$rational = (\lambda\_164946{::}real.\ \exists (m{::}real)\ n{::}real.\ integer\ m \wedge integer\ n \wedge n \neq (0{::}real) \wedge \_164946 = m\ /\ n)$

thm rational:

$\forall x{::}real.\ rational\ x = (\exists (m{::}real)\ n{::}real.\ integer\ m \wedge integer\ n \wedge n \neq (0{::}real) \wedge x = m\ /\ n)$

thm RATIONAL_INTEGER:

$\forall x{::}real.\ integer\ x \longrightarrow rational\ x$

thm RATIONAL_NUM:

$\forall n{::}nat.\ rational\ (real\_of\_nat\ n)$

thm RATIONAL_NEG:

$\forall x{::}real.\ rational\ x \longrightarrow rational\ (-\ x)$

thm RATIONAL_ABS:

$\forall x{::}real.\ rational\ x \longrightarrow rational\ |x|$

thm RATIONAL_INV:

$\forall x{::}real.\ rational\ x \longrightarrow rational\ (inverse\_class.inverse\ x)$

thm RATIONAL_ADD:

$\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x + y)$

thm RATIONAL_SUB:

$\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x - y)$

thm RATIONAL_MUL:

$\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x * y)$

thm RATIONAL_DIV:

$\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x\ /\ y)$

thm RATIONAL_POW:

$\forall (x{::}real)\ n{::}nat.\ rational\ x \longrightarrow rational\ x^{n}$

thm RATIONAL_CLOSED:

$(\forall n{::}nat.\ rational\ (real\_of\_nat\ n)) \wedge (\forall x{::}real.\ integer\ x \longrightarrow rational\ x) \wedge (\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x + y)) \wedge (\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x - y)) \wedge (\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x * y)) \wedge (\forall (x{::}real)\ y{::}real.\ rational\ x \wedge rational\ y \longrightarrow rational\ (x\ /\ y)) \wedge (\forall (x{::}real)\ r{::}nat.\ rational\ x \longrightarrow rational\ x^{r}) \wedge (\forall x{::}real.\ rational\ x \longrightarrow rational\ (-\ x)) \wedge (\forall x{::}real.\ rational\ x \longrightarrow rational\ (inverse\_class.inverse\ x)) \wedge (\forall x{::}real.\ rational\ x \longrightarrow rational\ |x|)$

thm RATIONAL_NEG_EQ:

$\forall x{::}real.\ rational\ (-\ x) = rational\ x$

thm RATIONAL_INV_EQ:

$\forall x{::}real.\ rational\ (inverse\_class.inverse\ x) = rational\ x$

thm RATIONAL_ALT:

$\forall x{::}real.\ rational\ x = (\exists (p{::}nat)\ q{::}nat.\ q \neq (0{::}nat) \wedge |x| = real\_of\_nat\ p\ /\ real\_of\_nat\ q)$

thm REAL_TRUNCATE_POS:

$\forall x{\geq}0{::}real.\ \exists (n{::}nat)\ r{::}real.\ (0{::}real) \leq r \wedge r < (1{::}real) \wedge x = real\_of\_nat\ n + r$

thm REAL_TRUNCATE:

$\forall x{::}real.\ \exists (n{::}real)\ r{::}real.\ integer\ n \wedge (0{::}real) \leq r \wedge r < (1{::}real) \wedge x = n + r$

thm DEF_floor:

$HOL\_Light\_Import.floor = (SOME\ n{::}nat \Rightarrow real \Rightarrow real.\ \forall \_165396{::}nat.\ \exists r{::}real \Rightarrow real.\ \forall x{::}real.\ integer\ (n\ \_165396\ x) \wedge (0{::}real) \leq r\ x \wedge r\ x < (1{::}real) \wedge x = n\ \_165396\ x + r\ x)\ (49{::}nat)$

thm DEF_frac:

$frac = (SOME\ r{::}nat \Rightarrow real \Rightarrow real.\ \forall (\_165397{::}nat)\ x{::}real.\ integer\ (HOL\_Light\_Import.floor\ x) \wedge (0{::}real) \leq r\ \_165397\ x \wedge r\ \_165397\ x < (1{::}real) \wedge x = HOL\_Light\_Import.floor\ x + r\ \_165397\ x)\ (50{::}nat)$

thm FLOOR_FRAC:

$\forall x{::}real.\ integer\ (HOL\_Light\_Import.floor\ x) \wedge (0{::}real) \leq frac\ x \wedge frac\ x < (1{::}real) \wedge x = HOL\_Light\_Import.floor\ x + frac\ x$

thm FLOOR_UNIQUE:

$\forall (x{::}real)\ a{::}real.\ (integer\ a \wedge a \leq x \wedge x < a + (1{::}real)) = (HOL\_Light\_Import.floor\ x = a)$

thm FLOOR_EQ_0:

$\forall x{::}real.\ (HOL\_Light\_Import.floor\ x = (0{::}real)) = ((0{::}real) \leq x \wedge x < (1{::}real))$

thm FLOOR:

$\forall x{::}real.\ integer\ (HOL\_Light\_Import.floor\ x) \wedge HOL\_Light\_Import.floor\ x \leq x \wedge x < HOL\_Light\_Import.floor\ x + (1{::}real)$

thm FLOOR_DOUBLE:

$\forall u::real.\ real\_of\_nat\ (2::nat) * HOL\_Light\_Import.floor\ u \le HOL\_Light\_Import.floor\ (real\_of\_nat\ (2::nat) * u) \land HOL\_Light\_Import.floor\ (real\_of\_nat\ (2::nat) * u) \le real\_of\_nat\ (2::nat) * HOL\_Light\_Import.floor\ u + (1::real)$

thm FRAC_FLOOR:

$\forall x::real.\ frac\ x = x - HOL\_Light\_Import.floor\ x$

thm FLOOR_NUM:

$\forall n::nat.\ HOL\_Light\_Import.floor\ (real\_of\_nat\ n) = real\_of\_nat\ n$

thm REAL_LE_FLOOR:

$\forall (x::real)\ n::real.\ integer\ n \longrightarrow (n \le HOL\_Light\_Import.floor\ x) = (n \le x)$

thm REAL_FLOOR_LE:

$\forall (x::real)\ n::real.\ integer\ n \longrightarrow (HOL\_Light\_Import.floor\ x \le n) = (x - (1::real) < n)$

thm FLOOR_POS:

$\forall x \ge 0::real.\ \exists n::nat.\ HOL\_Light\_Import.floor\ x = real\_of\_nat\ n$

thm FLOOR_DIV_DIV:

$\forall (m::nat)\ n::nat.\ m \ne (0::nat) \longrightarrow HOL\_Light\_Import.floor\ (real\_of\_nat\ n\ /\ real\_of\_nat\ m) = real\_of\_nat\ (n\ div\ m)$

thm FLOOR_MONO:

$\forall (x::real)\ y::real.\ x \le y \longrightarrow HOL\_Light\_Import.floor\ x \le HOL\_Light\_Import.floor\ y$

thm REAL_FLOOR_EQ:

$\forall x::real.\ (HOL\_Light\_Import.floor\ x = x) = integer\ x$

thm REAL_FLOOR_LT:

$\forall x::real.\ (HOL\_Light\_Import.floor\ x < x) = (\neg\ integer\ x)$

thm REAL_FRAC_EQ_0:

$\forall x::real.\ (frac\ x = (0::real)) = integer\ x$

thm REAL_FRAC_POS_LT:

$\forall x::real.\ ((0::real) < frac\ x) = (\neg\ integer\ x)$

thm FRAC_NUM:

$\forall n::nat.\ frac\ (real\_of\_nat\ n) = (0::real)$

thm REAL_FLOOR_REFL:

$\forall x::real.\ integer\ x \longrightarrow HOL\_Light\_Import.floor\ x = x$

thm REAL_FRAC_ZERO:

$\forall x$::*real*. *integer* $x \longrightarrow$ *frac* $x = (0$::*real*)

thm REAL_FLOOR_ADD:

$\forall (x$::*real*) y::*real*. *HOL_Light_Import.floor* $(x + y) =$ (*if frac* x + *frac* y $<$ (*1*::*real*) *then HOL_Light_Import.floor* x + *HOL_Light_Import.floor* y *else HOL_Light_Import.floor* x + *HOL_Light_Import.floor* y + (*1*::*real*))

thm REAL_FRAC_ADD:

$\forall (x$::*real*) y::*real*. *frac* $(x + y) =$ (*if frac* x + *frac* y $<$ (*1*::*real*) *then frac* x + *frac* y *else frac* x + *frac* y $-$ (*1*::*real*))

thm FLOOR_POS_LE:

$\forall x$::*real*. ((*0*::*real*) $\leq$ *HOL_Light_Import.floor* x) = ((*0*::*real*) $\leq x$)

thm FRAC_UNIQUE:

$\forall (x$::*real*) a::*real*. (*integer* $(x - a) \wedge$ (*0*::*real*) $\leq a \wedge a <$ (*1*::*real*)) = (*frac* x $= a$)

thm REAL_FRAC_EQ:

$\forall x$::*real*. (*frac* $x = x$) = ((*0*::*real*) $\leq x \wedge x <$ (*1*::*real*))

thm INT_OF_REAL_OF_INT:

$\forall i$::*int*. $\lfloor$*real_of_int* $i\rfloor = i$

thm REAL_OF_INT_OF_REAL:

$\forall x$::*real*. *integer* $x \longrightarrow$ *real_of_int* $\lfloor x\rfloor = x$

thm HAS_SIZE_INTSEG_NUM:

$\forall (m$::*nat*) n::*nat*. *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%421*::*real*. $\exists x$::*real*. *SETSPEC GEN%PVAR%421* (*integer* $x \wedge$ *real_of_nat* $m \leq x \wedge x \leq$ *real_of_nat* n) x)) (n + (*1*::*nat*) $-$ m)

thm FINITE_INTSEG:

$\forall (a$::*real*) b::*real*. *FINITE* (*GSPEC* (λ*GEN%PVAR%423*::*real*. $\exists x$::*real*. *SETSPEC GEN%PVAR%423* (*integer* $x \wedge a \leq x \wedge x \leq b$) x))

thm HAS_SIZE_INTSEG_INT:

$\forall (a$::*real*) b::*real*. *integer* $a \wedge$ *integer* $b \longrightarrow$ *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%426*::*real*. $\exists x$::*real*. *SETSPEC GEN%PVAR%426* (*integer* $x \wedge a \leq x \wedge x \leq b$) x)) (*if* b $< a$ *then* *0*::*nat* *else num_of_int* $\lfloor b - a +$ (*1*::*real*)$\rfloor$)

thm CARD_INTSEG_INT:

$\forall (a$::*real*) b::*real*. *integer* $a \wedge$ *integer* $b \longrightarrow$ *CARD* (*GSPEC* (λ*GEN%PVAR%427*::*real*. $\exists x$::*real*. *SETSPEC GEN%PVAR%427* (*integer* $x \wedge a \leq x \wedge x \leq b$) x)) = (*if* $b < a$ *then* *0*::*nat* *else num_of_int* $\lfloor b - a +$ (*1*::*real*)$\rfloor$)

thm REAL_CARD_INTSEG_INT:

$\forall (a::real)\ b::real.\ integer\ a \wedge integer\ b \longrightarrow real\_of\_nat\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%428::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%428\ (integer\ x \wedge a \le x \wedge x \le b)\ x))) = (if\ b < a\ then\ 0::real\ else\ b - a + (1::real))$

thm INFINITE_INTEGER:

INFINITE integer

thm INFINITE_RATIONAL:

INFINITE rational

thm RATIONAL_CLOSED_conjunct6:

$\forall (x::real)\ r::nat.\ rational\ x \longrightarrow rational\ x^r$

thm RATIONAL_APPROXIMATION:

$\forall (x::real)\ e::real.\ (0::real) < e \longrightarrow (\exists r::real.\ rational\ r \wedge |r - x| < e)$

thm RATIONAL_APPROXIMATION_STRADDLE:

$\forall (x::real)\ e::real.\ (0::real) < e \longrightarrow (\exists (a::real)\ b::real.\ rational\ a \wedge rational\ b \wedge a < x \wedge x < b \wedge |b - a| < e)$

thm product:

$product = iterate\ op\ *$

thm PRODUCT_CLAUSES:

$(\forall f::?'b::type \Rightarrow real.\ product\ EMPTY\ f = (1::real)) \wedge (\forall (x::?'a::type)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ product\ s\ f\ else\ f\ x * product\ s\ f))$

thm PRODUCT_UNION:

$\forall (f::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \wedge DISJOINT\ s\ t \longrightarrow product\ (HOL\_Light\_Import.UNION\ s\ t)\ f = product\ s\ f * product\ t\ f$

thm PRODUCT_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow real)\ s::?'b::type \Rightarrow bool.\ (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow product\ (IMAGE\ f\ s)\ g = product\ s\ (g \circ f)$

thm PRODUCT_ADD_SPLIT:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ (n::nat)\ p::nat.\ m \le n + (1::nat) \longrightarrow product\ (dotdot\ m\ (n + p))\ f = product\ (dotdot\ m\ n)\ f * product\ (dotdot\ (n + (1::nat))\ (n + p))\ f$

thm PRODUCT_CLAUSES_conjunct1:

$\forall (x::?'a::type)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ product\ s\ f\ else\ f\ x * product\ s\ f)$

thm PRODUCT_CLAUSES_conjunct0:

$\forall f::?'a::type \Rightarrow real.\ product\ EMPTY\ f = (1::real)$

thm PRODUCT_POS_LE:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow (0::real) \leq product\ s\ f$

thm PRODUCT_POS_LE_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall x::nat.\ m \leq x \wedge x \leq n \longrightarrow (0::real) \leq f\ x) \longrightarrow (0::real) \leq product\ (dotdot\ m\ n)\ f$

thm PRODUCT_POS_LT:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) < f\ x) \longrightarrow (0::real) < product\ s\ f$

thm PRODUCT_POS_LT_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall x::nat.\ m \leq x \wedge x \leq n \longrightarrow (0::real) < f\ x) \longrightarrow (0::real) < product\ (dotdot\ m\ n)\ f$

thm PRODUCT_OFFSET:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ p::nat.\ product\ (dotdot\ (m + p)\ ((?n::nat) + p))\ f = product\ (dotdot\ m\ ?n)\ (\lambda i::nat.\ f\ (i + p))$

thm PRODUCT_SING:

$\forall (f::?'a::type \Rightarrow real)\ x::?'a::type.\ product\ (INSERT\ x\ EMPTY)\ f = f\ x$

thm PRODUCT_SING_NUMSEG:

$\forall (f::nat \Rightarrow real)\ n::nat.\ product\ (dotdot\ n\ n)\ f = f\ n$

thm PRODUCT_CLAUSES_NUMSEG:

$(\forall m::nat.\ product\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow real) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ (1::real))) \wedge (\forall (m::nat)\ n::nat.\ product\ (dotdot\ m\ (Suc\ n))\ ?f = (if\ m \leq Suc\ n\ then\ product\ (dotdot\ m\ n)\ ?f * ?f\ (Suc\ n)\ else\ product\ (dotdot\ m\ n)\ ?f))$

thm PRODUCT_EQ:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow product\ s\ f = product\ s\ g$

thm PRODUCT_EQ_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\ i = g\ i) \longrightarrow product\ (dotdot\ m\ n)\ f = product\ (dotdot\ m\ n)\ g$

thm PRODUCT_EQ_0:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow (product\ s\ f = (0::real)) = (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x = (0::real))$

thm PRODUCT_EQ_0_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (product\ (dotdot\ m\ n)\ f = (0::real)) = (\exists x \geq m.\ x \leq n \wedge f\,x = (0::real))$

thm PRODUCT_LE:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\,x \wedge f\,x \leq (?g::?'a::type \Rightarrow real)\ x) \longrightarrow product\ s\,f \leq product\ s\ ?g$

thm PRODUCT_LE_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow (0::real) \leq f\,i \wedge f\ i \leq (?g::nat \Rightarrow real)\ i) \longrightarrow product\ (dotdot\ m\ n)\ f \leq product\ (dotdot\ m\ n)\ ?g$

thm PRODUCT_EQ_1:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = (1::real)) \longrightarrow product\ s\,f = (1::real)$

thm PRODUCT_EQ_1_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall i::nat.\ m \leq i \wedge i \leq n \longrightarrow f\,i = (1::real)) \longrightarrow product\ (dotdot\ m\ n)\ f = (1::real)$

thm PRODUCT_MUL:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ s\ (\lambda x::?'a::type.\ f\,x * g\ x) = product\ s\,f * product\ s\ g$

thm PRODUCT_MUL_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ product\ (dotdot\ m\ n)\ (\lambda x::nat.\ f\,x * g\ x) = product\ (dotdot\ m\ n)\ f * product\ (dotdot\ m\ n)\ g$

thm PRODUCT_CONST:

$\forall (c::real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ s\ (\lambda x::?'a::type.\ c) = c^{CARD\ s}$

thm PRODUCT_CONST_NUMSEG:

$\forall (c::real)\ (m::nat)\ n::nat.\ product\ (dotdot\ m\ n)\ (\lambda x::nat.\ c) = c^{n + (1::nat) - m}$

thm PRODUCT_CONST_NUMSEG_1:

$\forall (c::real)\ n::nat.\ product\ (dotdot\ (1::nat)\ n)\ (\lambda x::nat.\ c) = c^n$

thm PRODUCT_INV:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\,s \longrightarrow product\ s\ (\lambda x::?'a::type.\ inverse\_class.inverse\ (f\,x)) = inverse\_class.inverse\ (product\ s\,f)$

thm PRODUCT_DIV:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ s\ (\lambda x::?'a::type.\ f\ x\ /\ g\ x) = product\ s\ f\ /\ product\ s\ g$

thm PRODUCT_DIV_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ product\ (dotdot\ m\ n)\ (\lambda x::nat.\ f\ x\ /\ g\ x) = product\ (dotdot\ m\ n)\ f\ /\ product\ (dotdot\ m\ n)\ g$

thm PRODUCT_ONE:

$\forall s::?'a::type \Rightarrow bool.\ product\ s\ (\lambda n::?'a::type.\ 1::real) = (1::real)$

thm PRODUCT_LE_1:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x \wedge f\ x \leq (1::real)) \longrightarrow product\ s\ f \leq (1::real)$

thm PRODUCT_ABS:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow product\ s\ (\lambda x::?'a::type.\ |f\ x|) = |product\ s\ f|$

thm PRODUCT_CLOSED:

$\forall (P::real \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ P\ (1::real) \wedge (\forall (x::real)\ y::real.\ P\ x \wedge P\ y \longrightarrow P\ (x * y)) \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow P\ (f\ a)) \longrightarrow P\ (product\ s\ f)$

thm DEF_trace:

$trace = (\lambda\_166432::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (\$\ \_166432\ i)\ i))$

thm trace:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ trace\ A = sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (\$\ A\ i)\ i)$

thm TRACE_0:

$trace\ (mat\ (0::nat)) = (0::real)$

thm TRACE_I:

$trace\ (mat\ (1::nat)) = real\_of\_nat\ (dimindex\ HOL\_Light\_Import.UNIV)$

thm TRACE_ADD:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ trace\ (matrix\_add\ A\ B) = trace\ A + trace\ B$

thm TRACE_SUB:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ trace\ (matrix\_sub\ A\ B) = trace\ A - trace\ B$

thm TRACE_MUL_SYM:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ trace\ (matrix\_mul\ A\ B) = trace\ (matrix\_mul\ B\ A)$

thm DEF_det:

$det = (\lambda\_166445::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%431::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%431\ (permutes\ p\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)))\ p))\ (\lambda p::nat \Rightarrow nat.\ sign\ p * product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (\$\ \_166445\ i)\ (p\ i))))$

thm det:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ det\ A = sum\ (GSPEC\ (\lambda GEN\%PVAR\%431::nat \Rightarrow nat.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%431\ (permutes\ p\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)))\ p))\ (\lambda p::nat \Rightarrow nat.\ sign\ p * product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (\$\ A\ i)\ (p\ i)))$

thm IN_DIMINDEX_SWAP:

$\forall (m::nat)\ (n::nat)\ j::nat.\ (1::nat) \le m \wedge m \le dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \le n \wedge n \le dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \le j \wedge j \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (1::nat) \le swap\ (m,\ n)\ j \wedge swap\ (m,\ n)\ j \le dimindex\ HOL\_Light\_Import.UNIV$

thm LAMBDA_BETA_PERM:

$\forall (p::nat \Rightarrow nat)\ i::nat.\ permutes\ p\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) \wedge (1::nat) \le i \wedge i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (lambda\ (?g::nat \Rightarrow ?'a::type))\ (p\ i) = ?g\ (p\ i)$

thm PRODUCT_PERMUTE:

$\forall (f::?'a::type \Rightarrow real)\ (p::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \longrightarrow product\ s\ f = product\ s\ (f \circ p)$

thm PRODUCT_PERMUTE_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (p::nat \Rightarrow nat)\ (m::nat)\ n::nat.\ permutes\ p\ (dotdot\ m\ n) \longrightarrow product\ (dotdot\ m\ n)\ f = product\ (dotdot\ m\ n)\ (f \circ p)$

thm REAL_MUL_SUM:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ (f::?'b::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ FINITE\ s \wedge FINITE\ t \longrightarrow sum\ s\ f * sum\ t\ g = sum\ s\ (\lambda i::?'b::type.\ sum\ t\ (\lambda j::?'a::type.\ f\ i * g\ j))$

thm REAL_MUL_SUM_NUMSEG:

$\forall (m::nat)\ (n::nat)\ (p::nat)\ q::nat.\ sum\ (dotdot\ m\ n)\ (?f::nat \Rightarrow real) * sum\ (dotdot\ p\ q)\ (?g::nat \Rightarrow real) = sum\ (dotdot\ m\ n)\ (\lambda i::nat.\ sum\ (dotdot\ p\ q)\ (\lambda j::nat.\ ?f\ i * ?g\ j))$

thm DET_TRANSP:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *det* (*HOL_Light_Import*.*transp* *A*) = *det* *A*

thm DET_LOWERTRIANGULAR:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *i* < *j* ⟶ $ ($ *A* *i*) *j* = (*0*::*real*)) ⟶ *det* *A* = *product* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import*.*UNIV*)) (λ*i*::*nat*. $ ($ *A* *i*) *i*)

thm DET_UPPERTRIANGULAR:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *j* < *i* ⟶ $ ($ *A* *i*) *j* = (*0*::*real*)) ⟶ *det* *A* = *product* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import*.*UNIV*)) (λ*i*::*nat*. $ ($ *A* *i*) *i*)

thm DET_DIAGONAL:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *i* ≠ *j* ⟶ $ ($ *A* *i*) *j* = (*0*::*real*)) ⟶ *det* *A* = *product* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import*.*UNIV*)) (λ*i*::*nat*. $ ($ *A* *i*) *i*)

thm DET_I:

det (*mat* (*1*::*nat*)) = (*1*::*real*)

thm DET_0:

det (*mat* (*0*::*nat*)) = (*0*::*real*)

thm DET_PERMUTE_ROWS:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *p*::*nat* ⇒ *nat*. *permutes* *p* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import*.*UNIV*)) ⟶ *det* (*lambda* (λ*i*::*nat*. $ *A* (*p* *i*))) = *sign* *p* ∗ *det* *A*

thm DET_PERMUTE_COLUMNS:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *p*::*nat* ⇒ *nat*. *permutes* *p* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import*.*UNIV*)) ⟶ *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. $ ($ *A* *i*) (*p* *j*)))) = *sign* *p* ∗ *det* *A*

thm DET_IDENTICAL_ROWS:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *i* ≠ *j* ∧ *row* *i* *A* = *row* *j* *A* ⟶ *det* *A* = (*0*::*real*)

thm DET_IDENTICAL_COLUMNS:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *i* ≠ *j* ∧ *column* *i* *A* = *column* *j* *A* ⟶ *det* *A* = (*0*::*real*)

thm DET_ZERO_ROW:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ A = vec\ (0::nat) \longrightarrow det\ A = (0::real)$

thm DET_ZERO_COLUMN:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge column\ i\ A = vec\ (0::nat) \longrightarrow det\ A = (0::real)$

thm DET_ROW_ADD:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::nat \Rightarrow (real, ?'a::type)\ cart)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ (lambda\ (\lambda i::nat.\ if\ i = k\ then\ vector\_add\ a\ b\ else\ c\ i)) = det\ (lambda\ (\lambda i::nat.\ if\ i = k\ then\ a\ else\ c\ i)) + det\ (lambda\ (\lambda i::nat.\ if\ i = k\ then\ b\ else\ c\ i))$

thm DET_ROW_MUL:

$\forall (a::(real, ?'a::type)\ cart)\ (b::nat \Rightarrow (real, ?'a::type)\ cart)\ (c::real)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ (lambda\ (\lambda i::nat.\ if\ i = k\ then\ \%\ c\ a\ else\ b\ i)) = c * det\ (lambda\ (\lambda i::nat.\ if\ i = k\ then\ a\ else\ b\ i))$

thm DET_ROW_OPERATION:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?j::nat) \wedge ?j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge i \neq ?j \longrightarrow det\ (lambda\ (\lambda k::nat.\ if\ k = i\ then\ vector\_add\ (row\ i\ A)\ (\%\ (?c::real)\ (row\ ?j\ A))\ else\ row\ k\ A)) = det\ A$

thm DET_ROW_SPAN:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ (i::nat)\ x::(real, ?'a::type)\ cart.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge IN\ x\ (span\ (GSPEC\ (\lambda GEN\%PVAR\%432::(real, ?'a::type)\ cart.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%432\ ((1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge j \neq i)\ (row\ j\ A)))) \longrightarrow det\ (lambda\ (\lambda k::nat.\ if\ k = i\ then\ vector\_add\ (row\ i\ A)\ x\ else\ row\ k\ A)) = det\ A$

thm DET_DEPENDENT_ROWS:

$\forall A::((real, ?'a::type)\ cart, ?'a::type)\ cart.\ dependent\ (rows\ A) \longrightarrow det\ A = (0::real)$

thm DET_DEPENDENT_COLUMNS:

$\forall A::((real, ?'a::type)\ cart, ?'a::type)\ cart.\ dependent\ (columns\ A) \longrightarrow det\ A = (0::real)$

thm DET_LINEAR_ROW_VSUM:

$\forall (a::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (c::nat \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ k::nat.\ FINITE\ s \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV$

⟶ det (lambda (λi::nat. if i = k then vsum s a else c i)) = sum s (λj::?'b::type. det (lambda (λi::nat. if i = k then a j else c i)))

thm BOUNDED_FUNCTIONS_BIJECTIONS_1:

∀p::nat × (nat ⇒ nat). IN p (GSPEC (λGEN%PVAR%434::nat × (nat ⇒ nat). ∃(y::nat) g::nat ⇒ nat. SETSPEC GEN%PVAR%434 (IN y (?s::nat ⇒ bool) ∧ IN g (GSPEC (λGEN%PVAR%433::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%433 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ (?k::nat) ⟶ IN (f i) ?s) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ ?k) ⟶ f i = i)) f))) (y, g))) ⟶ IN (GABS (λf::nat × (nat ⇒ nat) ⇒ nat ⇒ nat. ∀(y::nat) g::nat ⇒ nat. GEQ (f (y, g)) (λi::nat. if i = Suc ?k then y else g i)) p) (GSPEC (λGEN%PVAR%435::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%435 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ Suc ?k ⟶ IN (f i) ?s) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ Suc ?k) ⟶ f i = i)) f)) ∧ (GABS (λf::nat × (nat ⇒ nat) ⇒ nat ⇒ nat. ∀(y::nat) g::nat ⇒ nat. GEQ (f (y, g)) (λi::nat. if i = Suc ?k then y else g i)) p (Suc ?k), λi::nat. if i = Suc ?k then i else GABS (λf::nat × (nat ⇒ nat) ⇒ nat ⇒ nat. ∀(y::nat) g::nat ⇒ nat. GEQ (f (y, g)) (λi::nat. if i = Suc ?k then y else g i)) p i) = p

thm BOUNDED_FUNCTIONS_BIJECTIONS_2:

∀h::nat ⇒ nat. IN h (GSPEC (λGEN%PVAR%436::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%436 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ Suc (?k::nat) ⟶ IN (f i) (?s::nat ⇒ bool)) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ Suc ?k) ⟶ f i = i)) f)) ⟶ IN (h (Suc ?k), λi::nat. if i = Suc ?k then i else h i) (GSPEC (λGEN%PVAR%438::nat × (nat ⇒ nat). ∃(y::nat) g::nat ⇒ nat. SETSPEC GEN%PVAR%438 (IN y ?s ∧ IN g (GSPEC (λGEN%PVAR%437::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%437 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ ?k ⟶ IN (f i) ?s) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ ?k) ⟶ f i = i)) f))) (y, g))) ∧ GABS (λf::nat × (nat ⇒ nat) ⇒ nat ⇒ nat. ∀(y::nat) g::nat ⇒ nat. GEQ (f (y, g)) (λi::nat. if i = Suc ?k then y else g i)) (h (Suc ?k), λi::nat. if i = Suc ?k then i else h i) = h

thm FINITE_BOUNDED_FUNCTIONS:

∀(s::nat ⇒ bool) k::nat. FINITE s ⟶ FINITE (GSPEC (λGEN%PVAR%440::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%440 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ k ⟶ IN (f i) s) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ k) ⟶ f i = i)) f))

thm DET_LINEAR_ROWS_VSUM_LEMMA:

∀(s::nat ⇒ bool) (k::nat) (a::nat ⇒ nat ⇒ (real, ?'a::type) cart) c::nat ⇒ (real, ?'a::type) cart. FINITE s ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ det (lambda (λi::nat. if i ≤ k then vsum s (a i) else c i)) = sum (GSPEC (λGEN%PVAR%442::nat ⇒ nat. ∃f::nat ⇒ nat. SETSPEC GEN%PVAR%442 ((∀i::nat. (1::nat) ≤ i ∧ i ≤ k ⟶ IN (f i) s) ∧ (∀i::nat. ¬ ((1::nat) ≤ i ∧ i ≤ k) ⟶ f i = i)) f)) (λf::nat ⇒ nat. det (lambda (λi::nat. if i ≤ k then a i (f i) else c i)))

thm DET_LINEAR_ROWS_VSUM:

$\forall (s{::}nat \Rightarrow bool)\ a{::}nat \Rightarrow nat \Rightarrow (real,\ ?'a{::}type)\ cart.\ FINITE\ s \longrightarrow det\ (lambda\ (\lambda i{::}nat.\ vsum\ s\ (a\ i))) = sum\ (GSPEC\ (\lambda GEN\%PVAR\%443{::}nat \Rightarrow nat.\ \exists f{::}nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%443\ ((\forall i{::}nat.\ (1{::}nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (f\ i)\ s) \wedge (\forall i{::}nat.\ \neg ((1{::}nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV) \longrightarrow f\ i = i))\ f))\ (\lambda f{::}nat \Rightarrow nat.\ det\ (lambda\ (\lambda i{::}nat.\ a\ i\ (f\ i))))$

thm MATRIX_MUL_VSUM_ALT:

$\forall (A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart)\ B{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ matrix\_mul\ A\ B = lambda\ (\lambda i{::}nat.\ vsum\ (dotdot\ (1{::}nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda k{::}nat.\ \%\ (\$\ (\$\ A\ i)\ k)\ (\$\ B\ k)))$

thm DET_ROWS_MUL:

$\forall (a{::}nat \Rightarrow (real,\ ?'a{::}type)\ cart)\ c{::}nat \Rightarrow real.\ det\ (lambda\ (\lambda i{::}nat.\ \%\ (c\ i)\ (a\ i))) = product\ (dotdot\ (1{::}nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ c * det\ (lambda\ a)$

thm DET_MUL:

$\forall (A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart)\ B{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ det\ (matrix\_mul\ A\ B) = det\ A * det\ B$

thm INVERTIBLE_DET_NZ:

$\forall A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ invertible\ A = (det\ A \neq (0{::}real))$

thm DET_EQ_0:

$\forall A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ (det\ A = (0{::}real)) = (\neg\ invertible\ A)$

thm MATRIX_MUL_LINV:

$\forall A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ det\ A \neq (0{::}real) \longrightarrow matrix\_mul\ (matrix\_inv\ A)\ A = mat\ (1{::}nat)$

thm MATRIX_MUL_RINV:

$\forall A{::}((real,\ ?'a{::}type)\ cart,\ ?'a{::}type)\ cart.\ det\ A \neq (0{::}real) \longrightarrow matrix\_mul\ A\ (matrix\_inv\ A) = mat\ (1{::}nat)$

thm DET_MATRIX_EQ_0:

$\forall f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ linear\ f \longrightarrow (det\ (matrix\ f) = (0{::}real)) = (\neg\ (\exists g{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ linear\ g \wedge f \circ g = id \wedge g \circ f = id))$

thm DET_MATRIX_EQ_0_LEFT:

$\forall f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ linear\ f \longrightarrow (det\ (matrix\ f) = (0{::}real)) = (\neg\ (\exists g{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ linear\ g \wedge g \circ f = id))$

thm DET_MATRIX_EQ_0_RIGHT:

$\forall f$::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ⟶ (*det* (*matrix f*) = (*0*::*real*)) = (¬ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear g* ∧ *f* ∘ *g* = *id*))

thm DET_EQ_0_RANK:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (*det A* = (*0*::*real*)) = (*rank A* < *dimindex HOL_Light_Import.UNIV*)

thm HOMOGENEOUS_LINEAR_EQUATIONS_DET:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (∃ *x*::(*real*, *?'a*::*type*) *cart*. *x* ≠ *vec* (*0*::*nat*) ∧ *matrix_vector_mul A x* = *vec* (*0*::*nat*)) = (*det A* = (*0*::*real*))

thm CRAMER_LEMMA_TRANSP:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. (*1*::*nat*) ≤ (*?k*::*nat*) ∧ *?k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *det* (*lambda* (λ*i*::*nat*. *if i* = *?k then vsum* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) (λ*i*::*nat*. % ($ *x i*) (*row i A*)) *else row i A*)) = $ *x ?k* ∗ *det A*

thm CRAMER_LEMMA:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. (*1*::*nat*) ≤ (*?k*::*nat*) ∧ *?k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if j* = *?k then* $ (*matrix_vector_mul A x*) *i else* $ ($ *A i*) *j*))) = $ *x ?k* ∗ *det A*

thm CRAMER:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *det A* ≠ (*0*::*real*) ⟶ (*matrix_vector_mul A x* = *b*) = (*x* = *lambda* (λ*k*::*nat*. *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if j* = *k then* $ *b i else* $ ($ *A i*) *j*))) / *det A*))

thm CRAMER_MATRIX_LEFT:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*X*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *det A* ≠ (*0*::*real*) ⟶ (*matrix_mul X A* = *B*) = (*X* = *lambda* (λ*k*::*nat*. *lambda* (λ*l*::*nat*. *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if j* = *l then* $ ($ *B k*) *i else* $ ($ *A j*) *i*))) / *det A*)))

thm CRAMER_MATRIX_RIGHT:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*X*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *det A* ≠ (*0*::*real*) ⟶ (*matrix_mul A X* = *B*) = (*X* = *lambda* (λ*k*::*nat*. *lambda* (λ*l*::*nat*. *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if j* = *k then* $ ($ *B i*) *l else* $ ($ *A i*) *j*))) / *det A*)))

thm CRAMER_MATRIX_RIGHT_INVERSE:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *A'*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (*matrix_mul A A'* = *mat* (*1*::*nat*)) = (*det A* ≠ (*0*::*real*) ∧ *A'* = *lambda*

$(\lambda k::nat.\ lambda\ (\lambda l::nat.\ det\ (lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ if\ j = k\ then\ if\ i = l\ then\ 1::real\ else\ (0::real)\ else\ \$\ (\$\ A\ i)\ j)))\ /\ det\ A)))$

thm CRAMER_MATRIX_LEFT_INVERSE:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ A'::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ (matrix\_mul\ A'\ A = mat\ (1::nat)) = (det\ A \neq (0::real) \wedge A' = lambda\ (\lambda k::nat.\ lambda\ (\lambda l::nat.\ det\ (lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ if\ j = l\ then\ if\ i = k\ then\ 1::real\ else\ (0::real)\ else\ \$\ (\$\ A\ j)\ i)))\ /\ det\ A)))$

thm DEF_cofactor:

$cofactor = (\lambda\_171431::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ det\ (lambda\ (\lambda k::nat.\ lambda\ (\lambda l::nat.\ if\ k = i \wedge l = j\ then\ 1::real\ else\ if\ k = i \vee l = j\ then\ 0::real\ else\ \$\ (\$\ \_171431\ k)\ l))))))$

thm cofactor:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ cofactor\ A = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ det\ (lambda\ (\lambda k::nat.\ lambda\ (\lambda l::nat.\ if\ k = i \wedge l = j\ then\ 1::real\ else\ if\ k = i \vee l = j\ then\ 0::real\ else\ \$\ (\$\ A\ k)\ l)))))$

thm COFACTOR_TRANSP:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ cofactor\ (HOL\_Light\_Import.transp\ A) = HOL\_Light\_Import.transp\ (cofactor\ A)$

thm COFACTOR_COLUMN:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ cofactor\ A = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ det\ (lambda\ (\lambda k::nat.\ lambda\ (\lambda l::nat.\ if\ l = j\ then\ if\ k = i\ then\ 1::real\ else\ (0::real)\ else\ \$\ (\$\ A\ k)\ l)))))$

thm COFACTOR_ROW:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ cofactor\ A = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ det\ (lambda\ (\lambda k::nat.\ lambda\ (\lambda l::nat.\ if\ k = i\ then\ if\ l = j\ then\ 1::real\ else\ (0::real)\ else\ \$\ (\$\ A\ k)\ l)))))$

thm MATRIX_RIGHT_INVERSE_COFACTOR:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ A'::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ (matrix\_mul\ A\ A' = mat\ (1::nat)) = (det\ A \neq (0::real) \wedge A' = \%\%\ (inverse\_class.inverse\ (det\ A))\ (HOL\_Light\_Import.transp\ (cofactor\ A)))$

thm MATRIX_LEFT_INVERSE_COFACTOR:

$\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ A'::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ (matrix\_mul\ A'\ A = mat\ (1::nat)) = (det\ A \neq (0::real) \wedge A' = \%\%\ (inverse\_class.inverse\ (det\ A))\ (HOL\_Light\_Import.transp\ (cofactor\ A)))$

thm MATRIX_INV_COFACTOR:

$\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ det\ A \neq (0::real) \longrightarrow matrix\_inv\ A = \%\%\ (inverse\_class.inverse\ (det\ A))\ (HOL\_Light\_Import.transp\ (cofactor\ A))$

thm COFACTOR_MATRIX_INV:

$\forall A::((real, ?'a::type)\ cart, ?'a::type)\ cart.\ det\ A \neq (0::real) \longrightarrow cofactor\ A = \%\%\ (det\ A)\ (HOL\_Light\_Import.transp\ (matrix\_inv\ A))$

thm DET_COFACTOR_EXPANSION:

$\forall (A::((real, ?'a::type)\ cart, ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ A = sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda j::nat.\ \$\ (\$\ A\ i)\ j * \$\ (\$\ (cofactor\ A)\ i)\ j)$

thm MATRIX_MUL_RIGHT_COFACTOR:

$\forall A::((real, ?'a::type)\ cart, ?'a::type)\ cart.\ matrix\_mul\ A\ (HOL\_Light\_Import.transp\ (cofactor\ A)) = \%\%\ (det\ A)\ (mat\ (1::nat))$

thm MATRIX_MUL_LEFT_COFACTOR:

$\forall A::((real, ?'a::type)\ cart, ?'a::type)\ cart.\ matrix\_mul\ (HOL\_Light\_Import.transp\ (cofactor\ A))\ A = \%\%\ (det\ A)\ (mat\ (1::nat))$

thm PRODUCT_1:

$product\ (dotdot\ (1::nat)\ (1::nat))\ (?f::nat \Rightarrow real) = ?f\ (1::nat)$

thm PRODUCT_CLAUSES_NUMSEG_conjunct1:

$\forall (m::nat)\ n::nat.\ product\ (dotdot\ m\ (Suc\ n))\ (?f::nat \Rightarrow real) = (if\ m \leq Suc\ n\ then\ product\ (dotdot\ m\ n)\ ?f * ?f\ (Suc\ n)\ else\ product\ (dotdot\ m\ n)\ ?f)$

thm PRODUCT_CLAUSES_NUMSEG_conjunct0:

$\forall m::nat.\ product\ (dotdot\ m\ (0::nat))\ (?f::nat \Rightarrow real) = (if\ m = (0::nat)\ then\ ?f\ (0::nat)\ else\ (1::real))$

thm PRODUCT_2:

$\forall t::nat \Rightarrow real.\ product\ (dotdot\ (1::nat)\ (2::nat))\ t = t\ (1::nat) * t\ (2::nat)$

thm PRODUCT_3:

$\forall t::nat \Rightarrow real.\ product\ (dotdot\ (1::nat)\ (3::nat))\ t = t\ (1::nat) * (t\ (2::nat) * t\ (3::nat))$

thm DET_1:

$\forall A::((real, unit)\ cart, unit)\ cart.\ det\ A = \$\ (\$\ A\ (1::nat))\ (1::nat)$

thm DET_2:

$\forall A::((real, 2)\ cart, 2)\ cart.\ det\ A = \$\ (\$\ A\ (1::nat))\ (1::nat) * \$\ (\$\ A\ (2::nat))\ (2::nat) - \$\ (\$\ A\ (1::nat))\ (2::nat) * \$\ (\$\ A\ (2::nat))\ (1::nat)$

thm DET_3:

$\forall A::((real, 3)\ cart, 3)\ cart.\ det\ A = \$\ (\$\ A\ (1::nat))\ (1::nat) * (\$\ (\$\ A\ (2::nat))\ (2::nat) * \$\ (\$\ A\ (3::nat))\ (3::nat)) + (\$\ (\$\ A\ (1::nat))\ (2::nat) * (\$\ (\$\ A\ (2::nat))\ (3::nat) * \$\ (\$\ A\ (3::nat))\ (1::nat)) + (\$\ (\$\ A\ (1::nat))\ (3::nat) * (\$$

($ A (2::nat)) (1::nat) ∗ $ ($ A (3::nat)) (2::nat)) − $ ($ A (1::nat)) (1::nat) ∗ ($ ($ A (2::nat)) (3::nat) ∗ $ ($ A (3::nat)) (2::nat)) − $ ($ A (1::nat)) (2::nat) ∗ ($ ($ A (2::nat)) (1::nat) ∗ $ ($ A (3::nat)) (3::nat)) − $ ($ A (1::nat)) (3::nat) ∗ ($ ($ A (2::nat)) (2::nat) ∗ $ ($ A (3::nat)) (1::nat))))

thm GRASSMANN_PLUCKER_2:

∀ (x1::(real, 2) cart) (x2::(real, 2) cart) (y1::(real, 2) cart) y2::(real, 2) cart. det (vector [x1, x2]) ∗ det (vector [y1, y2]) = det (vector [y1, x2]) ∗ det (vector [x1, y2]) + det (vector [y2, x2]) ∗ det (vector [y1, x1])

thm GRASSMANN_PLUCKER_3:

∀ (x1::(real, 3) cart) (x2::(real, 3) cart) (x3::(real, 3) cart) (y1::(real, 3) cart) (y2::(real, 3) cart) y3::(real, 3) cart. det (vector [x1, x2, x3]) ∗ det (vector [y1, y2, y3]) = det (vector [y1, x2, x3]) ∗ det (vector [x1, y2, y3]) + (det (vector [y2, x2, x3]) ∗ det (vector [y1, x1, y3]) + det (vector [y3, x2, x3]) ∗ det (vector [y1, y2, x1]))

thm INTEGER_PRODUCT:

∀ (f::?'a::type ⇒ real) s::?'a::type ⇒ bool. (∀ x::?'a::type. IN x s ⟶ integer (f x)) ⟶ integer (product s f)

thm INTEGER_SIGN:

∀ p::?'a::type ⇒ ?'a::type. integer (sign p)

thm INTEGER_DET:

∀ M::((real, ?'a::type) cart, ?'a::type) cart. (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ⟶ integer ($ ($ M i) j)) ⟶ integer (det M)

thm DEF_diagonal_matrix:

diagonal_matrix = (λ_176890::((real, ?'b::type) cart, ?'a::type) cart. ∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ i ≠ j ⟶ $ ($ _176890 i) j = (0::real))

thm diagonal_matrix:

∀ A::((real, ?'b::type) cart, ?'a::type) cart. diagonal_matrix A = (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ i ≠ j ⟶ $ ($ A i) j = (0::real))

thm TRANSP_DIAGONAL_MATRIX:

∀ A::((real, ?'a::type) cart, ?'a::type) cart. diagonal_matrix A ⟶ HOL_Light_Import.transp A = A

thm DEF_orthogonal_transformation:

orthogonal_transformation = (λ*_176895*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear _176895* ∧ (∀ (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *dot* (*_176895 v*) (*_176895 w*) = *dot v w*))

thm orthogonal_transformation:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* = (*linear f* ∧ (∀ (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *dot* (*f v*) (*f w*) = *dot v w*))

thm ORTHOGONAL_TRANSFORMATION:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* = (*linear f* ∧ (∀ *v*::(*real*, *?'a*::*type*) *cart*. *vector_norm* (*f v*) = *vector_norm v*))

thm ORTHOGONAL_TRANSFORMATION_COMPOSE:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* ∧ *orthogonal_transformation g* ⟶ *orthogonal_transformation* (*f* ∘ *g*)

thm DEF_orthogonal_matrix:

orthogonal_matrix = (λ*_177026*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_mul* (*HOL_Light_Import.transp _177026*) *_177026* = *mat* (*1*::*nat*) ∧ *matrix_mul* *_177026* (*HOL_Light_Import.transp _177026*) = *mat* (*1*::*nat*))

thm orthogonal_matrix:

∀ *Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix Q* = (*matrix_mul* (*HOL_Light_Import.transp Q*) *Q* = *mat* (*1*::*nat*) ∧ *matrix_mul Q* (*HOL_Light_Import.transp Q*) = *mat* (*1*::*nat*))

thm ORTHOGONAL_MATRIX:

orthogonal_matrix (*?Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) = (*matrix_mul* (*HOL_Light_Import.transp ?Q*) *?Q* = *mat* (*1*::*nat*))

thm ORTHOGONAL_MATRIX_ALT:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (*matrix_mul A* (*HOL_Light_Import.transp A*) = *mat* (*1*::*nat*))

thm ORTHOGONAL_MATRIX_ID:

orthogonal_matrix (*mat* (*1*::*nat*))

thm ORTHOGONAL_MATRIX_MUL:

∀ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) *B*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* ∧ *orthogonal_matrix B* ⟶ *orthogonal_matrix* (*matrix_mul A B*)

thm ORTHOGONAL_TRANSFORMATION_MATRIX:

∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* = (*linear f* ∧ *orthogonal_matrix* (*matrix f*))

thm DET_ORTHOGONAL_MATRIX:

$\forall$ *Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix Q* $\longrightarrow$ *det Q* = (*1*::*real*) $\vee$ *det Q* = $-$ (*1*::*real*)

thm ORTHOGONAL_MATRIX_TRANSP:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix* (*HOL_Light_Import.transp A*) = *orthogonal_matrix A*

thm MATRIX_MUL_LTRANSP_DOT_COLUMN:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_mul* (*HOL_Light_Import.transp A*) *A* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *dot* (*column i A*) (*column j A*)))

thm MATRIX_MUL_RTRANSP_DOT_ROW:

$\forall$ *A*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. *matrix_mul A* (*HOL_Light_Import.transp A*) = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *dot* (*row i A*) (*row j A*)))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_COLUMNS:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *vector_norm* (*column i A*) = (*1*::*real*)) $\wedge$ ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *i* $\neq$ *j* $\longrightarrow$ *orthogonal* (*column i A*) (*column j A*)))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_ROWS:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *vector_norm* (*row i A*) = (*1*::*real*)) $\wedge$ ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *i* $\neq$ *j* $\longrightarrow$ *orthogonal* (*row i A*) (*row j A*)))

thm Trigonometry2.POW2_1:

(*1*::*real*)^2 = (*1*::*real*)

thm ORTHOGONAL_MATRIX_ORTHONORMAL_ROWS_INDEXED:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *vector_norm* (*row i A*) = (*1*::*real*)) $\wedge$ *pairwise* (λ(*i*::*nat*) *j*::*nat*. *orthogonal* (*row i A*) (*row j A*)) (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_ROWS_PAIRWISE:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (*CARD* (*rows A*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *vector_norm* (*row i A*) = (*1*::*real*)) $\wedge$ *pairwise orthogonal* (*rows A*))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_ROWS_SPAN:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (*span* (*rows A*) = *HOL_Light_Import.UNIV* ∧ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *vector_norm* (*row i A*) = (*1*::*real*)) ∧ *pairwise orthogonal* (*rows A*))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_COLUMNS_INDEXED:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *vector_norm* (*column i A*) = (*1*::*real*)) ∧ *pairwise* (λ(*i*::*nat*) *j*::*nat*. *orthogonal* (*column i A*) (*column j A*)) (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_COLUMNS_PAIRWISE:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (*CARD* (*columns A*) = *dimindex HOL_Light_Import.UNIV* ∧ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *vector_norm* (*column i A*) = (*1*::*real*)) ∧ *pairwise orthogonal* (*columns A*))

thm ORTHOGONAL_MATRIX_ORTHONORMAL_COLUMNS_SPAN:

∀ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* = (*span* (*columns A*) = *HOL_Light_Import.UNIV* ∧ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *vector_norm* (*column i A*) = (*1*::*real*)) ∧ *pairwise orthogonal* (*columns A*))

thm ORTHOGONAL_MATRIX_2:

∀ *A*::((*real*, *2*) *cart*, *2*) *cart*. *orthogonal_matrix A* = (($ ($ *A* (*1*::*nat*)) (*1*::*nat*))^2 + ($ ($ *A* (*2*::*nat*)) (*1*::*nat*))^2 = (*1*::*real*) ∧ ($ ($ *A* (*1*::*nat*)) (*2*::*nat*))^2 + ($ ($ *A* (*2*::*nat*)) (*2*::*nat*))^2 = (*1*::*real*) ∧ $ ($ *A* (*1*::*nat*)) (*1*::*nat*) ∗ $ ($ *A* (*1*::*nat*)) (*2*::*nat*) + $ ($ *A* (*2*::*nat*)) (*1*::*nat*) ∗ $ ($ *A* (*2*::*nat*)) (*2*::*nat*) = (*0*::*real*))

thm ORTHOGONAL_MATRIX_2_ALT:

∀ *A*::((*real*, *2*) *cart*, *2*) *cart*. *orthogonal_matrix A* = (($ ($ *A* (*1*::*nat*)) (*1*::*nat*))^2 + ($ ($ *A* (*2*::*nat*)) (*1*::*nat*))^2 = (*1*::*real*) ∧ ($ ($ *A* (*1*::*nat*)) (*1*::*nat*) = $ ($ *A* (*2*::*nat*)) (*2*::*nat*) ∧ $ ($ *A* (*1*::*nat*)) (*2*::*nat*) = − $ ($ *A* (*2*::*nat*)) (*1*::*nat*) ∨ $ ($ *A* (*1*::*nat*)) (*1*::*nat*) = − $ ($ *A* (*2*::*nat*)) (*2*::*nat*) ∧ $ ($ *A* (*1*::*nat*)) (*2*::*nat*) = $ ($ *A* (*2*::*nat*)) (*1*::*nat*)))

thm SCALING_LINEAR:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *c*::*real*. *f* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*) ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *distance* (*f x*, *f y*) = *c* ∗ *distance* (*x*, *y*)) ⟶ *linear f*

thm ISOMETRY_LINEAR:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *f* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*) ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *distance* (*f x*, *f y*) = *distance* (*x*, *y*)) ⟶ *linear f*

thm ORTHOGONAL_TRANSFORMATION_ISOMETRY:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation* *f* = (*f* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*) $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *distance* (*f x*, *f y*) = *distance* (*x*, *y*)))

thm ISOMETRY_SPHERE_EXTEND:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *vector_norm x* = (*1*::*real*) $\longrightarrow$ *vector_norm* (*f x*) = (*1*::*real*)) $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *vector_norm x* = (*1*::*real*) $\wedge$ *vector_norm y* = (*1*::*real*) $\longrightarrow$ *distance* (*f x*, *f y*) = *distance* (*x*, *y*)) $\longrightarrow$ ($\exists$ *g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation g* $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *vector_norm x* = (*1*::*real*) $\longrightarrow$ *g x* = *f x*))

thm ORTHOGONAL_TRANSFORMATION_LINEAR:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\longrightarrow$ *linear f*

thm ORTHOGONAL_TRANSFORMATION_INJECTIVE:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\longrightarrow$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*)

thm ORTHOGONAL_TRANSFORMATION_SURJECTIVE:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\longrightarrow$ ($\forall$ *y*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *f x* = *y*)

thm ORTHOGONAL_TRANSFORMATION_INVERSE_o:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\longrightarrow$ ($\exists$ *g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation g* $\wedge$ *g* $\circ$ *f* = *id* $\wedge$ *f* $\circ$ *g* = *id*)

thm ORTHOGONAL_TRANSFORMATION_INVERSE:

$\forall f$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\longrightarrow$ ($\exists$ *g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation g* $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *g* (*f x*) = *x*) $\wedge$ ($\forall$ *y*::(*real*, *?'a*::*type*) *cart*. *f* (*g y*) = *y*))

thm ORTHOGONAL_TRANSFORMATION_ID:

orthogonal_transformation (λ*x*::(*real*, *?'a*::*type*) *cart*. *x*)

thm ORTHOGONAL_TRANSFORMATION_I:

orthogonal_transformation id

thm FINITE_INDEX_NUMSEG_SPECIAL:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) *a*::*?'a*::*type*. *FINITE s* $\wedge$ *IN a s* $\longrightarrow$ ($\exists$ *f*::*nat* $\Rightarrow$ *?'a*::*type*. ($\forall$ (*i*::*nat*) *j*::*nat*. *IN i* (*dotdot* (*1*::*nat*) (*CARD s*)) $\wedge$ *IN j* (*dotdot* (*1*::*nat*) (*CARD s*)) $\wedge$ *f i* = *f j* $\longrightarrow$ *i* = *j*) $\wedge$ *s* = *IMAGE f* (*dotdot* (*1*::*nat*) (*CARD s*)) $\wedge$ *f* (*1*::*nat*) = *a*)

thm ORTHOGONAL_MATRIX_EXISTS_BASIS:

∀ *a*::(*real*, *?'a*::*type*) *cart*. *vector_norm a* = (*1*::*real*) ⟶ (∃ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix A* ∧ *matrix_vector_mul A* (*basis* (*1*::*nat*)) = *a*)

thm ORTHOGONAL_TRANSFORMATION_EXISTS_1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *vector_norm a* = (*1*::*real*) ∧ *vector_norm b* = (*1*::*real*) ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* ∧ *f a* = *b*)

thm ORTHOGONAL_TRANSFORMATION_EXISTS:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *vector_norm a* = *vector_norm b* ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* ∧ *f a* = *b*)

thm DEF_rotation_matrix:

rotation_matrix = (λ*_184211*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix _184211* ∧ *det _184211* = (*1*::*real*))

thm rotation_matrix:

∀ *Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *rotation_matrix Q* = (*orthogonal_matrix Q* ∧ *det Q* = (*1*::*real*))

thm DEF_rotoinversion_matrix:

rotoinversion_matrix = (λ*_184216*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix _184216* ∧ *det _184216* = − (*1*::*real*))

thm rotoinversion_matrix:

∀ *Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *rotoinversion_matrix Q* = (*orthogonal_matrix Q* ∧ *det Q* = − (*1*::*real*))

thm ORTHOGONAL_ROTATION_OR_ROTOINVERSION:

∀ *Q*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix Q* = (*rotation_matrix Q* ∨ *rotoinversion_matrix Q*)

thm ROTATION_MATRIX_2:

∀ *A*::((*real*, *2*) *cart*, *2*) *cart*. *rotation_matrix A* = (($ ($ *A* (*1*::*nat*)) (*1*::*nat*))^2 + ($ ($ *A* (*2*::*nat*)) (*1*::*nat*))^2 = (*1*::*real*) ∧ $ ($ *A* (*1*::*nat*)) (*1*::*nat*) = $ ($ *A* (*2*::*nat*)) (*2*::*nat*) ∧ $ ($ *A* (*1*::*nat*)) (*2*::*nat*) = − $ ($ *A* (*2*::*nat*)) (*1*::*nat*))

thm ROTATION_MATRIX_EXISTS_BASIS:

∀ *a*::(*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ∧ *vector_norm a* = (*1*::*real*) ⟶ (∃ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *rotation_matrix A* ∧ *matrix_vector_mul A* (*basis* (*1*::*nat*)) = *a*)

thm ROTATION_EXISTS_1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ vector_norm a = (1::real) ∧ vector_norm b = (1::real) ⟶ (∃ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ det (matrix f) = (1::real) ∧ f a = b)

thm ROTATION_EXISTS:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ vector_norm a = vector_norm b ⟶ (∃ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ det (matrix f) = (1::real) ∧ f a = b)

thm ROTATION_RIGHTWARD_LINE:

∀ (a::(real, ?'a::type) cart) k::nat. (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ (∃ (b::real) f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ ((2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ det (matrix f) = (1::real)) ∧ f (% b (basis k)) = a ∧ (0::real) ≤ b)

thm ROTATION_LOWDIM_HORIZONTAL:

∀ s::(real, ?'a::type) cart ⇒ bool. dim s < dimindex HOL_Light_Import.UNIV ⟶ (∃ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ det (matrix f) = (1::real) ∧ SUBSET (IMAGE f s) (GSPEC (λGEN%PVAR%448::(real, ?'a::type) cart. ∃ z::(real, ?'a::type) cart. SETSPEC GEN%PVAR%448 ($ z (dimindex HOL_Light_Import.UNIV) = (0::real)) z)))

thm ORTHOGONAL_TRANSFORMATION_LOWDIM_HORIZONTAL:

∀ s::(real, ?'a::type) cart ⇒ bool. dim s < dimindex HOL_Light_Import.UNIV ⟶ (∃ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ SUBSET (IMAGE f s) (GSPEC (λGEN%PVAR%449::(real, ?'a::type) cart. ∃ z::(real, ?'a::type) cart. SETSPEC GEN%PVAR%449 ($ z (dimindex HOL_Light_Import.UNIV) = (0::real)) z)))

thm UNION_UNIV_conjunct1:

∀ s::?'a::type ⇒ bool. HOL_Light_Import.UNION s HOL_Light_Import.UNIV = HOL_Light_Import.UNIV

thm UNION_UNIV_conjunct0:

∀ s::?'a::type ⇒ bool. HOL_Light_Import.UNION HOL_Light_Import.UNIV s = HOL_Light_Import.UNIV

thm ORTHOGONAL_TRANSFORMATION_BETWEEN_ORTHOGONAL_SETS:

∀ (v::nat ⇒ (real, ?'a::type) cart) (w::nat ⇒ (real, ?'a::type) cart) k::nat ⇒ bool. pairwise (λ(i::nat) j::nat. orthogonal (v i) (v j)) k ∧ pairwise (λ(i::nat) j::nat. orthogonal (w i) (w j)) k ∧ (∀ i::nat. IN i k ⟶ vector_norm (v i) = vector_norm (w i)) ⟶ (∃ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. orthogonal_transformation f ∧ (∀ i::nat. IN i k ⟶ f (v i) = w i))

thm COLLINEAR_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ (IMAGE\ (vector\_add\ a)\ s) = collinear\ s$

thm COLLINEAR_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ collinear\ s \longrightarrow collinear\ (IMAGE\ (vector\_add\ a)\ s)$

thm COLLINEAR_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ collinear\ s \wedge linear\ f \longrightarrow collinear\ (IMAGE\ f\ s)$

thm COLLINEAR_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow collinear\ (IMAGE\ f\ s) = collinear\ s$

thm DEF_istopology:

$istopology = (\lambda\_190372::(?'a::type \Rightarrow bool) \Rightarrow bool.\ IN\ EMPTY\ \_190372 \wedge (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ IN\ s\ \_190372 \wedge IN\ t\ \_190372 \longrightarrow IN\ (HOL\_Light\_Import.INTER\ s\ t)\ \_190372) \wedge (\forall k::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ k\ \_190372 \longrightarrow IN\ (UNIONS\ k)\ \_190372))$

thm istopology:

$\forall L::(?'a::type \Rightarrow bool) \Rightarrow bool.\ istopology\ L = (IN\ EMPTY\ L \wedge (\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ IN\ s\ L \wedge IN\ t\ L \longrightarrow IN\ (HOL\_Light\_Import.INTER\ s\ t)\ L) \wedge (\forall k::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ k\ L \longrightarrow IN\ (UNIONS\ k)\ L))$

thm topology_tybij_th:

$\exists t::(?'a::type \Rightarrow bool) \Rightarrow bool.\ istopology\ t$

thm TYDEF_topology:

$topology\ (open\_in\ (?a::?'a::type\ topology)) = ?a \wedge istopology\ (?r::(?'a::type \Rightarrow bool) \Rightarrow bool) = (open\_in\ (topology\ ?r) = ?r)$

thm topology_tybij_conjunct1:

$\forall r::(?'a::type \Rightarrow bool) \Rightarrow bool.\ istopology\ r = (open\_in\ (topology\ r) = r)$

thm topology_tybij_conjunct0:

$\forall a::?'a::type\ topology.\ topology\ (open\_in\ a) = a$

thm topology_tybij:

$(\forall a::?'a::type\ topology.\ topology\ (open\_in\ a) = a) \wedge (\forall r::(?'a::type \Rightarrow bool) \Rightarrow bool.\ istopology\ r = (open\_in\ (topology\ r) = r))$

thm ISTOPOLOGY_OPEN_IN:

istopology (*open_in* (*?top*::*?'a*::*type topology*))

thm TOPOLOGY_EQ:

∀ (*top1*::*?'a*::*type topology*) *top2*::*?'a*::*type topology*. (*top1* = *top2*) = (∀ *s*::*?'a*::*type* ⇒ *bool*. *open_in top1 s* = *open_in top2 s*)

thm DEF_topspace:

topspace = (λ*_190414*::*?'a*::*type topology*. *UNIONS* (*GSPEC* (λ*GEN%PVAR%475*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%475* (*open_in _190414 s*) *s*)))

thm topspace:

∀ *top*::*?'a*::*type topology*. *topspace top* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%475*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%475* (*open_in top s*) *s*))

thm OPEN_IN_CLAUSES:

∀ *top*::*?'a*::*type topology*. *open_in top EMPTY* ∧ (∀ (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *open_in top s* ∧ *open_in top t* ⟶ *open_in top* (*HOL_Light_Import.INTER s t*)) ∧ (∀ *k*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. (∀ *s*::*?'a*::*type* ⇒ *bool*. *IN s k* ⟶ *open_in top s*) ⟶ *open_in top* (*UNIONS k*))

thm OPEN_IN_SUBSET:

∀ (*top*::*?'a*::*type topology*) *s*::*?'a*::*type* ⇒ *bool*. *open_in top s* ⟶ *SUBSET s* (*topspace top*)

thm OPEN_IN_EMPTY:

∀ *top*::*?'a*::*type topology*. *open_in top EMPTY*

thm OPEN_IN_INTER:

∀ (*top*::*?'a*::*type topology*) (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *open_in top s* ∧ *open_in top t* ⟶ *open_in top* (*HOL_Light_Import.INTER s t*)

thm OPEN_IN_UNIONS:

∀ (*top*::*?'a*::*type topology*) *k*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. (∀ *s*::*?'a*::*type* ⇒ *bool*. *IN s k* ⟶ *open_in top s*) ⟶ *open_in top* (*UNIONS k*)

thm OPEN_IN_UNION:

∀ (*top*::*?'a*::*type topology*) (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *open_in top s* ∧ *open_in top t* ⟶ *open_in top* (*HOL_Light_Import.UNION s t*)

thm OPEN_IN_TOPSPACE:

∀ *top*::*?'a*::*type topology*. *open_in top* (*topspace top*)

thm INTER_UNIV_conjunct1:

∀ *s*::*?'a*::*type* ⇒ *bool*. *HOL_Light_Import.INTER s HOL_Light_Import.UNIV* = *s*

thm INTER_UNIV_conjunct0:

$\forall s::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ HOL\_Light\_Import.UNIV\ s = s$

thm OPEN_IN_INTERS:

$\forall (top::?'a::type\ topology)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow open\_in\ top\ t) \longrightarrow open\_in\ top\ (INTERS\ s)$

thm OPEN_IN_SUBOPEN:

$\forall (top::?'a::type\ topology)\ s::?'a::type \Rightarrow bool.\ open\_in\ top\ s = (\forall x::?'a::type.\ IN\ x\ s \longrightarrow (\exists t::?'a::type \Rightarrow bool.\ open\_in\ top\ t \wedge IN\ x\ t \wedge SUBSET\ t\ s))$

thm DEF_closed_in:

$closed\_in = (\lambda(\_190470::?'a::type\ topology)\ \_190471::?'a::type \Rightarrow bool.\ SUBSET\ \_190471\ (topspace\ \_190470) \wedge open\_in\ \_190470\ (DIFF\ (topspace\ \_190470)\ \_190471))$

thm closed_in:

$\forall (top::?'a::type\ topology)\ s::?'a::type \Rightarrow bool.\ closed\_in\ top\ s = (SUBSET\ s\ (topspace\ top) \wedge open\_in\ top\ (DIFF\ (topspace\ top)\ s))$

thm CLOSED_IN_SUBSET:

$\forall (top::?'a::type\ topology)\ s::?'a::type \Rightarrow bool.\ closed\_in\ top\ s \longrightarrow SUBSET\ s\ (topspace\ top)$

thm CLOSED_IN_EMPTY:

$\forall top::?'a::type\ topology.\ closed\_in\ top\ EMPTY$

thm CLOSED_IN_TOPSPACE:

$\forall top::?'a::type\ topology.\ closed\_in\ top\ (topspace\ top)$

thm CLOSED_IN_UNION:

$\forall (top::?'a::type\ topology)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ closed\_in\ top\ s \wedge closed\_in\ top\ t \longrightarrow closed\_in\ top\ (HOL\_Light\_Import.UNION\ s\ t)$

thm CLOSED_IN_INTERS:

$\forall (top::?'a::type\ topology)\ k::(?'a::type \Rightarrow bool) \Rightarrow bool.\ k \neq EMPTY \wedge (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ k \longrightarrow closed\_in\ top\ s) \longrightarrow closed\_in\ top\ (INTERS\ k)$

thm CLOSED_IN_INTER:

$\forall (top::?'a::type\ topology)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ closed\_in\ top\ s \wedge closed\_in\ top\ t \longrightarrow closed\_in\ top\ (HOL\_Light\_Import.INTER\ s\ t)$

thm OPEN_IN_CLOSED_IN_EQ:

$\forall (top::?'a::type\ topology)\ s::?'a::type \Rightarrow bool.\ open\_in\ top\ s = (SUBSET\ s\ (topspace\ top) \land closed\_in\ top\ (DIFF\ (topspace\ top)\ s))$

thm OPEN_IN_CLOSED_IN:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ s\ (topspace\ (?top::?'a::type\ topology)) \longrightarrow open\_in\ ?top\ s = closed\_in\ ?top\ (DIFF\ (topspace\ ?top)\ s)$

thm OPEN_IN_DIFF:

$\forall (top::?'a::type\ topology)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ open\_in\ top\ s \land closed\_in\ top\ t \longrightarrow open\_in\ top\ (DIFF\ s\ t)$

thm CLOSED_IN_DIFF:

$\forall (top::?'a::type\ topology)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ closed\_in\ top\ s \land open\_in\ top\ t \longrightarrow closed\_in\ top\ (DIFF\ s\ t)$

thm CLOSED_IN_UNIONS:

$\forall (top::?'a::type\ topology)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \land (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow closed\_in\ top\ t) \longrightarrow closed\_in\ top\ (UNIONS\ s)$

thm DEF_subtopology:

$subtopology = (\lambda(\_190660::?'a::type\ topology)\ \_190661::?'a::type \Rightarrow bool.\ topology\ (GSPEC\ (\lambda GEN\%PVAR\%477::?'a::type \Rightarrow bool.\ \exists s::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%477\ (open\_in\ \_190660\ s)\ (HOL\_Light\_Import.INTER\ s\ \_190661))))$

thm subtopology:

$\forall (top::?'a::type\ topology)\ u::?'a::type \Rightarrow bool.\ subtopology\ top\ u = topology\ (GSPEC\ (\lambda GEN\%PVAR\%477::?'a::type \Rightarrow bool.\ \exists s::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%477\ (open\_in\ top\ s)\ (HOL\_Light\_Import.INTER\ s\ u)))$

thm ISTOPLOGY_SUBTOPOLOGY:

$\forall (top::?'a::type\ topology)\ u::?'a::type \Rightarrow bool.\ istopology\ (GSPEC\ (\lambda GEN\%PVAR\%480::?'a::type \Rightarrow bool.\ \exists s::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%480\ (open\_in\ top\ s)\ (HOL\_Light\_Import.INTER\ s\ u)))$

thm OPEN_IN_SUBTOPOLOGY:

$\forall (top::?'a::type\ topology)\ (u::?'a::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ open\_in\ (subtopology\ top\ u)\ s = (\exists t::?'a::type \Rightarrow bool.\ open\_in\ top\ t \land s = HOL\_Light\_Import.INTER\ t\ u)$

thm TOPSPACE_SUBTOPOLOGY:

$\forall (top::?'a::type\ topology)\ u::?'a::type \Rightarrow bool.\ topspace\ (subtopology\ top\ u) = HOL\_Light\_Import.INTER\ (topspace\ top)\ u$

thm CLOSED_IN_SUBTOPOLOGY:

$\forall (top::?'a::type\ topology)\ (u::?'a::type \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ closed\_in\ (subtopology\ top\ u)\ s = (\exists\, t::?'a::type \Rightarrow bool.\ closed\_in\ top\ t \wedge s = HOL\_Light\_Import.INTER\ t\ u)$

thm OPEN_IN_SUBTOPOLOGY_REFL:

$\forall (top::?'a::type\ topology)\ u::?'a::type \Rightarrow bool.\ open\_in\ (subtopology\ top\ u)\ u = SUBSET\ u\ (topspace\ top)$

thm CLOSED_IN_SUBTOPOLOGY_REFL:

$\forall (top::?'a::type\ topology)\ u::?'a::type \Rightarrow bool.\ closed\_in\ (subtopology\ top\ u)\ u = SUBSET\ u\ (topspace\ top)$

thm SUBTOPOLOGY_SUPERSET:

$\forall (top::?'a::type\ topology)\ s::?'a::type \Rightarrow bool.\ SUBSET\ (topspace\ top)\ s \longrightarrow subtopology\ top\ s = top$

thm SUBTOPOLOGY_TOPSPACE:

$\forall\, top::?'a::type\ topology.\ subtopology\ top\ (topspace\ top) = top$

thm SUBTOPOLOGY_UNIV:

$\forall\, top::?'a::type\ topology.\ subtopology\ top\ HOL\_Light\_Import.UNIV = top$

thm DEF_open:

$HOL\_Light\_Import.open = (\lambda\_190857::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_190857 \longrightarrow (\exists\, e>0::real.\ \forall\, x'::(real,\ ?'a::type)\ cart.\ distance\ (x',\ x) < e \longrightarrow IN\ x'\ \_190857))$

thm open_def:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists\, e>0::real.\ \forall\, x'::(real,\ ?'a::type)\ cart.\ distance\ (x',\ x) < e \longrightarrow IN\ x'\ s))$

thm DEF_closed:

$HOL\_Light\_Import.closed = (\lambda\_190862::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ (DIFF\ HOL\_Light\_Import.UNIV\ \_190862))$

thm closed:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s = HOL\_Light\_Import.open\ (DIFF\ HOL\_Light\_Import.UNIV\ s)$

thm euclidean:

$euclidean = topology\ HOL\_Light\_Import.open$

thm OPEN_EMPTY:

$HOL\_Light\_Import.open\ EMPTY$

thm OPEN_UNIV:

HOL_Light_Import.open HOL_Light_Import.UNIV

thm OPEN_INTER:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *HOL_Light_Import.open t* ⟶ *HOL_Light_Import.open* (*HOL_Light_Import.INTER s t*)

thm OPEN_UNIONS:

(∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s* (*?f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ *HOL_Light_Import.open s*) ⟶ *HOL_Light_Import.open* (*UNIONS ?f*)

thm OPEN_IN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* = *open_in euclidean s*

thm TOPSPACE_EUCLIDEAN:

topspace euclidean = *HOL_Light_Import.UNIV*

thm TOPSPACE_EUCLIDEAN_SUBTOPOLOGY:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *topspace* (*subtopology euclidean s*) = *s*

thm CLOSED_IN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed s* = *closed_in euclidean s*

thm OPEN_UNION:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *HOL_Light_Import.open t* ⟶ *HOL_Light_Import.open* (*HOL_Light_Import.UNION s t*)

thm OPEN_SUBOPEN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* = (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open t* ∧ *IN x t* ∧ *SUBSET t s*))

thm CLOSED_EMPTY:

HOL_Light_Import.closed EMPTY

thm CLOSED_UNIV:

HOL_Light_Import.closed HOL_Light_Import.UNIV

thm CLOSED_UNION:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed s* ∧ *HOL_Light_Import.closed t* ⟶ *HOL_Light_Import.closed* (*HOL_Light_Import.UNION s t*)

thm CLOSED_INTER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed\ (HOL\_Light\_Import.INTER\ s\ t)$

thm CLOSED_INTERS:

$\forall f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow HOL\_Light\_Import.closed\ s) \longrightarrow HOL\_Light\_Import.closed\ (INTERS\ f)$

thm OPEN_CLOSED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = HOL\_Light\_Import.closed\ (DIFF\ HOL\_Light\_Import.UNIV\ s)$

thm OPEN_DIFF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.open\ (DIFF\ s\ t)$

thm CLOSED_DIFF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow HOL\_Light\_Import.closed\ (DIFF\ s\ t)$

thm OPEN_INTERS:

$\forall s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ s \longrightarrow HOL\_Light\_Import.open\ t) \longrightarrow HOL\_Light\_Import.open\ (INTERS\ s)$

thm CLOSED_UNIONS:

$\forall s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ s \longrightarrow HOL\_Light\_Import.closed\ t) \longrightarrow HOL\_Light\_Import.closed\ (UNIONS\ s)$

thm DEF_ball:

$ball = (\lambda\_190969::(real,\ ?'a::type)\ cart \times real.\ GSPEC\ (\lambda GEN\%PVAR\%481::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%481\ (distance\ (fst\ \_190969,\ y) < snd\ \_190969)\ y))$

thm ball:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ ball\ (x,\ e) = GSPEC\ (\lambda GEN\%PVAR\%481::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%481\ (distance\ (x,\ y) < e)\ y)$

thm DEF_cball:

$cball = (\lambda\_190978::(real,\ ?'a::type)\ cart \times real.\ GSPEC\ (\lambda GEN\%PVAR\%482::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%482\ (distance\ (fst\ \_190978,\ y) \leq snd\ \_190978)\ y))$

thm cball:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ cball\ (x,\ e) = GSPEC\ (\lambda GEN\%PVAR\%482::(real,\ ?'a::type)\ cart.\ \exists\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%482\ (distance\ (x,\ y) \leq e)\ y)$

thm IN_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ y\ (ball\ (x,\ e)) = (distance\ (x,\ y) < e)$

thm IN_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ y\ (cball\ (x,\ e)) = (distance\ (x,\ y) \leq e)$

thm IN_BALL_0:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ x\ (ball\ (vec\ (0::nat),\ e)) = (vector\_norm\ x < e)$

thm IN_CBALL_0:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ x\ (cball\ (vec\ (0::nat),\ e)) = (vector\_norm\ x \leq e)$

thm BALL_TRIVIAL:

$\forall\ x::(real,\ ?'a::type)\ cart.\ ball\ (x,\ 0::real) = EMPTY$

thm CBALL_TRIVIAL:

$\forall\ x::(real,\ ?'a::type)\ cart.\ cball\ (x,\ 0::real) = INSERT\ x\ EMPTY$

thm CENTRE_IN_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ x\ (cball\ (x,\ e)) = ((0::real) \leq e)$

thm BALL_SUBSET_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ SUBSET\ (ball\ (x,\ e))\ (cball\ (x,\ e))$

thm SUBSET_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (d::real)\ e::real.\ d \leq e \longrightarrow SUBSET\ (ball\ (x,\ d))\ (ball\ (x,\ e))$

thm SUBSET_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (d::real)\ e::real.\ d \leq e \longrightarrow SUBSET\ (cball\ (x,\ d))\ (cball\ (x,\ e))$

thm BALL_MAX_UNION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (r::real)\ s::real.\ ball\ (a,\ max\ r\ s) = HOL\_Light\_Import.UNION\ (ball\ (a,\ r))\ (ball\ (a,\ s))$

thm BALL_MIN_INTER:

$\forall (a::(real, ?'a::type)\ cart)\ (r::real)\ s::real.\ ball\ (a,\ min\ r\ s) = HOL\_Light\_Import.INTER\ (ball\ (a,\ r))\ (ball\ (a,\ s))$

thm BALL_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ r::real.\ ball\ (vector\_add\ a\ x,\ r) = IMAGE\ (vector\_add\ a)\ (ball\ (x,\ r))$

thm CBALL_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ r::real.\ cball\ (vector\_add\ a\ x,\ r) = IMAGE\ (vector\_add\ a)\ (cball\ (x,\ r))$

thm BALL_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ r::real.\ linear\ f \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \wedge (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow ball\ (f\ x,\ r) = IMAGE\ f\ (ball\ (x,\ r))$

thm CBALL_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ r::real.\ linear\ f \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \wedge (\forall x::(real, ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow cball\ (f\ x,\ r) = IMAGE\ f\ (cball\ (x,\ r))$

thm BALL_SCALING:

$\forall c>0::real.\ \forall (x::(real, ?'a::type)\ cart)\ r::real.\ ball\ (\%\ c\ x,\ c * r) = IMAGE\ (\%\ c)\ (ball\ (x,\ r))$

thm CBALL_SCALING:

$\forall c>0::real.\ \forall (x::(real, ?'a::type)\ cart)\ r::real.\ cball\ (\%\ c\ x,\ c * r) = IMAGE\ (\%\ c)\ (cball\ (x,\ r))$

thm CBALL_DIFF_BALL:

$\forall (a::(real, ?'a::type)\ cart)\ r::real.\ DIFF\ (cball\ (a,\ r))\ (ball\ (a,\ r)) = GSPEC\ (\lambda GEN\%PVAR\%483::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%483\ (distance\ (a,\ x) = r)\ x)$

thm BALL_UNION_SPHERE:

$\forall (a::(real, ?'a::type)\ cart)\ r::real.\ HOL\_Light\_Import.UNION\ (ball\ (a,\ r))\ (GSPEC\ (\lambda GEN\%PVAR\%484::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%484\ (distance\ (a,\ x) = r)\ x)) = cball\ (a,\ r)$

thm SPHERE_UNION_BALL:

$\forall (a::(real, ?'a::type)\ cart)\ r::real.\ HOL\_Light\_Import.UNION\ (GSPEC\ (\lambda GEN\%PVAR\%485::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%485\ (distance\ (a,\ x) = r)\ x))\ (ball\ (a,\ r)) = cball\ (a,\ r)$

thm OPEN_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ HOL\_Light\_Import.open\ (ball\ (x,\ e))$

thm CENTRE_IN_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ x\ (ball\ (x,\ e)) = ((0::real) < e)$

thm OPEN_CONTAINS_BALL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists e>0::real.\ SUBSET\ (ball\ (x,\ e))\ s))$

thm OPEN_CONTAINS_BALL_EQ:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s = (\exists e>0::real.\ SUBSET\ (ball\ (x,\ e))\ s))$

thm BALL_EQ_EMPTY:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ (ball\ (x,\ e) = EMPTY) = (e \leq (0::real))$

thm BALL_EMPTY:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ e \leq (0::real) \longrightarrow ball\ (x,\ e) = EMPTY$

thm INTER_ACI_conjunct4:

$HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ (HOL\_Light\_Import.INTER\ ?p\ (?q::?'a::type \Rightarrow bool)) = HOL\_Light\_Import.INTER\ ?p\ ?q$

thm INTER_ACI_conjunct3:

$HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ ?p = ?p$

thm INTER_ACI_conjunct2:

$HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ (HOL\_Light\_Import.INTER\ (?q::?'a::type \Rightarrow bool)\ (?r::?'a::type \Rightarrow bool)) = HOL\_Light\_Import.INTER\ ?q\ (HOL\_Light\_Import.INTER\ ?p\ ?r)$

thm INTER_ACI_conjunct1:

$HOL\_Light\_Import.INTER\ (HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ (?q::?'a::type \Rightarrow bool))\ (?r::?'a::type \Rightarrow bool) = HOL\_Light\_Import.INTER\ ?p\ (HOL\_Light\_Import.INTER\ ?q\ ?r)$

thm INTER_ACI_conjunct0:

$HOL\_Light\_Import.INTER\ (?p::?'a::type \Rightarrow bool)\ (?q::?'a::type \Rightarrow bool) = HOL\_Light\_Import.INTER\ ?q\ ?p$

thm OPEN_IN_OPEN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ u)\ s = (\exists t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge s = HOL\_Light\_Import.INTER\ u\ t)$

thm OPEN_IN_INTER_OPEN:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; (t::(real, ?'a::type) \; cart \Rightarrow bool) \; u::(real, ?'a::type) \; cart \Rightarrow bool. \; open\_in \; (subtopology \; euclidean \; u) \; s \wedge HOL\_Light\_Import.open \; t \longrightarrow open\_in \; (subtopology \; euclidean \; u) \; (HOL\_Light\_Import.INTER \; s \; t)$

thm OPEN_IN_OPEN_INTER:

$\forall (u::(real, ?'a::type) \; cart \Rightarrow bool) \; s::(real, ?'a::type) \; cart \Rightarrow bool. \; HOL\_Light\_Import.open \; s \longrightarrow open\_in \; (subtopology \; euclidean \; u) \; (HOL\_Light\_Import.INTER \; u \; s)$

thm OPEN_OPEN_IN_TRANS:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; HOL\_Light\_Import.open \; s \wedge HOL\_Light\_Import.open \; t \wedge SUBSET \; t \; s \longrightarrow open\_in \; (subtopology \; euclidean \; s) \; t$

thm OPEN_SUBSET:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; SUBSET \; s \; t \wedge HOL\_Light\_Import.open \; s \longrightarrow open\_in \; (subtopology \; euclidean \; t) \; s$

thm CLOSED_IN_CLOSED:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; u::(real, ?'a::type) \; cart \Rightarrow bool. \; closed\_in \; (subtopology \; euclidean \; u) \; s = (\exists \, t::(real, ?'a::type) \; cart \Rightarrow bool. \; HOL\_Light\_Import.closed \; t \wedge s = HOL\_Light\_Import.INTER \; u \; t)$

thm CLOSED_IN_INTER_CLOSED:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; (t::(real, ?'a::type) \; cart \Rightarrow bool) \; u::(real, ?'a::type) \; cart \Rightarrow bool. \; closed\_in \; (subtopology \; euclidean \; u) \; s \wedge HOL\_Light\_Import.closed \; t \longrightarrow closed\_in \; (subtopology \; euclidean \; u) \; (HOL\_Light\_Import.INTER \; s \; t)$

thm CLOSED_IN_CLOSED_INTER:

$\forall (u::(real, ?'a::type) \; cart \Rightarrow bool) \; s::(real, ?'a::type) \; cart \Rightarrow bool. \; HOL\_Light\_Import.closed \; s \longrightarrow closed\_in \; (subtopology \; euclidean \; u) \; (HOL\_Light\_Import.INTER \; u \; s)$

thm CLOSED_CLOSED_IN_TRANS:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; HOL\_Light\_Import.closed \; s \wedge HOL\_Light\_Import.closed \; t \wedge SUBSET \; t \; s \longrightarrow closed\_in \; (subtopology \; euclidean \; s) \; t$

thm CLOSED_SUBSET:

$\forall (s::(real, ?'a::type) \; cart \Rightarrow bool) \; t::(real, ?'a::type) \; cart \Rightarrow bool. \; SUBSET \; s \; t \wedge HOL\_Light\_Import.closed \; s \longrightarrow closed\_in \; (subtopology \; euclidean \; t) \; s$

thm open_in:

$\forall (u::(real, ?'a::type) \; cart \Rightarrow bool) \; s::(real, ?'a::type) \; cart \Rightarrow bool. \; open\_in \; (subtopology \; euclidean \; u) \; s = (SUBSET \; s \; u \wedge (\forall \, x::(real, ?'a::type) \; cart. \; IN \; x \; s \longrightarrow (\exists \, e>0::real. \; \forall \, x'::(real, ?'a::type) \; cart. \; IN \; x' \; u \wedge distance \; (x', x) < e \longrightarrow IN \; x' \; s)))$

thm OPEN_IN_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ t)\ s \wedge open\_in\ (subtopology\ euclidean\ u)\ t \longrightarrow open\_in\ (subtopology\ euclidean\ u)\ s$

thm OPEN_IN_OPEN_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ t)\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow HOL\_Light\_Import.open\ s$

thm CLOSED_IN_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ t)\ s \wedge closed\_in\ (subtopology\ euclidean\ u)\ t \longrightarrow closed\_in\ (subtopology\ euclidean\ u)\ s$

thm CLOSED_IN_CLOSED_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ t)\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed\ s$

thm OPEN_IN_SUBTOPOLOGY_INTER_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ u)\ (HOL\_Light\_Import.INTER\ u\ s) \wedge SUBSET\ v\ u \longrightarrow open\_in\ (subtopology\ euclidean\ v)\ (HOL\_Light\_Import.INTER\ v\ s)$

thm OPEN_IN_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (IMAGE\ (vector\_add\ a)\ t))\ (IMAGE\ (vector\_add\ a)\ s) = open\_in\ (subtopology\ euclidean\ t)\ s$

thm CLOSED_IN_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ (IMAGE\ (vector\_add\ a)\ t))\ (IMAGE\ (vector\_add\ a)\ s) = closed\_in\ (subtopology\ euclidean\ t)\ s$

thm OPEN_IN_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow open\_in\ (subtopology\ euclidean\ (IMAGE\ f\ t))\ (IMAGE\ f\ s) = open\_in\ (subtopology\ euclidean\ t)\ s$

thm CLOSED_IN_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow closed\_in\ (subtopology\ euclidean\ (IMAGE\ f\ t))\ (IMAGE\ f\ s) = closed\_in\ (subtopology\ euclidean\ t)\ s$

thm DEF_connected:

connected = (λ_192224::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. ¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open e1* ∧ *HOL_Light_Import.open e2* ∧ *SUBSET _192224* (*HOL_Light_Import.UNION e1 e2*) ∧ *HOL_Light_Import.INTER e1* (*HOL_Light_Import.INTER e2 _192224*) = *EMPTY* ∧ *HOL_Light_Import.INTER e1 _192224* ≠ *EMPTY* ∧ *HOL_Light_Import.INTER e2 _192224* ≠ *EMPTY*))

thm connected:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *connected s* = (¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open e1* ∧ *HOL_Light_Import.open e2* ∧ *SUBSET s* (*HOL_Light_Import.UNION e1 e2*) ∧ *HOL_Light_Import.INTER e1* (*HOL_Light_Import.INTER e2 s*) = *EMPTY* ∧ *HOL_Light_Import.INTER e1 s* ≠ *EMPTY* ∧ *HOL_Light_Import.INTER e2 s* ≠ *EMPTY*))

thm CONNECTED_CLOSED:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *connected s* = (¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed e1* ∧ *HOL_Light_Import.closed e2* ∧ *SUBSET s* (*HOL_Light_Import.UNION e1 e2*) ∧ *HOL_Light_Import.INTER e1* (*HOL_Light_Import.INTER e2 s*) = *EMPTY* ∧ *HOL_Light_Import.INTER e1 s* ≠ *EMPTY* ∧ *HOL_Light_Import.INTER e2 s* ≠ *EMPTY*))

thm CONNECTED_OPEN_IN:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *connected s* = (¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *open_in* (*subtopology euclidean s*) *e1* ∧ *open_in* (*subtopology euclidean s*) *e2* ∧ *SUBSET s* (*HOL_Light_Import.UNION e1 e2*) ∧ *HOL_Light_Import.INTER e1 e2* = *EMPTY* ∧ *e1* ≠ *EMPTY* ∧ *e2* ≠ *EMPTY*))

thm CONNECTED_OPEN_IN_EQ:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *connected s* = (¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *open_in* (*subtopology euclidean s*) *e1* ∧ *open_in* (*subtopology euclidean s*) *e2* ∧ *HOL_Light_Import.UNION e1 e2* = *s* ∧ *HOL_Light_Import.INTER e1 e2* = *EMPTY* ∧ *e1* ≠ *EMPTY* ∧ *e2* ≠ *EMPTY*))

thm CONNECTED_CLOSED_IN:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *connected s* = (¬ (∃ (*e1*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*) *e2*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *closed_in* (*subtopology euclidean s*) *e1* ∧ *closed_in* (*subtopology euclidean s*) *e2* ∧ *SUBSET s* (*HOL_Light_Import.UNION e1 e2*) ∧ *HOL_Light_Import.INTER e1 e2* = *EMPTY* ∧ *e1* ≠ *EMPTY* ∧ *e2* ≠ *EMPTY*))

thm CONNECTED_CLOSED_IN_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\neg (\exists (e1::(real, ?'a::type)\ cart \Rightarrow bool)\ e2::(real, ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ s)\ e1 \wedge closed\_in\ (subtopology\ euclidean\ s)\ e2 \wedge HOL\_Light\_Import.UNION\ e1\ e2 = s \wedge HOL\_Light\_Import.INTER\ e1\ e2 = EMPTY \wedge e1 \neq EMPTY \wedge e2 \neq EMPTY))$

thm CONNECTED_CLOPEN:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\forall t::(real, ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ s)\ t \wedge closed\_in\ (subtopology\ euclidean\ s)\ t \longrightarrow t = EMPTY \vee t = s)$

thm CONNECTED_CLOSED_SET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow connected\ s = (\neg (\exists (e1::(real, ?'a::type)\ cart \Rightarrow bool)\ e2::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ e1 \wedge HOL\_Light\_Import.closed\ e2 \wedge e1 \neq EMPTY \wedge e2 \neq EMPTY \wedge HOL\_Light\_Import.UNION\ e1\ e2 = s \wedge HOL\_Light\_Import.INTER\ e1\ e2 = EMPTY))$

thm CONNECTED_OPEN_SET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow connected\ s = (\neg (\exists (e1::(real, ?'a::type)\ cart \Rightarrow bool)\ e2::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ e1 \wedge HOL\_Light\_Import.open\ e2 \wedge e1 \neq EMPTY \wedge e2 \neq EMPTY \wedge HOL\_Light\_Import.UNION\ e1\ e2 = s \wedge HOL\_Light\_Import.INTER\ e1\ e2 = EMPTY))$

thm CONNECTED_EMPTY:

$connected\ EMPTY$

thm CONNECTED_SING:

$\forall a::(real, ?'a::type)\ cart.\ connected\ (INSERT\ a\ EMPTY)$

thm CONNECTED_UNIONS:

$\forall P::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ P \longrightarrow connected\ s) \wedge INTERS\ P \neq EMPTY \longrightarrow connected\ (UNIONS\ P)$

thm CONNECTED_UNION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s \wedge connected\ t \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow connected\ (HOL\_Light\_Import.UNION\ s\ t)$

thm UNION_ACI_conjunct4:

$HOL\_Light\_Import.UNION\ (?p::?'a::type \Rightarrow bool)\ (HOL\_Light\_Import.UNION\ ?p\ (?q::?'a::type \Rightarrow bool)) = HOL\_Light\_Import.UNION\ ?p\ ?q$

thm UNION_ACI_conjunct3:

HOL_Light_Import.UNION (?p::?'a::type ⇒ bool) ?p = ?p

thm UNION_ACI_conjunct2:

HOL_Light_Import.UNION (?p::?'a::type ⇒ bool) (HOL_Light_Import.UNION (?q::?'a::type ⇒ bool) (?r::?'a::type ⇒ bool)) = HOL_Light_Import.UNION ?q (HOL_Light_Import.UNION ?p ?r)

thm UNION_ACI_conjunct1:

HOL_Light_Import.UNION (HOL_Light_Import.UNION (?p::?'a::type ⇒ bool) (?q::?'a::type ⇒ bool)) (?r::?'a::type ⇒ bool) = HOL_Light_Import.UNION ?p (HOL_Light_Import.UNION ?q ?r)

thm UNION_ACI_conjunct0:

HOL_Light_Import.UNION (?p::?'a::type ⇒ bool) (?q::?'a::type ⇒ bool) = HOL_Light_Import.UNION ?q ?p

thm CONNECTED_DIFF_OPEN_FROM_CLOSED:

∀ (s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) u::(real, ?'a::type) cart ⇒ bool. SUBSET s t ∧ SUBSET t u ∧ HOL_Light_Import.open s ∧ HOL_Light_Import.closed t ∧ connected u ∧ connected (DIFF t s) ⟶ connected (DIFF u s)

thm DEF_limit_point_of:

limit_point_of = (λ(_192774::(real, ?'a::type) cart) _192775::(real, ?'a::type) cart ⇒ bool. ∀ t::(real, ?'a::type) cart ⇒ bool. IN _192774 t ∧ HOL_Light_Import.open t ⟶ (∃ y::(real, ?'a::type) cart. y ≠ _192774 ∧ IN y _192775 ∧ IN y t))

thm limit_point_of:

∀ (x::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. limit_point_of x s = (∀ t::(real, ?'a::type) cart ⇒ bool. IN x t ∧ HOL_Light_Import.open t ⟶ (∃ y::(real, ?'a::type) cart. y ≠ x ∧ IN y s ∧ IN y t))

thm LIMPT_SUBSET:

∀ (x::(real, ?'a::type) cart) (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. limit_point_of x s ∧ SUBSET s t ⟶ limit_point_of x t

thm LIMPT_APPROACHABLE:

∀ (x::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. limit_point_of x s = (∀ e>0::real. ∃ x'::(real, ?'a::type) cart. IN x' s ∧ x' ≠ x ∧ distance (x', x) < e)

thm LIMPT_APPROACHABLE_LE:

∀ (x::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. limit_point_of x s = (∀ e>0::real. ∃ x'::(real, ?'a::type) cart. IN x' s ∧ x' ≠ x ∧ distance (x', x) ≤ e)

thm REAL_HALF_conjunct2:

$\forall$ *e::real. real_of_nat (2::nat) * (e / real_of_nat (2::nat)) = e*

thm REAL_HALF_conjunct1:

$\forall$ *e::real. e / real_of_nat (2::nat) + e / real_of_nat (2::nat) = e*

thm REAL_HALF_conjunct0:

$\forall$ *e::real. ((0::real) < e / real_of_nat (2::nat)) = ((0::real) < e)*

thm LIMPT_UNIV:

$\forall$ *x::(real, ?'a::type) cart. limit_point_of x HOL_Light_Import.UNIV*

thm CLOSED_LIMPT:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. HOL_Light_Import.closed s* = ($\forall$ *x::(real, ?'a::type) cart. limit_point_of x s* $\longrightarrow$ *IN x s*)

thm LIMPT_EMPTY:

$\forall$ *x::(real, ?'a::type) cart.* $\neg$ *limit_point_of x EMPTY*

thm NO_LIMIT_POINT_IMP_CLOSED:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool.* $\neg$ ($\exists$ *x::(real, ?'a::type) cart. limit_point_of x s*) $\longrightarrow$ *HOL_Light_Import.closed s*

thm CLOSED_POSITIVE_ORTHANT:

HOL_Light_Import.closed (GSPEC (λ*GEN%PVAR%487::(real, ?'a::type) cart.* $\exists$ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%487 (*$\forall$ *i::nat. (1::nat)* $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *(0::real)* $\leq$ *\$ x i) x))*

thm FINITE_SET_AVOID:

$\forall$ *(a::(real, ?'a::type) cart) s::(real, ?'a::type) cart* $\Rightarrow$ *bool. FINITE s* $\longrightarrow$ ($\exists$ *d>0::real.* $\forall$ *x::(real, ?'a::type) cart. IN x s* $\wedge$ *x* $\neq$ *a* $\longrightarrow$ *d* $\leq$ *distance (a, x))*

thm LIMIT_POINT_FINITE:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) a::(real, ?'a::type) cart. FINITE s* $\longrightarrow$ $\neg$ *limit_point_of a s*

thm LIMPT_SING:

$\forall$ *(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart.* $\neg$ *limit_point_of x (INSERT y EMPTY)*

thm LIMIT_POINT_UNION:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) (t::(real, ?'a::type) cart* $\Rightarrow$ *bool) x::(real, ?'a::type) cart. limit_point_of x (HOL_Light_Import.UNION s t)* = *(limit_point_of x s* $\vee$ *limit_point_of x t)*

thm DISCRETE_IMP_CLOSED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ e::real.\ (0::real) < e \wedge (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge vector\_norm\ (vector\_sub\ y\ x) < e \longrightarrow y = x) \longrightarrow HOL\_Light\_Import.closed\ s$

thm DEF_interior:

$interior = (\lambda\_193478::(real, ?'a::type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%488::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%488\ (\exists t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge IN\ x\ t \wedge SUBSET\ t\ \_193478)\ x))$

thm interior:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ interior\ s = GSPEC\ (\lambda GEN\%PVAR\%488::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%488\ (\exists t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge IN\ x\ t \wedge SUBSET\ t\ s)\ x)$

thm INTERIOR_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (interior\ s = s) = HOL\_Light\_Import.open\ s$

thm INTERIOR_OPEN:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow interior\ s = s$

thm INTERIOR_EMPTY:

$interior\ EMPTY = EMPTY$

thm INTERIOR_UNIV:

$interior\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm OPEN_INTERIOR:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ (interior\ s)$

thm INTERIOR_INTERIOR:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ interior\ (interior\ s) = interior\ s$

thm INTERIOR_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (interior\ s)\ s$

thm SUBSET_INTERIOR:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (interior\ s)\ (interior\ t)$

thm INTERIOR_MAXIMAL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ t\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow SUBSET\ t\ (interior\ s)$

thm INTERIOR_UNIQUE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *HOL_Light_Import.open t* ∧ (∀ *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t' s* ∧ *HOL_Light_Import.open t'* ⟶ *SUBSET t' t*) ⟶ *interior s* = *t*

thm IN_INTERIOR:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN x* (*interior s*) = (∃ *e*>*0*::*real*. *SUBSET* (*ball* (*x*, *e*)) *s*)

thm OPEN_SUBSET_INTERIOR:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ *SUBSET s* (*interior t*) = *SUBSET s t*

thm INTERIOR_INTER:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *interior* (*HOL_Light_Import.INTER s t*) = *HOL_Light_Import.INTER* (*interior s*) (*interior t*)

thm INTERIOR_FINITE_INTERS:

∀ *s*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE s* ⟶ *interior* (*INTERS s*) = *INTERS* (*IMAGE interior s*)

thm INTERIOR_INTERS_SUBSET:

∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET* (*interior* (*INTERS f*)) (*INTERS* (*IMAGE interior f*))

thm INTERIOR_LIMIT_POINT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*interior s*) ⟶ *limit_point_of x s*

thm INTERIOR_CLOSED_UNION_EMPTY_INTERIOR:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed s* ∧ *interior t* = *EMPTY* ⟶ *interior* (*HOL_Light_Import.UNION s t*) = *interior s*

thm DEF_closure:

closure = (*λ_193790*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.UNION _193790* (*GSPEC* (*λGEN%PVAR%489*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%489* (*limit_point_of x _193790*) *x*)))

thm closure:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *closure s* = *HOL_Light_Import.UNION s* (*GSPEC* (*λGEN%PVAR%489*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%489* (*limit_point_of x s*) *x*))

thm CLOSURE_INTERIOR:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *closure s* = *DIFF HOL_Light_Import.UNIV* (*interior* (*DIFF HOL_Light_Import.UNIV s*))

thm INTERIOR_CLOSURE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *interior s* = *DIFF HOL_Light_Import.UNIV* (*closure* (*DIFF HOL_Light_Import.UNIV s*))

thm CLOSED_CLOSURE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed* (*closure s*)

thm CLOSURE_HULL:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *closure s* = *hull HOL_Light_Import.closed s*

thm CLOSURE_EQ:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*closure s* = *s*) = *HOL_Light_Import.closed s*

thm CLOSURE_CLOSED:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed s* ⟶ *closure s* = *s*

thm CLOSURE_CLOSURE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *closure* (*closure s*) = *closure s*

thm CLOSURE_SUBSET:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s* (*closure s*)

thm SUBSET_CLOSURE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ⟶ *SUBSET* (*closure s*) (*closure t*)

thm CLOSURE_UNION:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *closure* (*HOL_Light_Import.UNION s t*) = *HOL_Light_Import.UNION* (*closure s*) (*closure t*)

thm CLOSURE_INTER_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*closure* (*HOL_Light_Import.INTER s t*)) (*HOL_Light_Import.INTER* (*closure s*) (*closure t*))

thm CLOSURE_INTERS_SUBSET:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET* (*closure* (*INTERS f*)) (*INTERS* (*IMAGE closure f*))

thm CLOSURE_MINIMAL:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ∧ *HOL_Light_Import.closed t* ⟶ *SUBSET* (*closure s*) *t*

thm CLOSURE_MINIMAL_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ t \longrightarrow SUBSET\ (closure\ s)\ t = SUBSET\ s\ t$

thm CLOSURE_UNIQUE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge HOL\_Light\_Import.closed\ t \wedge (\forall t'::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t' \wedge HOL\_Light\_Import.closed\ t' \longrightarrow SUBSET\ t\ t') \longrightarrow closure\ s = t$

thm CLOSURE_EMPTY:

$closure\ EMPTY = EMPTY$

thm CLOSURE_UNIV:

$closure\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm CLOSURE_UNIONS:

$\forall f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \longrightarrow closure\ (UNIONS\ f) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%493::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists s::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%493\ (IN\ s\ f)\ (closure\ s)))$

thm CLOSURE_EQ_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (closure\ s = EMPTY) = (s = EMPTY)$

thm CLOSURE_SUBSET_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (closure\ s)\ s = HOL\_Light\_Import.closed\ s$

thm OPEN_INTER_CLOSURE_EQ_EMPTY:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow (HOL\_Light\_Import.INTER\ s\ (closure\ t) = EMPTY) = (HOL\_Light\_Import.INTER\ s\ t = EMPTY)$

thm OPEN_INTER_CLOSURE_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow SUBSET\ (HOL\_Light\_Import.INTER\ s\ (closure\ t))\ (closure\ (HOL\_Light\_Import.INTER\ s\ t))$

thm CLOSURE_OPEN_INTER_SUPERSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge SUBSET\ s\ (closure\ t) \longrightarrow closure\ (HOL\_Light\_Import.INTER\ s\ t) = closure\ s$

thm CLOSURE_COMPLEMENT:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ closure\ (DIFF\ HOL\_Light\_Import.UNIV\ s) = DIFF\ HOL\_Light\_Import.UNIV\ (interior\ s)$

thm INTERIOR_COMPLEMENT:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ interior\ (DIFF\ HOL\_Light\_Import.UNIV\ s) = DIFF\ HOL\_Light\_Import.UNIV\ (closure\ s)$

thm CONNECTED_INTERMEDIATE_CLOSURE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s \wedge SUBSET\ s\ t \wedge SUBSET\ t\ (closure\ s) \longrightarrow connected\ t$

thm CONNECTED_CLOSURE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s \longrightarrow connected\ (closure\ s)$

thm CONNECTED_UNION_STRONG:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s \wedge connected\ t \wedge HOL\_Light\_Import.INTER\ (closure\ s)\ t \neq EMPTY \longrightarrow connected\ (HOL\_Light\_Import.UNION\ s\ t)$

thm INTERIOR_DIFF:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ interior\ (DIFF\ s\ t) = DIFF\ (interior\ s)\ (closure\ t)$

thm DEF_frontier:

$frontier = (\lambda\_194246::(real, ?'a::type)\ cart \Rightarrow bool.\ DIFF\ (closure\ \_194246)\ (interior\ \_194246))$

thm frontier:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ frontier\ s = DIFF\ (closure\ s)\ (interior\ s)$

thm FRONTIER_CLOSED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ (frontier\ s)$

thm FRONTIER_CLOSURES:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ frontier\ s = HOL\_Light\_Import.INTER\ (closure\ s)\ (closure\ (DIFF\ HOL\_Light\_Import.UNIV\ s))$

thm FRONTIER_STRADDLE:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ (frontier\ s) = (\forall e>0::real.\ (\exists x::(real, ?'a::type)\ cart.\ IN\ x\ s \wedge distance\ (a, x) < e) \wedge (\exists x::(real, ?'a::type)\ cart.\ \neg\ IN\ x\ s \wedge distance\ (a, x) < e))$

thm FRONTIER_SUBSET_CLOSED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow SUBSET\ (frontier\ s)\ s$

thm FRONTIER_EMPTY:

$frontier\ EMPTY = EMPTY$

thm FRONTIER_UNIV:

frontier HOL_Light_Import.UNIV = EMPTY

thm FRONTIER_SUBSET_EQ:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. SUBSET (frontier s) s = HOL_Light_Import.closed s*

thm FRONTIER_COMPLEMENT:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. frontier (DIFF HOL_Light_Import.UNIV s) = frontier s*

thm FRONTIER_DISJOINT_EQ:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. (HOL_Light_Import.INTER (frontier s) s = EMPTY) = HOL_Light_Import.open s*

thm FRONTIER_INTER_SUBSET:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) t::(real, ?'a::type) cart* $\Rightarrow$ *bool. SUBSET (frontier (HOL_Light_Import.INTER s t)) (HOL_Light_Import.UNION (frontier s) (frontier t))*

thm FRONTIER_INTERIORS:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. frontier s = DIFF (DIFF HOL_Light_Import.UNIV (interior s)) (interior (DIFF HOL_Light_Import.UNIV s))*

thm CONNECTED_INTER_FRONTIER:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) t::(real, ?'a::type) cart* $\Rightarrow$ *bool. connected s* $\wedge$ *HOL_Light_Import.INTER s t* $\neq$ *EMPTY* $\wedge$ *DIFF s t* $\neq$ *EMPTY* $\longrightarrow$ *HOL_Light_Import.INTER s (frontier t)* $\neq$ *EMPTY*

thm TYDEF_net:

mk_net (netord (?a::?'a::type net)) = ?a $\wedge$ *(*$\forall$ *(x::?'a::type) y::?'a::type. (*$\forall$ *z::?'a::type. (?r::?'a::type* $\Rightarrow$ *?'a::type* $\Rightarrow$ *bool) z x* $\longrightarrow$ *?r z y)* $\vee$ *(*$\forall$ *z::?'a::type. ?r z y* $\longrightarrow$ *?r z x)) = (netord (mk_net ?r) = ?r)*

thm net_tybij_conjunct1:

$\forall$ *r::?'a::type* $\Rightarrow$ *?'a::type* $\Rightarrow$ *bool. (*$\forall$ *(x::?'a::type) y::?'a::type. (*$\forall$ *z::?'a::type. r z x* $\longrightarrow$ *r z y)* $\vee$ *(*$\forall$ *z::?'a::type. r z y* $\longrightarrow$ *r z x)) = (netord (mk_net r) = r)*

thm net_tybij_conjunct0:

$\forall$ *a::?'a::type net. mk_net (netord a) = a*

thm net_tybij:

($\forall$ *a::?'a::type net. mk_net (netord a) = a)* $\wedge$ *(*$\forall$ *r::?'a::type* $\Rightarrow$ *?'a::type* $\Rightarrow$ *bool. (*$\forall$ *(x::?'a::type) y::?'a::type. (*$\forall$ *z::?'a::type. r z x* $\longrightarrow$ *r z y)* $\vee$ *(*$\forall$ *z::?'a::type. r z y* $\longrightarrow$ *r z x)) = (netord (mk_net r) = r))*

thm NET:

$\forall (n::?'a::type\ net)\ (x::?'a::type)\ y::?'a::type.\ (\forall z::?'a::type.\ netord\ n\ z\ x \longrightarrow netord\ n\ z\ y) \vee (\forall z::?'a::type.\ netord\ n\ z\ y \longrightarrow netord\ n\ z\ x)$

thm OLDNET:

$\forall (n::?'a::type\ net)\ (x::?'a::type)\ y::?'a::type.\ netord\ n\ x\ x \wedge netord\ n\ y\ y \longrightarrow (\exists z::?'a::type.\ netord\ n\ z\ z \wedge (\forall w::?'a::type.\ netord\ n\ w\ z \longrightarrow netord\ n\ w\ x \wedge netord\ n\ w\ y))$

thm NET_DILEMMA:

$\forall net::?'a::type\ net.\ (\exists a::?'a::type.\ (\exists x::?'a::type.\ netord\ net\ x\ a) \wedge (\forall x::?'a::type.\ netord\ net\ x\ a \longrightarrow (?P::?'a::type \Rightarrow bool)\ x)) \wedge (\exists b::?'a::type.\ (\exists x::?'a::type.\ netord\ net\ x\ b) \wedge (\forall x::?'a::type.\ netord\ net\ x\ b \longrightarrow (?Q::?'a::type \Rightarrow bool)\ x)) \longrightarrow (\exists c::?'a::type.\ (\exists x::?'a::type.\ netord\ net\ x\ c) \wedge (\forall x::?'a::type.\ netord\ net\ x\ c \longrightarrow ?P\ x \wedge ?Q\ x))$

thm DEF_at:

$at = (\lambda\_196095::(real,\ ?'a::type)\ cart.\ mk\_net\ (\lambda(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (0::real) < distance\ (x,\ \_196095) \wedge distance\ (x,\ \_196095) \leq distance\ (y,\ \_196095)))$

thm at:

$\forall a::(real,\ ?'a::type)\ cart.\ at\ a = mk\_net\ (\lambda(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (0::real) < distance\ (x,\ a) \wedge distance\ (x,\ a) \leq distance\ (y,\ a))$

thm at_infinity:

$at\_infinity = mk\_net\ (\lambda(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ y \leq vector\_norm\ x)$

thm sequentially:

$sequentially = mk\_net\ (\lambda(m::nat)\ n::nat.\ n \leq m)$

thm DEF_within:

$within = (\lambda(\_196100::?'a::type\ net)\ \_196101::?'a::type \Rightarrow bool.\ mk\_net\ (\lambda(x::?'a::type)\ y::?'a::type.\ netord\ \_196100\ x\ y \wedge IN\ x\ \_196101))$

thm within:

$\forall (net::?'a::type\ net)\ s::?'a::type \Rightarrow bool.\ within\ net\ s = mk\_net\ (\lambda(x::?'a::type)\ y::?'a::type.\ netord\ net\ x\ y \wedge IN\ x\ s)$

thm DEF_in_direction:

$in\_direction = (\lambda(\_196112::(real,\ ?'a::type)\ cart)\ \_196113::(real,\ ?'a::type)\ cart.\ within\ (at\ \_196112)\ (GSPEC\ (\lambda GEN\%PVAR\%494::(real,\ ?'a::type)\ cart.\ \exists b::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%494\ (\exists c \geq 0::real.\ vector\_sub\ b\ \_196112 = \%\ c\ \_196113)\ b)))$

thm in_direction:

$\forall (a::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ in\_direction\ a\ v = within\ (at\ a)\ (GSPEC\ (\lambda GEN\%PVAR\%494::(real,\ ?'a::type)\ cart.\ \exists\, b::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%494\ (\exists\, c \geq 0::real.\ vector\_sub\ b\ a = \%\ c\ v)\ b))$

thm AT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ netord\ (at\ a)\ x\ y = ((0::real) < distance\ (x,\ a) \wedge distance\ (x,\ a) \leq distance\ (y,\ a))$

thm AT_INFINITY:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ netord\ at\_infinity\ x\ y = (vector\_norm\ y \leq vector\_norm\ x)$

thm SEQUENTIALLY:

$\forall (m::nat)\ n::nat.\ netord\ sequentially\ m\ n = (n \leq m)$

thm WITHIN:

$\forall (n::?'a::type\ net)\ (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ netord\ (within\ n\ s)\ x\ y = (netord\ n\ x\ y \wedge IN\ x\ s)$

thm IN_DIRECTION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ netord\ (in\_direction\ a\ v)\ x\ y = ((0::real) < distance\ (x,\ a) \wedge distance\ (x,\ a) \leq distance\ (y,\ a) \wedge (\exists\, c \geq 0::real.\ vector\_sub\ x\ a = \%\ c\ v))$

thm WITHIN_UNIV:

$\forall\, x::(real,\ ?'a::type)\ cart.\ within\ (at\ x)\ HOL\_Light\_Import.UNIV = at\ x$

thm WITHIN_WITHIN:

$\forall (net::?'a::type\ net)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ within\ (within\ net\ s)\ t = within\ net\ (HOL\_Light\_Import.INTER\ s\ t)$

thm DEF_trivial_limit:

$trivial\_limit = (\lambda\_196168::?'a::type\ net.\ (\forall (a::?'a::type)\ b::?'a::type.\ a = b) \vee (\exists (a::?'a::type)\ b::?'a::type.\ a \neq b \wedge (\forall\, x::?'a::type.\ \neg\ netord\ \_196168\ x\ a \wedge \neg\ netord\ \_196168\ x\ b)))$

thm trivial_limit:

$\forall\, net::?'a::type\ net.\ trivial\_limit\ net = ((\forall (a::?'a::type)\ b::?'a::type.\ a = b) \vee (\exists (a::?'a::type)\ b::?'a::type.\ a \neq b \wedge (\forall\, x::?'a::type.\ \neg\ netord\ net\ x\ a \wedge \neg\ netord\ net\ x\ b)))$

thm TRIVIAL_LIMIT_WITHIN:

$\forall\, a::(real,\ ?'a::type)\ cart.\ trivial\_limit\ (within\ (at\ a)\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool)) = (\neg\ limit\_point\_of\ a\ ?s)$

thm TRIVIAL_LIMIT_AT:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. $\neg$ *trivial_limit* (*at a*)

thm TRIVIAL_LIMIT_AT_INFINITY:

$\neg$ *trivial_limit at_infinity*

thm TRIVIAL_LIMIT_SEQUENTIALLY:

$\neg$ *trivial_limit sequentially*

thm LIM_WITHIN_CLOSED_TRIVIAL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed s* $\wedge$ $\neg$ *IN a s* $\longrightarrow$ *trivial_limit* (*within* (*at a*) *s*)

thm NONTRIVIAL_LIMIT_WITHIN:

$\forall$ (*net*::*?'a*::*type net*) *s*::*?'a*::*type* $\Rightarrow$ *bool*. *trivial_limit net* $\longrightarrow$ *trivial_limit* (*within net s*)

thm DEF_eventually:

eventually = (λ(*_196755*::*?'a*::*type* $\Rightarrow$ *bool*) *_196756*::*?'a*::*type net*. *trivial_limit _196756* $\vee$ ($\exists$ *y*::*?'a*::*type*. ($\exists$ *x*::*?'a*::*type*. *netord _196756 x y*) $\wedge$ ($\forall$ *x*::*?'a*::*type*. *netord _196756 x y* $\longrightarrow$ *_196755 x*)))

thm eventually:

$\forall$ (*net*::*?'a*::*type net*) *p*::*?'a*::*type* $\Rightarrow$ *bool*. *eventually p net* = (*trivial_limit net* $\vee$ ($\exists$ *y*::*?'a*::*type*. ($\exists$ *x*::*?'a*::*type*. *netord net x y*) $\wedge$ ($\forall$ *x*::*?'a*::*type*. *netord net x y* $\longrightarrow$ *p x*)))

thm EVENTUALLY_HAPPENS:

$\forall$ (*net*::*?'a*::*type net*) *p*::*?'a*::*type* $\Rightarrow$ *bool*. *eventually p net* $\longrightarrow$ *trivial_limit net* $\vee$ ($\exists$ *x*::*?'a*::*type*. *p x*)

thm EVENTUALLY_WITHIN_LE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *eventually p* (*within* (*at a*) *s*) = ($\exists$ *d*>*0*::*real*. $\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ (*0*::*real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) $\leq$ *d* $\longrightarrow$ *p x*)

thm EVENTUALLY_WITHIN:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *eventually p* (*within* (*at a*) *s*) = ($\exists$ *d*>*0*::*real*. $\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ (*0*::*real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) < *d* $\longrightarrow$ *p x*)

thm EVENTUALLY_AT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *eventually p* (*at a*) = ($\exists$ *d*>*0*::*real*. $\forall$ *x*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) < *d* $\longrightarrow$ *p x*)

thm EVENTUALLY_SEQUENTIALLY:

$\forall\, p$::*nat* $\Rightarrow$ *bool*. *eventually p sequentially* = ($\exists\, N$::*nat*. $\forall\, n \geq N$. *p n*)

thm EVENTUALLY_AT_INFINITY:

$\forall\, p$::(*real*, ?'*a*::*type*) *cart* $\Rightarrow$ *bool*. *eventually p at_infinity* = ($\exists\, b$::*real*. $\forall\, x$::(*real*, ?'*a*::*type*) *cart*. $b \leq$ *vector_norm x* $\longrightarrow$ *p x*)

thm ALWAYS_EVENTUALLY:

($\forall\, x$::?'*a*::*type*. (?*p*::?'*a*::*type* $\Rightarrow$ *bool*) *x*) $\longrightarrow$ *eventually* ?*p* (?*net*::?'*a*::*type net*)

thm EVENTUALLY_AND:

$\forall$ (*net*::?'*a*::*type net*) (*p*::?'*a*::*type* $\Rightarrow$ *bool*) *q*::?'*a*::*type* $\Rightarrow$ *bool*. *eventually* (λx::?'*a*::*type*. *p x* $\wedge$ *q x*) *net* = (*eventually p net* $\wedge$ *eventually q net*)

thm EVENTUALLY_MONO:

$\forall$ (*net*::?'*a*::*type net*) (*p*::?'*a*::*type* $\Rightarrow$ *bool*) *q*::?'*a*::*type* $\Rightarrow$ *bool*. ($\forall\, x$::?'*a*::*type*. *p x* $\longrightarrow$ *q x*) $\wedge$ *eventually p net* $\longrightarrow$ *eventually q net*

thm EVENTUALLY_MP:

$\forall$ (*net*::?'*a*::*type net*) (*p*::?'*a*::*type* $\Rightarrow$ *bool*) *q*::?'*a*::*type* $\Rightarrow$ *bool*. *eventually* (λx::?'*a*::*type*. *p x* $\longrightarrow$ *q x*) *net* $\wedge$ *eventually p net* $\longrightarrow$ *eventually q net*

thm EVENTUALLY_FALSE:

$\forall$ *net*::?'*a*::*type net*. *eventually* (λx::?'*a*::*type*. *False*) *net* = *trivial_limit net*

thm EVENTUALLY_TRUE:

$\forall$ *net*::?'*a*::*type net*. *eventually* (λx::?'*a*::*type*. *True*) *net* = *True*

thm NOT_EVENTUALLY:

$\forall$ (*net*::?'*a*::*type net*) *p*::?'*a*::*type* $\Rightarrow$ *bool*. ($\forall\, x$::?'*a*::*type*. $\neg$ *p x*) $\wedge$ $\neg$ *trivial_limit net* $\longrightarrow$ $\neg$ *eventually p net*

thm EVENTUALLY_FORALL:

$\forall$ (*net*::?'*b*::*type net*) (*p*::?'*a*::*type* $\Rightarrow$?'*b*::*type* $\Rightarrow$ *bool*) *s*::?'*a*::*type* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *eventually* (λx::?'*b*::*type*. $\forall\, a$::?'*a*::*type*. *IN a s* $\longrightarrow$ *p a x*) *net* = ($\forall\, a$::?'*a*::*type*. *IN a s* $\longrightarrow$ *eventually* (*p a*) *net*)

thm FORALL_EVENTUALLY:

$\forall$ (*net*::?'*b*::*type net*) (*p*::?'*a*::*type* $\Rightarrow$?'*b*::*type* $\Rightarrow$ *bool*) *s*::?'*a*::*type* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\forall\, a$::?'*a*::*type*. *IN a s* $\longrightarrow$ *eventually* (*p a*) *net*) = *eventually* (λx::?'*b*::*type*. $\forall\, a$::?'*a*::*type*. *IN a s* $\longrightarrow$ *p a x*) *net*

thm DEF_-->:

--> = (λ(*_197051*::?'*b*::*type* $\Rightarrow$ (*real*, ?'*a*::*type*) *cart*) (*_197052*::(*real*, ?'*a*::*type*) *cart*) *_197053*::?'*b*::*type net*. $\forall\, e>0$::*real*. *eventually* (λx::?'*b*::*type*. *distance* (*_197051 x*, *_197052*) < *e*) *_197053*)

thm tendsto:

∀(f::?'b::type ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) net::?'b::type net. −−> f l net = (∀e>0::real. eventually (λx::?'b::type. distance (f x, l) < e) net)

thm DEF_lim:

lim = (λ(_197072::?'b::type net) _197073::?'b::type ⇒ (real, ?'a::type) cart. SOME l::(real, ?'a::type) cart. −−> _197073 l _197072)

thm lim:

∀(f::?'b::type ⇒ (real, ?'a::type) cart) net::?'b::type net. lim net f = (SOME l::(real, ?'a::type) cart. −−> f l net)

thm LIM:

−−> (?f::?'b::type ⇒ (real, ?'a::type) cart) (?l::(real, ?'a::type) cart) (?net::?'b::type net) = (trivial_limit ?net ∨ (∀e>0::real. ∃y::?'b::type. (∃x::?'b::type. netord ?net x y) ∧ (∀x::?'b::type. netord ?net x y ⟶ distance (?f x, ?l) < e)))

thm LIM_WITHIN_LE:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) (a::(real, ?'b::type) cart) s::(real, ?'b::type) cart ⇒ bool. −−> f l (within (at a) s) = (∀e>0::real. ∃d>0::real. ∀x::(real, ?'b::type) cart. IN x s ∧ (0::real) < distance (x, a) ∧ distance (x, a) ≤ d ⟶ distance (f x, l) < e)

thm LIM_WITHIN:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) (a::(real, ?'b::type) cart) s::(real, ?'b::type) cart ⇒ bool. −−> f l (within (at a) s) = (∀e>0::real. ∃d>0::real. ∀x::(real, ?'b::type) cart. IN x s ∧ (0::real) < distance (x, a) ∧ distance (x, a) < d ⟶ distance (f x, l) < e)

thm LIM_AT:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) a::(real, ?'b::type) cart. −−> f l (at a) = (∀e>0::real. ∃d>0::real. ∀x::(real, ?'b::type) cart. (0::real) < distance (x, a) ∧ distance (x, a) < d ⟶ distance (f x, l) < e)

thm LIM_AT_INFINITY:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) l::(real, ?'a::type) cart. −−> f l at_infinity = (∀e>0::real. ∃b::real. ∀x::(real, ?'b::type) cart. b ≤ vector_norm x ⟶ distance (f x, l) < e)

thm LIM_SEQUENTIALLY:

∀(s::nat ⇒ (real, ?'a::type) cart) l::(real, ?'a::type) cart. −−> s l sequentially = (∀e>0::real. ∃N::nat. ∀n≥N. distance (s n, l) < e)

thm LIM_EVENTUALLY:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) *l*::(*real*, ?'a::*type*) *cart*. *eventually* (λx::?'b::*type*. *f x* = *l*) *net* $\longrightarrow$ --> *f l net*

thm LIM_WITHIN_EMPTY:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) *x*::(*real*, ?'b::*type*) *cart*. --> *f l* (*within* (*at x*) *EMPTY*)

thm LIM_WITHIN_SUBSET:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) (*a*::(*real*, ?'b::*type*) *cart*) *s*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*. --> *f l* (*within* (*at a*) *s*) $\wedge$ *SUBSET* (?t::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*) *s* $\longrightarrow$ --> *f l* (*within* (*at a*) ?t)

thm LIM_UNION:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*x*::(*real*, ?'b::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) (*s*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*. --> *f l* (*within* (*at x*) *s*) $\wedge$ --> *f l* (*within* (*at x*) *t*) $\longrightarrow$ --> *f l* (*within* (*at x*) (*HOL_Light_Import.UNION s t*))

thm LIM_UNION_UNIV:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*x*::(*real*, ?'b::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) (*s*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*. --> *f l* (*within* (*at x*) *s*) $\wedge$ --> *f l* (*within* (*at x*) *t*) $\wedge$ *HOL_Light_Import.UNION s t* = *HOL_Light_Import.UNIV* $\longrightarrow$ --> *f l* (*at x*)

thm LIM_AT_WITHIN:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) (*a*::(*real*, ?'b::*type*) *cart*) *s*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*. --> *f l* (*at a*) $\longrightarrow$ --> *f l* (*within* (*at a*) *s*)

thm LIM_WITHIN_OPEN:

$\forall$ (*f*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) (*a*::(*real*, ?'b::*type*) *cart*) *s*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ *bool*. *IN a s* $\wedge$ *HOL_Light_Import.open s* $\longrightarrow$ --> *f l* (*within* (*at a*) *s*) = --> *f l* (*at a*)

thm LIMPT_SEQUENTIAL:

$\forall$ (*x*::(*real*, ?'a::*type*) *cart*) *s*::(*real*, ?'a::*type*) *cart* $\Rightarrow$ *bool*. *limit_point_of x s* = ($\exists$ *f*::*nat* $\Rightarrow$ (*real*, ?'a::*type*) *cart*. ($\forall$ *n*::*nat*. *IN* (*f n*) (*DELETE s x*)) $\wedge$ --> *f x sequentially*)

thm LIM_LINEAR:

$\forall$ (*net*::?'c::*type net*) (*h*::(*real*, ?'b::*type*) *cart* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*f*::?'c::*type* $\Rightarrow$ (*real*, ?'b::*type*) *cart*) *l*::(*real*, ?'b::*type*) *cart*. --> *f l net* $\wedge$ *linear h* $\longrightarrow$ --> (λx::?'c::*type*. *h* (*f x*)) (*h l*) *net*

thm LIM_CONST:

∀ (*net*::*?'b*::*type net*) *a*::(*real*, *?'a*::*type*) *cart*. −−> (λ*x*::*?'b*::*type*. *a*) *a net*

thm LIM_CMUL:

∀ (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. −−> *f l* (*?net*::*?'b*::*type net*) ⟶ −−> (λ*x*::*?'b*::*type*. % *c* (*f x*)) (% *c l*) *?net*

thm LIM_CMUL_EQ:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. *c* ≠ (*0*::*real*) ⟶ −−> (λ*x*::*?'b*::*type*. % *c* (*f x*)) (% *c l*) *net* = −−> *f l net*

thm LIM_NEG:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. −−> *f l net* ⟶ −−> (λ*x*::*?'b*::*type*. *vector_neg* (*f x*)) (*vector_neg l*) *net*

thm LIM_NEG_EQ:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. −−> (λ*x*::*?'b*::*type*. *vector_neg* (*f x*)) (*vector_neg l*) *net* = −−> *f l net*

thm LIM_ADD:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *m*::(*real*, *?'a*::*type*) *cart*. −−> *f l net* ∧ −−> *g m net* ⟶ −−> (λ*x*::*?'b*::*type*. *vector_add* (*f x*) (*g x*)) (*vector_add l m*) *net*

thm LIM_ABS:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. −−> *f l net* ⟶ −−> (λ*x*::*?'b*::*type*. *lambda* (λ*i*::*nat*. |$ (*f x*) *i*|)) (*lambda* (λ*i*::*nat*. |$ *l i*|)) *net*

thm LIM_SUB:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *m*::(*real*, *?'a*::*type*) *cart*. −−> *f l net* ∧ −−> *g m net* ⟶ −−> (λ*x*::*?'b*::*type*. *vector_sub* (*f x*) (*g x*)) (*vector_sub l m*) *net*

thm LIM_MAX:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *m*::(*real*, *?'a*::*type*) *cart*. −−> *f l net* ∧ −−> *g m net* ⟶ −−> (λ*x*::*?'b*::*type*. *lambda* (λ*i*::*nat*. *max* ($ (*f x*) *i*) ($ (*g x*) *i*))) (*lambda* (λ*i*::*nat*. *max* ($ *l i*) ($ *m i*))) *net*

thm LIM_MIN:

∀ (*net*::*?'b*::*type net*) (*f*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *m*::(*real*, *?'a*::*type*) *cart*. −−> *f l net*

∧ --> g m net ⟶ --> (λx::?'b::type. lambda (λi::nat. min ($ (f x) i) ($ (g x) i))) (lambda (λi::nat. min ($ l i) ($ m i))) net

thm LIM_NULL:

∀(net::?'b::type net) (f::?'b::type ⇒ (real, ?'a::type) cart) l::(real, ?'a::type) cart. --> f l net = --> (λx::?'b::type. vector_sub (f x) l) (vec (0::nat)) net

thm LIM_NULL_NORM:

∀(net::?'b::type net) f::?'b::type ⇒ (real, ?'a::type) cart. --> f (vec (0::nat)) net = --> (λx::?'b::type. lift (vector_norm (f x))) (vec (0::nat)) net

thm LIM_NULL_CMUL_EQ:

∀(net::?'b::type net) (f::?'b::type ⇒ (real, ?'a::type) cart) c::real. c ≠ (0::real) ⟶ --> (λx::?'b::type. % c (f x)) (vec (0::nat)) net = --> f (vec (0::nat)) net

thm LIM_NULL_COMPARISON:

∀(net::?'b::type net) (f::?'b::type ⇒ (real, ?'a::type) cart) g::?'b::type ⇒ real. eventually (λx::?'b::type. vector_norm (f x) ≤ g x) net ∧ --> (λx::?'b::type. lift (g x)) (vec (0::nat)) net ⟶ --> f (vec (0::nat)) net

thm LIM_COMPONENT:

∀(net::?'b::type net) (f::?'b::type ⇒ (real, ?'a::type) cart) (i::nat) l::(real, ?'a::type) cart. --> f l net ∧ (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ --> (λa::?'b::type. lift ($ (f a) i)) (lift ($ l i)) net

thm LIM_TRANSFORM_BOUND:

∀(f::?'c::type ⇒ (real, ?'b::type) cart) g::?'c::type ⇒ (real, ?'a::type) cart. eventually (λn::?'c::type. vector_norm (f n) ≤ vector_norm (g n)) (?net::?'c::type net) ∧ --> g (vec (0::nat)) ?net ⟶ --> f (vec (0::nat)) ?net

thm LIM_NULL_CMUL_BOUNDED:

∀(f::?'b::type ⇒ real) (g::?'b::type ⇒ (real, ?'a::type) cart) B::real. eventually (λa::?'b::type. |f a| ≤ B) (?net::?'b::type net) ∧ --> g (vec (0::nat)) ?net ⟶ --> (λn::?'b::type. % (f n) (g n)) (vec (0::nat)) ?net

thm LIM_NULL_VMUL_BOUNDED:

∀(f::?'b::type ⇒ real) (g::?'b::type ⇒ (real, ?'a::type) cart) B::real. --> (lift ∘ f) (vec (0::nat)) (?net::?'b::type net) ∧ eventually (λa::?'b::type. vector_norm (g a) ≤ B) ?net ⟶ --> (λn::?'b::type. % (f n) (g n)) (vec (0::nat)) ?net

thm LIM_VSUM:

∀(f::?'c::type ⇒ ?'b::type ⇒ (real, ?'a::type) cart) s::?'c::type ⇒ bool. FINITE s ∧ (∀i::?'c::type. IN i s ⟶ --> (f i) ((?l::?'c::type ⇒ (real, ?'a::type) cart) i) (?net::?'b::type net)) ⟶ --> (λx::?'b::type. vsum s (λi::?'c::type. f i x)) (vsum s ?l) ?net

thm LIM_IN_CLOSED_SET:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*s*::(*real*, ?'a::*type*) *cart* $\Rightarrow$ *bool*) *l*::(*real*, ?'a::*type*) *cart*. *HOL_Light_Import.closed s* $\wedge$ *eventually* (λx::?'b::*type*. *IN* (*f x*) *s*) *net* $\wedge$ $\neg$ *trivial_limit net* $\wedge$ --> *f l net* $\longrightarrow$ *IN l s*

thm LIM_NORM_UBOUND:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) *b*::*real*. $\neg$ *trivial_limit net* $\wedge$ --> *f l net* $\wedge$ *eventually* (λx::?'b::*type*. *vector_norm* (*f x*) $\leq$ *b*) *net* $\longrightarrow$ *vector_norm l* $\leq$ *b*

thm LIM_NORM_LBOUND:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) *b*::*real*. $\neg$ *trivial_limit net* $\wedge$ --> *f l net* $\wedge$ *eventually* (λx::?'b::*type*. *b* $\leq$ *vector_norm* (*f x*)) *net* $\longrightarrow$ *b* $\leq$ *vector_norm l*

thm LIM_UNIQUE:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::(*real*, ?'a::*type*) *cart*) *l'*::(*real*, ?'a::*type*) *cart*. $\neg$ *trivial_limit net* $\wedge$ --> *f l net* $\wedge$ --> *f l'* *net* $\longrightarrow$ *l* = *l'*

thm TENDSTO_LIM:

$\forall$ (*net*::?'b::*type net*) (*f*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) *l*::(*real*, ?'a::*type*) *cart*. $\neg$ *trivial_limit net* $\wedge$ --> *f l net* $\longrightarrow$ *lim net f* = *l*

thm LIM_CONST_EQ:

$\forall$ (*net*::?'b::*type net*) (*c*::(*real*, ?'a::*type*) *cart*) *d*::(*real*, ?'a::*type*) *cart*. --> (λx::?'b::*type*. *c*) *d net* = (*trivial_limit net* $\vee$ *c* = *d*)

thm UNIFORM_LIM_ADD:

$\forall$ (*net*::?'c::*type net*) (*P*::?'b::*type* $\Rightarrow$ *bool*) (*f*::?'b::*type* $\Rightarrow$?'c::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*g*::?'b::*type* $\Rightarrow$?'c::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) *m*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*. ($\forall$ *e*>*0*::*real*. *eventually* (λx::?'c::*type*. $\forall$ *n*::?'b::*type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*f n x*) (*l n*)) < *e*) *net*) $\wedge$ ($\forall$ *e*>*0*::*real*. *eventually* (λx::?'c::*type*. $\forall$ *n*::?'b::*type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*g n x*) (*m n*)) < *e*) *net*) $\longrightarrow$ ($\forall$ *e*>*0*::*real*. *eventually* (λx::?'c::*type*. $\forall$ *n*::?'b::*type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*vector_add* (*f n x*) (*g n x*)) (*vector_add* (*l n*) (*m n*))) < *e*) *net*)

thm UNIFORM_LIM_SUB:

$\forall$ (*net*::?'c::*type net*) (*P*::?'b::*type* $\Rightarrow$ *bool*) (*f*::?'b::*type* $\Rightarrow$?'c::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*g*::?'b::*type* $\Rightarrow$?'c::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) (*l*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*) *m*::?'b::*type* $\Rightarrow$ (*real*, ?'a::*type*) *cart*. ($\forall$ *e*>*0*::*real*. *eventually* (λx::?'c::*type*. $\forall$ *n*::?'b::*type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*f n x*) (*l n*)) < *e*) *net*) $\wedge$ ($\forall$ *e*>*0*::*real*. *eventually* (λx::?'c::*type*. $\forall$ *n*::?'b::*type*. *P n*

⟶ vector_norm (vector_sub (g n x) (m n)) < e) net) ⟶ (∀ e>0::real. eventually (λx::?'c::type. ∀ n::?'b::type. P n ⟶ vector_norm (vector_sub (vector_sub (f n x) (g n x)) (vector_sub (l n) (m n))) < e) net)

thm UNIFORM_LIM_BILINEAR:

∀ (net::?'e::type net) (P::?'d::type ⇒ bool) (h::(real, ?'c::type) cart ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f::?'d::type ⇒ ?'e::type ⇒ (real, ?'c::type) cart) (g::?'d::type ⇒ ?'e::type ⇒ (real, ?'b::type) cart) (l::?'d::type ⇒ (real, ?'c::type) cart) (m::?'d::type ⇒ (real, ?'b::type) cart) (b1::real) b2::real. bilinear h ∧ eventually (λx::?'e::type. ∀ n::?'d::type. P n ⟶ vector_norm (l n) ≤ b1) net ∧ eventually (λx::?'e::type. ∀ n::?'d::type. P n ⟶ vector_norm (m n) ≤ b2) net ∧ (∀ e>0::real. eventually (λx::?'e::type. ∀ n::?'d::type. P n ⟶ vector_norm (vector_sub (f n x) (l n)) < e) net) ∧ (∀ e>0::real. eventually (λx::?'e::type. ∀ n::?'d::type. P n ⟶ vector_norm (vector_sub (g n x) (m n)) < e) net) ⟶ (∀ e>0::real. eventually (λx::?'e::type. ∀ n::?'d::type. P n ⟶ vector_norm (vector_sub (h (f n x) (g n x)) (h (l n) (m n))) < e) net)

thm LIM_BILINEAR:

∀ (net::?'d::type net) (h::(real, ?'c::type) cart ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f::?'d::type ⇒ (real, ?'c::type) cart) (g::?'d::type ⇒ (real, ?'b::type) cart) (l::(real, ?'c::type) cart) m::(real, ?'b::type) cart. --> f l net ∧ --> g m net ∧ bilinear h ⟶ --> (λx::?'d::type. h (f x) (g x)) (h l m) net

thm LIM_WITHIN_ID:

∀ (a::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. --> (λx::(real, ?'a::type) cart. x) a (within (at a) s)

thm LIM_AT_ID:

∀ a::(real, ?'a::type) cart. --> (λx::(real, ?'a::type) cart. x) a (at a)

thm LIM_AT_ZERO:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) a::(real, ?'b::type) cart. --> f l (at a) = --> (λx::(real, ?'b::type) cart. f (vector_add a x)) l (at (vec (0::nat)))

thm DEF_netlimit:

netlimit = (λ_198081::?'a::type net. SOME a::?'a::type. ∀ x::?'a::type. ¬ netord _198081 x a)

thm netlimit:

∀ net::?'a::type net. netlimit net = (SOME a::?'a::type. ∀ x::?'a::type. ¬ netord net x a)

thm NETLIMIT_WITHIN:

∀ (a::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. ¬ trivial_limit (within (at a) s) ⟶ netlimit (within (at a) s) = a

thm NETLIMIT_AT:

$\forall\, a{::}(real,\ ?'a{::}type)\ cart.\ netlimit\ (at\ a) = a$

thm LIM_TRANSFORM:

$\forall\, (net{::}?'b{::}type\ net)\ (f{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ l{::}(real,\ ?'a{::}type)\ cart.\ {-}{-}{>}\ (\lambda x{::}?'b{::}type.\ vector\_sub\ (f\ x)\ (g\ x))\ (vec\ (0{::}nat))\ net \wedge {-}{-}{>}\ f\ l\ net \longrightarrow {-}{-}{>}\ g\ l\ net$

thm LIM_TRANSFORM_EVENTUALLY:

$\forall\, (net{::}?'b{::}type\ net)\ (f{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ l{::}(real,\ ?'a{::}type)\ cart.\ eventually\ (\lambda x{::}?'b{::}type.\ f\ x = g\ x)\ net \wedge {-}{-}{>}\ f\ l\ net \longrightarrow {-}{-}{>}\ g\ l\ net$

thm LIM_TRANSFORM_WITHIN:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ d{::}real.\ (0{::}real) < d \wedge (\forall\, x'{::}(real,\ ?'b{::}type)\ cart.\ IN\ x'\ s \wedge (0{::}real) < distance\ (x',\ x) \wedge distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge {-}{-}{>}\ f\ (?l{::}(real,\ ?'a{::}type)\ cart)\ (within\ (at\ x)\ s) \longrightarrow {-}{-}{>}\ g\ ?l\ (within\ (at\ x)\ s)$

thm LIM_TRANSFORM_AT:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ d{::}real.\ (0{::}real) < d \wedge (\forall\, x'{::}(real,\ ?'b{::}type)\ cart.\ (0{::}real) < distance\ (x',\ x) \wedge distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge {-}{-}{>}\ f\ (?l{::}(real,\ ?'a{::}type)\ cart)\ (at\ x) \longrightarrow {-}{-}{>}\ g\ ?l\ (at\ x)$

thm LIM_TRANSFORM_EQ:

$\forall\, (net{::}?'b{::}type\ net)\ (f{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ l{::}(real,\ ?'a{::}type)\ cart.\ {-}{-}{>}\ (\lambda x{::}?'b{::}type.\ vector\_sub\ (f\ x)\ (g\ x))\ (vec\ (0{::}nat))\ net \longrightarrow {-}{-}{>}\ f\ l\ net = {-}{-}{>}\ g\ l\ net$

thm LIM_TRANSFORM_WITHIN_SET:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (a{::}(real,\ ?'b{::}type)\ cart)\ (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ t{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ eventually\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ IN\ x\ s = IN\ x\ t)\ (at\ a) \longrightarrow {-}{-}{>}\ f\ (?l{::}(real,\ ?'a{::}type)\ cart)\ (within\ (at\ a)\ s) = {-}{-}{>}\ f\ ?l\ (within\ (at\ a)\ t)$

thm LIM_TRANSFORM_AWAY_WITHIN:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (a{::}(real,\ ?'b{::}type)\ cart)\ (b{::}(real,\ ?'b{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ a \neq b \wedge (\forall\, x{::}(real,\ ?'b{::}type)\ cart.\ IN\ x\ s \wedge x \neq a \wedge x \neq b \longrightarrow f\ x = g\ x) \wedge {-}{-}{>}\ f\ (?l{::}(real,\ ?'a{::}type)\ cart)\ (within\ (at\ a)\ s) \longrightarrow {-}{-}{>}\ g\ ?l\ (within\ (at\ a)\ s)$

thm LIM_TRANSFORM_AWAY_AT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. a ≠ b ∧ (∀ x::(real, ?'b::type) cart. x ≠ a ∧ x ≠ b ⟶ f x = g x) ∧ --> f (?l::(real, ?'a::type) cart) (at a) ⟶ --> g ?l (at a)

thm LIM_TRANSFORM_WITHIN_OPEN:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'b::type) cart. HOL_Light_Import.open s ∧ IN a s ∧ (∀ x::(real, ?'b::type) cart. IN x s ∧ x ≠ a ⟶ f x = g x) ∧ --> f (?l::(real, ?'a::type) cart) (at a) ⟶ --> g ?l (at a)

thm LIM_CASES_FINITE_SEQUENTIALLY:

∀ (f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, ?'a::type) cart) l::(real, ?'a::type) cart. FINITE (GSPEC (λGEN%PVAR%495::nat. ∃ n::nat. SETSPEC GEN%PVAR%495 ((?P::nat ⇒ bool) n) n)) ⟶ --> (λn::nat. if ?P n then f n else g n) l sequentially = --> g l sequentially

thm LIM_CASES_COFINITE_SEQUENTIALLY:

∀ (f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, ?'a::type) cart) l::(real, ?'a::type) cart. FINITE (GSPEC (λGEN%PVAR%496::nat. ∃ n::nat. SETSPEC GEN%PVAR%496 (¬ (?P::nat ⇒ bool) n) n)) ⟶ --> (λn::nat. if ?P n then f n else g n) l sequentially = --> f l sequentially

thm LIM_CASES_SEQUENTIALLY:

∀ (f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, ?'a::type) cart) (l::(real, ?'a::type) cart) m::nat. --> (λn::nat. if m ≤ n then f n else g n) l sequentially = --> f l sequentially ∧ --> (λn::nat. if m < n then f n else g n) l sequentially = --> f l sequentially ∧ --> (λn::nat. if n ≤ m then f n else g n) l sequentially = --> g l sequentially ∧ --> (λn::nat. if n < m then f n else g n) l sequentially = --> g l sequentially

thm LIM_CONG_WITHIN:

(∀ x::(real, ?'b::type) cart. x ≠ (?a::(real, ?'b::type) cart) ⟶ (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x = (?g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x) ⟶ --> ?f (?l::(real, ?'a::type) cart) (within (at ?a) (?s::(real, ?'b::type) cart ⇒ bool)) = --> ?g ?l (within (at ?a) ?s)

thm LIM_CONG_AT:

(∀ x::(real, ?'b::type) cart. x ≠ (?a::(real, ?'b::type) cart) ⟶ (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x = (?g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x) ⟶ --> ?f (?l::(real, ?'a::type) cart) (at ?a) = --> ?g ?l (at ?a)

thm CLOSURE_SEQUENTIAL:

∀ (s::(real, ?'a::type) cart ⇒ bool) l::(real, ?'a::type) cart. IN l (closure s) = (∃ x::nat ⇒ (real, ?'a::type) cart. (∀ n::nat. IN (x n) s) ∧ --> x l sequentially)

thm CLOSED_SEQUENTIAL_LIMITS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s = (\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ (\forall n::nat.\ IN\ (x\ n)\ s) \wedge --> x\ l\ sequentially \longrightarrow IN\ l\ s)$

thm CLOSURE_APPROACHABLE:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (closure\ s) = (\forall e>0::real.\ \exists y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \wedge distance\ (y,\ x) < e)$

thm CLOSED_APPROACHABLE:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow (\forall e>0::real.\ \exists y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \wedge distance\ (y,\ x) < e) = IN\ x\ s$

thm SEQ_OFFSET:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ k::nat.\ --> f\ l\ sequentially \longrightarrow --> (\lambda i::nat.\ f\ (i + k))\ l\ sequentially$

thm SEQ_OFFSET_NEG:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ k::nat.\ --> f\ l\ sequentially \longrightarrow --> (\lambda i::nat.\ f\ (i - k))\ l\ sequentially$

thm SEQ_OFFSET_REV:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ k::nat.\ --> (\lambda i::nat.\ f\ (i + k))\ l\ sequentially \longrightarrow --> f\ l\ sequentially$

thm SEQ_HARMONIC:

$--> (\lambda n::nat.\ lift\ (inverse\_class.inverse\ (real\_of\_nat\ n)))\ (vec\ (0::nat))\ sequentially$

thm CLOSED_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ HOL\_Light\_Import.closed\ (cball\ (x,\ e))$

thm OPEN_CONTAINS_CBALL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists e>0::real.\ SUBSET\ (cball\ (x,\ e))\ s))$

thm OPEN_CONTAINS_CBALL_EQ:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s = (\exists e>0::real.\ SUBSET\ (cball\ (x,\ e))\ s))$

thm IN_INTERIOR_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (interior\ s) = (\exists e>0::real.\ SUBSET\ (cball\ (x,\ e))\ s)$

thm LIMPT_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ e::real.\ limit\_point\_of\ y\ (ball\ (x,\ e)) = ((0::real) < e \wedge IN\ y\ (cball\ (x,\ e)))$

thm CLOSURE_BALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \longrightarrow closure\ (ball\ (x,\ e)) = cball\ (x,\ e)$

thm INTERIOR_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ interior\ (cball\ (x,\ e)) = ball\ (x,\ e)$

thm FRONTIER_BALL:

$\forall (a::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \longrightarrow frontier\ (ball\ (a,\ e)) = GSPEC\ (\lambda GEN\%PVAR\%497::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%497\ (distance\ (a,\ x) = e)\ x)$

thm FRONTIER_CBALL:

$\forall (a::(real,\ ?'a::type)\ cart)\ e::real.\ frontier\ (cball\ (a,\ e)) = GSPEC\ (\lambda GEN\%PVAR\%498::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%498\ (distance\ (a,\ x) = e)\ x)$

thm CBALL_EQ_EMPTY:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ (cball\ (x,\ e) = EMPTY) = (e < (0::real))$

thm CBALL_EMPTY:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ e < (0::real) \longrightarrow cball\ (x,\ e) = EMPTY$

thm CBALL_EQ_SING:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ (cball\ (x,\ e) = INSERT\ x\ EMPTY) = (e = (0::real))$

thm CBALL_SING:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ e = (0::real) \longrightarrow cball\ (x,\ e) = INSERT\ x\ EMPTY$

thm EVENTUALLY_WITHIN_INTERIOR:

$\forall (p::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ (interior\ s) \longrightarrow eventually\ p\ (within\ (at\ x)\ s) = eventually\ p\ (at\ x)$

thm LIM_WITHIN_INTERIOR:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ IN\ x\ (interior\ s) \longrightarrow --> f\ l\ (within\ (at\ x)\ s) = --> f\ l\ (at\ x)$

thm NETLIMIT_WITHIN_INTERIOR:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ (interior\ s) \longrightarrow netlimit\ (within\ (at\ x)\ s) = x$

thm DEF_bounded:

$bounded = (\lambda\_200749::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, a::real.\ \forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_200749 \longrightarrow vector\_norm\ x \le a)$

thm bounded:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s = (\exists\, a::real.\ \forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ x \le a)$

thm BOUNDED_EMPTY:

$bounded\ EMPTY$

thm BOUNDED_SUBSET:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ t \wedge SUBSET\ s\ t \longrightarrow bounded\ s$

thm BOUNDED_INTERIOR:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow bounded\ (interior\ s)$

thm BOUNDED_CLOSURE:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow bounded\ (closure\ s)$

thm BOUNDED_CLOSURE_EQ:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ (closure\ s) = bounded\ s$

thm BOUNDED_CBALL:

$\forall\, (x::(real,\ ?'a::type)\ cart)\ e::real.\ bounded\ (cball\ (x,\ e))$

thm BOUNDED_BALL:

$\forall\, (x::(real,\ ?'a::type)\ cart)\ e::real.\ bounded\ (ball\ (x,\ e))$

thm FINITE_IMP_BOUNDED:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow bounded\ s$

thm BOUNDED_UNION:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ (HOL\_Light\_Import.UNION\ s\ t) = (bounded\ s \wedge bounded\ t)$

thm BOUNDED_UNIONS:

$\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow bounded\ s) \longrightarrow bounded\ (UNIONS\ f)$

thm BOUNDED_POS:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s = (\exists\, b>0::real.\ \forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ x \le b)$

thm BOUNDED_POS_LT:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* = ($\exists b>0$::*real*. $\forall x$::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *vector_norm x* < *b*)

thm BOUNDED_INTER:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∨ *bounded t* ⟶ *bounded* (*HOL_Light_Import.INTER s t*)

thm BOUNDED_DIFF:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *bounded* (*DIFF s t*)

thm BOUNDED_INSERT:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded* (*INSERT x s*) = *bounded s*

thm BOUNDED_SING:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. *bounded* (*INSERT a EMPTY*)

thm BOUNDED_INTERS:

$\forall$ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. ($\exists$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *bounded s*) ⟶ *bounded* (*INTERS f*)

thm NOT_BOUNDED_UNIV:

¬ *bounded HOL_Light_Import.UNIV*

thm COBOUNDED_IMP_UNBOUNDED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded* (*DIFF HOL_Light_Import.UNIV s*) ⟶ ¬ *bounded s*

thm BOUNDED_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *linear f* ⟶ *bounded* (*IMAGE f s*)

thm BOUNDED_LINEAR_IMAGE_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ∧ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *bounded* (*IMAGE f s*) = *bounded s*

thm BOUNDED_SCALING:

$\forall$ (*c*::*real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *bounded* (*IMAGE* (*% c*) *s*)

thm BOUNDED_NEGATIONS:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *bounded* (*IMAGE vector_neg s*)

thm BOUNDED_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow bounded\ (IMAGE\ (vector\_add\ a)\ s)$

thm BOUNDED_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ (IMAGE\ (vector\_add\ a)\ s) = bounded\ s$

thm BOUNDED_DIFFS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge bounded\ t \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%499::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%499\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm BOUNDED_SUMS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge bounded\ t \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%500::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%500\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm BOUNDED_SUMS_IMAGE:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ t::?'b::type \Rightarrow bool.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%501::(real, ?'a::type)\ cart.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%501\ (IN\ x\ t)\ (f\ x))) \wedge bounded\ (GSPEC\ (\lambda GEN\%PVAR\%502::(real, ?'a::type)\ cart.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%502\ (IN\ x\ t)\ (g\ x))) \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%503::(real, ?'a::type)\ cart.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%503\ (IN\ x\ t)\ (vector\_add\ (f\ x)\ (g\ x))))$

thm BOUNDED_SUMS_IMAGES:

$\forall (f::?'c::type \Rightarrow ?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (t::?'c::type \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'b::type.\ IN\ a\ s \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%504::(real, ?'a::type)\ cart.\ \exists x::?'c::type.\ SETSPEC\ GEN\%PVAR\%504\ (IN\ x\ t)\ (f\ x\ a)))) \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%505::(real, ?'a::type)\ cart.\ \exists x::?'c::type.\ SETSPEC\ GEN\%PVAR\%505\ (IN\ x\ t)\ (vsum\ s\ (f\ x))))$

thm BOUNDED_SUBSET_BALL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ bounded\ s \longrightarrow (\exists r>0::real.\ SUBSET\ s\ (ball\ (x, r)))$

thm BOUNDED_SUBSET_CBALL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ bounded\ s \longrightarrow (\exists r>0::real.\ SUBSET\ s\ (cball\ (x, r)))$

thm UNBOUNDED_INTER_COBOUNDED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ bounded\ s \wedge bounded\ (DIFF\ HOL\_Light\_Import.UNIV\ t) \longrightarrow HOL\_Light\_Import.INTER\ s\ t \neq EMPTY$

thm COBOUNDED_INTER_UNBOUNDED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ (DIFF\ HOL\_Light\_Import.UNIV\ s) \land \neg\ bounded\ t \longrightarrow HOL\_Light\_Import.INTER\ s\ t \neq EMPTY$

thm SUBSPACE_BOUNDED_EQ_TRIVIAL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow bounded\ s = (s = INSERT\ (vec\ (0::nat))\ EMPTY)$

thm BOUNDED_COMPONENTWISE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s = (\forall i::nat.\ (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow bounded\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (\$\ x\ i))\ s))$

thm BOUNDED_LIFT:

$\forall s::real \Rightarrow bool.\ bounded\ (IMAGE\ lift\ s) = (\exists a::real.\ \forall x::real.\ IN\ x\ s \longrightarrow |x| \leq a)$

thm BOUNDED_HAS_SUP:

$\forall s::real \Rightarrow bool.\ bounded\ (IMAGE\ lift\ s) \land s \neq EMPTY \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow x \leq HOL\_Light\_Import.sup\ s) \land (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow x \leq b) \longrightarrow HOL\_Light\_Import.sup\ s \leq b)$

thm SUP_INSERT:

$\forall (x::real)\ s::real \Rightarrow bool.\ bounded\ (IMAGE\ lift\ s) \longrightarrow HOL\_Light\_Import.sup\ (INSERT\ x\ s) = (if\ s = EMPTY\ then\ x\ else\ max\ x\ (HOL\_Light\_Import.sup\ s))$

thm BOUNDED_HAS_INF:

$\forall s::real \Rightarrow bool.\ bounded\ (IMAGE\ lift\ s) \land s \neq EMPTY \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.inf\ s \leq x) \land (\forall b::real.\ (\forall x::real.\ IN\ x\ s \longrightarrow b \leq x) \longrightarrow b \leq HOL\_Light\_Import.inf\ s)$

thm INF_INSERT:

$\forall (x::real)\ s::real \Rightarrow bool.\ bounded\ (IMAGE\ lift\ s) \longrightarrow HOL\_Light\_Import.inf\ (INSERT\ x\ s) = (if\ s = EMPTY\ then\ x\ else\ min\ x\ (HOL\_Light\_Import.inf\ s))$

thm SUBSET_BALLS:

$(\forall (a::(real,\ ?'a::type)\ cart)\ (a'::(real,\ ?'a::type)\ cart)\ (r::real)\ r'::real.\ SUBSET\ (ball\ (a,\ r))\ (ball\ (a',\ r')) = (distance\ (a,\ a') + r \leq r' \lor r \leq (0::real))) \land (\forall (a::(real,\ ?'a::type)\ cart)\ (a'::(real,\ ?'a::type)\ cart)\ (r::real)\ r'::real.\ SUBSET\ (ball\ (a,\ r))\ (cball\ (a',\ r')) = (distance\ (a,\ a') + r \leq r' \lor r \leq (0::real))) \land (\forall (a::(real,\ ?'a::type)\ cart)\ (a'::(real,\ ?'a::type)\ cart)\ (r::real)\ r'::real.\ SUBSET\ (cball\ (a,\ r))\ (ball\ (a',\ r')) = (distance\ (a,\ a') + r < r' \lor r < (0::real))) \land (\forall (a::(real,\ ?'a::type)\ cart)\ (a'::(real,\ ?'a::type)\ cart)\ (r::real)\ r'::real.\ SUBSET\ (cball\ (a,\ r))\ (cball\ (a',\ r')) = (distance\ (a,\ a') + r \leq r' \lor r < (0::real)))$

thm DEF_compact:

$compact = (\lambda\_201693::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall f::nat \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall n::nat.\ IN\ (f\ n)\ \_201693) \longrightarrow (\exists (l::(real,\ ?'a::type)\ cart)\ r::nat \Rightarrow nat.\ IN\ l\ \_201693 \wedge (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge --> (f \circ r)\ l\ sequentially))$

thm compact:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s = (\forall f::nat \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall n::nat.\ IN\ (f\ n)\ s) \longrightarrow (\exists (l::(real,\ ?'a::type)\ cart)\ r::nat \Rightarrow nat.\ IN\ l\ s \wedge (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge --> (f \circ r)\ l\ sequentially))$

thm MONOTONE_BIGGER:

$\forall r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \longrightarrow (\forall n::nat.\ n \leq r\ n)$

thm LIM_SUBSEQUENCE:

$\forall (s::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (r::nat \Rightarrow nat)\ l::(real,\ ?'a::type)\ cart.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge --> s\ l\ sequentially \longrightarrow --> (s \circ r)\ l\ sequentially$

thm MONOTONE_SUBSEQUENCE:

$\forall s::nat \Rightarrow real.\ \exists r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge ((\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow s\ (r\ m) \leq s\ (r\ n)) \vee (\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow s\ (r\ n) \leq s\ (r\ m)))$

thm CONVERGENT_BOUNDED_INCREASING:

$\forall (s::nat \Rightarrow real)\ b::real.\ (\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow s\ m \leq s\ n) \wedge (\forall n::nat.\ |s\ n| \leq b) \longrightarrow (\exists l::real.\ \forall e{>}0::real.\ \exists N::nat.\ \forall n{\geq}N.\ |s\ n - l| < e)$

thm CONVERGENT_BOUNDED_MONOTONE:

$\forall (s::nat \Rightarrow real)\ b::real.\ (\forall n::nat.\ |s\ n| \leq b) \wedge ((\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow s\ m \leq s\ n) \vee (\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow s\ n \leq s\ m)) \longrightarrow (\exists l::real.\ \forall e{>}0::real.\ \exists N::nat.\ \forall n{\geq}N.\ |s\ n - l| < e)$

thm COMPACT_REAL_LEMMA:

$\forall (s::nat \Rightarrow real)\ b::real.\ (\forall n::nat.\ |s\ n| \leq b) \longrightarrow (\exists (l::real)\ r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge (\forall e{>}0::real.\ \exists N::nat.\ \forall n{\geq}N.\ |s\ (r\ n) - l| < e))$

thm COMPACT_LEMMA:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge (\forall n::nat.\ IN\ ((?x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ n)\ s) \longrightarrow (\forall d{\leq}dimindex\ HOL\_Light\_Import.UNIV.\ \exists (l::(real,\ ?'a::type)\ cart)\ r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge (\forall e{>}0::real.\ \exists N::nat.\ \forall (n::nat)\ i::nat.\ (1::nat) \leq i \wedge i \leq d \longrightarrow N \leq n \longrightarrow |\$\ (?x\ (r\ n))\ i - \$\ l\ i| < e))$

thm BOUNDED_CLOSED_IMP_COMPACT:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *HOL_Light_Import.closed s* ⟶ *compact s*

thm DEF_cauchy:

cauchy = (λ*_203908*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. ∀ *e*>*0*::*real*. ∃ *N*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. *N* ≤ *m* ∧ *N* ≤ *n* ⟶ *distance* (*_203908 m*, *_203908 n*) < *e*)

thm cauchy:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. *cauchy s* = (∀ *e*>*0*::*real*. ∃ *N*::*nat*. ∀ (*m*::*nat*) *n*::*nat*. *N* ≤ *m* ∧ *N* ≤ *n* ⟶ *distance* (*s m*, *s n*) < *e*)

thm DEF_complete:

complete = (λ*_203913*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∀ *f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. (∀ *n*::*nat*. *IN* (*f n*) *_203913*) ∧ *cauchy f* ⟶ (∃ *l*::(*real*, *?'a*::*type*) *cart*. *IN l _203913* ∧ −−> *f l sequentially*))

thm complete:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *complete s* = (∀ *f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. (∀ *n*::*nat*. *IN* (*f n*) *s*) ∧ *cauchy f* ⟶ (∃ *l*::(*real*, *?'a*::*type*) *cart*. *IN l s* ∧ −−> *f l sequentially*))

thm CAUCHY:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. *cauchy s* = (∀ *e*>*0*::*real*. ∃ *N*::*nat*. ∀ *n*≥*N*. *distance* (*s n*, *s N*) < *e*)

thm CONVERGENT_IMP_CAUCHY:

∀ (*s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. −−> *s l sequentially* ⟶ *cauchy s*

thm CAUCHY_IMP_BOUNDED:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. *cauchy s* ⟶ *bounded* (*GSPEC* (λ*GEN%PVAR%506*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%506* (∃ *n*::*nat*. *y* = *s n*) *y*))

thm COMPACT_IMP_COMPLETE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ *complete s*

thm COMPLETE_UNIV:

complete HOL_Light_Import.UNIV

thm COMPLETE_EQ_CLOSED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *complete s* = *HOL_Light_Import.closed s*

thm CONVERGENT_EQ_CAUCHY:

$\forall s$::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. (∃ *l*::(*real*, *?'a*::*type*) *cart*. −−> *s l sequentially*) = *cauchy s*

thm CONVERGENT_IMP_BOUNDED:

∀ (*s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. −−> *s l sequentially* ⟶ *bounded* (*IMAGE s HOL_Light_Import.UNIV*)

thm COMPACT_IMP_TOTALLY_BOUNDED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ (∀ *e*>*0*::*real*. ∃ *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE k* ∧ *SUBSET k s* ∧ *SUBSET s* (*UNIONS* (*IMAGE* (λ*x*::(*real*, *?'a*::*type*) *cart*. *ball* (*x*, *e*)) *k*)))

thm HEINE_BOREL_LEMMA:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ (∀ *t*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET s* (*UNIONS t*) ∧ (∀ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN b t* ⟶ *HOL_Light_Import.open b*) ⟶ (∃ *e*>*0*::*real*. ∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (∃ *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN b t* ∧ *SUBSET* (*ball* (*x*, *e*)) *b*)))

thm COMPACT_IMP_HEINE_BOREL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ (∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t f* ⟶ *HOL_Light_Import.open t*) ∧ *SUBSET s* (*UNIONS f*) ⟶ (∃ *f'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET f' f* ∧ *FINITE f'* ∧ *SUBSET s* (*UNIONS f'*)))

thm HEINE_BOREL_IMP_BOLZANO_WEIERSTRASS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t f* ⟶ *HOL_Light_Import.open t*) ∧ *SUBSET s* (*UNIONS f*) ⟶ (∃ *f'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET f' f* ∧ *FINITE f'* ∧ *SUBSET s* (*UNIONS f'*))) ⟶ (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *INFINITE t* ∧ *SUBSET t s* ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *limit_point_of x t*))

thm BOLZANO_WEIERSTRASS_IMP_BOUNDED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *INFINITE t* ∧ *SUBSET t s* ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *limit_point_of x t*)) ⟶ *bounded s*

thm SEQUENCE_INFINITE_LEMMA:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. (∀ *n*::*nat*. *f n* ≠ *l*) ∧ −−> *f l sequentially* ⟶ *INFINITE* (*GSPEC* (λ*GEN%PVAR%512*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%512* (∃ *n*::*nat*. *y* = *f n*) *y*))

thm SEQUENCE_UNIQUE_LIMPT:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. (∀ *n*::*nat*. *f n* ≠ *l*) ∧ −−> *f l sequentially* ⟶ (∀ *l'*::(*real*, *?'a*::*type*) *cart*. *limit_point_of l'* (*GSPEC*

$(\lambda GEN\%PVAR\%514::(real,\ ?'a::type)\ cart.\ \exists\, y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%514\ (\exists\, n::nat.\ y = f\ n)\ y)) \longrightarrow l' = l)$

thm BOLZANO_WEIERSTRASS_IMP_CLOSED:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ INFINITE\ t \wedge SUBSET\ t\ s \longrightarrow (\exists\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge limit\_point\_of\ x\ t)) \longrightarrow HOL\_Light\_Import.closed\ s$

thm COMPACT_EQ_HEINE_BOREL:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s = (\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ f \longrightarrow HOL\_Light\_Import.open\ t) \wedge SUBSET\ s\ (UNIONS\ f) \longrightarrow (\exists\, f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SUBSET\ f'\ f \wedge FINITE\ f' \wedge SUBSET\ s\ (UNIONS\ f')))$

thm COMPACT_EQ_BOLZANO_WEIERSTRASS:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s = (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ INFINITE\ t \wedge SUBSET\ t\ s \longrightarrow (\exists\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge limit\_point\_of\ x\ t))$

thm COMPACT_EQ_BOUNDED_CLOSED:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s = (bounded\ s \wedge HOL\_Light\_Import.closed\ s)$

thm COMPACT_IMP_BOUNDED:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow bounded\ s$

thm COMPACT_IMP_CLOSED:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow HOL\_Light\_Import.closed\ s$

thm COMPACT_EQ_HEINE_BOREL_SUBTOPOLOGY:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s = (\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ f \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ t) \wedge SUBSET\ s\ (UNIONS\ f) \longrightarrow (\exists\, f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SUBSET\ f'\ f \wedge FINITE\ f' \wedge SUBSET\ s\ (UNIONS\ f')))$

thm COMPACT_CLOSURE:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ (closure\ s) = bounded\ s$

thm BOLZANO_WEIERSTRASS_CONTRAPOS:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge SUBSET\ t\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow \neg\ limit\_point\_of\ x\ t) \longrightarrow FINITE\ t$

thm DISCRETE_BOUNDED_IMP_FINITE:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ (0::real) < e \wedge (\forall\, (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge vector\_norm\ (vector\_sub\ y\ x) < e \longrightarrow y = x) \wedge bounded\ s \longrightarrow FINITE\ s$

thm BOLZANO_WEIERSTRASS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge INFINITE\ s \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ limit\_point\_of\ x\ s)$

thm COMPACT_EMPTY:

$compact\ EMPTY$

thm COMPACT_UNION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge compact\ t \longrightarrow compact\ (HOL\_Light\_Import.UNION\ s\ t)$

thm COMPACT_INTER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge compact\ t \longrightarrow compact\ (HOL\_Light\_Import.INTER\ s\ t)$

thm COMPACT_INTER_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow compact\ (HOL\_Light\_Import.INTER\ s\ t)$

thm CLOSED_INTER_COMPACT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge compact\ t \longrightarrow compact\ (HOL\_Light\_Import.INTER\ s\ t)$

thm COMPACT_INTERS:

$\forall f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow compact\ s) \wedge f \neq EMPTY \longrightarrow compact\ (INTERS\ f)$

thm FINITE_IMP_CLOSED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow HOL\_Light\_Import.closed\ s$

thm FINITE_IMP_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow compact\ s$

thm COMPACT_SING:

$\forall a::(real,\ ?'a::type)\ cart.\ compact\ (INSERT\ a\ EMPTY)$

thm COMPACT_INSERT:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow compact\ (INSERT\ a\ s)$

thm CLOSED_SING:

$\forall a::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ (INSERT\ a\ EMPTY)$

thm CLOSED_INSERT:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (INSERT\ a\ s)$

thm COMPACT_CBALL:

$\forall (x::(real,\ ?'a::type)\ cart)\ e::real.\ compact\ (cball\ (x,\ e))$

thm COMPACT_FRONTIER_BOUNDED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow compact\ (frontier\ s)$

thm COMPACT_FRONTIER:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow compact\ (frontier\ s)$

thm FRONTIER_SUBSET_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow SUBSET\ (frontier\ s)\ s$

thm OPEN_DELETE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (DELETE\ s\ x)$

thm CLOSED_INTERS_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s = (\forall e::real.\ compact\ (HOL\_Light\_Import.INTER\ (cball\ (vec\ (0::nat),\ e))\ s))$

thm COMPACT_UNIONS:

$\forall s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ s \longrightarrow compact\ t) \longrightarrow compact\ (UNIONS\ s)$

thm COMPACT_DIFF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow compact\ (DIFF\ s\ t)$

thm COMPACT_SPHERE:

$\forall (a::(real,\ ?'a::type)\ cart)\ r::real.\ compact\ (GSPEC\ (\lambda GEN\%PVAR\%518::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%518\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x))$

thm COMPACT_SPHERE_0:

$\forall a::real.\ compact\ (GSPEC\ (\lambda GEN\%PVAR\%519::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%519\ (vector\_norm\ x = a)\ x))$

thm COMPACT_IMP_FIP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ compact\ s \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ f \longrightarrow HOL\_Light\_Import.closed\ t) \wedge (\forall f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f' \wedge SUBSET\ f'\ f \longrightarrow HOL\_Light\_Import.INTER\ s\ (INTERS\ f') \neq EMPTY) \longrightarrow HOL\_Light\_Import.INTER\ s\ (INTERS\ f) \neq EMPTY$

thm CLOSED_FIP:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. ($\forall t$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t f* ⟶ *HOL_Light_Import.closed t*) ∧ (∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t f* ∧ *bounded t*) ∧ (∀ *f'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE f'* ∧ *SUBSET f' f* ⟶ *INTERS f'* ≠ *EMPTY*) ⟶ *INTERS f* ≠ *EMPTY*

thm COMPACT_FIP:

∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t f* ⟶ *compact t*) ∧ (∀ *f'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE f'* ∧ *SUBSET f' f* ⟶ *INTERS f'* ≠ *EMPTY*) ⟶ *INTERS f* ≠ *EMPTY*

thm BOUNDED_CLOSED_NEST:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *HOL_Light_Import.closed* (*s n*)) ∧ (∀ *n*::*nat*. *s n* ≠ *EMPTY*) ∧ (∀ (*m*::*nat*) *n*::*nat*. *m* ≤ *n* ⟶ *SUBSET* (*s n*) (*s m*)) ∧ *bounded* (*s* (*0*::*nat*)) ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. ∀ *n*::*nat*. *IN a* (*s n*))

thm DECREASING_CLOSED_NEST:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *HOL_Light_Import.closed* (*s n*)) ∧ (∀ *n*::*nat*. *s n* ≠ *EMPTY*) ∧ (∀ (*m*::*nat*) *n*::*nat*. *m* ≤ *n* ⟶ *SUBSET* (*s n*) (*s m*)) ∧ (∀ *e*>*0*::*real*. ∃ *n*::*nat*. ∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x* (*s n*) ∧ *IN y* (*s n*) ⟶ *distance* (*x*, *y*) < *e*) ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. ∀ *n*::*nat*. *IN a* (*s n*))

thm DECREASING_CLOSED_NEST_SING:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *HOL_Light_Import.closed* (*s n*)) ∧ (∀ *n*::*nat*. *s n* ≠ *EMPTY*) ∧ (∀ (*m*::*nat*) *n*::*nat*. *m* ≤ *n* ⟶ *SUBSET* (*s n*) (*s m*)) ∧ (∀ *e*>*0*::*real*. ∃ *n*::*nat*. ∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x* (*s n*) ∧ *IN y* (*s n*) ⟶ *distance* (*x*, *y*) < *e*) ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *INTERS* (*GSPEC* (λ*GEN%PVAR%520*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%520* (∃ *n*::*nat*. *t* = *s n*) *t*)) = *INSERT a EMPTY*)

thm BOUNDED_CLOSED_CHAIN:

∀ (*f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *b*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *HOL_Light_Import.closed s* ∧ *s* ≠ *EMPTY*) ∧ (∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *IN t f* ⟶ *SUBSET s t* ∨ *SUBSET t s*) ∧ *IN b f* ∧ *bounded b* ⟶ *INTERS f* ≠ *EMPTY*

thm COMPACT_CHAIN:

∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *compact s* ∧ *s* ≠ *EMPTY*) ∧ (∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *IN t f* ⟶ *SUBSET s t* ∨ *SUBSET t s*) ⟶ *INTERS f* ≠ *EMPTY*

thm COMPACT_NEST:

$\forall s::nat \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool.\ (\forall n::nat.\ compact\ (s\ n) \wedge s\ n \neq EMPTY) \wedge (\forall (m::nat)\ n::nat.\ m \leq n \longrightarrow SUBSET\ (s\ n)\ (s\ m)) \longrightarrow INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%521::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%521\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))) \neq EMPTY$

thm UNIFORMLY_CONVERGENT_EQ_CAUCHY:

$\forall (P::?'b::type \Rightarrow bool)\ s::nat \Rightarrow ?'b::type \Rightarrow (real, ?'a::type)\ cart.\ (\exists l::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ \forall e>0::real.\ \exists N::nat.\ \forall (n::nat)\ x::?'b::type.\ N \leq n \wedge P\ x \longrightarrow distance\ (s\ n\ x,\ l\ x) < e) = (\forall e>0::real.\ \exists N::nat.\ \forall (m::nat)\ (n::nat)\ x::?'b::type.\ N \leq m \wedge N \leq n \wedge P\ x \longrightarrow distance\ (s\ m\ x,\ s\ n\ x) < e)$

thm UNIFORMLY_CAUCHY_IMP_UNIFORMLY_CONVERGENT:

$\forall (P::?'b::type \Rightarrow bool)\ (s::nat \Rightarrow ?'b::type \Rightarrow (real, ?'a::type)\ cart)\ l::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ (\forall e>0::real.\ \exists N::nat.\ \forall (m::nat)\ (n::nat)\ x::?'b::type.\ N \leq m \wedge N \leq n \wedge P\ x \longrightarrow distance\ (s\ m\ x,\ s\ n\ x) < e) \wedge (\forall x::?'b::type.\ P\ x \longrightarrow (\forall e>0::real.\ \exists N::nat.\ \forall n \geq N.\ distance\ (s\ n\ x,\ l\ x) < e)) \longrightarrow (\forall e>0::real.\ \exists N::nat.\ \forall (n::nat)\ x::?'b::type.\ N \leq n \wedge P\ x \longrightarrow distance\ (s\ n\ x,\ l\ x) < e)$

thm DEF_continuous:

$continuous = (\lambda(\_209354::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ \_209355::?'b::type\ net.\ -->\ \_209354\ (\_209354\ (netlimit\ \_209355))\ \_209355)$

thm continuous:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ net::?'b::type\ net.\ continuous\ f\ net = -->\ f\ (f\ (netlimit\ net))\ net$

thm CONTINUOUS_TRIVIAL_LIMIT:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ net::?'b::type\ net.\ trivial\_limit\ net \longrightarrow continuous\ f\ net$

thm CONTINUOUS_WITHIN:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ continuous\ f\ (within\ (at\ x)\ (?s::(real, ?'b::type)\ cart \Rightarrow bool)) = -->\ f\ (f\ x)\ (within\ (at\ x)\ ?s)$

thm CONTINUOUS_AT:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ continuous\ f\ (at\ x) = -->\ f\ (f\ x)\ (at\ x)$

thm CONTINUOUS_AT_WITHIN:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ continuous\ f\ (at\ x) \longrightarrow continuous\ f\ (within\ (at\ x)\ s)$

thm CONTINUOUS_WITHIN_CLOSED_NONTRIVIAL:

∀ (a::(real, ?'b::type) cart) s::(real, ?'b::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ ¬ IN a s ⟶ continuous (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (within (at a) s)

thm CONTINUOUS_TRANSFORM_WITHIN:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (x::(real, ?'b::type) cart) d::real. (0::real) < d ∧ IN x s ∧ (∀ x'::(real, ?'b::type) cart. IN x' s ∧ distance (x', x) < d ⟶ f x' = g x') ∧ continuous f (within (at x) s) ⟶ continuous g (within (at x) s)

thm CONTINUOUS_TRANSFORM_AT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) d::real. (0::real) < d ∧ (∀ x'::(real, ?'b::type) cart. distance (x', x) < d ⟶ f x' = g x') ∧ continuous f (at x) ⟶ continuous g (at x)

thm continuous_within:

continuous (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (within (at (?x::(real, ?'b::type) cart)) (?s::(real, ?'b::type) cart ⇒ bool)) = (∀ e>0::real. ∃ d>0::real. ∀ x'::(real, ?'b::type) cart. IN x' ?s ∧ distance (x', ?x) < d ⟶ distance (?f x', ?f ?x) < e)

thm continuous_at:

continuous (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (at (?x::(real, ?'b::type) cart)) = (∀ e>0::real. ∃ d>0::real. ∀ x'::(real, ?'b::type) cart. distance (x', ?x) < d ⟶ distance (?f x', ?f ?x) < e)

thm CONTINUOUS_WITHIN_BALL:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) x::(real, ?'b::type) cart. continuous f (within (at x) s) = (∀ e>0::real. ∃ d>0::real. SUBSET (IMAGE f (HOL_Light_Import.INTER (ball (x, d)) s)) (ball (f x, e)))

thm CONTINUOUS_AT_BALL:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x::(real, ?'b::type) cart. continuous f (at x) = (∀ e>0::real. ∃ d>0::real. SUBSET (IMAGE f (ball (x, d))) (ball (f x, e)))

thm DEF_continuous_on:

continuous_on = (λ(_210042::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) _210043::(real, ?'b::type) cart ⇒ bool. ∀ x::(real, ?'b::type) cart. IN x _210043 ⟶ (∀ e>0::real. ∃ d>0::real. ∀ x'::(real, ?'b::type) cart. IN x' _210043 ∧ distance (x', x) < d ⟶ distance (_210042 x', _210042 x) < e))

thm continuous_on:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ s = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (\forall e>0::real.\ \exists d>0::real.\ \forall x'::(real,\ ?'b::type)\ cart.\ IN\ x'\ s \wedge distance\ (x',\ x) < d \longrightarrow distance\ (f\ x',\ f\ x) < e))$

thm DEF_uniformly_continuous_on:

$uniformly\_continuous\_on = (\lambda(\_210054::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ \_210055::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \forall e>0::real.\ \exists d>0::real.\ \forall (x::(real,\ ?'b::type)\ cart)\ x'::(real,\ ?'b::type)\ cart.\ IN\ x\ \_210055 \wedge IN\ x'\ \_210055 \wedge distance\ (x',\ x) < d \longrightarrow distance\ (\_210054\ x',\ \_210054\ x) < e)$

thm uniformly_continuous_on:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ uniformly\_continuous\_on\ f\ s = (\forall e>0::real.\ \exists d>0::real.\ \forall (x::(real,\ ?'b::type)\ cart)\ x'::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ x'\ s \wedge distance\ (x',\ x) < d \longrightarrow distance\ (f\ x',\ f\ x) < e)$

thm UNIFORMLY_CONTINUOUS_IMP_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \longrightarrow continuous\_on\ f\ s$

thm CONTINUOUS_AT_IMP_CONTINUOUS_ON:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow continuous\ f\ (at\ x)) \longrightarrow continuous\_on\ f\ s$

thm CONTINUOUS_ON_EQ_CONTINUOUS_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow continuous\ f\ (within\ (at\ x)\ s))$

thm CONTINUOUS_ON:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow --> f\ (f\ x)\ (within\ (at\ x)\ s))$

thm CONTINUOUS_ON_EQ_CONTINUOUS_AT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow continuous\_on\ f\ s = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow continuous\ f\ (at\ x))$

thm CONTINUOUS_WITHIN_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ continuous\ f\ (within\ (at\ x)\ s) \wedge SUBSET\ t\ s \longrightarrow continuous\ f\ (within\ (at\ x)\ t)$

thm CONTINUOUS_ON_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow continuous\_on\ f\ t$

thm UNIFORMLY_CONTINUOUS_ON_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow uniformly\_continuous\_on\ f\ t$

thm CONTINUOUS_ON_INTERIOR:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ continuous\_on\ f\ s \wedge IN\ x\ (interior\ s) \longrightarrow continuous\ f\ (at\ x)$

thm CONTINUOUS_ON_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge continuous\_on\ f\ s \longrightarrow continuous\_on\ g\ s$

thm UNIFORMLY_CONTINUOUS_ON_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge uniformly\_continuous\_on\ f\ s \longrightarrow uniformly\_continuous\_on\ g\ s$

thm CONTINUOUS_ON_SING:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ continuous\_on\ f\ (INSERT\ a\ EMPTY)$

thm CONTINUOUS_ON_EMPTY:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ EMPTY$

thm CONTINUOUS_WITHIN_SEQUENTIALLY:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ continuous\ f\ (within\ (at\ a)\ (?s::(real,\ ?'b::type)\ cart \Rightarrow bool)) = (\forall x::nat \Rightarrow (real,\ ?'b::type)\ cart.\ (\forall n::nat.\ IN\ (x\ n)\ ?s) \wedge --> x\ a\ sequentially \longrightarrow --> (f \circ x)\ (f\ a)\ sequentially)$

thm CONTINUOUS_AT_SEQUENTIALLY:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ continuous\ f\ (at\ a) = (\forall x::nat \Rightarrow (real,\ ?'b::type)\ cart.\ --> x\ a\ sequentially \longrightarrow --> (f \circ x)\ (f\ a)\ sequentially)$

thm CONTINUOUS_ON_SEQUENTIALLY:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall (x::nat \Rightarrow (real,\ ?'b::type)\ cart)\ a::(real,\ ?'b::type)$

cart. IN a s ∧ (∀ *n::nat. IN (x n) s*) ∧ *−−> x a sequentially* ⟶ *−−>* (*f* ∘ *x*) (*f a*) *sequentially*)

thm UNIFORMLY_CONTINUOUS_ON_SEQUENTIALLY:

∀ (*f::*(*real, ?'b::type*) *cart* ⇒ (*real, ?'a::type*) *cart*) *s::*(*real, ?'b::type*) *cart* ⇒ *bool. uniformly_continuous_on f s* = (∀ (*x::nat* ⇒ (*real, ?'b::type*) *cart*) *y::nat* ⇒ (*real, ?'b::type*) *cart*. (∀ *n::nat. IN (x n) s*) ∧ (∀ *n::nat. IN (y n) s*) ∧ *−−>* (*λn::nat. vector_sub (x n) (y n)*) (*vec (0::nat)*) *sequentially* ⟶ *−−>* (*λn::nat. vector_sub (f (x n)) (f (y n))*) (*vec (0::nat)*) *sequentially*)

thm LIM_CONTINUOUS_FUNCTION:

∀ (*f::*(*real, ?'c::type*) *cart* ⇒ (*real, ?'b::type*) *cart*) (*net::?'a::type net*) (*g::?'a::type* ⇒ (*real, ?'c::type*) *cart*) *l::*(*real, ?'c::type*) *cart. continuous f (at l)* ∧ *−−> g l net* ⟶ *−−>* (*λx::?'a::type. f (g x)*) (*f l*) *net*

thm CONTINUOUS_CONST:

∀ (*net::?'b::type net*) *c::*(*real, ?'a::type*) *cart. continuous* (*λx::?'b::type. c*) *net*

thm CONTINUOUS_CMUL:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) (*c::real*) *net::?'b::type net. continuous f net* ⟶ *continuous* (*λx::?'b::type.* % *c* (*f x*)) *net*

thm CONTINUOUS_NEG:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) *net::?'b::type net. continuous f net* ⟶ *continuous* (*λx::?'b::type. vector_neg (f x)*) *net*

thm CONTINUOUS_ADD:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) (*g::?'b::type* ⇒ (*real, ?'a::type*) *cart*) *net::?'b::type net. continuous f net* ∧ *continuous g net* ⟶ *continuous* (*λx::?'b::type. vector_add (f x) (g x)*) *net*

thm CONTINUOUS_SUB:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) (*g::?'b::type* ⇒ (*real, ?'a::type*) *cart*) *net::?'b::type net. continuous f net* ∧ *continuous g net* ⟶ *continuous* (*λx::?'b::type. vector_sub (f x) (g x)*) *net*

thm CONTINUOUS_ABS:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) *net::?'b::type net. continuous f net* ⟶ *continuous* (*λx::?'b::type. lambda* (*λi::nat.* |$ (*f x*) *i*|)) *net*

thm CONTINUOUS_MAX:

∀ (*f::?'b::type* ⇒ (*real, ?'a::type*) *cart*) (*g::?'b::type* ⇒ (*real, ?'a::type*) *cart*) *net::?'b::type net. continuous f net* ∧ *continuous g net* ⟶ *continuous* (*λx::?'b::type. lambda* (*λi::nat. max* ($ (*f x*) *i*) ($ (*g x*) *i*))) *net*

thm CONTINUOUS_MIN:

$\forall (f{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)$ $net{::}?'b{::}type\ net.\ continuous\ f\ net \wedge continuous\ g\ net \longrightarrow continuous\ (\lambda x{::}?'b{::}type.$ $lambda\ (\lambda i{::}nat.\ min\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ net$

thm CONTINUOUS_VSUM:

$\forall (net{::}?'c{::}type\ net)\ (f{::}?'b{::}type \Rightarrow ?'c{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}?'b{::}type$ $\Rightarrow bool.\ FINITE\ s \wedge (\forall\ a{::}?'b{::}type.\ IN\ a\ s \longrightarrow continuous\ (f\ a)\ net) \longrightarrow con$-$tinuous\ (\lambda x{::}?'c{::}type.\ vsum\ s\ (\lambda a{::}?'b{::}type.\ f\ a\ x))\ net$

thm CONTINUOUS_ON_CONST:

$\forall (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ c{::}(real,\ ?'a{::}type)\ cart.\ continuous\_on\ (\lambda x{::}(real,$ $?'b{::}type)\ cart.\ c)\ s$

thm CONTINUOUS_ON_CMUL:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (c{::}real)\ s{::}(real,\ ?'b{::}type)$ $cart \Rightarrow bool.\ continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.$ $\%\ c\ (f\ x))\ s$

thm CONTINUOUS_ON_NEG:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow$ $bool.\ continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ vector\_neg$ $(f\ x))\ s$

thm CONTINUOUS_ON_ADD:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart \Rightarrow$ $(real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge$ $continuous\_on\ g\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ vector\_add$ $(f\ x)\ (g\ x))\ s$

thm CONTINUOUS_ON_SUB:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart$ $\Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s$ $\wedge\ continuous\_on\ g\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ vector\_sub$ $(f\ x)\ (g\ x))\ s$

thm CONTINUOUS_ON_ABS:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow$ $bool.\ continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ lambda$ $(\lambda i{::}nat.\ |\$\ (f\ x)\ i|))\ s$

thm CONTINUOUS_ON_MAX:

$\forall (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'b{::}type)\ cart$ $\Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s$ $\wedge\ continuous\_on\ g\ s \longrightarrow continuous\_on\ (\lambda x{::}(real,\ ?'b{::}type)\ cart.\ lambda$ $(\lambda i{::}nat.\ max\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm CONTINUOUS_ON_MIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge continuous\_on\ g\ s \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ min\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm CONTINUOUS_ON_VSUM:

$\forall (t::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'b::type.\ IN\ a\ s \longrightarrow continuous\_on\ (f\ a)\ t) \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'c::type)\ cart.\ vsum\ s\ (\lambda a::?'b::type.\ f\ a\ x))\ t$

thm UNIFORMLY_CONTINUOUS_ON_CONST:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ c::(real,\ ?'a::type)\ cart.\ uniformly\_continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ c)\ s$

thm UNIFORMLY_CONTINUOUS_ON_CMUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \longrightarrow uniformly\_continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ c\ (f\ x))\ s$

thm UNIFORMLY_CONTINUOUS_ON_NEG:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \longrightarrow uniformly\_continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f\ x))\ s$

thm UNIFORMLY_CONTINUOUS_ON_ADD:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \wedge uniformly\_continuous\_on\ g\ s \longrightarrow uniformly\_continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x))\ s$

thm UNIFORMLY_CONTINUOUS_ON_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \wedge uniformly\_continuous\_on\ g\ s \longrightarrow uniformly\_continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm UNIFORMLY_CONTINUOUS_ON_VSUM:

$\forall (t::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'b::type.\ IN\ a\ s \longrightarrow uniformly\_continuous\_on\ (f\ a)\ t) \longrightarrow uniformly\_continuous\_on\ (\lambda x::(real,\ ?'c::type)\ cart.\ vsum\ s\ (\lambda a::?'b::type.\ f\ a\ x))\ t$

thm CONTINUOUS_WITHIN_ID:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ (within\ (at\ a)\ s)$

thm CONTINUOUS_AT_ID:

$\forall\, a::(real,\ ?'a::type)\ cart.\ continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ (at\ a)$

thm CONTINUOUS_ON_ID:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ s$

thm UNIFORMLY_CONTINUOUS_ON_ID:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ s$

thm CONTINUOUS_WITHIN_COMPOSE:

$\forall\, (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ continuous\ f\ (within\ (at\ x)\ s) \wedge continuous\ g\ (within\ (at\ (f\, x))\ (IMAGE\ f\ s)) \longrightarrow continuous\ (g \circ f)\ (within\ (at\ x)\ s)$

thm CONTINUOUS_AT_COMPOSE:

$\forall\, (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'c::type)\ cart.\ continuous\ f\ (at\ x) \wedge continuous\ g\ (at\ (f\, x)) \longrightarrow continuous\ (g \circ f)\ (at\ x)$

thm CONTINUOUS_ON_COMPOSE:

$\forall\, (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge continuous\_on\ g\ (IMAGE\ f\ s) \longrightarrow continuous\_on\ (g \circ f)\ s$

thm UNIFORMLY_CONTINUOUS_ON_COMPOSE:

$\forall\, (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \wedge uniformly\_continuous\_on\ g\ (IMAGE\ f\ s) \longrightarrow uniformly\_continuous\_on\ (g \circ f)\ s$

thm CONTINUOUS_AT_OPEN:

$\forall\, (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ continuous\ f\ (at\ x) = (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge IN\ (f\, x)\ t \longrightarrow (\exists\, s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge IN\ x\ s \wedge (\forall\, x'::(real,\ ?'b::type)\ cart.\ IN\ x'\ s \longrightarrow IN\ (f\, x')\ t)))$

thm CONTINUOUS_ON_OPEN:

$\forall\, (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\, s = (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (IMAGE\ f\ s))\ t \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%522::(real,\ ?'b::type)\ cart.\ \exists\, x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%522\ (IN\ x\ s \wedge IN\ (f\, x)\ t)\ x)))$

thm CONTINUOUS_ON_CLOSED:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ (IMAGE\ f\ s))\ t \longrightarrow closed\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%523::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%523\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x)))$

thm CONTINUOUS_OPEN_IN_PREIMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%524::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%524\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x))$

thm CONTINUOUS_CLOSED_IN_PREIMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow closed\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%525::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%525\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x))$

thm CONTINUOUS_OPEN_PREIMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.open\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%527::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%527\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x))$

thm CONTINUOUS_CLOSED_PREIMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.closed\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%529::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%529\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x))$

thm CONTINUOUS_OPEN_PREIMAGE_UNIV:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall\ x::(real,\ ?'b::type)\ cart.\ continuous\ f\ (at\ x)) \wedge HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%530::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%530\ (IN\ (f\ x)\ s)\ x))$

thm CONTINUOUS_CLOSED_PREIMAGE_UNIV:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall\ x::(real,\ ?'b::type)\ cart.\ continuous\ f\ (at\ x)) \wedge HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%531::(real,\ ?'b::type)\ cart.\ \exists\ x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%531\ (IN\ (f\ x)\ s)\ x))$

thm CONTINUOUS_OPEN_IN_PREIMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%532::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%532\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x)))$

thm CONTINUOUS_CLOSED_IN_PREIMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ t \longrightarrow closed\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%533::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%533\ (IN\ x\ s \wedge IN\ (f\ x)\ t)\ x)))$

thm LIM_LIFT_DOT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'a::type)\ cart.\ -->\ f\ (?l::(real,\ ?'a::type)\ cart)\ (?net::(real,\ ?'b::type)\ cart\ net) \longrightarrow -->\ (lift \circ (\lambda y::(real,\ ?'b::type)\ cart.\ dot\ a\ (f\ y)))\ (lift\ (dot\ a\ ?l))\ ?net$

thm CONTINUOUS_AT_LIFT_DOT:

$\forall (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ continuous\ (lift \circ dot\ a)\ (at\ x)$

thm CONTINUOUS_ON_LIFT_DOT:

$\forall\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (lift \circ dot\ (?a::(real,\ ?'a::type)\ cart))\ s$

thm CLOSED_INTERVAL_LEFT:

$\forall\ b::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%534::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%534\ (\forall\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ i \leq \$\ b\ i)\ x))$

thm CLOSED_INTERVAL_RIGHT:

$\forall\ a::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%535::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%535\ (\forall\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ x\ i)\ x))$

thm CLOSED_HALFSPACE_LE:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%538::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%538\ (dot\ a\ x \leq b)\ x))$

thm CLOSED_HALFSPACE_GE:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%539::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%539\ (b \leq dot\ a\ x)\ x))$

thm CLOSED_HYPERPLANE:

$\forall (a::(real, ?'a::type)\ cart)\ b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%543::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%543\ (dot\ a\ x = b)\ x))$

thm CLOSED_STANDARD_HYPERPLANE:

$\forall (k::nat)\ a::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%544::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%544\ (\$\ x\ k = a)\ x))$

thm CLOSED_HALFSPACE_COMPONENT_LE:

$\forall (a::real)\ k::nat.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%545::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%545\ (\$\ x\ k \leq a)\ x))$

thm CLOSED_HALFSPACE_COMPONENT_GE:

$\forall (a::real)\ k::nat.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%546::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%546\ (a \leq \$\ x\ k)\ x))$

thm OPEN_HALFSPACE_LT:

$\forall (a::(real, ?'a::type)\ cart)\ b::real.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%549::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%549\ (dot\ a\ x < b)\ x))$

thm OPEN_HALFSPACE_COMPONENT_LT:

$\forall (a::real)\ k::nat.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%550::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%550\ (\$\ x\ k < a)\ x))$

thm OPEN_HALFSPACE_GT:

$\forall (a::(real, ?'a::type)\ cart)\ b::real.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%553::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%553\ (b < dot\ a\ x)\ x))$

thm OPEN_HALFSPACE_COMPONENT_GT:

$\forall (a::real)\ k::nat.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%554::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%554\ (a < \$\ x\ k)\ x))$

thm OPEN_POSITIVE_MULTIPLES:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%555::(real, ?'a::type)\ cart.\ \exists (c::real)\ x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%555\ ((0::real) < c \wedge IN\ x\ s)\ (\%\ c\ x)))$

thm INTERIOR_HALFSPACE_LE:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \longrightarrow interior\ (GSPEC\ (\lambda GEN\%PVAR\%556::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%556\ (dot\ a\ x \leq b)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%557::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%557\ (dot\ a\ x < b)\ x)$

thm INTERIOR_HALFSPACE_GE:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \longrightarrow interior\ (GSPEC\ (\lambda GEN\%PVAR\%558::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%558\ (b \leq dot\ a\ x)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%559::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%559\ (b < dot\ a\ x)\ x)$

thm INTERIOR_HALFSPACE_COMPONENT_LE:

$\forall (a::real)\ k::nat.\ interior\ (GSPEC\ (\lambda GEN\%PVAR\%560::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%560\ (\$\ x\ k \leq a)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%561::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%561\ (\$\ x\ k < a)\ x)$

thm INTERIOR_HALFSPACE_COMPONENT_GE:

$\forall (a::real)\ k::nat.\ interior\ (GSPEC\ (\lambda GEN\%PVAR\%562::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%562\ (a \leq \$\ x\ k)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%563::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%563\ (a < \$\ x\ k)\ x)$

thm CLOSURE_HALFSPACE_LT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \longrightarrow closure\ (GSPEC\ (\lambda GEN\%PVAR\%566::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%566\ (dot\ a\ x < b)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%567::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%567\ (dot\ a\ x \leq b)\ x)$

thm CLOSURE_HALFSPACE_GT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \longrightarrow closure\ (GSPEC\ (\lambda GEN\%PVAR\%568::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%568\ (b < dot\ a\ x)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%569::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%569\ (b \leq dot\ a\ x)\ x)$

thm CLOSURE_HALFSPACE_COMPONENT_LT:

$\forall (a::real)\ k::nat.\ closure\ (GSPEC\ (\lambda GEN\%PVAR\%570::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%570\ (\$\ x\ k < a)\ x)) = GSPEC\ (\lambda GEN\%PVAR\%571::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%571\ (\$\ x\ k \leq a)\ x)$

thm CLOSURE_HALFSPACE_COMPONENT_GT:

$\forall (a::real)\ k::nat.\ closure\ (GSPEC\ (\lambda GEN\%PVAR\%572::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%572\ (a < \$\ x\ k)\ x)) =$

GSPEC (λ*GEN%PVAR%573*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%573* (*a* ≤ $ *x k*) *x*)

thm INTERIOR_HYPERPLANE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *a* ≠ *vec* (*0*::*nat*) ⟶ *interior* (*GSPEC* (λ*GEN%PVAR%577*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%577* (*dot a x* = *b*) *x*)) = *EMPTY*

thm INTERIOR_STANDARD_HYPERPLANE:

∀ (*k*::*nat*) *a*::*real*. *interior* (*GSPEC* (λ*GEN%PVAR%578*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%578* ($ *x k* = *a*) *x*)) = *EMPTY*

thm EMPTY_INTERIOR_LOWDIM:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dim s* < *dimindex HOL_Light_Import.UNIV* ⟶ *interior s* = *EMPTY*

thm UNBOUNDED_HALFSPACE_COMPONENT_LE:

∀ (*a*::*real*) *k*::*nat*. ¬ *bounded* (*GSPEC* (λ*GEN%PVAR%580*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%580* ($ *x k* ≤ *a*) *x*))

thm UNBOUNDED_HALFSPACE_COMPONENT_GE:

∀ (*a*::*real*) *k*::*nat*. ¬ *bounded* (*GSPEC* (λ*GEN%PVAR%581*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%581* (*a* ≤ $ *x k*) *x*))

thm UNBOUNDED_HALFSPACE_COMPONENT_LT:

∀ (*a*::*real*) *k*::*nat*. ¬ *bounded* (*GSPEC* (λ*GEN%PVAR%582*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%582* ($ *x k* < *a*) *x*))

thm UNBOUNDED_HALFSPACE_COMPONENT_GT:

∀ (*a*::*real*) *k*::*nat*. ¬ *bounded* (*GSPEC* (λ*GEN%PVAR%583*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%583* (*a* < $ *x k*) *x*))

thm BOUNDED_HALFSPACE_LE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *bounded* (*GSPEC* (λ*GEN%PVAR%585*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%585* (*dot a x* ≤ *b*) *x*)) = (*a* = *vec* (*0*::*nat*) ∧ *b* < (*0*::*real*))

thm BOUNDED_HALFSPACE_GE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *bounded* (*GSPEC* (λ*GEN%PVAR%586*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%586* (*b* ≤ *dot a x*) *x*)) = (*a* = *vec* (*0*::*nat*) ∧ (*0*::*real*) < *b*)

thm BOUNDED_HALFSPACE_LT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *bounded* (*GSPEC* (λ*GEN%PVAR%588*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%588* (*dot a x* < *b*) *x*)) = (*a* = *vec* (*0*::*nat*) ∧ *b* ≤ (*0*::*real*))

thm BOUNDED_HALFSPACE_GT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%589::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%589\ (b < dot\ a\ x)\ x)) = (a = vec\ (0::nat) \wedge (0::real) \leq b)$

thm FORALL_IN_CLOSURE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ t \wedge continuous\_on\ f\ (closure\ s) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closure\ s) \longrightarrow IN\ (f\ x)\ t)$

thm CONTINUOUS_LE_ON_CLOSURE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::real.\ continuous\_on\ (lift \circ f)\ (closure\ s) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\ x \leq a) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closure\ s) \longrightarrow f\ x \leq a)$

thm CONTINUOUS_GE_ON_CLOSURE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::real.\ continuous\_on\ (lift \circ f)\ (closure\ s) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow a \leq f\ x) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closure\ s) \longrightarrow a \leq f\ x)$

thm CONTINUOUS_CONSTANT_ON_CLOSURE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ continuous\_on\ f\ (closure\ s) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closure\ s) \longrightarrow f\ x = a)$

thm CONTINUOUS_AGREE_ON_CLOSURE:

$\forall (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ g\ (closure\ (?s::(real,\ ?'b::type)\ cart \Rightarrow bool)) \wedge continuous\_on\ h\ (closure\ ?s) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ ?s \longrightarrow g\ x = h\ x) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closure\ ?s) \longrightarrow g\ x = h\ x)$

thm CONTINUOUS_CLOSED_IN_PREIMAGE_CONSTANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ continuous\_on\ f\ s \longrightarrow closed\_in\ (subtopology\ euclidean\ s)\ (GSPEC\ (\lambda GEN\%PVAR\%595::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%595\ (IN\ x\ s \wedge f\ x = a)\ x))$

thm CONTINUOUS_CLOSED_PREIMAGE_CONSTANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%599::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%599\ (IN\ x\ s \wedge f\ x = (?a::(real,\ ?'a::type)\ cart))\ x))$

thm CONTINUOUS_ON_CLOSURE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ (closure\ s) = (\forall (x::(real,\ ?'b::type)\ cart)\ e::real.\ IN\ x\ (closure\ s) \wedge (0::real) < e \longrightarrow (\exists\, d>0::real.\ \forall\, y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \wedge distance\ (y,\ x) < d \longrightarrow distance\ (f\, y,\ f\, x) < e))$

thm CONTINUOUS_ON_CLOSURE_SEQUENTIALLY:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\, f\ (closure\ s) = (\forall (x::nat \Rightarrow (real,\ ?'b::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ IN\ a\ (closure\ s) \wedge (\forall\, n::nat.\ IN\ (x\, n)\ s) \wedge -->\ x\ a\ sequentially \longrightarrow -->\ (f \circ x)\ (f\, a)\ sequentially)$

thm UNIFORMLY_CONTINUOUS_ON_CLOSURE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\, f\, s \wedge continuous\_on\, f\ (closure\ s) \longrightarrow uniformly\_continuous\_on\ f\ (closure\ s)$

thm CONTINUOUS_AT_SQRT:

$\forall (a::(real,\ unit)\ cart)\ s::?'a::type.\ (0::real) < HOL\_Light\_Import.drop\ a \longrightarrow continuous\ (lift \circ (sqrt \circ HOL\_Light\_Import.drop))\ (at\ a)$

thm Hdplygy.CONTINUOUS_AT_SQRT:

$\forall\, a::(real,\ unit)\ cart.\ (0::real) < HOL\_Light\_Import.drop\ a \longrightarrow continuous\ (lift \circ (sqrt \circ HOL\_Light\_Import.drop))\ (at\ a)$

thm CONTINUOUS_WITHIN_LIFT_SQRT:

$\forall (a::(real,\ unit)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ (\forall\, x::(real,\ unit)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ x) \longrightarrow continuous\ (lift \circ (sqrt \circ HOL\_Light\_Import.drop))\ (within\ (at\ a)\ s)$

thm CONTINUOUS_ON_LIFT_SQRT:

$\forall\, s::(real,\ unit)\ cart \Rightarrow bool.\ (\forall\, x::(real,\ unit)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ x) \longrightarrow continuous\_on\ (lift \circ (sqrt \circ HOL\_Light\_Import.drop))\ s$

thm CONTINUOUS_ON_LIFT_SQRT_COMPOSE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (lift \circ f)\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (sqrt\ (f\, x)))\ s$

thm UNIFORMLY_CONTINUOUS_IMP_CAUCHY_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\, f\, s \longrightarrow (\forall\, x::nat \Rightarrow (real,\ ?'b::type)\ cart.\ cauchy\ x \wedge (\forall\, n::nat.\ IN\ (x\ n)\ s) \longrightarrow cauchy\ (f \circ x))$

thm CONTINUOUS_CLOSED_IMP_CAUCHY_CONTINUOUS:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. continuous_on f s ∧ HOL_Light_Import.closed s ⟶ (∀ x::nat ⇒ (real, ?'b::type) cart. cauchy x ∧ (∀ n::nat. IN (x n) s) ⟶ cauchy (f ∘ x))

thm CAUCHY_CONTINUOUS_UNIQUENESS_LEMMA:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀ x::nat ⇒ (real, ?'b::type) cart. cauchy x ∧ (∀ n::nat. IN (x n) s) ⟶ cauchy (f ∘ x)) ⟶ (∀ (a::(real, ?'b::type) cart) x::nat ⇒ (real, ?'b::type) cart. (∀ n::nat. IN (x n) s) ∧ --> x a sequentially ⟶ (∃ l::(real, ?'a::type) cart. --> (f ∘ x) l sequentially ∧ (∀ y::nat ⇒ (real, ?'b::type) cart. (∀ n::nat. IN (y n) s) ∧ --> y a sequentially ⟶ --> (f ∘ y) l sequentially)))

thm CAUCHY_CONTINUOUS_EXTENDS_TO_CLOSURE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀ x::nat ⇒ (real, ?'b::type) cart. cauchy x ∧ (∀ n::nat. IN (x n) s) ⟶ cauchy (f ∘ x)) ⟶ (∃ g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on g (closure s) ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g x = f x))

thm UNIFORMLY_CONTINUOUS_EXTENDS_TO_CLOSURE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. uniformly_continuous_on f s ⟶ (∃ g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. uniformly_continuous_on g (closure s) ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g x = f x) ∧ (∀ h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on h (closure s) ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ h x = f x) ⟶ (∀ x::(real, ?'b::type) cart. IN x (closure s) ⟶ h x = g x)))

thm CAUCHY_CONTINUOUS_IMP_CONTINUOUS:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀ x::nat ⇒ (real, ?'b::type) cart. cauchy x ∧ (∀ n::nat. IN (x n) s) ⟶ cauchy (f ∘ x)) ⟶ continuous_on f s

thm LINEAR_LIM_0:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. linear f ⟶ --> f (vec (0::nat)) (at (vec (0::nat)))

thm LINEAR_CONTINUOUS_AT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) a::(real, ?'b::type) cart. linear f ⟶ continuous f (at a)

thm LINEAR_CONTINUOUS_WITHIN:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) x::(real, ?'b::type) cart. linear f ⟶ continuous f (within (at x) s)

thm LINEAR_CONTINUOUS_ON:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. linear f ⟶ continuous_on f s

thm LINEAR_UNIFORMLY_CONTINUOUS_ON:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \longrightarrow uniformly\_continuous\_on\ f\ s$

thm BILINEAR_CONTINUOUS_AT_COMPOSE:

$\forall (f::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'c::type)\ cart)\ (g::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'d::type)\ cart.\ continuous\ f\ (at\ x) \wedge continuous\ g\ (at\ x) \wedge bilinear\ h \longrightarrow continuous\ (\lambda x::(real,\ ?'d::type)\ cart.\ h\ (f\ x)\ (g\ x))\ (at\ x)$

thm BILINEAR_CONTINUOUS_WITHIN_COMPOSE:

$\forall (f::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'c::type)\ cart)\ (g::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'d::type)\ cart)\ s::(real,\ ?'d::type)\ cart \Rightarrow bool.\ continuous\ f\ (within\ (at\ a)\ s) \wedge continuous\ g\ (within\ (at\ a)\ s) \wedge bilinear\ h \longrightarrow continuous\ (\lambda x::(real,\ ?'d::type)\ cart.\ h\ (f\ x)\ (g\ x))\ (within\ (at\ a)\ s)$

thm BILINEAR_CONTINUOUS_ON_COMPOSE:

$\forall (f::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'c::type)\ cart)\ (g::(real,\ ?'d::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'d::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge continuous\_on\ g\ s \wedge bilinear\ h \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'d::type)\ cart.\ h\ (f\ x)\ (g\ x))\ s$

thm CONTINUOUS_ON_LIFT_DOT2:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge continuous\_on\ g\ s \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (dot\ (f\ x)\ (g\ x)))\ s$

thm CONTINUOUS_AT_COMPOSE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ continuous\ g\ (at\ (?x::(real,\ ?'b::type)\ cart)) \wedge continuous\ h\ (at\ (g\ ?x)) \wedge (\forall y::(real,\ ?'b::type)\ cart.\ g\ (h\ y) = y) \wedge h\ (g\ ?x) = ?x \longrightarrow continuous\ f\ (at\ (g\ ?x)) = continuous\ (\lambda x::(real,\ ?'b::type)\ cart.\ f\ (g\ x))\ (at\ ?x)$

thm CONTINUOUS_AT_TRANSLATION:

$\forall (a::(real,\ ?'b::type)\ cart)\ (z::(real,\ ?'b::type)\ cart)\ f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\ f\ (at\ (vector\_add\ a\ z)) = continuous\ (\lambda x::(real,\ ?'b::type)\ cart.\ f\ (vector\_add\ a\ x))\ (at\ z)$

thm CONTINUOUS_AT_LINEAR_IMAGE:

$\forall (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (z::(real,\ ?'b::type)\ cart)\ f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ h \wedge (\forall x::(real,\ ?'b::type)$

cart. vector_norm (h x) = vector_norm x) ⟶ continuous f (at (h z)) = continuous (λx::(real, ?'b::type) cart. f (h x)) (at z)

thm INTERIOR_IMAGE_SUBSET:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀x::(real, ?'b::type) cart. continuous f (at x)) ∧ (∀(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f x = f y ⟶ x = y) ⟶ SUBSET (interior (IMAGE f s)) (IMAGE f (interior s))

thm CONTINUOUS_WITHIN_AVOID:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'a::type) cart. continuous f (within (at x) s) ∧ IN x s ∧ f x ≠ a ⟶ (∃e>0::real. ∀y::(real, ?'b::type) cart. IN y s ∧ distance (x, y) < e ⟶ f y ≠ a)

thm CONTINUOUS_AT_AVOID:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) a::(real, ?'a::type) cart. continuous f (at x) ∧ f x ≠ a ⟶ (∃e>0::real. ∀y::(real, ?'b::type) cart. distance (x, y) < e ⟶ f y ≠ a)

thm CONTINUOUS_ON_AVOID:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'a::type) cart. continuous_on f s ∧ IN x s ∧ f x ≠ a ⟶ (∃e>0::real. ∀y::(real, ?'b::type) cart. IN y s ∧ distance (x, y) < e ⟶ f y ≠ a)

thm CONTINUOUS_ON_OPEN_AVOID:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'a::type) cart. continuous_on f s ∧ HOL_Light_Import.open s ∧ IN x s ∧ f x ≠ a ⟶ (∃e>0::real. ∀y::(real, ?'b::type) cart. distance (x, y) < e ⟶ f y ≠ a)

thm CONTINUOUS_LEVELSET_OPEN_IN_CASES:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'a::type) cart. connected s ∧ continuous_on f s ∧ open_in (subtopology euclidean s) (GSPEC (λGEN%PVAR%604::(real, ?'b::type) cart. ∃x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%604 (IN x s ∧ f x = a) x)) ⟶ (∀x::(real, ?'b::type) cart. IN x s ⟶ f x ≠ a) ∨ (∀x::(real, ?'b::type) cart. IN x s ⟶ f x = a)

thm CONTINUOUS_LEVELSET_OPEN_IN:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) a::(real, ?'a::type) cart. connected s ∧ continuous_on f s ∧ open_in (subtopology euclidean s) (GSPEC (λGEN%PVAR%605::(real, ?'b::type) cart. ∃x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%605 (IN x s ∧ f x = a)

$x)) \wedge (\exists x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = a) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a)$

thm CONTINUOUS_LEVELSET_OPEN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ connected\ s \wedge continuous\_on\ f\ s \wedge HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%607::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%607\ (IN\ x\ s \wedge f\ x = a)\ x)) \wedge (\exists x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = a) \longrightarrow (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a)$

thm OPEN_SCALING:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ c \neq (0::real) \wedge HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ (\%\ c)\ s)$

thm OPEN_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ vector\_neg\ s)$

thm OPEN_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ (vector\_add\ a)\ s)$

thm OPEN_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ (IMAGE\ (vector\_add\ a)\ s) = HOL\_Light\_Import.open\ s$

thm OPEN_AFFINITY:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ c::real.\ HOL\_Light\_Import.open\ s \wedge c \neq (0::real) \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ a\ (\%\ c\ x))\ s)$

thm INTERIOR_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ interior\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (interior\ s)$

thm OPEN_SUMS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \vee HOL\_Light\_Import.open\ t \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%608::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%608\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm COMPACT_CONTINUOUS_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge compact\ s \longrightarrow compact\ (IMAGE\ f\ s)$

thm COMPACT_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ compact\ s \wedge linear\ f \longrightarrow compact\ (IMAGE\ f\ s)$

thm COMPACT_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow compact\ (IMAGE\ f\ s) = compact\ s$

thm CONNECTED_CONTINUOUS_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge connected\ s \longrightarrow connected\ (IMAGE\ f\ s)$

thm CONNECTED_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ s \longrightarrow connected\ (IMAGE\ (vector\_add\ a)\ s)$

thm CONNECTED_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ (IMAGE\ (vector\_add\ a)\ s) = connected\ s$

thm CONNECTED_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ connected\ s \wedge linear\ f \longrightarrow connected\ (IMAGE\ f\ s)$

thm CONNECTED_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow connected\ (IMAGE\ f\ s) = connected\ s$

thm BOUNDED_UNIFORMLY_CONTINUOUS_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ uniformly\_continuous\_on\ f\ s \wedge bounded\ s \longrightarrow bounded\ (IMAGE\ f\ s)$

thm DEF_connected_component:

$connected\_component = (\lambda(\_217274::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (\_217275::(real,\ ?'a::type)\ cart)\ \_217276::(real,\ ?'a::type)\ cart.\ \exists\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ t \wedge SUBSET\ t\ \_217274 \wedge IN\ \_217275\ t \wedge IN\ \_217276\ t)$

thm connected_component:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ y = (\exists\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ t \wedge SUBSET\ t\ s \wedge IN\ x\ t \wedge IN\ y\ t)$

thm CONNECTED_COMPONENT_IN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ y \longrightarrow IN\ x\ s \wedge IN\ y\ s$

thm CONNECTED_COMPONENT_REFL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow connected\_component\ s\ x\ x$

thm CONNECTED_COMPONENT_REFL_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ x = IN\ x\ s$

thm CONNECTED_COMPONENT_SYM:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ y \longrightarrow connected\_component\ s\ y\ x$

thm CONNECTED_COMPONENT_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ y \wedge connected\_component\ s\ y\ (?z::(real,\ ?'a::type)\ cart) \longrightarrow connected\_component\ s\ x\ ?z$

thm CONNECTED_COMPONENT_OF_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ t \wedge connected\_component\ s\ x\ (?y::(real,\ ?'a::type)\ cart) \longrightarrow connected\_component\ t\ x\ ?y$

thm CONNECTED_COMPONENT_SET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x = GSPEC\ (\lambda GEN\%PVAR\%612::(real,\ ?'a::type)\ cart.\ \exists\, y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%612\ (\exists\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ t \wedge SUBSET\ t\ s \wedge IN\ x\ t \wedge IN\ y\ t)\ y)$

thm CONNECTED_COMPONENT_UNIONS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%613::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%613\ (connected\ t \wedge IN\ x\ t \wedge SUBSET\ t\ s)\ t))$

thm CONNECTED_COMPONENT_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ SUBSET\ (connected\_component\ s\ x)\ s$

thm CONNECTED_CONNECTED_COMPONENT_SET:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow connected\_component\ s\ x = s)$

thm CONNECTED_COMPONENT_EQ_SELF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\ s \wedge IN\ x\ s \longrightarrow connected\_component\ s\ x = s$

thm CONNECTED_IFF_CONNECTED_COMPONENT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow connected\_component\ s\ x\ y)$

thm CONNECTED_COMPONENT_MAXIMAL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ t \wedge connected\ t \wedge SUBSET\ t\ s \longrightarrow SUBSET\ t\ (connected\_component\ s\ x)$

thm CONNECTED_COMPONENT_MONO:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ t \longrightarrow SUBSET\ (connected\_component\ s\ x)\ (connected\_component\ t\ x)$

thm CONNECTED_CONNECTED_COMPONENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\ (connected\_component\ s\ x)$

thm CONNECTED_COMPONENT_EQ_EMPTY:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ (connected\_component\ s\ x = EMPTY) = (\neg\ IN\ x\ s)$

thm CONNECTED_COMPONENT_EMPTY:

$\forall x::(real,\ ?'a::type)\ cart.\ connected\_component\ EMPTY\ x = EMPTY$

thm CONNECTED_COMPONENT_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ y\ (connected\_component\ s\ x) \longrightarrow connected\_component\ s\ y = connected\_component\ s\ x$

thm CLOSED_CONNECTED_COMPONENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (connected\_component\ s\ x)$

thm CONNECTED_COMPONENT_DISJOINT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ DISJOINT\ (connected\_component\ s\ a)\ (connected\_component\ s\ b) = (\neg\ IN\ a\ (connected\_component\ s\ b))$

thm CONNECTED_COMPONENT_NONOVERLAP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (HOL\_Light\_Import.INTER\ (connected\_component\ s\ a)\ (connected\_component\ s\ b) = EMPTY) = (\neg\ IN\ a\ s \vee \neg\ IN\ b\ s \vee connected\_component\ s\ a \neq connected\_component\ s\ b)$

thm CONNECTED_COMPONENT_OVERLAP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (HOL\_Light\_Import.INTER\ (connected\_component\ s\ a)\ (connected\_component\ s\ b) \neq EMPTY) = (IN\ a\ s \wedge IN\ b\ s \wedge connected\_component\ s\ a = connected\_component\ s\ b)$

thm CONNECTED_COMPONENT_SYM_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\_component\ s\ x\ y = connected\_component\ s\ y\ x$

thm CONNECTED_COMPONENT_EQ_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (connected\_component\ s\ x = connected\_component\ s\ y) = (\neg\ IN\ x\ s \wedge \neg\ IN\ y\ s \vee IN\ x\ s \wedge IN\ y\ s \wedge connected\_component\ s\ x\ y)$

thm CONNECTED_EQ_CONNECTED_COMPONENT_EQ:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow connected\_component\ s\ x = connected\_component\ s\ y)$

thm CONNECTED_COMPONENT_IDEMP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\_component\ (connected\_component\ s\ x)\ x = connected\_component\ s\ x$

thm CONNECTED_COMPONENT_UNIQUE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (c::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ c \wedge SUBSET\ c\ s \wedge connected\ c \wedge (\forall c'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ c' \wedge SUBSET\ c'\ s \wedge connected\ c' \longrightarrow SUBSET\ c'\ c) \longrightarrow connected\_component\ s\ x = c$

thm JOINABLE_CONNECTED_COMPONENT_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ connected\ t \wedge SUBSET\ t\ s \wedge HOL\_Light\_Import.INTER\ (connected\_component\ s\ x)\ t \neq EMPTY \wedge HOL\_Light\_Import.INTER\ (connected\_component\ s\ y)\ t \neq EMPTY \longrightarrow connected\_component\ s\ x = connected\_component\ s\ y$

thm CONNECTED_COMPONENT_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ connected\_component\ (IMAGE\ (vector\_add\ a)\ s)\ (vector\_add\ a\ x) = IMAGE\ (vector\_add\ a)\ (connected\_component\ s\ x)$

thm CONNECTED_COMPONENT_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \longrightarrow connected\_component\ (IMAGE\ f\ s)\ (f\ x) = IMAGE\ f\ (connected\_component\ s\ x)$

thm UNIONS_CONNECTED_COMPONENT:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%614::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%614\ (IN\ x\ s)\ (connected\_component\ s\ x))) = s$

thm CLOSED_IN_CONNECTED_COMPONENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ closed\_in\ (subtopology\ euclidean\ s)\ (connected\_component\ s\ x)$

thm OPEN_IN_CONNECTED_COMPONENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%616::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%616\ (IN\ x\ s)\ (connected\_component\ s\ x))) \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ (connected\_component\ s\ x)$

thm DEF_components:

$components = (\lambda\_217635::(real, ?'a::type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%617::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%617\ (IN\ x\ \_217635)\ (connected\_component\ \_217635\ x)))$

thm components:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ components\ s = GSPEC\ (\lambda GEN\%PVAR\%617::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%617\ (IN\ x\ s)\ (connected\_component\ s\ x))$

thm IN_COMPONENTS:

$\forall (u::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ (components\ u) = (\exists x::(real, ?'a::type)\ cart.\ IN\ x\ u \land s = connected\_component\ u\ x)$

thm UNIONS_COMPONENTS:

$\forall u::(real, ?'a::type)\ cart \Rightarrow bool.\ u = UNIONS\ (components\ u)$

thm PAIRWISE_DISJOINT_COMPONENTS:

$\forall u::(real, ?'a::type)\ cart \Rightarrow bool.\ pairwise\ DISJOINT\ (components\ u)$

thm IN_COMPONENTS_NONEMPTY:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \longrightarrow c \neq EMPTY$

thm IN_COMPONENTS_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \longrightarrow SUBSET\ c\ s$

thm IN_COMPONENTS_CONNECTED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \longrightarrow connected\ c$

thm IN_COMPONENTS_MAXIMAL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) = (c \neq EMPTY \wedge SUBSET\ c\ s \wedge connected\ c \wedge (\forall c'::(real, ?'a::type)\ cart \Rightarrow bool.\ c' \neq EMPTY \wedge SUBSET\ c\ c' \wedge SUBSET\ c'\ s \wedge connected\ c' \longrightarrow c' = c))$

thm JOINABLE_COMPONENTS_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ (c1::(real, ?'a::type)\ cart \Rightarrow bool)\ c2::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ t \wedge SUBSET\ t\ s \wedge IN\ c1\ (components\ s) \wedge IN\ c2\ (components\ s) \wedge HOL\_Light\_Import.INTER\ c1\ t \neq EMPTY \wedge HOL\_Light\_Import.INTER\ c2\ t \neq EMPTY \longrightarrow c1 = c2$

thm CLOSED_COMPONENTS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge IN\ c\ (components\ s) \longrightarrow HOL\_Light\_Import.closed\ c$

thm CONTINUOUS_ON_COMPONENTS_GEN:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ (\forall c::(real, ?'b::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \longrightarrow open\_in\ (subtopology\ euclidean\ s)\ c \wedge continuous\_on\ f\ c) \longrightarrow continuous\_on\ f\ s$

thm COMPONENTS_NONOVERLAP:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (c::(real, ?'a::type)\ cart \Rightarrow bool)\ c'::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \wedge IN\ c'\ (components\ s) \longrightarrow (HOL\_Light\_Import.INTER\ c\ c' = EMPTY) = (c \neq c')$

thm COMPONENTS_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (c::(real, ?'a::type)\ cart \Rightarrow bool)\ c'::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \wedge IN\ c'\ (components\ s) \longrightarrow (c = c') = (HOL\_Light\_Import.INTER\ c\ c' \neq EMPTY)$

thm COMPONENTS_EQ_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (components\ s = EMPTY) = (s = EMPTY)$

thm CONNECTED_EQ_CONNECTED_COMPONENTS_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ s = (\forall (c::(real, ?'a::type)\ cart \Rightarrow bool)\ c'::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ c\ (components\ s) \wedge IN\ c'\ (components\ s) \longrightarrow c = c')$

thm COMPONENTS_EQ_SING_EXISTS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (\exists a::(real, ?'a::type)\ cart \Rightarrow bool.\ components\ s = INSERT\ a\ EMPTY) = (connected\ s \wedge s \neq EMPTY)$

thm COMPONENTS_EQ_SING:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (components\ s = INSERT\ s\ EMPTY) = (connected\ s \wedge s \neq EMPTY)$

thm COMPACT_UNIFORMLY_EQUICONTINUOUS:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) s::(real, ?'b::type) cart ⇒ bool. (∀ (x::(real, ?'b::type) cart) e::real. IN x s ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x'::(real, ?'b::type) cart. IN f fs ∧ IN x' s ∧ distance (x', x) < d ⟶ distance (f x', f x) < e)) ∧ compact s ⟶ (∀ e>0::real. ∃ d>0::real. ∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) x'::(real, ?'b::type) cart. IN f fs ∧ IN x s ∧ IN x' s ∧ distance (x', x) < d ⟶ distance (f x', f x) < e)

thm COMPACT_UNIFORMLY_CONTINUOUS:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. continuous_on f s ∧ compact s ⟶ uniformly_continuous_on f s

thm CONTINUOUS_ON_INVERSE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'a::type) cart ⇒ (real, ?'b::type) cart) s::(real, ?'b::type) cart ⇒ bool. continuous_on f s ∧ compact s ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g (f x) = x) ⟶ continuous_on g (IMAGE f s)

thm CONTINUOUS_UNIFORM_LIMIT:

∀ (net::?'c::type net) (f::?'c::type ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. ¬ trivial_limit net ∧ eventually (λn::?'c::type. continuous_on (f n) s) net ∧ (∀ e>0::real. eventually (λn::?'c::type. ∀ x::(real, ?'b::type) cart. IN x s ⟶ vector_norm (vector_sub (f n x) (g x)) < e) net) ⟶ continuous_on g s

thm OPEN_LIFT:

∀ s::real ⇒ bool. HOL_Light_Import.open (IMAGE lift s) = (∀ x::real. IN x s ⟶ (∃ e>0::real. ∀ x'::real. |x' − x| < e ⟶ IN x' s))

thm LIMPT_APPROACHABLE_LIFT:

∀ (x::real) s::real ⇒ bool. limit_point_of (lift x) (IMAGE lift s) = (∀ e>0::real. ∃ x'::real. IN x' s ∧ x' ≠ x ∧ |x' − x| < e)

thm CLOSED_LIFT:

∀ s::real ⇒ bool. HOL_Light_Import.closed (IMAGE lift s) = (∀ x::real. (∀ e>0::real. ∃ x'::real. IN x' s ∧ x' ≠ x ∧ |x' − x| < e) ⟶ IN x s)

thm CONTINUOUS_AT_LIFT_RANGE:

∀ (f::(real, ?'a::type) cart ⇒ real) x::(real, ?'a::type) cart. continuous (lift ∘ f) (at x) = (∀ e>0::real. ∃ d>0::real. ∀ x'::(real, ?'a::type) cart. vector_norm (vector_sub x' x) < d ⟶ |f x' − f x| < e)

thm CONTINUOUS_ON_LIFT_RANGE:

∀ (f::(real, ?'a::type) cart ⇒ real) s::(real, ?'a::type) cart ⇒ bool. continuous_on (lift ∘ f) s = (∀ x::(real, ?'a::type) cart. IN x s ⟶ (∀ e>0::real. ∃ d>0::real.

$\forall x'::(real, ?'a::type)\ cart.\ IN\ x'\ s \wedge vector\_norm\ (vector\_sub\ x'\ x) < d \longrightarrow |f\ x' - f\ x| < e))$

thm CONTINUOUS_LIFT_NORM_COMPOSE:

$\forall (net::?'b::type\ net)\ f::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ continuous\ f\ net \longrightarrow continuous\ (\lambda x::?'b::type.\ lift\ (vector\_norm\ (f\ x)))\ net$

thm CONTINUOUS_ON_LIFT_NORM_COMPOSE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x::(real, ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\ x)))\ s$

thm CONTINUOUS_AT_LIFT_NORM:

$\forall x::(real, ?'a::type)\ cart.\ continuous\ (lift \circ vector\_norm)\ (at\ x)$

thm CONTINUOUS_ON_LIFT_NORM:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (lift \circ vector\_norm)\ s$

thm CONTINUOUS_AT_LIFT_COMPONENT:

$\forall (i::nat)\ a::(real, ?'a::type)\ cart.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow continuous\ (\lambda x::(real, ?'a::type)\ cart.\ lift\ (\$\ x\ i))\ (at\ a)$

thm CONTINUOUS_ON_LIFT_COMPONENT:

$\forall (i::nat)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow continuous\_on\ (\lambda x::(real, ?'a::type)\ cart.\ lift\ (\$\ x\ i))\ s$

thm CONTINUOUS_AT_LIFT_INFNORM:

$\forall x::(real, ?'a::type)\ cart.\ continuous\ (lift \circ infnorm)\ (at\ x)$

thm CONTINUOUS_AT_LIFT_DIST:

$\forall (a::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ continuous\ (lift \circ (\lambda x::(real, ?'a::type)\ cart.\ distance\ (a, x)))\ (at\ x)$

thm CONTINUOUS_ON_LIFT_DIST:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (lift \circ (\lambda x::(real, ?'a::type)\ cart.\ distance\ (a, x)))\ s$

thm COMPACT_ATTAINS_SUP:

$\forall s::real \Rightarrow bool.\ compact\ (IMAGE\ lift\ s) \wedge s \neq EMPTY \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow y \leq x))$

thm COMPACT_ATTAINS_INF:

$\forall s::real \Rightarrow bool.\ compact\ (IMAGE\ lift\ s) \wedge s \neq EMPTY \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow x \leq y))$

thm CONTINUOUS_ATTAINS_SUP:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge s \neq EMPTY \wedge continuous\_on\ (lift \circ f)\ s \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow f\ y \leq f\ x))$

thm CONTINUOUS_ATTAINS_INF:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge s \neq EMPTY \wedge continuous\_on\ (lift \circ f)\ s \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow f\ x \leq f\ y))$

thm DISTANCE_ATTAINS_SUP:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ compact\ s \wedge s \neq EMPTY \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow distance\ (a,\ y) \leq distance\ (a,\ x)))$

thm DISTANCE_ATTAINS_INF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ y)))$

thm LIM_MUL:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (c::?'b::type \Rightarrow real)\ d::real.\ --> (lift \circ c)\ (lift\ d)\ net \wedge --> f\ l\ net \longrightarrow --> (\lambda x::?'b::type.\ \%\ (c\ x)\ (f\ x))\ (\%\ d\ l)\ net$

thm LIM_VMUL:

$\forall (net::?'b::type\ net)\ (c::?'b::type \Rightarrow real)\ (d::real)\ v::(real,\ ?'a::type)\ cart.\ --> (lift \circ c)\ (lift\ d)\ net \longrightarrow --> (\lambda x::?'b::type.\ \%\ (c\ x)\ v)\ (\%\ d\ v)\ net$

thm CONTINUOUS_VMUL:

$\forall (net::?'b::type\ net)\ (c::?'b::type \Rightarrow real)\ v::(real,\ ?'a::type)\ cart.\ continuous\ (lift \circ c)\ net \longrightarrow continuous\ (\lambda x::?'b::type.\ \%\ (c\ x)\ v)\ net$

thm CONTINUOUS_MUL:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ c::?'b::type \Rightarrow real.\ continuous\ (lift \circ c)\ net \wedge continuous\ f\ net \longrightarrow continuous\ (\lambda x::?'b::type.\ \%\ (c\ x)\ (f\ x))\ net$

thm CONTINUOUS_ON_VMUL:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (c::(real,\ ?'b::type)\ cart \Rightarrow real)\ v::(real,\ ?'a::type)\ cart.\ continuous\_on\ (lift \circ c)\ s \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ (c\ x)\ v)\ s$

thm CONTINUOUS_ON_MUL:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (c::(real,\ ?'b::type)\ cart \Rightarrow real)\ f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ (lift \circ c)\ s \wedge continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ (c\ x)\ (f\ x))\ s$

thm LIM_INV:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *l*::*real*. −−> (*lift* $\circ$ *f*) (*lift l*) *net* $\wedge$ *l* $\neq$ (*0*::*real*) $\longrightarrow$ −−> (*lift* $\circ$ (*inverse_class.inverse* $\circ$ *f*)) (*lift* (*inverse_class.inverse* *l*)) *net*

thm CONTINUOUS_INV:

$\forall$ (*net*::*?'a*::*type net*) *f*::*?'a*::*type* $\Rightarrow$ *real*. *continuous* (*lift* $\circ$ *f*) *net* $\wedge$ *f* (*netlimit* *net*) $\neq$ (*0*::*real*) $\longrightarrow$ *continuous* (*lift* $\circ$ (*inverse_class.inverse* $\circ$ *f*)) *net*

thm CONTINUOUS_AT_WITHIN_INV:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous* (*lift* $\circ$ *f*) (*within* (*at a*) *s*) $\wedge$ *f a* $\neq$ (*0*::*real*) $\longrightarrow$ *continuous* (*lift* $\circ$ (*inverse_class.inverse* $\circ$ *f*)) (*within* (*at a*) *s*)

thm CONTINUOUS_AT_INV:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous* (*lift* $\circ$ *f*) (*at a*) $\wedge$ *f a* $\neq$ (*0*::*real*) $\longrightarrow$ *continuous* (*lift* $\circ$ (*inverse_class.inverse* $\circ$ *f*)) (*at a*)

thm CONTINUOUS_ON_INV:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *continuous_on* (*lift* $\circ$ *f*) *s* $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ *f x* $\neq$ (*0*::*real*)) $\longrightarrow$ *continuous_on* (*lift* $\circ$ (*inverse_class.inverse* $\circ$ *f*)) *s*

thm BOUNDED_PASTECART:

$\forall$ (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded* *s* $\wedge$ *bounded t* $\longrightarrow$ *bounded* (*GSPEC* (λ*GEN%PVAR%623*::(*real*, (*?'b*::*type*, *?'a*::*type*) *finite_sum*) *cart*. $\exists$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%623* (*IN x s* $\wedge$ *IN y t*) (*pastecart x y*)))

thm CLOSED_PASTECART:

$\forall$ (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed* *s* $\wedge$ *HOL_Light_Import.closed t* $\longrightarrow$ *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%624*::(*real*, (*?'b*::*type*, *?'a*::*type*) *finite_sum*) *cart*. $\exists$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%624* (*IN x s* $\wedge$ *IN y t*) (*pastecart x y*)))

thm COMPACT_PASTECART:

$\forall$ (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact* *s* $\wedge$ *compact t* $\longrightarrow$ *compact* (*GSPEC* (λ*GEN%PVAR%625*::(*real*, (*?'b*::*type*, *?'a*::*type*) *finite_sum*) *cart*. $\exists$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%625* (*IN x s* $\wedge$ *IN y t*) (*pastecart x y*)))

thm LIM_PASTECART:

$\forall$ (*net*::*?'c*::*type net*) (*f*::*?'c*::*type* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) *g*::*?'c*::*type* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. −−> *f* (*?a*::(*real*, *?'b*::*type*) *cart*) *net* $\wedge$ −−> *g* (*?b*::(*real*,

?'a::type) cart) net ⟶ −−> (λx::?'c::type. pastecart (f x) (g x)) (pastecart ?a ?b) net

thm CONTINUOUS_PASTECART:

∀(net::?'c::type net) (f::?'c::type ⇒ (real, ?'b::type) cart) g::?'c::type ⇒ (real, ?'a::type) cart. continuous f net ∧ continuous g net ⟶ continuous (λx::?'c::type. pastecart (f x) (g x)) net

thm CONTINUOUS_ON_PASTECART:

∀(f::(real, ?'c::type) cart ⇒ (real, ?'b::type) cart) (g::(real, ?'c::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'c::type) cart ⇒ bool. continuous_on f s ∧ continuous_on g s ⟶ continuous_on (λx::(real, ?'c::type) cart. pastecart (f x) (g x)) s

thm OPEN_PASTECART:

∀(s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.open s ∧ HOL_Light_Import.open t ⟶ HOL_Light_Import.open (GSPEC (λGEN%PVAR%629::(real, (?'b::type, ?'a::type) finite_sum) cart. ∃(x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%629 (IN x s ∧ IN y t) (pastecart x y)))

thm CONNECTED_PASTECART:

∀(s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. connected s ∧ connected t ⟶ connected (GSPEC (λGEN%PVAR%630::(real, (?'b::type, ?'a::type) finite_sum) cart. ∃(x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%630 (IN x s ∧ IN y t) (pastecart x y)))

thm COMPACT_SCALING:

∀(s::(real, ?'a::type) cart ⇒ bool) c::real. compact s ⟶ compact (IMAGE (% c) s)

thm COMPACT_NEGATIONS:

∀s::(real, ?'a::type) cart ⇒ bool. compact s ⟶ compact (IMAGE vector_neg s)

thm COMPACT_SUMS:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. compact s ∧ compact t ⟶ compact (GSPEC (λGEN%PVAR%633::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%633 (IN x s ∧ IN y t) (vector_add x y)))

thm COMPACT_DIFFERENCES:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. compact s ∧ compact t ⟶ compact (GSPEC (λGEN%PVAR%636::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%636 (IN x s ∧ IN y t) (vector_sub x y)))

thm COMPACT_TRANSLATION_EQ:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact* (*IMAGE* (*vector_add a*) *s*) = *compact s*

thm COMPACT_TRANSLATION:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *compact s* $\longrightarrow$ *compact* (*IMAGE* (*vector_add a*) *s*)

thm COMPACT_AFFINITY:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. *compact s* $\longrightarrow$ *compact* (*IMAGE* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vector_add a* (% *c x*)) *s*)

thm COMPACT_SUP_MAXDISTANCE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\wedge$ ($\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u s* $\wedge$ *IN v s* $\longrightarrow$ *vector_norm* (*vector_sub u v*) $\leq$ *vector_norm* (*vector_sub x y*)))

thm DEF_diameter:

diameter = (λ*_221737*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *if _221737* = *EMPTY then 0*::*real else HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%638*::*real*. $\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%638* (*IN x _221737* $\wedge$ *IN y _221737*) (*vector_norm* (*vector_sub x y*)))))

thm diameter:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *diameter s* = (*if s* = *EMPTY then 0*::*real else HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%638*::*real*. $\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%638* (*IN x s* $\wedge$ *IN y s*) (*vector_norm* (*vector_sub x y*)))))

thm DIAMETER_BOUNDED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\longrightarrow$ *vector_norm* (*vector_sub x y*) $\leq$ *diameter s*) $\wedge$ ($\forall$ *d*::*real*. (*0*::*real*) $\leq$ *d* $\wedge$ *d* < *diameter s* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\wedge$ *d* < *vector_norm* (*vector_sub x y*)))

thm DIAMETER_BOUNDED_BOUND:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *bounded s* $\wedge$ *IN x s* $\wedge$ *IN y s* $\longrightarrow$ *vector_norm* (*vector_sub x y*) $\leq$ *diameter s*

thm DIAMETER_COMPACT_ATTAINED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\wedge$ *vector_norm* (*vector_sub x y*) = *diameter s*)

thm DIAMETER_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *diameter* (*IMAGE* (*vector_add a*) *s*) = *diameter s*

thm DIAMETER_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\longrightarrow$ *diameter* (*IMAGE f s*) = *diameter s*

thm DIAMETER_EMPTY:

diameter EMPTY = (*0*::*real*)

thm DIAMETER_SING:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. *diameter* (*INSERT a EMPTY*) = (*0*::*real*)

thm DIAMETER_POS_LE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ (*0*::*real*) $\leq$ *diameter s*

thm DIAMETER_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET s t* $\wedge$ *bounded t* $\longrightarrow$ *diameter s* $\leq$ *diameter t*

thm DIAMETER_CLOSURE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ *diameter* (*closure s*) = *diameter s*

thm DIAMETER_SUBSET_CBALL_NONEMPTY:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *z*::(*real*, *?'a*::*type*) *cart*. *IN z s* $\wedge$ *SUBSET s* (*cball* (*z*, *diameter s*)))

thm DIAMETER_SUBSET_CBALL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ ($\exists$ *z*::(*real*, *?'a*::*type*) *cart*. *SUBSET s* (*cball* (*z*, *diameter s*)))

thm LEBESGUE_COVERING_LEMMA:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *c*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *compact s* $\wedge$ *c* $\neq$ *EMPTY* $\wedge$ *SUBSET s* (*UNIONS c*) $\wedge$ ($\forall$ *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN b c* $\longrightarrow$ *HOL_Light_Import.open b*) $\longrightarrow$ ($\exists$ *d*>*0*::*real*. $\forall$ *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET t s* $\wedge$ *diameter t* $\leq$ *d* $\longrightarrow$ ($\exists$ *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN b c* $\wedge$ *SUBSET t b*))

thm DIAMETER_EQ_0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ (*diameter s* = (*0*::*real*)) = (*s* = *EMPTY* $\vee$ ($\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *s* = *INSERT a EMPTY*))

thm CLOSED_SCALING:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *c*::*real*. *HOL_Light_Import.closed s* $\longrightarrow$ *HOL_Light_Import.closed* (*IMAGE* (% *c*) *s*)

thm CLOSED_NEGATIONS:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ vector\_neg\ s)$

thm COMPACT_CLOSED_SUMS:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%643{::}(real,\ ?'a{::}type)\ cart.\ \exists\, (x{::}(real,\ ?'a{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%643\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm CLOSED_COMPACT_SUMS:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge compact\ t \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%646{::}(real,\ ?'a{::}type)\ cart.\ \exists\, (x{::}(real,\ ?'a{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%646\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm COMPACT_CLOSED_DIFFERENCES:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%649{::}(real,\ ?'a{::}type)\ cart.\ \exists\, (x{::}(real,\ ?'a{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%649\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm CLOSED_COMPACT_DIFFERENCES:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge compact\ t \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%652{::}(real,\ ?'a{::}type)\ cart.\ \exists\, (x{::}(real,\ ?'a{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%652\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm CLOSED_TRANSLATION_EQ:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ (IMAGE\ (vector\_add\ a)\ s) = HOL\_Light\_Import.closed\ s$

thm CLOSED_TRANSLATION:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ a{::}(real,\ ?'a{::}type)\ cart.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ (vector\_add\ a)\ s)$

thm COMPLETE_TRANSLATION_EQ:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ complete\ (IMAGE\ (vector\_add\ a)\ s) = complete\ s$

thm TRANSLATION_UNIV:

$\forall\, a{::}(real,\ ?'a{::}type)\ cart.\ IMAGE\ (vector\_add\ a)\ HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm TRANSLATION_DIFF:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. IMAGE (vector_add (?a::(real, ?'a::type) cart)) (DIFF s t) = DIFF (IMAGE (vector_add ?a) s) (IMAGE (vector_add ?a) t)

thm CLOSURE_TRANSLATION:

∀(a::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. closure (IMAGE (vector_add a) s) = IMAGE (vector_add a) (closure s)

thm FRONTIER_TRANSLATION:

∀(a::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. frontier (IMAGE (vector_add a) s) = IMAGE (vector_add a) (frontier s)

thm SEPARATE_POINT_CLOSED:

∀(s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. HOL_Light_Import.closed s ∧ ¬ IN a s ⟶ (∃d>0::real. ∀x::(real, ?'a::type) cart. IN x s ⟶ d ≤ distance (a, x))

thm SEPARATE_COMPACT_CLOSED:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. compact s ∧ HOL_Light_Import.closed t ∧ HOL_Light_Import.INTER s t = EMPTY ⟶ (∃d>0::real. ∀(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y t ⟶ d ≤ distance (x, y))

thm SEPARATE_CLOSED_COMPACT:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ compact t ∧ HOL_Light_Import.INTER s t = EMPTY ⟶ (∃d>0::real. ∀(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y t ⟶ d ≤ distance (x, y))

thm OPEN_UNION_COMPACT_SUBSETS:

∀s::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.open s ⟶ (∃f::nat ⇒ (real, ?'a::type) cart ⇒ bool. (∀n::nat. compact (f n)) ∧ (∀n::nat. SUBSET (f n) s) ∧ (∀n::nat. SUBSET (f n) (interior (f (n + (1::nat))))) ∧ UNIONS (GSPEC (λGEN%PVAR%661::(real, ?'a::type) cart ⇒ bool. ∃n::nat. SETSPEC GEN%PVAR%661 (IN n HOL_Light_Import.UNIV) (f n))) = s ∧ (∀k::(real, ?'a::type) cart ⇒ bool. compact k ∧ SUBSET k s ⟶ (∃N::nat. ∀n≥N. SUBSET k (f n))))

thm DEF_open_interval:

open_interval = (λ_223756::(real, ?'a::type) cart × (real, ?'a::type) cart. GSPEC (λGEN%PVAR%662::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%662 (∀i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ (fst _223756) i < $ x i ∧ $ x i < $ (snd _223756) i) x))

thm open_interval:

∀(a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. open_interval (a, b) = GSPEC (λGEN%PVAR%662::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart.

SETSPEC GEN%PVAR%662 (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *a i* < $ *x i* ∧ $ *x i* < $ *b i*) *x*)

thm DEF_closed_interval:

closed_interval = (λ*_223765*::((*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*) *list*. *GSPEC* (λ*GEN%PVAR%663*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%663* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ (*fst* (*hd _223765*)) *i* ≤ $ *x i* ∧ $ *x i* ≤ $ (*snd* (*hd _223765*)) *i*) *x*))

thm closed_interval:

∀ *l*::((*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*) *list*. *closed_interval l* = *GSPEC* (λ*GEN%PVAR%663*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%663* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ (*fst* (*hd l*)) *i* ≤ $ *x i* ∧ $ *x i* ≤ $ (*snd* (*hd l*)) *i*) *x*)

thm interval:

open_interval (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*) = *GSPEC* (λ*GEN%PVAR%664*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%664* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *x i* ∧ $ *x i* < $ *?b i*) *x*) ∧ *closed_interval* [(*?a*, *?b*)] = *GSPEC* (λ*GEN%PVAR%665*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%665* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *x i* ∧ $ *x i* ≤ $ *?b i*) *x*)

thm interval_conjunct1:

closed_interval [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)] = *GSPEC* (λ*GEN%PVAR%665*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%665* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *x i* ∧ $ *x i* ≤ $ *?b i*) *x*)

thm interval_conjunct0:

open_interval (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*) = *GSPEC* (λ*GEN%PVAR%664*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%664* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *x i* ∧ $ *x i* < $ *?b i*) *x*)

thm IN_INTERVAL:

(∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*open_interval* (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)) = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *x i* ∧ $ *x i* < $ *?b i*)) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*?a*, *?b*)]) = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *x i* ∧ $ *x i* ≤ $ *?b i*))

thm IN_INTERVAL_conjunct1:

$\forall\, x::(real,\ ?'a::type)\ cart.\ IN\, x\ (closed\_interval\ [(?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart)]) = (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ ?b\ i)$

thm IN_INTERVAL_conjunct0:

$\forall\, x::(real,\ ?'a::type)\ cart.\ IN\, x\ (open\_interval\ (?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart)) = (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i < \$\ x\ i \wedge \$\ x\ i < \$\ ?b\ i)$

thm IN_INTERVAL_REFLECT:

$(\forall\, (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ IN\ (vector\_neg\, x)\ (closed\_interval\ [(vector\_neg\, b,\ vector\_neg\, a)]) = IN\, x\ (closed\_interval\ [(a,\ b)])) \wedge (\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ (vector\_neg\ x)\ (open\_interval\ (vector\_neg\ b,\ vector\_neg\ a)) = IN\ x\ (open\_interval\ (a,\ b)))$

thm IN_INTERVAL_REFLECT_conjunct1:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ (vector\_neg\, x)\ (open\_interval\ (vector\_neg\, b,\ vector\_neg\, a)) = IN\, x\ (open\_interval\ (a,\ b))$

thm IN_INTERVAL_REFLECT_conjunct0:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ (vector\_neg\, x)\ (closed\_interval\ [(vector\_neg\, b,\ vector\_neg\, a)]) = IN\, x\ (closed\_interval\ [(a,\ b)])$

thm REFLECT_INTERVAL_conjunct1:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IMAGE\ vector\_neg\ (open\_interval\ (a,\ b)) = open\_interval\ (vector\_neg\ b,\ vector\_neg\ a)$

thm REFLECT_INTERVAL_conjunct0:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IMAGE\ vector\_neg\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(vector\_neg\ b,\ vector\_neg\ a)]$

thm REFLECT_INTERVAL:

$(\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IMAGE\ vector\_neg\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(vector\_neg\, b,\ vector\_neg\, a)]) \wedge (\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IMAGE\ vector\_neg\ (open\_interval\ (a,\ b)) = open\_interval\ (vector\_neg\ b,\ vector\_neg\ a))$

thm INTERVAL_EQ_EMPTY:

$(closed\_interval\ [(?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart)] = EMPTY) = (\exists\, i \geq 1::nat.\ i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ ?b\ i < \$\ ?a\ i) \wedge (open\_interval\ (?a,\ ?b) = EMPTY) = (\exists\, i \geq 1::nat.\ i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ ?b\ i \leq \$\ ?a\ i)$

thm INTERVAL_EQ_EMPTY_conjunct1:

$(open\_interval\ (?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart) = EMPTY) = (\exists\, i \geq 1::nat.\ i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ ?b\ i \leq \$\ ?a\ i)$

thm INTERVAL_EQ_EMPTY_conjunct0:

$(closed\_interval\ [(?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart)] = EMPTY) = (\exists\, i \geq 1::nat.\ i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ ?b\ i < \$\ ?a\ i)$

thm INTERVAL_NE_EMPTY:

$(closed\_interval\ [(?a::(real,\ ?'a::type)\ cart,\ ?b::(real,\ ?'a::type)\ cart)] \neq EMPTY) = (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i \leq \$\ ?b\ i) \wedge (open\_interval\ (?a,\ ?b) \neq EMPTY) = (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i < \$\ ?b\ i)$

thm SUBSET_INTERVAL_IMP:

$((\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (?a::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?c::(real,\ ?'a::type)\ cart)\ i \wedge \$\ (?d::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?b::(real,\ ?'a::type)\ cart)\ i) \longrightarrow SUBSET\ (closed\_interval\ [(?c,\ ?d)])\ (closed\_interval\ [(?a,\ ?b)])) \wedge ((\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i < \$\ ?c\ i \wedge \$\ ?d\ i < \$\ ?b\ i) \longrightarrow SUBSET\ (closed\_interval\ [(?c,\ ?d)])\ (open\_interval\ (?a,\ ?b))) \wedge ((\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i \leq \$\ ?c\ i \wedge \$\ ?d\ i \leq \$\ ?b\ i) \longrightarrow SUBSET\ (open\_interval\ (?c,\ ?d))\ (closed\_interval\ [(?a,\ ?b)])) \wedge ((\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ ?a\ i \leq \$\ ?c\ i \wedge \$\ ?d\ i \leq \$\ ?b\ i) \longrightarrow SUBSET\ (open\_interval\ (?c,\ ?d))\ (open\_interval\ (?a,\ ?b)))$

thm INTERVAL_SING:

$closed\_interval\ [(?a::(real,\ ?'a::type)\ cart,\ ?a)] = INSERT\ ?a\ EMPTY \wedge open\_interval\ (?a,\ ?a) = EMPTY$

thm SUBSET_INTERVAL_IMP_conjunct3:

$(\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (?a::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?c::(real,\ ?'a::type)\ cart)\ i \wedge \$\ (?d::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?b::(real,\ ?'a::type)\ cart)\ i) \longrightarrow SUBSET\ (open\_interval\ (?c,\ ?d))\ (open\_interval\ (?a,\ ?b))$

thm SUBSET_INTERVAL_IMP_conjunct2:

$(\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (?a::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?c::(real,\ ?'a::type)\ cart)\ i \wedge \$\ (?d::(real,\ ?'a::type)\ cart)\ i \leq \$\ (?b::(real,\ ?'a::type)\ cart)\ i) \longrightarrow SUBSET\ (open\_interval\ (?c,\ ?d))\ (closed\_interval\ [(?a,\ ?b)])$

thm SUBSET_INTERVAL_IMP_conjunct1:

$(\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (?a::(real,\ ?'a::type)\ cart)\ i < \$\ (?c::(real,\ ?'a::type)\ cart)\ i \wedge \$\ (?d::(real,\ ?'a::type)$

cart) *i* < $ (*?b*::(*real*, *?'a*::*type*) *cart*) *i*) ⟶ *SUBSET* (*closed_interval* [(*?c*, *?d*)]) (*open_interval* (*?a*, *?b*))

thm SUBSET_INTERVAL_IMP_conjunct0:

(∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ (*?a*::(*real*, *?'a*::*type*) *cart*) *i* ≤ $ (*?c*::(*real*, *?'a*::*type*) *cart*) *i* ∧ $ (*?d*::(*real*, *?'a*::*type*) *cart*) *i* ≤ $ (*?b*::(*real*, *?'a*::*type*) *cart*) *i*) ⟶ *SUBSET* (*closed_interval* [(*?c*, *?d*)]) (*closed_interval* [(*?a*, *?b*)])

thm SUBSET_INTERVAL_conjunct3:

SUBSET (*open_interval* (*?c*::(*real*, *?'a*::*type*) *cart*, *?d*::(*real*, *?'a*::*type*) *cart*)) (*open_interval* (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* < $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*))

thm SUBSET_INTERVAL_conjunct2:

SUBSET (*open_interval* (*?c*::(*real*, *?'a*::*type*) *cart*, *?d*::(*real*, *?'a*::*type*) *cart*)) (*closed_interval* [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)]) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* < $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*))

thm SUBSET_INTERVAL_conjunct1:

SUBSET (*closed_interval* [(*?c*::(*real*, *?'a*::*type*) *cart*, *?d*::(*real*, *?'a*::*type*) *cart*)]) (*open_interval* (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* ≤ $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *?c i* ∧ $ *?d i* < $ *?b i*))

thm SUBSET_INTERVAL_conjunct0:

SUBSET (*closed_interval* [(*?c*::(*real*, *?'a*::*type*) *cart*, *?d*::(*real*, *?'a*::*type*) *cart*)]) (*closed_interval* [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)]) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* ≤ $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*))

thm SUBSET_INTERVAL:

SUBSET (*closed_interval* [(*?c*::(*real*, *?'a*::*type*) *cart*, *?d*::(*real*, *?'a*::*type*) *cart*)]) (*closed_interval* [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)]) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* ≤ $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*)) ∧ *SUBSET* (*closed_interval* [(*?c*, *?d*)]) (*open_interval* (*?a*, *?b*)) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* ≤ $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *?c i* ∧ $ *?d i* < $ *?b i*)) ∧ *SUBSET* (*open_interval* (*?c*,

?d)) (*closed_interval* [(*?a*, *?b*)]) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* < $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*)) ∧ *SUBSET* (*open_interval* (*?c*, *?d*)) (*open_interval* (*?a*, *?b*)) = ((∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?c i* < $ *?d i*) ⟶ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?c i* ∧ $ *?d i* ≤ $ *?b i*))

thm DISJOINT_INTERVAL:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) = *EMPTY*) = (∃ *i*≥*1*::*nat*. *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ ($ *b i* < $ *a i* ∨ $ *d i* < $ *c i* ∨ $ *b i* < $ *c i* ∨ $ *d i* < $ *a i*)) ∧ (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*open_interval* (*c*, *d*)) = *EMPTY*) = (∃ *i*≥*1*::*nat*. *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ ($ *b i* < $ *a i* ∨ $ *d i* ≤ $ *c i* ∨ $ *b i* ≤ $ *c i* ∨ $ *d i* ≤ $ *a i*)) ∧ (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*closed_interval* [(*c*, *d*)]) = *EMPTY*) = (∃ *i*≥*1*::*nat*. *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ ($ *b i* ≤ $ *a i* ∨ $ *d i* < $ *c i* ∨ $ *b i* ≤ $ *c i* ∨ $ *d i* ≤ $ *a i*)) ∧ (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*)) = *EMPTY*) = (∃ *i*≥*1*::*nat*. *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ ($ *b i* ≤ $ *a i* ∨ $ *d i* ≤ $ *c i* ∨ $ *b i* ≤ $ *c i* ∨ $ *d i* ≤ $ *a i*))

thm INTERVAL_NE_EMPTY_conjunct1:

(*open_interval* (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*) ≠ *EMPTY*) = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* < $ *?b i*)

thm INTERVAL_NE_EMPTY_conjunct0:

(*closed_interval* [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)] ≠ *EMPTY*) = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *?a i* ≤ $ *?b i*)

thm ENDS_IN_INTERVAL:

(∀ (*a*::(*real*, *?'d*::*type*) *cart*) *b*::(*real*, *?'d*::*type*) *cart*. *IN a* (*closed_interval* [(*a*, *b*)]) = (*closed_interval* [(*a*, *b*)] ≠ *EMPTY*)) ∧ (∀ (*a*::(*real*, *?'c*::*type*) *cart*) *b*::(*real*, *?'c*::*type*) *cart*. *IN b* (*closed_interval* [(*a*, *b*)]) = (*closed_interval* [(*a*, *b*)] ≠ *EMPTY*)) ∧ (∀ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. ¬ *IN a* (*open_interval* (*a*, *b*))) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *IN b* (*open_interval* (*a*, *b*)))

thm ENDS_IN_INTERVAL_conjunct3:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *IN b* (*open_interval* (*a*, *b*))

thm ENDS_IN_INTERVAL_conjunct2:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *IN a* (*open_interval* (*a*, *b*))

thm ENDS_IN_INTERVAL_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ IN\ b\ (closed\_interval\ [(a, b)]) = (closed\_interval\ [(a, b)] \neq EMPTY)$

thm ENDS_IN_INTERVAL_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ IN\ a\ (closed\_interval\ [(a, b)]) = (closed\_interval\ [(a, b)] \neq EMPTY)$

thm ENDS_IN_UNIT_INTERVAL:

$IN\ (vec\ (0::nat))\ (closed\_interval\ [(vec\ (0::nat), vec\ (1::nat))]) \wedge IN\ (vec\ (1::nat))\ (closed\_interval\ [(vec\ (0::nat), vec\ (1::nat))]) \wedge \neg\ IN\ (vec\ (0::nat))\ (open\_interval\ (vec\ (0::nat), vec\ (1::nat))) \wedge \neg\ IN\ (vec\ (1::nat))\ (open\_interval\ (vec\ (0::nat), vec\ (1::nat)))$

thm INTER_INTERVAL:

$HOL\_Light\_Import.INTER\ (closed\_interval\ [(?a::(real, ?'a::type)\ cart, ?b::(real, ?'a::type)\ cart)])\ (closed\_interval\ [(?c::(real, ?'a::type)\ cart, ?d::(real, ?'a::type)\ cart)]) = closed\_interval\ [(lambda\ (\lambda i::nat.\ max\ (\$\ ?a\ i)\ (\$\ ?c\ i)), lambda\ (\lambda i::nat.\ min\ (\$\ ?b\ i)\ (\$\ ?d\ i)))]$

thm INTERVAL_OPEN_SUBSET_CLOSED:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ SUBSET\ (open\_interval\ (a, b))\ (closed\_interval\ [(a, b)])$

thm OPEN_INTERVAL_LEMMA:

$\forall (a::real)\ (b::real)\ x::real.\ a < x \wedge x < b \longrightarrow (\exists d>0::real.\ \forall x'::real.\ |x' - x| < d \longrightarrow a < x' \wedge x' < b)$

thm OPEN_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.open\ (open\_interval\ (a, b))$

thm CLOSED_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ (closed\_interval\ [(a, b)])$

thm INTERIOR_CLOSED_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ interior\ (closed\_interval\ [(a, b)]) = open\_interval\ (a, b)$

thm BOUNDED_CLOSED_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ bounded\ (closed\_interval\ [(a, b)])$

thm BOUNDED_INTERVAL_conjunct0:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *bounded* (*closed_interval* [(*a*, *b*)])

thm BOUNDED_INTERVAL:

($\forall$ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *bounded* (*closed_interval* [(*a*, *b*)])) $\wedge$ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *bounded* (*open_interval* (*a*, *b*)))

thm NOT_INTERVAL_UNIV:

($\forall$ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *closed_interval* [(*a*, *b*)] $\neq$ *HOL_Light_Import.UNIV*) $\wedge$ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *HOL_Light_Import.UNIV*)

thm BOUNDED_INTERVAL_conjunct1:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *bounded* (*open_interval* (*a*, *b*))

thm COMPACT_INTERVAL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *compact* (*closed_interval* [(*a*, *b*)])

thm OPEN_INTERVAL_MIDPOINT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\longrightarrow$ *IN* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) (*vector_add* *a* *b*)) (*open_interval* (*a*, *b*))

thm OPEN_CLOSED_INTERVAL_CONVEX:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *e*::*real*. *IN* *x* (*open_interval* (*a*, *b*)) $\wedge$ *IN* *y* (*closed_interval* [(*a*, *b*)]) $\wedge$ (*0*::*real*) $<$ *e* $\wedge$ *e* $\leq$ (*1*::*real*) $\longrightarrow$ *IN* (*vector_add* (% *e* *x*) (% ((*1*::*real*) $-$ *e*) *y*)) (*open_interval* (*a*, *b*))

thm CLOSURE_OPEN_INTERVAL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\longrightarrow$ *closure* (*open_interval* (*a*, *b*)) = *closed_interval* [(*a*, *b*)]

thm BOUNDED_SUBSET_OPEN_INTERVAL_SYMMETRIC:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded* *s* $\longrightarrow$ ($\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *SUBSET* *s* (*open_interval* (*vector_neg* *a*, *a*)))

thm BOUNDED_SUBSET_OPEN_INTERVAL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded* *s* $\longrightarrow$ ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *SUBSET* *s* (*open_interval* (*a*, *b*)))

thm BOUNDED_SUBSET_CLOSED_INTERVAL_SYMMETRIC:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded* *s* $\longrightarrow$ ($\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *SUBSET* *s* (*closed_interval* [(*vector_neg* *a*, *a*)]))

thm BOUNDED_SUBSET_CLOSED_INTERVAL:

$\forall\, s$::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ (∃ (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *SUBSET s* (*closed_interval* [(*a*, *b*)]))

thm FRONTIER_CLOSED_INTERVAL:

∀ (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *frontier* (*closed_interval* [(*a*, *b*)]) = *DIFF* (*closed_interval* [(*a*, *b*)]) (*open_interval* (*a*, *b*))

thm FRONTIER_OPEN_INTERVAL:

∀ (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *frontier* (*open_interval* (*a*, *b*)) = (*if open_interval* (*a*, *b*) = *EMPTY then EMPTY else DIFF* (*closed_interval* [(*a*, *b*)]) (*open_interval* (*a*, *b*)))

thm INTER_INTERVAL_MIXED_EQ_EMPTY:

∀ (*a*::(*real*, ?'*a*::*type*) *cart*) (*b*::(*real*, ?'*a*::*type*) *cart*) (*c*::(*real*, ?'*a*::*type*) *cart*) *d*::(*real*, ?'*a*::*type*) *cart*. *open_interval* (*c*, *d*) ≠ *EMPTY* ⟶ (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*closed_interval* [(*c*, *d*)]) = *EMPTY*) = (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*)) = *EMPTY*)

thm INTERVAL_TRANSLATION:

(∀ (*c*::(*real*, ?'*b*::*type*) *cart*) (*a*::(*real*, ?'*b*::*type*) *cart*) *b*::(*real*, ?'*b*::*type*) *cart*. *closed_interval* [(*vector_add c a*, *vector_add c b*)] = *IMAGE* (*vector_add c*) (*closed_interval* [(*a*, *b*)])) ∧ (∀ (*c*::(*real*, ?'*a*::*type*) *cart*) (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *open_interval* (*vector_add c a*, *vector_add c b*) = *IMAGE* (*vector_add c*) (*open_interval* (*a*, *b*)))

thm INTERVAL_TRANSLATION_conjunct1:

∀ (*c*::(*real*, ?'*a*::*type*) *cart*) (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *open_interval* (*vector_add c a*, *vector_add c b*) = *IMAGE* (*vector_add c*) (*open_interval* (*a*, *b*))

thm INTERVAL_TRANSLATION_conjunct0:

∀ (*c*::(*real*, ?'*a*::*type*) *cart*) (*a*::(*real*, ?'*a*::*type*) *cart*) *b*::(*real*, ?'*a*::*type*) *cart*. *closed_interval* [(*vector_add c a*, *vector_add c b*)] = *IMAGE* (*vector_add c*) (*closed_interval* [(*a*, *b*)])

thm EMPTY_AS_INTERVAL:

EMPTY = *closed_interval* [(*vec* (*1*::*nat*), *vec* (*0*::*nat*))]

thm UNIT_INTERVAL_NONEMPTY:

closed_interval [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))] ≠ *EMPTY* ∧ *open_interval* (*vec* (*0*::*nat*), *vec* (*1*::*nat*)) ≠ *EMPTY*

thm REAL_MIN_ACI_conjunct4:

min (?*x*::*real*) (*min* ?*x* (?*y*::*real*)) = *min* ?*x* ?*y*

thm REAL_MIN_ACI_conjunct3:

$min\ (?x::real)\ ?x = ?x$

thm REAL_MIN_ACI_conjunct2:

$min\ (?x::real)\ (min\ (?y::real)\ (?z::real)) = min\ ?y\ (min\ ?x\ ?z)$

thm REAL_MIN_ACI_conjunct1:

$min\ (min\ (?x::real)\ (?y::real))\ (?z::real) = min\ ?x\ (min\ ?y\ ?z)$

thm REAL_MAX_ACI_conjunct4:

$max\ (?x::real)\ (max\ ?x\ (?y::real)) = max\ ?x\ ?y$

thm REAL_MAX_ACI_conjunct3:

$max\ (?x::real)\ ?x = ?x$

thm REAL_MAX_ACI_conjunct2:

$max\ (?x::real)\ (max\ (?y::real)\ (?z::real)) = max\ ?y\ (max\ ?x\ ?z)$

thm REAL_MAX_ACI_conjunct1:

$max\ (max\ (?x::real)\ (?y::real))\ (?z::real) = max\ ?x\ (max\ ?y\ ?z)$

thm IMAGE_STRETCH_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ m::nat \Rightarrow real.\ IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda k::nat.\ m\ k * \$\ x\ k))\ (closed\_interval\ [(a,\ b)]) = (if\ closed\_interval\ [(a,\ b)] = EMPTY\ then\ EMPTY\ else\ closed\_interval\ [(lambda\ (\lambda k::nat.\ min\ (m\ k * \$\ a\ k)\ (m\ k * \$\ b\ k)),\ lambda\ (\lambda k::nat.\ max\ (m\ k * \$\ a\ k)\ (m\ k * \$\ b\ k)))])$

thm INTERVAL_IMAGE_STRETCH_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ m::nat \Rightarrow real.\ \exists (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda k::nat.\ m\ k * \$\ x\ k))\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(u,\ v)]$

thm UNIT_INTERVAL_NONEMPTY_conjunct1:

$open\_interval\ (vec\ (0::nat),\ vec\ (1::nat)) \neq EMPTY$

thm UNIT_INTERVAL_NONEMPTY_conjunct0:

$closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))] \neq EMPTY$

thm CLOSED_INTERVAL_IMAGE_UNIT_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow closed\_interval\ [(a,\ b)] = IMAGE\ (vector\_add\ a)\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ (\$\ b\ i - \$\ a\ i) * \$\ x\ i))\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]))$

thm SUMS_INTERVALS:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge closed\_interval\ [(c,\ d)] \neq EMPTY \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%666::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%666\ (IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(c,\ d)]))\ (vector\_add\ x\ y)) = closed\_interval\ [(vector\_add\ a\ c,\ vector\_add\ b\ d)]$

thm PASTECART_INTERVAL:

$\forall (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%667::(real,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart.\ \exists (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%667\ (IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(c,\ d)]))\ (pastecart\ x\ y)) = closed\_interval\ [(pastecart\ a\ c,\ pastecart\ b\ d)]$

thm OPEN_CONTAINS_OPEN_INTERVAL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \wedge SUBSET\ (open\_interval\ (a,\ b))\ s))$

thm OPEN_CONTAINS_INTERVAL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s = (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ s))$

thm INTERVAL_CASES_1:

$\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(?a::(real,\ unit)\ cart,\ ?b::(real,\ unit)\ cart)]) \longrightarrow IN\ x\ (open\_interval\ (?a,\ ?b)) \vee x = ?a \vee x = ?b$

thm IN_INTERVAL_1:

$\forall (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) = (HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ x \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ b) \wedge IN\ x\ (open\_interval\ (a,\ b)) = (HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ x \wedge HOL\_Light\_Import.drop\ x < HOL\_Light\_Import.drop\ b)$

thm INTERVAL_EQ_EMPTY_1:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (closed\_interval\ [(a,\ b)] = EMPTY) = (HOL\_Light\_Import.drop\ b < HOL\_Light\_Import.drop\ a) \wedge (open\_interval\ (a,\ b) = EMPTY) = (HOL\_Light\_Import.drop\ b \leq HOL\_Light\_Import.drop\ a)$

thm INTERVAL_NE_EMPTY_1:

$(\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (closed\_interval\ [(a,\ b)] \neq EMPTY) = (HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b)) \wedge (\forall (a::(real,$

unit) *cart*) *b*::(*real*, *unit*) *cart*. (*open_interval* (*a*, *b*) ≠ *EMPTY*) = (*HOL_Light_Import.drop* *a* < *HOL_Light_Import.drop* *b*))

thm SUBSET_INTERVAL_1:

∀ (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*c*::(*real*, *unit*) *cart*) *d*::(*real*, *unit*) *cart*. *SUBSET* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) = (*HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *c* ≤ *HOL_Light_Import.drop* *a* ∧ *HOL_Light_Import.drop* *a* ≤ *HOL_Light_Import.drop* *b* ∧ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *d*) ∧ *SUBSET* (*closed_interval* [(*a*, *b*)]) (*open_interval* (*c*, *d*)) = (*HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *c* < *HOL_Light_Import.drop* *a* ∧ *HOL_Light_Import.drop* *a* ≤ *HOL_Light_Import.drop* *b* ∧ *HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *d*) ∧ *SUBSET* (*open_interval* (*a*, *b*)) (*closed_interval* [(*c*, *d*)]) = (*HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *c* ≤ *HOL_Light_Import.drop* *a* ∧ *HOL_Light_Import.drop* *a* < *HOL_Light_Import.drop* *b* ∧ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *d*) ∧ *SUBSET* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*)) = (*HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *c* ≤ *HOL_Light_Import.drop* *a* ∧ *HOL_Light_Import.drop* *a* < *HOL_Light_Import.drop* *b* ∧ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *d*)

thm EQ_INTERVAL_1:

∀ (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*c*::(*real*, *unit*) *cart*) *d*::(*real*, *unit*) *cart*. (*closed_interval* [(*a*, *b*)] = *closed_interval* [(*c*, *d*)]) = (*HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *a* ∧ *HOL_Light_Import.drop* *d* < *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *a* = *HOL_Light_Import.drop* *c* ∧ *HOL_Light_Import.drop* *b* = *HOL_Light_Import.drop* *d*)

thm DISJOINT_INTERVAL_1:

∀ (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*c*::(*real*, *unit*) *cart*) *d*::(*real*, *unit*) *cart*. (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) = *EMPTY*) = (*HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *d* < *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *d* < *HOL_Light_Import.drop* *a*) ∧ (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*open_interval* (*c*, *d*)) = *EMPTY*) = (*HOL_Light_Import.drop* *b* < *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *d* ≤ *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *d* ≤ *HOL_Light_Import.drop* *a*) ∧ (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*closed_interval* [(*c*, *d*)]) = *EMPTY*) = (*HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *d* < *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *d* ≤ *HOL_Light_Import.drop* *a*) ∧ (*HOL_Light_Import.INTER* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*)) = *EMPTY*) = (*HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *a* ∨ *HOL_Light_Import.drop* *d* ≤ *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *b* ≤ *HOL_Light_Import.drop* *c* ∨ *HOL_Light_Import.drop* *d* ≤ *HOL_Light_Import.drop* *a*)

thm OPEN_CLOSED_INTERVAL_1:

$\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *open_interval* (*a*, *b*) = *DIFF* (*closed_interval* [(*a*, *b*)]) (*INSERT a* (*INSERT b EMPTY*))

thm CLOSED_OPEN_INTERVAL_1:

$\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *HOL_Light_Import.drop a* $\leq$ *HOL_Light_Import.drop b* $\longrightarrow$ *closed_interval* [(*a*, *b*)] = *HOL_Light_Import.UNION* (*open_interval* (*a*, *b*)) (*INSERT a* (*INSERT b EMPTY*))

thm BALL_1:

$\forall$ (*x*::(*real*, *unit*) *cart*) *r*::*real*. *cball* (*x*, *r*) = *closed_interval* [(*vector_sub x* (*lift r*), *vector_add x* (*lift r*))] $\wedge$ *ball* (*x*, *r*) = *open_interval* (*vector_sub x* (*lift r*), *vector_add x* (*lift r*))

thm FINITE_INTERVAL_1_conjunct0:

$\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *FINITE* (*closed_interval* [(*a*, *b*)]) = (*HOL_Light_Import.drop b* $\leq$ *HOL_Light_Import.drop a*)

thm FINITE_INTERVAL_1:

($\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *FINITE* (*closed_interval* [(*a*, *b*)]) = (*HOL_Light_Import.drop b* $\leq$ *HOL_Light_Import.drop a*)) $\wedge$ ($\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *FINITE* (*open_interval* (*a*, *b*)) = (*HOL_Light_Import.drop b* $\leq$ *HOL_Light_Import.drop a*))

thm BALL_INTERVAL:

$\forall$ (*x*::(*real*, *unit*) *cart*) *e*::*real*. *ball* (*x*, *e*) = *open_interval* (*vector_sub x* (*lift e*), *vector_add x* (*lift e*))

thm CBALL_INTERVAL:

$\forall$ (*x*::(*real*, *unit*) *cart*) *e*::*real*. *cball* (*x*, *e*) = *closed_interval* [(*vector_sub x* (*lift e*), *vector_add x* (*lift e*))]

thm BALL_INTERVAL_0:

$\forall$ *e*::*real*. *ball* (*vec* (*0*::*nat*), *e*) = *open_interval* (*vector_neg* (*lift e*), *lift e*)

thm CBALL_INTERVAL_0:

$\forall$ *e*::*real*. *cball* (*vec* (*0*::*nat*), *e*) = *closed_interval* [(*vector_neg* (*lift e*), *lift e*)]

thm INTER_INTERVAL_1:

$\forall$ (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*c*::(*real*, *unit*) *cart*) *d*::(*real*, *unit*) *cart*. *HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) = *closed_interval* [(*lift* (*max* (*HOL_Light_Import.drop a*) (*HOL_Light_Import.drop c*)), *lift* (*min* (*HOL_Light_Import.drop b*) (*HOL_Light_Import.drop d*)))]

thm DEF_is_interval:

$is\_interval = (\lambda\_225285::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ a\ \_225285 \wedge IN\ b\ \_225285 \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ b\ i \vee \$\ b\ i \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ a\ i) \longrightarrow IN\ x\ \_225285)$

thm is_interval:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ is\_interval\ s = (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge IN\ b\ s \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ b\ i \vee \$\ b\ i \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ a\ i) \longrightarrow IN\ x\ s)$

thm IS_INTERVAL_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ is\_interval\ (open\_interval\ (a,\ b)) \wedge is\_interval\ (closed\_interval\ [(a,\ b)])$

thm IS_INTERVAL_EMPTY:

$is\_interval\ EMPTY$

thm IS_INTERVAL_UNIV:

$is\_interval\ HOL\_Light\_Import.UNIV$

thm IS_INTERVAL_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ is\_interval\ (IMAGE\ (vector\_add\ a)\ s) = is\_interval\ s$

thm IS_INTERVAL_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ is\_interval\ s \longrightarrow is\_interval\ (IMAGE\ (vector\_add\ a)\ s)$

thm IS_INTERVAL_POINTWISE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ is\_interval\ s \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists a::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge \$\ a\ i = \$\ x\ i)) \longrightarrow IN\ x\ s$

thm IS_INTERVAL_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (is\_interval\ s \wedge compact\ s) = (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ s = closed\_interval\ [(a,\ b)])$

thm IS_INTERVAL_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ is\_interval\ s = (\forall (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ x::(real,\ unit)\ cart.\ IN\ a\ s \wedge IN\ b\ s \wedge HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ x \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ b \longrightarrow IN\ x\ s)$

thm IS_INTERVAL_1_CASES:

$\forall s$::(*real*, *unit*) *cart* ⇒ *bool*. *is_interval s* = (*s* = *EMPTY* ∨ *s* = *HOL_Light_Import*.*UNIV* ∨ (∃ *a*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%675*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%675* (*a* < *HOL_Light_Import*.*drop x*) *x*)) ∨ (∃ *a*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%676*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%676* (*a* ≤ *HOL_Light_Import*.*drop x*) *x*)) ∨ (∃ *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%677*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%677* (*HOL_Light_Import*.*drop x* ≤ *b*) *x*)) ∨ (∃ *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%678*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%678* (*HOL_Light_Import*.*drop x* < *b*) *x*)) ∨ (∃ (*a*::*real*) *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%679*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%679* (*a* < *HOL_Light_Import*.*drop x* ∧ *HOL_Light_Import*.*drop x* < *b*) *x*)) ∨ (∃ (*a*::*real*) *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%680*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%680* (*a* < *HOL_Light_Import*.*drop x* ∧ *HOL_Light_Import*.*drop x* ≤ *b*) *x*)) ∨ (∃ (*a*::*real*) *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%681*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%681* (*a* ≤ *HOL_Light_Import*.*drop x* ∧ *HOL_Light_Import*.*drop x* < *b*) *x*)) ∨ (∃ (*a*::*real*) *b*::*real*. *s* = *GSPEC* (λ*GEN%PVAR%682*::(*real*, *unit*) *cart*. ∃ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%682* (*a* ≤ *HOL_Light_Import*.*drop x* ∧ *HOL_Light_Import*.*drop x* ≤ *b*) *x*)))

thm DEF_closed_segment:

closed_segment = (*SOME closed_segment*::*nat* ⇒ ((*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*) *list* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∀ (*_227360*::*nat*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *closed_segment _227360* [(*a*, *b*)] = *GSPEC* (λ*GEN%PVAR%683*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%683* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (*vector_add* (% ((*1*::*real*) − *u*) *a*) (% *u b*)))) (*51*::*nat*)

thm closed_segment:

closed_segment [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)] = *GSPEC* (λ*GEN%PVAR%683*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%683* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (*vector_add* (% ((*1*::*real*) − *u*) *?a*) (% *u ?b*)))

thm DEF_open_segment:

open_segment = (λ*_227361*::(*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*. *DIFF* (*closed_segment* [(*fst _227361*, *snd _227361*)]) (*INSERT* (*fst _227361*) (*INSERT* (*snd _227361*) *EMPTY*)))

thm open_segment:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *b*) = *DIFF* (*closed_segment* [(*a*, *b*)]) (*INSERT a* (*INSERT b EMPTY*))

thm OPEN_SEGMENT_ALT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* ≠ *b* ⟶ *open_segment* (*a*, *b*) = *GSPEC* (λ*GEN%PVAR%684*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC*

GEN%PVAR%684 ((*0*::*real*) < *u* ∧ *u* < (*1*::*real*)) (*vector_add* (% ((*1*::*real*) − *u*) *a*) (% *u* *b*)))

thm segment_conjunct1:

open_segment (*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*) = *DIFF* (*closed_segment* [(*?a*, *?b*)]) (*INSERT* *?a* (*INSERT* *?b* *EMPTY*))

thm segment:

closed_segment [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)] = *GSPEC* (λ*GEN%PVAR%685*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC* *GEN%PVAR%685* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (*vector_add* (% ((*1*::*real*) − *u*) *?a*) (% *u* *?b*))) ∧ *open_segment* (*?a*, *?b*) = *DIFF* (*closed_segment* [(*?a*, *?b*)]) (*INSERT* *?a* (*INSERT* *?b* *EMPTY*))

thm segment_conjunct0:

closed_segment [(*?a*::(*real*, *?'a*::*type*) *cart*, *?b*::(*real*, *?'a*::*type*) *cart*)] = *GSPEC* (λ*GEN%PVAR%685*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC* *GEN%PVAR%685* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (*vector_add* (% ((*1*::*real*) − *u*) *?a*) (% *u* *?b*)))

thm SEGMENT_REFL:

(∀ *a*::(*real*, *?'b*::*type*) *cart*. *closed_segment* [(*a*, *a*)] = *INSERT* *a* *EMPTY*) ∧ (∀ *a*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *a*) = *EMPTY*)

thm SEGMENT_REFL_conjunct1:

∀ *a*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *a*) = *EMPTY*

thm SEGMENT_REFL_conjunct0:

∀ *a*::(*real*, *?'a*::*type*) *cart*. *closed_segment* [(*a*, *a*)] = *INSERT* *a* *EMPTY*

thm IN_SEGMENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *IN* *x* (*closed_segment* [(*a*, *b*)]) = (∃ *u*≥*0*::*real*. *u* ≤ (*1*::*real*) ∧ *x* = *vector_add* (% ((*1*::*real*) − *u*) *a*) (% *u* *b*)) ∧ *IN* *x* (*open_segment* (*a*, *b*)) = (*a* ≠ *b* ∧ (∃ *u*>*0*::*real*. *u* < (*1*::*real*) ∧ *x* = *vector_add* (% ((*1*::*real*) − *u*) *a*) (% *u* *b*)))

thm SEGMENT_SYM_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *closed_segment* [(*a*, *b*)] = *closed_segment* [(*b*, *a*)]

thm SEGMENT_SYM:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *closed_segment* [(*a*, *b*)] = *closed_segment* [(*b*, *a*)]) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *b*) = *open_segment* (*b*, *a*))

thm ENDS_IN_SEGMENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN* *a* (*closed_segment* [(*a*, *b*)]) ∧ *IN* *b* (*closed_segment* [(*a*, *b*)])

thm ENDS_NOT_IN_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ \neg\ IN\ a\ (open\_segment\ (a,\ b))\ \wedge\ \neg\ IN\ b\ (open\_segment\ (a,\ b))$

thm SEGMENT_CLOSED_OPEN:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ closed\_segment\ [(a,\ b)] = HOL\_Light\_Import.UNION\ (open\_segment\ (a,\ b))\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm MIDPOINT_IN_SEGMENT:

$(\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ (midpoint\ (a,\ b))\ (closed\_segment\ [(a,\ b)]))\ \wedge\ (\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ (midpoint\ (a,\ b))\ (open\_segment\ (a,\ b)) = (a \neq b))$

thm BETWEEN_IN_SEGMENT:

$\forall (x::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ between\ x\ (a,\ b) = IN\ x\ (closed\_segment\ [(a,\ b)])$

thm SEGMENT_1_conjunct1:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ open\_segment\ (a,\ b) = (if\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b\ then\ open\_interval\ (a,\ b)\ else\ open\_interval\ (b,\ a))$

thm SEGMENT_1_conjunct0:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_segment\ [(a,\ b)] = (if\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b\ then\ closed\_interval\ [(a,\ b)]\ else\ closed\_interval\ [(b,\ a)])$

thm SEGMENT_1:

$(\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_segment\ [(a,\ b)] = (if\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b\ then\ closed\_interval\ [(a,\ b)]\ else\ closed\_interval\ [(b,\ a)]))\ \wedge\ (\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ open\_segment\ (a,\ b) = (if\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b\ then\ open\_interval\ (a,\ b)\ else\ open\_interval\ (b,\ a)))$

thm OPEN_SEGMENT_1:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.open\ (open\_segment\ (a,\ b))$

thm SEGMENT_TRANSLATION:

$(\forall (c::(real,\ ?'b::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ closed\_segment\ [(vector\_add\ c\ a,\ vector\_add\ c\ b)] = IMAGE\ (vector\_add\ c)\ (closed\_segment\ [(a,\ b)]))\ \wedge\ (\forall (c::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ open\_segment\ (vector\_add\ c\ a,\ vector\_add\ c\ b) = IMAGE\ (vector\_add\ c)\ (open\_segment\ (a,\ b)))$

thm SEGMENT_TRANSLATION_conjunct1:

$\forall (c::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.$ $open\_segment\ (vector\_add\ c\ a,\ vector\_add\ c\ b) = IMAGE\ (vector\_add\ c)\ (open\_segment\ (a,\ b))$

thm SEGMENT_TRANSLATION_conjunct0:

$\forall (c::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.$ $closed\_segment\ [(vector\_add\ c\ a,\ vector\_add\ c\ b)] = IMAGE\ (vector\_add\ c)\ (closed\_segment\ [(a,\ b)])$

thm CLOSED_SEGMENT_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow closed\_segment\ [(f\ a,\ f\ b)] = IMAGE\ f\ (closed\_segment\ [(a,\ b)])$

thm OPEN_SEGMENT_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow open\_segment\ (f\ a, f\ b) = IMAGE\ f\ (open\_segment\ (a,\ b))$

thm IN_OPEN_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $IN\ x\ (open\_segment\ (a,\ b)) = (IN\ x\ (closed\_segment\ [(a,\ b)]) \wedge x \neq a \wedge x \neq b)$

thm IN_OPEN_SEGMENT_ALT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $IN\ x\ (open\_segment\ (a,\ b)) = (IN\ x\ (closed\_segment\ [(a,\ b)]) \wedge x \neq a \wedge x \neq b \wedge a \neq b)$

thm COLLINEAR_DIST_IN_CLOSED_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $collinear\ (INSERT\ x\ (INSERT\ a\ (INSERT\ b\ EMPTY))) \wedge distance\ (x,\ a) \leq distance\ (a,\ b) \wedge distance\ (x,\ b) \leq distance\ (a,\ b) \longrightarrow IN\ x\ (closed\_segment\ [(a,\ b)])$

thm COLLINEAR_DIST_IN_OPEN_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.$ $collinear\ (INSERT\ x\ (INSERT\ a\ (INSERT\ b\ EMPTY))) \wedge distance\ (x,\ a) < distance\ (a,\ b) \wedge distance\ (x,\ b) < distance\ (a,\ b) \longrightarrow IN\ x\ (open\_segment\ (a,\ b))$

thm SEGMENT_SCALAR_MULTIPLE_conjunct0:

$\forall (a::real)\ (b::real)\ v::(real,\ ?'a::type)\ cart.\ closed\_segment\ [(\%\ a\ v,\ \%\ b\ v)] = GSPEC\ (\lambda GEN\%PVAR\%690::(real,\ ?'a::type)\ cart.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%690\ (a \leq x \wedge x \leq b \vee b \leq x \wedge x \leq a)\ (\%\ x\ v))$

thm SEGMENT_SCALAR_MULTIPLE:

$(\forall (a::real)\ (b::real)\ v::(real,\ ?'a::type)\ cart.\ closed\_segment\ [(\%\ a\ v,\ \%\ b\ v)] = GSPEC\ (\lambda GEN\%PVAR\%690::(real,\ ?'a::type)\ cart.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%690\ (a \leq x \wedge x \leq b \vee b \leq x \wedge x \leq a)\ (\%\ x\ v))) \wedge (\forall (a::real)\ (b::real)\ v::(real,\ ?'a::type)\ cart.\ v \neq vec\ (0::nat) \longrightarrow open\_segment\ (\%\ a\ v,\ \%\ b\ v) = GSPEC\ (\lambda GEN\%PVAR\%691::(real,\ ?'a::type)\ cart.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%691\ (a < x \wedge x < b \vee b < x \wedge x < a)\ (\%\ x\ v)))$

thm SEGMENT_SCALAR_MULTIPLE_conjunct1:

$\forall (a::real)\ (b::real)\ v::(real,\ ?'a::type)\ cart.\ v \neq vec\ (0::nat) \longrightarrow open\_segment\ (\%\ a\ v,\ \%\ b\ v) = GSPEC\ (\lambda GEN\%PVAR\%691::(real,\ ?'a::type)\ cart.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%691\ (a < x \wedge x < b \vee b < x \wedge x < a)\ (\%\ x\ v))$

thm FINITE_INTER_COLLINEAR_OPEN_SEGMENTS:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \longrightarrow FINITE\ (HOL\_Light\_Import.INTER\ (open\_segment\ (a,\ b))\ (open\_segment\ (c,\ d))) = (HOL\_Light\_Import.INTER\ (open\_segment\ (a,\ b))\ (open\_segment\ (c,\ d)) = EMPTY)$

thm DIST_IN_OPEN_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ (open\_segment\ (a,\ b)) \longrightarrow distance\ (x,\ a) < distance\ (a,\ b) \wedge distance\ (x,\ b) < distance\ (a,\ b)$

thm DIST_IN_CLOSED_SEGMENT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closed\_segment\ [(a,\ b)]) \longrightarrow distance\ (x,\ a) \leq distance\ (a,\ b) \wedge distance\ (x,\ b) \leq distance\ (a,\ b)$

thm LIM_COMPONENT_UBOUND:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (b::real)\ k::nat.\ \neg\ trivial\_limit\ net \wedge -->\ f\ l\ net \wedge eventually\ (\lambda x::?'b::type.\ \$\ (f\ x)\ k \leq b)\ net \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ l\ k \leq b$

thm LIM_COMPONENT_LBOUND:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (b::real)\ k::nat.\ \neg\ trivial\_limit\ net \wedge -->\ f\ l\ net \wedge eventually\ (\lambda x::?'b::type.\ b \leq \$\ (f\ x)\ k)\ net \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow b \leq \$\ l\ k$

thm LIM_COMPONENT_EQ:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ (i::nat)\ (l::(real,\ ?'a::type)\ cart)\ b::real.\ -->\ f\ l\ net \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV$

∧ ¬ trivial_limit net ∧ eventually (λx::?'b::type. $ (f x) i = b) net ⟶ $ l i = b

thm LIM_COMPONENT_LE:

∀ (net::?'b::type net) (f::?'b::type ⇒ (real, ?'a::type) cart) (g::?'b::type ⇒ (real, ?'a::type) cart) (k::nat) (l::(real, ?'a::type) cart) m::(real, ?'a::type) cart. ¬ trivial_limit net ∧ --> f l net ∧ --> g m net ∧ eventually (λx::?'b::type. $ (f x) k ≤ $ (g x) k) net ∧ (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ $ l k ≤ $ m k

thm LIM_DROP_LE:

∀ (net::?'a::type net) (f::?'a::type ⇒ (real, unit) cart) (g::?'a::type ⇒ (real, unit) cart) (l::(real, unit) cart) m::(real, unit) cart. ¬ trivial_limit net ∧ --> f l net ∧ --> g m net ∧ eventually (λx::?'a::type. HOL_Light_Import.drop (f x) ≤ HOL_Light_Import.drop (g x)) net ⟶ HOL_Light_Import.drop l ≤ HOL_Light_Import.drop m

thm LIM_DROP_UBOUND:

∀ (net::?'a::type net) (f::?'a::type ⇒ (real, unit) cart) (l::(real, unit) cart) b::real. --> f l net ∧ ¬ trivial_limit net ∧ eventually (λx::?'a::type. HOL_Light_Import.drop (f x) ≤ b) net ⟶ HOL_Light_Import.drop l ≤ b

thm LIM_DROP_LBOUND:

∀ (net::?'a::type net) (f::?'a::type ⇒ (real, unit) cart) (l::(real, unit) cart) b::real. --> f l net ∧ ¬ trivial_limit net ∧ eventually (λx::?'a::type. b ≤ HOL_Light_Import.drop (f x)) net ⟶ b ≤ HOL_Light_Import.drop l

thm IMAGE_CLOSURE_SUBSET:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. continuous_on f (closure s) ∧ HOL_Light_Import.closed t ∧ SUBSET (IMAGE f s) t ⟶ SUBSET (IMAGE f (closure s)) t

thm CONTINUOUS_ON_CLOSURE_NORM_LE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (x::(real, ?'b::type) cart) b::real. continuous_on f (closure s) ∧ (∀ y::(real, ?'b::type) cart. IN y s ⟶ vector_norm (f y) ≤ b) ∧ IN x (closure s) ⟶ vector_norm (f x) ≤ b

thm CONTINUOUS_ON_CLOSURE_COMPONENT_LE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (x::(real, ?'b::type) cart) (b::real) k::nat. continuous_on f (closure s) ∧ (∀ y::(real, ?'b::type) cart. IN y s ⟶ $ (f y) k ≤ b) ∧ IN x (closure s) ⟶ $ (f x) k ≤ b

thm CONTINUOUS_ON_CLOSURE_COMPONENT_GE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (x::(real, ?'b::type) cart) (b::real) k::nat. continuous_on f (closure s) ∧

$(\forall y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \longrightarrow b \leq \$\ (f\ y)\ k) \land IN\ x\ (closure\ s) \longrightarrow b \leq \$\ (f\ x)\ k$

thm LIM_WITHIN_UNION:

$--> (?f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (?l::(real,\ ?'a::type)\ cart)\ (within\ (at\ (?x::(real,\ ?'b::type)\ cart))\ (HOL\_Light\_Import.UNION\ (?s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (?t::(real,\ ?'b::type)\ cart \Rightarrow bool))) = (--> ?f\ ?l\ (within\ (at\ ?x)\ ?s) \land --> ?f\ ?l\ (within\ (at\ ?x)\ ?t))$

thm CONTINUOUS_ON_UNION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \land HOL\_Light\_Import.closed\ t \land continuous\_on\ f\ s \land continuous\_on\ f\ t \longrightarrow continuous\_on\ f\ (HOL\_Light\_Import.UNION\ s\ t)$

thm CONTINUOUS_ON_CASES:

$\forall (P::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \land HOL\_Light\_Import.closed\ t \land continuous\_on\ f\ s \land continuous\_on\ g\ t \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \land \neg\ P\ x \lor IN\ x\ t \land P\ x \longrightarrow f\ x = g\ x) \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ if\ P\ x\ then\ f\ x\ else\ g\ x)\ (HOL\_Light\_Import.UNION\ s\ t)$

thm CLOSED_IN_LIMPT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ t)\ s = (SUBSET\ s\ t \land (\forall x::(real,\ ?'a::type)\ cart.\ limit\_point\_of\ x\ s \land IN\ x\ t \longrightarrow IN\ x\ s))$

thm CONTINUOUS_ON_UNION_LOCAL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ (?t::(real,\ ?'b::type)\ cart \Rightarrow bool)))\ s \land closed\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ ?t))\ ?t \land continuous\_on\ f\ s \land continuous\_on\ f\ ?t \longrightarrow continuous\_on\ f\ (HOL\_Light\_Import.UNION\ s\ ?t)$

thm CONTINUOUS_ON_CASES_LOCAL:

$\forall (P::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ closed\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ t))\ s \land closed\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ t))\ t \land continuous\_on\ f\ s \land continuous\_on\ g\ t \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \land \neg\ P\ x \lor IN\ x\ t \land P\ x \longrightarrow f\ x = g\ x) \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ if\ P\ x\ then\ f\ x\ else\ g\ x)\ (HOL\_Light\_Import.UNION\ s\ t)$

thm CONTINUOUS_ON_CASES_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow real)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ a::real.\ continuous\_on\ f\ (GSPEC\ (\lambda GEN\%PVAR\%713::(real,\ ?'b::type)\ cart.\ \exists\, t::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%713\ (IN\ t\ s \land h\ t \le a)\ t)) \land continuous\_on\ g\ (GSPEC\ (\lambda GEN\%PVAR\%714::(real,\ ?'b::type)\ cart.\ \exists\, t::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%714\ (IN\ t\ s \land a \le h\ t)\ t)) \land continuous\_on\ (lift \circ h)\ s \land (\forall\, t::(real,\ ?'b::type)\ cart.\ IN\ t\ s \land h\ t = a \longrightarrow f\ t = g\ t) \longrightarrow continuous\_on\ (\lambda t::(real,\ ?'b::type)\ cart.\ if\ h\ t \le a\ then\ f\ t\ else\ g\ t)\ s$

thm CONTINUOUS_ON_CASES_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)\ a::real.\ continuous\_on\ f\ (GSPEC\ (\lambda GEN\%PVAR\%715::(real,\ unit)\ cart.\ \exists\, t::(real,\ unit)\ cart.\ SETSPEC\ GEN\%PVAR\%715\ (IN\ t\ s \land HOL\_Light\_Import.drop\ t \le a)\ t)) \land continuous\_on\ g\ (GSPEC\ (\lambda GEN\%PVAR\%716::(real,\ unit)\ cart.\ \exists\, t::(real,\ unit)\ cart.\ SETSPEC\ GEN\%PVAR\%716\ (IN\ t\ s \land a \le HOL\_Light\_Import.drop\ t)\ t)) \land (IN\ (lift\ a)\ s \longrightarrow f\ (lift\ a) = g\ (lift\ a)) \longrightarrow continuous\_on\ (\lambda t::(real,\ unit)\ cart.\ if\ HOL\_Light\_Import.drop\ t \le a\ then\ f\ t\ else\ g\ t)\ s$

thm CONTINUOUS_FINITE_RANGE_CONSTANT_EQ:

$\forall\, s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ connected\ s = (\forall\, f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ s \land FINITE\ (IMAGE\ f\ s) \longrightarrow (\exists\, a::(real,\ ?'a::type)\ cart.\ \forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a))$

thm CONTINUOUS_DISCRETE_RANGE_CONSTANT_EQ:

$\forall\, s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ connected\ s = (\forall\, f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ s \land (\forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (\exists\, e>0::real.\ \forall\, y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \land f\ y \ne f\ x \longrightarrow e \le vector\_norm\ (vector\_sub\ (f\ y)\ (f\ x)))) \longrightarrow (\exists\, a::(real,\ ?'a::type)\ cart.\ \forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a))$

thm CONTINUOUS_DISCRETE_RANGE_CONSTANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ connected\ s \land continuous\_on\ f\ s \land (\forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (\exists\, e>0::real.\ \forall\, y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \land f\ y \ne f\ x \longrightarrow e \le vector\_norm\ (vector\_sub\ (f\ y)\ (f\ x)))) \longrightarrow (\exists\, a::(real,\ ?'a::type)\ cart.\ \forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a)$

thm CONTINUOUS_FINITE_RANGE_CONSTANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ connected\ s \land continuous\_on\ f\ s \land FINITE\ (IMAGE\ f\ s) \longrightarrow (\exists\, a::(real,\ ?'a::type)\ cart.\ \forall\, x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = a)$

thm LIM_COMPONENTWISE_LIFT:

$\forall (net::?'b::type\ net)\ f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ -->\ f\ (?l::(real,\ ?'a::type)\ cart)\ net = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow -->\ (\lambda x::?'b::type.\ lift\ (\$\ (f\ x)\ i))\ (lift\ (\$\ ?l\ i))\ net)$

thm CONTINUOUS_COMPONENTWISE_LIFT:

$\forall (net::?'b::type\ net)\ f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\ f\ net = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow continuous\ (\lambda x::?'b::type.\ lift\ (\$\ (f\ x)\ i))\ net)$

thm CONTINUOUS_ON_COMPONENTWISE_LIFT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow continuous\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (\$\ (f\ x)\ i))\ s)$

thm CONNECTED_IVT_HYPERPLANE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::real.\ connected\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge dot\ a\ x \leq b \wedge b \leq dot\ a\ y \longrightarrow (\exists z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \wedge dot\ a\ z = b)$

thm CONNECTED_IVT_COMPONENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (a::real)\ k::nat.\ connected\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ x\ k \leq a \wedge a \leq \$\ y\ k \longrightarrow (\exists z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \wedge \$\ z\ k = a)$

thm BOUNDED_INCREASING_CONVERGENT:

$\forall s::nat \Rightarrow (real,\ unit)\ cart.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%720::(real,\ unit)\ cart.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%720\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))) \wedge (\forall n::nat.\ HOL\_Light\_Import.drop\ (s\ n) \leq HOL\_Light\_Import.drop\ (s\ (Suc\ n))) \longrightarrow (\exists l::(real,\ unit)\ cart.\ -->\ s\ l\ sequentially)$

thm BOUNDED_DECREASING_CONVERGENT:

$\forall s::nat \Rightarrow (real,\ unit)\ cart.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%721::(real,\ unit)\ cart.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%721\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))) \wedge (\forall n::nat.\ HOL\_Light\_Import.drop\ (s\ (Suc\ n)) \leq HOL\_Light\_Import.drop\ (s\ n)) \longrightarrow (\exists l::(real,\ unit)\ cart.\ -->\ s\ l\ sequentially)$

thm DEF_homeomorphism:

$homeomorphism = (\lambda(\_231384::((real,\ ?'b::type)\ cart \Rightarrow bool) \times ((real,\ ?'a::type)\ cart \Rightarrow bool))\ \_231385::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \times ((real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart).\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (fst\ \_231384) \longrightarrow snd\ \_231385\ (fst\ \_231385\ x) = x) \wedge IMAGE\ (fst\ \_231385)\ (fst\ \_231384) = snd\ \_231384 \wedge continuous\_on\ (fst\ \_231385)\ (fst\ \_231384) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ (snd\ \_231384) \longrightarrow fst\ \_231385\ (snd\ \_231385\ y) = y) \wedge IMAGE\ (snd\ \_231385)\ (snd\ \_231384) = fst\ \_231384 \wedge continuous\_on\ (snd\ \_231385)\ (snd\ \_231384))$

thm homeomorphism:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphism\ (s,\ t)\ (f,\ g) = ((\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \wedge IMAGE\ f\ s = t \wedge continuous\_on\ f\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ t \longrightarrow f\ (g\ y) = y) \wedge IMAGE\ g\ t = s \wedge continuous\_on\ g\ t)$

thm DEF_homeomorphic:

$homeomorphic = (\lambda(\_231406::(real,\ ?'b::type)\ cart \Rightarrow bool)\ \_231407::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ homeomorphism\ (\_231406,\ \_231407)\ (f,\ g))$

thm homeomorphic:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t = (\exists (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ homeomorphism\ (s,\ t)\ (f,\ g))$

thm HOMEOMORPHIC_REFL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ s$

thm HOMEOMORPHIC_SYM:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t = homeomorphic\ t\ s$

thm HOMEOMORPHIC_TRANS:

$\forall (s::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \wedge homeomorphic\ t\ u \longrightarrow homeomorphic\ s\ u$

thm HOMEOMORPHIC_IMP_CARD_EQ:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \longrightarrow =\_c\ s\ t$

thm HOMEOMORPHIC_EMPTY:

$(\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ EMPTY = (s = EMPTY)) \wedge (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ homeomorphic\ EMPTY\ s = (s = EMPTY))$

thm HOMEOMORPHIC_MINIMAL:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t = (\exists (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ t \wedge g\ (f\ x) = x) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ t \longrightarrow IN\ (g\ y)\ s \wedge f\ (g\ y) = y) \wedge continuous\_on\ f\ s \wedge continuous\_on\ g\ t)$

thm HOMEOMORPHIC_INJECTIVE_LINEAR_IMAGE_SELF:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *homeomorphic* (*IMAGE f s*) *s*

thm HOMEOMORPHIC_INJECTIVE_LINEAR_IMAGE_LEFT_EQ:

∀ (*f*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'c*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *linear f* ∧ (∀ (*x*::(*real*, *?'c*::*type*) *cart*) *y*::(*real*, *?'c*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *homeomorphic* (*IMAGE f s*) *t* = *homeomorphic s t*

thm HOMEOMORPHIC_INJECTIVE_LINEAR_IMAGE_RIGHT_EQ:

∀ (*f*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'c*::*type*) *cart* ⇒ *bool*. *linear f* ∧ (∀ (*x*::(*real*, *?'c*::*type*) *cart*) *y*::(*real*, *?'c*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *homeomorphic s* (*IMAGE f t*) = *homeomorphic s t*

thm HOMEOMORPHIC_TRANSLATION_SELF:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic* (*IMAGE* (*vector_add a*) *s*) *s*

thm HOMEOMORPHIC_TRANSLATION_LEFT_EQ:

∀ (*a*::(*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic* (*IMAGE* (*vector_add a*) *s*) *t* = *homeomorphic s t*

thm HOMEOMORPHIC_TRANSLATION_RIGHT_EQ:

∀ (*a*::(*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *homeomorphic s* (*IMAGE* (*vector_add a*) *t*) = *homeomorphic s t*

thm HOMEOMORPHISM_COMPACT:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ *continuous_on f s* ∧ *IMAGE f s* = *t* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x s* ∧ *IN y s* ∧ *f x* = *f y* ⟶ *x* = *y*) ⟶ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *homeomorphism* (*s*, *t*) (*f*, *g*))

thm HOMEOMORPHIC_COMPACT:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ *continuous_on f s* ∧ *IMAGE f s* = *t* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x s* ∧ *IN y s* ∧ *f x* = *f y* ⟶ *x* = *y*) ⟶ *homeomorphic s t*

thm HOMEOMORPHIC_COMPACTNESS:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic s t* ⟶ *compact s* = *compact t*

thm HOMEOMORPHIC_CONNECTEDNESS:

∀(*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic s t* ⟶ *connected s* = *connected t*

thm HOMEOMORPHIC_SCALING:

∀(*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *c*::*real*. *c* ≠ (*0*::*real*) ⟶ *homeomorphic s* (*IMAGE* (% *c*) *s*)

thm HOMEOMORPHIC_TRANSLATION:

∀(*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *homeomorphic s* (*IMAGE* (*vector_add a*) *s*)

thm HOMEOMORPHIC_AFFINITY:

∀(*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. *c* ≠ (*0*::*real*) ⟶ *homeomorphic s* (*IMAGE* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vector_add a* (% *c x*)) *s*)

thm HOMEOMORPHIC_CBALLS:

∀(*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*d*::*real*) *e*::*real*. (*0*::*real*) < *d* ∧ (*0*::*real*) < *e* ⟶ *homeomorphic* (*cball* (*a*, *d*)) (*cball* (*b*, *e*))

thm HOMEOMORPHIC_BALLS:

∀(*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*d*::*real*) *e*::*real*. (*0*::*real*) < *d* ∧ (*0*::*real*) < *e* ⟶ *homeomorphic* (*ball* (*a*, *d*)) (*ball* (*b*, *e*))

thm HOMEOMORPHIC_OPEN_INTERVALS_1:

∀(*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*c*::(*real*, *unit*) *cart*) *d*::(*real*, *unit*) *cart*. *HOL_Light_Import.drop a* < *HOL_Light_Import.drop b* ∧ *HOL_Light_Import.drop c* < *HOL_Light_Import.drop d* ⟶ *homeomorphic* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*))

thm HOMEOMORPHIC_OPEN_INTERVAL_UNIV_1:

∀(*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *HOL_Light_Import.drop a* < *HOL_Light_Import.drop b* ⟶ *homeomorphic* (*open_interval* (*a*, *b*)) *HOL_Light_Import.UNIV*

thm HOMEOMORPHIC_OPEN_INTERVALS:

∀(*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. (*open_interval* (*a*, *b*) = *EMPTY*) = (*open_interval* (*c*, *d*) = *EMPTY*) ⟶ *homeomorphic* (*open_interval* (*a*, *b*)) (*open_interval* (*c*, *d*))

thm HOMEOMORPHIC_OPEN_INTERVAL_UNIV:

∀(*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) ≠ *EMPTY* ⟶ *homeomorphic* (*open_interval* (*a*, *b*)) *HOL_Light_Import.UNIV*

thm HOMEOMORPHIC_BALL_UNIV:

∀(*a*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) < *r* ⟶ *homeomorphic* (*ball* (*a*, *r*)) *HOL_Light_Import.UNIV*

thm CAUCHY_ISOMETRIC:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (e::real)\ x::nat \Rightarrow (real, ?'b::type)\ cart.\ (0::real) < e \wedge subspace\ s \wedge linear\ f \wedge (\forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \longrightarrow e * vector\_norm\ x \leq vector\_norm\ (f\ x)) \wedge (\forall n::nat.\ IN\ (x\ n)\ s) \wedge cauchy\ (f \circ x) \longrightarrow cauchy\ x$

thm COMPLETE_ISOMETRIC_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ e::real.\ (0::real) < e \wedge subspace\ s \wedge linear\ f \wedge (\forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \longrightarrow e * vector\_norm\ x \leq vector\_norm\ (f\ x)) \wedge complete\ s \longrightarrow complete\ (IMAGE\ f\ s)$

thm INJECTIVE_IMP_ISOMETRIC:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge subspace\ s \wedge linear\ f \wedge (\forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = vec\ (0::nat) \longrightarrow x = vec\ (0::nat)) \longrightarrow (\exists e>0::real.\ \forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \longrightarrow e * vector\_norm\ x \leq vector\_norm\ (f\ x))$

thm CLOSED_INJECTIVE_IMAGE_SUBSPACE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ subspace\ s \wedge linear\ f \wedge (\forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = vec\ (0::nat) \longrightarrow x = vec\ (0::nat)) \wedge HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ f\ s)$

thm OPEN_SURJECTIVE_LINEAR_IMAGE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow (\forall s::(real, ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ f\ s))$

thm OPEN_BIJECTIVE_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ f\ s) = HOL\_Light\_Import.open\ s$

thm CLOSED_INJECTIVE_LINEAR_IMAGE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\forall s::(real, ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ f\ s))$

thm CLOSED_INJECTIVE_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ f\ s) = HOL\_Light\_Import.closed\ s$

thm CLOSURE_LINEAR_IMAGE_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \longrightarrow SUBSET\ (IMAGE\ f\ (closure\ s))\ (closure\ (IMAGE\ f\ s))$

thm CLOSURE_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow closure\ (IMAGE\ f\ s) = IMAGE\ f\ (closure\ s)$

thm CLOSURE_BOUNDED_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge bounded\ s \longrightarrow closure\ (IMAGE\ f\ s) = IMAGE\ f\ (closure\ s)$

thm LINEAR_INTERIOR_IMAGE_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow SUBSET\ (interior\ (IMAGE\ f\ s))\ (IMAGE\ f\ (interior\ s))$

thm LINEAR_IMAGE_SUBSET_INTERIOR:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \longrightarrow SUBSET\ (IMAGE\ f\ (interior\ s))\ (interior\ (IMAGE\ f\ s))$

thm INTERIOR_BIJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \longrightarrow interior\ (IMAGE\ f\ s) = IMAGE\ f\ (interior\ s)$

thm FRONTIER_BIJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \longrightarrow frontier\ (IMAGE\ f\ s) = IMAGE\ f\ (frontier\ s)$

thm IN_INTERIOR_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ linear\ f \wedge linear\ g \wedge f \circ g = id \wedge IN\ x\ (interior\ s) \longrightarrow IN\ (f\ x)\ (interior\ (IMAGE\ f\ s))$

thm LINEAR_OPEN_MAPPING:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge linear\ g \wedge f \circ g = id \longrightarrow (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ f\ s))$

thm INTERIOR_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow interior\ (IMAGE\ f\ s) = IMAGE\ f\ (interior\ s)$

thm INTERIOR_SURJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ f\ x = y) \longrightarrow interior\ (IMAGE\ f\ s) = IMAGE\ f\ (interior\ s)$

thm CLOSURE_SURJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ f\ x = y) \longrightarrow closure\ (IMAGE\ f\ s) = IMAGE\ f\ (closure\ s)$

thm FRONTIER_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow frontier\ (IMAGE\ f\ s) = IMAGE\ f\ (frontier\ s)$

thm FRONTIER_SURJECTIVE_LINEAR_IMAGE:

$\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ f\ x = y) \longrightarrow frontier\ (IMAGE\ f\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool)) = IMAGE\ f\ (frontier\ ?s)$

thm COMPLETE_INJECTIVE_LINEAR_IMAGE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ complete\ s \longrightarrow complete\ (IMAGE\ f\ s))$

thm COMPLETE_INJECTIVE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow complete\ (IMAGE\ f\ s) = complete\ s$

thm LIMPT_INJECTIVE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow limit\_point\_of\ (f\ (?x::(real,\ ?'b::type)\ cart))\ (IMAGE\ f\ s) = limit\_point\_of\ ?x\ s$

thm LIMPT_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ limit\_point\_of\ (vector\_add\ a\ x)\ (IMAGE\ (vector\_add\ a)\ s) = limit\_point\_of\ x\ s$

thm OPEN_OPEN_PROJECTION:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ (?'b::type,\ ?'a::type)\ finite\_sum)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge HOL\_Light\_Import.open\ t \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%733::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%733\ (IN\ x\ s \wedge (\exists y::(real,\ ?'a::type)\ cart.\ IN\ (pastecart\ x\ y)\ t))\ x))$

thm INTERIOR_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ interior\ (IMAGE\ vector\_neg\ s) = IMAGE\ vector\_neg\ (interior\ s)$

thm SYMMETRIC_INTERIOR:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (vector\_neg\ x)\ s) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (interior\ s) \longrightarrow IN\ (vector\_neg\ x)\ (interior\ s))$

thm CLOSURE_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closure\ (IMAGE\ vector\_neg\ s) = IMAGE\ vector\_neg\ (closure\ s)$

thm SYMMETRIC_CLOSURE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow IN\ (vector\_neg\ x)\ s) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closure\ s) \longrightarrow IN\ (vector\_neg\ x)\ (closure\ s))$

thm SUBSPACE_SUBSTANDARD:

$\forall d::nat.\ subspace\ (GSPEC\ (\lambda GEN\%PVAR\%735::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%735\ (\forall i::nat.\ d < i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ i = (0::real))\ x))$

thm CLOSED_SUBSTANDARD:

$\forall d::nat.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%740::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%740\ (\forall i::nat.\ d < i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ i = (0::real))\ x))$

thm DIM_SUBSTANDARD:

$\forall d \leq dimindex\ HOL\_Light\_Import.UNIV.\ dim\ (GSPEC\ (\lambda GEN\%PVAR\%742::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%742\ (\forall i::nat.\ d < i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ i = (0::real))\ x)) = d$

thm CLOSED_SUBSPACE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow HOL\_Light\_Import.closed\ s$

thm COMPLETE_SUBSPACE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow complete\ s$

thm CLOSED_SPAN:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ (span\ s)$

thm DIM_CLOSURE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dim\ (closure\ s) = dim\ s$

thm CLOSED_BOUNDEDPREIM_CONTINUOUS_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge continuous\_on\ f\ s \wedge (\forall e::real.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%745::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%745\ (IN\ x\ s \wedge vector\_norm\ (f\ x) \leq e)\ x))) \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ f\ s)$

thm CLOSED_INJECTIVE_IMAGE_SUBSET_SUBSPACE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge SUBSET\ s\ t \wedge subspace\ t \wedge linear\ f \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ t \wedge f\ x = vec\ (0::nat) \longrightarrow x = vec\ (0::nat)) \longrightarrow HOL\_Light\_Import.closed\ (IMAGE\ f\ s)$

thm BASIS_COORDINATES_LIPSCHITZ:

$\forall b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ independent\ b \longrightarrow (\exists B>0::real.\ \forall (c::(real,\ ?'a::type)\ cart \Rightarrow real)\ v::(real,\ ?'a::type)\ cart.\ IN\ v\ b \longrightarrow |c\ v| \leq B * vector\_norm\ (vsum\ b\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (c\ v)\ v)))$

thm BASIS_COORDINATES_CONTINUOUS:

$\forall (b::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ independent\ b \wedge (0::real) < e \longrightarrow (\exists d>0::real.\ \forall c::(real,\ ?'a::type)\ cart \Rightarrow real.\ vector\_norm\ (vsum\ b\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (c\ v)\ v)) < d \longrightarrow (\forall v::(real,\ ?'a::type)\ cart.\ IN\ v\ b \longrightarrow |c\ v| < e))$

thm AFFINITY_INVERSES:

$\forall (m::real)\ c::(real,\ ?'a::type)\ cart.\ m \neq (0::real) \longrightarrow (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c) \circ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ (inverse\_class.inverse\ m)\ x)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c))) = id \wedge (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ (inverse\_class.inverse\ m)\ x)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c))) \circ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c) = id$

thm REAL_AFFINITY_LE:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ (0::real) < m \longrightarrow (m * x + c \leq y) = (x \leq inverse\_class.inverse\ m * y + -\ (c\ /\ m))$

thm REAL_LE_AFFINITY:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ (0::real) < m \longrightarrow (y \leq m * x + c) = (inverse\_class.inverse\ m * y + -\ (c\ /\ m) \leq x)$

thm REAL_AFFINITY_LT:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ (0::real) < m \longrightarrow (m * x + c < y) = (x < inverse\_class.inverse\ m * y + - (c / m))$

thm REAL_LT_AFFINITY:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ (0::real) < m \longrightarrow (y < m * x + c) = (inverse\_class.inverse\ m * y + - (c / m) < x)$

thm REAL_AFFINITY_EQ:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ m \neq (0::real) \longrightarrow (m * x + c = y) = (x = inverse\_class.inverse\ m * y + - (c / m))$

thm REAL_EQ_AFFINITY:

$\forall (m::real)\ (c::real)\ (x::real)\ y::real.\ m \neq (0::real) \longrightarrow (y = m * x + c) = (inverse\_class.inverse\ m * y + - (c / m) = x)$

thm VECTOR_AFFINITY_EQ:

$\forall (m::real)\ (c::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ m \neq (0::real) \longrightarrow (vector\_add\ (\%\ m\ x)\ c = y) = (x = vector\_add\ (\%\ (inverse\_class.inverse\ m)\ y)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c)))$

thm VECTOR_EQ_AFFINITY:

$\forall (m::real)\ (c::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ m \neq (0::real) \longrightarrow (y = vector\_add\ (\%\ m\ x)\ c) = (vector\_add\ (\%\ (inverse\_class.inverse\ m)\ y)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c)) = x)$

thm INTERVAL_SING_conjunct1:

$open\_interval\ (?a::(real, ?'a::type)\ cart, ?a) = EMPTY$

thm INTERVAL_SING_conjunct0:

$closed\_interval\ [(?a::(real, ?'a::type)\ cart, ?a)] = INSERT\ ?a\ EMPTY$

thm IMAGE_AFFINITY_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (m::real)\ c::(real, ?'a::type)\ cart.\ IMAGE\ (\lambda x::(real, ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c)\ (closed\_interval\ [(a, b)]) = (if\ closed\_interval\ [(a, b)] = EMPTY\ then\ EMPTY\ else\ if\ (0::real) \leq m\ then\ closed\_interval\ [(vector\_add\ (\%\ m\ a)\ c, vector\_add\ (\%\ m\ b)\ c)]\ else\ closed\_interval\ [(vector\_add\ (\%\ m\ b)\ c, vector\_add\ (\%\ m\ a)\ c)])$

thm EXISTS_IN_GSPEC_conjunct2:

$\forall (P::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow bool)\ f::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type.\ (\exists z::?'a::type.\ IN\ z\ (GSPEC\ (\lambda GEN\%PVAR\%26::?'a::type.\ \exists (w::?'d::type)\ (x::?'c::type)\ y::?'b::type.\ SETSPEC\ GEN\%PVAR\%26\ (P\ w\ x\ y)\ (f\ w\ x\ y))) \wedge (?Q::?'a::type \Rightarrow bool)\ z) = (\exists (w::?'d::type)\ (x::?'c::type)\ y::?'b::type.\ P\ w\ x\ y \wedge ?Q\ (f\ w\ x\ y))$

thm EXISTS_IN_GSPEC_conjunct1:

∀ (P::?'c::type ⇒ ?'b::type ⇒ bool) f::?'c::type ⇒ ?'b::type ⇒ ?'a::type. (∃ z::?'a::type. IN z (GSPEC (λGEN%PVAR%25::?'a::type. ∃ (x::?'c::type) y::?'b::type. SETSPEC GEN%PVAR%25 (P x y) (f x y))) ∧ (?Q::?'a::type ⇒ bool) z) = (∃ (x::?'c::type) y::?'b::type. P x y ∧ ?Q (f x y))

thm EXISTS_IN_GSPEC_conjunct0:

∀ (P::?'b::type ⇒ bool) f::?'b::type ⇒ ?'a::type. (∃ z::?'a::type. IN z (GSPEC (λGEN%PVAR%24::?'a::type. ∃ x::?'b::type. SETSPEC GEN%PVAR%24 (P x) (f x))) ∧ (?Q::?'a::type ⇒ bool) z) = (∃ x::?'b::type. P x ∧ ?Q (f x))

thm SELF_ADJOINT_HAS_EIGENVECTOR_IN_SUBSPACE:

∀ (f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. linear f ∧ adjoint f = f ∧ subspace s ∧ s ≠ INSERT (vec (0::nat)) EMPTY ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ IN (f x) s) ⟶ (∃ (v::(real, ?'a::type) cart) c::real. IN v s ∧ vector_norm v = (1::real) ∧ f v = % c v)

thm SELF_ADJOINT_HAS_EIGENVECTOR:

∀ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. linear f ∧ adjoint f = f ⟶ (∃ (v::(real, ?'a::type) cart) c::real. vector_norm v = (1::real) ∧ f v = % c v)

thm SELF_ADJOINT_HAS_EIGENVECTOR_BASIS_OF_SUBSPACE:

∀ (f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. linear f ∧ adjoint f = f ∧ subspace s ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ IN (f x) s) ⟶ (∃ b::(real, ?'a::type) cart ⇒ bool. SUBSET b s ∧ pairwise orthogonal b ∧ (∀ x::(real, ?'a::type) cart. IN x b ⟶ vector_norm x = (1::real) ∧ (∃ c::real. f x = % c x)) ∧ independent b ∧ span b = s ∧ HAS_SIZE b (dim s))

thm SELF_ADJOINT_HAS_EIGENVECTOR_BASIS:

∀ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. linear f ∧ adjoint f = f ⟶ (∃ b::(real, ?'a::type) cart ⇒ bool. pairwise orthogonal b ∧ (∀ x::(real, ?'a::type) cart. IN x b ⟶ vector_norm x = (1::real) ∧ (∃ c::real. f x = % c x)) ∧ independent b ∧ span b = HOL_Light_Import.UNIV ∧ HAS_SIZE b (dimindex HOL_Light_Import.UNIV))

thm SYMMETRIC_MATRIX_DIAGONALIZABLE_EXPLICIT:

∀ A::((real, ?'a::type) cart, ?'a::type) cart. HOL_Light_Import.transp A = A ⟶ (∃ (P::((real, ?'a::type) cart, ?'a::type) cart) d::nat ⇒ real. orthogonal_matrix P ∧ matrix_mul (HOL_Light_Import.transp P) (matrix_mul A P) = lambda (λi::nat. lambda (λj::nat. if i = j then d i else (0::real))))

thm SYMMETRIC_MATRIX_IMP_DIAGONALIZABLE:

∀ A::((real, ?'a::type) cart, ?'a::type) cart. HOL_Light_Import.transp A = A ⟶ (∃ P::((real, ?'a::type) cart, ?'a::type) cart. orthogonal_matrix P ∧ diagonal_matrix (matrix_mul (HOL_Light_Import.transp P) (matrix_mul A P)))

thm SYMMETRIC_MATRIX_EQ_DIAGONALIZABLE:

$\forall$ *A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. (*HOL_Light_Import*.*transp A* = *A*) = ($\exists$ *P*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*. *orthogonal_matrix P* $\wedge$ *diagonal_matrix* (*matrix_mul* (*HOL_Light_Import*.*transp P*) (*matrix_mul A P*)))

thm RIGID_TRANSFORMATION_BETWEEN_CONGRUENT_SETS:

$\forall$ (*x*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*) *y*::((*real*, *?'b*::*type*) *cart*, *?'a*::*type*) *cart*. ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import*.*UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import*.*UNIV* $\longrightarrow$ *distance* ($ *x i*, $ *x j*) = *distance* ($ *y i*, $ *y j*)) $\longrightarrow$ ($\exists$ (*a*::(*real*, *?'b*::*type*) *cart*) *f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*. *orthogonal_transformation f* $\wedge$ ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import*.*UNIV* $\longrightarrow$ $ *y i* = *vector_add a* (*f* ($ *x i*))))

thm RIGID_TRANSFORMATION_BETWEEN_3:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*b'*::(*real*, *?'a*::*type*) *cart*) *c'*::(*real*, *?'a*::*type*) *cart*. *distance* (*a*, *b*) = *distance* (*a'*, *b'*) $\wedge$ *distance* (*b*, *c*) = *distance* (*b'*, *c'*) $\wedge$ *distance* (*c*, *a*) = *distance* (*c'*, *a'*) $\longrightarrow$ ($\exists$ (*k*::(*real*, *?'a*::*type*) *cart*) *f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\wedge$ *a'* = *vector_add k* (*f a*) $\wedge$ *b'* = *vector_add k* (*f b*) $\wedge$ *c'* = *vector_add k* (*f c*))

thm RIGID_TRANSFORMATION_BETWEEN_2:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) *b'*::(*real*, *?'a*::*type*) *cart*. *distance* (*a*, *b*) = *distance* (*a'*, *b'*) $\longrightarrow$ ($\exists$ (*k*::(*real*, *?'a*::*type*) *cart*) *f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *orthogonal_transformation f* $\wedge$ *a'* = *vector_add k* (*f a*) $\wedge$ *b'* = *vector_add k* (*f b*))

thm DEF_sums:

sums = (λ(*_243483*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*_243484*::(*real*, *?'a*::*type*) *cart*) *_243485*::*nat* $\Rightarrow$ *bool*. −−> (λ*n*::*nat*. *vsum* (*HOL_Light_Import*.*INTER _243485* (*dotdot* (*0*::*nat*) *n*)) *_243483*) *_243484 sequentially*)

thm sums:

$\forall$ (*s*::*nat* $\Rightarrow$ *bool*) (*f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *l*::(*real*, *?'a*::*type*) *cart*. *sums f l s* = −−> (λ*n*::*nat*. *vsum* (*HOL_Light_Import*.*INTER s* (*dotdot* (*0*::*nat*) *n*)) *f*) *l sequentially*

thm DEF_infsum:

infsum = (λ(*_243504*::*nat* $\Rightarrow$ *bool*) *_243505*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *SOME l*::(*real*, *?'a*::*type*) *cart*. *sums _243505 l _243504*)

thm infsum:

$\forall$ (*f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::*nat* $\Rightarrow$ *bool*. *infsum s f* = (*SOME l*::(*real*, *?'a*::*type*) *cart*. *sums f l s*)

thm DEF_summable:

summable = (λ(*_243516*::*nat* ⇒ *bool*) *_243517*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. ∃ *l*::(*real*, *?'a*::*type*) *cart*. *sums _243517 l _243516*)

thm summable:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *summable s f* = (∃ *l*::(*real*, *?'a*::*type*) *cart*. *sums f l s*)

thm SUMS_SUMMABLE:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *sums f l s* ⟶ *summable s f*

thm SUMS_INFSUM:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *sums f* (*infsum s f*) *s* = *summable s f*

thm SUMS_LIM:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *sums f* (*lim sequentially* (λ*n*::*nat*. *vsum* (*HOL_Light_Import.INTER s* (*dotdot* (*0*::*nat*) *n*)) *f*)) *s* = *summable s f*

thm DEF_from:

from = (λ*_243613*::*nat*. *GSPEC* (λ*GEN%PVAR%750*::*nat*. ∃ *m*::*nat*. *SET-SPEC GEN%PVAR%750* (*_243613* ≤ *m*) *m*))

thm from:

∀ *n*::*nat*. *from n* = *GSPEC* (λ*GEN%PVAR%750*::*nat*. ∃ *m*::*nat*. *SETSPEC GEN%PVAR%750* (*n* ≤ *m*) *m*)

thm FROM_0:

from (*0*::*nat*) = *HOL_Light_Import.UNIV*

thm FINITE_INTER_NUMSEG:

∀ (*s*::*nat* ⇒ *bool*) (*m*::*nat*) *n*::*nat*. *FINITE* (*HOL_Light_Import.INTER s* (*dotdot m n*))

thm FROM_INTER_NUMSEG_GEN:

∀ (*k*::*nat*) (*m*::*nat*) *n*::*nat*. *HOL_Light_Import.INTER* (*from k*) (*dotdot m n*) = (*if m* < *k then dotdot k n else dotdot m n*)

thm FROM_INTER_NUMSEG:

∀ (*k*::*nat*) *n*::*nat*. *HOL_Light_Import.INTER* (*from k*) (*dotdot* (*0*::*nat*) *n*) = *dotdot k n*

thm IN_FROM:

∀ (*m*::*nat*) *n*::*nat*. *IN m* (*from n*) = (*n* ≤ *m*)

thm SERIES_FROM:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ k::nat.\ sums\ f\ l\ (from\ k) = --> (\lambda n::nat.\ vsum\ (dotdot\ k\ n)\ f)\ l\ sequentially$

thm SERIES_UNIQUE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (l'::(real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \wedge sums\ f\ l'\ s \longrightarrow l = l'$

thm INFSUM_UNIQUE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow infsum\ s\ f = l$

thm SERIES_FINITE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ FINITE\ s \longrightarrow sums\ f\ (vsum\ s\ f)\ s$

thm SERIES_LINEAR:

$\forall (f::nat \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'b::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \wedge linear\ h \longrightarrow sums\ (\lambda n::nat.\ h\ (f\ n))\ (h\ l)\ s$

thm SERIES_0:

$\forall s::nat \Rightarrow bool.\ sums\ (\lambda n::nat.\ vec\ (0::nat))\ (vec\ (0::nat))\ s$

thm SERIES_ADD:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (x0::(real,\ ?'a::type)\ cart)\ (y::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (y0::(real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ x\ x0\ s \wedge sums\ y\ y0\ s \longrightarrow sums\ (\lambda n::nat.\ vector\_add\ (x\ n)\ (y\ n))\ (vector\_add\ x0\ y0)\ s$

thm SERIES_SUB:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (x0::(real,\ ?'a::type)\ cart)\ (y::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (y0::(real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ x\ x0\ s \wedge sums\ y\ y0\ s \longrightarrow sums\ (\lambda n::nat.\ vector\_sub\ (x\ n)\ (y\ n))\ (vector\_sub\ x0\ y0)\ s$

thm SERIES_CMUL:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (x0::(real,\ ?'a::type)\ cart)\ (c::real)\ s::nat \Rightarrow bool.\ sums\ x\ x0\ s \longrightarrow sums\ (\lambda n::nat.\ \%\ c\ (x\ n))\ (\%\ c\ x0)\ s$

thm SERIES_NEG:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (x0::(real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ sums\ x\ x0\ s \longrightarrow sums\ (\lambda n::nat.\ vector\_neg\ (x\ n))\ (vector\_neg\ x0)\ s$

thm SUMS_IFF:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \longrightarrow sums\ f\ (?l::(real,\ ?'a::type)\ cart)\ k = sums\ g\ ?l\ k$

thm SUMS_EQ:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \land sums\ f\ (?l::(real,\ ?'a::type)\ cart)\ k \longrightarrow sums\ g\ ?l\ k$

thm SUMS_0:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ (\forall n::nat.\ IN\ n\ s \longrightarrow f\ n = vec\ (0::nat)) \longrightarrow sums\ f\ (vec\ (0::nat))\ s$

thm SERIES_FINITE_SUPPORT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ k::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.INTER\ s\ k) \land (\forall x::nat.\ \neg\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ k) \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow sums\ f\ (vsum\ (HOL\_Light\_Import.INTER\ s\ k)\ f)\ k$

thm SERIES_COMPONENT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ (l::(real,\ ?'a::type)\ cart)\ k::nat.\ sums\ f\ l\ s \land (1::nat) \leq k \land k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow sums\ (\lambda i::nat.\ lift\ (\$\ (f\ i)\ k))\ (lift\ (\$\ l\ k))\ s$

thm SERIES_DIFFS:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat.\ -->\ f\ (vec\ (0::nat))\ sequentially \longrightarrow sums\ (\lambda n::nat.\ vector\_sub\ (f\ n)\ (f\ (n + (1::nat))))\ (f\ k)\ (from\ k)$

thm SERIES_TRIVIAL:

$\forall f::nat \Rightarrow (real,\ ?'a::type)\ cart.\ sums\ f\ (vec\ (0::nat))\ EMPTY$

thm SERIES_RESTRICT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (k::nat \Rightarrow bool)\ l::(real,\ ?'a::type)\ cart.\ sums\ (\lambda n::nat.\ if\ IN\ n\ k\ then\ f\ n\ else\ vec\ (0::nat))\ l\ HOL\_Light\_Import.UNIV = sums\ f\ l\ k$

thm SERIES_VSUM:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (k::nat \Rightarrow bool)\ s::nat \Rightarrow bool.\ FINITE\ s \land SUBSET\ s\ k \land (\forall x::nat.\ \neg\ IN\ x\ s \longrightarrow f\ x = vec\ (0::nat)) \land vsum\ s\ f = l \longrightarrow sums\ f\ l\ k$

thm SUMS_REINDEX:

$\forall (k::nat)\ (a::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ n::nat.\ sums\ (\lambda x::nat.\ a\ (x + k))\ l\ (from\ n) = sums\ a\ l\ (from\ (n + k))$

thm SUMMABLE_LINEAR:

$\forall (f::nat \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ f \land linear\ h \longrightarrow summable\ s\ (\lambda n::nat.\ h\ (f\ n))$

thm SUMMABLE_0:

$\forall s::nat \Rightarrow bool.\ summable\ s\ (\lambda n::nat.\ vec\ (0::nat))$

thm SUMMABLE_ADD:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (y::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ x \wedge summable\ s\ y \longrightarrow summable\ s\ (\lambda n::nat.\ vector\_add\ (x\ n)\ (y\ n))$

thm SUMMABLE_SUB:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (y::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ x \wedge summable\ s\ y \longrightarrow summable\ s\ (\lambda n::nat.\ vector\_sub\ (x\ n)\ (y\ n))$

thm SUMMABLE_CMUL:

$\forall (s::nat \Rightarrow bool)\ (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ c::real.\ summable\ s\ x \longrightarrow summable\ s\ (\lambda n::nat.\ \%\ c\ (x\ n))$

thm SUMMABLE_NEG:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ x \longrightarrow summable\ s\ (\lambda n::nat.\ vector\_neg\ (x\ n))$

thm SUMMABLE_IFF:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \longrightarrow summable\ k\ f = summable\ k\ g$

thm SUMMABLE_EQ:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \wedge summable\ k\ f \longrightarrow summable\ k\ g$

thm SUMMABLE_COMPONENT:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ k::nat.\ summable\ s\ f \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow summable\ s\ (\lambda i::nat.\ lift\ (\$\ (f\ i)\ k))$

thm SERIES_SUBSET:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ (t::nat \Rightarrow bool)\ l::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ t \wedge sums\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ x\ i\ else\ vec\ (0::nat))\ l\ t \longrightarrow sums\ x\ l\ s$

thm SUMMABLE_SUBSET:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ SUBSET\ s\ t \wedge summable\ t\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ x\ i\ else\ vec\ (0::nat)) \longrightarrow summable\ s\ x$

thm SUMMABLE_TRIVIAL:

$\forall f::nat \Rightarrow (real,\ ?'a::type)\ cart.\ summable\ EMPTY\ f$

thm SUMMABLE_RESTRICT:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *k*::*nat* ⇒ *bool*. *summable HOL_Light_Import.UNIV* (λ*n*::*nat*. *if IN n k then f n else vec* (*0*::*nat*)) = *summable k f*

thm SUMS_FINITE_DIFF:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*t*::*nat* ⇒ *bool*) (*s*::*nat* ⇒ *bool*) *l*::(*real*, *?'a*::*type*) *cart*. *SUBSET t s* ∧ *FINITE t* ∧ *sums f l s* ⟶ *sums f* (*vector_sub l* (*vsum t f*)) (*DIFF s t*)

thm SUMS_FINITE_UNION:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::*nat* ⇒ *bool*) (*t*::*nat* ⇒ *bool*) *l*::(*real*, *?'a*::*type*) *cart*. *FINITE t* ∧ *sums f l s* ⟶ *sums f* (*vector_add l* (*vsum* (*DIFF t s*) *f*)) (*HOL_Light_Import.UNION s t*)

thm SUMS_OFFSET:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) (*m*::*nat*) *n*::*nat*. *sums f l* (*from m*) ∧ *m* < *n* ⟶ *sums f* (*vector_sub l* (*vsum* (*dotdot m* (*n* − (*1*::*nat*))) *f*)) (*from n*)

thm SUMS_OFFSET_REV:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*l*::(*real*, *?'a*::*type*) *cart*) (*m*::*nat*) *n*::*nat*. *sums f l* (*from m*) ∧ *n* < *m* ⟶ *sums f* (*vector_add l* (*vsum* (*dotdot n* (*m* − (*1*::*nat*))) *f*)) (*from n*)

thm SUMMABLE_REINDEX:

∀ (*k*::*nat*) (*a*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *n*::*nat*. *summable* (*from n*) (λ*x*::*nat*. *a* (*x* + *k*)) = *summable* (*from* (*n* + *k*)) *a*

thm INFSUM_LINEAR:

∀ (*f*::*nat* ⇒ (*real*, *?'b*::*type*) *cart*) (*h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *summable s f* ∧ *linear h* ⟶ *infsum s* (λ*n*::*nat*. *h* (*f n*)) = *h* (*infsum s f*)

thm INFSUM_0:

infsum (*?s*::*nat* ⇒ *bool*) (λ*i*::*nat*. *vec* (*0*::*nat*)) = *vec* (*0*::*nat*)

thm INFSUM_ADD:

∀ (*x*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*y*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *summable s x* ∧ *summable s y* ⟶ *infsum s* (λ*i*::*nat*. *vector_add* (*x i*) (*y i*)) = *vector_add* (*infsum s x*) (*infsum s y*)

thm INFSUM_SUB:

∀ (*x*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*y*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::*nat* ⇒ *bool*. *summable s x* ∧ *summable s y* ⟶ *infsum s* (λ*i*::*nat*. *vector_sub* (*x i*) (*y i*)) = *vector_sub* (*infsum s x*) (*infsum s y*)

thm INFSUM_CMUL:

∀(s::nat ⇒ bool) (x::nat ⇒ (real, ?'a::type) cart) c::real. summable s x ⟶ infsum s (λn::nat. % c (x n)) = % c (infsum s x)

thm INFSUM_NEG:

∀(s::nat ⇒ bool) x::nat ⇒ (real, ?'a::type) cart. summable s x ⟶ infsum s (λn::nat. vector_neg (x n)) = vector_neg (infsum s x)

thm INFSUM_EQ:

∀(f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, ?'a::type) cart) k::nat ⇒ bool. summable k f ∧ summable k g ∧ (∀x::nat. IN x k ⟶ f x = g x) ⟶ infsum k f = infsum k g

thm INFSUM_RESTRICT:

∀(k::nat ⇒ bool) a::nat ⇒ (real, ?'a::type) cart. infsum HOL_Light_Import.UNIV (λn::nat. if IN n k then a n else vec (0::nat)) = infsum k a

thm PARTIAL_SUMS_COMPONENT_LE_INFSUM:

∀(f::nat ⇒ (real, ?'a::type) cart) (s::nat ⇒ bool) (k::nat) n::nat. (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ∧ (∀i::nat. IN i s ⟶ (0::real) ≤ $ (f i) k) ∧ summable s f ⟶ $ (vsum (HOL_Light_Import.INTER s (dotdot (0::nat) n)) f) k ≤ $ (infsum s f) k

thm PARTIAL_SUMS_DROP_LE_INFSUM:

∀(f::nat ⇒ (real, unit) cart) (s::nat ⇒ bool) n::nat. (∀i::nat. IN i s ⟶ (0::real) ≤ HOL_Light_Import.drop (f i)) ∧ summable s f ⟶ HOL_Light_Import.drop (vsum (HOL_Light_Import.INTER s (dotdot (0::nat) n)) f) ≤ HOL_Light_Import.drop (infsum s f)

thm SEQUENCE_CAUCHY_WLOG:

∀(P::nat ⇒ bool) s::nat ⇒ (real, ?'a::type) cart. (∀(m::nat) n::nat. P m ∧ P n ⟶ distance (s m, s n) < (?e::real)) = (∀(m::nat) n::nat. P m ∧ P n ∧ m ≤ n ⟶ distance (s m, s n) < ?e)

thm VSUM_DIFF_LEMMA:

∀(f::nat ⇒ (real, ?'a::type) cart) (k::nat ⇒ bool) (m::nat) n::nat. m ≤ n ⟶ vector_sub (vsum (HOL_Light_Import.INTER k (dotdot (0::nat) n)) f) (vsum (HOL_Light_Import.INTER k (dotdot (0::nat) m)) f) = vsum (HOL_Light_Import.INTER k (dotdot (m + (1::nat)) n)) f

thm NORM_VSUM_TRIVIAL_LEMMA:

∀e>0::real. ((?P::bool) ⟶ vector_norm (vsum (HOL_Light_Import.INTER (?s::nat ⇒ bool) (dotdot (?m::nat) (?n::nat))) (?f::nat ⇒ (real, ?'a::type) cart)) < e) = (?P ⟶ ?n < ?m ∨ vector_norm (vsum (HOL_Light_Import.INTER ?s (dotdot ?m ?n)) ?f) < e)

thm SERIES_CAUCHY:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ (\exists\, l::(real,\ ?'a::type)\ cart.\ sums\, f\, l\, s) = (\forall\, e>0::real.\ \exists\, N::nat.\ \forall (m::nat)\ n::nat.\ N \leq m \longrightarrow vector\_norm\ (vsum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ m\ n))\ f) < e)$

thm SUMMABLE_CAUCHY:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ f = (\forall\, e>0::real.\ \exists\, N::nat.\ \forall (m::nat)\ n::nat.\ N \leq m \longrightarrow vector\_norm\ (vsum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ m\ n))\ f) < e)$

thm SUMMABLE_IFF_EVENTUALLY:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\exists\, N::nat.\ \forall\, n::nat.\ N \leq n \wedge IN\ n\ k \longrightarrow f\, n = g\ n) \longrightarrow summable\ k\, f = summable\ k\ g$

thm SUMMABLE_EQ_EVENTUALLY:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\exists\, N::nat.\ \forall\, n::nat.\ N \leq n \wedge IN\ n\ k \longrightarrow f\, n = g\ n) \wedge summable\ k\, f \longrightarrow summable\ k\ g$

thm SUMMABLE_IFF_COFINITE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow summable\ s\ f = summable\ t\, f$

thm SUMMABLE_EQ_COFINITE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \wedge summable\ s\ f \longrightarrow summable\ t\, f$

thm SUMMABLE_FROM_ELSEWHERE:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ summable\ (from\ m)\ f \longrightarrow summable\ (from\ n)\ f$

thm SERIES_CAUCHY_UNIFORM:

$\forall (P::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ (\exists\, l::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ \forall\, e>0::real.\ \exists\, N::nat.\ \forall (n::nat)\ x::?'b::type.\ N \leq n \wedge P\ x \longrightarrow distance\ (vsum\ (HOL\_Light\_Import.INTER\ k\ (dotdot\ (0::nat)\ n))\ (f\ x),\ l\ x) < e) = (\forall\, e>0::real.\ \exists\, N::nat.\ \forall (m::nat)\ (n::nat)\ x::?'b::type.\ N \leq m \wedge P\, x \longrightarrow vector\_norm\ (vsum\ (HOL\_Light\_Import.INTER\ k\ (dotdot\ m\ n))\ (f\, x)) < e)$

thm SERIES_GOESTOZERO:

$\forall (s::nat \Rightarrow bool)\ x::nat \Rightarrow (real,\ ?'a::type)\ cart.\ summable\ s\, x \longrightarrow (\forall\, e>0::real.\ eventually\ (\lambda n::nat.\ IN\ n\ s \longrightarrow vector\_norm\ (x\ n) < e)\ sequentially)$

thm SUMMABLE_IMP_TOZERO:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ summable\ k\ f \longrightarrow -->$ $(\lambda n::nat.\ if\ IN\ n\ k\ then\ f\ n\ else\ vec\ (0::nat))\ (vec\ (0::nat))\ sequentially$

thm SUMMABLE_IMP_BOUNDED:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ summable\ k\ f \longrightarrow bounded$ $(IMAGE\ f\ k)$

thm SUMMABLE_IMP_SUMS_BOUNDED:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat.\ summable\ (from\ k)\ f \longrightarrow bounded$ $(GSPEC\ (\lambda GEN\%PVAR\%751::(real,\ ?'a::type)\ cart.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%751$ $(IN\ n\ HOL\_Light\_Import.UNIV)\ (vsum\ (dotdot\ k\ n)\ f)))$

thm SERIES_COMPARISON:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ (\exists l::(real,$ $unit)\ cart.\ sums\ (lift \circ g)\ l\ s) \land (\exists N::nat.\ \forall n::nat.\ N \leq n \land IN\ n\ s \longrightarrow$ $vector\_norm\ (f\ n) \leq g\ n) \longrightarrow (\exists l::(real,\ ?'a::type)\ cart.\ sums\ f\ l\ s)$

thm SUMMABLE_COMPARISON:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ summable\ s$ $(lift \circ g) \land (\exists N::nat.\ \forall n::nat.\ N \leq n \land IN\ n\ s \longrightarrow vector\_norm\ (f\ n) \leq g\ n)$ $\longrightarrow summable\ s\ f$

thm SERIES_LIFT_ABSCONV_IMP_CONV:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat \Rightarrow bool.\ summable\ k\ (\lambda n::nat.\ lift$ $(vector\_norm\ (x\ n))) \longrightarrow summable\ k\ x$

thm SUMMABLE_SUBSET_ABSCONV:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ summable\ s$ $(\lambda n::nat.\ lift\ (vector\_norm\ (x\ n))) \land SUBSET\ t\ s \longrightarrow summable\ t\ (\lambda n::nat.$ $lift\ (vector\_norm\ (x\ n)))$

thm SERIES_COMPARISON_UNIFORM:

$\forall (f::?'b::type \Rightarrow nat \Rightarrow (real,\ ?'a::type)\ cart)\ (g::nat \Rightarrow real)\ (P::?'b::type \Rightarrow$ $bool)\ s::nat \Rightarrow bool.\ (\exists l::(real,\ unit)\ cart.\ sums\ (lift \circ g)\ l\ s) \land (\exists N::nat.$ $\forall (n::nat)\ x::?'b::type.\ N \leq n \land IN\ n\ s \land P\ x \longrightarrow vector\_norm\ (f\ x\ n) \leq g$ $n) \longrightarrow (\exists l::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ \forall e>0::real.\ \exists N::nat.\ \forall (n::nat)$ $x::?'b::type.\ N \leq n \land P\ x \longrightarrow distance\ (vsum\ (HOL\_Light\_Import.INTER\ s$ $(dotdot\ (0::nat)\ n))\ (f\ x),\ l\ x) < e)$

thm SERIES_RATIO:

$\forall (c::real)\ (a::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ N::nat.\ c < (1::real)$ $\land (\forall n \geq N.\ vector\_norm\ (a\ (Suc\ n)) \leq c * vector\_norm\ (a\ n)) \longrightarrow (\exists l::(real,$ $?'a::type)\ cart.\ sums\ a\ l\ s)$

thm BOUNDED_PARTIAL_SUMS:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart)\ k::nat.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%754::(real,$ $?'a::type)\ cart.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%754\ (IN\ n\ HOL\_Light\_Import.UNIV)$

(vsum (dotdot k n) f))) ⟶ *bounded (GSPEC (λGEN%PVAR%755::(real, ?'a::type) cart.* ∃ *(m::nat) n::nat. SETSPEC GEN%PVAR%755 (IN m HOL_Light_Import.UNIV* ∧ *IN n HOL_Light_Import.UNIV) (vsum (dotdot m n) f)))*

thm SUMMABLE_BILINEAR_PARTIAL_PRE:

∀ *(f::nat* ⇒ *(real, ?'c::type) cart) (g::nat* ⇒ *(real, ?'b::type) cart) (h::(real, ?'c::type) cart* ⇒ *(real, ?'b::type) cart* ⇒ *(real, ?'a::type) cart) (l::(real, ?'a::type) cart) k::nat. bilinear h* ∧ *−−> (λn::nat. h (f (n + (1::nat))) (g n)) l sequentially* ∧ *summable (from k) (λn::nat. h (vector_sub (f (n + (1::nat))) (f n)) (g n))* ⟶ *summable (from k) (λn::nat. h (f n) (vector_sub (g n) (g (n − (1::nat)))))*

thm SERIES_DIRICHLET_BILINEAR:

∀ *(f::nat* ⇒ *(real, ?'c::type) cart) (g::nat* ⇒ *(real, ?'b::type) cart) (h::(real, ?'b::type) cart* ⇒ *(real, ?'c::type) cart* ⇒ *(real, ?'a::type) cart) (k::nat) (m::nat) (p::nat) l::(real, ?'a::type) cart. bilinear h* ∧ *bounded (GSPEC (λGEN%PVAR%756::(real, ?'c::type) cart.* ∃ *n::nat. SETSPEC GEN%PVAR%756 (IN n HOL_Light_Import.UNIV) (vsum (dotdot m n) f)))* ∧ *summable (from p) (λn::nat. lift (vector_norm (vector_sub (g (n + (1::nat))) (g n))))* ∧ *−−> (λn::nat. h (g (n + (1::nat))) (vsum (dotdot (1::nat) n) f)) l sequentially* ⟶ *summable (from k) (λn::nat. h (g n) (f n))*

thm SERIES_DIRICHLET:

∀ *(f::nat* ⇒ *(real, ?'a::type) cart) (g::nat* ⇒ *real) (N::nat) (k::nat) m::nat. bounded (GSPEC (λGEN%PVAR%757::(real, ?'a::type) cart.* ∃ *n::nat. SETSPEC GEN%PVAR%757 (IN n HOL_Light_Import.UNIV) (vsum (dotdot m n) f)))* ∧ *(*∀ *n≥N. g (n + (1::nat))* ≤ *g n)* ∧ *−−> (lift* ∘ *g) (vec (0::nat)) sequentially* ⟶ *summable (from k) (λn::nat. % (g n) (f n))*

thm SERIES_INJECTIVE_IMAGE_STRONG:

∀ *(x::nat* ⇒ *(real, ?'a::type) cart) (s::nat* ⇒ *bool) f::nat* ⇒ *nat. summable (IMAGE f s) (λn::nat. lift (vector_norm (x n)))* ∧ *(*∀ *(m::nat) n::nat. IN m s* ∧ *IN n s* ∧ *f m = f n* ⟶ *m = n)* ⟶ *−−> (λn::nat. vector_sub (vsum (HOL_Light_Import.INTER (IMAGE f s) (dotdot (0::nat) n)) x) (vsum (HOL_Light_Import.INTER s (dotdot (0::nat) n)) (x* ∘ *f))) (vec (0::nat)) sequentially*

thm SERIES_INJECTIVE_IMAGE:

∀ *(x::nat* ⇒ *(real, ?'a::type) cart) (s::nat* ⇒ *bool) (f::nat* ⇒ *nat) l::(real, ?'a::type) cart. summable (IMAGE f s) (λn::nat. lift (vector_norm (x n)))* ∧ *(*∀ *(m::nat) n::nat. IN m s* ∧ *IN n s* ∧ *f m = f n* ⟶ *m = n)* ⟶ *sums (x* ∘ *f) l s = sums x l (IMAGE f s)*

thm SERIES_REARRANGE_EQ:

∀ *(x::nat* ⇒ *(real, ?'a::type) cart) (s::nat* ⇒ *bool) (p::nat* ⇒ *nat) l::(real, ?'a::type) cart. summable s (λn::nat. lift (vector_norm (x n)))* ∧ *permutes p s* ⟶ *sums (x* ∘ *p) l s = sums x l s*

thm SERIES_REARRANGE:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ (p::nat \Rightarrow nat)\ l::(real,\ ?'a::type)\ cart.\ summable\ s\ (\lambda n::nat.\ lift\ (vector\_norm\ (x\ n))) \wedge permutes\ p\ s \wedge sums\ x\ l\ s \longrightarrow sums\ (x \circ p)\ l\ s$

thm SUMMABLE_REARRANGE:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (s::nat \Rightarrow bool)\ p::nat \Rightarrow nat.\ summable\ s\ (\lambda n::nat.\ lift\ (vector\_norm\ (x\ n))) \wedge permutes\ p\ s \longrightarrow summable\ s\ (x \circ p)$

thm BANACH_FIX:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ complete\ s \wedge s \neq EMPTY \wedge (0::real) \leq c \wedge c < (1::real) \wedge SUBSET\ (IMAGE\ f\ s)\ s \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow distance\ (f\ x,\ f\ y) \leq c * distance\ (x,\ y)) \longrightarrow (\exists !x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge f\ x = x)$

thm EDELSTEIN_FIX:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge s \neq EMPTY \wedge SUBSET\ (IMAGE\ f\ s)\ s \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow distance\ (f\ x,\ f\ y) < distance\ (x,\ y)) \longrightarrow (\exists !x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge f\ x = x)$

thm DINI:

$\forall (f::nat \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge (\forall n::nat.\ continuous\_on\ (f\ n)\ s) \wedge continuous\_on\ g\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow -->\ (\lambda n::nat.\ f\ n\ x)\ (g\ x)\ sequentially) \wedge (\forall (n::nat)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ n\ x) \leq HOL\_Light\_Import.drop\ (f\ (n + (1::nat))\ x)) \longrightarrow (\forall e>0::real.\ eventually\ (\lambda n::nat.\ \forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (vector\_sub\ (f\ n\ x)\ (g\ x)) < e)\ sequentially)$

thm DEF_closest_point:

$closest\_point = (\lambda(\_249756::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_249757::(real,\ ?'a::type)\ cart.\ SOME\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_249756 \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ \_249756 \longrightarrow distance\ (\_249757,\ x) \leq distance\ (\_249757,\ y)))$

thm closest_point:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ closest\_point\ s\ a = (SOME\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ y)))$

thm CLOSEST_POINT_EXISTS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow IN\ (closest\_point\ s\ a)\ s \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow distance\ (a,\ closest\_point\ s\ a) \leq distance\ (a,\ y))$

thm CLOSEST_POINT_IN_SET:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ a::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow IN \ (closest\_point \ s \ a) \ s$

thm CLOSEST_POINT_LE:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ (a::(real, ?'a::type) \ cart) \ x::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge IN \ x \ s \longrightarrow distance \ (a, \ closest\_point \ s \ a) \leq distance \ (a, \ x)$

thm CLOSEST_POINT_SELF:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ x::(real, ?'a::type) \ cart. \ IN \ x \ s \longrightarrow closest\_point \ s \ x = x$

thm CLOSEST_POINT_REFL:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ x::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow (closest\_point \ s \ x = x) = IN \ x \ s$

thm DIST_CLOSEST_POINT_LIPSCHITZ:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ (x::(real, ?'a::type) \ cart) \ y::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow |distance \ (x, \ closest\_point \ s \ x) - distance \ (y, \ closest\_point \ s \ y)| \leq distance \ (x, \ y)$

thm CONTINUOUS_AT_DIST_CLOSEST_POINT:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ x::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow continuous \ (\lambda x::(real, ?'a::type) \ cart. \ lift \ (distance \ (x, \ closest\_point \ s \ x))) \ (at \ x)$

thm CONTINUOUS_ON_DIST_CLOSEST_POINT:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ t::(real, ?'a::type) \ cart \Rightarrow bool. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow continuous\_on \ (\lambda x::(real, ?'a::type) \ cart. \ lift \ (distance \ (x, \ closest\_point \ s \ x))) \ t$

thm SEGMENT_TO_CLOSEST_POINT:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ a::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER \ (open\_segment \ (a, \ closest\_point \ s \ a)) \ s = EMPTY$

thm SEGMENT_TO_POINT_EXISTS:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ a::(real, ?'a::type) \ cart. \ HOL\_Light\_Import.closed \ s \wedge s \neq EMPTY \longrightarrow (\exists \ b::(real, ?'a::type) \ cart. \ IN \ b \ s \wedge HOL\_Light\_Import.INTER \ (open\_segment \ (a, \ b)) \ s = EMPTY)$

thm DEF_setdist:

$setdist = (\lambda\_250067::((real, ?'a::type) \ cart \Rightarrow bool) \times ((real, ?'a::type) \ cart \Rightarrow bool). \ if \ fst \ \_250067 = EMPTY \vee snd \ \_250067 = EMPTY \ then \ 0::real \ else \ HOL\_Light\_Import.inf \ (GSPEC \ (\lambda GEN\%PVAR\%759::real. \ \exists (x::(real, ?'a::type)$

cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%759 (IN x (fst _250067) ∧ IN y (snd _250067)) (distance (x, y)))))

thm setdist:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. setdist (s, t) = (if s = EMPTY ∨ t = EMPTY then 0::real else HOL_Light_Import.inf (GSPEC (λGEN%PVAR%759::real. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%759 (IN x s ∧ IN y t) (distance (x, y)))))

thm SETDIST_EMPTY:

(∀t::(real, ?'b::type) cart ⇒ bool. setdist (EMPTY, t) = (0::real)) ∧ (∀s::(real, ?'a::type) cart ⇒ bool. setdist (s, EMPTY) = (0::real))

thm SETDIST_POS_LE:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. (0::real) ≤ setdist (s, t)

thm REAL_LE_SETDIST:

∀(s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) d::real. s ≠ EMPTY ∧ t ≠ EMPTY ∧ (∀(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y t ⟶ d ≤ distance (x, y)) ⟶ d ≤ setdist (s, t)

thm SETDIST_LE_DIST:

∀(s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y t ⟶ setdist (s, t) ≤ distance (x, y)

thm SETDIST_EMPTY_conjunct1:

∀s::(real, ?'a::type) cart ⇒ bool. setdist (s, EMPTY) = (0::real)

thm SETDIST_EMPTY_conjunct0:

∀t::(real, ?'a::type) cart ⇒ bool. setdist (EMPTY, t) = (0::real)

thm REAL_LE_SETDIST_EQ:

∀(d::real) (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. (d ≤ setdist (s, t)) = ((∀(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y t ⟶ d ≤ distance (x, y)) ∧ (s = EMPTY ∨ t = EMPTY ⟶ d ≤ (0::real)))

thm SETDIST_REFL:

∀s::(real, ?'a::type) cart ⇒ bool. setdist (s, s) = (0::real)

thm SETDIST_SYM:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. setdist (s, t) = setdist (t, s)

thm SETDIST_TRIANGLE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ setdist\ (s,\ t) \le setdist\ (s,\ INSERT\ a\ EMPTY) + setdist\ (INSERT\ a\ EMPTY,\ t)$

thm SETDIST_SINGS:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ setdist\ (INSERT\ x\ EMPTY,\ INSERT\ y\ EMPTY) = distance\ (x,\ y)$

thm SETDIST_LIPSCHITZ:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::?'a::type)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ |setdist\ (INSERT\ x\ EMPTY,\ s) - setdist\ (INSERT\ y\ EMPTY,\ s)| \le distance\ (x,\ y)$

thm CONTINUOUS_AT_LIFT_SETDIST:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ continuous\ (\lambda y::(real,\ ?'a::type)\ cart.\ lift\ (setdist\ (INSERT\ y\ EMPTY,\ s)))\ (at\ x)$

thm CONTINUOUS_ON_LIFT_SETDIST:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (\lambda y::(real,\ ?'a::type)\ cart.\ lift\ (setdist\ (INSERT\ y\ EMPTY,\ s)))\ t$

thm SETDIST_DIFFERENCES:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ setdist\ (s,\ t) = setdist\ (INSERT\ (vec\ (0::nat))\ EMPTY,\ GSPEC\ (\lambda GEN\%PVAR\%764::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%764\ (IN\ x\ s \land IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm SETDIST_SUBSET_RIGHT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ t \neq EMPTY \land SUBSET\ t\ u \longrightarrow setdist\ (s,\ u) \le setdist\ (s,\ t)$

thm SETDIST_SUBSET_LEFT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ s \neq EMPTY \land SUBSET\ s\ t \longrightarrow setdist\ (t,\ u) \le setdist\ (s,\ u)$

thm SETDIST_CLOSURE_conjunct0:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ setdist\ (closure\ s,\ t) = setdist\ (s,\ t)$

thm SETDIST_CLOSURE:

$(\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ setdist\ (closure\ s,\ t) = setdist\ (s,\ t)) \land (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ setdist\ (s,\ closure\ t) = setdist\ (s,\ t))$

thm SETDIST_COMPACT_CLOSED:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact s* $\wedge$ *HOL_Light_Import.closed t* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ *t* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y t* $\wedge$ *distance* (*x*, *y*) = *setdist* (*s*, *t*))

thm SETDIST_CLOSED_COMPACT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed s* $\wedge$ *compact t* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ *t* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y t* $\wedge$ *distance* (*x*, *y*) = *setdist* (*s*, *t*))

thm SETDIST_EQ_0_COMPACT_CLOSED:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *compact s* $\wedge$ *HOL_Light_Import.closed t* $\longrightarrow$ (*setdist* (*s*, *t*) = (*0*::*real*)) = (*s* = *EMPTY* $\vee$ *t* = *EMPTY* $\vee$ *HOL_Light_Import.INTER s t* $\neq$ *EMPTY*)

thm SETDIST_EQ_0_CLOSED_COMPACT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed s* $\wedge$ *compact t* $\longrightarrow$ (*setdist* (*s*, *t*) = (*0*::*real*)) = (*s* = *EMPTY* $\vee$ *t* = *EMPTY* $\vee$ *HOL_Light_Import.INTER s t* $\neq$ *EMPTY*)

thm SETDIST_EQ_0_BOUNDED:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\vee$ *bounded t* $\longrightarrow$ (*setdist* (*s*, *t*) = (*0*::*real*)) = (*s* = *EMPTY* $\vee$ *t* = *EMPTY* $\vee$ *HOL_Light_Import.INTER* (*closure s*) (*closure t*) $\neq$ *EMPTY*)

thm SETDIST_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *setdist* (*IMAGE* (*vector_add a*) *s*, *IMAGE* (*vector_add a*) *t*) = *setdist* (*s*, *t*)

thm SETDIST_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\longrightarrow$ *setdist* (*IMAGE f s*, *IMAGE f t*) = *setdist* (*s*, *t*)

thm SETDIST_UNIQUE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *d*::*real*. *IN a s* $\wedge$ *IN b t* $\wedge$ *distance* (*a*, *b*) = *d* $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y t* $\longrightarrow$ *distance* (*a*, *b*) $\leq$ *distance* (*x*, *y*)) $\longrightarrow$ *setdist* (*s*, *t*) = *d*

thm SETDIST_CLOSEST_POINT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *setdist* (*INSERT a EMPTY*, *s*) = *distance* (*a*, *closest_point s a*)

thm SEPARATION_CLOSURES:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.INTER\ s\ (closure\ t) = EMPTY \land HOL\_Light\_Import.INTER\ t\ (closure\ s) = EMPTY \longrightarrow (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ DISJOINT\ u\ v \land HOL\_Light\_Import.open\ u \land HOL\_Light\_Import.open\ v \land SUBSET\ s\ u \land SUBSET\ t\ v)$

thm SEPARATION_NORMAL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \land HOL\_Light\_Import.closed\ t \land HOL\_Light\_Import.INTER\ s\ t = EMPTY \longrightarrow (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land HOL\_Light\_Import.open\ v \land SUBSET\ s\ u \land SUBSET\ t\ v \land HOL\_Light\_Import.INTER\ u\ v = EMPTY)$

thm SEPARATION_NORMAL_COMPACT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \land HOL\_Light\_Import.closed\ t \land HOL\_Light\_Import.INTER\ s\ t = EMPTY \longrightarrow (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land compact\ (closure\ u) \land HOL\_Light\_Import.open\ v \land SUBSET\ s\ u \land SUBSET\ t\ v \land HOL\_Light\_Import.INTER\ u\ v = EMPTY)$

thm SEPARATION_HAUSDORFF:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ x \neq y \longrightarrow (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land HOL\_Light\_Import.open\ v \land IN\ x\ u \land IN\ y\ v \land HOL\_Light\_Import.INTER\ u\ v = EMPTY)$

thm SEPARATION_T2:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (x \neq y) = (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land HOL\_Light\_Import.open\ v \land IN\ x\ u \land IN\ y\ v \land HOL\_Light\_Import.INTER\ u\ v = EMPTY)$

thm SEPARATION_T1:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (x \neq y) = (\exists (u::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land HOL\_Light\_Import.open\ v \land IN\ x\ u \land \neg\ IN\ y\ u \land \neg\ IN\ x\ v \land IN\ y\ v)$

thm SEPARATION_T0:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (x \neq y) = (\exists\ u::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ u \land IN\ x\ u \neq IN\ y\ u)$

thm DIST_PASTECART_CANCEL_conjunct1:

$\forall (x::(real,\ ?'b::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ y'::(real,\ ?'a::type)\ cart.\ distance\ (pastecart\ x\ y,\ pastecart\ x\ y') = distance\ (y,\ y')$

thm DIST_PASTECART_CANCEL_conjunct0:

$\forall (x::(real, ?'b::type)\ cart)\ (x'::(real, ?'b::type)\ cart)\ y::(real, ?'a::type)\ cart.$ $distance\ (pastecart\ x\ y,\ pastecart\ x'\ y) = distance\ (x,\ x')$

thm CLOSED_COMPACT_PROJECTION:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, (?'b::type, ?'a::type)\ finite\_sum)$ $cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.closed\ t \longrightarrow HOL\_Light\_Import.closed$ $(GSPEC\ (\lambda GEN\%PVAR\%778::(real, ?'a::type)\ cart.\ \exists\, y::(real, ?'a::type)\ cart.$ $SETSPEC\ GEN\%PVAR\%778\ (\exists\, x::(real, ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ (pastecart$ $x\ y)\ t)\ y))$

thm SEGMENT_SYM_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ open\_segment\ (a,\ b) =$ $open\_segment\ (b,\ a)$

thm MIDPOINT_IN_SEGMENT_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ IN\ (midpoint\ (a,\ b))\ (open\_segment$ $(a,\ b)) = (a \neq b)$

thm MIDPOINT_IN_SEGMENT_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ IN\ (midpoint\ (a,\ b))\ (closed\_segment$ $[(a,\ b)])$

thm URYSOHN_STRONG:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (t::(real, ?'b::type)\ cart \Rightarrow bool)\ (a::(real,$ $?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ s \wedge HOL\_Light\_Import.closed$ $t \wedge HOL\_Light\_Import.INTER\ s\ t = EMPTY \wedge a \neq b \longrightarrow (\exists\, f::(real, ?'b::type)$ $cart \Rightarrow (real, ?'a::type)\ cart.\ continuous\_on\ f\ HOL\_Light\_Import.UNIV \wedge$ $(\forall\, x::(real, ?'b::type)\ cart.\ IN\ (f\ x)\ (closed\_segment\ [(a,\ b)])) \wedge (\forall\, x::(real,$ $?'b::type)\ cart.\ (f\ x = a) = IN\ x\ s) \wedge (\forall\, x::(real, ?'b::type)\ cart.\ (f\ x = b) =$ $IN\ x\ t))$

thm URYSOHN:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (t::(real, ?'b::type)\ cart \Rightarrow bool)\ (a::(real,$ $?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ s \wedge HOL\_Light\_Import.closed$ $t \wedge HOL\_Light\_Import.INTER\ s\ t = EMPTY \longrightarrow (\exists\, f::(real, ?'b::type)\ cart$ $\Rightarrow (real, ?'a::type)\ cart.\ continuous\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall\, x::(real,$ $?'b::type)\ cart.\ IN\ (f\ x)\ (closed\_segment\ [(a,\ b)])) \wedge (\forall\, x::(real, ?'b::type)\ cart.$ $IN\ x\ s \longrightarrow f\ x = a) \wedge (\forall\, x::(real, ?'b::type)\ cart.\ IN\ x\ t \longrightarrow f\ x = b))$

thm TIETZE_STEP:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, unit)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)$ $B::real.\ (0::real) < B \wedge HOL\_Light\_Import.closed\ s \wedge continuous\_on\ f\ s \wedge$ $(\forall\, x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) \leq B) \longrightarrow (\exists\, g::(real,$ $?'a::type)\ cart \Rightarrow (real, unit)\ cart.\ continuous\_on\ g\ HOL\_Light\_Import.UNIV$

∧ (∀ x::(real, ?'a::type) cart. vector_norm (g x) ≤ B / real_of_nat (3::nat)) ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ vector_norm (vector_sub (f x) (g x)) ≤ real_of_nat (2::nat) / real_of_nat (3::nat) ∗ B))

thm TIETZE:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) (s::(real, ?'a::type) cart ⇒ bool) B::real. (0::real) ≤ B ∧ HOL_Light_Import.closed s ∧ continuous_on f s ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ vector_norm (f x) ≤ B) ⟶ (∃ g::(real, ?'a::type) cart ⇒ (real, unit) cart. continuous_on g HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ g x = f x) ∧ (∀ x::(real, ?'a::type) cart. vector_norm (g x) ≤ B))

thm TIETZE_CLOSED_INTERVAL_1:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) (s::(real, ?'a::type) cart ⇒ bool) (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b ∧ HOL_Light_Import.closed s ∧ continuous_on f s ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ IN (f x) (closed_interval [(a, b)])) ⟶ (∃ g::(real, ?'a::type) cart ⇒ (real, unit) cart. continuous_on g HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ g x = f x) ∧ (∀ x::(real, ?'a::type) cart. IN (g x) (closed_interval [(a, b)])))

thm TIETZE_OPEN_INTERVAL_1:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) (s::(real, ?'a::type) cart ⇒ bool) (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.drop a < HOL_Light_Import.drop b ∧ HOL_Light_Import.closed s ∧ continuous_on f s ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ IN (f x) (open_interval (a, b))) ⟶ (∃ g::(real, ?'a::type) cart ⇒ (real, unit) cart. continuous_on g HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ g x = f x) ∧ (∀ x::(real, ?'a::type) cart. IN (g x) (open_interval (a, b))))

thm TIETZE_UNBOUNDED_1:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ continuous_on f s ⟶ (∃ g::(real, ?'a::type) cart ⇒ (real, unit) cart. continuous_on g HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ g x = f x))

thm TIETZE_CLOSED_INTERVAL:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closed_interval [(a, b)] ≠ EMPTY ∧ HOL_Light_Import.closed s ∧ continuous_on f s ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ IN (f x) (closed_interval [(a, b)])) ⟶ (∃ g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on g HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g x = f x) ∧ (∀ x::(real, ?'b::type) cart. IN (g x) (closed_interval [(a, b)])))

thm TIETZE_OPEN_INTERVAL:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\wedge$ *HOL_Light_Import.closed s* $\wedge$ *continuous_on f s* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* $\longrightarrow$ *IN* (*f x*) (*open_interval* (*a*, *b*))) $\longrightarrow$ ($\exists$ *g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *continuous_on g HOL_Light_Import.UNIV* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* $\longrightarrow$ *g x* = *f x*) $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN* (*g x*) (*open_interval* (*a*, *b*))))

thm TIETZE_UNBOUNDED:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.closed s* $\wedge$ *continuous_on f s* $\longrightarrow$ ($\exists$ *g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *continuous_on g HOL_Light_Import.UNIV* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* $\longrightarrow$ *g x* = *f x*))

thm COUNTABLE_INTEGER:

COUNTABLE integer

thm CARD_EQ_INTEGER:

=_*c integer HOL_Light_Import.UNIV*

thm COUNTABLE_RATIONAL:

COUNTABLE rational

thm CARD_EQ_RATIONAL:

=_*c rational HOL_Light_Import.UNIV*

thm COUNTABLE_INTEGER_COORDINATES:

COUNTABLE (*GSPEC* (λ*GEN%PVAR%785*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%785* ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *integer* ($ *x i*)) *x*))

thm COUNTABLE_RATIONAL_COORDINATES:

COUNTABLE (*GSPEC* (λ*GEN%PVAR%787*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%787* ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *rational* ($ *x i*)) *x*))

thm CLOSURE_DYADIC_RATIONALS:

closure (*GSPEC* (λ*GEN%PVAR%788*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*n*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%788* ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *integer* ($ *x i*)) (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))^n) *x*))) = *HOL_Light_Import.UNIV*

thm CLOSURE_RATIONAL_COORDINATES:

closure (*GSPEC* (λ*GEN%PVAR%790*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%790* ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *rational* ($ *x i*)) *x*)) = *HOL_Light_Import.UNIV*

thm CLOSURE_DYADIC_RATIONALS_IN_OPEN_SET:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ *closure* (*HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%791*::(*real*, *?'a*::*type*) *cart*. ∃ (*n*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%791* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *integer* ($ *x i*)) (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))^n) *x*)))) = *closure s*

thm CLOSURE_RATIONALS_IN_OPEN_SET:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ *closure* (*HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%792*::(*real*, *?'a*::*type*) *cart*. ∃ (*n*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%792* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *integer* ($ *x i*)) (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))^n) *x*)))) = *closure s*

thm UNIV_SECOND_COUNTABLE:

∃ *b*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *COUNTABLE b* ∧ (∀ *c*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN c b* ⟶ *HOL_Light_Import.open c*) ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ (∃ *u*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET u b* ∧ *s* = *UNIONS u*))

thm UNIV_SECOND_COUNTABLE_SEQUENCE:

∃ *b*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ (*m*::*nat*) *n*::*nat*. (*b m* = *b n*) = (*m* = *n*)) ∧ (∀ *n*::*nat*. *HOL_Light_Import.open* (*b n*)) ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ (∃ *k*::*nat* ⇒ *bool*. *s* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%799*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%799* (*IN n k*) (*b n*)))))

thm SUBSET_SECOND_COUNTABLE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *b*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *COUNTABLE b* ∧ (∀ *c*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN c b* ⟶ *c* ≠ *EMPTY* ∧ *open_in* (*subtopology euclidean s*) *c*) ∧ (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *open_in* (*subtopology euclidean s*) *t* ⟶ (∃ *u*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET u b* ∧ *t* = *UNIONS u*))

thm SEPARABLE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *COUNTABLE t* ∧ *SUBSET t s* ∧ *SUBSET s* (*closure t*)

thm OPEN_SET_RATIONAL_COORDINATES:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *s* ≠ *EMPTY* ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *rational* ($ *x i*)))

thm OPEN_COUNTABLE_UNION_CLOSED_INTERVALS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ (∃ *D*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *COUNTABLE D* ∧ (∀ *i*::(*real*, *?'a*::*type*) *cart*

$\Rightarrow$ *bool. IN i D* $\longrightarrow$ *SUBSET i s* $\wedge$ ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *i* = *closed_interval* [(*a*, *b*)])) $\wedge$ *UNIONS D* = *s*)

thm OPEN_COUNTABLE_UNION_OPEN_INTERVALS:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\longrightarrow$ ($\exists$ *D*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *COUNTABLE D* $\wedge$ ($\forall$ *i*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN i D* $\longrightarrow$ *SUBSET i s* $\wedge$ ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *i* = *open_interval* (*a*, *b*))) $\wedge$ *UNIONS D* = *s*)

thm LINDELOF:

$\forall$ *f*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. ($\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN s f* $\longrightarrow$ *HOL_Light_Import.open s*) $\longrightarrow$ ($\exists$ *f'*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *SUBSET f' f* $\wedge$ *COUNTABLE f'* $\wedge$ *UNIONS f'* = *UNIONS f*)

thm COUNTABLE_DISJOINT_OPEN_SUBSETS:

$\forall$ *f*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. ($\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN s f* $\longrightarrow$ *HOL_Light_Import.open s*) $\wedge$ *pairwise DISJOINT f* $\longrightarrow$ *COUNTABLE f*

thm BROUWER_REDUCTION_THEOREM_GEN:

$\forall$ (*P*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *n*::*nat*. *HOL_Light_Import.closed* (*f n*) $\wedge$ *P* (*f n*)) $\wedge$ ($\forall$ *n*::*nat*. *SUBSET* (*f* (*Suc n*)) (*f n*)) $\longrightarrow$ *P* (*INTERS* (*GSPEC* (λ*GEN%PVAR%811*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%811* (*IN n HOL_Light_Import.UNIV*) (*f n*))))) $\wedge$ *HOL_Light_Import.closed s* $\wedge$ *P s* $\longrightarrow$ ($\exists$ *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET t s* $\wedge$ *HOL_Light_Import.closed t* $\wedge$ *P t* $\wedge$ ($\forall$ *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET u s* $\wedge$ *HOL_Light_Import.closed u* $\wedge$ *P u* $\longrightarrow$ $\neg$ *PSUBSET u t*))

thm BROUWER_REDUCTION_THEOREM:

$\forall$ (*P*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *n*::*nat*. *compact* (*f n*) $\wedge$ *f n* $\neq$ *EMPTY* $\wedge$ *P* (*f n*)) $\wedge$ ($\forall$ *n*::*nat*. *SUBSET* (*f* (*Suc n*)) (*f n*)) $\longrightarrow$ *P* (*INTERS* (*GSPEC* (λ*GEN%PVAR%812*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%812* (*IN n HOL_Light_Import.UNIV*) (*f n*))))) $\wedge$ *compact s* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ *P s* $\longrightarrow$ ($\exists$ *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET t s* $\wedge$ *compact t* $\wedge$ *t* $\neq$ *EMPTY* $\wedge$ *P t* $\wedge$ ($\forall$ *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET u s* $\wedge$ *HOL_Light_Import.closed u* $\wedge$ *u* $\neq$ *EMPTY* $\wedge$ *P u* $\longrightarrow$ $\neg$ *PSUBSET u t*))

thm SUBSEQUENCE_DIAGONALIZATION_LEMMA:

$\forall$ *P*::*nat* $\Rightarrow$ (*nat* $\Rightarrow$ *?'a*::*type*) $\Rightarrow$ *bool*. ($\forall$ (*i*::*nat*) *r*::*nat* $\Rightarrow$ *?'a*::*type*. $\exists$ *k*::*nat* $\Rightarrow$ *nat*. ($\forall$ (*m*::*nat*) *n*::*nat*. *m* < *n* $\longrightarrow$ *k m* < *k n*) $\wedge$ *P i* (*r* $\circ$ *k*)) $\wedge$ ($\forall$ (*i*::*nat*) (*r*::*nat* $\Rightarrow$ *?'a*::*type*) (*k1*::*nat* $\Rightarrow$ *nat*) (*k2*::*nat* $\Rightarrow$ *nat*) *N*::*nat*. *P i* (*r* $\circ$ *k1*) $\wedge$ ($\forall$ *j*$\geq$*N*. $\exists$ *j'*$\geq$*j*. *k2 j* = *k1 j'*) $\longrightarrow$ *P i* (*r* $\circ$ *k2*)) $\longrightarrow$ ($\forall$ *r*::*nat* $\Rightarrow$ *?'a*::*type*.

$\exists k::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow k\ m < k\ n) \wedge (\forall i::nat.\ P\ i\ (r \circ k)))$

thm FUNCTION_CONVERGENT_SUBSEQUENCE:

$\forall (f::nat \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ M::real.\ COUNTABLE\ s \wedge (\forall (n::nat)\ x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ n\ x) \leq M) \longrightarrow (\exists k::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow k\ m < k\ n) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (\exists l::(real,\ ?'a::type)\ cart.\ --> (\lambda n::nat.\ f\ (k\ n)\ x)\ l\ sequentially)))$

thm ARZELA_ASCOLI:

$\forall (f::nat \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ M::real.\ compact\ s \wedge (\forall (n::nat)\ x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ n\ x) \leq M) \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ e::real.\ IN\ x\ s \wedge (0::real) < e \longrightarrow (\exists d>0::real.\ \forall (n::nat)\ y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \wedge vector\_norm\ (vector\_sub\ x\ y) < d \longrightarrow vector\_norm\ (vector\_sub\ (f\ n\ x)\ (f\ n\ y)) < e)) \longrightarrow (\exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ g\ s \wedge (\exists r::nat \Rightarrow nat.\ (\forall (m::nat)\ n::nat.\ m < n \longrightarrow r\ m < r\ n) \wedge (\forall e>0::real.\ \exists N::nat.\ \forall (n::nat)\ x::(real,\ ?'b::type)\ cart.\ N \leq n \wedge IN\ x\ s \longrightarrow vector\_norm\ (vector\_sub\ (f\ (r\ n)\ x)\ (g\ x)) < e)))$

thm TRANSLATION_GALOIS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ (s = IMAGE\ (vector\_add\ a)\ t) = (t = IMAGE\ (vector\_add\ (vector\_neg\ a))\ s)$

thm TRANSLATION_EQ_IMP:

$\forall P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ P\ (IMAGE\ (vector\_add\ a)\ s) = P\ s) = (\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ P\ s \longrightarrow P\ (IMAGE\ (vector\_add\ a)\ s))$

thm DIM_HYPERPLANE:

$\forall a::(real,\ ?'a::type)\ cart.\ a \neq vec\ (0::nat) \longrightarrow dim\ (GSPEC\ (\lambda GEN\%PVAR\%813::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%813\ (dot\ a\ x = (0::real))\ x)) = dimindex\ HOL\_Light\_Import.UNIV - (1::nat)$

thm LOWDIM_EQ_HYPERPLANE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dim\ s = dimindex\ HOL\_Light\_Import.UNIV - (1::nat) \longrightarrow (\exists a::(real,\ ?'a::type)\ cart.\ a \neq vec\ (0::nat) \wedge span\ s = GSPEC\ (\lambda GEN\%PVAR\%814::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%814\ (dot\ a\ x = (0::real))\ x))$

thm DIM_EQ_HYPERPLANE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (dim\ s = dimindex\ HOL\_Light\_Import.UNIV - (1::nat)) = (\exists a::(real,\ ?'a::type)\ cart.\ a \neq vec\ (0::nat) \wedge span\ s = GSPEC$

$(\lambda GEN\%PVAR\%815::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%815\ (dot\ a\ x = (0::real))\ x))$

thm DEF_affine:

$affine = (\lambda\_258701::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ \_258701 \wedge IN\ y\ \_258701 \wedge u + v = (1::real) \longrightarrow IN\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y))\ \_258701)$

thm affine:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ s \wedge IN\ y\ s \wedge u + v = (1::real) \longrightarrow IN\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y))\ s)$

thm AFFINE_ALT:

$affine\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) = (\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ u::real.\ IN\ x\ ?s \wedge IN\ y\ ?s \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - u)\ x)\ (\%\ u\ y))\ ?s)$

thm AFFINE_SCALING:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ affine\ s \longrightarrow affine\ (IMAGE\ (\%\ c)\ s)$

thm AFFINE_SCALING_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ c \neq (0::real) \longrightarrow affine\ (IMAGE\ (\%\ c)\ s) = affine\ s$

thm AFFINE_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ s \longrightarrow affine\ (IMAGE\ vector\_neg\ s)$

thm AFFINE_SUMS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge affine\ t \longrightarrow affine\ (GSPEC\ (\lambda GEN\%PVAR\%816::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%816\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm AFFINE_DIFFERENCES:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge affine\ t \longrightarrow affine\ (GSPEC\ (\lambda GEN\%PVAR\%817::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%817\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm AFFINE_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ (IMAGE\ (vector\_add\ a)\ s) = affine\ s$

thm AFFINE_TRANSLATION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ a::(real, ?'a::type)\ cart.\ affine\ s \longrightarrow affine\ (IMAGE\ (vector\_add\ a)\ s)$

thm AFFINE_AFFINITY:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (a::(real, ?'a::type)\ cart)\ c::real.\ affine\ s \longrightarrow affine\ (IMAGE\ (\lambda x::(real, ?'a::type)\ cart.\ vector\_add\ a\ (\%\ c\ x))\ s)$

thm AFFINE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ affine\ s \wedge linear\ f \longrightarrow affine\ (IMAGE\ f\ s)$

thm AFFINE_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow affine\ (IMAGE\ f\ s) = affine\ s$

thm AFFINE_EMPTY:

$affine\ EMPTY$

thm AFFINE_SING:

$\forall x::(real, ?'a::type)\ cart.\ affine\ (INSERT\ x\ EMPTY)$

thm AFFINE_UNIV:

$affine\ HOL\_Light\_Import.UNIV$

thm AFFINE_HYPERPLANE:

$\forall (a::(real, ?'a::type)\ cart)\ b::real.\ affine\ (GSPEC\ (\lambda GEN\%PVAR\%818::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%818\ (dot\ a\ x = b)\ x))$

thm AFFINE_INTERS:

$(\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ (?f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool) \longrightarrow affine\ s) \longrightarrow affine\ (INTERS\ ?f)$

thm AFFINE_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge affine\ t \longrightarrow affine\ (HOL\_Light\_Import.INTER\ s\ t)$

thm AFFINE_AFFINE_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ (hull\ affine\ s)$

thm AFFINE_HULL_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (hull\ affine\ s = s) = affine\ s$

thm IS_AFFINE_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s = (\exists t::(real, ?'a::type)\ cart \Rightarrow bool.\ s = hull\ affine\ t)$

thm AFFINE_HULL_UNIV:

hull affine HOL_Light_Import.UNIV = HOL_Light_Import.UNIV

thm AFFINE_HULLS_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s* (*hull affine t*) ∧ *SUBSET t* (*hull affine s*) ⟶ *hull affine s* = *hull affine t*

thm AFFINE_HULL_TRANSLATION:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *hull affine* (*IMAGE* (*vector_add a*) *s*) = *IMAGE* (*vector_add a*) (*hull affine s*)

thm AFFINE_HULL_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ⟶ *hull affine* (*IMAGE f s*) = *IMAGE f* (*hull affine s*)

thm IN_AFFINE_HULL_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'b*::*type*) *cart*. *linear f* ∧ *IN x* (*hull affine s*) ⟶ *IN* (*f x*) (*hull affine* (*IMAGE f s*))

thm IN_AFFINE_ADD_MUL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *d*::*real*. *affine s* ∧ *IN a s* ∧ *IN* (*vector_add a x*) *s* ⟶ *IN* (*vector_add a* (% *d x*)) *s*

thm IN_AFFINE_ADD_MUL_DIFF:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::*real*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *IN x s* ∧ *IN y s* ∧ *IN z s* ⟶ *IN* (*vector_add x* (% *a* (*vector_sub y z*))) *s*

thm IN_AFFINE_MUL_DIFF_ADD:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::*real*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *IN x s* ∧ *IN y s* ∧ *IN z s* ⟶ *IN* (*vector_add* (% *a* (*vector_sub x y*)) *z*) *s*

thm IN_AFFINE_SUB_MUL_DIFF:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::*real*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *IN x s* ∧ *IN y s* ∧ *IN z s* ⟶ *IN* (*vector_sub x* (% *a* (*vector_sub y z*))) *s*

thm AFFINE_DIFFS_SUBSPACE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *IN a s* ⟶ *subspace* (*GSPEC* (λ*GEN%PVAR%819*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%819* (*IN x s*) (*vector_sub x a*)))

thm AFFINE_VSUM:

$\forall$ (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) (*k*::*?'a*::*type* $\Rightarrow$ *bool*) (*u*::*?'a*::*type* $\Rightarrow$ *real*) *x*::*?'a*::*type* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*. *FINITE k* $\wedge$ *affine s* $\wedge$ *sum k u* = (*1*::*real*) $\wedge$ ($\forall$ *i*::*?'a*::*type*. *IN i k* $\longrightarrow$ *IN* (*x i*) *s*) $\longrightarrow$ *IN* (*vsum k* (λ*i*::*?'a*::*type*. % (*u i*) (*x i*))) *s*

thm AFFINE_INDEXED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affine s* = ($\forall$ (*k*::*nat*) (*u*::*nat* $\Rightarrow$ *real*) *x*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *k* $\longrightarrow$ *IN* (*x i*) *s*) $\wedge$ *sum* (*dotdot* (*1*::*nat*) *k*) *u* = (*1*::*real*) $\longrightarrow$ *IN* (*vsum* (*dotdot* (*1*::*nat*) *k*) (λ*i*::*nat*. % (*u i*) (*x i*))) *s*)

thm AFFINE_HULL_INDEXED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *hull affine s* = *GSPEC* (λ*GEN%PVAR%821*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%821* ($\exists$ (*k*::*nat*) (*u*::*nat* $\Rightarrow$ *real*) *x*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *k* $\longrightarrow$ *IN* (*x i*) *s*) $\wedge$ *sum* (*dotdot* (*1*::*nat*) *k*) *u* = (*1*::*real*) $\wedge$ *vsum* (*dotdot* (*1*::*nat*) *k*) (λ*i*::*nat*. % (*u i*) (*x i*)) = *y*) *y*)

thm AFFINE:

$\forall$ *V*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affine V* = ($\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ *SUBSET s V* $\wedge$ *sum s u* = (*1*::*real*) $\longrightarrow$ *IN* (*vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *V*)

thm AFFINE_EXPLICIT:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affine s* = ($\forall$ (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *FINITE t* $\wedge$ *SUBSET t s* $\wedge$ *sum t u* = (*1*::*real*) $\longrightarrow$ *IN* (*vsum t* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *s*)

thm AFFINE_HULL_EXPLICIT:

$\forall$ *p*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *hull affine p* = *GSPEC* (λ*GEN%PVAR%824*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%824* ($\exists$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ *SUBSET s p* $\wedge$ *sum s u* = (*1*::*real*) $\wedge$ *vsum s* (λ*v*::(*real*, *?'a*::*type*) *cart*. % (*u v*) *v*) = *y*) *y*)

thm AFFINE_HULL_EXPLICIT_ALT:

$\forall$ *p*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *hull affine p* = *GSPEC* (λ*GEN%PVAR%825*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%825* ($\exists$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *FINITE s* $\wedge$ *SUBSET s p* $\wedge$ *sum s u* = (*1*::*real*) $\wedge$ *vsum s* (λ*v*::(*real*, *?'a*::*type*) *cart*. % (*u v*) *v*) = *y*) *y*)

thm AFFINE_HULL_FINITE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *hull affine s* = *GSPEC* (λ*GEN%PVAR%827*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%827* ($\exists$ *u*::(*real*,

?'a::type) cart ⇒ real. sum s u = (1::real) ∧ vsum s (λv::(real, ?'a::type) cart. % (u v) v) = y) y)

thm AFFINE_HULL_EMPTY:

hull affine EMPTY = EMPTY

thm AFFINE_HULL_EQ_EMPTY:

∀ s::(real, ?'a::type) cart ⇒ bool. (hull affine s = EMPTY) = (s = EMPTY)

thm AFFINE_HULL_FINITE_STEP_GEN:

∀ P::(real, ?'a::type) cart ⇒ real ⇒ bool. (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x EMPTY ⟶ P x (u x)) ∧ sum EMPTY u = (?w::real) ∧ vsum EMPTY (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (?w = (0::real) ∧ ?y = vec (0::nat)) ∧ (FINITE (?s::(real, ?'a::type) cart ⇒ bool) ∧ (∀ y::real. IN (?a::(real, ?'a::type) cart) ?s ∧ P ?a y ⟶ P ?a (y / real_of_nat (2::nat))) ∧ (∀ (x::real) y::real. IN ?a ?s ∧ P ?a x ∧ P ?a y ⟶ P ?a (x + y)) ⟶ (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x (INSERT ?a ?s) ⟶ P x (u x)) ∧ sum (INSERT ?a ?s) u = ?w ∧ vsum (INSERT ?a ?s) (λx::(real, ?'a::type) cart. % (u x) x) = ?y) = (∃ (v::real) u::(real, ?'a::type) cart ⇒ real. P ?a v ∧ (∀ x::(real, ?'a::type) cart. IN x ?s ⟶ P x (u x)) ∧ sum ?s u = ?w − v ∧ vsum ?s (λx::(real, ?'a::type) cart. % (u x) x) = vector_sub ?y (% v ?a)))

thm AFFINE_HULL_FINITE_STEP:

(∃ u::(real, ?'a::type) cart ⇒ real. sum EMPTY u = (?w::real) ∧ vsum EMPTY (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (?w = (0::real) ∧ ?y = vec (0::nat)) ∧ (FINITE (?s::(real, ?'a::type) cart ⇒ bool) ⟶ (∃ u::(real, ?'a::type) cart ⇒ real. sum (INSERT (?a::(real, ?'a::type) cart) ?s) u = ?w ∧ vsum (INSERT ?a ?s) (λx::(real, ?'a::type) cart. % (u x) x) = ?y) = (∃ (v::real) u::(real, ?'a::type) cart ⇒ real. sum ?s u = ?w − v ∧ vsum ?s (λx::(real, ?'a::type) cart. % (u x) x) = vector_sub ?y (% v ?a)))

thm AFFINE_HULL_FINITE_STEP_conjunct1:

FINITE (?s::(real, ?'a::type) cart ⇒ bool) ⟶ (∃ u::(real, ?'a::type) cart ⇒ real. sum (INSERT (?a::(real, ?'a::type) cart) ?s) u = (?w::real) ∧ vsum (INSERT ?a ?s) (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (∃ (v::real) u::(real, ?'a::type) cart ⇒ real. sum ?s u = ?w − v ∧ vsum ?s (λx::(real, ?'a::type) cart. % (u x) x) = vector_sub ?y (% v ?a))

thm AFFINE_HULL_FINITE_STEP_conjunct0:

(∃ u::(real, ?'a::type) cart ⇒ real. sum EMPTY u = (?w::real) ∧ vsum EMPTY (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (?w = (0::real) ∧ ?y = vec (0::nat))

thm AFFINE_HULL_2:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. hull affine (INSERT a (INSERT b EMPTY)) = GSPEC (λGEN%PVAR%830::(real, ?'a::type) cart. ∃ (u::real)

v::real. SETSPEC GEN%PVAR%830 (*u* + *v* = (*1::real*)) (*vector_add* (*%* *u* *a*) (*%* *v* *b*)))

thm AFFINE_HULL_2_ALT:

∀ (*a::(real, ?'a::type) cart*) *b::(real, ?'a::type) cart. hull affine* (*INSERT a* (*INSERT b EMPTY*)) = *GSPEC* (λ*GEN%PVAR%831::(real, ?'a::type) cart.* ∃ *u::real. SETSPEC GEN%PVAR%831* (*IN u HOL_Light_Import.UNIV*) (*vector_add a* (*%* *u* (*vector_sub b a*))))

thm AFFINE_HULL_3:

hull affine (*INSERT* (*?a::(real, ?'a::type) cart*) (*INSERT* (*?b::(real, ?'a::type) cart*) (*INSERT* (*?c::(real, ?'a::type) cart*) *EMPTY*))) = *GSPEC* (λ*GEN%PVAR%832::(real, ?'a::type) cart.* ∃ (*u::real*) (*v::real*) *w::real. SETSPEC GEN%PVAR%832* (*u* + (*v* + *w*) = (*1::real*)) (*vector_add* (*%* *u* *?a*) (*vector_add* (*%* *v* *?b*) (*%* *w* *?c*))))

thm AFFINE_HULL_INSERT_SUBSET_SPAN:

∀ (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* ⇒ *bool. SUBSET* (*hull affine* (*INSERT a s*)) (*GSPEC* (λ*GEN%PVAR%834::(real, ?'a::type) cart.* ∃ *v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%834* (*IN v* (*span* (*GSPEC* (λ*GEN%PVAR%833::(real, ?'a::type) cart.* ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%833* (*IN x s*) (*vector_sub x a*))))) (*vector_add a v*)))

thm AFFINE_HULL_INSERT_SPAN:

∀ (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* ⇒ *bool.* ¬ *IN a s* ⟶ *hull affine* (*INSERT a s*) = *GSPEC* (λ*GEN%PVAR%836::(real, ?'a::type) cart.* ∃ *v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%836* (*IN v* (*span* (*GSPEC* (λ*GEN%PVAR%835::(real, ?'a::type) cart.* ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%835* (*IN x s*) (*vector_sub x a*))))) (*vector_add a v*))

thm AFFINE_HULL_SPAN:

∀ (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* ⇒ *bool. IN a s* ⟶ *hull affine s* = *GSPEC* (λ*GEN%PVAR%838::(real, ?'a::type) cart.* ∃ *v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%838* (*IN v* (*span* (*GSPEC* (λ*GEN%PVAR%837::(real, ?'a::type) cart.* ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%837* (*IN x* (*DELETE s a*)) (*vector_sub x a*))))) (*vector_add a v*))

thm DIFFS_AFFINE_HULL_SPAN:

∀ (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* ⇒ *bool. IN a s* ⟶ *GSPEC* (λ*GEN%PVAR%839::(real, ?'a::type) cart.* ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%839* (*IN x* (*hull affine s*)) (*vector_sub x a*)) = *span* (*GSPEC* (λ*GEN%PVAR%840::(real, ?'a::type) cart.* ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%840* (*IN x s*) (*vector_sub x a*)))

thm AFFINE_HULL_SING:

∀ *a::(real, ?'a::type) cart. hull affine* (*INSERT a EMPTY*) = *INSERT a EMPTY*

thm AFFINE_HULL_EQ_SING:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ (hull\ affine\ s = INSERT\ a\ EMPTY) = (s = INSERT\ a\ EMPTY)$

thm DEF_convex:

$convex = (\lambda\_262576::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ \_262576 \wedge IN\ y\ \_262576 \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow IN\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y))\ \_262576)$

thm convex:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ s \wedge IN\ y\ s \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow IN\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y))\ s)$

thm CONVEX_ALT:

$convex\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) = (\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ u::real.\ IN\ x\ ?s \wedge IN\ y\ ?s \wedge (0::real) \leq u \wedge u \leq (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - u)\ x)\ (\%\ u\ y))\ ?s)$

thm IN_CONVEX_SET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ u::real.\ convex\ s \wedge IN\ a\ s \wedge IN\ b\ s \wedge (0::real) \leq u \wedge u \leq (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - u)\ a)\ (\%\ u\ b))\ s$

thm CONVEX_EMPTY:

$convex\ EMPTY$

thm CONVEX_SING:

$\forall a::(real,\ ?'a::type)\ cart.\ convex\ (INSERT\ a\ EMPTY)$

thm CONVEX_UNIV:

$convex\ HOL\_Light\_Import.UNIV$

thm CONVEX_INTERS:

$(\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ (?f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool) \longrightarrow convex\ s) \longrightarrow convex\ (INTERS\ ?f)$

thm CONVEX_INTER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow convex\ (HOL\_Light\_Import.INTER\ s\ t)$

thm CONVEX_HULLS_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ (hull\ convex\ t) \wedge SUBSET\ t\ (hull\ convex\ s) \longrightarrow hull\ convex\ s = hull\ convex\ t$

thm CONVEX_HALFSPACE_LE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%843*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%843* (*dot a x* $\leq$ *b*) *x*))

thm CONVEX_HALFSPACE_COMPONENT_LE:

$\forall$ (*a*::*real*) *k*::*nat*. *convex* (*GSPEC* (λ*GEN%PVAR%844*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%844* ($ *x k* $\leq$ *a*) *x*))

thm CONVEX_HALFSPACE_GE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%847*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%847* (*b* $\leq$ *dot a x*) *x*))

thm CONVEX_HALFSPACE_COMPONENT_GE:

$\forall$ (*a*::*real*) *k*::*nat*. *convex* (*GSPEC* (λ*GEN%PVAR%848*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%848* (*a* $\leq$ $ *x k*) *x*))

thm CONVEX_HYPERPLANE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%852*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%852* (*dot a x* = *b*) *x*))

thm CONVEX_STANDARD_HYPERPLANE:

$\forall$ (*k*::*nat*) *a*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%853*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%853* ($ *x k* = *a*) *x*))

thm CONVEX_HALFSPACE_LT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%854*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%854* (*dot a x* < *b*) *x*))

thm CONVEX_HALFSPACE_COMPONENT_LT:

$\forall$ (*a*::*real*) *k*::*nat*. *convex* (*GSPEC* (λ*GEN%PVAR%855*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%855* ($ *x k* < *a*) *x*))

thm CONVEX_HALFSPACE_GT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%856*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%856* (*b* < *dot a x*) *x*))

thm CONVEX_HALFSPACE_COMPONENT_GT:

$\forall$ (*a*::*real*) *k*::*nat*. *convex* (*GSPEC* (λ*GEN%PVAR%857*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%857* (*a* < $ *x k*) *x*))

thm CONVEX_POSITIVE_ORTHANT:

convex (*GSPEC* (λ*GEN%PVAR%858*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%858* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ (*0*::*real*) ≤ $ *x i*) *x*))

thm LIMPT_OF_CONVEX:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *IN x s* ⟶ *limit_point_of x s* = (*s* ≠ *INSERT x EMPTY*)

thm TRIVIAL_LIMIT_WITHIN_CONVEX:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *IN x s* ⟶ *trivial_limit* (*within* (*at x*) *s*) = (*s* = *INSERT x EMPTY*)

thm CONVEX_VSUM:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*k*::*?'a*::*type* ⇒ *bool*) (*u*::*?'a*::*type* ⇒ *real*) *x*::*?'a*::*type* ⇒ (*real*, *?'b*::*type*) *cart*. *FINITE k* ∧ *convex s* ∧ *sum k u* = (*1*::*real*) ∧ (∀ *i*::*?'a*::*type*. *IN i k* ⟶ (*0*::*real*) ≤ *u i* ∧ *IN* (*x i*) *s*) ⟶ *IN* (*vsum k* (λ*i*::*?'a*::*type*. % (*u i*) (*x i*))) *s*

thm CONVEX_INDEXED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* = (∀ (*k*::*nat*) (*u*::*nat* ⇒ *real*) *x*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*. (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *k* ⟶ (*0*::*real*) ≤ *u i* ∧ *IN* (*x i*) *s*) ∧ *sum* (*dotdot* (*1*::*nat*) *k*) *u* = (*1*::*real*) ⟶ *IN* (*vsum* (*dotdot* (*1*::*nat*) *k*) (λ*i*::*nat*. % (*u i*) (*x i*))) *s*)

thm CONVEX_EXPLICIT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* = (∀ (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *FINITE t* ∧ *SUBSET t s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x t* ⟶ (*0*::*real*) ≤ *u x*) ∧ *sum t u* = (*1*::*real*) ⟶ *IN* (*vsum t* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *s*)

thm CONVEX:

∀ *V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex V* = (∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *FINITE s* ∧ *s* ≠ *EMPTY* ∧ *SUBSET s V* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *u x*) ∧ *sum s u* = (*1*::*real*) ⟶ *IN* (*vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *V*)

thm CONVEX_FINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *convex s* = (∀ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *u x*) ∧ *sum s u* = (*1*::*real*) ⟶ *IN* (*vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *s*)

thm DEF_conic:

conic = (λ*_263819*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∀ (*x*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. *IN x _263819* ∧ (*0*::*real*) ≤ *c* ⟶ *IN* (% *c x*) *_263819*)

thm conic:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. conic s = (\forall (x::(real, ?'a::type) cart) c::real. IN x s \wedge (0::real) \leq c \longrightarrow IN (\% c x) s)$

thm SUBSPACE_IMP_CONIC:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. subspace s \longrightarrow conic s$

thm CONIC_EMPTY:

conic EMPTY

thm CONIC_UNIV:

conic HOL_Light_Import.UNIV

thm CONIC_INTERS:

$(\forall s::(real, ?'a::type) cart \Rightarrow bool. IN s (?f::((real, ?'a::type) cart \Rightarrow bool) \Rightarrow bool) \longrightarrow conic s) \longrightarrow conic (INTERS ?f)$

thm CONIC_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. conic s \wedge linear f \longrightarrow conic (IMAGE f s)$

thm CONIC_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. linear f \wedge (\forall (x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f x = f y \longrightarrow x = y) \longrightarrow conic (IMAGE f s) = conic s$

thm CONIC_CONIC_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. conic (hull conic s)$

thm CONIC_HULL_EQ:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. (hull conic s = s) = conic s$

thm CONIC_NEGATIONS:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. conic s \longrightarrow conic (IMAGE vector\_neg s)$

thm CONIC_SPAN:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. conic (span s)$

thm CONIC_HULL_EXPLICIT:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. hull conic s = GSPEC (\lambda GEN\%PVAR\%860::(real, ?'a::type) cart. \exists (c::real) x::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%860 ((0::real) \leq c \wedge IN x s) (\% c x))$

thm CONIC_HULL_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. linear f \longrightarrow hull conic (IMAGE f s) = IMAGE f (hull conic s)$

thm CONVEX_CONIC_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow convex\ (hull\ conic\ s)$

thm CONIC_HALFSPACE_LE:

$\forall a::(real, ?'a::type)\ cart.\ conic\ (GSPEC\ (\lambda GEN\%PVAR\%865::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%865\ (dot\ a\ x \leq (0::real))\ x))$

thm CONIC_HALFSPACE_GE:

$\forall a::(real, ?'a::type)\ cart.\ conic\ (GSPEC\ (\lambda GEN\%PVAR\%866::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%866\ ((0::real) \leq dot\ a\ x)\ x))$

thm CONIC_HULL_EMPTY:

$hull\ conic\ EMPTY = EMPTY$

thm CONIC_CONTAINS_0:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ conic\ s \longrightarrow IN\ (vec\ (0::nat))\ s = (s \neq EMPTY)$

thm CONIC_HULL_EQ_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (hull\ conic\ s = EMPTY) = (s = EMPTY)$

thm CONIC_SUMS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ conic\ s \wedge conic\ t \longrightarrow conic\ (GSPEC\ (\lambda GEN\%PVAR\%867::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%867\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm CONIC_POSITIVE_ORTHANT:

$conic\ (GSPEC\ (\lambda GEN\%PVAR\%868::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%868\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (0::real) \leq \$\ x\ i)\ x))$

thm SEPARATE_CLOSED_CONES:

$\forall (c::(real, ?'a::type)\ cart \Rightarrow bool)\ d::(real, ?'a::type)\ cart \Rightarrow bool.\ conic\ c \wedge HOL\_Light\_Import.closed\ c \wedge conic\ d \wedge HOL\_Light\_Import.closed\ d \wedge SUBSET\ (HOL\_Light\_Import.INTER\ c\ d)\ (INSERT\ (vec\ (0::nat))\ EMPTY) \longrightarrow (\exists e>0::real.\ \forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ c \wedge IN\ y\ d \longrightarrow e * max\ (vector\_norm\ x)\ (vector\_norm\ y) \leq distance\ (x, y))$

thm CONTINUOUS_ON_COMPACT_SURFACE_PROJECTION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (v::(real, ?'a::type)\ cart \Rightarrow bool)\ d::(real, ?'a::type)\ cart \Rightarrow real.\ compact\ s \wedge SUBSET\ s\ (DELETE\ v\ (vec\ (0::nat))) \wedge conic\ v \wedge (\forall (x::(real, ?'a::type)\ cart)\ k::real.\ IN\ x\ (DELETE\ v\ (vec\ (0::nat))) \longrightarrow ((0::real) < k \wedge IN\ (\%\ k\ x)\ s) = (d\ x = k)) \longrightarrow continuous\_on\ (\lambda x::(real, ?'a::type)\ cart.\ \%\ (d\ x)\ x)\ (DELETE\ v\ (vec\ (0::nat)))$

thm DEF_affine_dependent:

$affine\_dependent = (\lambda\_264854::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_264854 \wedge IN\ x\ (hull\ affine\ (DELETE\ \_264854\ x)))$

thm affine_dependent:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\_dependent\ s = (\exists\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ x\ (hull\ affine\ (DELETE\ s\ x)))$

thm AFFINE_DEPENDENT_EXPLICIT:

$\forall\, p::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\_dependent\ p = (\exists\,(s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ u::(real,\ ?'a::type)\ cart \Rightarrow real.\ FINITE\ s \wedge SUBSET\ s\ p \wedge sum\ s\ u = (0::real) \wedge (\exists\, v::(real,\ ?'a::type)\ cart.\ IN\ v\ s \wedge u\ v \neq (0::real)) \wedge vsum\ s\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = vec\ (0::nat))$

thm AFFINE_DEPENDENT_EXPLICIT_FINITE:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow affine\_dependent\ s = (\exists\, u::(real,\ ?'a::type)\ cart \Rightarrow real.\ sum\ s\ u = (0::real) \wedge (\exists\, v::(real,\ ?'a::type)\ cart.\ IN\ v\ s \wedge u\ v \neq (0::real)) \wedge vsum\ s\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = vec\ (0::nat))$

thm AFFINE_DEPENDENT_TRANSLATION_EQ:

$\forall\,(a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\_dependent\ (IMAGE\ (vector\_add\ a)\ s) = affine\_dependent\ s$

thm AFFINE_DEPENDENT_TRANSLATION:

$\forall\,(s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ affine\_dependent\ s \longrightarrow affine\_dependent\ (IMAGE\ (vector\_add\ a)\ s)$

thm AFFINE_DEPENDENT_LINEAR_IMAGE_EQ:

$\forall\,(f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall\,(x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow affine\_dependent\ (IMAGE\ f\ s) = affine\_dependent\ s$

thm AFFINE_DEPENDENT_LINEAR_IMAGE:

$\forall\,(f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall\,(x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \wedge affine\_dependent\ s \longrightarrow affine\_dependent\ (IMAGE\ f\ s)$

thm AFFINE_DEPENDENT_MONO:

$\forall\,(s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\_dependent\ s \wedge SUBSET\ s\ t \longrightarrow affine\_dependent\ t$

thm AFFINE_INDEPENDENT_EMPTY:

$\neg\ affine\_dependent\ EMPTY$

thm AFFINE_INDEPENDENT_1:

$\forall\, a::(real,\ ?'a::type)\ cart.\ \neg\ affine\_dependent\ (INSERT\ a\ EMPTY)$

thm AFFINE_INDEPENDENT_2:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ \neg\ affine\_dependent\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm AFFINE_INDEPENDENT_SUBSET:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ t \wedge SUBSET\ s\ t \longrightarrow \neg\ affine\_dependent\ s$

thm AFFINE_INDEPENDENT_DELETE:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ \neg\ affine\_dependent\ s \longrightarrow \neg\ affine\_dependent\ (DELETE\ s\ a)$

thm DEF_coplanar:

$coplanar = (\lambda\_265772::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ SUBSET\ \_265772\ (hull\ affine\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))))$

thm coplanar:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ coplanar\ s = (\exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ (hull\ affine\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))))$

thm COLLINEAR_AFFINE_HULL:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ s = (\exists\, (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ (hull\ affine\ (INSERT\ u\ (INSERT\ v\ EMPTY))))$

thm COLLINEAR_IMP_COPLANAR:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ s \longrightarrow coplanar\ s$

thm COPLANAR_SMALL:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \wedge CARD\ s \leq (3::nat) \longrightarrow coplanar\ s$

thm COPLANAR_EMPTY:

$coplanar\ EMPTY$

thm COPLANAR_SING:

$\forall\, a::(real,\ ?'a::type)\ cart.\ coplanar\ (INSERT\ a\ EMPTY)$

thm COPLANAR_2:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ coplanar\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm COPLANAR_3:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *coplanar* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))

thm COLLINEAR_AFFINE_HULL_COLLINEAR:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *collinear* (*hull affine s*) = *collinear s*

thm COPLANAR_AFFINE_HULL_COPLANAR:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *coplanar* (*hull affine s*) = *coplanar s*

thm COPLANAR_TRANSLATION_EQ:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *coplanar* (*IMAGE* (*vector_add a*) *s*) = *coplanar s*

thm COPLANAR_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *coplanar s* $\longrightarrow$ *coplanar* (*IMAGE* (*vector_add a*) *s*)

thm COPLANAR_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *coplanar s* $\wedge$ *linear f* $\longrightarrow$ *coplanar* (*IMAGE f s*)

thm COPLANAR_LINEAR_IMAGE_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *coplanar* (*IMAGE f s*) = *coplanar s*

thm COPLANAR_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *coplanar t* $\wedge$ *SUBSET s t* $\longrightarrow$ *coplanar s*

thm AFFINE_HULL_3_IMP_COLLINEAR:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *IN c* (*hull affine* (*INSERT a* (*INSERT b EMPTY*))) $\longrightarrow$ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))

thm COLLINEAR_3_AFFINE_HULL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) = *IN c* (*hull affine* (*INSERT a* (*INSERT b EMPTY*)))

thm COLLINEAR_3_EQ_AFFINE_DEPENDENT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) = (*a* = *b* $\vee$ *a* = *c* $\vee$ *b* = *c* $\vee$ *affine_dependent* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))))

thm AFFINE_DEPENDENT_IMP_COLLINEAR_3:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *affine_dependent* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) ⟶ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))

thm COLLINEAR_3_IN_AFFINE_HULL:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *v1* ≠ *v0* ⟶ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) = *IN x* (*hull affine* (*INSERT v0* (*INSERT v1 EMPTY*)))

thm CONVEX_CONNECTED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ⟶ *connected s*

thm CONNECTED_UNIV:

connected HOL_Light_Import.UNIV

thm CONNECTED_COMPONENT_UNIV:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *connected_component HOL_Light_Import.UNIV x* = *HOL_Light_Import.UNIV*

thm CONNECTED_COMPONENT_EQ_UNIV:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. (*connected_component s x* = *HOL_Light_Import.UNIV*) = (*s* = *HOL_Light_Import.UNIV*)

thm CLOPEN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*HOL_Light_Import.closed s* ∧ *HOL_Light_Import.open s*) = (*s* = *EMPTY* ∨ *s* = *HOL_Light_Import.UNIV*)

thm FINITE_IMP_NOT_OPEN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ *s* ≠ *EMPTY* ⟶ ¬ *HOL_Light_Import.open s*

thm OPEN_IMP_INFINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ *s* = *EMPTY* ∨ *INFINITE s*

thm EMPTY_INTERIOR_FINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *interior s* = *EMPTY*

thm FRONTIER_NOT_EMPTY:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *s* ≠ *EMPTY* ∧ *s* ≠ *HOL_Light_Import.UNIV* ⟶ *frontier s* ≠ *EMPTY*

thm FRONTIER_EQ_EMPTY:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*frontier s* = *EMPTY*) = (*s* = *EMPTY* ∨ *s* = *HOL_Light_Import.UNIV*)

thm NOT_INTERVAL_UNIV_conjunct1:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ open\_interval\ (a,\ b) \neq HOL\_Light\_Import.UNIV$

thm NOT_INTERVAL_UNIV_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq HOL\_Light\_Import.UNIV$

thm EQ_INTERVAL_conjunct3:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (open\_interval\ (a,\ b) = open\_interval\ (c,\ d)) = (open\_interval\ (a,\ b) = EMPTY \wedge open\_interval\ (c,\ d) = EMPTY \vee a = c \wedge b = d)$

thm EQ_INTERVAL_conjunct2:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (open\_interval\ (a,\ b) = closed\_interval\ [(c,\ d)]) = (open\_interval\ (a,\ b) = EMPTY \wedge closed\_interval\ [(c,\ d)] = EMPTY)$

thm EQ_INTERVAL_conjunct1:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (closed\_interval\ [(a,\ b)] = open\_interval\ (c,\ d)) = (closed\_interval\ [(a,\ b)] = EMPTY \wedge open\_interval\ (c,\ d) = EMPTY)$

thm EQ_INTERVAL_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (closed\_interval\ [(a,\ b)] = closed\_interval\ [(c,\ d)]) = (closed\_interval\ [(a,\ b)] = EMPTY \wedge closed\_interval\ [(c,\ d)] = EMPTY \vee a = c \wedge b = d)$

thm EQ_INTERVAL:

$(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (closed\_interval\ [(a,\ b)] = closed\_interval\ [(c,\ d)]) = (closed\_interval\ [(a,\ b)] = EMPTY \wedge closed\_interval\ [(c,\ d)] = EMPTY \vee a = c \wedge b = d)) \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (closed\_interval\ [(a,\ b)] = open\_interval\ (c,\ d)) = (closed\_interval\ [(a,\ b)] = EMPTY \wedge open\_interval\ (c,\ d) = EMPTY)) \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (open\_interval\ (a,\ b) = closed\_interval\ [(c,\ d)]) = (open\_interval\ (a,\ b) = EMPTY \wedge closed\_interval\ [(c,\ d)] = EMPTY)) \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ (open\_interval\ (a,\ b) = open\_interval\ (c,\ d)) = (open\_interval\ (a,\ b) = EMPTY \wedge open\_interval\ (c,\ d) = EMPTY \vee a = c \wedge b = d))$

thm CONNECTED_CHAIN:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. ($\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *compact s* ∧ *connected s*) ∧ ($\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *IN t f* ⟶ *SUBSET s t* ∨ *SUBSET t s*) ⟶ *connected* (*INTERS f*)

thm CONNECTED_CHAIN_GEN:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. ($\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *HOL_Light_Import.closed s* ∧ *connected s*) ∧ ($\exists s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *compact s*) ∧ ($\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ∧ *IN t f* ⟶ *SUBSET s t* ∨ *SUBSET t s*) ⟶ *connected* (*INTERS f*)

thm CONNECTED_NEST:

$\forall s$::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. ($\forall n$::*nat*. *compact* (*s n*) ∧ *connected* (*s n*)) ∧ ($\forall$ (*m*::*nat*) *n*::*nat*. *m* ≤ *n* ⟶ *SUBSET* (*s n*) (*s m*)) ⟶ *connected* (*INTERS* (*GSPEC* (λ*GEN%PVAR%874*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. $\exists n$::*nat*. *SETSPEC GEN%PVAR%874* (*IN n HOL_Light_Import.UNIV*) (*s n*))))

thm CONNECTED_NEST_GEN:

$\forall s$::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. ($\forall n$::*nat*. *HOL_Light_Import.closed* (*s n*) ∧ *connected* (*s n*)) ∧ ($\exists n$::*nat*. *compact* (*s n*)) ∧ ($\forall$ (*m*::*nat*) *n*::*nat*. *m* ≤ *n* ⟶ *SUBSET* (*s n*) (*s m*)) ⟶ *connected* (*INTERS* (*GSPEC* (λ*GEN%PVAR%875*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. $\exists n$::*nat*. *SETSPEC GEN%PVAR%875* (*IN n HOL_Light_Import.UNIV*) (*s n*))))

thm SUBSET_BALLS_conjunct3:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. *SUBSET* (*cball* (*a*, *r*)) (*cball* (*a'*, *r'*)) = (*distance* (*a*, *a'*) + *r* ≤ *r'* ∨ *r* < (*0*::*real*))

thm SUBSET_BALLS_conjunct2:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. *SUBSET* (*cball* (*a*, *r*)) (*ball* (*a'*, *r'*)) = (*distance* (*a*, *a'*) + *r* < *r'* ∨ *r* < (*0*::*real*))

thm SUBSET_BALLS_conjunct1:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. *SUBSET* (*ball* (*a*, *r*)) (*cball* (*a'*, *r'*)) = (*distance* (*a*, *a'*) + *r* ≤ *r'* ∨ *r* ≤ (*0*::*real*))

thm SUBSET_BALLS_conjunct0:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. *SUBSET* (*ball* (*a*, *r*)) (*ball* (*a'*, *r'*)) = (*distance* (*a*, *a'*) + *r* ≤ *r'* ∨ *r* ≤ (*0*::*real*))

thm EQ_BALLS:

($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. (*ball* (*a*, *r*) = *ball* (*a'*, *r'*)) = (*a* = *a'* ∧ *r* = *r'* ∨ *r* ≤ (*0*::*real*) ∧ *r'* ≤ (*0*::*real*))) ∧ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*a'*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. (*ball* (*a*,

$r) = cball\ (a', r')) = (r \leq (0::real) \wedge r' < (0::real))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (a'::(real, ?'a::type)\ cart)\ (r::real)\ r'::real.\ (cball\ (a, r) = ball\ (a', r')) = (r < (0::real) \wedge r' \leq (0::real))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (a'::(real, ?'a::type)\ cart)\ (r::real)\ r'::real.\ (cball\ (a, r) = cball\ (a', r')) = (a = a' \wedge r = r' \vee r < (0::real) \wedge r' < (0::real)))$

thm DEF_convex_on:

$convex\_on = (\lambda(\_270762::(real, ?'a::type)\ cart \Rightarrow real)\ \_270763::(real, ?'a::type)\ cart \Rightarrow bool.\ \forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ \_270763 \wedge IN\ y\ \_270763 \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow \_270762\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y)) \leq u * \_270762\ x + v * \_270762\ y)$

thm convex_on:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ f::(real, ?'a::type)\ cart \Rightarrow real.\ convex\_on\ f\ s = (\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ (u::real)\ v::real.\ IN\ x\ s \wedge IN\ y\ s \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow f\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y)) \leq u * f\ x + v * f\ y)$

thm CONVEX_ON_SUBSET:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\_on\ f\ t \wedge SUBSET\ s\ t \longrightarrow convex\_on\ f\ s$

thm CONVEX_ADD:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (f::(real, ?'a::type)\ cart \Rightarrow real)\ g::(real, ?'a::type)\ cart \Rightarrow real.\ convex\_on\ f\ s \wedge convex\_on\ g\ s \longrightarrow convex\_on\ (\lambda x::(real, ?'a::type)\ cart.\ f\ x + g\ x)\ s$

thm CONVEX_CMUL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (c::real)\ f::(real, ?'a::type)\ cart \Rightarrow real.\ (0::real) \leq c \wedge convex\_on\ f\ s \longrightarrow convex\_on\ (\lambda x::(real, ?'a::type)\ cart.\ c * f\ x)\ s$

thm CONVEX_LOWER:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ convex\_on\ f\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge (0::real) \leq (?u::real) \wedge (0::real) \leq (?v::real) \wedge ?u + ?v = (1::real) \longrightarrow f\ (vector\_add\ (\%\ ?u\ x)\ (\%\ ?v\ y)) \leq max\ (f\ x)\ (f\ y)$

thm CONVEX_LOCAL_GLOBAL_MINIMUM:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ convex\_on\ f\ s \wedge IN\ x\ t \wedge HOL\_Light\_Import.open\ t \wedge SUBSET\ t\ s \wedge (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ t \longrightarrow f\ x \leq f\ y) \longrightarrow (\forall y::(real, ?'a::type)\ cart.\ IN\ y\ s \longrightarrow f\ x \leq f\ y)$

thm CONVEX_DISTANCE:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. convex_on (λx::(real, ?'a::type) cart. distance (a, x)) s

thm CONVEX_BALL:

∀ (x::(real, ?'a::type) cart) e::real. convex (ball (x, e))

thm CONNECTED_BALL:

∀ (x::(real, ?'a::type) cart) e::real. connected (ball (x, e))

thm CONVEX_CBALL:

∀ (x::(real, ?'a::type) cart) e::real. convex (cball (x, e))

thm CONNECTED_CBALL:

∀ (x::(real, ?'a::type) cart) e::real. connected (cball (x, e))

thm FRONTIER_OF_CONNECTED_COMPONENT_SUBSET:

∀ (s::(real, ?'b::type) cart ⇒ bool) (c::?'a::type) x::(real, ?'b::type) cart. SUBSET (frontier (connected_component s x)) (frontier s)

thm FRONTIER_OF_COMPONENTS_SUBSET:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. IN c (components s) ⟶ SUBSET (frontier c) (frontier s)

thm FRONTIER_OF_COMPONENTS_CLOSED_COMPLEMENT:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ IN c (components (DIFF HOL_Light_Import.UNIV s)) ⟶ SUBSET (frontier c) s

thm SURA_BURA_COMPACT:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. compact s ∧ IN c (components s) ⟶ c = INTERS (GSPEC (λGEN%PVAR%877::(real, ?'a::type) cart ⇒ bool. ∃ t::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%877 (SUBSET c t ∧ open_in (subtopology euclidean s) t ∧ closed_in (subtopology euclidean s) t) t))

thm SURA_BURA_CLOSED:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ IN c (components s) ∧ compact c ⟶ c = INTERS (GSPEC (λGEN%PVAR%880::(real, ?'a::type) cart ⇒ bool. ∃ k::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%880 (SUBSET c k ∧ compact k ∧ open_in (subtopology euclidean s) k) k))

thm CONVEX_SCALING:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::real. convex s ⟶ convex (IMAGE (% c) s)

thm CONVEX_SCALING_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ c \neq (0::real) \longrightarrow convex\ (IMAGE\ (\%\ c)\ s) = convex\ s$

thm CONVEX_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow convex\ (IMAGE\ vector\_neg\ s)$

thm CONVEX_SUMS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow convex\ (GSPEC\ (\lambda GEN\%PVAR\%881::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%881\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm CONVEX_DIFFERENCES:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow convex\ (GSPEC\ (\lambda GEN\%PVAR\%882::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%882\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_sub\ x\ y)))$

thm CONVEX_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ (IMAGE\ (vector\_add\ a)\ s) = convex\ s$

thm CONVEX_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ convex\ s \longrightarrow convex\ (IMAGE\ (vector\_add\ a)\ s)$

thm CONVEX_AFFINITY:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ c::real.\ convex\ s \longrightarrow convex\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ a\ (\%\ c\ x))\ s)$

thm CONVEX_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ convex\ s \wedge linear\ f \longrightarrow convex\ (IMAGE\ f\ s)$

thm CONVEX_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow convex\ (IMAGE\ f\ s) = convex\ s$

thm CONVEX_LINEAR_PREIMAGE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge convex\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) \longrightarrow convex\ (GSPEC\ (\lambda GEN\%PVAR\%883::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%883\ (IN\ (f\ x)\ ?s)\ x))$

thm CONVEX_CONVEX_HULL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ (hull\ convex\ s)$

thm CONVEX_HULL_EQ:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. (hull convex s = s) = convex s$

thm IS_CONVEX_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. convex s = (\exists t::(real, ?'a::type) cart \Rightarrow bool. s = hull convex t)$

thm CONVEX_HULL_UNIV:

$hull convex HOL\_Light\_Import.UNIV = HOL\_Light\_Import.UNIV$

thm BOUNDED_CONVEX_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. bounded s \longrightarrow bounded (hull convex s)$

thm BOUNDED_CONVEX_HULL_EQ:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. bounded (hull convex s) = bounded s$

thm FINITE_IMP_BOUNDED_CONVEX_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. FINITE s \longrightarrow bounded (hull convex s)$

thm CONVEX_HULL_EMPTY:

$hull convex EMPTY = EMPTY$

thm CONVEX_HULL_EQ_EMPTY:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. (hull convex s = EMPTY) = (s = EMPTY)$

thm CONVEX_HULL_SING:

$\forall a::(real, ?'a::type) cart. hull convex (INSERT a EMPTY) = INSERT a EMPTY$

thm CONVEX_HULL_EQ_SING:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) a::(real, ?'a::type) cart. (hull convex s = INSERT a EMPTY) = (s = INSERT a EMPTY)$

thm CONVEX_HULL_INSERT:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) a::(real, ?'a::type) cart. s \neq EMPTY \longrightarrow hull convex (INSERT a s) = GSPEC (\lambda GEN\%PVAR\%884::(real, ?'a::type) cart. \exists x::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%884 (\exists (u::real) (v::real) b::(real, ?'a::type) cart. (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \wedge IN b (hull convex s) \wedge x = vector\_add (\% u a) (\% v b)) x)$

thm CONVEX_HULL_INSERT_ALT:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) a::(real, ?'a::type) cart. hull convex (INSERT a s) = (if s = EMPTY then INSERT a EMPTY else GSPEC (\lambda GEN\%PVAR\%885::(real, ?'a::type) cart. \exists (u::real) x::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%885 ((0::real) \leq u \wedge u \leq (1::real) \wedge IN x (hull convex s)) (vector\_add (\% ((1::real) - u) a) (\% u x))))$

thm CONVEX_HULL_INDEXED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ convex\ s = GSPEC\ (\lambda GEN\%PVAR\%886::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%886\ (\exists (k::nat)\ (u::nat \Rightarrow real)\ x::nat \Rightarrow (real, ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq k \longrightarrow (0::real) \leq u\ i \wedge IN\ (x\ i)\ s) \wedge sum\ (dotdot\ (1::nat)\ k)\ u = (1::real) \wedge vsum\ (dotdot\ (1::nat)\ k)\ (\lambda i::nat.\ \%\ (u\ i)\ (x\ i)) = y)\ y)$

thm CONVEX_HULL_EXPLICIT:

$\forall p::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ convex\ p = GSPEC\ (\lambda GEN\%PVAR\%888::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%888\ (\exists (s::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow real.\ FINITE\ s \wedge SUBSET\ s\ p \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq u\ x) \wedge sum\ s\ u = (1::real) \wedge vsum\ s\ (\lambda v::(real, ?'a::type)\ cart.\ \%\ (u\ v)\ v) = y)\ y)$

thm CONVEX_HULL_FINITE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ convex\ s = GSPEC\ (\lambda GEN\%PVAR\%890::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%890\ (\exists u::(real, ?'a::type)\ cart \Rightarrow real.\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq u\ x) \wedge sum\ s\ u = (1::real) \wedge vsum\ s\ (\lambda x::(real, ?'a::type)\ cart.\ \%\ (u\ x)\ x) = y)\ y)$

thm CONVEX_HULL_UNION_EXPLICIT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow hull\ convex\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.UNION\ s\ (HOL\_Light\_Import.UNION\ t\ (GSPEC\ (\lambda GEN\%PVAR\%891::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ (u::real)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%891\ (IN\ x\ s \wedge IN\ y\ t \wedge (0::real) \leq u \wedge u \leq (1::real))\ (vector\_add\ (\%\ ((1::real) - u)\ x)\ (\%\ u\ y)))))$

thm CONVEX_HULL_UNION_NONEMPTY_EXPLICIT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge s \neq EMPTY \wedge convex\ t \wedge t \neq EMPTY \longrightarrow hull\ convex\ (HOL\_Light\_Import.UNION\ s\ t) = GSPEC\ (\lambda GEN\%PVAR\%892::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ (u::real)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%892\ (IN\ x\ s \wedge IN\ y\ t \wedge (0::real) \leq u \wedge u \leq (1::real))\ (vector\_add\ (\%\ ((1::real) - u)\ x)\ (\%\ u\ y)))$

thm CONVEX_HULL_UNION_UNIONS:

$\forall (f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (UNIONS\ f) \wedge f \neq EMPTY \longrightarrow hull\ convex\ (HOL\_Light\_Import.UNION\ s\ (UNIONS\ f)) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%893::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists t::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%893\ (IN\ t\ f)\ (hull\ convex\ (HOL\_Light\_Import.UNION\ s\ t))))$

thm CONVEX_HULL_FINITE_STEP:

$(\exists u::(real, ?'a::type)\ cart \Rightarrow real.\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ EMPTY \longrightarrow (0::real) \leq u\ x) \wedge sum\ EMPTY\ u = (?w::real) \wedge vsum\ EMPTY\ (\lambda x::(real,$

?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (?w = (0::real) ∧ ?y = vec (0::nat)) ∧ (FINITE (?s::(real, ?'a::type) cart ⇒ bool) ⟶ (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x (INSERT (?a::(real, ?'a::type) cart) ?s) ⟶ (0::real) ≤ u x) ∧ sum (INSERT ?a ?s) u = ?w ∧ vsum (INSERT ?a ?s) (λx::(real, ?'a::type) cart. % (u x) x) = ?y) = (∃ v≥0::real. ∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x ?s ⟶ (0::real) ≤ u x) ∧ sum ?s u = ?w − v ∧ vsum ?s (λx::(real, ?'a::type) cart. % (u x) x) = vector_sub ?y (% v ?a)))

thm CONVEX_HULL_FINITE_STEP_conjunct1:

FINITE (?s::(real, ?'a::type) cart ⇒ bool) ⟶ (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x (INSERT (?a::(real, ?'a::type) cart) ?s) ⟶ (0::real) ≤ u x) ∧ sum (INSERT ?a ?s) u = (?w::real) ∧ vsum (INSERT ?a ?s) (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (∃ v≥0::real. ∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x ?s ⟶ (0::real) ≤ u x) ∧ sum ?s u = ?w − v ∧ vsum ?s (λx::(real, ?'a::type) cart. % (u x) x) = vector_sub ?y (% v ?a))

thm CONVEX_HULL_FINITE_STEP_conjunct0:

(∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x EMPTY ⟶ (0::real) ≤ u x) ∧ sum EMPTY u = (?w::real) ∧ vsum EMPTY (λx::(real, ?'a::type) cart. % (u x) x) = (?y::(real, ?'a::type) cart)) = (?w = (0::real) ∧ ?y = vec (0::nat))

thm CONVEX_HULL_2:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. hull convex (INSERT a (INSERT b EMPTY)) = GSPEC (λGEN%PVAR%894::(real, ?'a::type) cart. ∃ (u::real) v::real. SETSPEC GEN%PVAR%894 ((0::real) ≤ u ∧ (0::real) ≤ v ∧ u + v = (1::real)) (vector_add (% u a) (% v b)))

thm CONVEX_HULL_2_ALT:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. hull convex (INSERT a (INSERT b EMPTY)) = GSPEC (λGEN%PVAR%895::(real, ?'a::type) cart. ∃ u::real. SETSPEC GEN%PVAR%895 ((0::real) ≤ u ∧ u ≤ (1::real)) (vector_add a (% u (vector_sub b a))))

thm CONVEX_HULL_3:

hull convex (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) EMPTY))) = GSPEC (λGEN%PVAR%896::(real, ?'a::type) cart. ∃ (u::real) (v::real) w::real. SETSPEC GEN%PVAR%896 ((0::real) ≤ u ∧ (0::real) ≤ v ∧ (0::real) ≤ w ∧ u + (v + w) = (1::real)) (vector_add (% u ?a) (vector_add (% v ?b) (% w ?c))))

thm CONVEX_HULL_3_ALT:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) c::(real, ?'a::type) cart. hull convex (INSERT a (INSERT b (INSERT c EMPTY))) = GSPEC (λGEN%PVAR%897::(real,

?'a::type) cart. ∃(u::real) v::real. SETSPEC GEN%PVAR%897 ((0::real) ≤ u ∧ (0::real) ≤ v ∧ u + v ≤ (1::real)) (vector_add a (vector_add (% u (vector_sub b a)) (% v (vector_sub c a)))))

thm CONVEX_HULL_SUMS:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. hull convex (GSPEC (λGEN%PVAR%898::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%898 (IN x s ∧ IN y t) (vector_add x y))) = GSPEC (λGEN%PVAR%899::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%899 (IN x (hull convex s) ∧ IN y (hull convex t)) (vector_add x y))

thm SUBSPACE_IMP_AFFINE:

∀s::(real, ?'a::type) cart ⇒ bool. subspace s ⟶ affine s

thm AFFINE_IMP_CONVEX:

∀s::(real, ?'a::type) cart ⇒ bool. affine s ⟶ convex s

thm SUBSPACE_IMP_CONVEX:

∀s::(real, ?'a::type) cart ⇒ bool. subspace s ⟶ convex s

thm AFFINE_HULL_SUBSET_SPAN:

∀s::(real, ?'a::type) cart ⇒ bool. SUBSET (hull affine s) (span s)

thm CONVEX_HULL_SUBSET_SPAN:

∀s::(real, ?'a::type) cart ⇒ bool. SUBSET (hull convex s) (span s)

thm CONVEX_HULL_SUBSET_AFFINE_HULL:

∀s::(real, ?'a::type) cart ⇒ bool. SUBSET (hull convex s) (hull affine s)

thm COLLINEAR_CONVEX_HULL_COLLINEAR:

∀s::(real, ?'a::type) cart ⇒ bool. collinear (hull convex s) = collinear s

thm AFFINE_SPAN:

∀s::(real, ?'a::type) cart ⇒ bool. affine (span s)

thm CONVEX_SPAN:

∀s::(real, ?'a::type) cart ⇒ bool. convex (span s)

thm AFFINE_EQ_SUBSPACE:

∀s::(real, ?'a::type) cart ⇒ bool. IN (vec (0::nat)) s ⟶ affine s = subspace s

thm AFFINE_IMP_SUBSPACE:

∀s::(real, ?'a::type) cart ⇒ bool. affine s ∧ IN (vec (0::nat)) s ⟶ subspace s

thm AFFINE_HULL_EQ_SPAN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* (*vec* (*0*::*nat*)) (*hull affine s*) ⟶ *hull affine s* = *span s*

thm CLOSED_AFFINE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *affine s* ⟶ *HOL_Light_Import.closed s*

thm CLOSED_AFFINE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed* (*hull affine s*)

thm CLOSURE_SUBSET_AFFINE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*closure s*) (*hull affine s*)

thm AFFINE_HULL_CLOSURE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *hull affine* (*closure s*) = *hull affine s*

thm AFFINE_HULL_EQ_SPAN_EQ:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*hull affine s* = *span s*) = *IN* (*vec* (*0*::*nat*)) (*hull affine s*)

thm AFFINE_DEPENDENT_IMP_DEPENDENT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *affine_dependent s* ⟶ *dependent s*

thm DEPENDENT_AFFINE_DEPENDENT_CASES:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dependent s* = (*affine_dependent s* ∨ *IN* (*vec* (*0*::*nat*)) (*hull affine s*))

thm DEPENDENT_IMP_AFFINE_DEPENDENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *dependent* (*GSPEC* (λ*GEN%PVAR%900*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%900* (*IN x s*) (*vector_sub x a*))) ∧ ¬ *IN a s* ⟶ *affine_dependent* (*INSERT a s*)

thm AFFINE_DEPENDENT_BIGGERSET:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*FINITE s* ⟶ *dimindex HOL_Light_Import.UNIV* + (*2*::*nat*) ≤ *CARD s*) ⟶ *affine_dependent s*

thm AFFINE_DEPENDENT_BIGGERSET_GENERAL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*FINITE s* ⟶ *dim s* + (*2*::*nat*) ≤ *CARD s*) ⟶ *affine_dependent s*

thm AFFINE_INDEPENDENT_IMP_FINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *FINITE s*

thm AFFINE_INDEPENDENT_CARD_LE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *CARD s* ≤ *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*)

thm AFFINE_INDEPENDENT_CONVEX_AFFINE_HULL:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. \neg affine\_dependent s \wedge SUBSET t s \longrightarrow hull convex t = HOL\_Light\_Import.INTER (hull affine t) (hull convex s)$

thm DISJOINT_AFFINE_HULL:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) (t::(real, ?'a::type) cart \Rightarrow bool) u::(real, ?'a::type) cart \Rightarrow bool. \neg affine\_dependent s \wedge SUBSET t s \wedge SUBSET u s \wedge DISJOINT t u \longrightarrow DISJOINT (hull affine t) (hull affine u)$

thm AFFINE_INDEPENDENT_SPAN_EQ:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. \neg affine\_dependent s \wedge CARD s = dimindex HOL\_Light\_Import.UNIV + (1::nat) \longrightarrow hull affine s = HOL\_Light\_Import.UNIV$

thm AFFINE_INDEPENDENT_SPAN_GT:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. \neg affine\_dependent s \wedge dimindex HOL\_Light\_Import.UNIV < CARD s \longrightarrow hull affine s = HOL\_Light\_Import.UNIV$

thm EMPTY_INTERIOR_AFFINE_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. FINITE s \wedge CARD s \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow interior (hull affine s) = EMPTY$

thm EMPTY_INTERIOR_CONVEX_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. FINITE s \wedge CARD s \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow interior (hull convex s) = EMPTY$

thm AFFINE_DEPENDENT_CHOOSE:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) a::(real, ?'a::type) cart. \neg affine\_dependent s \longrightarrow affine\_dependent (INSERT a s) = (\neg IN a s \wedge IN a (hull affine s))$

thm AFFINE_INDEPENDENT_INSERT:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) a::(real, ?'a::type) cart. \neg affine\_dependent s \wedge \neg IN a (hull affine s) \longrightarrow \neg affine\_dependent (INSERT a s)$

thm AFFINE_HULL_EXPLICIT_UNIQUE:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) (u::(real, ?'a::type) cart \Rightarrow real) u'::(real, ?'a::type) cart \Rightarrow real. \neg affine\_dependent s \wedge sum s u = (1::real) \wedge sum s u' = (1::real) \wedge vsum s (\lambda x::(real, ?'a::type) cart. \% (u x) x) = vsum s (\lambda x::(real, ?'a::type) cart. \% (u' x) x) \longrightarrow (\forall x::(real, ?'a::type) cart. IN x s \longrightarrow u x = u' x)$

thm INDEPENDENT_IMP_AFFINE_DEPENDENT_0:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. independent s \longrightarrow \neg affine\_dependent (INSERT (vec (0::nat)) s)$

thm AFFINE_INDEPENDENT_STDBASIS:

$\neg$ *affine_dependent* (*INSERT* (*vec* (*0*::*nat*)) (*GSPEC* (λ*GEN%PVAR%906*::(*real*, *?'a*::*type*) *cart*. $\exists$ *i*::*nat*. *SETSPEC GEN%PVAR%906* ((*1*::*nat*) $\leq$ *i* $\land$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV*) (*basis i*))))

thm AFFINE_TRANSLATION_SUBSPACE:

$\forall$ *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*affine t* $\land$ *t* $\neq$ *EMPTY*) = ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *subspace s* $\land$ *t* = *IMAGE* (*vector_add a*) *s*)

thm AFFINE_TRANSLATION_UNIQUE_SUBSPACE:

$\forall$ *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*affine t* $\land$ *t* $\neq$ *EMPTY*) = ($\exists$!*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *subspace s* $\land$ *t* = *IMAGE* (*vector_add a*) *s*)

thm AFFINE_TRANSLATION_SUBSPACE_EXPLICIT:

$\forall$ (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *affine t* $\land$ *IN a t* $\longrightarrow$ *subspace* (*GSPEC* (λ*GEN%PVAR%907*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%907* (*IN x t*) (*vector_sub x a*))) $\land$ *t* = *IMAGE* (*vector_add a*) (*GSPEC* (λ*GEN%PVAR%908*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%908* (*IN x t*) (*vector_sub x a*)))

thm AFFINE_PARALLEL_SLICE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *affine s* $\longrightarrow$ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%912*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%912* (*dot a x* $\leq$ *b*) *x*)) = *EMPTY* $\lor$ *SUBSET s* (*GSPEC* (λ*GEN%PVAR%913*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%913* (*dot a x* $\leq$ *b*) *x*)) $\lor$ ($\exists$ (*a'*::(*real*, *?'a*::*type*) *cart*) *b'*::*real*. *a'* $\neq$ *vec* (*0*::*nat*) $\land$ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%914*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%914* (*dot a' x* $\leq$ *b'*) *x*)) = *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%915*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%915* (*dot a x* $\leq$ *b*) *x*)) $\land$ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%916*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%916* (*dot a' x* = *b'*) *x*)) = *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%917*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%917* (*dot a x* = *b*) *x*)) $\land$ ($\forall$ *w*::(*real*, *?'a*::*type*) *cart*. *IN w s* $\longrightarrow$ *IN* (*vector_add w a'*) *s*))

thm MAXIMAL_AFFINE_INDEPENDENT_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *b*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET b s* $\land$ $\neg$ *affine_dependent b* $\land$ ($\forall$ *b'*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET b b'* $\land$ *SUBSET b' s* $\land$ $\neg$ *affine_dependent b'* $\longrightarrow$ *b'* = *b*) $\longrightarrow$ *SUBSET s* (*hull affine b*)

thm MAXIMAL_AFFINE_INDEPENDENT_SUBSET_AFFINE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ b::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge SUBSET\ b\ s \wedge \neg\ affine\_dependent\ b \wedge (\forall b'::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ b\ b' \wedge SUBSET\ b'\ s \wedge \neg\ affine\_dependent\ b' \longrightarrow b' = b) \longrightarrow hull\ affine\ b = s$

thm EXTEND_TO_AFFINE_BASIS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \wedge SUBSET\ s\ u \longrightarrow (\exists t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ t \wedge SUBSET\ s\ t \wedge SUBSET\ t\ u \wedge hull\ affine\ t = hull\ affine\ u)$

thm AFFINE_BASIS_EXISTS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists b::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ b \wedge SUBSET\ b\ s \wedge hull\ affine\ b = hull\ affine\ s$

thm DEF_aff_dim:

$aff\_dim = (\lambda\_282728::(real, ?'a::type)\ cart \Rightarrow bool.\ SOME\ d::int.\ \exists b::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ b = hull\ affine\ \_282728 \wedge \neg\ affine\_dependent\ b \wedge int\ (CARD\ b) = d + int\ (1::nat))$

thm aff_dim:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ s = (SOME\ d::int.\ \exists b::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ b = hull\ affine\ s \wedge \neg\ affine\_dependent\ b \wedge int\ (CARD\ b) = d + int\ (1::nat))$

thm AFF_DIM:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists b::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ b = hull\ affine\ s \wedge \neg\ affine\_dependent\ b \wedge aff\_dim\ s = int\ (CARD\ b) - int\ (1::nat)$

thm AFF_DIM_EMPTY:

$aff\_dim\ EMPTY = -\ int\ (1::nat)$

thm AFF_DIM_AFFINE_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ (hull\ affine\ s) = aff\_dim\ s$

thm AFF_DIM_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ (IMAGE\ (vector\_add\ a)\ s) = aff\_dim\ s$

thm AFFINE_INDEPENDENT_CARD_DIM_DIFFS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ a::(real, ?'a::type)\ cart.\ \neg\ affine\_dependent\ s \wedge IN\ a\ s \longrightarrow CARD\ s = dim\ (GSPEC\ (\lambda GEN\%PVAR\%919::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%919\ (IN\ x\ s)\ (vector\_sub\ x\ a))) + (1::nat)$

thm AFF_DIM_DIM_AFFINE_DIFFS:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge IN\ a\ s \longrightarrow aff\_dim\ s = int\ (dim\ (GSPEC\ (\lambda GEN\%PVAR\%922::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%922\ (IN\ x\ s)\ (vector\_sub\ x\ a))))$

thm AFF_DIM_DIM_0:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ (vec\ (0::nat))\ (hull\ affine\ s) \longrightarrow aff\_dim\ s = int\ (dim\ s)$

thm AFF_DIM_DIM_SUBSPACE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ subspace\ s \longrightarrow aff\_dim\ s = int\ (dim\ s)$

thm AFF_DIM_LINEAR_IMAGE_LE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \longrightarrow aff\_dim\ (IMAGE\ f\ s) \leq aff\_dim\ s$

thm AFF_DIM_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow aff\_dim\ (IMAGE\ f\ s) = aff\_dim\ s$

thm AFF_DIM_AFFINE_INDEPENDENT:

$\forall b::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ b \longrightarrow aff\_dim\ b = int\ (CARD\ b) - int\ (1::nat)$

thm AFF_DIM_UNIQUE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ b::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ b = hull\ affine\ s \wedge \neg\ affine\_dependent\ b \longrightarrow aff\_dim\ s = int\ (CARD\ b) - int\ (1::nat)$

thm AFF_DIM_SING:

$\forall a::(real, ?'a::type)\ cart.\ aff\_dim\ (INSERT\ a\ EMPTY) = int\ (0::nat)$

thm AFF_DIM_LE_CARD:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow aff\_dim\ s \leq int\ (CARD\ s) - int\ (1::nat)$

thm AFF_DIM_GE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ -\ int\ (1::nat) \leq aff\_dim\ s$

thm AFF_DIM_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow aff\_dim\ s \leq aff\_dim\ t$

thm AFF_DIM_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ (hull\ convex\ s) = aff\_dim\ s$

thm AFF_DIM_2:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ aff\_dim\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = (if\ a = b\ then\ int\ (0::nat)\ else\ int\ (1::nat))$

thm AFF_DIM_EQ_MINUS1:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (aff\_dim\ s = -\ int\ (1::nat)) = (s = EMPTY)$

thm AFF_DIM_POS_LE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (int\ (0::nat) \leq aff\_dim\ s) = (s \neq EMPTY)$

thm AFF_DIM_EQ_0:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (aff\_dim\ s = int\ (0::nat)) = (\exists a::(real, ?'a::type)\ cart.\ s = INSERT\ a\ EMPTY)$

thm AFF_DIM_UNIV:

$aff\_dim\ HOL\_Light\_Import.UNIV = int\ (dimindex\ HOL\_Light\_Import.UNIV)$

thm AFF_DIM_EQ_AFFINE_HULL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge aff\_dim\ t \leq aff\_dim\ s \longrightarrow hull\ affine\ s = hull\ affine\ t$

thm AFF_DIM_SUMS_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge affine\ t \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow aff\_dim\ (GSPEC\ (\lambda GEN\%PVAR\%931::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%931\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y))) = aff\_dim\ s + aff\_dim\ t - aff\_dim\ (HOL\_Light\_Import.INTER\ s\ t)$

thm AFF_DIM_PSUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ PSUBSET\ (hull\ affine\ s)\ (hull\ affine\ t) \longrightarrow aff\_dim\ s < aff\_dim\ t$

thm AFF_DIM_EQ_FULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (aff\_dim\ s = int\ (dimindex\ HOL\_Light\_Import.UNIV)) = (hull\ affine\ s = HOL\_Light\_Import.UNIV)$

thm AFF_DIM_LE_UNIV:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ s \leq int\ (dimindex\ HOL\_Light\_Import.UNIV)$

thm AFFINE_INDEPENDENT_IFF_CARD:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (\neg\ affine\_dependent\ s) = (FINITE\ s \wedge aff\_dim\ s = int\ (CARD\ s) - int\ (1::nat))$

thm AFFINE_HULL_CONVEX_INTER_OPEN:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge HOL\_Light\_Import.open\ t \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow hull\ affine\ (HOL\_Light\_Import.INTER\ s\ t) = hull\ affine\ s$

thm CONVEX_AND_AFFINE_INTER_OPEN:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex s* $\wedge$ *affine t* $\wedge$ *HOL_Light_Import.open u* $\wedge$ *HOL_Light_Import.INTER s u* = *HOL_Light_Import.INTER t u* $\wedge$ *HOL_Light_Import.INTER s u* $\neq$ *EMPTY* $\longrightarrow$ *hull affine s* = *t*

thm AFFINE_HULL_OPEN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *hull affine s* = *HOL_Light_Import.UNIV*

thm AFFINE_HULL_NONEMPTY_INTERIOR:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *interior s* $\neq$ *EMPTY* $\longrightarrow$ *hull affine s* = *HOL_Light_Import.UNIV*

thm AFF_DIM_OPEN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *aff_dim s* = *int* (*dimindex HOL_Light_Import.UNIV*)

thm AFF_DIM_NONEMPTY_INTERIOR:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *interior s* $\neq$ *EMPTY* $\longrightarrow$ *aff_dim s* = *int* (*dimindex HOL_Light_Import.UNIV*)

thm SPAN_OPEN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *span s* = *HOL_Light_Import.UNIV*

thm DIM_OPEN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *dim s* = *dimindex HOL_Light_Import.UNIV*

thm AFF_DIM_INSERT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *aff_dim* (*INSERT a s*) = (*if IN a* (*hull affine s*) *then aff_dim s else aff_dim s* + *int* (*1*::*nat*))

thm AFFINE_BOUNDED_EQ_TRIVIAL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affine s* $\longrightarrow$ *bounded s* = (*s* = *EMPTY* $\vee$ ($\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *s* = *INSERT a EMPTY*))

thm AFFINE_BOUNDED_EQ_LOWDIM:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affine s* $\longrightarrow$ *bounded s* = (*aff_dim s* $\leq$ *int* (*0*::*nat*))

thm COLLINEAR_AFF_DIM:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *collinear s* = (*aff_dim s* $\leq$ *int* (*1*::*nat*))

thm CONVEX_HULL_CARATHEODORY_AFF_DIM:

$\forall$ *p::(real, ?'a::type) cart ⇒ bool. hull convex p = GSPEC (λGEN%PVAR%933::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%933 (∃ (s::(real, ?'a::type) cart ⇒ bool) u::(real, ?'a::type) cart ⇒ real. FINITE s ∧ SUBSET s p ∧ int (CARD s) ≤ aff_dim p + int (1::nat) ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ (0::real) ≤ u x) ∧ sum s u = (1::real) ∧ vsum s (λv::(real, ?'a::type) cart. % (u v) v) = y) y)*

thm CARATHEODORY_AFF_DIM:

∀ *p::(real, ?'a::type) cart ⇒ bool. hull convex p = GSPEC (λGEN%PVAR%934::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%934 (∃ s::(real, ?'a::type) cart ⇒ bool. FINITE s ∧ SUBSET s p ∧ int (CARD s) ≤ aff_dim p + int (1::nat) ∧ IN x (hull convex s)) x)*

thm CONVEX_HULL_CARATHEODORY:

∀ *p::(real, ?'a::type) cart ⇒ bool. hull convex p = GSPEC (λGEN%PVAR%935::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%935 (∃ (s::(real, ?'a::type) cart ⇒ bool) u::(real, ?'a::type) cart ⇒ real. FINITE s ∧ SUBSET s p ∧ CARD s ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ (0::real) ≤ u x) ∧ sum s u = (1::real) ∧ vsum s (λv::(real, ?'a::type) cart. % (u v) v) = y) y)*

thm CARATHEODORY:

∀ *p::(real, ?'a::type) cart ⇒ bool. hull convex p = GSPEC (λGEN%PVAR%936::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%936 (∃ s::(real, ?'a::type) cart ⇒ bool. FINITE s ∧ SUBSET s p ∧ CARD s ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ∧ IN x (hull convex s)) x)*

thm AFFINE_HULL_INTER:

∀ *(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. ¬ affine_dependent (HOL_Light_Import.UNION s t) ⟶ HOL_Light_Import.INTER (hull affine s) (hull affine t) = hull affine (HOL_Light_Import.INTER s t)*

thm CONVEX_HULL_INTER:

∀ *(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. ¬ affine_dependent (HOL_Light_Import.UNION s t) ⟶ HOL_Light_Import.INTER (hull convex s) (hull convex t) = hull convex (HOL_Light_Import.INTER s t)*

thm AFFINE_HULL_INTERS:

∀ *s::((real, ?'a::type) cart ⇒ bool) ⇒ bool. ¬ affine_dependent (UNIONS s) ⟶ hull affine (INTERS s) = INTERS (GSPEC (λGEN%PVAR%943::(real, ?'a::type) cart ⇒ bool. ∃ t::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%943 (IN t s) (hull affine t)))*

thm CONVEX_HULL_INTERS:

∀ *s::((real, ?'a::type) cart ⇒ bool) ⇒ bool. ¬ affine_dependent (UNIONS s) ⟶ hull convex (INTERS s) = INTERS (GSPEC (λGEN%PVAR%948::(real,*

?'a::type) cart ⇒ bool. ∃ t::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%948 (IN t s) (hull convex t)))

thm IN_CONVEX_HULL_EXCHANGE:

∀ (s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) x::(real, ?'a::type) cart. IN a (hull convex s) ∧ IN x (hull convex s) ⟶ (∃ b::(real, ?'a::type) cart. IN b s ∧ IN x (hull convex (INSERT a (DELETE s b))))

thm IN_CONVEX_HULL_EXCHANGE_UNIQUE:

∀ (s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) (t'::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) x::(real, ?'a::type) cart. ¬ affine_dependent s ∧ IN a (hull convex s) ∧ SUBSET t s ∧ SUBSET t' s ∧ IN x (hull convex (INSERT a t)) ∧ IN x (hull convex (INSERT a t')) ⟶ IN x (hull convex (INSERT a (HOL_Light_Import.INTER t t')))

thm CONVEX_HULL_EXCHANGE_UNION:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. IN a (hull convex s) ⟶ hull convex s = UNIONS (GSPEC (λGEN%PVAR%953::(real, ?'a::type) cart ⇒ bool. ∃ b::(real, ?'a::type) cart. SETSPEC GEN%PVAR%953 (IN b s) (hull convex (INSERT a (DELETE s b)))))

thm CONVEX_HULL_EXCHANGE_INTER:

∀ (s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) (t::(real, ?'a::type) cart ⇒ bool) t'::(real, ?'a::type) cart ⇒ bool. ¬ affine_dependent s ∧ IN a (hull convex s) ∧ SUBSET t s ∧ SUBSET t' s ⟶ HOL_Light_Import.INTER (hull convex (INSERT a t)) (hull convex (INSERT a t')) = hull convex (INSERT a (HOL_Light_Import.INTER t t'))

thm AFF_DIM_EQ_HYPERPLANE:

∀ s::(real, ?'a::type) cart ⇒ bool. (aff_dim s = int (dimindex HOL_Light_Import.UNIV) − int (1::nat)) = (∃ (a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ hull affine s = GSPEC (λGEN%PVAR%954::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%954 (dot a x = b) x))

thm AFF_DIM_HYPERPLANE:

∀ (a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ⟶ aff_dim (GSPEC (λGEN%PVAR%955::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%955 (dot a x = b) x)) = int (dimindex HOL_Light_Import.UNIV) − int (1::nat)

thm BOUNDED_HYPERPLANE_EQ_TRIVIAL:

∀ (a::(real, ?'a::type) cart) b::real. bounded (GSPEC (λGEN%PVAR%957::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%957 (dot a x = b) x)) = (if a = vec (0::nat) then b ≠ (0::real) else dimindex HOL_Light_Import.UNIV = (1::nat))

thm AFFINE_HULL_FINITE_INTERSECTION_HYPERPLANES:

∀ s::(real, ?'a::type) cart ⇒ bool. ∃ f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ int (CARD f) + aff_dim s = int (dimindex HOL_Light_Import.UNIV) ∧ hull affine s = INTERS f ∧ (∀ h::(real, ?'a::type) cart ⇒ bool. IN h f ⟶ (∃ (a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ h = GSPEC (λGEN%PVAR%959::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%959 (dot a x = b) x)))

thm AFFINE_HYPERPLANE_SUMS_EQ_UNIV:

∀ (a::(real, ?'a::type) cart) (b::real) s::(real, ?'a::type) cart ⇒ bool. affine s ∧ HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%966::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%966 (dot a v = b) v)) ≠ EMPTY ∧ DIFF s (GSPEC (λGEN%PVAR%967::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%967 (dot a v = b) v)) ≠ EMPTY ⟶ GSPEC (λGEN%PVAR%969::(real, ?'a::type) cart. ∃ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%969 (IN x s ∧ IN y (GSPEC (λGEN%PVAR%968::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%968 (dot a v = b) v))) (vector_add x y)) = HOL_Light_Import.UNIV

thm AFF_DIM_AFFINE_INTER_HYPERPLANE:

∀ (a::(real, ?'a::type) cart) (b::real) s::(real, ?'a::type) cart ⇒ bool. affine s ⟶ aff_dim (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%973::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%973 (dot a x = b) x))) = (if HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%974::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%974 (dot a v = b) v)) = EMPTY then − int (1::nat) else if SUBSET s (GSPEC (λGEN%PVAR%975::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%975 (dot a v = b) v)) then aff_dim s else aff_dim s − int (1::nat))

thm AFF_DIM_LT_FULL:

∀ s::(real, ?'a::type) cart ⇒ bool. (aff_dim s < int (dimindex HOL_Light_Import.UNIV)) = (hull affine s ≠ HOL_Light_Import.UNIV)

thm AFF_LOWDIM_SUBSET_HYPERPLANE:

∀ s::(real, ?'a::type) cart ⇒ bool. aff_dim s < int (dimindex HOL_Light_Import.UNIV) ⟶ (∃ (a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ SUBSET s (GSPEC (λGEN%PVAR%976::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%976 (dot a x = b) x)))

thm HYPERPLANE_EQ_EMPTY:

∀ (a::(real, ?'a::type) cart) b::real. (GSPEC (λGEN%PVAR%977::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%977 (dot a x = b) x) = EMPTY) = (a = vec (0::nat) ∧ b ≠ (0::real))

thm HYPERPLANE_EQ_UNIV:

$\forall (a::(real, ?'a::type)\ cart)\ b::real.\ (GSPEC\ (\lambda GEN\%PVAR\%978::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%978\ (dot\ a\ x = b)\ x) = HOL\_Light\_Import.UNIV) = (a = vec\ (0::nat) \wedge b = (0::real))$

thm SUBSET_HYPERPLANES:

$\forall (a::(real, ?'a::type)\ cart)\ (b::real)\ (a'::(real, ?'a::type)\ cart)\ b'::real.\ SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%987::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%987\ (dot\ a\ x = b)\ x))\ (GSPEC\ (\lambda GEN\%PVAR\%988::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%988\ (dot\ a'\ x = b')\ x)) = (GSPEC\ (\lambda GEN\%PVAR\%989::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%989\ (dot\ a\ x = b)\ x) = EMPTY \vee GSPEC\ (\lambda GEN\%PVAR\%990::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%990\ (dot\ a'\ x = b')\ x) = HOL\_Light\_Import.UNIV \vee GSPEC\ (\lambda GEN\%PVAR\%991::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%991\ (dot\ a\ x = b)\ x) = GSPEC\ (\lambda GEN\%PVAR\%992::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%992\ (dot\ a'\ x = b')\ x))$

thm OPEN_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (hull\ convex\ s)$

thm COMPACT_CONVEX_COMBINATIONS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge compact\ t \longrightarrow compact\ (GSPEC\ (\lambda GEN\%PVAR\%996::(real, ?'a::type)\ cart.\ \exists (x::(real, ?'a::type)\ cart)\ (u::real)\ y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%996\ ((0::real) \leq u \wedge u \leq (1::real) \wedge IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ (\%\ ((1::real) - u)\ x)\ (\%\ u\ y))))$

thm COMPACT_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow compact\ (hull\ convex\ s)$

thm FINITE_IMP_COMPACT_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow compact\ (hull\ convex\ s)$

thm DIST_INCREASES_ONLINE:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ d::(real, ?'a::type)\ cart.\ d \neq vec\ (0::nat) \longrightarrow distance\ (a, b) < distance\ (a, vector\_add\ b\ d) \vee distance\ (a, b) < distance\ (a, vector\_sub\ b\ d)$

thm NORM_INCREASES_ONLINE:

$\forall (a::(real, ?'a::type)\ cart)\ d::(real, ?'a::type)\ cart.\ d \neq vec\ (0::nat) \longrightarrow vector\_norm\ a < vector\_norm\ (vector\_add\ a\ d) \vee vector\_norm\ a < vector\_norm\ (vector\_sub\ a\ d)$

thm SIMPLEX_FURTHEST_LT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*hull convex s*) $\wedge$ $\neg$ *IN x s* $\longrightarrow$ ($\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *IN y* (*hull convex s*) $\wedge$ *vector_norm* (*vector_sub x a*) < *vector_norm* (*vector_sub y a*)))

thm SIMPLEX_FURTHEST_LE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*hull convex s*) $\longrightarrow$ *vector_norm* (*vector_sub x a*) $\leq$ *vector_norm* (*vector_sub y a*)))

thm SIMPLEX_FURTHEST_LE_EXISTS:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*hull convex s*) $\longrightarrow$ ($\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* $\wedge$ *vector_norm* (*vector_sub x a*) $\leq$ *vector_norm* (*vector_sub y a*)))

thm SIMPLEX_EXTREMAL_LE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u s* $\wedge$ *IN v s* $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x* (*hull convex s*) $\wedge$ *IN y* (*hull convex s*) $\longrightarrow$ *vector_norm* (*vector_sub x y*) $\leq$ *vector_norm* (*vector_sub u v*)))

thm SIMPLEX_EXTREMAL_LE_EXISTS:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *IN x* (*hull convex s*) $\wedge$ *IN y* (*hull convex s*) $\longrightarrow$ ($\exists$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u s* $\wedge$ *IN v s* $\wedge$ *vector_norm* (*vector_sub x y*) $\leq$ *vector_norm* (*vector_sub u v*))

thm DIAMETER_CONVEX_HULL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *diameter* (*hull convex s*) = *diameter s*

thm DIAMETER_SIMPLEX:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ *diameter* (*hull convex s*) = *HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%1005*::*real*. $\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1005* (*IN x s* $\wedge$ *IN y s*) (*distance* (*x*, *y*))))

thm CLOSER_POINTS_LEMMA:

$\forall$ (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *dot y z* $\longrightarrow$ ($\exists$ *u*>*0*::*real*. $\forall$ *v*::*real*. (*0*::*real*) < *v* $\wedge$ *v* $\leq$ *u* $\longrightarrow$ *vector_norm* (*vector_sub* (% *v z*) *y*) < *vector_norm y*)

thm CLOSER_POINT_LEMMA:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *dot* (*vector_sub y x*) (*vector_sub z x*) $\longrightarrow$ ($\exists$ *u*>*0*::*real*. *u* $\leq$ (*1*::*real*) $\wedge$ *distance* (*vector_add x* (% *u* (*vector_sub z x*)), *y*) < *distance* (*x*, *y*))

thm ANY_CLOSEST_POINT_DOT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge (\forall z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ z)) \longrightarrow dot\ (vector\_sub\ a\ x)\ (vector\_sub\ y\ x) \leq (0::real)$

thm ANY_CLOSEST_POINT_UNIQUE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge (\forall z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ z)) \wedge (\forall z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \longrightarrow distance\ (a,\ y) \leq distance\ (a,\ z)) \longrightarrow x = y$

thm CLOSEST_POINT_UNIQUE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge IN\ x\ s \wedge (\forall z::(real,\ ?'a::type)\ cart.\ IN\ z\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ z)) \longrightarrow x = closest\_point\ s\ a$

thm CLOSEST_POINT_DOT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge IN\ x\ s \longrightarrow dot\ (vector\_sub\ a\ (closest\_point\ s\ a))\ (vector\_sub\ x\ (closest\_point\ s\ a)) \leq (0::real)$

thm CLOSEST_POINT_LT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge IN\ x\ s \wedge x \neq closest\_point\ s\ a \longrightarrow distance\ (a,\ closest\_point\ s\ a) < distance\ (a,\ x)$

thm CLOSEST_POINT_LIPSCHITZ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow distance\ (closest\_point\ s\ x,\ closest\_point\ s\ y) \leq distance\ (x,\ y)$

thm CONTINUOUS_AT_CLOSEST_POINT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow continuous\ (closest\_point\ s)\ (at\ x)$

thm CONTINUOUS_ON_CLOSEST_POINT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow continuous\_on\ (closest\_point\ s)\ t$

thm ANY_CLOSEST_POINT_AFFINE_ORTHOGONAL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ affine\ s \wedge IN\ b\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow distance\ (a,$

b) ≤ *distance* (*a*, *x*)) ⟶ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *orthogonal* (*vector_sub x b*) (*vector_sub a b*))

thm ORTHOGONAL_ANY_CLOSEST_POINT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN b s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *orthogonal* (*vector_sub x b*) (*vector_sub a b*)) ⟶ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *distance* (*a*, *b*) ≤ *distance* (*a*, *x*))

thm CLOSEST_POINT_AFFINE_ORTHOGONAL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *s* ≠ *EMPTY* ∧ *IN x s* ⟶ *orthogonal* (*vector_sub x* (*closest_point s a*)) (*vector_sub a* (*closest_point s a*))

thm CLOSEST_POINT_AFFINE_ORTHOGONAL_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *affine s* ∧ *IN b s* ⟶ (*closest_point s a* = *b*) = (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *orthogonal* (*vector_sub x b*) (*vector_sub a b*))

thm SUPPORTING_HYPERPLANE_COMPACT_POINT_SUP:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ *s* ≠ *EMPTY* ⟶ (∃ (*b*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* ∧ *dot a* (*vector_sub y c*) = *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a* (*vector_sub x c*) ≤ *b*))

thm SUPPORTING_HYPERPLANE_COMPACT_POINT_INF:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ *s* ≠ *EMPTY* ⟶ (∃ (*b*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* ∧ *dot a* (*vector_sub y c*) = *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a* (*vector_sub x c*)))

thm SUPPORTING_HYPERPLANE_CLOSED_POINT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *z*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *HOL_Light_Import.closed s* ∧ *s* ≠ *EMPTY* ∧ ¬ *IN z s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *dot a z* < *b* ∧ *IN y s* ∧ *dot a y* = *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a x*))

thm SEPARATING_HYPERPLANE_CLOSED_POINT_INSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *z*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *HOL_Light_Import.closed s* ∧ *s* ≠ *EMPTY* ∧ ¬ *IN z s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *IN a s* ∧ *dot* (*vector_sub a z*) *z* < *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* < *dot* (*vector_sub a z*) *x*))

thm SEPARATING_HYPERPLANE_CLOSED_0_INSET:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *HOL_Light_Import.closed s* ∧ *s* ≠ *EMPTY* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*.

IN a s ∧ *a* ≠ *vec* (*0*::*nat*) ∧ (*0*::*real*) < *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* < *dot a x*))

thm SEPARATING_HYPERPLANE_CLOSED_POINT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *z*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *HOL_Light_Import*.*closed s* ∧ ¬ *IN z s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *dot a z* < *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* < *dot a x*))

thm SEPARATING_HYPERPLANE_CLOSED_0:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *HOL_Light_Import*.*closed s* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *a* ≠ *vec* (*0*::*nat*) ∧ (*0*::*real*) < *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* < *dot a x*))

thm SEPARATING_HYPERPLANE_CLOSED_COMPACT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *HOL_Light_Import*.*closed s* ∧ *convex t* ∧ *compact t* ∧ *t* ≠ *EMPTY* ∧ *DISJOINT s t* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a x* < *b*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x t* ⟶ *b* < *dot a x*))

thm SEPARATING_HYPERPLANE_COMPACT_CLOSED:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *compact s* ∧ *s* ≠ *EMPTY* ∧ *convex t* ∧ *HOL_Light_Import*.*closed t* ∧ *DISJOINT s t* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a x* < *b*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x t* ⟶ *b* < *dot a x*))

thm SEPARATING_HYPERPLANE_SET_0_INSPAN:

∀ *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *s* ≠ *EMPTY* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ (∃ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::*?'a*::*type*. *IN a* (*span s*) ∧ *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *dot a x*))

thm SEPARATING_HYPERPLANE_SET_POINT_INAFF:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *z*::(*real*, *?'a*::*type*) *cart*. *convex s* ∧ *s* ≠ *EMPTY* ∧ ¬ *IN z s* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *IN* (*vector_add z a*) (*hull affine* (*INSERT z s*)) ∧ *a* ≠ *vec* (*0*::*nat*) ∧ *dot a z* ≤ *b* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a x*))

thm SEPARATING_HYPERPLANE_SET_0:

∀ *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *convex s* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ (∃ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::*?'a*::*type*. *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *dot a x*))

thm SEPARATING_HYPERPLANE_SETS:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *convex t* ∧ *s* ≠ *EMPTY* ∧ *t* ≠ *EMPTY* ∧ *DISJOINT s t* ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *a* ≠ *vec* (*0*::*nat*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a x* ≤ *b*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x t* ⟶ *b* ≤ *dot a x*))

thm CONVEX_CLOSURE:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow convex\ (closure\ s)$

thm CONVEX_INTERIOR:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow convex\ (interior\ s)$

thm CONVEX_HULL_TRANSLATION:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ hull\ convex\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (hull\ convex\ s)$

thm CONVEX_HULL_SCALING:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ hull\ convex\ (IMAGE\ (\%\ c)\ s) = IMAGE\ (\%\ c)\ (hull\ convex\ s)$

thm CONVEX_HULL_AFFINITY:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ c::real.\ hull\ convex\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ a\ (\%\ c\ x))\ s) = IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ a\ (\%\ c\ x))\ (hull\ convex\ s)$

thm CONVEX_HALFSPACE_INTERSECTION:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge convex\ s \longrightarrow s = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%1010::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1010\ (SUBSET\ s\ h \wedge (\exists\, (a::(real,\ ?'a::type)\ cart)\ b::real.\ h = GSPEC\ (\lambda GEN\%PVAR\%1009::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1009\ (dot\ a\ x \le b)\ x)))\ h))$

thm RADON_EX_LEMMA:

$\forall\, c::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ c \wedge affine\_dependent\ c \longrightarrow (\exists\, u::(real,\ ?'a::type)\ cart \Rightarrow real.\ sum\ c\ u = (0::real) \wedge (\exists\, v::(real,\ ?'a::type)\ cart.\ IN\ v\ c \wedge u\ v \neq (0::real)) \wedge vsum\ c\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (u\ v)\ v) = vec\ (0::nat))$

thm RADON_S_LEMMA:

$\forall\, (s::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow real.\ FINITE\ s \wedge sum\ s\ f = (0::real) \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%1018::?'a::type.\ \exists\, x::?'a::type.\ SETSPEC\ GEN\%PVAR\%1018\ (IN\ x\ s \wedge (0::real) < f\ x)\ x))\ f = -\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%1019::?'a::type.\ \exists\, x::?'a::type.\ SETSPEC\ GEN\%PVAR\%1019\ (IN\ x\ s \wedge f\ x < (0::real))\ x))\ f$

thm RADON_V_LEMMA:

$\forall\, (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ g::?'b::type \Rightarrow real.\ FINITE\ s \wedge vsum\ s\ f = vec\ (0::nat) \wedge (\forall\, x::?'b::type.\ g\ x = (0::real) \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow vsum\ (GSPEC\ (\lambda GEN\%PVAR\%1026::?'b::type.\ \exists\, x::?'b::type.\ SETSPEC\ GEN\%PVAR\%1026\ (IN\ x\ s \wedge (0::real) < g\ x)\ x))\ f$

= *vector_neg (vsum (GSPEC (λGEN%PVAR%1027::?'b::type. ∃x::?'b::type. SETSPEC GEN%PVAR%1027 (IN x s ∧ g x < (0::real)) x)) f)*

thm RADON_PARTITION:

∀c::(real, ?'a::type) cart ⇒ bool. FINITE c ∧ affine_dependent c ⟶ (∃(m::(real, ?'a::type) cart ⇒ bool) p::(real, ?'a::type) cart ⇒ bool. DISJOINT m p ∧ HOL_Light_Import.UNION m p = c ∧ ¬ DISJOINT (hull convex m) (hull convex p))

thm RADON:

∀c::(real, ?'a::type) cart ⇒ bool. affine_dependent c ⟶ (∃(m::(real, ?'a::type) cart ⇒ bool) p::(real, ?'a::type) cart ⇒ bool. SUBSET m c ∧ SUBSET p c ∧ DISJOINT m p ∧ ¬ DISJOINT (hull convex m) (hull convex p))

thm HELLY_INDUCT:

∀(n::nat) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. HAS_SIZE f n ∧ dimindex HOL_Light_Import.UNIV + (1::nat) ≤ n ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex s) ∧ (∀t::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET t f ∧ CARD t = dimindex HOL_Light_Import.UNIV + (1::nat) ⟶ INTERS t ≠ EMPTY) ⟶ INTERS f ≠ EMPTY

thm HELLY:

∀f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ dimindex HOL_Light_Import.UNIV + (1::nat) ≤ CARD f ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex s) ∧ (∀t::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET t f ∧ CARD t = dimindex HOL_Light_Import.UNIV + (1::nat) ⟶ INTERS t ≠ EMPTY) ⟶ INTERS f ≠ EMPTY

thm HELLY_ALT:

∀f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex s) ∧ (∀t::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET t f ∧ CARD t ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ⟶ INTERS t ≠ EMPTY) ⟶ INTERS f ≠ EMPTY

thm HELLY_CLOSED_ALT:

∀f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex s ∧ HOL_Light_Import.closed s) ∧ (∃s::(real, ?'a::type) cart ⇒ bool. IN s f ∧ bounded s) ∧ (∀t::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET t f ∧ FINITE t ∧ CARD t ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ⟶ INTERS t ≠ EMPTY) ⟶ INTERS f ≠ EMPTY

thm HELLY_COMPACT_ALT:

∀f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex s ∧ compact s) ∧ (∀t::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET t f ∧ FINITE t ∧ CARD t ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ⟶ INTERS t ≠ EMPTY) ⟶ INTERS f ≠ EMPTY

thm HELLY_CLOSED:

$\forall f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (FINITE\, f \longrightarrow dimindex\ HOL\_Light\_Import.UNIV + (1::nat) \leq CARD\, f) \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow convex\ s \wedge HOL\_Light\_Import.closed\ s) \wedge (\exists s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \wedge bounded\ s) \wedge (\forall t::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SUBSET\ t\ f \wedge FINITE\ t \wedge CARD\ t = dimindex\ HOL\_Light\_Import.UNIV + (1::nat) \longrightarrow INTERS\ t \neq EMPTY) \longrightarrow INTERS\ f \neq EMPTY$

thm HELLY_COMPACT:

$\forall f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (FINITE\, f \longrightarrow dimindex\ HOL\_Light\_Import.UNIV + (1::nat) \leq CARD\, f) \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow convex\ s \wedge compact\ s) \wedge (\forall t::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SUBSET\ t\ f \wedge FINITE\ t \wedge CARD\ t = dimindex\ HOL\_Light\_Import.UNIV + (1::nat) \longrightarrow INTERS\ t \neq EMPTY) \longrightarrow INTERS\ f \neq EMPTY$

thm CONVEX_HULL_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \longrightarrow hull\ convex\ (IMAGE\ f\ s) = IMAGE\ f\ (hull\ convex\ s)$

thm IN_CONVEX_HULL_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ linear\ f \wedge IN\ x\ (hull\ convex\ s) \longrightarrow IN\ (f\ x)\ (hull\ convex\ (IMAGE\ f\ s))$

thm IS_INTERVAL_CONVEX:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ is\_interval\ s \longrightarrow convex\ s$

thm IS_INTERVAL_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ is\_interval\ s \longrightarrow connected\ s$

thm IS_INTERVAL_CONNECTED_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ is\_interval\ s = connected\ s$

thm CONVEX_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ convex\ (closed\_interval\ [(a,\ b)]) \wedge convex\ (open\_interval\ (a,\ b))$

thm IS_INTERVAL_CONVEX_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ is\_interval\ s = convex\ s$

thm CONVEX_CONNECTED_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ convex\ s = connected\ s$

thm CONNECTED_CONVEX_1:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ connected\ s = convex\ s$

thm CONNECTED_COMPACT_INTERVAL_1:

$\forall\, s{::}(real,\ unit)\ cart \Rightarrow bool.\ (connected\ s \wedge compact\ s) = (\exists\, (a{::}(real,\ unit)\ cart)\ b{::}(real,\ unit)\ cart.\ s = closed\_interval\ [(a,\ b)])$

thm CONVEX_CONNECTED_1_GEN:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV = (1{::}nat) \longrightarrow convex\ s = connected\ s$

thm EQ_BALLS_conjunct3:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ (a'{::}(real,\ ?'a{::}type)\ cart)\ (r{::}real)\ r'{::}real.\ (cball\ (a,\ r) = cball\ (a',\ r')) = (a = a' \wedge r = r' \vee r < (0{::}real) \wedge r' < (0{::}real))$

thm EQ_BALLS_conjunct2:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ (a'{::}(real,\ ?'a{::}type)\ cart)\ (r{::}real)\ r'{::}real.\ (cball\ (a,\ r) = ball\ (a',\ r')) = (r < (0{::}real) \wedge r' \leq (0{::}real))$

thm EQ_BALLS_conjunct1:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ (a'{::}(real,\ ?'a{::}type)\ cart)\ (r{::}real)\ r'{::}real.\ (ball\ (a,\ r) = cball\ (a',\ r')) = (r \leq (0{::}real) \wedge r' < (0{::}real))$

thm EQ_BALLS_conjunct0:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ (a'{::}(real,\ ?'a{::}type)\ cart)\ (r{::}real)\ r'{::}real.\ (ball\ (a,\ r) = ball\ (a',\ r')) = (a = a' \wedge r = r' \vee r \leq (0{::}real) \wedge r' \leq (0{::}real))$

thm JUNG:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ r{::}real.\ bounded\ s \wedge sqrt\ (real\_of\_nat\ (dimindex\ HOL\_Light\_Import.UNIV) / real\_of\_nat\ ((2{::}nat) * dimindex\ HOL\_Light\_Import.UNIV + (2{::}nat))) * diameter\ s \leq r \longrightarrow (\exists\, a{::}(real,\ ?'a{::}type)\ cart.\ SUBSET\ s\ (cball\ (a,\ r)))$

thm COMPACT_FRONTIER_LINE_LEMMA:

$\forall\, (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ x{::}(real,\ ?'a{::}type)\ cart.\ compact\ s \wedge IN\ (vec\ (0{::}nat))\ s \wedge x \neq vec\ (0{::}nat) \longrightarrow (\exists\, u{\geq}0{::}real.\ IN\ (\%\ u\ x)\ (frontier\ s) \wedge (\forall\, v{>}u.\ \neg\ IN\ (\%\ v\ x)\ s))$

thm STARLIKE_COMPACT_PROJECTIVE:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ compact\ s \wedge SUBSET\ (cball\ (vec\ (0{::}nat),\ 1{::}real))\ s \wedge (\forall\, (x{::}(real,\ ?'a{::}type)\ cart)\ u{::}real.\ IN\ x\ s \wedge (0{::}real) \leq u \wedge u < (1{::}real) \longrightarrow IN\ (\%\ u\ x)\ (DIFF\ s\ (frontier\ s))) \longrightarrow homeomorphic\ s\ (cball\ (vec\ (0{::}nat),\ 1{::}real))$

thm HOMEOMORPHIC_CONVEX_COMPACT_LEMMA:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \wedge (0{::}real) < (?e{::}real) \wedge SUBSET\ (cball\ (vec\ (0{::}nat),\ 1{::}real))\ s \longrightarrow homeomorphic\ s\ (cball\ (vec\ (0{::}nat),\ 1{::}real))$

thm HOMEOMORPHIC_CONVEX_COMPACT_CBALL:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*b*::(*real*, *?'a*::*type*) *cart*) *e*::*real*. *convex s* $\wedge$ *compact s* $\wedge$ *interior s* $\neq$ *EMPTY* $\wedge$ (*0*::*real*) $<$ *e* $\longrightarrow$ *homeomorphic s* (*cball* (*b*, *e*))

thm HOMEOMORPHIC_CONVEX_COMPACT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex s* $\wedge$ *compact s* $\wedge$ *interior s* $\neq$ *EMPTY* $\wedge$ *convex t* $\wedge$ *compact t* $\wedge$ *interior t* $\neq$ *EMPTY* $\longrightarrow$ *homeomorphic s t*

thm HOMEOMORPHIC_CLOSED_INTERVALS:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\wedge$ *open_interval* (*c*, *d*) $\neq$ *EMPTY* $\longrightarrow$ *homeomorphic* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)])

thm DEF_convex_cone:

convex_cone = (λ*_320326*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *_320326* $\neq$ *EMPTY* $\wedge$ *convex _320326* $\wedge$ *conic _320326*)

thm convex_cone:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex_cone s* = (*s* $\neq$ *EMPTY* $\wedge$ *convex s* $\wedge$ *conic s*)

thm CONVEX_CONE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex_cone s* = (*IN* (*vec* (*0*::*nat*)) *s* $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\longrightarrow$ *IN* (*vector_add x y*) *s*) $\wedge$ ($\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *c*::*real*. *IN x s* $\wedge$ (*0*::*real*) $\leq$ *c* $\longrightarrow$ *IN* (% *c x*) *s*))

thm CONVEX_CONE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *convex_cone s* $\wedge$ *linear f* $\longrightarrow$ *convex_cone* (*IMAGE f s*)

thm CONVEX_CONE_LINEAR_IMAGE_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *convex_cone* (*IMAGE f s*) = *convex_cone s*

thm CONVEX_CONE_HALFSPACE_GE:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. *convex_cone* (*GSPEC* (λ*GEN%PVAR%1047*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1047* ((*0*::*real*) $\leq$ *dot a x*) *x*))

thm CONVEX_CONE_HALFSPACE_LE:

$\forall$ *a::(real, ?'a::type) cart. convex_cone (GSPEC (λGEN%PVAR%1048::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1048 (dot a x ≤ (0::real)) x))*

thm CONVEX_CONE_CONTAINS_0:

∀s::(real, ?'a::type) cart ⇒ bool. convex_cone s ⟶ IN (vec (0::nat)) s

thm CONVEX_CONE_INTERS:

∀f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ convex_cone s) ⟶ convex_cone (INTERS f)

thm CONVEX_CONE_CONVEX_CONE_HULL:

∀s::(real, ?'a::type) cart ⇒ bool. convex_cone (hull convex_cone s)

thm CONVEX_CONVEX_CONE_HULL:

∀s::(real, ?'a::type) cart ⇒ bool. convex (hull convex_cone s)

thm CONIC_CONVEX_CONE_HULL:

∀s::(real, ?'a::type) cart ⇒ bool. conic (hull convex_cone s)

thm CONVEX_CONE_HULL_NONEMPTY:

∀s::(real, ?'a::type) cart ⇒ bool. hull convex_cone s ≠ EMPTY

thm CONVEX_CONE_HULL_CONTAINS_0:

∀s::(real, ?'a::type) cart ⇒ bool. IN (vec (0::nat)) (hull convex_cone s)

thm CONVEX_CONE_HULL_ADD:

∀(s::(real, ?'a::type) cart ⇒ bool) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x (hull convex_cone s) ∧ IN y (hull convex_cone s) ⟶ IN (vector_add x y) (hull convex_cone s)

thm CONVEX_CONE_HULL_MUL:

∀(s::(real, ?'a::type) cart ⇒ bool) (c::real) x::(real, ?'a::type) cart. (0::real) ≤ c ∧ IN x (hull convex_cone s) ⟶ IN (% c x) (hull convex_cone s)

thm CONVEX_CONE_SUMS:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. convex_cone s ∧ convex_cone t ⟶ convex_cone (GSPEC (λGEN%PVAR%1049::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1049 (IN x s ∧ IN y t) (vector_add x y)))

thm CONVEX_CONE_HULL_UNION:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. hull convex_cone (HOL_Light_Import.UNION s t) = GSPEC (λGEN%PVAR%1050::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1050 (IN x (hull convex_cone s) ∧ IN y (hull convex_cone t)) (vector_add x y))

thm CONVEX_CONE_SING:

convex_cone (INSERT (vec (0::nat)) EMPTY)

thm CONVEX_HULL_SUBSET_CONVEX_CONE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*hull convex s*) (*hull convex_cone s*)

thm CONIC_HULL_SUBSET_CONVEX_CONE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*hull conic s*) (*hull convex_cone s*)

thm CONVEX_CONE_HULL_SEPARATE_NONEMPTY:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *s* ≠ *EMPTY* ⟶ *hull convex_cone s* = *hull conic* (*hull convex s*)

thm CONVEX_CONE_HULL_EMPTY:

hull convex_cone EMPTY = *INSERT* (*vec* (*0*::*nat*)) *EMPTY*

thm CONVEX_CONE_HULL_SEPARATE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *hull convex_cone s* = *INSERT* (*vec* (*0*::*nat*)) (*hull conic* (*hull convex s*))

thm CONVEX_CONE_HULL_CONVEX_HULL_NONEMPTY:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *s* ≠ *EMPTY* ⟶ *hull convex_cone s* = *GSPEC* (λ*GEN%PVAR%1051*::(*real*, *?'a*::*type*) *cart*. ∃ (*c*::*real*) *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1051* ((*0*::*real*) ≤ *c* ∧ *IN x* (*hull convex s*)) (% *c x*))

thm CONVEX_CONE_HULL_CONVEX_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *hull convex_cone s* = *INSERT* (*vec* (*0*::*nat*)) (*GSPEC* (λ*GEN%PVAR%1052*::(*real*, *?'a*::*type*) *cart*. ∃ (*c*::*real*) *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1052* ((*0*::*real*) ≤ *c* ∧ *IN x* (*hull convex s*)) (% *c x*)))

thm CONVEX_CONE_HULL_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ⟶ *hull convex_cone* (*IMAGE f s*) = *IMAGE f* (*hull convex_cone s*)

thm SUBSPACE_IMP_CONVEX_CONE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *subspace s* ⟶ *convex_cone s*

thm CONVEX_CONE_SPAN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex_cone* (*span s*)

thm CONVEX_CONE_NEGATIONS:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex_cone* *s* ⟶ *convex_cone* (*IMAGE* *vector_neg* *s*)

thm SUBSPACE_CONVEX_CONE_SYMMETRIC:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *subspace* *s* = (*convex_cone* *s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN* *x* *s* ⟶ *IN* (*vector_neg* *x*) *s*))

thm SPAN_CONVEX_CONE_ALLSIGNS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *span* *s* = *hull* *convex_cone* (*HOL_Light_Import.UNION* *s* (*IMAGE* *vector_neg* *s*))

thm DEF_epigraph:

epigraph = (λ(*_323328*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *_323329*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *GSPEC* (λ*GEN%PVAR%1054*::(*real*, (*?'a*::*type*, *unit*) *finite_sum*) *cart*. ∃ *xy*::(*real*, (*?'a*::*type*, *unit*) *finite_sum*) *cart*. *SETSPEC* *GEN%PVAR%1054* (*IN* (*fstcart* *xy*) *_323328* ∧ *_323329* (*fstcart* *xy*) ≤ *HOL_Light_Import.drop* (*sndcart* *xy*)) *xy*))

thm epigraph:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *epigraph* *s* *f* = *GSPEC* (λ*GEN%PVAR%1054*::(*real*, (*?'a*::*type*, *unit*) *finite_sum*) *cart*. ∃ *xy*::(*real*, (*?'a*::*type*, *unit*) *finite_sum*) *cart*. *SETSPEC* *GEN%PVAR%1054* (*IN* (*fstcart* *xy*) *s* ∧ *f* (*fstcart* *xy*) ≤ *HOL_Light_Import.drop* (*sndcart* *xy*)) *xy*)

thm IN_EPIGRAPH:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::*real*. *IN* (*pastecart* *x* (*lift* *y*)) (*epigraph* (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*?f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*)) = (*IN* *x* *?s* ∧ *?f* *x* ≤ *y*)

thm CONVEX_EPIGRAPH:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*convex_on* *f* *s* ∧ *convex* *s*) = *convex* (*epigraph* *s* *f*)

thm CONVEX_EPIGRAPH_CONVEX:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex* *s* ⟶ *convex_on* *f* *s* = *convex* (*epigraph* *s* *f*)

thm FORALL_OF_PASTECART:

(∀ *p*::*?'d*::*type* ⇒ (*?'c*::*type*, (*?'b*::*type*, *?'a*::*type*) *finite_sum*) *cart*. (*?P*::(*?'d*::*type* ⇒ (*?'c*::*type*, *?'b*::*type*) *cart*) ⇒ (*?'d*::*type* ⇒ (*?'c*::*type*, *?'a*::*type*) *cart*) ⇒ *bool*) (*fstcart* ∘ *p*) (*sndcart* ∘ *p*)) = (∀ (*x*::*?'d*::*type* ⇒ (*?'c*::*type*, *?'b*::*type*) *cart*) *y*::*?'d*::*type* ⇒ (*?'c*::*type*, *?'a*::*type*) *cart*. *?P* *x* *y*)

thm FORALL_OF_DROP:

(∀ *v*::*?'a*::*type* ⇒ (*real*, *unit*) *cart*. (*?P*::(*?'a*::*type* ⇒ *real*) ⇒ *bool*) (*HOL_Light_Import.drop* ∘ *v*)) = (∀ *x*::*?'a*::*type* ⇒ *real*. *?P* *x*)

thm CONVEX_ON_JENSEN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow convex\_on\ f\ s = (\forall (k::nat)\ (u::nat \Rightarrow real)\ x::nat \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq k \longrightarrow (0::real) \leq u\ i \wedge IN\ (x\ i)\ s) \wedge sum\ (dotdot\ (1::nat)\ k)\ u = (1::real) \longrightarrow f\ (vsum\ (dotdot\ (1::nat)\ k)\ (\lambda i::nat.\ \%\ (u\ i)\ (x\ i))) \leq sum\ (dotdot\ (1::nat)\ k)\ (\lambda i::nat.\ u\ i * f\ (x\ i)))$

thm IVT_INCREASING_COMPONENT_ON_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ (y::real)\ k::nat.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ a)\ k \leq y \wedge y \leq \$\ (f\ b)\ k \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ x)\ k = y)$

thm IVT_INCREASING_COMPONENT_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ (y::real)\ k::nat.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow continuous\ f\ (at\ x)) \wedge \$\ (f\ a)\ k \leq y \wedge y \leq \$\ (f\ b)\ k \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ x)\ k = y)$

thm IVT_DECREASING_COMPONENT_ON_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ (y::real)\ k::nat.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ b)\ k \leq y \wedge y \leq \$\ (f\ a)\ k \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ x)\ k = y)$

thm IVT_DECREASING_COMPONENT_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ (y::real)\ k::nat.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow continuous\ f\ (at\ x)) \wedge \$\ (f\ b)\ k \leq y \wedge y \leq \$\ (f\ a)\ k \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (f\ x)\ k = y)$

thm CONVEX_ON_CONVEX_HULL_BOUND:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ b::real.\ convex\_on\ f\ (hull\ convex\ s) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\ x \leq b) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (hull\ convex\ s) \longrightarrow f\ x \leq b)$

thm UNIT_INTERVAL_CONVEX_HULL:

$closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))] = hull\ convex\ (GSPEC\ (\lambda GEN\%PVAR\%1063::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1063\ (\forall i::nat.$

$(1::nat) \leq i \wedge i \leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ \$ $x\ i = (0::real) \vee$ \$ $x\ i = (1::real))\ x))$

thm CLOSED_INTERVAL_AS_CONVEX_HULL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ \exists s::(real, ?'a::type)\ cart \Rightarrow bool.$ *FINITE s* $\wedge$ *closed_interval* $[(a,\ b)] =$ *hull convex s*

thm CONVEX_ON_BOUNDED_CONTINUOUS:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ b::real.$ *HOL_Light_Import.open s* $\wedge$ *convex_on f s* $\wedge$ $(\forall x::(real, ?'a::type)\ cart.$ *IN x s* $\longrightarrow |f\ x| \leq b) \longrightarrow$ *continuous_on* $(lift \circ f)\ s$

thm CONVEX_BOUNDS_LEMMA:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ (x::(real, ?'a::type)\ cart)\ e::real.$ *convex_on f* $(cball\ (x,\ e)) \wedge (\forall y::(real, ?'a::type)\ cart.$ *IN y* $(cball\ (x,\ e)) \longrightarrow f\ y \leq (?b::real)) \longrightarrow (\forall y::(real, ?'a::type)\ cart.$ *IN y* $(cball\ (x,\ e)) \longrightarrow |f\ y| \leq ?b +$ *real_of_nat* $(2::nat) * |f\ x|)$

thm CONVEX_ON_CONTINUOUS:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ s::(real, ?'a::type)\ cart \Rightarrow bool.$ *HOL_Light_Import.open s* $\wedge$ *convex_on f s* $\longrightarrow$ *continuous_on* $(lift \circ f)\ s$

thm CONVEX_ON_RIGHT_SECANT_MUL:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ s::(real, ?'a::type)\ cart \Rightarrow bool.$ *convex_on f s* $= (\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.$ *IN a s* $\wedge$ *IN b s* $\wedge$ *IN x* $(closed\_segment\ [(a,\ b)]) \longrightarrow (f\ b - f\ a) *$ *vector_norm* $(vector\_sub\ b\ x) \leq (f\ b - f\ x) *$ *vector_norm* $(vector\_sub\ b\ a))$

thm CONVEX_ON_LEFT_SECANT_MUL:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ s::(real, ?'a::type)\ cart \Rightarrow bool.$ *convex_on f s* $= (\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.$ *IN a s* $\wedge$ *IN b s* $\wedge$ *IN x* $(closed\_segment\ [(a,\ b)]) \longrightarrow (f\ x - f\ a) *$ *vector_norm* $(vector\_sub\ b\ a) \leq (f\ b - f\ a) *$ *vector_norm* $(vector\_sub\ x\ a))$

thm CONVEX_ON_RIGHT_SECANT:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ s::(real, ?'a::type)\ cart \Rightarrow bool.$ *convex_on f s* $= (\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.$ *IN a s* $\wedge$ *IN b s* $\wedge$ *IN x* $(open\_segment\ (a,\ b)) \longrightarrow (f\ b - f\ a)\ /$ *vector_norm* $(vector\_sub\ b\ a) \leq (f\ b - f\ x)\ /$ *vector_norm* $(vector\_sub\ b\ x))$

thm CONVEX_ON_LEFT_SECANT:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ s::(real, ?'a::type)\ cart \Rightarrow bool.$ *convex_on f s* $= (\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.$ *IN a s* $\wedge$ *IN b s* $\wedge$ *IN x* $(open\_segment\ (a,\ b)) \longrightarrow (f\ x - f\ a)\ /$ *vector_norm* $(vector\_sub\ x\ a) \leq (f\ b - f\ a)\ /$ *vector_norm* $(vector\_sub\ b\ a))$

thm DEF_starlike:

$starlike = (\lambda\_325238::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, a::(real,\ ?'a::type)\ cart.\ IN\ a\ \_325238 \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_325238 \longrightarrow SUBSET\ (closed\_segment\ [(a,\ x)])\ \_325238))$

thm starlike:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ starlike\ s = (\exists\, a::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow SUBSET\ (closed\_segment\ [(a,\ x)])\ s))$

thm CONVEX_CONTAINS_SEGMENT:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s = (\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge IN\ b\ s \longrightarrow SUBSET\ (closed\_segment\ [(a,\ b)])\ s)$

thm CONVEX_IMP_STARLIKE:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge s \neq EMPTY \longrightarrow starlike\ s$

thm SEGMENT_CONVEX_HULL:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ closed\_segment\ [(a,\ b)] = hull\ convex\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm SEGMENT_FURTHEST_LE:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ (closed\_segment\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ y\ x) \leq vector\_norm\ (vector\_sub\ y\ a) \vee vector\_norm\ (vector\_sub\ y\ x) \leq vector\_norm\ (vector\_sub\ y\ b)$

thm SEGMENT_BOUND:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closed\_segment\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ x\ a) \leq vector\_norm\ (vector\_sub\ b\ a) \wedge vector\_norm\ (vector\_sub\ x\ b) \leq vector\_norm\ (vector\_sub\ b\ a)$

thm BETWEEN_IN_CONVEX_HULL:

$\forall\, (x::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ between\ x\ (a,\ b) = IN\ x\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ EMPTY)))$

thm STARLIKE_LINEAR_IMAGE:

$\forall\, (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ starlike\ s \wedge linear\ f \longrightarrow starlike\ (IMAGE\ f\ s)$

thm STARLIKE_LINEAR_IMAGE_EQ:

$\forall\, (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall\, (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow starlike\ (IMAGE\ f\ s) = starlike\ s$

thm STARLIKE_TRANSLATION_EQ:

$\forall\,(a{::}(real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ starlike\ (IMAGE\ (vector\_add\ a)\ s) = starlike\ s$

thm BETWEEN_LINEAR_IMAGE_EQ:

$\forall\,(f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ (y{::}(real,\ ?'b{::}type)\ cart)\ z{::}(real,\ ?'b{::}type)\ cart.\ linear\,f \wedge (\forall\,(x{::}(real,\ ?'b{::}type)\ cart)\ y{::}(real,\ ?'b{::}type)\ cart.\ f\,x = f\,y \longrightarrow x = y) \longrightarrow between\ (f\,x)\ (f\,y,\ f\,z) = between\ x\ (y,\ z)$

thm BETWEEN_TRANSLATION:

$\forall\,(a{::}(real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'a{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ between\ (vector\_add\ a\ x)\ (vector\_add\ a\ y,\ vector\_add\ a\ (?z{::}(real,\ ?'a{::}type)\ cart)) = between\ x\ (y,\ ?z)$

thm STARLIKE_CLOSURE:

$\forall\,s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ starlike\ s \longrightarrow starlike\ (closure\ s)$

thm STARLIKE_UNIV:

$starlike\ HOL\_Light\_Import.UNIV$

thm IN_INTERIOR_CONVEX_SHRINK:

$\forall\,(s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (e{::}real)\ (x{::}(real,\ ?'a{::}type)\ cart)\ c{::}(real,\ ?'a{::}type)\ cart.\ convex\ s \wedge IN\ c\ (interior\ s) \wedge IN\ x\ s \wedge (0{::}real) < e \wedge e \leq (1{::}real) \longrightarrow IN\ (vector\_sub\ x\ (\%\ e\ (vector\_sub\ x\ c)))\ (interior\ s)$

thm IN_INTERIOR_CLOSURE_CONVEX_SHRINK:

$\forall\,(s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (e{::}real)\ (x{::}(real,\ ?'a{::}type)\ cart)\ c{::}(real,\ ?'a{::}type)\ cart.\ convex\ s \wedge IN\ c\ (interior\ s) \wedge IN\ x\ (closure\ s) \wedge (0{::}real) < e \wedge e \leq (1{::}real) \longrightarrow IN\ (vector\_sub\ x\ (\%\ e\ (vector\_sub\ x\ c)))\ (interior\ s)$

thm IN_INTERIOR_CLOSURE_CONVEX_SEGMENT:

$\forall\,(s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (a{::}(real,\ ?'a{::}type)\ cart)\ b{::}(real,\ ?'a{::}type)\ cart.\ convex\ s \wedge IN\ a\ (interior\ s) \wedge IN\ b\ (closure\ s) \longrightarrow SUBSET\ (open\_segment\ (a,\ b))\ (interior\ s)$

thm DEF_relative_interior:

$relative\_interior = (\lambda\_326075{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1067{::}(real,\ ?'a{::}type)\ cart.\ \exists\,x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1067\ (\exists\,t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (hull\ affine\ \_326075))\ t \wedge IN\ x\ t \wedge SUBSET\ t\ \_326075)\ x))$

thm relative_interior:

$\forall\,s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ relative\_interior\ s = GSPEC\ (\lambda GEN\%PVAR\%1067{::}(real,\ ?'a{::}type)\ cart.\ \exists\,x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1067\ (\exists\,t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (hull\ affine\ s))\ t \wedge IN\ x\ t \wedge SUBSET\ t\ s)\ x)$

thm RELATIVE_INTERIOR:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *relative_interior s* = *GSPEC* (λ*GEN%PVAR%1068*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1068* (*IN x s* ∧ (∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open t* ∧ *IN x t* ∧ *SUBSET* (*HOL_Light_Import.INTER t* (*hull affine s*)) *s*)) *x*)

thm RELATIVE_INTERIOR_EQ:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*relative_interior s* = *s*) = *open_in* (*subtopology euclidean* (*hull affine s*)) *s*

thm RELATIVE_INTERIOR_OPEN_IN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *open_in* (*subtopology euclidean* (*hull affine s*)) *s* ⟶ *relative_interior s* = *s*

thm RELATIVE_INTERIOR_EMPTY:

relative_interior EMPTY = *EMPTY*

thm RELATIVE_INTERIOR_AFFINE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *affine s* ⟶ *relative_interior s* = *s*

thm RELATIVE_INTERIOR_UNIV:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *relative_interior* (*hull affine s*) = *hull affine s*

thm OPEN_IN_RELATIVE_INTERIOR:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *open_in* (*subtopology euclidean* (*hull affine s*)) (*relative_interior s*)

thm RELATIVE_INTERIOR_SUBSET:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*relative_interior s*) *s*

thm SUBSET_RELATIVE_INTERIOR:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ∧ *hull affine s* = *hull affine t* ⟶ *SUBSET* (*relative_interior s*) (*relative_interior t*)

thm RELATIVE_INTERIOR_MAXIMAL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *open_in* (*subtopology euclidean* (*hull affine s*)) *t* ⟶ *SUBSET t* (*relative_interior s*)

thm RELATIVE_INTERIOR_UNIQUE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *open_in* (*subtopology euclidean* (*hull affine s*)) *t* ∧ (∀ *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t' s* ∧ *open_in* (*subtopology euclidean* (*hull affine s*)) *t'* ⟶ *SUBSET t' t*) ⟶ *relative_interior s* = *t*

thm IN_RELATIVE_INTERIOR:

$\forall (x::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (relative\_interior\ s) = (IN\ x\ s \wedge (\exists\ e>0::real.\ SUBSET\ (HOL\_Light\_Import.INTER\ (ball\ (x, e))\ (hull\ affine\ s))\ s))$

thm IN_RELATIVE_INTERIOR_CBALL:

$\forall (x::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ (relative\_interior\ s) = (IN\ x\ s \wedge (\exists\ e>0::real.\ SUBSET\ (HOL\_Light\_Import.INTER\ (cball\ (x, e))\ (hull\ affine\ s))\ s))$

thm OPEN_IN_SUBSET_RELATIVE_INTERIOR:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (hull\ affine\ t))\ s \longrightarrow SUBSET\ s\ (relative\_interior\ t) = SUBSET\ s\ t$

thm RELATIVE_INTERIOR_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ relative\_interior\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (relative\_interior\ s)$

thm RELATIVE_INTERIOR_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow relative\_interior\ (IMAGE\ f\ s) = IMAGE\ f\ (relative\_interior\ s)$

thm RELATIVE_INTERIOR_EQ_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow (relative\_interior\ s = EMPTY) = (s = EMPTY)$

thm RELATIVE_INTERIOR_INTERIOR:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ s = HOL\_Light\_Import.UNIV \longrightarrow relative\_interior\ s = interior\ s$

thm RELATIVE_INTERIOR_OPEN:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow relative\_interior\ s = s$

thm RELATIVE_INTERIOR_NONEMPTY_INTERIOR:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ interior\ s \neq EMPTY \longrightarrow relative\_interior\ s = interior\ s$

thm AFFINE_HULL_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ (hull\ convex\ s) = hull\ affine\ s$

thm INTERIOR_SIMPLEX_NONEMPTY:

∀ s::(real, ?'a::type) cart ⇒ bool. independent s ∧ HAS_SIZE s (dimindex HOL_Light_Import.UNIV) ⟶ (∃ a::(real, ?'a::type) cart. IN a (interior (hull convex (INSERT (vec (0::nat)) s))))

thm INTERIOR_SUBSET_RELATIVE_INTERIOR:

∀ s::(real, ?'a::type) cart ⇒ bool. SUBSET (interior s) (relative_interior s)

thm CONVEX_RELATIVE_INTERIOR:

∀ s::(real, ?'a::type) cart ⇒ bool. convex s ⟶ convex (relative_interior s)

thm IN_RELATIVE_INTERIOR_CONVEX_SHRINK:

∀ (s::(real, ?'a::type) cart ⇒ bool) (e::real) (x::(real, ?'a::type) cart) c::(real, ?'a::type) cart. convex s ∧ IN c (relative_interior s) ∧ IN x s ∧ (0::real) < e ∧ e ≤ (1::real) ⟶ IN (vector_sub x (% e (vector_sub x c))) (relative_interior s)

thm IN_RELATIVE_INTERIOR_CLOSURE_CONVEX_SHRINK:

∀ (s::(real, ?'a::type) cart ⇒ bool) (e::real) (x::(real, ?'a::type) cart) c::(real, ?'a::type) cart. convex s ∧ IN c (relative_interior s) ∧ IN x (closure s) ∧ (0::real) < e ∧ e ≤ (1::real) ⟶ IN (vector_sub x (% e (vector_sub x c))) (relative_interior s)

thm IN_RELATIVE_INTERIOR_CLOSURE_CONVEX_SEGMENT:

∀ (s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. convex s ∧ IN a (relative_interior s) ∧ IN b (closure s) ⟶ SUBSET (open_segment (a, b)) (relative_interior s)

thm RELATIVE_INTERIOR_SING:

∀ a::(real, ?'a::type) cart. relative_interior (INSERT a EMPTY) = INSERT a EMPTY

thm RELATIVE_INTERIOR_PROLONG:

∀ (s::(real, ?'a::type) cart ⇒ bool) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x (relative_interior s) ∧ IN y s ⟶ (∃ t>1::real. IN (vector_add y (% t (vector_sub x y))) s)

thm RELATIVE_INTERIOR_CONVEX_PROLONG:

∀ s::(real, ?'a::type) cart ⇒ bool. convex s ⟶ relative_interior s = GSPEC (λGEN%PVAR%1069::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1069 (IN x s ∧ (∀ y::(real, ?'a::type) cart. IN y s ⟶ (∃ t>1::real. IN (vector_add y (% t (vector_sub x y))) s))) x)

thm RELATIVE_INTERIOR_EQ_CLOSURE:

∀ s::(real, ?'a::type) cart ⇒ bool. (relative_interior s = closure s) = affine s

thm RAY_TO_RELATIVE_FRONTIER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ bounded\ s \wedge IN\ a\ (relative\_interior\ s) \wedge IN\ (vector\_add\ a\ l)\ (hull\ affine\ s) \wedge l \neq vec\ (0::nat) \longrightarrow (\exists d>0::real.\ IN\ (vector\_add\ a\ (\%\ d\ l))\ (DIFF\ (closure\ s)\ (relative\_interior\ s)) \wedge (\forall e::real.\ (0::real) \leq e \wedge e < d \longrightarrow IN\ (vector\_add\ a\ (\%\ e\ l))\ (relative\_interior\ s)))$

thm RAY_TO_FRONTIER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ bounded\ s \wedge IN\ a\ (interior\ s) \wedge l \neq vec\ (0::nat) \longrightarrow (\exists d>0::real.\ IN\ (vector\_add\ a\ (\%\ d\ l))\ (frontier\ s) \wedge (\forall e::real.\ (0::real) \leq e \wedge e < d \longrightarrow IN\ (vector\_add\ a\ (\%\ e\ l))\ (interior\ s)))$

thm CONVEX_CLOSURE_INTERIOR:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge interior\ s \neq EMPTY \longrightarrow closure\ (interior\ s) = closure\ s$

thm EMPTY_INTERIOR_SUBSET_HYPERPLANE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge interior\ s = EMPTY \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge SUBSET\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1073::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1073\ (dot\ a\ x = b)\ x)))$

thm CONVEX_INTERIOR_CLOSURE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow interior\ (closure\ s) = interior\ s$

thm CONVEX_CLOSURE_RELATIVE_INTERIOR:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow closure\ (relative\_interior\ s) = closure\ s$

thm AFFINE_HULL_RELATIVE_INTERIOR:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow hull\ affine\ (relative\_interior\ s) = hull\ affine\ s$

thm CONVEX_RELATIVE_INTERIOR_CLOSURE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow relative\_interior\ (closure\ s) = relative\_interior\ s$

thm CONNECTED_INTER_RELATIVE_FRONTIER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ connected\ s \wedge SUBSET\ s\ (hull\ affine\ t) \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \wedge DIFF\ s\ t \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER\ s\ (DIFF\ (closure\ t)\ (relative\_interior\ t)) \neq EMPTY$

thm CLOSED_RELATIVE_FRONTIER:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ (DIFF\ (closure\ s)\ (relative\_interior\ s))$

thm CLOSED_RELATIVE_BOUNDARY:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (DIFF\ s\ (relative\_interior\ s))$

thm COMPACT_RELATIVE_BOUNDARY:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow compact\ (DIFF\ s\ (relative\_interior\ s))$

thm CONVEX_SAME_RELATIVE_INTERIOR_CLOSURE:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow (relative\_interior\ s = relative\_interior\ t) = (closure\ s = closure\ t)$

thm CONVEX_SAME_RELATIVE_INTERIOR_CLOSURE_STRADDLE:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \longrightarrow (relative\_interior\ s = relative\_interior\ t) = (SUBSET\ (relative\_interior\ s)\ t \wedge SUBSET\ t\ (closure\ s))$

thm RELATIVE_INTERIOR_LINEAR_IMAGE_CONVEX:

$\forall\, (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge convex\ s \longrightarrow relative\_interior\ (IMAGE\ f\ s) = IMAGE\ f\ (relative\_interior\ s)$

thm CLOSURE_INTERS_CONVEX:

$\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow convex\ s) \wedge INTERS\ (IMAGE\ relative\_interior\ f) \neq EMPTY \longrightarrow closure\ (INTERS\ f) = INTERS\ (IMAGE\ closure\ f)$

thm CLOSURE_INTERS_CONVEX_OPEN:

$\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow convex\ s \wedge HOL\_Light\_Import.open\ s) \longrightarrow closure\ (INTERS\ f) = (if\ INTERS\ f = EMPTY\ then\ EMPTY\ else\ INTERS\ (IMAGE\ closure\ f))$

thm CLOSURE_INTER_CONVEX:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge convex\ t \wedge HOL\_Light\_Import.INTER\ (relative\_interior\ s)\ (relative\_interior\ t) \neq EMPTY \longrightarrow closure\ (HOL\_Light\_Import.INTER\ s\ t) = HOL\_Light\_Import.INTER\ (closure\ s)\ (closure\ t)$

thm CLOSURE_INTER_CONVEX_OPEN:

$\forall\, (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge HOL\_Light\_Import.open\ s \wedge convex\ t \wedge HOL\_Light\_Import.open\ t \longrightarrow closure\ (HOL\_Light\_Import.INTER\ s\ t) = (if\ HOL\_Light\_Import.INTER\ s\ t = EMPTY\ then\ EMPTY\ else\ HOL\_Light\_Import.INTER\ (closure\ s)\ (closure\ t))$

thm CLOSURE_CONVEX_INTER_SUPERSET:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. convex s \wedge interior s \neq EMPTY \wedge SUBSET (interior s) (closure t) \longrightarrow closure (HOL\_Light\_Import.INTER s t) = closure s$

thm CLOSURE_DYADIC_RATIONALS_IN_CONVEX_SET:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. convex s \wedge interior s \neq EMPTY \longrightarrow closure (HOL\_Light\_Import.INTER s (GSPEC (\lambda GEN\%PVAR\%1075::(real, ?'a::type) cart. \exists (n::nat) x::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%1075 (\forall i::nat. (1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow integer (\$ x i)) (\% (inverse\_class.inverse (real\_of\_nat (2::nat))^n) x)))) = closure s$

thm CLOSURE_RATIONALS_IN_CONVEX_SET:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. convex s \wedge interior s \neq EMPTY \longrightarrow closure (HOL\_Light\_Import.INTER s (GSPEC (\lambda GEN\%PVAR\%1076::(real, ?'a::type) cart. \exists x::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%1076 (\forall i::nat. (1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow rational (\$ x i)) x))) = closure s$

thm SEPARATING_HYPERPLANE_RELATIVE_INTERIORS:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. convex s \wedge convex t \wedge \neg (s = EMPTY \wedge t = HOL\_Light\_Import.UNIV \vee s = HOL\_Light\_Import.UNIV \wedge t = EMPTY) \wedge DISJOINT (relative\_interior s) (relative\_interior t) \longrightarrow (\exists (a::(real, ?'a::type) cart) b::real. a \neq vec (0::nat) \wedge (\forall x::(real, ?'a::type) cart. IN x s \longrightarrow dot a x \leq b) \wedge (\forall x::(real, ?'a::type) cart. IN x t \longrightarrow b \leq dot a x))$

thm SUPPORTING_HYPERPLANE_RELATIVE_BOUNDARY:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) x::(real, ?'a::type) cart. convex s \wedge IN x s \wedge \neg IN x (relative\_interior s) \longrightarrow (\exists a::(real, ?'a::type) cart. a \neq vec (0::nat) \wedge (\forall y::(real, ?'a::type) cart. IN y s \longrightarrow dot a x \leq dot a y) \wedge (\forall y::(real, ?'a::type) cart. IN y (relative\_interior s) \longrightarrow dot a x < dot a y))$

thm SUPPORTING_HYPERPLANE_RELATIVE_FRONTIER:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) x::(real, ?'a::type) cart. convex s \wedge IN x (closure s) \wedge \neg IN x (relative\_interior s) \longrightarrow (\exists a::(real, ?'a::type) cart. a \neq vec (0::nat) \wedge (\forall y::(real, ?'a::type) cart. IN y (closure s) \longrightarrow dot a x \leq dot a y) \wedge (\forall y::(real, ?'a::type) cart. IN y (relative\_interior s) \longrightarrow dot a x < dot a y))$

thm EXPLICIT_SUBSET_RELATIVE_INTERIOR_CONVEX_HULL:

$\forall s::(real, ?'a::type) cart \Rightarrow bool. FINITE s \longrightarrow SUBSET (GSPEC (\lambda GEN\%PVAR\%1077::(real, ?'a::type) cart. \exists y::(real, ?'a::type) cart. SETSPEC GEN\%PVAR\%1077 (\exists u::(real, ?'a::type) cart \Rightarrow real. (\forall x::(real, ?'a::type) cart. IN x s \longrightarrow (0::real) < u x \wedge u x < (1::real)) \wedge sum s u = (1::real) \wedge vsum s (\lambda x::(real, ?'a::type) cart. \% (u x) x) = y) y)) (relative\_interior (hull convex s))$

thm EXPLICIT_SUBSET_RELATIVE_INTERIOR_CONVEX_HULL_MINIMAL:

$\forall s$::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *SUBSET* (*GSPEC* (λ*GEN%PVAR%1078*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1078* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x*) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)) (*relative_interior* (*hull convex s*))

thm RELATIVE_INTERIOR_CONVEX_HULL_EXPLICIT:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *relative_interior* (*hull convex s*) = *GSPEC* (λ*GEN%PVAR%1081*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1081* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x*) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)

thm EXPLICIT_SUBSET_INTERIOR_CONVEX_HULL:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ *hull affine s* = *HOL_Light_Import*.*UNIV* ⟶ *SUBSET* (*GSPEC* (λ*GEN%PVAR%1082*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1082* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x* ∧ *u x* < (*1*::*real*)) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)) (*interior* (*hull convex s*))

thm EXPLICIT_SUBSET_INTERIOR_CONVEX_HULL_MINIMAL:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ *hull affine s* = *HOL_Light_Import*.*UNIV* ⟶ *SUBSET* (*GSPEC* (λ*GEN%PVAR%1083*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1083* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x*) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)) (*interior* (*hull convex s*))

thm INTERIOR_CONVEX_HULL_EXPLICIT_MINIMAL:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *interior* (*hull convex s*) = (*if CARD s* ≤ *dimindex HOL_Light_Import*.*UNIV then EMPTY else GSPEC* (λ*GEN%PVAR%1084*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1084* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x*) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*))

thm INTERIOR_CONVEX_HULL_EXPLICIT:

∀ *s*::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *interior* (*hull convex s*) = (*if CARD s* ≤ *dimindex HOL_Light_Import*.*UNIV then EMPTY else GSPEC* (λ*GEN%PVAR%1085*::(*real*, ?'*a*::*type*) *cart*. ∃ *y*::(*real*, ?'*a*::*type*) *cart*. *SETSPEC GEN%PVAR%1085* (∃ *u*::(*real*, ?'*a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, ?'*a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) < *u x* ∧ *u x* < (*1*::*real*)) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, ?'*a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*))

thm INTERIOR_CONVEX_HULL_3_MINIMAL:

∀ (*a*::(*real*, *2*) *cart*) (*b*::(*real*, *2*) *cart*) *c*::(*real*, *2*) *cart*. ¬ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) ⟶ *interior* (*hull convex* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))) = *GSPEC* (λ*GEN%PVAR%1086*::(*real*, *2*) *cart*. ∃ *v*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%1086* (∃ (*x*::*real*) (*y*::*real*) *z*::*real*. (*0*::*real*) < *x* ∧ (*0*::*real*) < *y* ∧ (*0*::*real*) < *z* ∧ *x* + (*y* + *z*) = (*1*::*real*) ∧ *vector_add* (% *x a*) (*vector_add* (% *y b*) (% *z c*)) = *v*) *v*)

thm INTERIOR_CONVEX_HULL_3:

∀ (*a*::(*real*, *2*) *cart*) (*b*::(*real*, *2*) *cart*) *c*::(*real*, *2*) *cart*. ¬ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) ⟶ *interior* (*hull convex* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))) = *GSPEC* (λ*GEN%PVAR%1087*::(*real*, *2*) *cart*. ∃ *v*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%1087* (∃ (*x*::*real*) (*y*::*real*) *z*::*real*. (*0*::*real*) < *x* ∧ *x* < (*1*::*real*) ∧ (*0*::*real*) < *y* ∧ *y* < (*1*::*real*) ∧ (*0*::*real*) < *z* ∧ *z* < (*1*::*real*) ∧ *x* + (*y* + *z*) = (*1*::*real*) ∧ *vector_add* (% *x a*) (*vector_add* (% *y b*) (% *z c*)) = *v*) *v*)

thm CLOSURE_CONVEX_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ *closure* (*hull convex s*) = *hull convex s*

thm RELATIVE_FRONTIER_CONVEX_HULL_EXPLICIT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *DIFF* (*closure* (*hull convex s*)) (*relative_interior* (*hull convex s*)) = *GSPEC* (λ*GEN%PVAR%1088*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1088* (∃ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *u x*) ∧ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *u x* = (*0*::*real*)) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)

thm FRONTIER_CONVEX_HULL_EXPLICIT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *frontier* (*hull convex s*) = *GSPEC* (λ*GEN%PVAR%1089*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1089* (∃ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (*0*::*real*) ≤ *u x*) ∧ (*dimindex HOL_Light_Import.UNIV* < *CARD s* ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *u x* = (*0*::*real*))) ∧ *sum s u* = (*1*::*real*) ∧ *vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*) = *y*) *y*)

thm RELATIVE_FRONTIER_CONVEX_HULL_CASES:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *DIFF* (*closure* (*hull convex s*)) (*relative_interior* (*hull convex s*)) = *UNIONS* (*GSPEC* (λ*GEN%PVAR%1090*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *a*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1090* (*IN a s*) (*hull convex* (*DELETE s a*))))

thm FRONTIER_CONVEX_HULL_CASES:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ⟶ *frontier* (*hull convex s*) = (*if CARD s* ≤ *dimindex HOL_Light_Import.UNIV then hull convex s else UNIONS* (*GSPEC* (λ*GEN%PVAR%1091*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*.

$\exists$ *a::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1091 (IN a s) (hull convex (DELETE s a)))))*

thm IN_FRONTIER_CONVEX_HULL:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) x::(real, ?'a::type) cart. FINITE s* $\wedge$ *CARD s* $\leq$ *dimindex HOL_Light_Import.UNIV + (1::nat)* $\wedge$ *IN x s* $\longrightarrow$ *IN x (frontier (hull convex s))*

thm NOT_IN_INTERIOR_CONVEX_HULL:

$\forall$ *(s::(real, ?'a::type) cart* $\Rightarrow$ *bool) x::(real, ?'a::type) cart. FINITE s* $\wedge$ *CARD s* $\leq$ *dimindex HOL_Light_Import.UNIV + (1::nat)* $\wedge$ *IN x s* $\longrightarrow$ $\neg$ *IN x (interior (hull convex s))*

thm INTERIOR_CONVEX_HULL_EQ_EMPTY:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. HAS_SIZE s (dimindex HOL_Light_Import.UNIV + (1::nat))* $\longrightarrow$ *(interior (hull convex s) = EMPTY) = affine_dependent s*

thm SIMPLEX_EXPLICIT:

$\forall$ *s::(real, ?'a::type) cart* $\Rightarrow$ *bool. FINITE s* $\wedge$ $\neg$ *IN (vec (0::nat)) s* $\longrightarrow$ *hull convex (INSERT (vec (0::nat)) s) = GSPEC (*λ*GEN%PVAR%1092::(real, ?'a::type) cart.* $\exists$ *y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1092 (*$\exists$ *u::(real, ?'a::type) cart* $\Rightarrow$ *real. (*$\forall$ *x::(real, ?'a::type) cart. IN x s* $\longrightarrow$ *(0::real)* $\leq$ *u x)* $\wedge$ *sum s u* $\leq$ *(1::real)* $\wedge$ *vsum s (*λ*x::(real, ?'a::type) cart. % (u x) x) = y) y)*

thm STD_SIMPLEX:

hull convex (INSERT (vec (0::nat)) (GSPEC (λ*GEN%PVAR%1093::(real, ?'a::type) cart.* $\exists$ *i::nat. SETSPEC GEN%PVAR%1093 ((1::nat)* $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV) (basis i)))) = GSPEC (*λ*GEN%PVAR%1094::(real, ?'a::type) cart.* $\exists$ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1094 ((*$\forall$ *i::nat. (1::nat)* $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *(0::real)* $\leq$ *$ x i)* $\wedge$ *sum (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) ($ x)* $\leq$ *(1::real)) x)*

thm INTERIOR_STD_SIMPLEX:

interior (hull convex (INSERT (vec (0::nat)) (GSPEC (λ*GEN%PVAR%1095::(real, ?'a::type) cart.* $\exists$ *i::nat. SETSPEC GEN%PVAR%1095 ((1::nat)* $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV) (basis i))))) = GSPEC (*λ*GEN%PVAR%1096::(real, ?'a::type) cart.* $\exists$ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1096 ((*$\forall$ *i::nat. (1::nat)* $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *(0::real) < $ x i)* $\wedge$ *sum (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) ($ x) < (1::real)) x)*

thm DEF_path:

path = (λ*_332988::(real, unit) cart* $\Rightarrow$ *(real, ?'a::type) cart. continuous_on _332988 (closed_interval [(vec (0::nat), vec (1::nat))]))*

thm path:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\ g = continuous\_on\ g\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))])$

thm DEF_pathstart:

$pathstart = (\lambda\_332993::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ \_332993\ (vec\ (0::nat)))$

thm pathstart:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ pathstart\ g = g\ (vec\ (0::nat))$

thm DEF_pathfinish:

$pathfinish = (\lambda\_332998::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ \_332998\ (vec\ (1::nat)))$

thm pathfinish:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ pathfinish\ g = g\ (vec\ (1::nat))$

thm DEF_path_image:

$path\_image = (\lambda\_333003::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ IMAGE\ \_333003\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]))$

thm path_image:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\_image\ g = IMAGE\ g\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))])$

thm DEF_reversepath:

$reversepath = (\lambda(\_333008::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ x::(real, unit)\ cart.\ \_333008\ (vector\_sub\ (vec\ (1::nat))\ x))$

thm reversepath:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ reversepath\ g = (\lambda x::(real, unit)\ cart.\ g\ (vector\_sub\ (vec\ (1::nat))\ x))$

thm DEF_++:

$++ = (\lambda(\_333013::(real, unit)\ cart \Rightarrow ?'a::type)\ (\_333014::(real, unit)\ cart \Rightarrow ?'a::type)\ x::(real, unit)\ cart.\ if\ HOL\_Light\_Import.drop\ x \le (1::real)\ /\ real\_of\_nat\ (2::nat)\ then\ \_333013\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ else\ \_333014\ (vector\_sub\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ (vec\ (1::nat))))$

thm joinpaths:

$\forall (g1::(real, unit)\ cart \Rightarrow ?'a::type)\ g2::(real, unit)\ cart \Rightarrow ?'a::type.\ ++\ g1\ g2 = (\lambda x::(real, unit)\ cart.\ if\ HOL\_Light\_Import.drop\ x \le (1::real)\ /\ real\_of\_nat\ (2::nat)\ then\ g1\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ else\ g2\ (vector\_sub\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ (vec\ (1::nat))))$

thm DEF_simple_path:

simple_path = (λ*_333025*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path _333025* ∧ (∀ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN y* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *_333025 x* = *_333025 y* ⟶ *x* = *y* ∨ *x* = *vec* (*0*::*nat*) ∧ *y* = *vec* (*1*::*nat*) ∨ *x* = *vec* (*1*::*nat*) ∧ *y* = *vec* (*0*::*nat*)))

thm simple_path:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *simple_path g* = (*path g* ∧ (∀ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN y* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *g x* = *g y* ⟶ *x* = *y* ∨ *x* = *vec* (*0*::*nat*) ∧ *y* = *vec* (*1*::*nat*) ∨ *x* = *vec* (*1*::*nat*) ∧ *y* = *vec* (*0*::*nat*)))

thm DEF_arc:

arc = (λ*_333030*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path _333030* ∧ (∀ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN y* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *_333030 x* = *_333030 y* ⟶ *x* = *y*))

thm arc:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* = (*path g* ∧ (∀ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN y* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *g x* = *g y* ⟶ *x* = *y*))

thm PATH_CONTINUOUS_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *path g* ∧ *continuous_on f* (*path_image g*) ⟶ *path* (*f* ∘ *g*)

thm PATH_TRANSLATION_EQ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path* (*vector_add a* ∘ *g*) = *path g*

thm PATH_LINEAR_IMAGE_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *path* (*f* ∘ *g*) = *path g*

thm PATHSTART_TRANSLATION:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathstart* (*vector_add a* ∘ *g*) = *vector_add a* (*pathstart g*)

thm PATHSTART_LINEAR_IMAGE_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *linear f* ⟶ *pathstart* (*f* ∘ *g*) = *f* (*pathstart g*)

thm PATHFINISH_TRANSLATION:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish* (*vector_add a* ∘ *g*) = *vector_add a* (*pathfinish g*)

thm PATHFINISH_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow pathfinish\ (f \circ g) = f\ (pathfinish\ g)$

thm PATH_IMAGE_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\_image\ (vector\_add\ a \circ g) = IMAGE\ (vector\_add\ a)\ (path\_image\ g)$

thm PATH_IMAGE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow path\_image\ (f \circ g) = IMAGE\ f\ (path\_image\ g)$

thm REVERSEPATH_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ reversepath\ (vector\_add\ a \circ g) = vector\_add\ a \circ reversepath\ g$

thm REVERSEPATH_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow reversepath\ (f \circ g) = f \circ reversepath\ g$

thm JOINPATHS_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ ++\ (vector\_add\ a \circ g1)\ (vector\_add\ a \circ g2) = vector\_add\ a \circ ++\ g1\ g2$

thm JOINPATHS_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow ++\ (f \circ g1)\ (f \circ g2) = f \circ ++\ g1\ g2$

thm SIMPLE_PATH_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ simple\_path\ (vector\_add\ a \circ g) = simple\_path\ g$

thm SIMPLE_PATH_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow simple\_path\ (f \circ g) = simple\_path\ g$

thm ARC_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ arc\ (vector\_add\ a \circ g) = arc\ g$

thm ARC_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow arc\ (f \circ g) = arc\ g$

thm ARC_IMP_SIMPLE_PATH:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* ⟶ *simple_path g*

thm ARC_IMP_PATH:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* ⟶ *path g*

thm SIMPLE_PATH_IMP_PATH:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *simple_path g* ⟶ *path g*

thm SIMPLE_PATH_CASES:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *simple_path g* ⟶ *arc g* ∨ *pathfinish g* = *pathstart g*

thm SIMPLE_PATH_IMP_ARC:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *simple_path g* ∧ *pathfinish g* ≠ *pathstart g* ⟶ *arc g*

thm ARC_DISTINCT_ENDS:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* ⟶ *pathfinish g* ≠ *pathstart g*

thm ARC_SIMPLE_PATH:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* = (*simple_path g* ∧ *pathfinish g* ≠ *pathstart g*)

thm SIMPLE_PATH_EQ_ARC:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathstart g* ≠ *pathfinish g* ⟶ *simple_path g* = *arc g*

thm PATH_IMAGE_NONEMPTY:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path_image g* ≠ *EMPTY*

thm PATHSTART_IN_PATH_IMAGE:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *IN* (*pathstart g*) (*path_image g*)

thm PATHFINISH_IN_PATH_IMAGE:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *IN* (*pathfinish g*) (*path_image g*)

thm CONNECTED_PATH_IMAGE:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path g* ⟶ *connected* (*path_image g*)

thm COMPACT_PATH_IMAGE:

$\forall g$::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path g* ⟶ *compact* (*path_image g*)

thm BOUNDED_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\ g \longrightarrow bounded\ (path\_image\ g)$

thm CLOSED_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\ g \longrightarrow HOL\_Light\_Import.closed\ (path\_image\ g)$

thm CONNECTED_SIMPLE_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ simple\_path\ g \longrightarrow connected\ (path\_image\ g)$

thm COMPACT_SIMPLE_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ simple\_path\ g \longrightarrow compact\ (path\_image\ g)$

thm BOUNDED_SIMPLE_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ simple\_path\ g \longrightarrow bounded\ (path\_image\ g)$

thm CLOSED_SIMPLE_PATH_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ simple\_path\ g \longrightarrow HOL\_Light\_Import.closed\ (path\_image\ g)$

thm CONNECTED_ARC_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ arc\ g \longrightarrow connected\ (path\_image\ g)$

thm COMPACT_ARC_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ arc\ g \longrightarrow compact\ (path\_image\ g)$

thm BOUNDED_ARC_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ arc\ g \longrightarrow bounded\ (path\_image\ g)$

thm CLOSED_ARC_IMAGE:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ arc\ g \longrightarrow HOL\_Light\_Import.closed\ (path\_image\ g)$

thm PATHSTART_COMPOSE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ p::(real, unit)\ cart \Rightarrow (real, ?'b::type)\ cart.\ pathstart\ (f \circ p) = f\ (pathstart\ p)$

thm PATHFINISH_COMPOSE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ p::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ pathfinish\ (f \circ p) = f\ (pathfinish\ p)$

thm PATH_IMAGE_COMPOSE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ p::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ path\_image\ (f \circ p) = IMAGE\ f\ (path\_image\ p)$

thm PATH_COMPOSE_JOIN:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (p::(real,\ unit)\ cart \Rightarrow ?'b::type)\ q::(real,\ unit)\ cart \Rightarrow ?'b::type.\ f \circ ++\ p\ q = ++\ (f \circ p)\ (f \circ q)$

thm PATH_COMPOSE_REVERSEPATH:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ p::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ f \circ reversepath\ p = reversepath\ (f \circ p)$

thm SIMPLE_PATH_ENDLESS:

$\forall c::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ simple\_path\ c \longrightarrow DIFF\ (path\_image\ c)\ (INSERT\ (pathstart\ c)\ (INSERT\ (pathfinish\ c)\ EMPTY)) = IMAGE\ c\ (open\_interval\ (vec\ (0::nat),\ vec\ (1::nat)))$

thm CONNECTED_SIMPLE_PATH_ENDLESS:

$\forall c::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ simple\_path\ c \longrightarrow connected\ (DIFF\ (path\_image\ c)\ (INSERT\ (pathstart\ c)\ (INSERT\ (pathfinish\ c)\ EMPTY)))$

thm NONEMPTY_SIMPLE_PATH_ENDLESS:

$\forall c::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ simple\_path\ c \longrightarrow DIFF\ (path\_image\ c)\ (INSERT\ (pathstart\ c)\ (INSERT\ (pathfinish\ c)\ EMPTY)) \neq EMPTY$

thm JOINPATHS:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ g1 = pathstart\ g2 \longrightarrow ++\ g1\ g2 = (\lambda x::(real,\ unit)\ cart.\ if\ HOL\_Light\_Import.drop\ x < (1::real)\ /\ real\_of\_nat\ (2::nat)\ then\ g1\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ else\ g2\ (vector\_sub\ (\%\ (real\_of\_nat\ (2::nat))\ x)\ (vec\ (1::nat))))$

thm REVERSEPATH_REVERSEPATH:

$\forall g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ reversepath\ (reversepath\ g) = g$

thm PATHSTART_REVERSEPATH:

$pathstart\ (reversepath\ (?g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)) = pathfinish\ ?g$

thm PATHFINISH_REVERSEPATH:

$pathfinish\ (reversepath\ (?g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)) = pathstart\ ?g$

thm PATHSTART_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathstart\ (++\ g1\ g2) = pathstart\ g1$

thm PATHFINISH_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ (++\ g1\ g2) = pathfinish\ g2$

thm PATH_IMAGE_REVERSEPATH:

$\forall g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\_image\ (reversepath\ g) = path\_image\ g$

thm PATH_REVERSEPATH:

$\forall g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\ (reversepath\ g) = path\ g$

thm PATH_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ g1 = pathstart\ g2 \longrightarrow path\ (++\ g1\ g2) = (path\ g1 \wedge path\ g2)$

thm PATH_JOIN_IMP:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\ g1 \wedge path\ g2 \wedge pathfinish\ g1 = pathstart\ g2 \longrightarrow path\ (++\ g1\ g2)$

thm PATH_IMAGE_JOIN_SUBSET:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ SUBSET\ (path\_image\ (++\ g1\ g2))\ (HOL\_Light\_Import.UNION\ (path\_image\ g1)\ (path\_image\ g2))$

thm SUBSET_PATH_IMAGE_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (path\_image\ g1)\ s \wedge SUBSET\ (path\_image\ g2)\ s \longrightarrow SUBSET\ (path\_image\ (++\ g1\ g2))\ s$

thm PATH_IMAGE_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ g1 = pathstart\ g2 \longrightarrow path\_image\ (++\ g1\ g2) = HOL\_Light\_Import.UNION\ (path\_image\ g1)\ (path\_image\ g2)$

thm NOT_IN_PATH_IMAGE_JOIN:

$\forall (g1::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g2::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ \neg\ IN\ x\ (path\_image\ g1) \wedge \neg\ IN\ x\ (path\_image\ g2) \longrightarrow \neg\ IN\ x\ (path\_image\ (++\ g1\ g2))$

thm ARC_REVERSEPATH:

$\forall g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ arc\ g \longrightarrow arc\ (reversepath\ g)$

thm SIMPLE_PATH_REVERSEPATH:

$\forall g$::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *simple_path g* $\longrightarrow$ *simple_path* (*reversepath g*)

thm SIMPLE_PATH_JOIN_LOOP:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *arc g1* $\wedge$ *arc g2* $\wedge$ *pathfinish g1* = *pathstart g2* $\wedge$ *pathfinish g2* = *pathstart g1* $\wedge$ *SUBSET* (*HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*)) (*INSERT* (*pathstart g1*) (*INSERT* (*pathstart g2*) *EMPTY*)) $\longrightarrow$ *simple_path* (++ *g1 g2*)

thm ARC_JOIN:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *arc g1* $\wedge$ *arc g2* $\wedge$ *pathfinish g1* = *pathstart g2* $\wedge$ *SUBSET* (*HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*)) (*INSERT* (*pathstart g2*) *EMPTY*) $\longrightarrow$ *arc* (++ *g1 g2*)

thm REVERSEPATH_JOINPATHS:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *pathfinish g1* = *pathstart g2* $\longrightarrow$ *reversepath* (++ *g1 g2*) = ++ (*reversepath g2*) (*reversepath g1*)

thm ENDS_IN_UNIT_INTERVAL_conjunct3:

$\neg$ *IN* (*vec* (*1*::*nat*)) (*open_interval* (*vec* (*0*::*nat*), *vec* (*1*::*nat*)))

thm ENDS_IN_UNIT_INTERVAL_conjunct2:

$\neg$ *IN* (*vec* (*0*::*nat*)) (*open_interval* (*vec* (*0*::*nat*), *vec* (*1*::*nat*)))

thm ENDS_IN_UNIT_INTERVAL_conjunct1:

IN (*vec* (*1*::*nat*)) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))])

thm ENDS_IN_UNIT_INTERVAL_conjunct0:

IN (*vec* (*0*::*nat*)) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))])

thm PATH_JOIN_PATH_ENDS:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *path g2* $\wedge$ *path* (++ *g1 g2*) $\longrightarrow$ *pathfinish g1* = *pathstart g2*

thm PATH_JOIN_EQ:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *path g1* $\wedge$ *path g2* $\longrightarrow$ *path* (++ *g1 g2*) = (*pathfinish g1* = *pathstart g2*)

thm SIMPLE_PATH_JOIN_IMP:

$\forall$ (*g1*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *simple_path* (++ *g1 g2*) $\wedge$ *pathfinish g1* = *pathstart g2* $\longrightarrow$

arc g1 ∧ *arc g2* ∧ *SUBSET* (*HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*)) (*INSERT* (*pathstart g1*) (*INSERT* (*pathstart g2*) *EMPTY*))

thm SIMPLE_PATH_JOIN_LOOP_EQ:

∀(*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish g2* = *pathstart g1* ∧ *pathfinish g1* = *pathstart g2* ⟶ *simple_path* (++ *g1 g2*) = (*arc g1* ∧ *arc g2* ∧ *SUBSET* (*HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*)) (*INSERT* (*pathstart g1*) (*INSERT* (*pathstart g2*) *EMPTY*)))

thm ARC_JOIN_EQ:

∀(*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish g1* = *pathstart g2* ⟶ *arc* (++ *g1 g2*) = (*arc g1* ∧ *arc g2* ∧ *SUBSET* (*HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*)) (*INSERT* (*pathstart g2*) *EMPTY*))

thm ARC_JOIN_EQ_ALT:

∀(*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish g1* = *pathstart g2* ⟶ *arc* (++ *g1 g2*) = (*arc g1* ∧ *arc g2* ∧ *HOL_Light_Import.INTER* (*path_image g1*) (*path_image g2*) = *INSERT* (*pathstart g2*) *EMPTY*)

thm PATH_ASSOC:

∀(*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish p* = *pathstart q* ∧ *pathfinish q* = *pathstart r* ⟶ *path* (++ *p* (++ *q r*)) = *path* (++ (++ *p q*) *r*)

thm SIMPLE_PATH_ASSOC:

∀(*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish p* = *pathstart q* ∧ *pathfinish q* = *pathstart r* ⟶ *simple_path* (++ *p* (++ *q r*)) = *simple_path* (++ (++ *p q*) *r*)

thm ARC_ASSOC:

∀(*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish p* = *pathstart q* ∧ *pathfinish q* = *pathstart r* ⟶ *arc* (++ *p* (++ *q r*)) = *arc* (++ (++ *p q*) *r*)

thm PATH_SYM:

∀(*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish p* = *pathstart q* ∧ *pathfinish q* = *pathstart p* ⟶ *path* (++ *p q*) = *path* (++ *q p*)

thm SIMPLE_PATH_SYM:

$\forall (p::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ q::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ p = pathstart\ q \wedge pathfinish\ q = pathstart\ p \longrightarrow simple\_path\ (++\ p\ q) = simple\_path\ (++\ q\ p)$

thm PATH_IMAGE_SYM:

$\forall (p::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ q::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ p = pathstart\ q \wedge pathfinish\ q = pathstart\ p \longrightarrow path\_image\ (++\ p\ q) = path\_image\ (++\ q\ p)$

thm DEF_shiftpath:

$shiftpath = (\lambda(\_337727::(real,\ unit)\ cart)\ (\_337728::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ unit)\ cart.\ if\ HOL\_Light\_Import.drop\ (vector\_add\ \_337727\ x) \leq (1::real)\ then\ \_337728\ (vector\_add\ \_337727\ x)\ else\ \_337728\ (vector\_add\ \_337727\ (vector\_sub\ x\ (vec\ (1::nat)))))$

thm shiftpath:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ unit)\ cart.\ shiftpath\ a\ f = (\lambda x::(real,\ unit)\ cart.\ if\ HOL\_Light\_Import.drop\ (vector\_add\ a\ x) \leq (1::real)\ then\ f\ (vector\_add\ a\ x)\ else\ f\ (vector\_add\ a\ (vector\_sub\ x\ (vec\ (1::nat)))))$

thm SHIFTPATH_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (t::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ shiftpath\ t\ (vector\_add\ a \circ g) = vector\_add\ a \circ shiftpath\ t\ g$

thm SHIFTPATH_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (t::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ linear\ f \longrightarrow shiftpath\ t\ (f \circ g) = f \circ shiftpath\ t\ g$

thm PATHSTART_SHIFTPATH:

$\forall (a::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.drop\ a \leq (1::real) \longrightarrow pathstart\ (shiftpath\ a\ g) = g\ a$

thm PATHFINISH_SHIFTPATH:

$\forall (a::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (0::real) \leq HOL\_Light\_Import.drop\ a \wedge pathfinish\ g = pathstart\ g \longrightarrow pathfinish\ (shiftpath\ a\ g) = g\ a$

thm ENDPOINTS_SHIFTPATH:

$\forall (a::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ g = pathstart\ g \wedge IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \longrightarrow pathfinish\ (shiftpath\ a\ g) = g\ a \wedge pathstart\ (shiftpath\ a\ g) = g\ a$

thm CLOSED_SHIFTPATH:

$\forall (a::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ pathfinish\ g = pathstart\ g \wedge IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \longrightarrow pathfinish\ (shiftpath\ a\ g) = pathstart\ (shiftpath\ a\ g)$

thm PATH_SHIFTPATH:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ unit)\ cart.\ path\ g \wedge pathfinish\ g = pathstart\ g \wedge IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \longrightarrow path\ (shiftpath\ a\ g)$

thm SHIFTPATH_SHIFTPATH:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ x::(real,\ unit)\ cart.\ IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \wedge pathfinish\ g = pathstart\ g \wedge IN\ x\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \longrightarrow shiftpath\ (vector\_sub\ (vec\ (1::nat))\ a)\ (shiftpath\ a\ g)\ x = g\ x$

thm PATH_IMAGE_SHIFTPATH:

$\forall (a::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \wedge pathfinish\ g = pathstart\ g \longrightarrow path\_image\ (shiftpath\ a\ g) = path\_image\ g$

thm SIMPLE_PATH_SHIFTPATH:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ unit)\ cart.\ simple\_path\ g \wedge pathfinish\ g = pathstart\ g \wedge IN\ a\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \longrightarrow simple\_path\ (shiftpath\ a\ g)$

thm DEF_subpath:

$subpath = (\lambda(\_337889::(real,\ unit)\ cart)\ (\_337890::(real,\ unit)\ cart)\ (\_337891::(real,\ unit)\ cart \Rightarrow ?'a::type)\ x::(real,\ unit)\ cart.\ \_337891\ (vector\_add\ \_337889\ (\%\ (HOL\_Light\_Import.drop\ (vector\_sub\ \_337890\ \_337889))\ x)))$

thm subpath:

$\forall (g::(real,\ unit)\ cart \Rightarrow ?'a::type)\ (v::(real,\ unit)\ cart)\ u::(real,\ unit)\ cart.\ subpath\ u\ v\ g = (\lambda x::(real,\ unit)\ cart.\ g\ (vector\_add\ u\ (\%\ (HOL\_Light\_Import.drop\ (vector\_sub\ v\ u))\ x)))$

thm SUBPATH_SCALING_LEMMA:

$\forall (u::(real,\ unit)\ cart)\ v::(real,\ unit)\ cart.\ IMAGE\ (\lambda x::(real,\ unit)\ cart.\ vector\_add\ u\ (\%\ (HOL\_Light\_Import.drop\ (vector\_sub\ v\ u))\ x))\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) = closed\_segment\ [(u,\ v)]$

thm PATH_IMAGE_SUBPATH_GEN:

$\forall (u::(real,\ unit)\ cart)\ (v::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\_image\ (subpath\ u\ v\ g) = IMAGE\ g\ (closed\_segment\ [(u,\ v)])$

thm PATH_IMAGE_SUBPATH:

$\forall (u::(real,\ unit)\ cart)\ (v::(real,\ unit)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.drop\ u \leq HOL\_Light\_Import.drop\ v \longrightarrow path\_image\ (subpath\ u\ v\ g) = IMAGE\ g\ (closed\_interval\ [(u,\ v)])$

thm PATH_SUBPATH:

$\forall (u::(real, unit)\ cart)\ (v::(real, unit)\ cart)\ g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\ g \wedge IN\ u\ (closed\_interval\ [(vec\ (0::nat), vec\ (1::nat))]) \wedge IN\ v\ (closed\_interval\ [(vec\ (0::nat), vec\ (1::nat))]) \longrightarrow path\ (subpath\ u\ v\ g)$

thm PATHSTART_SUBPATH:

$\forall (u::(real, unit)\ cart)\ (v::(real, unit)\ cart)\ g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ pathstart\ (subpath\ u\ v\ g) = g\ u$

thm PATHFINISH_SUBPATH:

$\forall (u::(real, unit)\ cart)\ (v::(real, unit)\ cart)\ g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ pathfinish\ (subpath\ u\ v\ g) = g\ v$

thm SUBPATH_TRIVIAL:

$\forall g::(real, unit)\ cart \Rightarrow ?'a::type.\ subpath\ (vec\ (0::nat))\ (vec\ (1::nat))\ g = g$

thm SUBPATH_REVERSEPATH:

$\forall g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ subpath\ (vec\ (1::nat))\ (vec\ (0::nat))\ g = reversepath\ g$

thm REVERSEPATH_SUBPATH:

$\forall (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ reversepath\ (subpath\ u\ v\ g) = subpath\ v\ u\ g$

thm SUBPATH_TRANSLATION:

$\forall (a::real)\ (g::(real, unit)\ cart \Rightarrow real)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ subpath\ u\ v\ (op + a \circ g) = op + a \circ subpath\ u\ v\ g$

thm SUBPATH_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, unit)\ cart \Rightarrow (real, ?'b::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ linear\ f \longrightarrow subpath\ u\ v\ (f \circ g) = f \circ subpath\ u\ v\ g$

thm SIMPLE_PATH_SUBPATH_EQ:

$\forall (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ simple\_path\ (subpath\ u\ v\ g) = (path\ (subpath\ u\ v\ g) \wedge u \neq v \wedge (\forall (x::(real, unit)\ cart)\ y::(real, unit)\ cart.\ IN\ x\ (closed\_segment\ [(u, v)]) \wedge IN\ y\ (closed\_segment\ [(u, v)]) \wedge g\ x = g\ y \longrightarrow x = y \vee x = u \wedge y = v \vee x = v \wedge y = u))$

thm ARC_SUBPATH_EQ:

$\forall (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ arc\ (subpath\ u\ v\ g) = (path\ (subpath\ u\ v\ g) \wedge u \neq v \wedge (\forall (x::(real, unit)\ cart)\ y::(real, unit)\ cart.\ IN\ x\ (closed\_segment\ [(u, v)]) \wedge IN\ y\ (closed\_segment\ [(u, v)]) \wedge g\ x = g\ y \longrightarrow x = y))$

thm SIMPLE_PATH_SUBPATH:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *unit*) *cart*) *v*::(*real*, *unit*) *cart*. *simple_path g* $\wedge$ *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *IN v* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *u* $\neq$ *v* $\longrightarrow$ *simple_path* (*subpath u v g*)

thm ARC_SIMPLE_PATH_SUBPATH:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *unit*) *cart*) *v*::(*real*, *unit*) *cart*. *simple_path g* $\wedge$ *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *IN v* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *g u* $\neq$ *g v* $\longrightarrow$ *arc* (*subpath u v g*)

thm ARC_SIMPLE_PATH_SUBPATH_INTERIOR:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *unit*) *cart*) *v*::(*real*, *unit*) *cart*. *simple_path g* $\wedge$ *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *IN v* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *u* $\neq$ *v* $\wedge$ |*HOL_Light_Import.drop* *u* − *HOL_Light_Import.drop v*| < (*1*::*real*) $\longrightarrow$ *arc* (*subpath u v g*)

thm PATH_IMAGE_SUBPATH_SUBSET:

$\forall$ (*u*::(*real*, *unit*) *cart*) (*v*::(*real*, *unit*) *cart*) *g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *path g* $\wedge$ *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\wedge$ *IN v* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) $\longrightarrow$ *SUBSET* (*path_image* (*subpath u v g*)) (*path_image g*)

thm EXISTS_SUBPATH_OF_PATH:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *path g* $\wedge$ *IN a* (*path_image g*) $\wedge$ *IN b* (*path_image g*) $\longrightarrow$ ($\exists$ *h*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *path h* $\wedge$ *pathstart h* = *a* $\wedge$ *pathfinish h* = *b* $\wedge$ *SUBSET* (*path_image h*) (*path_image g*))

thm EXISTS_SUBPATH_OF_ARC_NOENDS:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *arc g* $\wedge$ *IN a* (*path_image g*) $\wedge$ *IN b* (*path_image g*) $\wedge$ *HOL_Light_Import.INTER* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT* (*pathstart g*) (*INSERT* (*pathfinish g*) *EMPTY*)) = *EMPTY* $\longrightarrow$ ($\exists$ *h*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *path h* $\wedge$ *pathstart h* = *a* $\wedge$ *pathfinish h* = *b* $\wedge$ *SUBSET* (*path_image h*) (*DIFF* (*path_image g*) (*INSERT* (*pathstart g*) (*INSERT* (*pathfinish g*) *EMPTY*))))

thm EXISTS_SUBARC_OF_ARC_NOENDS:

$\forall$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *arc g* $\wedge$ *IN a* (*path_image g*) $\wedge$ *IN b* (*path_image g*) $\wedge$ *a* $\neq$ *b* $\wedge$ *HOL_Light_Import.INTER* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT* (*pathstart g*) (*INSERT* (*pathfinish g*) *EMPTY*)) = *EMPTY* $\longrightarrow$ ($\exists$ *h*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *arc h* $\wedge$ *pathstart h* = *a* $\wedge$ *pathfinish h* = *b* $\wedge$ *SUBSET* (*path_image h*) (*DIFF* (*path_image g*) (*INSERT* (*pathstart g*) (*INSERT* (*pathfinish g*) *EMPTY*))))

thm EXISTS_ARC_PSUBSET_SIMPLE_PATH:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *simple_path g* ∧ *HOL_Light_Import.closed* (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ∧ *PSUBSET ?s* (*path_image g*) ⟶ (∃ *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc h* ∧ *SUBSET ?s* (*path_image h*) ∧ *SUBSET* (*path_image h*) (*path_image g*))

thm EXISTS_DOUBLE_ARC:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *simple_path g* ∧ *pathfinish g* = *pathstart g* ∧ *IN a* (*path_image g*) ∧ *IN b* (*path_image g*) ∧ *a* ≠ *b* ⟶ (∃ (*u*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *d*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc u* ∧ *arc d* ∧ *pathstart u* = *a* ∧ *pathfinish u* = *b* ∧ *pathstart d* = *b* ∧ *pathfinish d* = *a* ∧ *HOL_Light_Import.INTER* (*path_image u*) (*path_image d*) = *INSERT a* (*INSERT b EMPTY*) ∧ *HOL_Light_Import.UNION* (*path_image u*) (*path_image d*) = *path_image g*)

thm SUBPATH_TO_FRONTIER_EXPLICIT:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path g* ∧ *IN* (*pathstart g*) *s* ∧ ¬ *IN* (*pathfinish g*) *s* ⟶ (∃ *u*::(*real*, *unit*) *cart*. *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ (∀ *x*::(*real*, *unit*) *cart*. (*0*::*real*) ≤ *HOL_Light_Import.drop x* ∧ *HOL_Light_Import.drop x* < *HOL_Light_Import.drop u* ⟶ *IN* (*g x*) (*interior s*)) ∧ ¬ *IN* (*g u*) (*interior s*) ∧ (*u* = *vec* (*0*::*nat*) ∨ *IN* (*g u*) (*closure s*)))

thm SUBPATH_TO_FRONTIER_STRONG:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path g* ∧ *IN* (*pathstart g*) *s* ∧ ¬ *IN* (*pathfinish g*) *s* ⟶ (∃ *u*::(*real*, *unit*) *cart*. *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ ¬ *IN* (*pathfinish* (*subpath* (*vec* (*0*::*nat*)) *u g*)) (*interior s*) ∧ (*u* = *vec* (*0*::*nat*) ∨ (∀ *x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *x* ≠ *vec* (*1*::*nat*) ⟶ *IN* (*subpath* (*vec* (*0*::*nat*)) *u g x*) (*interior s*)) ∧ *IN* (*pathfinish* (*subpath* (*vec* (*0*::*nat*)) *u g*)) (*closure s*)))

thm SUBPATH_TO_FRONTIER:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path g* ∧ *IN* (*pathstart g*) *s* ∧ ¬ *IN* (*pathfinish g*) *s* ⟶ (∃ *u*::(*real*, *unit*) *cart*. *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN* (*pathfinish* (*subpath* (*vec* (*0*::*nat*)) *u g*)) (*frontier s*) ∧ *SUBSET* (*DELETE* (*path_image* (*subpath* (*vec* (*0*::*nat*)) *u g*)) (*pathfinish* (*subpath* (*vec* (*0*::*nat*)) *u g*))) (*interior s*))

thm EXISTS_PATH_SUBPATH_TO_FRONTIER:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path g* ∧ *IN* (*pathstart g*) *s* ∧ ¬ *IN* (*pathfinish g*) *s* ⟶ (∃ *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path h* ∧ *pathstart h* = *pathstart g* ∧ *SUBSET* (*path_image h*) (*path_image g*) ∧ *SUBSET* (*DELETE* (*path_image h*) (*pathfinish h*)) (*interior s*) ∧ *IN* (*pathfinish h*) (*frontier s*))

thm EXISTS_PATH_SUBPATH_TO_FRONTIER_CLOSED:

∀ (g::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed s ∧ path g ∧ IN (pathstart g) s ∧ ¬ IN (pathfinish g) s ⟶ (∃ h::(real, unit) cart ⇒ (real, ?'a::type) cart. path h ∧ pathstart h = pathstart g ∧ SUBSET (path_image h) (HOL_Light_Import.INTER (path_image g) s) ∧ IN (pathfinish h) (frontier s))

thm DEF_linepath:

linepath = (λ(_343744::(real, ?'a::type) cart × (real, ?'a::type) cart) x::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop x) (fst _343744)) (% (HOL_Light_Import.drop x) (snd _343744)))

thm linepath:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. linepath (a, b) = (λx::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop x) a) (% (HOL_Light_Import.drop x) b))

thm LINEPATH_TRANSLATION:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) c::(real, ?'a::type) cart. linepath (vector_add a b, vector_add a c) = vector_add a ∘ linepath (b, c)

thm LINEPATH_LINEAR_IMAGE:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. linear f ⟶ (∀ (b::(real, ?'b::type) cart) c::(real, ?'b::type) cart. linepath (f b, f c) = f ∘ linepath (b, c))

thm PATHSTART_LINEPATH:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. pathstart (linepath (a, b)) = a

thm PATHFINISH_LINEPATH:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. pathfinish (linepath (a, b)) = b

thm CONTINUOUS_LINEPATH_AT:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) x::(real, unit) cart. continuous (linepath (a, b)) (at x)

thm CONTINUOUS_ON_LINEPATH:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) s::(real, unit) cart ⇒ bool. continuous_on (linepath (a, b)) s

thm PATH_LINEPATH:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. path (linepath (a, b))

thm PATH_IMAGE_LINEPATH:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. path_image (linepath (a, b)) = closed_segment [(a, b)]

thm REVERSEPATH_LINEPATH:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ reversepath\ (linepath\ (a,\ b)) = linepath\ (b,\ a)$

thm ARC_LINEPATH:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ a \neq b \longrightarrow arc\ (linepath\ (a,\ b))$

thm SIMPLE_PATH_LINEPATH:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ a \neq b \longrightarrow simple\_path\ (linepath\ (a,\ b))$

thm SIMPLE_PATH_LINEPATH_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ simple\_path\ (linepath\ (a,\ b)) = (a \neq b)$

thm ARC_LINEPATH_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ arc\ (linepath\ (a,\ b)) = (a \neq b)$

thm LINEPATH_REFL:

$\forall\ a::(real,\ ?'a::type)\ cart.\ linepath\ (a,\ a) = (\lambda x::(real,\ unit)\ cart.\ a)$

thm SHIFTPATH_TRIVIAL:

$\forall (t::(real,\ unit)\ cart)\ a::(real,\ ?'a::type)\ cart.\ shiftpath\ t\ (linepath\ (a,\ a)) = linepath\ (a,\ a)$

thm SUBPATH_REFL:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ unit)\ cart.\ subpath\ a\ a\ g = linepath\ (g\ a,\ g\ a)$

thm NOT_ON_PATH_BALL:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ path\ g \wedge \neg\ IN\ z\ (path\_image\ g) \longrightarrow (\exists\ e{>}0::real.\ HOL\_Light\_Import.INTER\ (ball\ (z,\ e))\ (path\_image\ g) = EMPTY)$

thm NOT_ON_PATH_CBALL:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ path\ g \wedge \neg\ IN\ z\ (path\_image\ g) \longrightarrow (\exists\ e{>}0::real.\ HOL\_Light\_Import.INTER\ (cball\ (z,\ e))\ (path\_image\ g) = EMPTY)$

thm DEF_path_component:

$path\_component = (\lambda(\_343911::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (\_343912::(real,\ ?'a::type)\ cart)\ \_343913::(real,\ ?'a::type)\ cart.\ \exists\ g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\ g \wedge SUBSET\ (path\_image\ g)\ \_343911 \wedge pathstart\ g = \_343912 \wedge pathfinish\ g = \_343913)$

thm path_component:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x\ y = (\exists g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\ g \wedge SUBSET\ (path\_image\ g)\ s \wedge pathstart\ g = x \wedge pathfinish\ g = y)$

thm PATH_COMPONENT_IN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x\ y \longrightarrow IN\ x\ s \wedge IN\ y\ s$

thm PATH_COMPONENT_REFL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow path\_component\ s\ x\ x$

thm PATH_COMPONENT_REFL_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x\ x = IN\ x\ s$

thm PATH_COMPONENT_SYM:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x\ y \longrightarrow path\_component\ s\ y\ x$

thm PATH_COMPONENT_TRANS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x\ y \wedge path\_component\ s\ y\ (?z::(real,\ ?'a::type)\ cart) \longrightarrow path\_component\ s\ x\ ?z$

thm PATH_COMPONENT_SET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ path\_component\ s\ x = GSPEC\ (\lambda GEN\%PVAR\%1101::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1101\ (\exists g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path\ g \wedge SUBSET\ (path\_image\ g)\ s \wedge pathstart\ g = x \wedge pathfinish\ g = y)\ y)$

thm PATH_COMPONENT_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ SUBSET\ (path\_component\ s\ x)\ s$

thm PATH_COMPONENT_EQ_EMPTY:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ (path\_component\ s\ x = EMPTY) = (\neg\ IN\ x\ s)$

thm PATH_COMPONENT_EMPTY:

$\forall x::(real,\ ?'a::type)\ cart.\ path\_component\ EMPTY\ x = EMPTY$

thm PATH_COMPONENT_TRANSLATION:

∀ (a::(real, ?'a::type) cart) (s::(real, ?'a::type) cart ⇒ bool) x::(real, ?'a::type) cart. path_component (IMAGE (vector_add a) s) (vector_add a x) = IMAGE (vector_add a) (path_component s x)

thm PATH_COMPONENT_LINEAR_IMAGE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) x::(real, ?'b::type) cart. linear f ∧ (∀ (x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f x = f y ⟶ x = y) ∧ (∀ y::(real, ?'a::type) cart. ∃ x::(real, ?'b::type) cart. f x = y) ⟶ path_component (IMAGE f s) (f x) = IMAGE f (path_component s x)

thm DEF_path_connected:

path_connected = (λ_344101::(real, ?'a::type) cart ⇒ bool. ∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x _344101 ∧ IN y _344101 ⟶ (∃ g::(real, unit) cart ⇒ (real, ?'a::type) cart. path g ∧ SUBSET (path_image g) _344101 ∧ pathstart g = x ∧ pathfinish g = y))

thm path_connected:

∀ s::(real, ?'a::type) cart ⇒ bool. path_connected s = (∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y s ⟶ (∃ g::(real, unit) cart ⇒ (real, ?'a::type) cart. path g ∧ SUBSET (path_image g) s ∧ pathstart g = x ∧ pathfinish g = y))

thm PATH_CONNECTED_IFF_PATH_COMPONENT:

∀ s::(real, ?'a::type) cart ⇒ bool. path_connected s = (∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y s ⟶ path_component s x y)

thm PATH_CONNECTED_COMPONENT_SET:

∀ s::(real, ?'a::type) cart ⇒ bool. path_connected s = (∀ x::(real, ?'a::type) cart. IN x s ⟶ path_component s x = s)

thm PATH_COMPONENT_MONO:

∀ (s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) x::(real, ?'a::type) cart. SUBSET s t ⟶ SUBSET (path_component s x) (path_component t x)

thm PATH_COMPONENT_OF_SUBSET:

∀ (s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SUBSET s t ∧ path_component s x y ⟶ path_component t x y

thm PATH_COMPONENT_MAXIMAL:

∀ (s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) x::(real, ?'a::type) cart. IN x t ∧ path_connected t ∧ SUBSET t s ⟶ SUBSET t (path_component s x)

thm PATH_COMPONENT_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ y\ (path\_component\ s\ x) \longrightarrow path\_component\ s\ y = path\_component\ s\ x$

thm PATH_COMPONENT_PATH_IMAGE_PATHSTART:

$\forall (p::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ path\ p \land IN\ x\ (path\_image\ p) \longrightarrow path\_component\ (path\_image\ p)\ (pathstart\ p)\ x$

thm PATH_CONNECTED_PATH_IMAGE:

$\forall p::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ path\ p \longrightarrow path\_connected\ (path\_image\ p)$

thm PATH_CONNECTED_PATH_COMPONENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ path\_connected\ (path\_component\ s\ x)$

thm PATH_COMPONENT_PATH_COMPONENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ path\_component\ (path\_component\ s\ x)\ x = path\_component\ s\ x$

thm PATH_CONNECTED_LINEPATH:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ SUBSET\ (closed\_segment\ [(a, b)])\ s \longrightarrow path\_component\ s\ a\ b$

thm PATH_COMPONENT_DISJOINT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ DISJOINT\ (path\_component\ s\ a)\ (path\_component\ s\ b) = (\neg\ IN\ a\ (path\_component\ s\ b))$

thm OPEN_GENERAL_COMPONENT:

$\forall c::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool.\ (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow IN\ x\ s \land IN\ y\ s) \land (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow c\ s\ y\ x) \land (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ c\ s\ x\ y \land c\ s\ y\ z \longrightarrow c\ s\ x\ z) \land (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ SUBSET\ s\ t \land c\ s\ x\ y \longrightarrow c\ t\ x\ y) \land (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ e::real.\ IN\ y\ (ball\ (x, e)) \land SUBSET\ (ball\ (x, e))\ s \longrightarrow c\ (ball\ (x, e))\ x\ y) \longrightarrow (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (c\ s\ x))$

thm OPEN_NON_GENERAL_COMPONENT:

$\forall c::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool.\ (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)$

$y::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow IN\ x\ s \wedge IN\ y\ s) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow c\ s\ y\ x) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \wedge c\ s\ y\ z \longrightarrow c\ s\ x\ z) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ t \wedge c\ s\ x\ y \longrightarrow c\ t\ x\ y) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ y\ (ball\ (x,\ e)) \wedge SUBSET\ (ball\ (x,\ e))\ s \longrightarrow c\ (ball\ (x,\ e))\ x\ y) \longrightarrow (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (DIFF\ s\ (c\ s\ x)))$

thm GENERAL_CONNECTED_OPEN:

$\forall c::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow IN\ x\ s \wedge IN\ y\ s) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \longrightarrow c\ s\ y\ x) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ c\ s\ x\ y \wedge c\ s\ y\ z \longrightarrow c\ s\ x\ z) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SUBSET\ s\ t \wedge c\ s\ x\ y \longrightarrow c\ t\ x\ y) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ e::real.\ IN\ y\ (ball\ (x,\ e)) \wedge SUBSET\ (ball\ (x,\ e))\ s \longrightarrow c\ (ball\ (x,\ e))\ x\ y) \longrightarrow (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \wedge connected\ s \wedge IN\ x\ s \wedge IN\ y\ s \longrightarrow c\ s\ x\ y)$

thm CONVEX_IMP_PATH_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow path\_connected\ s$

thm PATH_CONNECTED_UNIV:

$path\_connected\ HOL\_Light\_Import.UNIV$

thm PATH_COMPONENT_UNIV:

$\forall x::(real,\ ?'a::type)\ cart.\ path\_component\ HOL\_Light\_Import.UNIV\ x = HOL\_Light\_Import.UNIV$

thm PATH_CONNECTED_IMP_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ s \longrightarrow connected\ s$

thm OPEN_PATH_COMPONENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (path\_component\ s\ x)$

thm OPEN_NON_PATH_COMPONENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (DIFF\ s\ (path\_component\ s\ x))$

thm CONNECTED_OPEN_PATH_CONNECTED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \land connected\ s \longrightarrow path\_connected\ s$

thm PATH_CONNECTED_EQ_CONNECTED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow path\_connected\ s = connected\ s$

thm PATH_CONNECTED_CONTINUOUS_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \land path\_connected\ s \longrightarrow path\_connected\ (IMAGE\ f\ s)$

thm HOMEOMORPHIC_PATH_CONNECTEDNESS:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \longrightarrow path\_connected\ s = path\_connected\ t$

thm PATH_CONNECTED_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ path\_connected\ s \land linear\ f \longrightarrow path\_connected\ (IMAGE\ f\ s)$

thm PATH_CONNECTED_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \land (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow path\_connected\ (IMAGE\ f\ s) = path\_connected\ s$

thm PATH_CONNECTED_EMPTY:

$path\_connected\ EMPTY$

thm PATH_CONNECTED_SING:

$\forall a::(real, ?'a::type)\ cart.\ path\_connected\ (INSERT\ a\ EMPTY)$

thm PATH_CONNECTED_UNION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ s \land path\_connected\ t \land HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow path\_connected\ (HOL\_Light\_Import.UNION\ s\ t)$

thm PATH_CONNECTED_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ s \longrightarrow path\_connected\ (IMAGE\ (vector\_add\ a)\ s)$

thm PATH_CONNECTED_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ (IMAGE\ (vector\_add\ a)\ s) = path\_connected\ s$

thm PATH_CONNECTED_PASTECART:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ s \land path\_connected\ t \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1102::(real,$

(?'b::type, ?'a::type) finite_sum) cart. ∃ (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1102 (IN x s ∧ IN y t) (pastecart x y)))

thm SEGMENT_OPEN_SUBSET_CLOSED:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. SUBSET (open_segment (a, b)) (closed_segment [(a, b)])

thm CLOSED_SEGMENT:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. HOL_Light_Import.closed (closed_segment [(a, b)])

thm SEGMENT_IMAGE_INTERVAL_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. a ≠ b ⟶ open_segment (a, b) = IMAGE (λu::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop u) a) (% (HOL_Light_Import.drop u) b)) (open_interval (vec (0::nat), vec (1::nat)))

thm SEGMENT_IMAGE_INTERVAL_conjunct0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closed_segment [(a, b)] = IMAGE (λu::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop u) a) (% (HOL_Light_Import.drop u) b)) (closed_interval [(vec (0::nat), vec (1::nat))])

thm SEGMENT_IMAGE_INTERVAL:

(∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. closed_segment [(a, b)] = IMAGE (λu::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop u) a) (% (HOL_Light_Import.drop u) b)) (closed_interval [(vec (0::nat), vec (1::nat))])) ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. a ≠ b ⟶ open_segment (a, b) = IMAGE (λu::(real, unit) cart. vector_add (% ((1::real) − HOL_Light_Import.drop u) a) (% (HOL_Light_Import.drop u) b)) (open_interval (vec (0::nat), vec (1::nat))))

thm CLOSURE_SEGMENT_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closure (open_segment (a, b)) = (if a = b then EMPTY else closed_segment [(a, b)])

thm CLOSURE_SEGMENT:

(∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closure (closed_segment [(a, b)]) = closed_segment [(a, b)]) ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closure (open_segment (a, b)) = (if a = b then EMPTY else closed_segment [(a, b)]))

thm CLOSURE_SEGMENT_conjunct0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. closure (closed_segment [(a, b)]) = closed_segment [(a, b)]

thm AFFINE_HULL_SEGMENT_conjunct1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*open_segment* (*a*, *b*)) = (*if a* = *b then EMPTY else hull affine* (*INSERT a* (*INSERT b EMPTY*)))

thm AFFINE_HULL_SEGMENT:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*closed_segment* [(*a*, *b*)]) = *hull affine* (*INSERT a* (*INSERT b EMPTY*))) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*open_segment* (*a*, *b*)) = (*if a* = *b then EMPTY else hull affine* (*INSERT a* (*INSERT b EMPTY*))))

thm SEGMENT_AS_BALL_conjunct1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *b*) = *HOL_Light_Import.INTER* (*hull affine* (*INSERT a* (*INSERT b EMPTY*))) (*ball* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) (*vector_add a b*), *vector_norm* (*vector_sub b a*) / *real_of_nat* (*2*::*nat*)))

thm SEGMENT_AS_BALL_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *closed_segment* [(*a*, *b*)] = *HOL_Light_Import.INTER* (*hull affine* (*INSERT a* (*INSERT b EMPTY*))) (*cball* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) (*vector_add a b*), *vector_norm* (*vector_sub b a*) / *real_of_nat* (*2*::*nat*)))

thm SEGMENT_AS_BALL:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *closed_segment* [(*a*, *b*)] = *HOL_Light_Import.INTER* (*hull affine* (*INSERT a* (*INSERT b EMPTY*))) (*cball* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) (*vector_add a b*), *vector_norm* (*vector_sub b a*) / *real_of_nat* (*2*::*nat*)))) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *b*) = *HOL_Light_Import.INTER* (*hull affine* (*INSERT a* (*INSERT b EMPTY*))) (*ball* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) (*vector_add a b*), *vector_norm* (*vector_sub b a*) / *real_of_nat* (*2*::*nat*))))

thm CONVEX_SEGMENT:

(∀ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *convex* (*closed_segment* [(*a*, *b*)])) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *convex* (*open_segment* (*a*, *b*)))

thm CONVEX_SEGMENT_conjunct1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *convex* (*open_segment* (*a*, *b*))

thm CONVEX_SEGMENT_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *convex* (*closed_segment* [(*a*, *b*)])

thm AFFINE_HULL_SEGMENT_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*closed_segment* [(*a*, *b*)]) = *hull affine* (*INSERT a* (*INSERT b EMPTY*))

thm RELATIVE_INTERIOR_SEGMENT_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ relative\_interior\ (open\_segment\ (a, b)) = open\_segment\ (a, b)$

thm RELATIVE_INTERIOR_SEGMENT:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ relative\_interior\ (closed\_segment\ [(a, b)]) = (if\ a = b\ then\ INSERT\ a\ EMPTY\ else\ open\_segment\ (a, b))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ relative\_interior\ (open\_segment\ (a, b)) = open\_segment\ (a, b))$

thm PATH_CONNECTED_SEGMENT:

$(\forall (a::(real, ?'b::type)\ cart)\ b::(real, ?'b::type)\ cart.\ path\_connected\ (closed\_segment\ [(a, b)])) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ path\_connected\ (open\_segment\ (a, b)))$

thm CONNECTED_SEGMENT:

$(\forall (a::(real, ?'b::type)\ cart)\ b::(real, ?'b::type)\ cart.\ connected\ (closed\_segment\ [(a, b)])) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ connected\ (open\_segment\ (a, b)))$

thm CONVEX_SEMIOPEN_SEGMENT_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ convex\ (DELETE\ (closed\_segment\ [(a, b)])\ a)$

thm CONVEX_SEMIOPEN_SEGMENT:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ convex\ (DELETE\ (closed\_segment\ [(a, b)])\ a)) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ convex\ (DELETE\ (closed\_segment\ [(a, b)])\ b))$

thm CONVEX_SEMIOPEN_SEGMENT_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ convex\ (DELETE\ (closed\_segment\ [(a, b)])\ b)$

thm PATH_CONNECTED_SEMIOPEN_SEGMENT:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ path\_connected\ (DELETE\ (closed\_segment\ [(a, b)])\ a)) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ path\_connected\ (DELETE\ (closed\_segment\ [(a, b)])\ b))$

thm CONNECTED_SEMIOPEN_SEGMENT:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ connected\ (DELETE\ (closed\_segment\ [(a, b)])\ a)) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ connected\ (DELETE\ (closed\_segment\ [(a, b)])\ b))$

thm SEGMENT_EQ_EMPTY_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ (open\_segment\ (a, b) = EMPTY) = (a = b)$

thm SEGMENT_EQ_EMPTY:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ closed\_segment\ [(a, b)] \neq EMPTY) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ (open\_segment\ (a, b) = EMPTY) = (a = b))$

thm FINITE_INTERVAL_1_conjunct1:

$\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ FINITE\ (open\_interval\ (a, b)) = (HOL\_Light\_Import.drop\ b \leq HOL\_Light\_Import.drop\ a)$

thm FINITE_SEGMENT_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ FINITE\ (closed\_segment\ [(a, b)]) = (a = b)$

thm FINITE_SEGMENT:

$(\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ FINITE\ (closed\_segment\ [(a, b)]) = (a = b)) \wedge (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ FINITE\ (open\_segment\ (a, b)) = (a = b))$

thm FINITE_SEGMENT_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ FINITE\ (open\_segment\ (a, b)) = (a = b)$

thm SEGMENT_EQ_SING_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ open\_segment\ (a, b) \neq INSERT\ c\ EMPTY$

thm SEGMENT_EQ_SING:

$(\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ (closed\_segment\ [(a, b)] = INSERT\ c\ EMPTY) = (a = c \wedge b = c)) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ open\_segment\ (a, b) \neq INSERT\ c\ EMPTY)$

thm SEGMENT_EQ_EMPTY_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ closed\_segment\ [(a, b)] \neq EMPTY$

thm SEGMENT_EQ_SING_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ (closed\_segment\ [(a, b)] = INSERT\ c\ EMPTY) = (a = c \wedge b = c)$

thm SUBSET_SEGMENT_OPEN_CLOSED:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::(real, ?'a::type)\ cart)\ d::(real, ?'a::type)\ cart.\ SUBSET\ (open\_segment\ (a, b))\ (open\_segment\ (c, d)) = (a = b \vee SUBSET\ (closed\_segment\ [(a, b)])\ (closed\_segment\ [(c, d)]))$

thm SUBSET_SEGMENT_conjunct1:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_segment\ [(a,\ b)])\ (open\_segment\ (c,$ $d)) = (IN\ a\ (open\_segment\ (c,\ d)) \wedge IN\ b\ (open\_segment\ (c,\ d)))$

thm SUBSET_SEGMENT_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_segment\ [(a,\ b)])\ (closed\_segment$ $[(c,\ d)]) = (IN\ a\ (closed\_segment\ [(c,\ d)]) \wedge IN\ b\ (closed\_segment\ [(c,\ d)]))$

thm SUBSET_SEGMENT:

$(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_segment\ [(a,\ b)])\ (closed\_segment$ $[(c,\ d)]) = (IN\ a\ (closed\_segment\ [(c,\ d)]) \wedge IN\ b\ (closed\_segment\ [(c,\ d)]))) \wedge$ $(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_segment\ [(a,\ b)])\ (open\_segment\ (c,$ $d)) = (IN\ a\ (open\_segment\ (c,\ d)) \wedge IN\ b\ (open\_segment\ (c,\ d)))) \wedge (\forall (a::(real,$ $?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,$ $?'a::type)\ cart.\ SUBSET\ (open\_segment\ (a,\ b))\ (closed\_segment\ [(c,\ d)]) =$ $(a = b \vee IN\ a\ (closed\_segment\ [(c,\ d)]) \wedge IN\ b\ (closed\_segment\ [(c,\ d)]))) \wedge$ $(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ SUBSET\ (open\_segment\ (a,\ b))\ (open\_segment\ (c,\ d))$ $= (a = b \vee IN\ a\ (closed\_segment\ [(c,\ d)]) \wedge IN\ b\ (closed\_segment\ [(c,\ d)])))$

thm RELATIVE_INTERIOR_SEGMENT_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ relative\_interior\ (closed\_segment$ $[(a,\ b)]) = (if\ a = b\ then\ INSERT\ a\ EMPTY\ else\ open\_segment\ (a,\ b))$

thm INTERIOR_SEGMENT:

$(\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ interior\ (closed\_segment$ $[(a,\ b)]) = (if\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV\ then\ EMPTY\ else$ $open\_segment\ (a,\ b))) \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.$ $interior\ (open\_segment\ (a,\ b)) = (if\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV$ $then\ EMPTY\ else\ open\_segment\ (a,\ b)))$

thm SEGMENT_EQ_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ (closed\_segment\ [(a,\ b)] = closed\_segment\ [(c,\ d)]) =$ $(INSERT\ a\ (INSERT\ b\ EMPTY) = INSERT\ c\ (INSERT\ d\ EMPTY))$

thm SEGMENT_EQ:

$(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ (closed\_segment\ [(a,\ b)] = closed\_segment\ [(c,\ d)])$ $= (INSERT\ a\ (INSERT\ b\ EMPTY) = INSERT\ c\ (INSERT\ d\ EMPTY))) \wedge$ $(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$ $d::(real,\ ?'a::type)\ cart.\ closed\_segment\ [(a,\ b)] \neq open\_segment\ (c,\ d)) \wedge$ $(\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)$

d::(*real*, *?'a*::*type*) *cart*. *open_segment* (*a*, *b*) ≠ *closed_segment* [(*c*, *d*)]) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. (*open_segment* (*a*, *b*) = *open_segment* (*c*, *d*)) = (*a* = *b* ∧ *c* = *d* ∨ *INSERT a* (*INSERT b EMPTY*) = *INSERT c* (*INSERT d EMPTY*)))

thm COMPACT_SEGMENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *compact* (*closed_segment* [(*a*, *b*)])

thm BOUNDED_SEGMENT:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *bounded* (*closed_segment* [(*a*, *b*)])) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *bounded* (*open_segment* (*a*, *b*)))

thm COLLINEAR_SEGMENT:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *collinear* (*closed_segment* [(*a*, *b*)])) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *collinear* (*open_segment* (*a*, *b*)))

thm UNION_SEGMENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *IN b* (*closed_segment* [(*a*, *c*)]) ⟶ *HOL_Light_Import.UNION* (*closed_segment* [(*a*, *b*)]) (*closed_segment* [(*b*, *c*)]) = *closed_segment* [(*a*, *c*)]

thm INTER_SEGMENT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *IN b* (*closed_segment* [(*a*, *c*)]) ∨ ¬ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) ⟶ *HOL_Light_Import.INTER* (*closed_segment* [(*a*, *b*)]) (*closed_segment* [(*b*, *c*)]) = *INSERT b EMPTY*

thm CARD_EQ_EUCLIDEAN:

=_*c* *HOL_Light_Import.UNIV HOL_Light_Import.UNIV*

thm UNCOUNTABLE_EUCLIDEAN:

¬ *COUNTABLE HOL_Light_Import.UNIV*

thm CARD_EQ_INTERVAL:

(∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) ≠ *EMPTY* ⟶ =_*c* (*closed_interval* [(*a*, *b*)]) *HOL_Light_Import.UNIV*) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) ≠ *EMPTY* ⟶ =_*c* (*open_interval* (*a*, *b*)) *HOL_Light_Import.UNIV*)

thm CARD_EQ_INTERVAL_conjunct1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) ≠ *EMPTY* ⟶ =_*c* (*open_interval* (*a*, *b*)) *HOL_Light_Import.UNIV*

thm CARD_EQ_INTERVAL_conjunct0:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\longrightarrow$ =_*c* (*closed_interval* [(*a*, *b*)]) *HOL_Light_Import.UNIV*

thm UNCOUNTABLE_INTERVAL:

($\forall$ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\longrightarrow$ $\neg$ *COUNTABLE* (*closed_interval* [(*a*, *b*)])) $\wedge$ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *open_interval* (*a*, *b*) $\neq$ *EMPTY* $\longrightarrow$ $\neg$ *COUNTABLE* (*open_interval* (*a*, *b*)))

thm COUNTABLE_OPEN_INTERVAL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *COUNTABLE* (*open_interval* (*a*, *b*)) = (*open_interval* (*a*, *b*) = *EMPTY*)

thm CARD_EQ_OPEN:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ =_*c s HOL_Light_Import.UNIV*

thm CARD_EQ_BALL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) < *r* $\longrightarrow$ =_*c* (*ball* (*a*, *r*)) *HOL_Light_Import.UNIV*

thm CARD_EQ_CBALL:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) < *r* $\longrightarrow$ =_*c* (*cball* (*a*, *r*)) *HOL_Light_Import.UNIV*

thm CARD_EQ_SEGMENT_conjunct1:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ =_*c* (*open_segment* (*a*, *b*)) *HOL_Light_Import.UNIV*

thm CARD_EQ_SEGMENT_conjunct0:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ =_*c* (*closed_segment* [(*a*, *b*)]) *HOL_Light_Import.UNIV*

thm CARD_EQ_SEGMENT:

($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ =_*c* (*closed_segment* [(*a*, *b*)]) *HOL_Light_Import.UNIV*) $\wedge$ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ =_*c* (*open_segment* (*a*, *b*)) *HOL_Light_Import.UNIV*)

thm UNCOUNTABLE_SEGMENT:

($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ $\neg$ *COUNTABLE* (*closed_segment* [(*a*, *b*)])) $\wedge$ ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *b* $\longrightarrow$ $\neg$ *COUNTABLE* (*open_segment* (*a*, *b*)))

thm CARD_EQ_CONVEX:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *convex s* $\wedge$ *IN a s* $\wedge$ *IN b s* $\wedge$ *a* $\neq$ *b* $\longrightarrow$ =_*c s HOL_Light_Import.UNIV*

thm UNCOUNTABLE_CONVEX:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ convex\ s \wedge IN\ a\ s \wedge IN\ b\ s \wedge a \neq b \longrightarrow \neg\ COUNTABLE\ s$

thm INTERVAL_NE_EMPTY_1_conjunct1:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (open\_interval\ (a,\ b) \neq EMPTY) = (HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b)$

thm INTERVAL_NE_EMPTY_1_conjunct0:

$\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (closed\_interval\ [(a,\ b)] \neq EMPTY) = (HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b)$

thm Float.REAL_RINV_2:

$real\_of\_nat\ (2::nat) * inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) = (1::real)$

thm HOMEOMORPHIC_MONOTONE_IMAGE_INTERVAL:

$\forall f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \wedge (\forall y::(real,\ ?'a::type)\ cart.\ connected\ (GSPEC\ (\lambda GEN\%PVAR\%1121::(real,\ unit)\ cart.\ \exists x::(real,\ unit)\ cart.\ SETSPEC\ GEN\%PVAR\%1121\ (IN\ x\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \wedge f\ x = y)\ x))) \wedge f\ (vec\ (1::nat)) \neq f\ (vec\ (0::nat)) \longrightarrow homeomorphic\ (IMAGE\ f\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]))\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))])$

thm CONNECTED_SEGMENT_conjunct1:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ connected\ (open\_segment\ (a,\ b))$

thm CONNECTED_SEGMENT_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ connected\ (closed\_segment\ [(a,\ b)])$

thm PATH_CONTAINS_ARC:

$\forall (p::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ path\ p \wedge pathstart\ p = a \wedge pathfinish\ p = b \wedge a \neq b \longrightarrow (\exists q::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ arc\ q \wedge SUBSET\ (path\_image\ q)\ (path\_image\ p) \wedge pathstart\ q = a \wedge pathfinish\ q = b)$

thm PATH_CONNECTED_ARCWISE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ path\_connected\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow (\exists g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ arc\ g \wedge SUBSET\ (path\_image\ g)\ s \wedge pathstart\ g = x \wedge pathfinish\ g = y))$

thm ARC_CONNECTED_TRANS:

$\forall (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ h::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ arc\ g \wedge arc\ h \wedge pathfinish\ g = pathstart\ h \wedge pathstart\ g \neq$

pathfinish h ⟶ (∃ *i*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc i* ∧ *SUBSET* (*path_image i*) (*HOL_Light_Import.UNION* (*path_image g*) (*path_image h*)) ∧ *pathstart i* = *pathstart g* ∧ *pathfinish i* = *pathfinish h*)

thm CONNECTED_OPEN_ARC_CONNECTED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *connected s* ⟶ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *IN y s* ⟶ *x* = *y* ∨ (∃ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc g* ∧ *SUBSET* (*path_image g*) *s* ∧ *pathstart g* = *x* ∧ *pathfinish g* = *y*))

thm OPEN_CONNECTED_COMPONENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.open s* ⟶ *HOL_Light_Import.open* (*connected_component s x*)

thm IN_CLOSURE_CONNECTED_COMPONENT:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x* (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ∧ *HOL_Light_Import.open ?s* ⟶ *IN x* (*closure* (*connected_component ?s y*)) = *IN x* (*connected_component ?s y*)

thm PATH_COMPONENT_SUBSET_CONNECTED_COMPONENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*path_component s x*) (*connected_component s x*)

thm OPEN_PATH_CONNECTED_COMPONENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.open s* ⟶ *path_component s x* = *connected_component s x*

thm OPEN_COMPONENTS:

∀ (*u*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open u* ∧ *IN s* (*components u*) ⟶ *HOL_Light_Import.open s*

thm CONTINUOUS_ON_COMPONENTS:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ (∀ *c*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *IN c* (*components s*) ⟶ *continuous_on f c*) ⟶ *continuous_on f s*

thm CONTINUOUS_ON_COMPONENTS_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ⟶ *continuous_on f s* = (∀ *c*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *IN c* (*components s*) ⟶ *continuous_on f c*)

thm CLOSED_UNION_COMPLEMENT_COMPONENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *c*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed s* ∧ *IN c* (*components* (*DIFF HOL_Light_Import.UNIV s*)) ⟶ *HOL_Light_Import.closed* (*HOL_Light_Import.UNION s c*)

thm COUNTABLE_COMPONENTS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import*.*open s* ⟶ *COUNTABLE* (*components s*)

thm FRONTIER_MINIMAL_SEPARATING_CLOSED:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *c*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import*.*closed s* ∧ ¬ *connected* (*DIFF HOL_Light_Import*.*UNIV s*) ∧ (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import*.*closed t* ∧ *PSUBSET t s* ⟶ *connected* (*DIFF HOL_Light_Import*.*UNIV t*)) ∧ *IN c* (*components* (*DIFF HOL_Light_Import*.*UNIV s*)) ⟶ *frontier c* = *s*

thm NORM_SEGMENT_LOWERBOUND:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *d*::*real*. (*0*::*real*) < *r* ∧ *vector_norm a* = *r* ∧ *vector_norm b* = *r* ∧ *IN x* (*closed_segment* [(*a*, *b*)]) ∧ *dot a b* = *d* ∗ r^2 ⟶ *sqrt* (((*1*::*real*) − |*d*|) / *real_of_nat* (*2*::*nat*)) ∗ *r* ≤ *vector_norm x*

thm NORM_SEGMENT_ORTHOGONAL_LOWERBOUND:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. *r* ≤ *vector_norm a* ∧ *r* ≤ *vector_norm b* ∧ *orthogonal a b* ∧ *IN x* (*closed_segment* [(*a*, *b*)]) ⟶ *r* / *real_of_nat* (*2*::*nat*) ≤ *vector_norm x*

thm PATH_CONNECTED_CONVEX_DIFF_CARD_LT:

∀ (*u*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex u* ∧ ¬ *collinear u* ∧ *<_c s HOL_Light_Import*.*UNIV* ⟶ *path_connected* (*DIFF u s*)

thm PATH_CONNECTED_COMPLEMENT_CARD_LT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *<_c s HOL_Light_Import*.*UNIV* ⟶ *path_connected* (*DIFF HOL_Light_Import*.*UNIV s*)

thm PATH_CONNECTED_OPEN_DIFF_CARD_LT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *HOL_Light_Import*.*open s* ∧ *connected s* ∧ *<_c t HOL_Light_Import*.*UNIV* ⟶ *path_connected* (*DIFF s t*)

thm CONNECTED_OPEN_DIFF_CARD_LT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *HOL_Light_Import*.*open s* ∧ *connected s* ∧ *<_c t HOL_Light_Import*.*UNIV* ⟶ *connected* (*DIFF s t*)

thm PATH_CONNECTED_OPEN_DIFF_COUNTABLE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import*.*UNIV* ∧ *HOL_Light_Import*.*open s* ∧ *connected s* ∧ *COUNTABLE t* ⟶ *path_connected* (*DIFF s t*)

thm CONNECTED_OPEN_DIFF_COUNTABLE:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ HOL_Light_Import.open s ∧ connected s ∧ COUNTABLE t ⟶ connected (DIFF s t)

thm PATH_CONNECTED_OPEN_DELETE:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ HOL_Light_Import.open s ∧ connected s ⟶ path_connected (DELETE s a)

thm CONNECTED_OPEN_DELETE:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ HOL_Light_Import.open s ∧ connected s ⟶ connected (DELETE s a)

thm PATH_CONNECTED_PUNCTURED_UNIVERSE:

∀ a::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ path_connected (DIFF HOL_Light_Import.UNIV (INSERT a EMPTY))

thm CONNECTED_PUNCTURED_UNIVERSE:

∀ a::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ connected (DIFF HOL_Light_Import.UNIV (INSERT a EMPTY))

thm PATH_CONNECTED_PUNCTURED_BALL:

∀ (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ path_connected (DELETE (ball (a, r)) a)

thm CONNECTED_PUNCTURED_BALL:

∀ (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ connected (DELETE (ball (a, r)) a)

thm PATH_CONNECTED_SPHERE:

∀ (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ path_connected (GSPEC (λGEN%PVAR%1132::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1132 (vector_norm (vector_sub x a) = r) x))

thm CONNECTED_SPHERE:

∀ (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ⟶ connected (GSPEC (λGEN%PVAR%1133::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1133 (vector_norm (vector_sub x a) = r) x))

thm IN_IMAGE_LIFT_DROP_conjunct1:

∀ (x::real) s::(real, unit) cart ⇒ bool. IN x (IMAGE HOL_Light_Import.drop s) = IN (lift x) s

thm IN_IMAGE_LIFT_DROP_conjunct0:

$\forall (x::(real, unit)\ cart)\ s::real \Rightarrow bool.\ IN\ x\ (IMAGE\ lift\ s) = IN\ (HOL\_Light\_Import.drop\ x)\ s$

thm PATH_CONNECTED_ANNULUS_conjunct3:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1143::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1143\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm PATH_CONNECTED_ANNULUS_conjunct2:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1142::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1142\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm PATH_CONNECTED_ANNULUS_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1141::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1141\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) \leq r2)\ x))$

thm PATH_CONNECTED_ANNULUS_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1140::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1140\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm PATH_CONNECTED_ANNULUS:

$(\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1140::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1140\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1141::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1141\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) \leq r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1142::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1142\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow path\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1143::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1143\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x)))$

thm CONNECTED_ANNULUS_conjunct3:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1147::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1147\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm CONNECTED_ANNULUS_conjunct2:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1146::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1146\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm CONNECTED_ANNULUS_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1145::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1145\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) \leq r2)\ x))$

thm CONNECTED_ANNULUS_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1144::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1144\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))$

thm CONNECTED_ANNULUS:

$(\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1144::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1144\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1145::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1145\ (r1 < vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) \leq r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1146::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1146\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x))) \wedge (\forall (a::(real, ?'a::type)\ cart)\ (r1::real)\ r2::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow connected\ (GSPEC\ (\lambda GEN\%PVAR\%1147::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1147\ (r1 \leq vector\_norm\ (vector\_sub\ x\ a) \wedge vector\_norm\ (vector\_sub\ x\ a) < r2)\ x)))$

thm PATH_CONNECTED_COMPLEMENT_BOUNDED_CONVEX:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \wedge bounded\ s \wedge convex\ s \longrightarrow path\_connected\ (DIFF\ HOL\_Light\_Import.UNIV\ s)$

thm CONNECTED_COMPLEMENT_BOUNDED_CONVEX:

∀ *s::(real, ?'a::type) cart ⇒ bool. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ bounded s ∧ convex s ⟶ connected (DIFF HOL_Light_Import.UNIV s)*

thm CONNECTED_DIFF_BALL:

∀ *(s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ connected s ∧ SUBSET (cball (a, r)) s ⟶ connected (DIFF s (ball (a, r)))*

thm PATH_CONNECTED_DIFF_BALL:

∀ *(s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ path_connected s ∧ SUBSET (cball (a, r)) s ⟶ path_connected (DIFF s (ball (a, r)))*

thm CONNECTED_OPEN_DIFF_CBALL:

∀ *(s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) r::real. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ HOL_Light_Import.open s ∧ connected s ∧ SUBSET (cball (a, r)) s ⟶ connected (DIFF s (cball (a, r)))*

thm COBOUNDED_UNBOUNDED_COMPONENT:

∀ *s::(real, ?'a::type) cart ⇒ bool. bounded (DIFF HOL_Light_Import.UNIV s) ⟶ (∃ x::(real, ?'a::type) cart. IN x s ∧ ¬ bounded (connected_component s x))*

thm COBOUNDED_UNIQUE_UNBOUNDED_COMPONENT:

∀ *(s::(real, ?'a::type) cart ⇒ bool) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ bounded (DIFF HOL_Light_Import.UNIV s) ∧ ¬ bounded (connected_component s x) ∧ ¬ bounded (connected_component s y) ⟶ connected_component s x = connected_component s y*

thm COBOUNDED_UNIQUE_UNBOUNDED_COMPONENTS:

∀ *(s::(real, ?'a::type) cart ⇒ bool) (c::(real, ?'a::type) cart ⇒ bool) c'::(real, ?'a::type) cart ⇒ bool. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ bounded (DIFF HOL_Light_Import.UNIV s) ∧ IN c (components s) ∧ ¬ bounded c ∧ IN c' (components s) ∧ ¬ bounded c' ⟶ c' = c*

thm DEF_inside:

inside = (λ_409808::(real, ?'a::type) cart ⇒ bool. GSPEC (λGEN%PVAR%1152::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1152 (¬ IN x _409808 ∧ bounded (connected_component (DIFF HOL_Light_Import.UNIV _409808) x)) x))

thm inside:

∀ *s::(real, ?'a::type) cart ⇒ bool. inside s = GSPEC (λGEN%PVAR%1152::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1152 (¬ IN x s ∧ bounded (connected_component (DIFF HOL_Light_Import.UNIV s) x)) x)*

thm DEF_outside:

$outside = (\lambda\_409813::(real, ?'a::type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1153::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1153\ (\neg IN\ x\ \_409813 \wedge \neg\ bounded\ (connected\_component\ (DIFF\ HOL\_Light\_Import.UNIV\ \_409813)\ x))\ x))$

thm outside:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ outside\ s = GSPEC\ (\lambda GEN\%PVAR\%1153::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1153\ (\neg IN\ x\ s \wedge \neg\ bounded\ (connected\_component\ (DIFF\ HOL\_Light\_Import.UNIV\ s)\ x))\ x)$

thm INSIDE_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ inside\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (inside\ s)$

thm OUTSIDE_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ outside\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (outside\ s)$

thm INSIDE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow inside\ (IMAGE\ f\ s) = IMAGE\ f\ (inside\ s)$

thm OUTSIDE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow outside\ (IMAGE\ f\ s) = IMAGE\ f\ (outside\ s)$

thm OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ outside\ s = GSPEC\ (\lambda GEN\%PVAR\%1154::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1154\ (\neg\ bounded\ (connected\_component\ (DIFF\ HOL\_Light\_Import.UNIV\ s)\ x))\ x)$

thm INSIDE_NO_OVERLAP:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (inside\ s)\ s = EMPTY$

thm OUTSIDE_NO_OVERLAP:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (outside\ s)\ s = EMPTY$

thm INSIDE_INTER_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (inside\ s)\ (outside\ s) = EMPTY$

thm INSIDE_UNION_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (inside\ s)\ (outside\ s) = DIFF\ HOL\_Light\_Import.UNIV\ s$

thm INSIDE_EQ_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (inside\ s = outside\ s) = (s = HOL\_Light\_Import.UNIV)$

thm INSIDE_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ inside\ s = DIFF\ HOL\_Light\_Import.UNIV\ (HOL\_Light\_Import.UNION\ s\ (outside\ s))$

thm OUTSIDE_INSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ outside\ s = DIFF\ HOL\_Light\_Import.UNIV\ (HOL\_Light\_Import.UNION\ s\ (inside\ s))$

thm UNION_WITH_INSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.UNION\ s\ (inside\ s) = DIFF\ HOL\_Light\_Import.UNIV\ (outside\ s)$

thm UNION_WITH_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.UNION\ s\ (outside\ s) = DIFF\ HOL\_Light\_Import.UNIV\ (inside\ s)$

thm OUTSIDE_MONO:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (outside\ t)\ (outside\ s)$

thm INSIDE_MONO:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow SUBSET\ (DIFF\ (inside\ s)\ t)\ (inside\ t)$

thm COBOUNDED_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow bounded\ (DIFF\ HOL\_Light\_Import.UNIV\ (outside\ s))$

thm UNBOUNDED_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow \neg\ bounded\ (outside\ s)$

thm BOUNDED_INSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow bounded\ (inside\ s)$

thm CONNECTED_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ (2::nat) \le dimindex\ HOL\_Light\_Import.UNIV \wedge bounded\ s \longrightarrow connected\ (outside\ s)$

thm OUTSIDE_CONNECTED_COMPONENT_LT:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *outside s* = *GSPEC* (λ*GEN%PVAR%1157*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1157* ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B <$ *vector_norm y* $\wedge$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*) *x*)

thm OUTSIDE_CONNECTED_COMPONENT_LE:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *outside s* = *GSPEC* (λ*GEN%PVAR%1158*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1158* ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B \leq$ *vector_norm y* $\wedge$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*) *x*)

thm NOT_OUTSIDE_CONNECTED_COMPONENT_LT:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*2*::*nat*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\longrightarrow$ *DIFF HOL_Light_Import.UNIV* (*outside s*) = *GSPEC* (λ*GEN%PVAR%1159*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1159* ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B <$ *vector_norm y* $\wedge \neg$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*) *x*)

thm NOT_OUTSIDE_CONNECTED_COMPONENT_LE:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*2*::*nat*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\longrightarrow$ *DIFF HOL_Light_Import.UNIV* (*outside s*) = *GSPEC* (λ*GEN%PVAR%1160*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1160* ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B \leq$ *vector_norm y* $\wedge \neg$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*) *x*)

thm INSIDE_CONNECTED_COMPONENT_LT:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*2*::*nat*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\longrightarrow$ *inside s* = *GSPEC* (λ*GEN%PVAR%1161*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1161* ($\neg$ *IN x s* $\wedge$ ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B <$ *vector_norm y* $\wedge \neg$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*)) *x*)

thm INSIDE_CONNECTED_COMPONENT_LE:

$\forall\, s$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. (*2*::*nat*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\longrightarrow$ *inside s* = *GSPEC* (λ*GEN%PVAR%1162*::(*real*, *?'a*::*type*) *cart*. $\exists\, x$::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1162* ($\neg$ *IN x s* $\wedge$ ($\forall\, B$::*real*. $\exists\, y$::(*real*, *?'a*::*type*) *cart*. $B \leq$ *vector_norm y* $\wedge \neg$ *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y*)) *x*)

thm OUTSIDE_UNION_OUTSIDE_UNION:

$\forall$ (*c*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*c1*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *c2*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *HOL_Light_Import.INTER c* (*outside* (*HOL_Light_Import.UNION c1 c2*)) = *EMPTY* $\longrightarrow$ *SUBSET* (*outside* (*HOL_Light_Import.UNION c1 c2*)) (*outside* (*HOL_Light_Import.UNION c1 c*))

thm INSIDE_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *connected u* $\wedge$ $\neg$ *bounded u* $\wedge$ *HOL_Light_Import.UNION t u* = *DIFF HOL_Light_Import.UNIV s* $\longrightarrow$ *SUBSET* (*inside s*) *t*

thm INSIDE_UNIQUE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *connected t* $\wedge$ *bounded t* $\wedge$ *connected u* $\wedge$ $\neg$ *bounded u* $\wedge$ $\neg$ *connected* (*DIFF HOL_Light_Import.UNIV s*) $\wedge$ *HOL_Light_Import.UNION t u* = *DIFF HOL_Light_Import.UNIV s* $\longrightarrow$ *inside s* = *t*

thm INSIDE_OUTSIDE_UNIQUE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *connected t* $\wedge$ *bounded t* $\wedge$ *connected u* $\wedge$ $\neg$ *bounded u* $\wedge$ $\neg$ *connected* (*DIFF HOL_Light_Import.UNIV s*) $\wedge$ *HOL_Light_Import.UNION t u* = *DIFF HOL_Light_Import.UNIV s* $\longrightarrow$ *inside s* = *t* $\wedge$ *outside s* = *u*

thm INTERIOR_INSIDE_FRONTIER:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ *SUBSET* (*interior s*) (*inside* (*frontier s*))

thm INSIDE_EMPTY:

inside EMPTY = *EMPTY*

thm OUTSIDE_EMPTY:

outside EMPTY = *HOL_Light_Import.UNIV*

thm INSIDE_SAME_COMPONENT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y* $\wedge$ *IN x* (*inside s*) $\longrightarrow$ *IN y* (*inside s*)

thm OUTSIDE_SAME_COMPONENT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *connected_component* (*DIFF HOL_Light_Import.UNIV s*) *x y* $\wedge$ *IN x* (*outside s*) $\longrightarrow$ *IN y* (*outside s*)

thm OUTSIDE_CONVEX:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex s* $\longrightarrow$ *outside s* = *DIFF HOL_Light_Import.UNIV s*

thm INSIDE_CONVEX:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex s* $\longrightarrow$ *inside s* = *EMPTY*

thm OUTSIDE_SUBSET_CONVEX:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ t \wedge SUBSET\ s\ t \longrightarrow SUBSET\ (DIFF\ HOL\_Light\_Import.UNIV\ t)\ (outside\ s)$

thm OUTSIDE_FRONTIER_MISSES_CLOSURE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow SUBSET\ (outside\ (frontier\ s))\ (DIFF\ HOL\_Light\_Import.UNIV\ (closure\ s))$

thm OUTSIDE_FRONTIER_EQ_COMPLEMENT_CLOSURE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge convex\ s \longrightarrow outside\ (frontier\ s) = DIFF\ HOL\_Light\_Import.UNIV\ (closure\ s)$

thm INSIDE_FRONTIER_EQ_INTERIOR:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge convex\ s \longrightarrow inside\ (frontier\ s) = interior\ s$

thm OPEN_INSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.open\ (inside\ s)$

thm OPEN_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.open\ (outside\ s)$

thm CLOSURE_INSIDE_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow SUBSET\ (closure\ (inside\ s))\ (HOL\_Light\_Import.UNION\ s\ (inside\ s))$

thm FRONTIER_INSIDE_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow SUBSET\ (frontier\ (inside\ s))\ s$

thm CLOSURE_OUTSIDE_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow SUBSET\ (closure\ (outside\ s))\ (HOL\_Light\_Import.UNION\ s\ (outside\ s))$

thm FRONTIER_OUTSIDE_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow SUBSET\ (frontier\ (outside\ s))\ s$

thm INSIDE_COMPLEMENT_UNBOUNDED_CONNECTED_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ (DIFF\ HOL\_Light\_Import.UNIV\ s) \wedge \neg\ bounded\ (DIFF\ HOL\_Light\_Import.UNIV\ s) \longrightarrow inside\ s = EMPTY$

thm INSIDE_BOUNDED_COMPLEMENT_CONNECTED_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ (DIFF\ HOL\_Light\_Import.UNIV\ s) \wedge bounded\ s \longrightarrow inside\ s = EMPTY$

thm INSIDE_INSIDE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ (inside\ t) \longrightarrow SUBSET\ (DIFF\ (inside\ s)\ t)\ (inside\ t)$

thm INSIDE_INSIDE_SUBSET:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (inside\ (inside\ s))\ s$

thm INSIDE_OUTSIDE_INTERSECT_CONNECTED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ connected\ t \wedge HOL\_Light\_Import.INTER\ (inside\ s)\ t \neq EMPTY \wedge HOL\_Light\_Import.INTER\ (outside\ s)\ t \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER\ s\ t \neq EMPTY$

thm OUTSIDE_BOUNDED_NONEMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow outside\ s \neq EMPTY$

thm OUTSIDE_COMPACT_IN_OPEN:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge HOL\_Light\_Import.open\ t \wedge SUBSET\ s\ t \wedge t \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER\ (outside\ s)\ t \neq EMPTY$

thm INSIDE_INSIDE_COMPACT_CONNECTED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge compact\ t \wedge SUBSET\ s\ (inside\ t) \wedge connected\ t \longrightarrow SUBSET\ (inside\ s)\ (inside\ t)$

thm CONNECTED_WITH_INSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge connected\ s \longrightarrow connected\ (HOL\_Light\_Import.UNION\ s\ (inside\ s))$

thm CONNECTED_WITH_OUTSIDE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge connected\ s \longrightarrow connected\ (HOL\_Light\_Import.UNION\ s\ (outside\ s))$

thm INSIDE_INSIDE_EQ_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge connected\ s \longrightarrow inside\ (inside\ s) = EMPTY$

thm INSIDE_IN_COMPONENTS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ (inside\ s)\ (components\ (DIFF\ HOL\_Light\_Import.UNIV\ s)) = (connected\ (inside\ s) \wedge inside\ s \neq EMPTY)$

thm OUTSIDE_IN_COMPONENTS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ (outside\ s)\ (components\ (DIFF\ HOL\_Light\_Import.UNIV\ s)) = (connected\ (outside\ s) \wedge outside\ s \neq EMPTY)$

thm BOUNDED_UNIQUE_OUTSIDE:

∀ (c::(real, ?'a::type) cart ⇒ bool) s::(real, ?'a::type) cart ⇒ bool. (2::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ bounded s ⟶ (IN c (components (DIFF HOL_Light_Import.UNIV s)) ∧ ¬ bounded c) = (c = outside s)

thm DEF_homotopic_with:

homotopic_with = (λ(_411733::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (_411734::((real, ?'b::type) cart ⇒ bool) × ((real, ?'a::type) cart ⇒ bool)) (_411735::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) _411736::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. ∃ h::(real, (unit, ?'b::type) finite_sum) cart ⇒ (real, ?'a::type) cart. continuous_on h (GSPEC (λGEN%PVAR%1163::(real, (unit, ?'b::type) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1163 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x (fst _411734)) (pastecart t x))) ∧ SUBSET (IMAGE h (GSPEC (λGEN%PVAR%1164::(real, (unit, ?'b::type) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1164 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x (fst _411734)) (pastecart t x)))) (snd _411734) ∧ (∀ x::(real, ?'b::type) cart. h (pastecart (vec (0::nat)) x) = _411735 x) ∧ (∀ x::(real, ?'b::type) cart. h (pastecart (vec (1::nat)) x) = _411736 x) ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ _411733 (λx::(real, ?'b::type) cart. h (pastecart t x))))

thm homotopic_with:

∀ (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (p::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (q::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool. homotopic_with P (X, Y) p q = (∃ h::(real, (unit, ?'b::type) finite_sum) cart ⇒ (real, ?'a::type) cart. continuous_on h (GSPEC (λGEN%PVAR%1163::(real, (unit, ?'b::type) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1163 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x X) (pastecart t x))) ∧ SUBSET (IMAGE h (GSPEC (λGEN%PVAR%1164::(real, (unit, ?'b::type) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1164 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x X) (pastecart t x)))) Y ∧ (∀ x::(real, ?'b::type) cart. h (pastecart (vec (0::nat)) x) = p x) ∧ (∀ x::(real, ?'b::type) cart. h (pastecart (vec (1::nat)) x) = q x) ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ P (λx::(real, ?'b::type) cart. h (pastecart t x))))

thm HOMOTOPIC_WITH:

(∀ (h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) k::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ x::(real, ?'b::type) cart. IN x (?X::(real, ?'b::type) cart ⇒ bool) ⟶ h x = k x) ⟶ (?P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) h = ?P k) ⟶ homotopic_with ?P (?X, ?Y::(real, ?'a::type) cart ⇒ bool) (?p::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (?q::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) = (∃ h::(real, (unit, ?'b::type) finite_sum) cart ⇒ (real, ?'a::type) cart. continuous_on h (GSPEC (λGEN%PVAR%1165::(real, (unit, ?'b::type) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, ?'b::type) *cart. SETSPEC GEN%PVAR%1165 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x ?X) (pastecart t x))) ∧ SUBSET (IMAGE h (GSPEC (λGEN%PVAR%1166::(real, (unit, ?'b::type) finite_sum) cart. ∃(t::(real, unit) cart) x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1166 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x ?X) (pastecart t x)))) ?Y ∧ (∀x::(real, ?'b::type) cart. IN x ?X ⟶ h (pastecart (vec (0::nat)) x) = ?p x) ∧ (∀x::(real, ?'b::type) cart. IN x ?X ⟶ h (pastecart (vec (1::nat)) x) = ?q x) ∧ (∀t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ ?P (λx::(real, ?'b::type) cart. h (pastecart t x))))*

thm HOMOTOPIC_WITH_EQ:

∀(P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f'::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) g'::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_with P (X, Y) f g ∧ (∀x::(real, ?'b::type) cart. IN x X ⟶ f' x = f x ∧ g' x = g x) ∧ (∀(h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) k::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀x::(real, ?'b::type) cart. IN x X ⟶ h x = k x) ⟶ P h = P k) ⟶ homotopic_with P (X, Y) f' g'

thm HOMOTOPIC_WITH_IMP_PROPERTY:

∀(P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_with P (X, Y) f g ⟶ P f ∧ P g

thm HOMOTOPIC_WITH_IMP_CONTINUOUS:

∀(P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_with P (X, Y) f g ⟶ continuous_on f X ∧ continuous_on g X

thm HOMOTOPIC_WITH_IMP_SUBSET:

∀(P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_with P (X, Y) f g ⟶ SUBSET (IMAGE f X) Y ∧ SUBSET (IMAGE g X) Y

thm HOMOTOPIC_WITH_MONO:

∀(P::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (Q::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (X::(real, ?'b::type) cart ⇒ bool) (Y::(real, ?'a::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_with P (X, Y) f g ∧ (∀h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on h X ∧ SUBSET (IMAGE h X) Y ∧ P h ⟶ Q h) ⟶ homotopic_with Q (X, Y) f g

thm HOMOTOPIC_WITH_SUBSET_LEFT:

$\forall (P::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (X::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (Y::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (Z::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ homotopic\_with\ P\ (X,\ Y)\ f\ g \wedge SUBSET\ Z\ X \longrightarrow homotopic\_with\ P\ (Z,\ Y)\ f\ g$

thm HOMOTOPIC_WITH_SUBSET_RIGHT:

$\forall (P::((real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart) \Rightarrow bool)\ (X::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (Y::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (Z::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ h::?'a::type.\ homotopic\_with\ P\ (X,\ Y)\ f\ g \wedge SUBSET\ Y\ Z \longrightarrow homotopic\_with\ P\ (X,\ Z)\ f\ g$

thm HOMOTOPIC_WITH_COMPOSE_CONTINUOUS_RIGHT:

$\forall (p::((real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart) \Rightarrow bool)\ (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (W::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (X::(real,\ ?'a::type)\ cart \Rightarrow bool)\ Y::(real,\ ?'b::type)\ cart \Rightarrow bool.\ homotopic\_with\ (\lambda f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ p\ (f \circ h))\ (X,\ Y)\ f\ g \wedge continuous\_on\ h\ W \wedge SUBSET\ (IMAGE\ h\ W)\ X \longrightarrow homotopic\_with\ p\ (W,\ Y)\ (f \circ h)\ (g \circ h)$

thm HOMOTOPIC_COMPOSE_CONTINUOUS_RIGHT:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'c::type)\ cart)\ (W::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (X::(real,\ ?'c::type)\ cart \Rightarrow bool)\ Y::(real,\ ?'b::type)\ cart \Rightarrow bool.\ homotopic\_with\ (\lambda f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ True)\ (X,\ Y)\ f\ g \wedge continuous\_on\ h\ W \wedge SUBSET\ (IMAGE\ h\ W)\ X \longrightarrow homotopic\_with\ (\lambda f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ True)\ (W,\ Y)\ (f \circ h)\ (g \circ h)$

thm HOMOTOPIC_WITH_COMPOSE_CONTINUOUS_LEFT:

$\forall (p::((real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart) \Rightarrow bool)\ (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (X::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (Y::(real,\ ?'a::type)\ cart \Rightarrow bool)\ Z::(real,\ ?'b::type)\ cart \Rightarrow bool.\ homotopic\_with\ (\lambda f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ p\ (h \circ f))\ (X,\ Y)\ f\ g \wedge continuous\_on\ h\ Y \wedge SUBSET\ (IMAGE\ h\ Y)\ Z \longrightarrow homotopic\_with\ p\ (X,\ Z)\ (h \circ f)\ (h \circ g)$

thm HOMOTOPIC_COMPOSE_CONTINUOUS_LEFT:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (X::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (Y::(real,\ ?'b::type)\ cart \Rightarrow bool)\ Z::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homotopic\_with\ (\lambda f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart.\ True)\ (X,\ Y)\ f\ g \wedge continuous\_on\ h\ Y \wedge SUBSET\ (IMAGE\ h$

Y) *Z* ⟶ *homotopic_with* (λ*f*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *True*) (*X*, *Z*) (*h* ∘ *f*) (*h* ∘ *g*)

thm HOMOTOPIC_WITH_REFL:

∀ (*P*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*X*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_with* *P* (*X*, *Y*) *f f* = (*continuous_on f X* ∧ *SUBSET* (*IMAGE f X*) *Y* ∧ *P f*)

thm HOMOTOPIC_WITH_SYM:

∀ (*P*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*X*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_with* *P* (*X*, *Y*) *f g* = *homotopic_with* *P* (*X*, *Y*) *g f*

thm HOMOTOPIC_WITH_TRANS:

∀ (*P*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*X*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_with* *P* (*X*, *Y*) *f g* ∧ *homotopic_with* *P* (*X*, *Y*) *g h* ⟶ *homotopic_with* *P* (*X*, *Y*) *f h*

thm DEF_homotopic_paths:

homotopic_paths = (λ(*_414161*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *_414162*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_with* (λ*r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathstart r* = *pathstart _414162* ∧ *pathfinish r* = *pathfinish _414162*) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))], *_414161*) *_414162*)

thm homotopic_paths:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* = *homotopic_with* (λ*r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathstart r* = *pathstart p* ∧ *pathfinish r* = *pathfinish p*) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))], *s*) *p q*

thm HOMOTOPIC_PATHS_IMP_PATHSTART:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ⟶ *pathstart p* = *pathstart q*

thm HOMOTOPIC_PATHS_IMP_PATHFINISH:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ⟶ *pathfinish p* = *pathfinish q*

thm HOMOTOPIC_PATHS_IMP_PATH:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ⟶ *path p* ∧ *path q*

thm HOMOTOPIC_PATHS_IMP_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ⟶ *SUBSET* (*path_image p*) *s* ∧ *SUBSET* (*path_image q*) *s*

thm HOMOTOPIC_PATHS_REFL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p p* = (*path p* ∧ *SUBSET* (*path_image p*) *s*)

thm HOMOTOPIC_PATHS_SYM:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* = *homotopic_paths s q p*

thm HOMOTOPIC_PATHS_TRANS:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ∧ *homotopic_paths s q r* ⟶ *homotopic_paths s p r*

thm HOMOTOPIC_PATHS_EQ:

∀ (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *pathstart p* = *pathstart q* ∧ *pathfinish p* = *pathfinish q* ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *p t* = *q t*) ⟶ *homotopic_paths s p q*

thm HOMOTOPIC_PATHS_REPARAMETRIZE:

∀ (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *f*::(*real*, *unit*) *cart* ⇒ (*real*, *unit*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ∧ (∃ *f*::(*real*, *unit*) *cart* ⇒ (*real*, *unit*) *cart*. *continuous_on f* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *SUBSET* (*IMAGE f* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))])) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *f* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*) ∧ *f* (*vec* (*1*::*nat*)) = *vec* (*1*::*nat*) ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *q t* = *p* (*f t*))) ⟶ *homotopic_paths ?s p q*

thm HOMOTOPIC_PATHS_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_paths s p q* ∧ *SUBSET s* (*?t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⟶ *homotopic_paths ?t p q*

thm HOMOTOPIC_JOIN_LEMMA:

∀ (p::(real, unit) cart ⇒ (real, unit) cart ⇒ (real, ?'a::type) cart) q::(real, unit) cart ⇒ (real, unit) cart ⇒ (real, ?'a::type) cart. continuous_on (λy::(real, (unit, unit) finite_sum) cart. p (fstcart y) (sndcart y)) (GSPEC (λGEN%PVAR%1172::(real, (unit, unit) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, unit) cart. SETSPEC GEN%PVAR%1172 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x (closed_interval [(vec (0::nat), vec (1::nat))])) (pastecart t x))) ∧ continuous_on (λy::(real, (unit, unit) finite_sum) cart. q (fstcart y) (sndcart y)) (GSPEC (λGEN%PVAR%1173::(real, (unit, unit) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, unit) cart. SETSPEC GEN%PVAR%1173 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x (closed_interval [(vec (0::nat), vec (1::nat))])) (pastecart t x))) ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ pathfinish (p t) = pathstart (q t)) ⟶ continuous_on (λy::(real, (unit, unit) finite_sum) cart. ++ (p (fstcart y)) (q (fstcart y)) (sndcart y)) (GSPEC (λGEN%PVAR%1174::(real, (unit, unit) finite_sum) cart. ∃ (t::(real, unit) cart) x::(real, unit) cart. SETSPEC GEN%PVAR%1174 (IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN x (closed_interval [(vec (0::nat), vec (1::nat))])) (pastecart t x)))

thm HOMOTOPIC_PATHS_REVERSEPATH:

∀ (s::(real, ?'a::type) cart ⇒ bool) (p::(real, unit) cart ⇒ (real, ?'a::type) cart) q::(real, unit) cart ⇒ (real, ?'a::type) cart. homotopic_paths s (reversepath p) (reversepath q) = homotopic_paths s p q

thm HOMOTOPIC_PATHS_JOIN:

∀ (s::(real, ?'a::type) cart ⇒ bool) (p::(real, unit) cart ⇒ (real, ?'a::type) cart) (q::(real, unit) cart ⇒ (real, ?'a::type) cart) (p'::(real, unit) cart ⇒ (real, ?'a::type) cart) q'::(real, unit) cart ⇒ (real, ?'a::type) cart. homotopic_paths s p p' ∧ homotopic_paths s q q' ∧ pathfinish p = pathstart q ⟶ homotopic_paths s (++ p q) (++ p' q')

thm HOMOTOPIC_PATHS_CONTINUOUS_IMAGE:

∀ (f::(real, unit) cart ⇒ (real, ?'b::type) cart) (g::(real, unit) cart ⇒ (real, ?'b::type) cart) h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. homotopic_paths (?s::(real, ?'b::type) cart ⇒ bool) f g ∧ continuous_on h ?s ∧ SUBSET (IMAGE h ?s) (?t::(real, ?'a::type) cart ⇒ bool) ⟶ homotopic_paths ?t (h ∘ f) (h ∘ g)

thm HOMOTOPIC_PATHS_RID:

∀ (s::(real, ?'a::type) cart ⇒ bool) p::(real, unit) cart ⇒ (real, ?'a::type) cart. path p ∧ SUBSET (path_image p) s ⟶ homotopic_paths s (++ p (linepath (pathfinish p, pathfinish p))) p

thm HOMOTOPIC_PATHS_LID:

∀ (s::(real, ?'a::type) cart ⇒ bool) p::(real, unit) cart ⇒ (real, ?'a::type) cart. path p ∧ SUBSET (path_image p) s ⟶ homotopic_paths s (++ (linepath (pathstart p, pathstart p)) p) p

thm HOMOTOPIC_PATHS_ASSOC:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *path q* ∧ *SUBSET* (*path_image q*) *s* ∧ *path r* ∧ *SUBSET* (*path_image r*) *s* ∧ *pathfinish p* = *pathstart q* ∧ *pathfinish q* = *pathstart r* ⟶ *homotopic_paths s* (++ *p* (++ *q r*)) (++ (++ *p q*) *r*)

thm HOMOTOPIC_PATHS_RINV:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ⟶ *homotopic_paths s* (++ *p* (*reversepath p*)) (*linepath* (*pathstart p*, *pathstart p*))

thm HOMOTOPIC_PATHS_LINV:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ⟶ *homotopic_paths s* (++ (*reversepath p*) *p*) (*linepath* (*pathfinish p*, *pathfinish p*))

thm DEF_homotopic_loops:

homotopic_loops = (λ*_417572*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homotopic_with* (λ*r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish r* = *pathstart r*) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))], *_417572*))

thm homotopic_loops:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_loops s p q* = *homotopic_with* (λ*r*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish r* = *pathstart r*) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))], *s*) *p q*

thm HOMOTOPIC_LOOPS_IMP_LOOP:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_loops s p q* ⟶ *pathfinish p* = *pathstart p* ∧ *pathfinish q* = *pathstart q*

thm HOMOTOPIC_LOOPS_IMP_PATH:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_loops s p q* ⟶ *path p* ∧ *path q*

thm HOMOTOPIC_LOOPS_IMP_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *homotopic_loops s p q* ⟶ *SUBSET* (*path_image p*) *s* ∧ *SUBSET* (*path_image q*) *s*

thm HOMOTOPIC_LOOPS_REFL:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.$ $homotopic\_loops\ s\ p\ p = (path\ p \wedge SUBSET\ (path\_image\ p)\ s \wedge pathfinish\ p = pathstart\ p)$

thm HOMOTOPIC_LOOPS_SYM:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ homotopic\_loops\ s\ p\ q = homotopic\_loops\ s\ q\ p$

thm HOMOTOPIC_LOOPS_TRANS:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ r{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ homotopic\_loops\ s\ p\ q \wedge homotopic\_loops\ s\ q\ r \longrightarrow homotopic\_loops\ s\ p\ r$

thm HOMOTOPIC_LOOPS_SUBSET:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ homotopic\_loops\ s\ p\ q \wedge SUBSET\ s\ (?t{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool) \longrightarrow homotopic\_loops\ ?t\ p\ q$

thm HOMOTOPIC_PATHS_IMP_HOMOTOPIC_LOOPS:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ homotopic\_paths\ s\ p\ q \wedge pathfinish\ p = pathstart\ p \wedge pathfinish\ q = pathstart\ p \longrightarrow homotopic\_loops\ s\ p\ q$

thm HOMOTOPIC_LOOPS_IMP_HOMOTOPIC_PATHS_NULL:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ a{::}(real,\ ?'a{::}type)\ cart.\ homotopic\_loops\ s\ p\ (linepath\ (a,\ a)) \longrightarrow homotopic\_paths\ s\ p\ (linepath\ (pathstart\ p,\ pathstart\ p))$

thm HOMOTOPIC_LOOPS_CONJUGATE:

$\forall (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ path\ p \wedge SUBSET\ (path\_image\ p)\ s \wedge path\ q \wedge SUBSET\ (path\_image\ q)\ s \wedge pathfinish\ p = pathstart\ q \wedge pathfinish\ q = pathstart\ q \longrightarrow homotopic\_loops\ s\ (++\ p\ (++\ q\ (reversepath\ p)))\ q$

thm PATH_COMPONENT_IMP_HOMOTOPIC_POINTS:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (a{::}(real,\ ?'a{::}type)\ cart)\ b{::}(real,\ ?'a{::}type)\ cart.\ path\_component\ s\ a\ b \longrightarrow homotopic\_loops\ s\ (linepath\ (a,\ a))\ (linepath\ (b,\ b))$

thm HOMOTOPIC_LOOPS_IMP_PATH_COMPONENT_VALUE:

$\forall (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ (p{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (q{::}(real,\ unit)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ t{::}(real,\ unit)\ cart.\ homotopic\_loops$

s p q ∧ *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *path_component* *s* (*p t*) (*q t*)

thm HOMOTOPIC_POINTS_EQ_PATH_COMPONENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *homotopic_loops s* (*linepath* (*a*, *a*)) (*linepath* (*b*, *b*)) = *path_component s a b*

thm PATH_CONNECTED_EQ_HOMOTOPIC_POINTS:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path_connected s* = (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN a s* ∧ *IN b s* ⟶ *homotopic_loops s* (*linepath* (*a*, *a*)) (*linepath* (*b*, *b*)))

thm HOMOTOPIC_PATHS_LINEAR:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *path g* ∧ *path h* ∧ *pathstart h* = *pathstart g* ∧ *pathfinish h* = *pathfinish g* ∧ (∀ (*t*::(*real*, *unit*) *cart*) *x*::*?'a*::*type*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *SUBSET* (*closed_segment* [(*g t*, *h t*)]) *s*) ⟶ *homotopic_paths s g h*

thm HOMOTOPIC_LOOPS_LINEAR:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*. *path g* ∧ *path h* ∧ *pathfinish g* = *pathstart g* ∧ *pathfinish h* = *pathstart h* ∧ (∀ (*t*::(*real*, *unit*) *cart*) *x*::*?'a*::*type*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *SUBSET* (*closed_segment* [(*g t*, *h t*)]) *s*) ⟶ *homotopic_loops s g h*

thm HOMOTOPIC_LOOPS_NEARBY_EXPLICIT:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path g* ∧ *path h* ∧ *pathfinish g* = *pathstart g* ∧ *pathfinish h* = *pathstart h* ∧ (∀ (*t*::(*real*, *unit*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ ¬ *IN x s* ⟶ *vector_norm* (*vector_sub* (*h t*) (*g t*)) < *vector_norm* (*vector_sub* (*g t*) *x*)) ⟶ *homotopic_loops s g h*

thm HOMOTOPIC_PATHS_NEARBY_EXPLICIT:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *h*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path g* ∧ *path h* ∧ *pathstart h* = *pathstart g* ∧ *pathfinish h* = *pathfinish g* ∧ (∀ (*t*::(*real*, *unit*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *IN t* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ ¬ *IN x s* ⟶ *vector_norm* (*vector_sub* (*h t*) (*g t*)) < *vector_norm* (*vector_sub* (*g t*) *x*)) ⟶ *homotopic_paths s g h*

thm HOMOTOPIC_NEARBY_LOOPS:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *path g* ∧ *pathfinish g* = *pathstart g* ∧ *HOL_Light_Import.open s* ∧ *SUBSET* *(path_image g) s ⟶ (∃ e>0::real. ∀ h::(real, unit) cart ⇒ (real, ?'a::type) cart. path h ∧ pathfinish h = pathstart h ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ vector_norm (vector_sub (h t) (g t)) < e) ⟶ homotopic_loops s g h)*

thm HOMOTOPIC_NEARBY_PATHS:

∀ (g::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. path g ∧ HOL_Light_Import.open s ∧ SUBSET (path_image g) s ⟶ (∃ e>0::real. ∀ h::(real, unit) cart ⇒ (real, ?'a::type) cart. path h ∧ pathstart h = pathstart g ∧ pathfinish h = pathfinish g ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ vector_norm (vector_sub (h t) (g t)) < e) ⟶ homotopic_paths s g h)

thm HOMOTOPIC_JOIN_SUBPATHS:

∀ (g::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. path g ∧ SUBSET (path_image g) s ∧ IN (?u::(real, unit) cart) (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN (?v::(real, unit) cart) (closed_interval [(vec (0::nat), vec (1::nat))]) ∧ IN (?w::(real, unit) cart) (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ homotopic_paths s (++ (subpath ?u ?v g) (subpath ?v ?w g)) (subpath ?u ?w g)

thm HOMOTOPIC_LOOPS_SHIFTPATH:

∀ (s::(real, ?'a::type) cart ⇒ bool) (p::(real, unit) cart ⇒ (real, ?'a::type) cart) (q::(real, unit) cart ⇒ (real, ?'a::type) cart) u::(real, unit) cart. homotopic_loops s p q ∧ IN u (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ homotopic_loops s (shiftpath u p) (shiftpath u q)

thm HOMOTOPIC_PATHS_LOOP_PARTS:

∀ (s::(real, ?'a::type) cart ⇒ bool) (p::(real, unit) cart ⇒ (real, ?'a::type) cart) (q::(real, unit) cart ⇒ (real, ?'a::type) cart) a::(real, ?'a::type) cart. homotopic_loops s (++ p (reversepath q)) (linepath (a, a)) ∧ path q ⟶ homotopic_paths s p q

thm DEF_simply_connected:

simply_connected = (λ_422494::(real, ?'a::type) cart ⇒ bool. ∀ (p::(real, unit) cart ⇒ (real, ?'a::type) cart) q::(real, unit) cart ⇒ (real, ?'a::type) cart. path p ∧ pathfinish p = pathstart p ∧ SUBSET (path_image p) _422494 ∧ path q ∧ pathfinish q = pathstart q ∧ SUBSET (path_image q) _422494 ⟶ homotopic_loops _422494 p q)

thm simply_connected:

∀ s::(real, ?'a::type) cart ⇒ bool. simply_connected s = (∀ (p::(real, unit) cart ⇒ (real, ?'a::type) cart) q::(real, unit) cart ⇒ (real, ?'a::type) cart. path p ∧ pathfinish p = pathstart p ∧ SUBSET (path_image p) s ∧ path q ∧ pathfinish q = pathstart q ∧ SUBSET (path_image q) s ⟶ homotopic_loops s p q)

thm SIMPLY_CONNECTED_EMPTY:

simply_connected EMPTY

thm SIMPLY_CONNECTED_IMP_PATH_CONNECTED:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* ⟶ *path_connected s*

thm SIMPLY_CONNECTED_IMP_CONNECTED:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* ⟶ *connected s*

thm SIMPLY_CONNECTED_EQ_CONTRACTIBLE_LOOP_ANY:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* = (∀ (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *a*::(*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *pathfinish p* = *pathstart p* ∧ *IN a s* ⟶ *homotopic_loops s p* (*linepath* (*a*, *a*)))

thm SIMPLY_CONNECTED_EQ_CONTRACTIBLE_LOOP_SOME:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* = (*path_connected s* ∧ (∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *pathfinish p* = *pathstart p* ⟶ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *IN a s* ∧ *homotopic_loops s p* (*linepath* (*a*, *a*)))))

thm SIMPLY_CONNECTED_EQ_CONTRACTIBLE_LOOP_ALL:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* = (*s* = *EMPTY* ∨ (∃ *a*::(*real*, *?'a*::*type*) *cart*. *IN a s* ∧ (∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *pathfinish p* = *pathstart p* ⟶ *homotopic_loops s p* (*linepath* (*a*, *a*)))))

thm SIMPLY_CONNECTED_EQ_CONTRACTIBLE_PATH:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* = (*path_connected s* ∧ (∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *pathfinish p* = *pathstart p* ⟶ *homotopic_paths s p* (*linepath* (*pathstart p*, *pathstart p*))))

thm SIMPLY_CONNECTED_EQ_HOMOTOPIC_PATHS:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *simply_connected s* = (*path_connected s* ∧ (∀ (*p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *q*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path p* ∧ *SUBSET* (*path_image p*) *s* ∧ *path q* ∧ *SUBSET* (*path_image q*) *s* ∧ *pathstart q* = *pathstart p* ∧ *pathfinish q* = *pathfinish p* ⟶ *homotopic_paths s p q*))

thm HOMEOMORPHIC_SIMPLY_CONNECTED:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic s t* ∧ *simply_connected s* ⟶ *simply_connected t*

thm HOMEOMORPHIC_SIMPLY_CONNECTED_EQ:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *homeomorphic s t* ⟶ *simply_connected s* = *simply_connected t*

thm SIMPLY_CONNECTED_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ simply\_connected\ (IMAGE\ (vector\_add\ a)\ s) = simply\_connected\ s$

thm SIMPLY_CONNECTED_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow simply\_connected\ (IMAGE\ f\ s) = simply\_connected\ s$

thm NULLHOMOTOPIC_FROM_SPHERE_EXTENSION:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (a::(real, ?'b::type)\ cart)\ r::real.\ (\exists\, c::(real, ?'a::type)\ cart.\ homotopic\_with\ (\lambda x::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ True)\ (GSPEC\ (\lambda GEN\%PVAR\%1184::(real, ?'b::type)\ cart.\ \exists\, x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1184\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x),\ s)\ f\ (\lambda x::(real, ?'b::type)\ cart.\ c)) = (\exists\, g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ continuous\_on\ g\ (cball\ (a, r)) \wedge SUBSET\ (IMAGE\ g\ (cball\ (a, r)))\ s \wedge (\forall\, x::(real, ?'b::type)\ cart.\ IN\ x\ (GSPEC\ (\lambda GEN\%PVAR\%1185::(real, ?'b::type)\ cart.\ \exists\, x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1185\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x)) \longrightarrow g\ x = f\ x))$

thm DEF_contractible:

$contractible = (\lambda\_424442::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists\, a::(real, ?'a::type)\ cart.\ homotopic\_with\ (\lambda x::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ True)\ (\_424442, \_424442)\ (\lambda x::(real, ?'a::type)\ cart.\ x)\ (\lambda x::(real, ?'a::type)\ cart.\ a))$

thm contractible:

$\forall\, s::(real, ?'a::type)\ cart \Rightarrow bool.\ contractible\ s = (\exists\, a::(real, ?'a::type)\ cart.\ homotopic\_with\ (\lambda x::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ True)\ (s, s)\ (\lambda x::(real, ?'a::type)\ cart.\ x)\ (\lambda x::(real, ?'a::type)\ cart.\ a))$

thm HOMEOMORPHIC_CONTRACTIBLE:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \wedge contractible\ s \longrightarrow contractible\ t$

thm HOMEOMORPHIC_CONTRACTIBLE_EQ:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \longrightarrow contractible\ s = contractible\ t$

thm CONTRACTIBLE_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ contractible\ (IMAGE\ (vector\_add\ a)\ s) = contractible\ s$

thm CONTRACTIBLE_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow contractible\ (IMAGE\ f\ s) = contractible\ s$

thm CONTRACTIBLE_IMP_SIMPLY_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ contractible\ s \longrightarrow simply\_connected\ s$

thm STARLIKE_IMP_CONTRACTIBLE_GEN:

$\forall (P::((real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall (a::(real,\ ?'a::type)\ cart)\ t::real.\ IN\ a\ s \wedge (0::real) \leq t \wedge t \leq (1::real) \longrightarrow P\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ ((1::real) - t)\ x)\ (\%\ t\ a))) \wedge starlike\ s \longrightarrow (\exists a::(real,\ ?'a::type)\ cart.\ homotopic\_with\ P\ (s,\ s)\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ (\lambda x::(real,\ ?'a::type)\ cart.\ a))$

thm STARLIKE_IMP_CONTRACTIBLE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ starlike\ s \longrightarrow contractible\ s$

thm CONTRACTIBLE_UNIV:

$contractible\ HOL\_Light\_Import.UNIV$

thm STARLIKE_IMP_SIMPLY_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ starlike\ s \longrightarrow simply\_connected\ s$

thm CONVEX_IMP_SIMPLY_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow simply\_connected\ s$

thm STARLIKE_IMP_PATH_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ starlike\ s \longrightarrow path\_connected\ s$

thm STARLIKE_IMP_CONNECTED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ starlike\ s \longrightarrow connected\ s$

thm NULLHOMOTOPIC_THROUGH_CONTRACTIBLE:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'c::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge SUBSET\ (IMAGE\ f\ s)\ t \wedge continuous\_on\ g\ t \wedge SUBSET\ (IMAGE\ g\ t)\ (?u::(real,\ ?'a::type)\ cart \Rightarrow bool) \wedge contractible\ t \longrightarrow (\exists c::(real,\ ?'a::type)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ True)\ (s,\ ?u)\ (g \circ f)\ (\lambda x::(real,\ ?'c::type)\ cart.\ c))$

thm NULLHOMOTOPIC_INTO_CONTRACTIBLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge SUBSET\ (IMAGE\ f\ s)\ t \wedge contractible\ t \longrightarrow (\exists c::(real,\ ?'a::type)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ True)\ (s,\ t)\ f\ (\lambda x::(real,\ ?'b::type)\ cart.\ c))$

thm NULLHOMOTOPIC_FROM_CONTRACTIBLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge SUBSET\ (IMAGE\ f\ s)\ t \wedge contractible\ s \longrightarrow (\exists\, c::(real,\ ?'a::type)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ True)\ (s,\ t)\ f\ (\lambda x::(real,\ ?'b::type)\ cart.\ c))$

thm CONTRACTIBLE_PUNCTURED_SPHERE:

$\forall (a::(real,\ ?'a::type)\ cart)\ (r::real)\ b::(real,\ ?'a::type)\ cart.\ (0::real) < r \wedge vector\_norm\ (vector\_sub\ b\ a) = r \longrightarrow contractible\ (DELETE\ (GSPEC\ (\lambda GEN\%PVAR\%1190::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1190\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x))\ b)$

thm SIMPLY_CONNECTED_UNION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ t))\ s \wedge open\_in\ (subtopology\ euclidean\ (HOL\_Light\_Import.UNION\ s\ t))\ t \wedge simply\_connected\ s \wedge simply\_connected\ t \wedge path\_connected\ (HOL\_Light\_Import.INTER\ s\ t) \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow simply\_connected\ (HOL\_Light\_Import.UNION\ s\ t)$

thm SIMPLY_CONNECTED_SPHERE:

$\forall (a::(real,\ ?'a::type)\ cart)\ r::real.\ (3::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow simply\_connected\ (GSPEC\ (\lambda GEN\%PVAR\%1201::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1201\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x))$

thm DEF_face_of:

$face\_of = (\lambda(\_427396::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_427397::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ \_427396\ \_427397 \wedge convex\ \_427396 \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ a\ \_427397 \wedge IN\ b\ \_427397 \wedge IN\ x\ \_427396 \wedge IN\ x\ (open\_segment\ (a,\ b)) \longrightarrow IN\ a\ \_427396 \wedge IN\ b\ \_427396))$

thm face_of:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s = (SUBSET\ t\ s \wedge convex\ t \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge IN\ b\ s \wedge IN\ x\ t \wedge IN\ x\ (open\_segment\ (a,\ b)) \longrightarrow IN\ a\ t \wedge IN\ b\ t))$

thm FACE_OF_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ (IMAGE\ (vector\_add\ a)\ f)\ (IMAGE\ (vector\_add\ a)\ s) = face\_of\ f\ s$

thm FACE_OF_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (c::(real, ?'b::type)\ cart \Rightarrow bool)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow face\_of\ (IMAGE\ f\ c)\ (IMAGE\ f\ s) = face\_of\ c\ s$

thm FACE_OF_REFL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow face\_of\ s\ s$

thm FACE_OF_REFL_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ s\ s = convex\ s$

thm EMPTY_FACE_OF:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ EMPTY\ s$

thm FACE_OF_EMPTY:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ s\ EMPTY = (s = EMPTY)$

thm FACE_OF_TRANS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ s\ t \wedge face\_of\ t\ u \longrightarrow face\_of\ s\ u$

thm FACE_OF_FACE:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s \longrightarrow face\_of\ f\ t = (face\_of\ f\ s \wedge SUBSET\ f\ t)$

thm FACE_OF_SUBSET:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ s \wedge SUBSET\ f\ t \wedge SUBSET\ t\ s \longrightarrow face\_of\ f\ t$

thm FACE_OF_SLICE:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ s \wedge convex\ t \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ f\ t)\ (HOL\_Light\_Import.INTER\ s\ t)$

thm FACE_OF_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t1::(real, ?'a::type)\ cart \Rightarrow bool)\ t2::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t1\ s \wedge face\_of\ t2\ s \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ t1\ t2)\ s$

thm FACE_OF_INTERS:

$\forall (P::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ P \neq EMPTY \wedge (\forall t::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ P \longrightarrow face\_of\ t\ s) \longrightarrow face\_of\ (INTERS\ P)\ s$

thm FACE_OF_INTER_INTER:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ t \wedge face\_of\ f'\ t' \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ f\ f')\ (HOL\_Light\_Import.INTER\ t\ t')$

thm FACE_OF_STILLCONVEX:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow face\_of\ t\ s = (SUBSET\ t\ s \wedge convex\ (DIFF\ s\ t) \wedge t = HOL\_Light\_Import.INTER\ (hull\ affine\ t)\ s)$

thm FACE_OF_INTER_SUPPORTING_HYPERPLANE_LE_STRONG:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::real.\ convex\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1202::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1202\ (dot\ a\ x = b)\ x))) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow dot\ a\ x \leq b) \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1203::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1203\ (dot\ a\ x = b)\ x)))\ s$

thm FACE_OF_INTER_SUPPORTING_HYPERPLANE_GE_STRONG:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::real.\ convex\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1204::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1204\ (dot\ a\ x = b)\ x))) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow b \leq dot\ a\ x) \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1205::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1205\ (dot\ a\ x = b)\ x)))\ s$

thm FACE_OF_INTER_SUPPORTING_HYPERPLANE_LE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::real.\ convex\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow dot\ a\ x \leq b) \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1206::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1206\ (dot\ a\ x = b)\ x)))\ s$

thm FACE_OF_INTER_SUPPORTING_HYPERPLANE_GE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::real.\ convex\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow b \leq dot\ a\ x) \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1207::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1207\ (dot\ a\ x = b)\ x)))\ s$

thm FACE_OF_IMP_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s \longrightarrow SUBSET\ t\ s$

thm FACE_OF_IMP_CONVEX:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s \longrightarrow convex\ t$

thm FACE_OF_IMP_CLOSED:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge HOL\_Light\_Import.closed\ s \wedge face\_of\ t\ s \longrightarrow HOL\_Light\_Import.closed\ t$

thm FACE_OF_IMP_COMPACT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \wedge face\_of\ t\ s \longrightarrow compact\ t$

thm SUBSET_OF_FACE_OF:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s \wedge SUBSET\ u\ s \wedge \neg\ DISJOINT\ t\ (relative\_interior\ u) \longrightarrow SUBSET\ u\ t$

thm FACE_OF_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ t\ s \wedge face\_of\ u\ s \wedge \neg\ DISJOINT\ (relative\_interior\ t)\ (relative\_interior\ u) \longrightarrow t = u$

thm FACE_OF_DISJOINT_RELATIVE_INTERIOR:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ s \wedge f \neq s \longrightarrow HOL\_Light\_Import.INTER\ f\ (relative\_interior\ s) = EMPTY$

thm FACE_OF_DISJOINT_INTERIOR:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ s \wedge f \neq s \longrightarrow HOL\_Light\_Import.INTER\ f\ (interior\ s) = EMPTY$

thm FACE_OF_AFF_DIM_LT:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge face\_of\ f\ s \wedge f \neq s \longrightarrow aff\_dim\ f < aff\_dim\ s$

thm FACE_OF_CONVEX_HULLS:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ f\ s \wedge DISJOINT\ (hull\ affine\ f)\ (hull\ convex\ (DIFF\ s\ f)) \longrightarrow face\_of\ (hull\ convex\ f)\ (hull\ convex\ s)$

thm FACE_OF_CONVEX_HULL_INSERT:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow bool)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ a::(real, ?'a::type)\ cart.\ FINITE\ s \wedge \neg\ IN\ a\ (hull\ affine\ s) \wedge face\_of\ f\ (hull\ convex\ s) \longrightarrow face\_of\ f\ (hull\ convex\ (INSERT\ a\ s))$

thm FACE_OF_AFFINE_TRIVIAL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ f::(real, ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge face\_of\ f\ s \longrightarrow f = EMPTY \vee f = s$

thm INTERS_FACES_FINITE_BOUND:

∀ (s::(real, ?'a::type) cart ⇒ bool) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. convex s ∧ (∀ c::(real, ?'a::type) cart ⇒ bool. IN c f ⟶ face_of c s) ⟶ (∃ f'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f' ∧ SUBSET f' f ∧ CARD f' ≤ dimindex HOL_Light_Import.UNIV + (1::nat) ∧ INTERS f' = INTERS f)

thm INTERS_FACES_FINITE_ALTBOUND:

∀ (s::(real, ?'a::type) cart ⇒ bool) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. (∀ c::(real, ?'a::type) cart ⇒ bool. IN c f ⟶ face_of c s) ⟶ (∃ f'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f' ∧ SUBSET f' f ∧ CARD f' ≤ dimindex HOL_Light_Import.UNIV + (2::nat) ∧ INTERS f' = INTERS f)

thm FACES_OF_TRANSLATION:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. GSPEC (λGEN%PVAR%1208::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1208 (face_of f (IMAGE (vector_add a) s)) f) = IMAGE (IMAGE (vector_add a)) (GSPEC (λGEN%PVAR%1209::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1209 (face_of f s) f))

thm FACES_OF_LINEAR_IMAGE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. linear f ∧ (∀ (x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f x = f y ⟶ x = y) ⟶ GSPEC (λGEN%PVAR%1214::(real, ?'a::type) cart ⇒ bool. ∃ t::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1214 (face_of t (IMAGE f s)) t) = IMAGE (IMAGE f) (GSPEC (λGEN%PVAR%1215::(real, ?'b::type) cart ⇒ bool. ∃ t::(real, ?'b::type) cart ⇒ bool. SETSPEC GEN%PVAR%1215 (face_of t s) t))

thm FACE_OF_CONIC:

∀ (s::(real, ?'a::type) cart ⇒ bool) f::(real, ?'a::type) cart ⇒ bool. conic s ∧ face_of f s ⟶ conic f

thm DEF_exposed_face_of:

exposed_face_of = (λ(_434681::(real, ?'a::type) cart ⇒ bool) _434682::(real, ?'a::type) cart ⇒ bool. face_of _434681 _434682 ∧ (∃ (a::(real, ?'a::type) cart) b::real. SUBSET _434682 (GSPEC (λGEN%PVAR%1216::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1216 (dot a x ≤ b) x)) ∧ _434681 = HOL_Light_Import.INTER _434682 (GSPEC (λGEN%PVAR%1217::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1217 (dot a x = b) x))))

thm exposed_face_of:

∀ (t::(real, ?'a::type) cart ⇒ bool) s::(real, ?'a::type) cart ⇒ bool. exposed_face_of t s = (face_of t s ∧ (∃ (a::(real, ?'a::type) cart) b::real. SUBSET s (GSPEC (λGEN%PVAR%1216::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SET-SPEC GEN%PVAR%1216 (dot a x ≤ b) x)) ∧ t = HOL_Light_Import.INTER *s (GSPEC (λGEN%PVAR%1217::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1217 (dot a x = b) x))))*

thm EMPTY_EXPOSED_FACE_OF:

∀s::(real, ?'a::type) cart ⇒ bool. exposed_face_of EMPTY s

thm EXPOSED_FACE_OF_REFL_EQ:

∀s::(real, ?'a::type) cart ⇒ bool. exposed_face_of s s = convex s

thm EXPOSED_FACE_OF_REFL:

∀s::(real, ?'a::type) cart ⇒ bool. convex s ⟶ exposed_face_of s s

thm EXPOSED_FACE_OF:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. exposed_face_of t s = (face_of t s ∧ (t = EMPTY ∨ t = s ∨ (∃(a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ SUBSET s (GSPEC (λGEN%PVAR%1218::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1218 (dot a x ≤ b) x)) ∧ t = HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%1219::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1219 (dot a x = b) x)))))

thm EXPOSED_FACE_OF_TRANSLATION_EQ:

∀(a::(real, ?'a::type) cart) (f::(real, ?'a::type) cart ⇒ bool) s::(real, ?'a::type) cart ⇒ bool. exposed_face_of (IMAGE (vector_add a) f) (IMAGE (vector_add a) s) = exposed_face_of f s

thm EXPOSED_FACE_OF_LINEAR_IMAGE:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (c::(real, ?'b::type) cart ⇒ bool) s::(real, ?'b::type) cart ⇒ bool. linear f ∧ (∀(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f x = f y ⟶ x = y) ∧ (∀y::(real, ?'a::type) cart. ∃x::(real, ?'b::type) cart. f x = y) ⟶ exposed_face_of (IMAGE f c) (IMAGE f s) = exposed_face_of c s

thm EXPOSED_FACE_OF_INTER:

∀(s::(real, ?'a::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) u::(real, ?'a::type) cart ⇒ bool. exposed_face_of t s ∧ exposed_face_of u s ⟶ exposed_face_of (HOL_Light_Import.INTER t u) s

thm EXPOSED_FACE_OF_INTERS:

∀(P::((real, ?'a::type) cart ⇒ bool) ⇒ bool) s::(real, ?'a::type) cart ⇒ bool. P ≠ EMPTY ∧ (∀t::(real, ?'a::type) cart ⇒ bool. IN t P ⟶ exposed_face_of t s) ⟶ exposed_face_of (INTERS P) s

thm EXPOSED_FACE_OF_INTER_SUPPORTING_HYPERPLANE_LE:

∀(s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::real. convex s ∧ (∀x::(real, ?'a::type) cart. IN x s ⟶ dot a x ≤ b) ⟶ exposed_face_of

(*HOL_Light_Import.INTER s* (*GSPEC* (*λGEN%PVAR%1222*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1222* (*dot a x* = *b*) *x*))) *s*

thm EXPOSED_FACE_OF_INTER_SUPPORTING_HYPERPLANE_GE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *convex s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a x*) ⟶ *exposed_face_of* (*HOL_Light_Import.INTER s* (*GSPEC* (*λGEN%PVAR%1223*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1223* (*dot a x* = *b*) *x*))) *s*

thm EXPOSED_FACE_OF_SUMS:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *f*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* ∧ *convex t* ∧ *exposed_face_of f* (*GSPEC* (*λGEN%PVAR%1227*::(*real*, *?'a*::*type*) *cart*. ∃ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1227* (*IN x s* ∧ *IN y t*) (*vector_add x y*))) ⟶ (∃ (*k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *l*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *exposed_face_of k s* ∧ *exposed_face_of l t* ∧ *f* = *GSPEC* (*λGEN%PVAR%1228*::(*real*, *?'a*::*type*) *cart*. ∃ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1228* (*IN x k* ∧ *IN y l*) (*vector_add x y*)))

thm EXPOSED_FACE_OF_PARALLEL:

∀ (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *exposed_face_of t s* = (*face_of t s* ∧ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *SUBSET s* (*GSPEC* (*λGEN%PVAR%1239*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1239* (*dot a x* ≤ *b*) *x*)) ∧ *t* = *HOL_Light_Import.INTER s* (*GSPEC* (*λGEN%PVAR%1240*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1240* (*dot a x* = *b*) *x*)) ∧ (*t* ≠ *EMPTY* ∧ *t* ≠ *s* ⟶ *a* ≠ *vec* (*0*::*nat*)) ∧ (∀ *w*::(*real*, *?'a*::*type*) *cart*. *IN w* (*hull affine s*) ∧ *t* ≠ *s* ⟶ *IN* (*vector_add w a*) (*hull affine s*))))

thm DEF_extreme_point_of:

extreme_point_of = (λ(*_435409*::(*real*, *?'a*::*type*) *cart*) *_435410*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN _435409 _435410* ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN a _435410* ∧ *IN b _435410* ⟶ ¬ *IN _435409* (*open_segment* (*a*, *b*))))

thm extreme_point_of:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *extreme_point_of x s* = (*IN x s* ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN a s* ∧ *IN b s* ⟶ ¬ *IN x* (*open_segment* (*a*, *b*))))

thm EXTREME_POINT_OF_STILLCONVEX:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *convex s* ⟶ *extreme_point_of x s* = (*IN x s* ∧ *convex* (*DELETE s x*))

thm FACE_OF_SING:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ (INSERT\ x\ EMPTY)\ s = extreme\_point\_of\ x\ s$

thm EXTREME_POINT_NOT_IN_RELATIVE_INTERIOR:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ extreme\_point\_of\ x\ s \wedge s \neq INSERT\ x\ EMPTY \longrightarrow \neg\ IN\ x\ (relative\_interior\ s)$

thm EXTREME_POINT_NOT_IN_INTERIOR:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ extreme\_point\_of\ x\ s \longrightarrow \neg\ IN\ x\ (interior\ s)$

thm EXTREME_POINT_OF_FACE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ face\_of\ f\ s \longrightarrow extreme\_point\_of\ v\ f = (extreme\_point\_of\ v\ s \wedge IN\ v\ f)$

thm EXTREME_POINT_OF_MIDPOINT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ convex\ s \longrightarrow extreme\_point\_of\ x\ s = (IN\ x\ s \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge IN\ b\ s \wedge x = midpoint\ (a,\ b) \longrightarrow x = a \wedge x = b))$

thm EXTREME_POINT_OF_CONVEX_HULL:

$\forall (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ extreme\_point\_of\ x\ (hull\ convex\ s) \longrightarrow IN\ x\ s$

thm EXTREME_POINTS_OF_CONVEX_HULL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%1241::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1241\ (extreme\_point\_of\ x\ (hull\ convex\ s))\ x))\ s$

thm EXTREME_POINT_OF_EMPTY:

$\forall x::(real,\ ?'a::type)\ cart.\ \neg\ extreme\_point\_of\ x\ EMPTY$

thm EXTREME_POINT_OF_SING:

$\forall (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ extreme\_point\_of\ x\ (INSERT\ a\ EMPTY) = (x = a)$

thm EXTREME_POINT_OF_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ extreme\_point\_of\ (vector\_add\ a\ x)\ (IMAGE\ (vector\_add\ a)\ s) = extreme\_point\_of\ x\ s$

thm EXTREME_POINT_OF_LINEAR_IMAGE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow extreme\_point\_of\ (f$

(*?x*::(*real*, *?'b*::*type*) *cart*)) (*IMAGE f* (*?s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*)) = *extreme_point_of ?x ?s*

thm EXTREME_POINTS_OF_TRANSLATION:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%1242*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1242* (*extreme_point_of x* (*IMAGE* (*vector_add a*) *s*)) *x*) = *IMAGE* (*vector_add a*) (*GSPEC* (λ*GEN%PVAR%1243*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1243* (*extreme_point_of x s*) *x*))

thm EXTREME_POINT_OF_INTER:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *extreme_point_of x s* ∧ *extreme_point_of x t* ⟶ *extreme_point_of x* (*HOL_Light_Import.INTER s t*)

thm EXTREME_POINTS_OF_LINEAR_IMAGE:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *GSPEC* (λ*GEN%PVAR%1244*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1244* (*extreme_point_of y* (*IMAGE f* (*?s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*))) *y*) = *IMAGE f* (*GSPEC* (λ*GEN%PVAR%1245*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1245* (*extreme_point_of x ?s*) *x*))

thm EXTREME_POINT_OF_INTER_SUPPORTING_HYPERPLANE_LE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::(*real*, *?'a*::*type*) *cart*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a x* ≤ *b*) ∧ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%1246*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1246* (*dot a x* = *b*) *x*)) = *INSERT c EMPTY* ⟶ *extreme_point_of c s*

thm EXTREME_POINT_OF_INTER_SUPPORTING_HYPERPLANE_GE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::(*real*, *?'a*::*type*) *cart*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a x*) ∧ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%1247*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1247* (*dot a x* = *b*) *x*)) = *INSERT c EMPTY* ⟶ *extreme_point_of c s*

thm EXPOSED_POINT_OF_INTER_SUPPORTING_HYPERPLANE_LE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::(*real*, *?'a*::*type*) *cart*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *dot a x* ≤ *b*) ∧ *HOL_Light_Import.INTER s* (*GSPEC* (λ*GEN%PVAR%1248*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1248* (*dot a x* = *b*) *x*)) = *INSERT c EMPTY* ⟶ *exposed_face_of* (*INSERT c EMPTY*) *s*

thm EXPOSED_POINT_OF_INTER_SUPPORTING_HYPERPLANE_GE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *c*::(*real*, *?'a*::*type*) *cart*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *b* ≤ *dot a x*) ∧ *HOL_Light_Import.INTER*

s $(GSPEC$ $(\lambda GEN\%PVAR\%1249::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1249\ (dot\ a\ x = b)\ x)) = INSERT\ c\ EMPTY \longrightarrow exposed\_face\_of\ (INSERT\ c\ EMPTY)\ s$

thm EXPOSED_POINT_OF_FURTHEST_POINT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ b\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow distance\ (a,\ x) \leq distance\ (a,\ b)) \longrightarrow exposed\_face\_of\ (INSERT\ b\ EMPTY)\ s$

thm COLLINEAR_EXTREME_POINTS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ collinear\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1250::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1250\ (extreme\_point\_of\ x\ s)\ x)) \wedge CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1251::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1251\ (extreme\_point\_of\ x\ s)\ x)) \leq (2::nat)$

thm EXTREME_POINT_OF_CONIC:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ conic\ s \wedge extreme\_point\_of\ x\ s \longrightarrow x = vec\ (0::nat)$

thm EXTREME_POINT_OF_CONVEX_HULL_INSERT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ FINITE\ s \wedge \neg\ IN\ a\ (hull\ convex\ s) \longrightarrow extreme\_point\_of\ a\ (hull\ convex\ (INSERT\ a\ s))$

thm DEF_facet_of:

$facet\_of = (\lambda(\_438215::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_438216::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ \_438215\ \_438216 \wedge \_438215 \neq EMPTY \wedge aff\_dim\ \_438215 = aff\_dim\ \_438216 - int\ (1::nat))$

thm facet_of:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ facet\_of\ f\ s = (face\_of\ f\ s \wedge f \neq EMPTY \wedge aff\_dim\ f = aff\_dim\ s - int\ (1::nat))$

thm FACET_OF_EMPTY:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ facet\_of\ s\ EMPTY$

thm FACET_OF_REFL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ facet\_of\ s\ s$

thm FACET_OF_IMP_FACE_OF:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ facet\_of\ f\ s \longrightarrow face\_of\ f\ s$

thm FACET_OF_IMP_SUBSET:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ facet\_of\ f\ s \longrightarrow SUBSET\ f\ s$

thm FACET_OF_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ facet\_of\ (IMAGE\ (vector\_add\ a)\ f)\ (IMAGE\ (vector\_add\ a)\ s) = facet\_of\ f\ s$

thm FACET_OF_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow facet\_of\ (IMAGE\ f\ c)\ (IMAGE\ f\ s) = facet\_of\ c\ s$

thm DEF_edge_of:

$edge\_of = (\lambda(\_438241::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_438242::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\ \_438241\ \_438242 \wedge aff\_dim\ \_438241 = int\ (1::nat))$

thm edge_of:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::(real,\ ?'a::type)\ cart \Rightarrow bool.\ edge\_of\ e\ s = (face\_of\ e\ s \wedge aff\_dim\ e = int\ (1::nat))$

thm EDGE_OF_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ edge\_of\ (IMAGE\ (vector\_add\ a)\ f)\ (IMAGE\ (vector\_add\ a)\ s) = edge\_of\ f\ s$

thm EDGE_OF_LINEAR_IMAGE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (c::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow edge\_of\ (IMAGE\ f\ c)\ (IMAGE\ f\ s) = edge\_of\ c\ s$

thm EDGE_OF_IMP_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ edge\_of\ s\ t \longrightarrow SUBSET\ s\ t$

thm DIFFERENT_NORM_3_COLLINEAR_POINTS:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ \neg (IN\ x\ (open\_segment\ (a,\ b)) \wedge vector\_norm\ a = vector\_norm\ b \wedge vector\_norm\ x = vector\_norm\ b)$

thm EXTREME_POINT_EXISTS_CONVEX:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge convex\ s \wedge s \neq EMPTY \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ extreme\_point\_of\ x\ s)$

thm KREIN_MILMAN:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \longrightarrow s = closure\ (hull\ convex\ (GSPEC\ (\lambda GEN\%PVAR\%1258::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1258\ (extreme\_point\_of\ x\ s)\ x)))$

thm KREIN_MILMAN_MINKOWSKI:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \longrightarrow s = hull\ convex\ (GSPEC\ (\lambda GEN\%PVAR\%1264::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1264\ (extreme\_point\_of\ x\ s)\ x))$

thm KREIN_MILMAN_POLYTOPE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow hull\ convex\ s = hull\ convex\ (GSPEC\ (\lambda GEN\%PVAR\%1265::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1265\ (extreme\_point\_of\ x\ (hull\ convex\ s))\ x))$

thm EXTREME_POINTS_OF_CONVEX_HULL_EQ:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge (\forall t::(real, ?'a::type)\ cart \Rightarrow bool.\ PSUBSET\ t\ s \longrightarrow hull\ convex\ t \neq hull\ convex\ s) \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%1267::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1267\ (extreme\_point\_of\ x\ (hull\ convex\ s))\ x) = s$

thm EXTREME_POINT_OF_CONVEX_HULL_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ compact\ s \wedge (\forall t::(real, ?'a::type)\ cart \Rightarrow bool.\ PSUBSET\ t\ s \longrightarrow hull\ convex\ t \neq hull\ convex\ s) \longrightarrow extreme\_point\_of\ x\ (hull\ convex\ s) = IN\ x\ s$

thm EXTREME_POINT_OF_CONVEX_HULL_CONVEX_INDEPENDENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ compact\ s \wedge (\forall a::(real, ?'a::type)\ cart.\ IN\ a\ s \longrightarrow \neg\ IN\ a\ (hull\ convex\ (DELETE\ s\ a))) \longrightarrow extreme\_point\_of\ x\ (hull\ convex\ s) = IN\ x\ s$

thm EXTREME_POINT_OF_CONVEX_HULL_AFFINE_INDEPENDENT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ \neg\ affine\_dependent\ s \longrightarrow extreme\_point\_of\ x\ (hull\ convex\ s) = IN\ x\ s$

thm EXTREME_POINT_OF_CONVEX_HULL_2:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ extreme\_point\_of\ x\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ EMPTY))) = (x = a \vee x = b)$

thm EXTREME_POINT_OF_SEGMENT:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ x::(real, ?'a::type)\ cart.\ extreme\_point\_of\ x\ (closed\_segment\ [(a, b)]) = (x = a \vee x = b)$

thm FACE_OF_CONVEX_HULL_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge face\_of\ t\ (hull\ convex\ s) \longrightarrow (\exists s'::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s'\ s \wedge t = hull\ convex\ s')$

thm FACE_OF_CONVEX_HULL_AFFINE_INDEPENDENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \longrightarrow face\_of\ t\ (hull\ convex\ s) = (\exists\ c::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ c\ s \wedge t = hull\ convex\ c)$

thm FACET_OF_CONVEX_HULL_AFFINE_INDEPENDENT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \longrightarrow facet\_of\ t\ (hull\ convex\ s) = (t \neq EMPTY \wedge (\exists\ u::(real,\ ?'a::type)\ cart.\ IN\ u\ s \wedge t = hull\ convex\ (DELETE\ s\ u)))$

thm FACET_OF_CONVEX_HULL_AFFINE_INDEPENDENT_ALT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \longrightarrow facet\_of\ t\ (hull\ convex\ s) = ((2::nat) \leq CARD\ s \wedge (\exists\ u::(real,\ ?'a::type)\ cart.\ IN\ u\ s \wedge t = hull\ convex\ (DELETE\ s\ u)))$

thm SEGMENT_FACE_OF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ face\_of\ (closed\_segment\ [(a,\ b)])\ s \longrightarrow extreme\_point\_of\ a\ s \wedge extreme\_point\_of\ b\ s$

thm SEGMENT_EDGE_OF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ edge\_of\ (closed\_segment\ [(a,\ b)])\ s \longrightarrow a \neq b \wedge extreme\_point\_of\ a\ s \wedge extreme\_point\_of\ b\ s$

thm COMPACT_CONVEX_COLLINEAR_SEGMENT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ s \neq EMPTY \wedge compact\ s \wedge convex\ s \wedge collinear\ s \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ s = closed\_segment\ [(a,\ b)])$

thm KREIN_MILMAN_RELATIVE_FRONTIER:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \wedge \neg (\exists\ a::(real,\ ?'a::type)\ cart.\ s = INSERT\ a\ EMPTY) \longrightarrow s = hull\ convex\ (DIFF\ s\ (relative\_interior\ s))$

thm KREIN_MILMAN_FRONTIER:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge compact\ s \longrightarrow s = hull\ convex\ (frontier\ s)$

thm EXTREME_POINT_OF_CONVEX_HULL_INSERT_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ FINITE\ s \wedge \neg\ IN\ a\ (hull\ affine\ s) \longrightarrow extreme\_point\_of\ x\ (hull\ convex\ (INSERT\ a\ s)) = (x = a \vee extreme\_point\_of\ x\ (hull\ convex\ s))$

thm FACE_OF_CONVEX_HULL_INSERT_EQ:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ FINITE\ s \wedge \neg\ IN\ a\ (hull\ affine\ s) \longrightarrow face\_of\ f\ (hull\ convex$

$(INSERT\ a\ s)) = (face\_of\, f\ (hull\ convex\ s) \lor (\exists f'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ face\_of\, f'\ (hull\ convex\ s) \land f = hull\ convex\ (INSERT\ a\ f')))$

thm DEF_polytope:

$polytope = (\lambda\_441572::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ v \land \_441572 = hull\ convex\ v)$

thm polytope:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s = (\exists v::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ v \land s = hull\ convex\ v)$

thm POLYTOPE_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ (IMAGE\ (vector\_add\ a)\ s) = polytope\ s$

thm POLYTOPE_LINEAR_IMAGE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\, f \land polytope\ (?p::(real,\ ?'b::type)\ cart \Rightarrow bool) \longrightarrow polytope\ (IMAGE\, f\ ?p)$

thm POLYTOPE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\, f \land (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\, x = f\, y \longrightarrow x = y) \land (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\, x = y) \longrightarrow polytope\ (IMAGE\, f\, s) = polytope\ s$

thm POLYTOPE_NEGATIONS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow polytope\ (IMAGE\ vector\_neg\ s)$

thm POLYTOPE_CONVEX_HULL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow polytope\ (hull\ convex\ s)$

thm FACE_OF_POLYTOPE_POLYTOPE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \land face\_of\, f\, s \longrightarrow polytope\, f$

thm FINITE_POLYTOPE_FACES:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\, s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1274::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1274\ (face\_of\, f\, s)\ f))$

thm FINITE_POLYTOPE_FACETS:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\, s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1278::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1278\ (facet\_of\, f\, s)\ f))$

thm POLYTOPE_SCALING:

$\forall (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow polytope\ (IMAGE\ (\%\ c)\ s)$

thm POLYTOPE_SCALING_EQ:

$\forall (c::real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ c \neq (0::real) \longrightarrow polytope\ (IMAGE\ (\%\ c)\ s) = polytope\ s$

thm POLYTOPE_SUMS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \wedge polytope\ t \longrightarrow polytope\ (GSPEC\ (\lambda GEN\%PVAR\%1280::(real,\ ?'a::type)\ cart.\ \exists (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1280\ (IN\ x\ s \wedge IN\ y\ t)\ (vector\_add\ x\ y)))$

thm POLYTOPE_IMP_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow compact\ s$

thm POLYTOPE_IMP_CONVEX:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow convex\ s$

thm POLYTOPE_IMP_CLOSED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow HOL\_Light\_Import.closed\ s$

thm POLYTOPE_IMP_BOUNDED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \longrightarrow bounded\ s$

thm POLYTOPE_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ polytope\ (closed\_interval\ [(a,\ b)])$

thm POLYTOPE_SING:

$\forall a::(real,\ ?'a::type)\ cart.\ polytope\ (INSERT\ a\ EMPTY)$

thm DEF_polyhedron:

$polyhedron = (\lambda\_441708::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge \_441708 = INTERS\ f \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1281::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1281\ (dot\ a\ x \leq b)\ x))))$

thm polyhedron:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (\exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = INTERS\ f \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1281::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1281\ (dot\ a\ x \leq b)\ x))))$

thm POLYHEDRON_INTER:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *polyhedron* *s* ∧ *polyhedron* *t* ⟶ *polyhedron* (*HOL_Light_Import.INTER* *s* *t*)

thm POLYHEDRON_UNIV:

polyhedron *HOL_Light_Import.UNIV*

thm POLYHEDRON_POSITIVE_ORTHANT:

polyhedron (*GSPEC* (λ*GEN%PVAR%1283*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1283* (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex* *HOL_Light_Import.UNIV* ⟶ (*0*::*real*) ≤ $ *x* *i*) *x*))

thm POLYHEDRON_INTERS:

∀ *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE* *f* ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* *s* *f* ⟶ *polyhedron* *s*) ⟶ *polyhedron* (*INTERS* *f*)

thm POLYHEDRON_EMPTY:

polyhedron *EMPTY*

thm POLYHEDRON_HALFSPACE_LE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *polyhedron* (*GSPEC* (λ*GEN%PVAR%1288*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1288* (*dot* *a* *x* ≤ *b*) *x*))

thm POLYHEDRON_HALFSPACE_GE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *polyhedron* (*GSPEC* (λ*GEN%PVAR%1289*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1289* (*b* ≤ *dot* *a* *x*) *x*))

thm POLYHEDRON_HYPERPLANE:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *polyhedron* (*GSPEC* (λ*GEN%PVAR%1293*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1293* (*dot* *a* *x* = *b*) *x*))

thm AFFINE_IMP_POLYHEDRON:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *affine* *s* ⟶ *polyhedron* *s*

thm POLYHEDRON_IMP_CLOSED:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *polyhedron* *s* ⟶ *HOL_Light_Import.closed* *s*

thm POLYHEDRON_IMP_CONVEX:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *polyhedron* *s* ⟶ *convex* *s*

thm POLYHEDRON_AFFINE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *polyhedron* (*hull* *affine* *s*)

thm POLYHEDRON_INTER_AFFINE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (\exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1294::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1294\ (dot\ a\ x \leq b)\ x))))$

thm POLYHEDRON_INTER_AFFINE_PARALLEL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (\exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1298::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1298\ (dot\ a\ x \leq b)\ x) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (hull\ affine\ s) \longrightarrow IN\ (vector\_add\ x\ a)\ (hull\ affine\ s)))))$

thm POLYHEDRON_INTER_AFFINE_PARALLEL_MINIMAL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (\exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1299::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1299\ (dot\ a\ x \leq b)\ x) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (hull\ affine\ s) \longrightarrow IN\ (vector\_add\ x\ a)\ (hull\ affine\ s)))) \wedge (\forall f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ f'\ f \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f'))))$

thm POLYHEDRON_INTER_AFFINE_MINIMAL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (\exists f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::real.\ a \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1300::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1300\ (dot\ a\ x \leq b)\ x))) \wedge (\forall f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ f'\ f \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f'))))$

thm RELATIVE_INTERIOR_POLYHEDRON_EXPLICIT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ b::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow real.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow a\ h \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1302::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1302\ (dot\ (a\ h)\ x \leq b\ h)\ x)) \wedge (\forall f'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ f'\ f \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f'))) \longrightarrow relative\_interior\ s = GSPEC\ (\lambda GEN\%PVAR\%1303::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$

cart. SETSPEC GEN%PVAR%1303 (IN x s ∧ (∀ h::(real, ?'a::type) cart ⇒ bool. IN h f ⟶ dot (a h) x < b h)) x)

thm FACET_OF_POLYHEDRON_EXPLICIT:

∀ (s::(real, ?'a::type) cart ⇒ bool) (f::((real, ?'a::type) cart ⇒ bool) ⇒ bool) (a::((real, ?'a::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart) b::((real, ?'a::type) cart ⇒ bool) ⇒ real. FINITE f ∧ s = HOL_Light_Import.INTER (hull affine s) (INTERS f) ∧ (∀ h::(real, ?'a::type) cart ⇒ bool. IN h f ⟶ a h ≠ vec (0::nat) ∧ h = GSPEC (λGEN%PVAR%1311::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1311 (dot (a h) x ≤ b h) x)) ∧ (∀ f'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. PSUBSET f' f ⟶ PSUBSET s (HOL_Light_Import.INTER (hull affine s) (INTERS f'))) ⟶ (∀ c::(real, ?'a::type) cart ⇒ bool. facet_of c s = (∃ h::(real, ?'a::type) cart ⇒ bool. IN h f ∧ c = HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%1312::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1312 (dot (a h) x = b h) x))))

thm FACE_OF_POLYHEDRON_SUBSET_EXPLICIT:

∀ (s::(real, ?'a::type) cart ⇒ bool) (f::((real, ?'a::type) cart ⇒ bool) ⇒ bool) (a::((real, ?'a::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart) b::((real, ?'a::type) cart ⇒ bool) ⇒ real. FINITE f ∧ s = HOL_Light_Import.INTER (hull affine s) (INTERS f) ∧ (∀ h::(real, ?'a::type) cart ⇒ bool. IN h f ⟶ a h ≠ vec (0::nat) ∧ h = GSPEC (λGEN%PVAR%1314::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1314 (dot (a h) x ≤ b h) x)) ∧ (∀ f'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. PSUBSET f' f ⟶ PSUBSET s (HOL_Light_Import.INTER (hull affine s) (INTERS f'))) ⟶ (∀ c::(real, ?'a::type) cart ⇒ bool. face_of c s ∧ c ≠ EMPTY ∧ c ≠ s ⟶ (∃ h::(real, ?'a::type) cart ⇒ bool. IN h f ∧ SUBSET c (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%1315::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1315 (dot (a h) x = b h) x)))))

thm FACE_OF_POLYHEDRON_EXPLICIT:

∀ (s::(real, ?'a::type) cart ⇒ bool) (f::((real, ?'a::type) cart ⇒ bool) ⇒ bool) (a::((real, ?'a::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart) b::((real, ?'a::type) cart ⇒ bool) ⇒ real. FINITE f ∧ s = HOL_Light_Import.INTER (hull affine s) (INTERS f) ∧ (∀ h::(real, ?'a::type) cart ⇒ bool. IN h f ⟶ a h ≠ vec (0::nat) ∧ h = GSPEC (λGEN%PVAR%1333::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1333 (dot (a h) x ≤ b h) x)) ∧ (∀ f'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. PSUBSET f' f ⟶ PSUBSET s (HOL_Light_Import.INTER (hull affine s) (INTERS f'))) ⟶ (∀ c::(real, ?'a::type) cart ⇒ bool. face_of c s ∧ c ≠ EMPTY ∧ c ≠ s ⟶ c = INTERS (GSPEC (λGEN%PVAR%1338::(real, ?'a::type) cart ⇒ bool. ∃ h::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1338 (IN h f ∧ SUBSET c (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%1337::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1337 (dot (a h) x = b h) x)))) (HOL_Light_Import.INTER *s (GSPEC (λGEN%PVAR%1336::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1336 (dot (a h) x = b h) x))))))*

thm FACET_OF_POLYHEDRON:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ facet_of c s ⟶ (∃ (a::(real, ?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ SUBSET s (GSPEC (λGEN%PVAR%1339::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1339 (dot a x ≤ b) x)) ∧ c = HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%1340::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1340 (dot a x = b) x)))

thm FACE_OF_POLYHEDRON:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ face_of c s ∧ c ≠ EMPTY ∧ c ≠ s ⟶ c = INTERS (GSPEC (λGEN%PVAR%1341::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1341 (facet_of f s ∧ SUBSET c f) f))

thm FACE_OF_POLYHEDRON_SUBSET_FACET:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ face_of c s ∧ c ≠ EMPTY ∧ c ≠ s ⟶ (∃ f::(real, ?'a::type) cart ⇒ bool. facet_of f s ∧ SUBSET c f)

thm EXPOSED_FACE_OF_POLYHEDRON:

∀ (s::(real, ?'a::type) cart ⇒ bool) f::(real, ?'a::type) cart ⇒ bool. polyhedron s ⟶ exposed_face_of f s = face_of f s

thm FACE_OF_POLYHEDRON_POLYHEDRON:

∀ (s::(real, ?'a::type) cart ⇒ bool) c::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ face_of c s ⟶ polyhedron c

thm FINITE_POLYHEDRON_FACES:

∀ s::(real, ?'a::type) cart ⇒ bool. polyhedron s ⟶ FINITE (GSPEC (λGEN%PVAR%1358::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1358 (face_of f s) f))

thm FINITE_POLYHEDRON_EXPOSED_FACES:

∀ s::(real, ?'a::type) cart ⇒ bool. polyhedron s ⟶ FINITE (GSPEC (λGEN%PVAR%1359::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1359 (exposed_face_of f s) f))

thm FINITE_POLYHEDRON_EXTREME_POINTS:

∀ s::(real, ?'a::type) cart ⇒ bool. polyhedron s ⟶ FINITE (GSPEC (λGEN%PVAR%1362::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1362 (extreme_point_of v s) v))

thm FINITE_POLYHEDRON_FACETS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1366::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1366\ (facet\_of\ f\ s)\ f))$

thm RELATIVE_INTERIOR_OF_POLYHEDRON:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s \longrightarrow relative\_interior\ s = DIFF\ s\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%1369::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1369\ (facet\_of\ f\ s)\ f)))$

thm RELATIVE_FRONTIER_OF_POLYHEDRON:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s \longrightarrow DIFF\ s\ (relative\_interior\ s) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%1370::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1370\ (facet\_of\ f\ s)\ f))$

thm FACETS_OF_POLYHEDRON_EXPLICIT_DISTINCT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ b::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow real.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f) \wedge (\forall h::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ h\ f \longrightarrow a\ h \neq vec\ (0::nat) \wedge h = GSPEC\ (\lambda GEN\%PVAR\%1374::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1374\ (dot\ (a\ h)\ x \leq b\ h)\ x)) \wedge (\forall f'::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ f'\ f \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ f'))) \longrightarrow (\forall (h1::(real, ?'a::type)\ cart \Rightarrow bool)\ h2::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ h1\ f \wedge IN\ h2\ f \wedge HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1375::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1375\ (dot\ (a\ h1)\ x = b\ h1)\ x)) = HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1376::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1376\ (dot\ (a\ h2)\ x = b\ h2)\ x)) \longrightarrow h1 = h2)$

thm POLYHEDRON_EQ_FINITE_EXPOSED_FACES:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (HOL\_Light\_Import.closed\ s \wedge convex\ s \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1389::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1389\ (exposed\_face\_of\ f\ s)\ f)))$

thm POLYHEDRON_EQ_FINITE_FACES:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s = (HOL\_Light\_Import.closed\ s \wedge convex\ s \wedge FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1391::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1391\ (face\_of\ f\ s)\ f)))$

thm POLYHEDRON_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ (IMAGE\ (vector\_add\ a)\ s) = polyhedron\ s$

thm POLYHEDRON_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow polyhedron\ (IMAGE\ f\ s) = polyhedron\ s$

thm POLYHEDRON_NEGATIONS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ s \longrightarrow polyhedron\ (IMAGE\ vector\_neg\ s)$

thm POLYTOPE_EQ_BOUNDED_POLYHEDRON:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ s = (polyhedron\ s \wedge bounded\ s)$

thm POLYTOPE_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ s \wedge polytope\ t \longrightarrow polytope\ (HOL\_Light\_Import.INTER\ s\ t)$

thm POLYTOPE_IMP_POLYHEDRON:

$\forall p::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \longrightarrow polyhedron\ p$

thm POLYTOPE_FACET_EXISTS:

$\forall p::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \wedge int\ (0::nat) < aff\_dim\ p \longrightarrow (\exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ facet\_of\ f\ p)$

thm POLYHEDRON_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ polyhedron\ (closed\_interval\ [(a, b)])$

thm POLYHEDRON_CONVEX_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow polyhedron\ (hull\ convex\ s)$

thm POLYTOPE_UNION_CONVEX_HULL_FACETS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ p::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \wedge int\ (0::nat) < aff\_dim\ p \wedge s \neq EMPTY \wedge SUBSET\ s\ p \longrightarrow p = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%1395::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1395\ (facet\_of\ f\ p)\ (hull\ convex\ (HOL\_Light\_Import.UNION\ s\ f))))$

thm POLYHEDRON_CONVEX_CONE_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow polyhedron\ (hull\ convex\_cone\ s)$

thm CLOSED_CONVEX_CONE_HULL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow HOL\_Light\_Import.closed\ (hull\ convex\_cone\ s)$

thm FINITELY_GENERATED_CONIC_POLYHEDRON:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ conic s ∧ s ≠ EMPTY ⟶ (∃ c::(real, ?'a::type) cart ⇒ bool. FINITE c ∧ s = hull convex_cone c)*

thm POLYHEDRON_POLYTOPE_SUMS:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ polytope t ⟶ polyhedron (GSPEC (λGEN%PVAR%1404::(real, ?'a::type) cart. ∃ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1404 (IN x s ∧ IN y t) (vector_add x y)))

thm POLYHEDRON_AS_CONE_PLUS_CONV:

∀ s::(real, ?'a::type) cart ⇒ bool. polyhedron s = (∃ (t::(real, ?'a::type) cart ⇒ bool) u::(real, ?'a::type) cart ⇒ bool. FINITE t ∧ FINITE u ∧ s = GSPEC (λGEN%PVAR%1415::(real, ?'a::type) cart. ∃ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1415 (IN x (hull convex_cone t) ∧ IN y (hull convex u)) (vector_add x y)))

thm POLYHEDRON_LINEAR_IMAGE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. linear f ∧ polyhedron s ⟶ polyhedron (IMAGE f s)

thm POLYHEDRON_SUMS:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ polyhedron t ⟶ polyhedron (GSPEC (λGEN%PVAR%1421::(real, ?'a::type) cart. ∃ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1421 (IN x s ∧ IN y t) (vector_add x y)))

thm FARKAS_LEMMA:

∀ (A::((real, ?'b::type) cart, ?'a::type) cart) b::(real, ?'a::type) cart. (∃ x::(real, ?'b::type) cart. matrix_vector_mul A x = b ∧ (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ (0::real) ≤ $ x i)) = (¬ (∃ y::(real, ?'a::type) cart. dot b y < (0::real) ∧ (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ (0::real) ≤ $ (matrix_vector_mul (HOL_Light_Import.transp A) y) i)))

thm FARKAS_LEMMA_ALT:

∀ (A::((real, ?'b::type) cart, ?'a::type) cart) b::(real, ?'a::type) cart. (∃ x::(real, ?'b::type) cart. ∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ (matrix_vector_mul A x) i ≤ $ b i) = (¬ (∃ y::(real, ?'a::type) cart. (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ (0::real) ≤ $ y i) ∧ vector_matrix_mul y A = vec (0::nat) ∧ dot b y < (0::real)))

thm SEPARATING_HYPERPLANE_POLYHEDRA:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. polyhedron s ∧ polyhedron t ∧ s ≠ EMPTY ∧ t ≠ EMPTY ∧ DISJOINT s t ⟶ (∃ (a::(real,

?'a::type) cart) b::real. a ≠ vec (0::nat) ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ dot a x < b) ∧ (∀ x::(real, ?'a::type) cart. IN x t ⟶ b < dot a x))

thm RELATIVE_FRONTIER_OF_CONVEX_HULL:

∀ s::(real, ?'a::type) cart ⇒ bool. ¬ affine_dependent s ⟶ DIFF (hull convex s) (relative_interior (hull convex s)) = UNIONS (GSPEC (λGEN%PVAR%1428::(real, ?'a::type) cart ⇒ bool. ∃ a::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1428 (IN a s) (hull convex (DELETE s a))))

thm FRONTIER_OF_CONVEX_HULL:

∀ s::(real, ?'a::type) cart ⇒ bool. HAS_SIZE s (dimindex HOL_Light_Import.UNIV + (1::nat)) ⟶ frontier (hull convex s) = UNIONS (GSPEC (λGEN%PVAR%1429::(real, ?'a::type) cart ⇒ bool. ∃ a::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1429 (IN a s) (hull convex (DELETE s a))))

thm RELATIVE_FRONTIER_OF_TRIANGLE:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) c::(real, ?'a::type) cart. ¬ collinear (INSERT a (INSERT b (INSERT c EMPTY))) ⟶ DIFF (hull convex (INSERT a (INSERT b (INSERT c EMPTY)))) (relative_interior (hull convex (INSERT a (INSERT b (INSERT c EMPTY))))) = HOL_Light_Import.UNION (closed_segment [(a, b)]) (HOL_Light_Import.UNION (closed_segment [(b, c)]) (closed_segment [(c, a)]))

thm INTERIOR_SEGMENT_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. interior (open_segment (a, b)) = (if (2::nat) ≤ dimindex HOL_Light_Import.UNIV then EMPTY else open_segment (a, b))

thm INTERIOR_SEGMENT_conjunct0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. interior (closed_segment [(a, b)]) = (if (2::nat) ≤ dimindex HOL_Light_Import.UNIV then EMPTY else open_segment (a, b))

thm FRONTIER_OF_TRIANGLE:

∀ (a::(real, 2) cart) (b::(real, 2) cart) c::(real, 2) cart. frontier (hull convex (INSERT a (INSERT b (INSERT c EMPTY)))) = HOL_Light_Import.UNION (closed_segment [(a, b)]) (HOL_Light_Import.UNION (closed_segment [(b, c)]) (closed_segment [(c, a)]))

thm INSIDE_OF_TRIANGLE:

∀ (a::(real, 2) cart) (b::(real, 2) cart) c::(real, 2) cart. inside (HOL_Light_Import.UNION (closed_segment [(a, b)]) (HOL_Light_Import.UNION (closed_segment [(b, c)]) (closed_segment [(c, a)]))) = interior (hull convex (INSERT a (INSERT b (INSERT c EMPTY))))

thm INTERIOR_OF_TRIANGLE:

$\forall (a::(real, 2)\ cart)\ (b::(real, 2)\ cart)\ c::(real, 2)\ cart.\ interior\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))) = DIFF\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))))\ (HOL\_Light\_Import.UNION\ (closed\_segment\ [(a, b)])\ (HOL\_Light\_Import.UNION\ (closed\_segment\ [(b, c)])\ (closed\_segment\ [(c, a)])))$

thm POLYHEDRON_RIDGE_TWO_FACETS:

$\forall (p::(real, ?'a::type)\ cart \Rightarrow bool)\ r::(real, ?'a::type)\ cart \Rightarrow bool.\ polyhedron\ p \wedge face\_of\ r\ p \wedge r \neq EMPTY \wedge aff\_dim\ r = aff\_dim\ p - int\ (2::nat) \longrightarrow (\exists (f1::(real, ?'a::type)\ cart \Rightarrow bool)\ f2::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f1\ p \wedge aff\_dim\ f1 = aff\_dim\ p - int\ (1::nat) \wedge face\_of\ f2\ p \wedge aff\_dim\ f2 = aff\_dim\ p - int\ (1::nat) \wedge f1 \neq f2 \wedge SUBSET\ r\ f1 \wedge SUBSET\ r\ f2 \wedge HOL\_Light\_Import.INTER\ f1\ f2 = r \wedge (\forall f::(real, ?'a::type)\ cart \Rightarrow bool.\ face\_of\ f\ p \wedge aff\_dim\ f = aff\_dim\ p - int\ (1::nat) \wedge SUBSET\ r\ f \longrightarrow f = f1 \vee f = f2))$

thm POLYTOPE_VERTEX_LOWER_BOUND:

$\forall p::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \longrightarrow aff\_dim\ p + int\ (1::nat) \leq int\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1454::(real, ?'a::type)\ cart.\ \exists v::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1454\ (extreme\_point\_of\ v\ p)\ v)))$

thm POLYTOPE_FACET_LOWER_BOUND:

$\forall p::(real, ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \wedge aff\_dim\ p \neq int\ (0::nat) \longrightarrow aff\_dim\ p + int\ (1::nat) \leq int\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1459::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1459\ (facet\_of\ f\ p)\ f)))$

thm DEF_simplex:

$simplex = (\lambda(\_696080::int)\ \_696081::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists c::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ c \wedge int\ (CARD\ c) = \_696080 + int\ (1::nat) \wedge \_696081 = hull\ convex\ c)$

thm simplex:

$\forall (n::int)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s = (\exists c::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ c \wedge int\ (CARD\ c) = n + int\ (1::nat) \wedge s = hull\ convex\ c)$

thm SIMPLEX:

$simplex\ (?n::int)\ (?s::(real, ?'a::type)\ cart \Rightarrow bool) = (\exists c::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ c \wedge \neg\ affine\_dependent\ c \wedge int\ (CARD\ c) = ?n + int\ (1::nat) \wedge ?s = hull\ convex\ c)$

thm CONVEX_SIMPLEX:

$\forall (n::int)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow convex\ s$

thm COMPACT_SIMPLEX:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow compact\ s$

thm SIMPLEX_IMP_POLYTOPE:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow polytope\ s$

thm SIMPLEX_DIM_GE:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow -\ int\ (1::nat) \leq n$

thm SIMPLEX_EMPTY:

$\forall n::int.\ simplex\ n\ EMPTY = (n = -\ int\ (1::nat))$

thm SIMPLEX_MINUS_1:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ (-\ int\ (1::nat))\ s = (s = EMPTY)$

thm AFF_DIM_SIMPLEX:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ n::int.\ simplex\ n\ s \longrightarrow aff\_dim\ s = n$

thm SIMPLEX_EXTREME_POINTS:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1462::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1462\ (extreme\_point\_of\ v\ s)\ v)) \wedge \neg\ affine\_dependent\ (GSPEC\ (\lambda GEN\%PVAR\%1463::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1463\ (extreme\_point\_of\ v\ s)\ v)) \wedge int\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1464::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1464\ (extreme\_point\_of\ v\ s)\ v))) = n + int\ (1::nat) \wedge s = hull\ convex\ (GSPEC\ (\lambda GEN\%PVAR\%1465::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1465\ (extreme\_point\_of\ v\ s)\ v))$

thm SIMPLEX_FACE_OF_SIMPLEX:

$\forall (n::int)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \wedge face\_of\ f\ s \longrightarrow (\exists m \leq n.\ simplex\ m\ f)$

thm FACE_OF_SIMPLEX_SUBSET:

$\forall (n::int)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \wedge face\_of\ f\ s \longrightarrow (\exists c::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ c\ (GSPEC\ (\lambda GEN\%PVAR\%1467::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1467\ (extreme\_point\_of\ x\ s)\ x)) \wedge f = hull\ convex\ c)$

thm SUBSET_FACE_OF_SIMPLEX:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (n::int)\ c::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \wedge SUBSET\ c\ (GSPEC\ (\lambda GEN\%PVAR\%1470::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1470\ (extreme\_point\_of\ x\ s)\ x)) \longrightarrow face\_of\ (hull\ convex\ c)\ s$

thm FACES_OF_SIMPLEX:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%1471::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1471\ (face\_of\ f\ s)\ f) = GSPEC\ (\lambda GEN\%PVAR\%1473::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists c::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1473\ (SUBSET\ c\ (GSPEC\ (\lambda GEN\%PVAR\%1472::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1472\ (extreme\_point\_of\ v\ s)\ v)))\ (hull\ convex\ c))$

thm HAS_SIZE_FACES_OF_SIMPLEX:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%1476::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1476\ (face\_of\ f\ s)\ f))\ (2::nat)^{num\_of\_int\ (n\ +\ int\ (1::nat))}$

thm FINITE_FACES_OF_SIMPLEX:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1477::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1477\ (face\_of\ f\ s)\ f))$

thm CARD_FACES_OF_SIMPLEX:

$\forall (n::int)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ simplex\ n\ s \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1478::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1478\ (face\_of\ f\ s)\ f)) = (2::nat)^{num\_of\_int\ (n\ +\ int\ (1::nat))}$

thm DEF_simplicial_complex:

$simplicial\_complex = (\lambda\_697136::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ \_697136 \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ \_697136 \longrightarrow (\exists n::int.\ simplex\ n\ s)) \wedge (\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ \_697136 \wedge face\_of\ f\ s \longrightarrow IN\ f\ \_697136) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ \_697136 \wedge IN\ s'\ \_697136 \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ s')\ s \wedge face\_of\ (HOL\_Light\_Import.INTER\ s\ s')\ s'))$

thm simplicial_complex:

$\forall c::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ simplicial\_complex\ c = (FINITE\ c \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ c \longrightarrow (\exists n::int.\ simplex\ n\ s)) \wedge (\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ c \wedge face\_of\ f\ s \longrightarrow IN\ f\ c) \wedge (\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ s'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ c \wedge IN\ s'\ c \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ s\ s')\ s \wedge face\_of\ (HOL\_Light\_Import.INTER\ s\ s')\ s'))$

thm DEF_triangulation:

$triangulation = (\lambda\_697141::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ \_697141 \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ \_697141 \longrightarrow (\exists n::int.\ simplex\ n\ t)) \wedge (\forall (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ \_697141 \wedge IN\ t'\ \_697141 \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ t\ t')\ t \wedge face\_of\ (HOL\_Light\_Import.INTER\ t\ t')\ t'))$

thm triangulation:

$\forall\, tr::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ triangulation\ tr = (FINITE\ tr \wedge (\forall\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ tr \longrightarrow (\exists\, n::int.\ simplex\ n\ t)) \wedge (\forall\, (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ tr \wedge IN\ t'\ tr \longrightarrow face\_of\ (HOL\_Light\_Import.INTER\ t\ t')\ t \wedge face\_of\ (HOL\_Light\_Import.INTER\ t\ t')\ t'))$

thm TRIANGULATION_INTER_SIMPLEX:

$\forall\, (tr::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ triangulation\ tr \wedge IN\ t\ tr \wedge IN\ t'\ tr \longrightarrow HOL\_Light\_Import.INTER\ t\ t' = hull\ convex\ (HOL\_Light\_Import.INTER\ (GSPEC\ (\lambda GEN\%PVAR\%1480::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1480\ (extreme\_point\_of\ x\ t)\ x))\ (GSPEC\ (\lambda GEN\%PVAR\%1481::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1481\ (extreme\_point\_of\ x\ t')\ x)))$

thm TRIANGULATION_SIMPLICIAL_COMPLEX:

$\forall\, tr::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ triangulation\ tr \longrightarrow simplicial\_complex\ (GSPEC\ (\lambda GEN\%PVAR\%1486::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1486\ (\exists\, t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ tr \wedge face\_of\ f\ t)\ f))$

thm DEF_binom:

$binom = (SOME\ binom::nat \Rightarrow nat \times nat \Rightarrow nat.\ \forall\, \_698166::nat.\ (\forall\, n::nat.\ binom\ \_698166\ (n,\ 0::nat) = (1::nat)) \wedge (\forall\, k::nat.\ binom\ \_698166\ (0::nat,\ Suc\ k) = (0::nat)) \wedge (\forall\, (n::nat)\ k::nat.\ binom\ \_698166\ (Suc\ n,\ Suc\ k) = binom\ \_698166\ (n,\ Suc\ k) + binom\ \_698166\ (n,\ k)))\ (52::nat)$

thm binom:

$(\forall\, n::nat.\ binom\ (n,\ 0::nat) = (1::nat)) \wedge (\forall\, k::nat.\ binom\ (0::nat,\ Suc\ k) = (0::nat)) \wedge (\forall\, (n::nat)\ k::nat.\ binom\ (Suc\ n,\ Suc\ k) = binom\ (n,\ Suc\ k) + binom\ (n,\ k))$

thm binom_conjunct2:

$\forall\, (n::nat)\ k::nat.\ binom\ (Suc\ n,\ Suc\ k) = binom\ (n,\ Suc\ k) + binom\ (n,\ k)$

thm binom_conjunct1:

$\forall\, k::nat.\ binom\ (0::nat,\ Suc\ k) = (0::nat)$

thm binom_conjunct0:

$\forall\, n::nat.\ binom\ (n,\ 0::nat) = (1::nat)$

thm BINOM_LT:

$\forall\, (n::nat)\ k::nat.\ n < k \longrightarrow binom\ (n,\ k) = (0::nat)$

thm BINOM_REFL:

$\forall\, n{::}nat.\ binom\ (n,\ n) = (1{::}nat)$

thm BINOM_1:

$\forall\, n{::}nat.\ binom\ (n,\ 1{::}nat) = n$

thm BINOM_FACT:

$\forall\, (n{::}nat)\ k{::}nat.\ fact\ n * (fact\ k * binom\ (n + k,\ k)) = fact\ (n + k)$

thm BINOM_EQ_0:

$\forall\, (n{::}nat)\ k{::}nat.\ (binom\ (n,\ k) = (0{::}nat)) = (n < k)$

thm BINOM_PENULT:

$\forall\, n{::}nat.\ binom\ (Suc\ n,\ n) = Suc\ n$

thm BINOM_TOP_STEP:

$\forall\, (n{::}nat)\ k{::}nat.\ (n + (1{::}nat) - k) * binom\ (n + (1{::}nat),\ k) = (n + (1{::}nat)) * binom\ (n,\ k)$

thm BINOM_BOTTOM_STEP:

$\forall\, (n{::}nat)\ k{::}nat.\ (k + (1{::}nat)) * binom\ (n,\ k + (1{::}nat)) = (n - k) * binom\ (n,\ k)$

thm BINOMIAL_THEOREM:

$\forall\, (n{::}nat)\ (x{::}nat)\ y{::}nat.\ (x + y)^n = nsum\ (dotdot\ (0{::}nat)\ n)\ (\lambda k{::}nat.\ binom\ (n,\ k) * (x^k * y^{n - k}))$

thm REAL_BINOMIAL_THEOREM:

$\forall\, (n{::}nat)\ (x{::}real)\ y{::}real.\ (x + y)^n = sum\ (dotdot\ (0{::}nat)\ n)\ (\lambda k{::}nat.\ real\_of\_nat\ (binom\ (n,\ k)) * (x^k * y^{n - k}))$

thm BINOM_TOP_STEP_REAL:

$\forall\, (n{::}nat)\ k{::}nat.\ real\_of\_nat\ (binom\ (n + (1{::}nat),\ k)) = (if\ k = n + (1{::}nat)\ then\ 1{::}real\ else\ (real\_of\_nat\ n + (1{::}real))\ /\ (real\_of\_nat\ n + ((1{::}real) - real\_of\_nat\ k)) * real\_of\_nat\ (binom\ (n,\ k)))$

thm BINOM_BOTTOM_STEP_REAL:

$\forall\, (n{::}nat)\ k{::}nat.\ real\_of\_nat\ (binom\ (n,\ k + (1{::}nat))) = (real\_of\_nat\ n - real\_of\_nat\ k)\ /\ (real\_of\_nat\ k + (1{::}real)) * real\_of\_nat\ (binom\ (n,\ k))$

thm REAL_OF_NUM_BINOM:

$\forall\, (n{::}nat)\ k{::}nat.\ real\_of\_nat\ (binom\ (n,\ k)) = (if\ k \le n\ then\ real\_of\_nat\ (fact\ n)\ /\ (real\_of\_nat\ (fact\ (n - k)) * real\_of\_nat\ (fact\ k))\ else\ (0{::}real))$

thm BINOM_BOTH_STEP_REAL:

$\forall\, (p{::}nat)\ k{::}nat.\ real\_of\_nat\ (binom\ (p + (1{::}nat),\ k + (1{::}nat))) = (real\_of\_nat\ p + (1{::}real))\ /\ (real\_of\_nat\ k + (1{::}real)) * real\_of\_nat\ (binom\ (p,\ k))$

thm BINOM_BOTH_STEP:

$\forall (p::nat)\ k::nat.\ (k + (1::nat)) * binom\ (p + (1::nat),\ k + (1::nat)) = (p + (1::nat)) * binom\ (p,\ k)$

thm BINOM_BOTH_STEP_DOWN:

$\forall (p::nat)\ k::nat.\ (k = (0::nat) \longrightarrow p = (0::nat)) \longrightarrow k * binom\ (p,\ k) = p * binom\ (p - (1::nat),\ k - (1::nat))$

thm BINOM:

$\forall (n::nat)\ k::nat.\ binom\ (n,\ k) = (if\ k \leq n\ then\ fact\ n\ div\ (fact\ (n - k) * fact\ k)\ else\ (0::nat))$

thm BROUWER_COMPACTNESS_LEMMA:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ compact\ s \wedge continuous\_on\ f\ s \wedge \neg (\exists x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge f\ x = vec\ (0::nat)) \longrightarrow (\exists d>0::real.\ \forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow d \leq vector\_norm\ (f\ x))$

thm KUHN_LABELLING_LEMMA:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (P::(real,\ ?'a::type)\ cart \Rightarrow bool)\ Q::nat \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ P\ x \longrightarrow P\ (f\ x)) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ P\ x \longrightarrow (\forall i::nat.\ Q\ i \longrightarrow (0::real) \leq \$\ x\ i \wedge \$\ x\ i \leq (1::real))) \longrightarrow (\exists l::(real,\ ?'a::type)\ cart \Rightarrow nat \Rightarrow nat.\ (\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ l\ x\ i \leq (1::nat)) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ P\ x \wedge Q\ i \wedge \$\ x\ i = (0::real) \longrightarrow l\ x\ i = (0::nat)) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ P\ x \wedge Q\ i \wedge \$\ x\ i = (1::real) \longrightarrow l\ x\ i = (1::nat)) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ P\ x \wedge Q\ i \wedge l\ x\ i = (0::nat) \longrightarrow \$\ x\ i \leq \$\ (f\ x)\ i) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ i::nat.\ P\ x \wedge Q\ i \wedge l\ x\ i = (1::nat) \longrightarrow \$\ (f\ x)\ i \leq \$\ x\ i))$

thm KUHN_COUNTING_LEMMA:

$\forall (face::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (faces::?'b::type \Rightarrow bool)\ (simplices::?'a::type \Rightarrow bool)\ (comp::?'a::type \Rightarrow bool)\ (comp'::?'b::type \Rightarrow bool)\ bnd::?'b::type \Rightarrow bool.\ FINITE\ faces \wedge FINITE\ simplices \wedge (\forall f::?'b::type.\ IN\ f\ faces \wedge bnd\ f \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1517::?'a::type.\ \exists s::?'a::type.\ SETSPEC\ GEN\%PVAR\%1517\ (IN\ s\ simplices \wedge face\ f\ s)\ s)) = (1::nat)) \wedge (\forall f::?'b::type.\ IN\ f\ faces \wedge \neg\ bnd\ f \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1518::?'a::type.\ \exists s::?'a::type.\ SETSPEC\ GEN\%PVAR\%1518\ (IN\ s\ simplices \wedge face\ f\ s)\ s)) = (2::nat)) \wedge (\forall s::?'a::type.\ IN\ s\ simplices \wedge comp\ s \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1519::?'b::type.\ \exists f::?'b::type.\ SETSPEC\ GEN\%PVAR\%1519\ (IN\ f\ faces \wedge face\ f\ s \wedge comp'\ f)\ f)) = (1::nat)) \wedge (\forall s::?'a::type.\ IN\ s\ simplices \wedge \neg\ comp\ s \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1520::?'b::type.\ \exists f::?'b::type.\ SETSPEC\ GEN\%PVAR\%1520\ (IN\ f\ faces \wedge face\ f\ s \wedge comp'\ f)\ f)) = (0::nat) \vee CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1521::?'b::type.\ \exists f::?'b::type.\ SETSPEC\ GEN\%PVAR\%1521\ (IN\ f\ faces \wedge face\ f\ s \wedge comp'\ f)\ f)) = (2::nat)) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1522::?'b::type.\ \exists f::?'b::type.$

SETSPEC GEN%PVAR%1522 (IN f faces ∧ comp′ f ∧ bnd f) f))) ⟶ ODD (CARD (GSPEC (λGEN%PVAR%1523::?′a::type. ∃ s::?′a::type. SETSPEC GEN%PVAR%1523 (IN s simplices ∧ comp s) s)))

thm HAS_SIZE_1_EXISTS:

∀ s::?′a::type ⇒ bool. HAS_SIZE s (1::nat) = (∃!x::?′a::type. IN x s)

thm HAS_SIZE_2_EXISTS:

∀ s::?′a::type ⇒ bool. HAS_SIZE s (2::nat) = (∃ (x::?′a::type) y::?′a::type. x ≠ y ∧ (∀ z::?′a::type. IN z s = (z = x ∨ z = y)))

thm IMAGE_LEMMA_0:

∀ (f::?′b::type ⇒ ?′a::type) (s::?′b::type ⇒ bool) n::nat. HAS_SIZE (GSPEC (λGEN%PVAR%1526::?′b::type. ∃ a::?′b::type. SETSPEC GEN%PVAR%1526 (IN a s ∧ IMAGE f (DELETE s a) = DELETE (?t::?′a::type ⇒ bool) (?b::?′a::type)) a)) n ⟶ HAS_SIZE (GSPEC (λGEN%PVAR%1527::?′b::type ⇒ bool. ∃ s′::?′b::type ⇒ bool. SETSPEC GEN%PVAR%1527 (∃ a::?′b::type. IN a s ∧ s′ = DELETE s a ∧ IMAGE f s′ = DELETE ?t ?b) s′)) n

thm IMAGE_LEMMA_1:

∀ (f::?′b::type ⇒ ?′a::type) (s::?′b::type ⇒ bool) (t::?′a::type ⇒ bool) b::?′a::type. FINITE s ∧ FINITE t ∧ CARD s = CARD t ∧ IMAGE f s = t ∧ IN b t ⟶ CARD (GSPEC (λGEN%PVAR%1528::?′b::type ⇒ bool. ∃ s′::?′b::type ⇒ bool. SETSPEC GEN%PVAR%1528 (∃ a::?′b::type. IN a s ∧ s′ = DELETE s a ∧ IMAGE f s′ = DELETE t b) s′)) = (1::nat)

thm IMAGE_LEMMA_2:

∀ (f::?′b::type ⇒ ?′a::type) (s::?′b::type ⇒ bool) (t::?′a::type ⇒ bool) b::?′a::type. FINITE s ∧ FINITE t ∧ CARD s = CARD t ∧ SUBSET (IMAGE f s) t ∧ IMAGE f s ≠ t ∧ IN b t ⟶ CARD (GSPEC (λGEN%PVAR%1530::?′b::type ⇒ bool. ∃ s′::?′b::type ⇒ bool. SETSPEC GEN%PVAR%1530 (∃ a::?′b::type. IN a s ∧ s′ = DELETE s a ∧ IMAGE f s′ = DELETE t b) s′)) = (0::nat) ∨ CARD (GSPEC (λGEN%PVAR%1531::?′b::type ⇒ bool. ∃ s′::?′b::type ⇒ bool. SETSPEC GEN%PVAR%1531 (∃ a::?′b::type. IN a s ∧ s′ = DELETE s a ∧ IMAGE f s′ = DELETE t b) s′)) = (2::nat)

thm KUHN_COMPLETE_LEMMA:

∀ (face::(?′a::type ⇒ bool) ⇒ (?′a::type ⇒ bool) ⇒ bool) (simplices::(?′a::type ⇒ bool) ⇒ bool) (rl::?′a::type ⇒ nat) (bnd::(?′a::type ⇒ bool) ⇒ bool) n::nat. FINITE simplices ∧ (∀ (f::?′a::type ⇒ bool) s::?′a::type ⇒ bool. face f s = (∃ a::?′a::type. IN a s ∧ f = DELETE s a)) ∧ (∀ s::?′a::type ⇒ bool. IN s simplices ⟶ HAS_SIZE s (n + (2::nat)) ∧ SUBSET (IMAGE rl s) (dotdot (0::nat) (n + (1::nat)))) ∧ (∀ f::?′a::type ⇒ bool. IN f (GSPEC (λGEN%PVAR%1536::?′a::type ⇒ bool. ∃ f::?′a::type ⇒ bool. SETSPEC GEN%PVAR%1536 (∃ s::?′a::type ⇒ bool. IN s simplices ∧ face f s) f)) ∧ bnd f ⟶ CARD (GSPEC (λGEN%PVAR%1537::?′a::type ⇒ bool. ∃ s::?′a::type ⇒ bool. SETSPEC GEN%PVAR%1537 (IN s simplices ∧

$face\, f\, s)\ s)) = (1{::}nat)) \wedge (\forall f{::}?'a{::}type \Rightarrow bool.\ IN\, f\ (GSPEC\ (\lambda GEN\%PVAR\%1538{::}?'a{::}type \Rightarrow bool.\ \exists f{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1538\ (\exists s{::}?'a{::}type \Rightarrow bool.\ IN\, s\ simplices \wedge face\, f\, s)\ f)) \wedge \neg\ bnd\, f \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1539{::}?'a{::}type \Rightarrow bool.\ \exists s{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1539\ (IN\, s\ simplices \wedge face\, f\, s)\ s)) = (2{::}nat)) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1541{::}?'a{::}type \Rightarrow bool.\ \exists f{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1541\ (IN\ f\ (GSPEC\ (\lambda GEN\%PVAR\%1540{::}?'a{::}type \Rightarrow bool.\ \exists f{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1540\ (\exists s{::}?'a{::}type \Rightarrow bool.\ IN\ s\ simplices \wedge face\ f\ s)\ f)) \wedge IMAGE\ rl\ f = dotdot\ (0{::}nat)\ n \wedge bnd\, f)\ f))) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1542{::}?'a{::}type \Rightarrow bool.\ \exists s{::}?'a{::}type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1542\ (IN\ s\ simplices \wedge IMAGE\ rl\ s = dotdot\ (0{::}nat)\ (n + (1{::}nat)))\ s)))$

thm DEF_kle:

$kle = (\lambda(\_702142{::}nat)\ (\_702143{::}nat \Rightarrow nat)\ \_702144{::}nat \Rightarrow nat.\ \exists k{::}nat \Rightarrow bool.\ SUBSET\ k\ (dotdot\ (1{::}nat)\ \_702142) \wedge (\forall j{::}nat.\ \_702144\ j = \_702143\ j + (if\ IN\ j\ k\ then\ 1{::}nat\ else\ (0{::}nat))))$

thm kle:

$\forall (n{::}nat)\ (y{::}nat \Rightarrow nat)\ x{::}nat \Rightarrow nat.\ kle\ n\ x\ y = (\exists k{::}nat \Rightarrow bool.\ SUBSET\ k\ (dotdot\ (1{::}nat)\ n) \wedge (\forall j{::}nat.\ y\ j = x\ j + (if\ IN\ j\ k\ then\ 1{::}nat\ else\ (0{::}nat))))$

thm KLE_REFL:

$\forall (n{::}nat)\ x{::}nat \Rightarrow nat.\ kle\ n\ x\ x$

thm KLE_ANTISYM:

$\forall (n{::}nat)\ (x{::}nat \Rightarrow nat)\ y{::}nat \Rightarrow nat.\ (kle\ n\ x\ y \wedge kle\ n\ y\ x) = (x = y)$

thm POINTWISE_MINIMAL:

$\forall s{::}(nat \Rightarrow nat) \Rightarrow bool.\ FINITE\ s \longrightarrow s \neq EMPTY \wedge (\forall (x{::}nat \Rightarrow nat)\ y{::}nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow (\forall j{::}nat.\ x\ j \leq y\ j) \vee (\forall j{::}nat.\ y\ j \leq x\ j)) \longrightarrow (\exists a{::}nat \Rightarrow nat.\ IN\ a\ s \wedge (\forall x{::}nat \Rightarrow nat.\ IN\ x\ s \longrightarrow (\forall j{::}nat.\ a\ j \leq x\ j)))$

thm POINTWISE_MAXIMAL:

$\forall s{::}(nat \Rightarrow nat) \Rightarrow bool.\ FINITE\ s \longrightarrow s \neq EMPTY \wedge (\forall (x{::}nat \Rightarrow nat)\ y{::}nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow (\forall j{::}nat.\ x\ j \leq y\ j) \vee (\forall j{::}nat.\ y\ j \leq x\ j)) \longrightarrow (\exists a{::}nat \Rightarrow nat.\ IN\ a\ s \wedge (\forall x{::}nat \Rightarrow nat.\ IN\ x\ s \longrightarrow (\forall j{::}nat.\ x\ j \leq a\ j)))$

thm KLE_IMP_POINTWISE:

$\forall (n{::}nat)\ (x{::}nat \Rightarrow nat)\ y{::}nat \Rightarrow nat.\ kle\ n\ x\ y \longrightarrow (\forall j{::}nat.\ x\ j \leq y\ j)$

thm POINTWISE_ANTISYM:

$\forall (x{::}nat \Rightarrow nat)\ y{::}nat \Rightarrow nat.\ ((\forall j{::}nat.\ x\ j \leq y\ j) \wedge (\forall j{::}nat.\ y\ j \leq x\ j)) = (x = y)$

thm KLE_TRANS:

$\forall (x::nat \Rightarrow nat)\ (y::nat \Rightarrow nat)\ (z::nat \Rightarrow nat)\ n::nat.\ kle\ n\ x\ y \wedge kle\ n\ y\ z \wedge (kle\ n\ x\ z \vee kle\ n\ z\ x) \longrightarrow kle\ n\ x\ z$

thm KLE_STRICT:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ kle\ n\ x\ y \wedge x \neq y \longrightarrow (\forall j::nat.\ x\ j \leq y\ j) \wedge (\exists k \geq 1::nat.\ k \leq n \wedge x\ k < y\ k)$

thm KLE_MINIMAL:

$\forall (s::(nat \Rightarrow nat) \Rightarrow bool)\ n::nat.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow kle\ n\ x\ y \vee kle\ n\ y\ x) \longrightarrow (\exists a::nat \Rightarrow nat.\ IN\ a\ s \wedge (\forall x::nat \Rightarrow nat.\ IN\ x\ s \longrightarrow kle\ n\ a\ x))$

thm KLE_MAXIMAL:

$\forall (s::(nat \Rightarrow nat) \Rightarrow bool)\ n::nat.\ FINITE\ s \wedge s \neq EMPTY \wedge (\forall (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow kle\ n\ x\ y \vee kle\ n\ y\ x) \longrightarrow (\exists a::nat \Rightarrow nat.\ IN\ a\ s \wedge (\forall x::nat \Rightarrow nat.\ IN\ x\ s \longrightarrow kle\ n\ x\ a))$

thm KLE_STRICT_SET:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ kle\ n\ x\ y \wedge x \neq y \longrightarrow (1::nat) \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1543::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1543\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < y\ k)\ k))$

thm KLE_RANGE_COMBINE:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ (y::nat \Rightarrow nat)\ (m1::nat)\ m2::nat.\ kle\ n\ x\ y \wedge kle\ n\ y\ (?z::nat \Rightarrow nat) \wedge (kle\ n\ x\ ?z \vee kle\ n\ ?z\ x) \wedge m1 \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1546::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1546\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < y\ k)\ k)) \wedge m2 \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1547::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1547\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge y\ k < ?z\ k)\ k)) \longrightarrow kle\ n\ x\ ?z \wedge m1 + m2 \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1548::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1548\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < ?z\ k)\ k))$

thm KLE_RANGE_COMBINE_L:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ (y::nat \Rightarrow nat)\ m::nat.\ kle\ n\ x\ y \wedge kle\ n\ y\ (?z::nat \Rightarrow nat) \wedge (kle\ n\ x\ ?z \vee kle\ n\ ?z\ x) \wedge m \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1550::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1550\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge y\ k < ?z\ k)\ k)) \longrightarrow kle\ n\ x\ ?z \wedge m \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1551::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1551\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < ?z\ k)\ k))$

thm KLE_RANGE_COMBINE_R:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ (y::nat \Rightarrow nat)\ m::nat.\ kle\ n\ x\ y \wedge kle\ n\ y\ (?z::nat \Rightarrow nat) \wedge (kle\ n\ x\ ?z \vee kle\ n\ ?z\ x) \wedge m \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1553::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1553\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < y\ k)\ k)) \longrightarrow kle\ n\ x\ ?z \wedge m \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1554::nat.$

$\exists k::nat.\ SETSPEC\ GEN\%PVAR\%1554\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < ?z\ k)\ k))$

thm KLE_RANGE_INDUCT:

$\forall (n::nat)\ (m::nat)\ s::(nat \Rightarrow nat) \Rightarrow bool.\ HAS\_SIZE\ s\ (Suc\ m) \longrightarrow (\forall (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow kle\ n\ x\ y \vee kle\ n\ y\ x) \longrightarrow (\exists (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ IN\ x\ s \wedge IN\ y\ s \wedge kle\ n\ x\ y \wedge m \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1555::nat.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%1555\ (IN\ k\ (dotdot\ (1::nat)\ n) \wedge x\ k < y\ k)\ k)))$

thm KLE_SUC:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ kle\ n\ x\ y \longrightarrow kle\ (n + (1::nat))\ x\ y$

thm KLE_TRANS_1:

$\forall (n::nat)\ (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ kle\ n\ x\ y \longrightarrow (\forall j::nat.\ x\ j \leq y\ j \wedge y\ j \leq x\ j + (1::nat))$

thm KLE_TRANS_2:

$\forall (a::nat \Rightarrow nat)\ (b::nat \Rightarrow nat)\ c::nat \Rightarrow nat.\ kle\ (?n::nat)\ a\ b \wedge kle\ ?n\ b\ c \wedge (\forall j::nat.\ c\ j \leq a\ j + (1::nat)) \longrightarrow kle\ ?n\ a\ c$

thm KLE_BETWEEN_R:

$\forall (a::nat \Rightarrow nat)\ (b::nat \Rightarrow nat)\ (c::nat \Rightarrow nat)\ x::nat \Rightarrow nat.\ kle\ (?n::nat)\ a\ b \wedge kle\ ?n\ b\ c \wedge kle\ ?n\ a\ x \wedge kle\ ?n\ c\ x \longrightarrow kle\ ?n\ b\ x$

thm KLE_BETWEEN_L:

$\forall (a::nat \Rightarrow nat)\ (b::nat \Rightarrow nat)\ (c::nat \Rightarrow nat)\ x::nat \Rightarrow nat.\ kle\ (?n::nat)\ a\ b \wedge kle\ ?n\ b\ c \wedge kle\ ?n\ x\ a \wedge kle\ ?n\ x\ c \longrightarrow kle\ ?n\ x\ b$

thm KLE_ADJACENT:

$\forall (a::nat \Rightarrow nat)\ (b::nat \Rightarrow nat)\ (x::nat \Rightarrow nat)\ k::nat.\ (1::nat) \leq k \wedge k \leq (?n::nat) \wedge (\forall j::nat.\ b\ j = (if\ j = k\ then\ a\ j + (1::nat)\ else\ a\ j)) \wedge kle\ ?n\ a\ x \wedge kle\ ?n\ x\ b \longrightarrow x = a \vee x = b$

thm DEF_ksimplex:

$ksimplex = (\lambda(\_704627::nat)\ (\_704628::nat)\ \_704629::(nat \Rightarrow nat) \Rightarrow bool.\ HAS\_SIZE\ \_704629\ (\_704628 + (1::nat)) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ IN\ x\ \_704629 \longrightarrow x\ j \leq \_704627) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ IN\ x\ \_704629 \wedge \neg ((1::nat) \leq j \wedge j \leq \_704628) \longrightarrow x\ j = \_704627) \wedge (\forall (x::nat \Rightarrow nat)\ y::nat \Rightarrow nat.\ IN\ x\ \_704629 \wedge IN\ y\ \_704629 \longrightarrow kle\ \_704628\ x\ y \vee kle\ \_704628\ y\ x))$

thm ksimplex:

$\forall (p::nat)\ (s::(nat \Rightarrow nat) \Rightarrow bool)\ n::nat.\ ksimplex\ p\ n\ s = (HAS\_SIZE\ s\ (n + (1::nat)) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ IN\ x\ s \longrightarrow x\ j \leq p) \wedge (\forall (x::nat \Rightarrow$

nat) *j*::*nat*. *IN x s* ∧ ¬ ((*1*::*nat*) ≤ *j* ∧ *j* ≤ *n*) ⟶ *x j* = *p*) ∧ (∀ (*x*::*nat* ⇒ *nat*) *y*::*nat* ⇒ *nat*. *IN x s* ∧ *IN y s* ⟶ *kle n x y* ∨ *kle n y x*))

thm KSIMPLEX_EXTREMA:

∀ (*p*::*nat*) (*n*::*nat*) *s*::(*nat* ⇒ *nat*) ⇒ *bool*. *ksimplex p n s* ⟶ (∃ (*a*::*nat* ⇒ *nat*) *b*::*nat* ⇒ *nat*. *IN a s* ∧ *IN b s* ∧ (∀ *x*::*nat* ⇒ *nat*. *IN x s* ⟶ *kle n a x* ∧ *kle n x b*) ∧ (∀ *i*::*nat*. *b i* = (*if* (*1*::*nat*) ≤ *i* ∧ *i* ≤ *n then a i* + (*1*::*nat*) *else a i*)))

thm KSIMPLEX_EXTREMA_STRONG:

∀ (*p*::*nat*) (*n*::*nat*) *s*::(*nat* ⇒ *nat*) ⇒ *bool*. *ksimplex p n s* ∧ *n* ≠ (*0*::*nat*) ⟶ (∃ (*a*::*nat* ⇒ *nat*) *b*::*nat* ⇒ *nat*. *IN a s* ∧ *IN b s* ∧ *a* ≠ *b* ∧ (∀ *x*::*nat* ⇒ *nat*. *IN x s* ⟶ *kle n a x* ∧ *kle n x b*) ∧ (∀ *i*::*nat*. *b i* = (*if* (*1*::*nat*) ≤ *i* ∧ *i* ≤ *n then a i* + (*1*::*nat*) *else a i*)))

thm KSIMPLEX_SUCCESSOR:

∀ (*a*::*nat* ⇒ *nat*) (*p*::*nat*) (*n*::*nat*) *s*::(*nat* ⇒ *nat*) ⇒ *bool*. *ksimplex p n s* ∧ *IN a s* ⟶ (∀ *x*::*nat* ⇒ *nat*. *IN x s* ⟶ *kle n x a*) ∨ (∃ *y*::*nat* ⇒ *nat*. *IN y s* ∧ (∃ *k*≥*1*::*nat*. *k* ≤ *n* ∧ (∀ *j*::*nat*. *y j* = (*if j* = *k then a j* + (*1*::*nat*) *else a j*))))

thm KSIMPLEX_PREDECESSOR:

∀ (*a*::*nat* ⇒ *nat*) (*p*::*nat*) (*n*::*nat*) *s*::(*nat* ⇒ *nat*) ⇒ *bool*. *ksimplex p n s* ∧ *IN a s* ⟶ (∀ *x*::*nat* ⇒ *nat*. *IN x s* ⟶ *kle n a x*) ∨ (∃ *y*::*nat* ⇒ *nat*. *IN y s* ∧ (∃ *k*≥*1*::*nat*. *k* ≤ *n* ∧ (∀ *j*::*nat*. *a j* = (*if j* = *k then y j* + (*1*::*nat*) *else y j*))))

thm FINITE_SIMPLICES:

∀ (*p*::*nat*) *n*::*nat*. *FINITE* (*GSPEC* (λ*GEN%PVAR%1581*::(*nat* ⇒ *nat*) ⇒ *bool*. ∃ *s*::(*nat* ⇒ *nat*) ⇒ *bool*. *SETSPEC GEN%PVAR%1581* (*ksimplex p n s*) *s*))

thm SIMPLEX_TOP_FACE:

(*0*::*nat*) < (*?p*::*nat*) ∧ (∀ *x*::*nat* ⇒ *nat*. *IN x* (*?f*::(*nat* ⇒ *nat*) ⇒ *bool*) ⟶ *x* ((*?n*::*nat*) + (*1*::*nat*)) = *?p*) ⟶ (∃ (*s*::(*nat* ⇒ *nat*) ⇒ *bool*) *a*::*nat* ⇒ *nat*. *ksimplex ?p* (*?n* + (*1*::*nat*)) *s* ∧ *IN a s* ∧ *?f* = *DELETE s a*) = *ksimplex ?p ?n ?f*

thm KSIMPLEX_FIX_PLANE:

∀ (*p*::*nat*) (*q*::*nat*) (*n*::*nat*) (*j*::*nat*) (*s*::(*nat* ⇒ *nat*) ⇒ *bool*) (*a*::*nat* ⇒ *nat*) (*a0*::*nat* ⇒ *nat*) *a1*::*nat* ⇒ *nat*. *ksimplex p n s* ∧ *IN a s* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *n* ∧ (∀ *x*::*nat* ⇒ *nat*. *IN x* (*DELETE s a*) ⟶ *x j* = *q*) ∧ *IN a0 s* ∧ *IN a1 s* ∧ (∀ *i*::*nat*. *a1 i* = (*if* (*1*::*nat*) ≤ *i* ∧ *i* ≤ *n then a0 i* + (*1*::*nat*) *else a0 i*)) ⟶ *a* = *a0* ∨ *a* = *a1*

thm KSIMPLEX_FIX_PLANE_0:

∀ (*p*::*nat*) (*n*::*nat*) (*j*::*nat*) (*s*::(*nat* ⇒ *nat*) ⇒ *bool*) (*a*::*nat* ⇒ *nat*) (*a0*::*nat* ⇒ *nat*) *a1*::*nat* ⇒ *nat*. *ksimplex p n s* ∧ *IN a s* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *n* ∧ (∀ *x*::*nat* ⇒ *nat*. *IN x* (*DELETE s a*) ⟶ *x j* = (*0*::*nat*)) ∧ *IN a0 s* ∧ *IN a1*

$s \wedge (\forall\, i{::}nat.\ a1\ i = (if\ \ (1{::}nat) \le i \wedge i \le n\ then\ a0\ i + (1{::}nat)\ else\ a0\ i)) \longrightarrow a = a1$

thm KSIMPLEX_FIX_PLANE_P:

$\forall\,(p{::}nat)\ (n{::}nat)\ (j{::}nat)\ (s{::}(nat \Rightarrow nat) \Rightarrow bool)\ (a{::}nat \Rightarrow nat)\ (a0{::}nat \Rightarrow nat)\ a1{::}nat \Rightarrow nat.\ ksimplex\ p\ n\ s \wedge IN\ a\ s \wedge (1{::}nat) \le j \wedge j \le n \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ (DELETE\ s\ a) \longrightarrow x\ j = p) \wedge IN\ a0\ s \wedge IN\ a1\ s \wedge (\forall\, i{::}nat.\ a1\ i = (if\ \ (1{::}nat) \le i \wedge i \le n\ then\ a0\ i + (1{::}nat)\ else\ a0\ i)) \longrightarrow a = a0$

thm KSIMPLEX_REPLACE_0:

$ksimplex\ (?p{::}nat)\ (?n{::}nat)\ (?s{::}(nat \Rightarrow nat) \Rightarrow bool) \wedge IN\ (?a{::}nat \Rightarrow nat)\ ?s \wedge ?n \ne (0{::}nat) \wedge (\exists\, j{\ge}1{::}nat.\ j \le ?n \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ (DELETE\ ?s\ ?a) \longrightarrow x\ j = (0{::}nat))) \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1582{::}(nat \Rightarrow nat) \Rightarrow bool.\ \exists\, s'{::}(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1582\ (ksimplex\ ?p\ ?n\ s' \wedge (\exists\, b{::}nat \Rightarrow nat.\ IN\ b\ s' \wedge DELETE\ s'\ b = DELETE\ ?s\ ?a))\ s')) = (1{::}nat)$

thm KSIMPLEX_REPLACE_1:

$ksimplex\ (?p{::}nat)\ (?n{::}nat)\ (?s{::}(nat \Rightarrow nat) \Rightarrow bool) \wedge IN\ (?a{::}nat \Rightarrow nat)\ ?s \wedge ?n \ne (0{::}nat) \wedge (\exists\, j{\ge}1{::}nat.\ j \le ?n \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ (DELETE\ ?s\ ?a) \longrightarrow x\ j = ?p)) \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1583{::}(nat \Rightarrow nat) \Rightarrow bool.\ \exists\, s'{::}(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1583\ (ksimplex\ ?p\ ?n\ s' \wedge (\exists\, b{::}nat \Rightarrow nat.\ IN\ b\ s' \wedge DELETE\ s'\ b = DELETE\ ?s\ ?a))\ s')) = (1{::}nat)$

thm KSIMPLEX_REPLACE_2:

$ksimplex\ (?p{::}nat)\ (?n{::}nat)\ (?s{::}(nat \Rightarrow nat) \Rightarrow bool) \wedge IN\ (?a{::}nat \Rightarrow nat)\ ?s \wedge ?n \ne (0{::}nat) \wedge \neg\ (\exists\, j{\ge}1{::}nat.\ j \le ?n \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ (DELETE\ ?s\ ?a) \longrightarrow x\ j = (0{::}nat))) \wedge \neg\ (\exists\, j{\ge}1{::}nat.\ j \le ?n \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ (DELETE\ ?s\ ?a) \longrightarrow x\ j = ?p)) \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1584{::}(nat \Rightarrow nat) \Rightarrow bool.\ \exists\, s'{::}(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1584\ (ksimplex\ ?p\ ?n\ s' \wedge (\exists\, b{::}nat \Rightarrow nat.\ IN\ b\ s' \wedge DELETE\ s'\ b = DELETE\ ?s\ ?a))\ s')) = (2{::}nat)$

thm KUHN_SIMPLEX_LEMMA:

$\forall\,(p{::}nat)\ n{::}nat.\ (\forall\, s{::}(nat \Rightarrow nat) \Rightarrow bool.\ ksimplex\ p\ (n + (1{::}nat))\ s \longrightarrow SUBSET\ (IMAGE\ (?rl{::}(nat \Rightarrow nat) \Rightarrow nat)\ s)\ (dotdot\ (0{::}nat)\ (n + (1{::}nat)))) \wedge ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1588{::}(nat \Rightarrow nat) \Rightarrow bool.\ \exists\, f{::}(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1588\ ((\exists\,(s{::}(nat \Rightarrow nat) \Rightarrow bool)\ a{::}nat \Rightarrow nat.\ ksimplex\ p\ (n + (1{::}nat))\ s \wedge IN\ a\ s \wedge f = DELETE\ s\ a) \wedge IMAGE\ ?rl\ f = dotdot\ (0{::}nat)\ n \wedge ((\exists\, j{\ge}1{::}nat.\ j \le n + (1{::}nat) \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ f \longrightarrow x\ j = (0{::}nat))) \vee (\exists\, j{\ge}1{::}nat.\ j \le n + (1{::}nat) \wedge (\forall\, x{::}nat \Rightarrow nat.\ IN\ x\ f \longrightarrow x\ j = p))))\ f))) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1590{::}(nat \Rightarrow nat) \Rightarrow bool.\ \exists\, s{::}(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1590\ (IN\ s\ (GSPEC\ (\lambda GEN\%PVAR\%1589{::}(nat$

$\Rightarrow nat) \Rightarrow bool.\ \exists s::(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1589\ (ksimplex\ p\ (n + (1::nat))\ s)\ s)) \wedge IMAGE\ ?rl\ s = dotdot\ (0::nat)\ (n + (1::nat)))\ s)))$

thm DEF_reduced:

$reduced = (\lambda(\_740221::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (\_740222::nat)\ \_740223::nat \Rightarrow nat.\ SOME\ k::nat.\ k \leq \_740222 \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < k + (1::nat) \longrightarrow \_740221\ \_740223\ i = (0::nat)) \wedge (k = \_740222 \vee \_740221\ \_740223\ (k + (1::nat)) \neq (0::nat)))$

thm reduced:

$\forall (n::nat)\ (label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ x::nat \Rightarrow nat.\ reduced\ label\ n\ x = (SOME\ k::nat.\ k \leq n \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < k + (1::nat) \longrightarrow label\ x\ i = (0::nat)) \wedge (k = n \vee label\ x\ (k + (1::nat)) \neq (0::nat)))$

thm REDUCED_LABELLING:

$\forall (label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (x::nat \Rightarrow nat)\ n::nat.\ reduced\ label\ n\ x \leq n \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < reduced\ label\ n\ x + (1::nat) \longrightarrow label\ x\ i = (0::nat)) \wedge (reduced\ label\ n\ x = n \vee label\ x\ (reduced\ label\ n\ x + (1::nat)) \neq (0::nat))$

thm REDUCED_LABELLING_UNIQUE:

$\forall (label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (x::nat \Rightarrow nat)\ n::nat.\ (?r::nat) \leq n \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < ?r + (1::nat) \longrightarrow label\ x\ i = (0::nat)) \wedge (?r = n \vee label\ x\ (?r + (1::nat)) \neq (0::nat)) \longrightarrow reduced\ label\ n\ x = ?r$

thm REDUCED_LABELLING_0:

$\forall (label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (n::nat)\ (x::nat \Rightarrow nat)\ j::nat.\ (1::nat) \leq j \wedge j \leq n \wedge label\ x\ j = (0::nat) \longrightarrow reduced\ label\ n\ x \neq j - (1::nat)$

thm REDUCED_LABELLING_1:

$\forall (label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (n::nat)\ (x::nat \Rightarrow nat)\ j::nat.\ (1::nat) \leq j \wedge j \leq n \wedge label\ x\ j \neq (0::nat) \longrightarrow reduced\ label\ n\ x < j$

thm REDUCED_LABELLING_SUC:

$\forall (lab::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (n::nat)\ x::nat \Rightarrow nat.\ reduced\ lab\ (n + (1::nat))\ x \neq n + (1::nat) \longrightarrow reduced\ lab\ (n + (1::nat))\ x = reduced\ lab\ n\ x$

thm COMPLETE_FACE_TOP:

$\forall (lab::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (f::(nat \Rightarrow nat) \Rightarrow bool)\ n::nat.\ (\forall (x::nat \Rightarrow nat)\ j::nat.\ IN\ x\ f \wedge (1::nat) \leq j \wedge j \leq n + (1::nat) \wedge x\ j = (0::nat) \longrightarrow lab\ x\ j = (0::nat)) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ IN\ x\ f \wedge (1::nat) \leq j \wedge j \leq n + (1::nat) \wedge x\ j = (?p::nat) \longrightarrow lab\ x\ j = (1::nat)) \longrightarrow (IMAGE\ (reduced\ lab\ (n + (1::nat)))\ f = dotdot\ (0::nat)\ n \wedge ((\exists j \geq 1::nat.\ j \leq n + (1::nat) \wedge (\forall x::nat \Rightarrow nat.\ IN\ x\ f \longrightarrow x\ j = (0::nat))) \vee (\exists j \geq 1::nat.\ j \leq n + (1::nat) \wedge (\forall x::nat \Rightarrow nat.\ IN\ x\ f \longrightarrow x\ j = ?p)))) = (IMAGE\ (reduced\ lab\ (n + (1::nat)))\ f = dotdot\ (0::nat)\ n \wedge (\forall x::nat \Rightarrow nat.\ IN\ x\ f \longrightarrow x\ (n + (1::nat)) = ?p))$

thm KUHN_INDUCTION:

$\forall (p::nat)\ n::nat.\ (0::nat) < p \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ (\forall j::nat.\ x\ j \leq p) \wedge (1::nat) \leq j \wedge j \leq n + (1::nat) \wedge x\ j = (0::nat) \longrightarrow (?lab::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ x\ j = (0::nat)) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ (\forall j::nat.\ x\ j \leq p) \wedge (1::nat) \leq j \wedge j \leq n + (1::nat) \wedge x\ j = p \longrightarrow ?lab\ x\ j = (1::nat)) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1591::(nat \Rightarrow nat) \Rightarrow bool.\ \exists f::(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1591\ (ksimplex\ p\ n\ f \wedge IMAGE\ (reduced\ ?lab\ n)\ f = dotdot\ (0::nat)\ n)\ f))) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1592::(nat \Rightarrow nat) \Rightarrow bool.\ \exists s::(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1592\ (ksimplex\ p\ (n + (1::nat))\ s \wedge IMAGE\ (reduced\ ?lab\ (n + (1::nat)))\ s = dotdot\ (0::nat)\ (n + (1::nat)))\ s)))$

thm KSIMPLEX_0:

$ksimplex\ (?p::nat)\ (0::nat)\ (?s::(nat \Rightarrow nat) \Rightarrow bool) = (?s = INSERT\ (\lambda x::nat.\ ?p)\ EMPTY)$

thm REDUCE_LABELLING_0:

$\forall (lab::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ x::nat \Rightarrow nat.\ reduced\ lab\ (0::nat)\ x = (0::nat)$

thm KUHN_COMBINATORIAL:

$\forall (lab::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat)\ (p::nat)\ n::nat.\ (0::nat) < p \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ (\forall j::nat.\ x\ j \leq p) \wedge (1::nat) \leq j \wedge j \leq n \wedge x\ j = (0::nat) \longrightarrow lab\ x\ j = (0::nat)) \wedge (\forall (x::nat \Rightarrow nat)\ j::nat.\ (\forall j::nat.\ x\ j \leq p) \wedge (1::nat) \leq j \wedge j \leq n \wedge x\ j = p \longrightarrow lab\ x\ j = (1::nat)) \longrightarrow ODD\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1594::(nat \Rightarrow nat) \Rightarrow bool.\ \exists s::(nat \Rightarrow nat) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1594\ (ksimplex\ p\ n\ s \wedge IMAGE\ (reduced\ lab\ n)\ s = dotdot\ (0::nat)\ n)\ s)))$

thm KUHN_LEMMA:

$\forall (n::nat)\ (p::nat)\ label::(nat \Rightarrow nat) \Rightarrow nat \Rightarrow nat.\ (0::nat) < p \wedge (0::nat) < n \wedge (\forall x::nat \Rightarrow nat.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow x\ i \leq p) \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow label\ x\ i = (0::nat) \vee label\ x\ i = (1::nat))) \wedge (\forall x::nat \Rightarrow nat.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow x\ i \leq p) \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \wedge x\ i = (0::nat) \longrightarrow label\ x\ i = (0::nat))) \wedge (\forall x::nat \Rightarrow nat.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow x\ i \leq p) \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \wedge x\ i = p \longrightarrow label\ x\ i = (1::nat))) \longrightarrow (\exists q::nat \Rightarrow nat.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow q\ i < p) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq n \longrightarrow (\exists (r::nat \Rightarrow nat)\ s::nat \Rightarrow nat.\ (\forall j::nat.\ (1::nat) \leq j \wedge j \leq n \longrightarrow q\ j \leq r\ j \wedge r\ j \leq q\ j + (1::nat)) \wedge (\forall j::nat.\ (1::nat) \leq j \wedge j \leq n \longrightarrow q\ j \leq s\ j \wedge s\ j \leq q\ j + (1::nat)) \wedge label\ r\ i \neq label\ s\ i)))$

thm BROUWER_CUBE:

$\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) \wedge SUBSET\ (IMAGE\ f\ (closed\_interval\ [(vec$

$(0::nat)$, vec $(1::nat))]]))$ $(closed\_interval$ $[(vec$ $(0::nat)$, vec $(1::nat))])$ $\longrightarrow$ $(\exists x::(real, ?'a::type)$ $cart$. IN x $(closed\_interval$ $[(vec$ $(0::nat)$, vec $(1::nat))])$ $\wedge$ f x $=$ $x)$

thm DEF_retraction:

$retraction = (\lambda(\_746417::((real, ?'a::type)\ cart \Rightarrow bool) \times ((real, ?'a::type)\ cart \Rightarrow bool))\ \_746418::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ SUBSET\ (snd\ \_746417)\ (fst\ \_746417) \wedge continuous\_on\ \_746418\ (fst\ \_746417) \wedge SUBSET\ (IMAGE\ \_746418\ (fst\ \_746417))\ (snd\ \_746417) \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ (snd\ \_746417) \longrightarrow \_746418\ x = x))$

thm retraction:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ r::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ retraction\ (s, t)\ r = (SUBSET\ t\ s \wedge continuous\_on\ r\ s \wedge SUBSET\ (IMAGE\ r\ s)\ t \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ t \longrightarrow r\ x = x))$

thm DEF_retract_of:

$retract\_of = (\lambda(\_746434::(real, ?'a::type)\ cart \Rightarrow bool)\ \_746435::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists r::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ retraction\ (\_746435, \_746434)\ r)$

thm retract_of:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ t\ s = (\exists r::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ retraction\ (s, t)\ r)$

thm RETRACTION_IDEMPOTENT:

$\forall (r::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retraction\ (s, t)\ r \longrightarrow (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow r\ (r\ x) = r\ x)$

thm RETRACTION_SUBSET:

$\forall (r::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (s'::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retraction\ (s, t)\ r \wedge SUBSET\ t\ s' \wedge SUBSET\ s'\ s \longrightarrow retraction\ (s', t)\ r$

thm RETRACT_OF_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (s'::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ t\ s \wedge SUBSET\ t\ s' \wedge SUBSET\ s'\ s \longrightarrow retract\_of\ t\ s'$

thm RETRACT_OF_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ t\ s \longrightarrow retract\_of\ (IMAGE\ (vector\_add\ a)\ t)\ (IMAGE\ (vector\_add\ a)\ s)$

thm RETRACT_OF_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ (IMAGE\ (vector\_add\ a)\ t)\ (IMAGE\ (vector\_add\ a)\ s) = retract\_of\ t\ s$

thm RETRACT_OF_INJECTIVE_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge retract\_of\ t\ s \longrightarrow retract\_of\ (IMAGE\ f\ t)\ (IMAGE\ f\ s)$

thm RETRACT_OF_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \wedge (\forall y::(real, ?'a::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ f\ x = y) \longrightarrow retract\_of\ (IMAGE\ f\ t)\ (IMAGE\ f\ s) = retract\_of\ t\ s$

thm RETRACTION_REFL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ retraction\ (s,\ s)\ (\lambda x::(real, ?'a::type)\ cart.\ x)$

thm RETRACT_OF_REFL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ s\ s$

thm RETRACT_OF_IMP_SUBSET:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ s\ t \longrightarrow SUBSET\ s\ t$

thm RETRACTION_o:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ retraction\ (s,\ t)\ f \wedge retraction\ (t,\ u)\ g \longrightarrow retraction\ (s,\ u)\ (g \circ f)$

thm RETRACT_OF_TRANS:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ (t::(real, ?'a::type)\ cart \Rightarrow bool)\ u::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ s\ t \wedge retract\_of\ t\ u \longrightarrow retract\_of\ s\ u$

thm CLOSED_IN_RETRACT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ retract\_of\ s\ t \longrightarrow closed\_in\ (subtopology\ euclidean\ t)\ s$

thm RETRACT_OF_CONTRACTIBLE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ contractible\ t \wedge retract\_of\ s\ t \longrightarrow contractible\ s$

thm ABSOLUTE_RETRACT_CONVEX_CLOSED:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. convex s ∧ HOL_Light_Import.closed s ∧ s ≠ EMPTY ∧ SUBSET s t ⟶ retract_of s t

thm ABSOLUTE_RETRACT_HOMEOMORPHIC_IMAGE_INTERVAL:

∀ (s::(real, ?'b::type) cart ⇒ bool) (t::(real, ?'b::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. homeomorphic s (closed_interval [(a, b)]) ∧ s ≠ EMPTY ∧ SUBSET s t ⟶ retract_of s t

thm ABSOLUTE_RETRACT_PATH_IMAGE_ARC:

∀ (g::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. arc g ∧ SUBSET (path_image g) s ⟶ retract_of (path_image g) s

thm RELATIVE_BOUNDARY_RETRACT_OF_PUNCTURED_AFFINE_HULL:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. convex s ∧ compact s ∧ IN a (relative_interior s) ⟶ retract_of (DIFF s (relative_interior s)) (DELETE (hull affine s) a)

thm FRONTIER_RETRACT_OF_PUNCTURED_UNIVERSE:

∀ (s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. convex s ∧ compact s ∧ IN a (interior s) ⟶ retract_of (frontier s) (DELETE HOL_Light_Import.UNIV a)

thm SPHERE_RETRACT_OF_PUNCTURED_UNIVERSE_GEN:

∀ (a::(real, ?'a::type) cart) (r::real) b::(real, ?'a::type) cart. IN b (ball (a, r)) ⟶ retract_of (GSPEC (λGEN%PVAR%1597::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1597 (distance (a, x) = r) x)) (DELETE HOL_Light_Import.UNIV b)

thm SPHERE_RETRACT_OF_PUNCTURED_UNIVERSE:

∀ (a::(real, ?'a::type) cart) r::real. (0::real) < r ⟶ retract_of (GSPEC (λGEN%PVAR%1598::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1598 (vector_norm (vector_sub x a) = r) x)) (DELETE HOL_Light_Import.UNIV a)

thm RETRACT_OF_PASTECART:

∀ (s::(real, ?'b::type) cart ⇒ bool) (s'::(real, ?'b::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) t'::(real, ?'a::type) cart ⇒ bool. retract_of s s' ∧ retract_of t t' ⟶ retract_of (GSPEC (λGEN%PVAR%1599::(real, (?'b::type, ?'a::type) finite_sum) cart. ∃ (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1599 (IN x s ∧ IN y t) (pastecart x y))) (GSPEC (λGEN%PVAR%1600::(real, (?'b::type, ?'a::type) finite_sum) cart. ∃ (x::(real, ?'b::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1600 (IN x s' ∧ IN y t') (pastecart x y)))

thm BORSUK_HOMOTOPY_EXTENSION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (t::(real, ?'b::type) cart ⇒ bool) (u::(real, ?'a::type) cart ⇒ bool) v::(real, ?'a::type) cart ⇒ bool.

HOL_Light_Import.closed s ∧ HOL_Light_Import.closed t ∧ retract_of u v ∧ HOL_Light_Import.open v ∧ continuous_on f t ∧ SUBSET (IMAGE f t) u ∧ continuous_on g s ∧ SUBSET (IMAGE g s) u ∧ homotopic_with (λx::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. True) (s, u) f g ⟶ (∃ g'::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on g' t ∧ SUBSET (IMAGE g' t) u ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g' x = g x))

thm NULLHOMOTOPIC_INTO_SPHERE_EXTENSION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (a::(real, ?'a::type) cart) r::real. HOL_Light_Import.closed s ∧ continuous_on f s ∧ s ≠ EMPTY ∧ SUBSET (IMAGE f s) (GSPEC (λGEN%PVAR%1610::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1610 (vector_norm (vector_sub x a) = r) x)) ⟶ (∃ c::(real, ?'a::type) cart. homotopic_with (λx::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. True) (s, GSPEC (λGEN%PVAR%1611::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1611 (vector_norm (vector_sub x a) = r) x)) f (λx::(real, ?'b::type) cart. c)) = (∃ g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on g HOL_Light_Import.UNIV ∧ SUBSET (IMAGE g HOL_Light_Import.UNIV) (GSPEC (λGEN%PVAR%1612::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1612 (vector_norm (vector_sub x a) = r) x)) ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ g x = f x))

thm INVERTIBLE_FIXPOINT_PROPERTY:

∀ (s::(real, ?'b::type) cart ⇒ bool) (t::(real, ?'a::type) cart ⇒ bool) (i::(real, ?'a::type) cart ⇒ (real, ?'b::type) cart) r::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on i t ∧ SUBSET (IMAGE i t) s ∧ continuous_on r s ∧ SUBSET (IMAGE r s) t ∧ (∀ y::(real, ?'a::type) cart. IN y t ⟶ r (i y) = y) ⟶ (∀ f::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart. continuous_on f s ∧ SUBSET (IMAGE f s) s ⟶ (∃ x::(real, ?'b::type) cart. IN x s ∧ f x = x)) ⟶ (∀ g::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. continuous_on g t ∧ SUBSET (IMAGE g t) t ⟶ (∃ y::(real, ?'a::type) cart. IN y t ∧ g y = y))

thm HOMEOMORPHIC_FIXPOINT_PROPERTY:

∀ (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. homeomorphic s t ⟶ (∀ f::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart. continuous_on f s ∧ SUBSET (IMAGE f s) s ⟶ (∃ x::(real, ?'b::type) cart. IN x s ∧ f x = x)) = (∀ g::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. continuous_on g t ∧ SUBSET (IMAGE g t) t ⟶ (∃ y::(real, ?'a::type) cart. IN y t ∧ g y = y))

thm RETRACT_FIXPOINT_PROPERTY:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. retract_of t s ∧ (∀ f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. continuous_on f s ∧ SUBSET (IMAGE f s) s ⟶ (∃ x::(real, ?'a::type) cart. IN x s ∧ f x = x)) ⟶ (∀ g::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart. continuous_on g t ∧ SUBSET (IMAGE g t) t ⟶ (∃ y::(real, ?'a::type) cart. IN y t ∧ g y = y))

thm BROUWER_WEAK:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge convex\ s \wedge interior\ s \neq EMPTY \wedge continuous\_on\ f\ s \wedge SUBSET\ (IMAGE\ f\ s)\ s \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge f\ x = x)$

thm BROUWER_BALL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \wedge continuous\_on\ f\ (cball\ (a,\ e)) \wedge SUBSET\ (IMAGE\ f\ (cball\ (a,\ e)))\ (cball\ (a,\ e)) \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ (cball\ (a,\ e)) \wedge f\ x = x)$

thm BROUWER:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge convex\ s \wedge s \neq EMPTY \wedge continuous\_on\ f\ s \wedge SUBSET\ (IMAGE\ f\ s)\ s \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge f\ x = x)$

thm NO_RETRACTION_CBALL:

$\forall (a::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \longrightarrow \neg\ retract\_of\ (frontier\ (cball\ (a,\ e)))\ (cball\ (a,\ e))$

thm FRONTIER_SUBSET_RETRACTION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ r::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ bounded\ s \wedge SUBSET\ (frontier\ s)\ t \wedge continuous\_on\ r\ (closure\ s) \wedge SUBSET\ (IMAGE\ r\ s)\ t \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ t \longrightarrow r\ x = x) \longrightarrow SUBSET\ s\ t$

thm NO_RETRACTION_FRONTIER_BOUNDED:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \wedge interior\ s \neq EMPTY \longrightarrow \neg\ retract\_of\ (frontier\ s)\ s$

thm CONTRACTIBLE_SPHERE:

$\forall (a::(real,\ ?'a::type)\ cart)\ r::real.\ contractible\ (GSPEC\ (\lambda GEN\%PVAR\%1613::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1613\ (vector\_norm\ (vector\_sub\ x\ a) = r)\ x)) = (r \leq (0::real))$

thm DEF_interval_bij:

$interval\_bij = (\lambda(\_755600::(real,\ ?'a::type)\ cart \times (real,\ ?'a::type)\ cart)\ (\_755601::(real,\ ?'a::type)\ cart \times (real,\ ?'a::type)\ cart)\ \_755602::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ \$\ (fst\ \_755601)\ i + (\$\ \_755602\ i - \$\ (fst\ \_755600)\ i)\ /\ (\$\ (snd\ \_755600)\ i - \$\ (fst\ \_755600)\ i) * (\$\ (snd\ \_755601)\ i - \$\ (fst\ \_755601)\ i)))$

thm interval_bij:

$\forall (x::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ interval\_bij\ (a,\ b)\ (u,\ v)\ x = lambda\ (\lambda i::nat.\ \$\ u\ i + (\$\ x\ i - \$\ a\ i)\ /\ (\$\ b\ i - \$\ a\ i) * (\$\ v\ i - \$\ u\ i))$

thm INTERVAL_BIJ_AFFINE:

interval_bij (?a::(real, ?'a::type) cart, ?b::(real, ?'a::type) cart) (?u::(real, ?'a::type) cart, ?v::(real, ?'a::type) cart) = (λx::(real, ?'a::type) cart. vector_add (lambda (λi::nat. ($?v i − $?u i) / ($?b i − $?a i) ∗ $ x i)) (lambda (λi::nat. $?u i − ($?v i − $?u i) / ($?b i − $?a i) ∗ $?a i)))

thm CONTINUOUS_INTERVAL_BIJ:

∀ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) x::(real, ?'a::type) cart. continuous (interval_bij (a, b) (u, v)) (at x)*

thm CONTINUOUS_ON_INTERVAL_BIJ:

∀ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) s::(real, ?'a::type) cart ⇒ bool. continuous_on (interval_bij (a, b) (u, v)) s*

thm IN_INTERVAL_INTERVAL_BIJ:

∀ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) x::(real, ?'a::type) cart. IN x (closed_interval [(a, b)]) ∧ closed_interval [(u, v)] ≠ EMPTY ⟶ IN (interval_bij (a, b) (u, v) x) (closed_interval [(u, v)])*

thm INTERVAL_BIJ_BIJ:

∀ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) x::(real, ?'a::type) cart. (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ a i < $ b i ∧ $ u i < $ v i) ⟶ interval_bij (a, b) (u, v) (interval_bij (u, v) (a, b) x) = x*

thm INFNORM_2:

infnorm (?x::(real, 2) cart) = max |$?x (1::nat)| |$?x (2::nat)|

thm INFNORM_EQ_1_2:

(infnorm (?x::(real, 2) cart) = (1::real)) = (|$?x (1::nat)| ≤ (1::real) ∧ |$?x (2::nat)| ≤ (1::real) ∧ ($?x (1::nat) = − (1::real) ∨ $?x (1::nat) = (1::real) ∨ $?x (2::nat) = − (1::real) ∨ $?x (2::nat) = (1::real)))

thm INFNORM_EQ_1_IMP:

infnorm (?x::(real, 2) cart) = (1::real) ⟶ |$?x (1::nat)| ≤ (1::real) ∧ |$?x (2::nat)| ≤ (1::real)

thm FASHODA_UNIT:

∀ *(f::(real, unit) cart ⇒ (real, 2) cart) g::(real, unit) cart ⇒ (real, 2) cart. SUBSET (IMAGE f (closed_interval [(vector_neg (vec (1::nat)), vec (1::nat))])) (closed_interval [(vector_neg (vec (1::nat)), vec (1::nat))]) ∧ SUBSET (IMAGE g (closed_interval [(vector_neg (vec (1::nat)), vec (1::nat))])) (closed_interval [(vector_neg (vec (1::nat)), vec (1::nat))]) ∧ continuous_on f (closed_interval [(vector_neg (vec (1::nat)), vec (1::nat))]) ∧ continuous_on g (closed_interval*

$[(vector\_neg\ (vec\ (1::nat)),\ vec\ (1::nat))]) \wedge \$\ (f\ (vector\_neg\ (vec\ (1::nat))))\ (1::nat) = -\ (1::real) \wedge \$\ (f\ (vec\ (1::nat)))\ (1::nat) = (1::real) \wedge \$\ (g\ (vector\_neg\ (vec\ (1::nat))))\ (2::nat) = -\ (1::real) \wedge \$\ (g\ (vec\ (1::nat)))\ (2::nat) = (1::real) \longrightarrow (\exists\, (s::(real,\ unit)\ cart)\ t::(real,\ unit)\ cart.\ IN\ s\ (closed\_interval\ [(vector\_neg\ (vec\ (1::nat)),\ vec\ (1::nat))]) \wedge IN\ t\ (closed\_interval\ [(vector\_neg\ (vec\ (1::nat)),\ vec\ (1::nat))]) \wedge f\ s = g\ t)$

thm FASHODA_UNIT_PATH:

$\forall\, (f::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart)\ g::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart.\ path\ f \wedge path\ g \wedge SUBSET\ (path\_image\ f)\ (closed\_interval\ [(vector\_neg\ (vec\ (1::nat)),\ vec\ (1::nat))]) \wedge SUBSET\ (path\_image\ g)\ (closed\_interval\ [(vector\_neg\ (vec\ (1::nat)),\ vec\ (1::nat))]) \wedge \$\ (pathstart\ f)\ (1::nat) = -\ (1::real) \wedge \$\ (pathfinish\ f)\ (1::nat) = (1::real) \wedge \$\ (pathstart\ g)\ (2::nat) = -\ (1::real) \wedge \$\ (pathfinish\ g)\ (2::nat) = (1::real) \longrightarrow (\exists\, z::(real,\ 2)\ cart.\ IN\ z\ (path\_image\ f) \wedge IN\ z\ (path\_image\ g))$

thm FASHODA:

$\forall\, (f::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart)\ (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ path\ f \wedge path\ g \wedge SUBSET\ (path\_image\ f)\ (closed\_interval\ [(a,\ b)]) \wedge SUBSET\ (path\_image\ g)\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (pathstart\ f)\ (1::nat) = \$\ a\ (1::nat) \wedge \$\ (pathfinish\ f)\ (1::nat) = \$\ b\ (1::nat) \wedge \$\ (pathstart\ g)\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ (pathfinish\ g)\ (2::nat) = \$\ b\ (2::nat) \longrightarrow (\exists\, z::(real,\ 2)\ cart.\ IN\ z\ (path\_image\ f) \wedge IN\ z\ (path\_image\ g))$

thm SEGMENT_VERTICAL:

$\forall\, (a::(real,\ 2)\ cart)\ (b::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ \$\ a\ (1::nat) = \$\ b\ (1::nat) \longrightarrow IN\ x\ (closed\_segment\ [(a,\ b)]) = (\$\ x\ (1::nat) = \$\ a\ (1::nat) \wedge \$\ x\ (1::nat) = \$\ b\ (1::nat) \wedge (\$\ a\ (2::nat) \leq \$\ x\ (2::nat) \wedge \$\ x\ (2::nat) \leq \$\ b\ (2::nat) \vee \$\ b\ (2::nat) \leq \$\ x\ (2::nat) \wedge \$\ x\ (2::nat) \leq \$\ a\ (2::nat)))$

thm SEGMENT_HORIZONTAL:

$\forall\, (a::(real,\ 2)\ cart)\ (b::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ \$\ a\ (2::nat) = \$\ b\ (2::nat) \longrightarrow IN\ x\ (closed\_segment\ [(a,\ b)]) = (\$\ x\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ x\ (2::nat) = \$\ b\ (2::nat) \wedge (\$\ a\ (1::nat) \leq \$\ x\ (1::nat) \wedge \$\ x\ (1::nat) \leq \$\ b\ (1::nat) \vee \$\ b\ (1::nat) \leq \$\ x\ (1::nat) \wedge \$\ x\ (1::nat) \leq \$\ a\ (1::nat)))$

thm FASHODA_INTERLACE:

$\forall\, (f::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ 2)\ cart)\ (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ path\ f \wedge path\ g \wedge SUBSET\ (path\_image\ f)\ (closed\_interval\ [(a,\ b)]) \wedge SUBSET\ (path\_image\ g)\ (closed\_interval\ [(a,\ b)]) \wedge \$\ (pathstart\ f)\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ (pathfinish\ f)\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ (pathstart\ g)\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ (pathfinish\ g)\ (2::nat) = \$\ a\ (2::nat) \wedge \$\ (pathstart\ f)\ (1::nat) < \$\ (pathstart\ g)\ (1::nat) \wedge \$\ (pathstart\ g)\ (1::nat) < \$\ (pathfinish\ f)\ (1::nat) \wedge \$\ (pathfinish\ f)\ (1::nat) < \$\ (pathfinish$

g) (*1*::*nat*) ⟶ (∃ *z*::(*real*, *2*) *cart*. *IN z* (*path_image f*) ∧ *IN z* (*path_image g*))

thm RETRACTION_INJECTIVE_IMAGE_INTERVAL:

∀ (*p*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *closed_interval* [(*a*, *b*)] ≠ *EMPTY* ∧ *continuous_on p* (*closed_interval* [(*a*, *b*)]) ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ∧ *IN y* (*closed_interval* [(*a*, *b*)]) ∧ *p x* = *p y* ⟶ *x* = *y*) ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *continuous_on f HOL_Light_Import.UNIV* ∧ *SUBSET* (*IMAGE f HOL_Light_Import.UNIV*) (*IMAGE p* (*closed_interval* [(*a*, *b*)])) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*IMAGE p* (*closed_interval* [(*a*, *b*)])) ⟶ *f x* = *x*))

thm HOMEOMORPHIC_EMPTY_conjunct1:

∀ *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *homeomorphic EMPTY s* = (*s* = *EMPTY*)

thm HOMEOMORPHIC_EMPTY_conjunct0:

∀ *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *homeomorphic s EMPTY* = (*s* = *EMPTY*)

thm CONNECTED_COMPLEMENT_HOMEOMORPHIC_INTERVAL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ∧ *homeomorphic s* (*closed_interval* [(*a*, *b*)]) ⟶ *connected* (*DIFF HOL_Light_Import.UNIV s*)

thm PATH_CONNECTED_COMPLEMENT_HOMEOMORPHIC_INTERVAL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ∧ *homeomorphic s* (*closed_interval* [(*a*, *b*)]) ⟶ *path_connected* (*DIFF HOL_Light_Import.UNIV s*)

thm RETRACTION_ARC:

∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc p* ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *continuous_on f HOL_Light_Import.UNIV* ∧ *SUBSET* (*IMAGE f HOL_Light_Import.UNIV*) (*path_image p*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*path_image p*) ⟶ *f x* = *x*))

thm PATH_CONNECTED_ARC_COMPLEMENT:

∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ∧ *arc p* ⟶ *path_connected* (*DIFF HOL_Light_Import.UNIV* (*path_image p*))

thm CONNECTED_ARC_COMPLEMENT:

∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ∧ *arc p* ⟶ *connected* (*DIFF HOL_Light_Import.UNIV* (*path_image p*))

thm INSIDE_ARC_EMPTY:

∀ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *arc p* ⟶ *inside* (*path_image p*) = *EMPTY*

thm INSIDE_SIMPLE_CURVE_IMP_CLOSED:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *simple_path g* ∧ *IN x* (*inside* (*path_image g*)) ⟶ *pathfinish g* = *pathstart g*

thm JORDAN_CURVE_THEOREM:

∀ *c*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*. *simple_path c* ∧ *pathfinish c* = *pathstart c* ⟶ (∃ (*ins*::(*real*, *2*) *cart* ⇒ *bool*) *out*::(*real*, *2*) *cart* ⇒ *bool*. *ins* ≠ *EMPTY* ∧ *HOL_Light_Import.open ins* ∧ *connected ins* ∧ *out* ≠ *EMPTY* ∧ *HOL_Light_Import.open out* ∧ *connected out* ∧ *bounded ins* ∧ ¬ *bounded out* ∧ *HOL_Light_Import.INTER ins out* = *EMPTY* ∧ *HOL_Light_Import.UNION ins out* = *DIFF HOL_Light_Import.UNIV* (*path_image c*) ∧ *frontier ins* = *path_image c* ∧ *frontier out* = *path_image c*)

thm JORDAN_DISCONNECTED:

∀ *c*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*. *simple_path c* ∧ *pathfinish c* = *pathstart c* ⟶ ¬ *connected* (*DIFF HOL_Light_Import.UNIV* (*path_image c*))

thm JORDAN_INSIDE_OUTSIDE:

∀ *c*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*. *simple_path c* ∧ *pathfinish c* = *pathstart c* ⟶ *inside* (*path_image c*) ≠ *EMPTY* ∧ *HOL_Light_Import.open* (*inside* (*path_image c*)) ∧ *connected* (*inside* (*path_image c*)) ∧ *outside* (*path_image c*) ≠ *EMPTY* ∧ *HOL_Light_Import.open* (*outside* (*path_image c*)) ∧ *connected* (*outside* (*path_image c*)) ∧ *bounded* (*inside* (*path_image c*)) ∧ ¬ *bounded* (*outside* (*path_image c*)) ∧ *HOL_Light_Import.INTER* (*inside* (*path_image c*)) (*outside* (*path_image c*)) = *EMPTY* ∧ *HOL_Light_Import.UNION* (*inside* (*path_image c*)) (*outside* (*path_image c*)) = *DIFF HOL_Light_Import.UNIV* (*path_image c*) ∧ *frontier* (*inside* (*path_image c*)) = *path_image c* ∧ *frontier* (*outside* (*path_image c*)) = *path_image c*

thm SPLIT_INSIDE_SIMPLE_CLOSED_CURVE:

∀ (*c1*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*) (*c2*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*) (*c*::(*real*, *unit*) *cart* ⇒ (*real*, *2*) *cart*) (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. *a* ≠ *b* ∧ *simple_path c1* ∧ *pathstart c1* = *a* ∧ *pathfinish c1* = *b* ∧ *simple_path c2* ∧ *pathstart c2* = *a* ∧ *pathfinish c2* = *b* ∧ *simple_path c* ∧ *pathstart c* = *a* ∧ *pathfinish c* = *b* ∧ *HOL_Light_Import.INTER* (*path_image c1*) (*path_image c2*) = *INSERT a* (*INSERT b EMPTY*) ∧ *HOL_Light_Import.INTER* (*path_image c1*) (*path_image c*) = *INSERT a* (*INSERT b EMPTY*) ∧ *HOL_Light_Import.INTER* (*path_image c2*) (*path_image c*) = *INSERT a* (*INSERT b EMPTY*) ∧ *HOL_Light_Import.INTER* (*path_image c*) (*inside* (*HOL_Light_Import.UNION* (*path_image c1*) (*path_image c2*))) ≠ *EMPTY* ⟶ *HOL_Light_Import.INTER* (*inside* (*HOL_Light_Import.UNION* (*path_image c1*) (*path_image c*))) (*inside* (*HOL_Light_Import.UNION* (*path_image c2*) (*path_image c*))) = *EMPTY* ∧ *HOL_Light_Import.UNION* (*inside* (*HOL_Light_Import.UNION* (*path_image c1*) (*path_image c*))) (*HOL_Light_Import.UNION* (*inside* (*HOL_Light_Import.UNION* (*path_image c2*) (*path_image c*))) (*DIFF* (*path_image c*) (*INSERT a* (*INSERT b EMPTY*)))) = *inside* (*HOL_Light_Import.UNION* (*path_image c1*) (*path_image c2*))

thm DEF_has_derivative:

has_derivative = (λ(*_1518840*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*_1518841*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *_1518842*::(*real*, *?'b*::*type*) *cart net*. *linear _1518841* ∧ −−> (λ*y*::(*real*, *?'b*::*type*) *cart*. % (*inverse_class.inverse* (*vector_norm* (*vector_sub y* (*netlimit _1518842*)))) (*vector_sub* (*_1518840 y*) (*vector_add* (*_1518840* (*netlimit _1518842*)) (*_1518841* (*vector_sub y* (*netlimit _1518842*)))))) (*vec* (*0*::*nat*)) *_1518842*)

thm has_derivative:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *has_derivative f f' net* = (*linear f'* ∧ −−> (λ*y*::(*real*, *?'b*::*type*) *cart*. % (*inverse_class.inverse* (*vector_norm* (*vector_sub y* (*netlimit net*)))) (*vector_sub* (*f y*) (*vector_add* (*f* (*netlimit net*)) (*f'* (*vector_sub y* (*netlimit net*)))))) (*vec* (*0*::*nat*)) *net*)

thm has_derivative_within:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *has_derivative f f'* (*within* (*at x*) *s*) = (*linear f'* ∧ −−> (λ*y*::(*real*, *?'b*::*type*) *cart*. % (*inverse_class.inverse* (*vector_norm* (*vector_sub y x*))) (*vector_sub* (*f y*) (*vector_add* (*f x*) (*f'* (*vector_sub y x*))))) (*vec* (*0*::*nat*)) (*within* (*at x*) *s*))

thm has_derivative_at:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'b*::*type*) *cart*. *has_derivative f f'* (*at x*) = (*linear f'* ∧ −−> (λ*y*::(*real*, *?'b*::*type*) *cart*. % (*inverse_class.inverse* (*vector_norm* (*vector_sub y x*))) (*vector_sub* (*f y*) (*vector_add* (*f x*) (*f'* (*vector_sub y x*))))) (*vec* (*0*::*nat*)) (*at x*))

thm HAS_DERIVATIVE_WITHIN:

has_derivative (*?f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*?f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*within* (*at* (*?x*::(*real*, *?'b*::*type*) *cart*)) (*?s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*)) = (*linear ?f'* ∧ (∀ *e*>*0*::*real*. ∃ *d*>*0*::*real*. ∀ *x'*::(*real*, *?'b*::*type*) *cart*. *IN x' ?s* ∧ (*0*::*real*) < *vector_norm* (*vector_sub x' ?x*) ∧ *vector_norm* (*vector_sub x' ?x*) < *d* ⟶ *vector_norm* (*vector_sub* (*vector_sub* (*?f x'*) (*?f ?x*)) (*?f'* (*vector_sub x' ?x*))) / *vector_norm* (*vector_sub x' ?x*) < *e*))

thm HAS_DERIVATIVE_AT:

has_derivative (*?f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*?f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*at* (*?x*::(*real*, *?'b*::*type*) *cart*)) = (*linear ?f'* ∧ (∀ *e*>*0*::*real*. ∃ *d*>*0*::*real*. ∀ *x'*::(*real*, *?'b*::*type*) *cart*. (*0*::*real*) < *vector_norm* (*vector_sub x' ?x*) ∧ *vector_norm* (*vector_sub x' ?x*) < *d* ⟶ *vector_norm* (*vector_sub* (*vector_sub* (*?f x'*) (*?f ?x*)) (*?f'* (*vector_sub x' ?x*))) / *vector_norm* (*vector_sub x' ?x*) < *e*))

thm HAS_DERIVATIVE_AT_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_derivative\ f\ (?f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (at\ x) \longrightarrow has\_derivative\ f\ ?f'\ (within\ (at\ x)\ s)$

thm HAS_DERIVATIVE_WITHIN_OPEN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ a\ s \land HOL\_Light\_Import.open\ s \longrightarrow has\_derivative\ f\ f'\ (within\ (at\ a)\ s) = has\_derivative\ f\ f'\ (at\ a)$

thm HAS_DERIVATIVE_LINEAR:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ linear\ f \longrightarrow has\_derivative\ f\ f\ net$

thm HAS_DERIVATIVE_ID:

$\forall net::(real,\ ?'a::type)\ cart\ net.\ has\_derivative\ (\lambda x::(real,\ ?'a::type)\ cart.\ x)\ (\lambda h::(real,\ ?'a::type)\ cart.\ h)\ net$

thm HAS_DERIVATIVE_CONST:

$\forall (c::(real,\ ?'b::type)\ cart)\ net::(real,\ ?'a::type)\ cart\ net.\ has\_derivative\ (\lambda x::(real,\ ?'a::type)\ cart.\ c)\ (\lambda h::(real,\ ?'a::type)\ cart.\ vec\ (0::nat))\ net$

thm HAS_DERIVATIVE_CMUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (net::(real,\ ?'b::type)\ cart\ net)\ c::real.\ has\_derivative\ f\ f'\ net \longrightarrow has\_derivative\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ c\ (f\ x))\ (\lambda h::(real,\ ?'b::type)\ cart.\ \%\ c\ (f'\ h))\ net$

thm HAS_DERIVATIVE_CMUL_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (net::(real,\ ?'b::type)\ cart\ net)\ c::real.\ c \neq (0::real) \longrightarrow has\_derivative\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ c\ (f\ x))\ (\lambda h::(real,\ ?'b::type)\ cart.\ \%\ c\ (f'\ h))\ net = has\_derivative\ f\ f'\ net$

thm HAS_DERIVATIVE_NEG:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ has\_derivative\ f\ f'\ net \longrightarrow has\_derivative\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f\ x))\ (\lambda h::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f'\ h))\ net$

thm HAS_DERIVATIVE_NEG_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ has\_derivative\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f\ x))\ (\lambda h::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f'\ h))\ net = has\_derivative\ f\ f'\ net$

thm HAS_DERIVATIVE_ADD:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *has_derivative f f' net* ∧ *has_derivative g g' net* ⟶ *has_derivative* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_add* (*f x*) (*g x*)) (λ*h*::(*real*, *?'b*::*type*) *cart*. *vector_add* (*f' h*) (*g' h*)) *net*

thm HAS_DERIVATIVE_SUB:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *has_derivative f f' net* ∧ *has_derivative g g' net* ⟶ *has_derivative* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_sub* (*f x*) (*g x*)) (λ*h*::(*real*, *?'b*::*type*) *cart*. *vector_sub* (*f' h*) (*g' h*)) *net*

thm HAS_DERIVATIVE_VSUM:

∀ (*f*::*?'c*::*type* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*net*::(*real*, *?'b*::*type*) *cart net*) *s*::*?'c*::*type* ⇒ *bool*. *FINITE s* ∧ (∀ *a*::*?'c*::*type*. *IN a s* ⟶ *has_derivative* (*f a*) ((*?f'*::*?'c*::*type* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *a*) *net*) ⟶ *has_derivative* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vsum s* (λ*a*::*?'c*::*type*. *f a x*)) (λ*h*::(*real*, *?'b*::*type*) *cart*. *vsum s* (λ*a*::*?'c*::*type*. *?f' a h*)) *net*

thm HAS_DERIVATIVE_VSUM_NUMSEG:

∀ (*f*::*nat* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*net*::(*real*, *?'b*::*type*) *cart net*) (*m*::*nat*) *n*::*nat*. (∀ *i*::*nat*. *m* ≤ *i* ∧ *i* ≤ *n* ⟶ *has_derivative* (*f i*) ((*?f'*::*nat* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *i*) *net*) ⟶ *has_derivative* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vsum* (*dotdot m n*) (λ*i*::*nat*. *f i x*)) (λ*h*::(*real*, *?'b*::*type*) *cart*. *vsum* (*dotdot m n*) (λ*i*::*nat*. *?f' i h*)) *net*

thm HAS_DERIVATIVE_VMUL_COMPONENT:

∀ (*c*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*c'*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*k*::*nat*) *v*::(*real*, *?'a*::*type*) *cart*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *has_derivative c c'* (*?net*::(*real*, *?'c*::*type*) *cart net*) ⟶ *has_derivative* (λ*x*::(*real*, *?'c*::*type*) *cart*. % ($ (*c x*) *k*) *v*) (λ*x*::(*real*, *?'c*::*type*) *cart*. % ($ (*c' x*) *k*) *v*) *?net*

thm HAS_DERIVATIVE_VMUL_DROP:

∀ (*c*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *unit*) *cart*) (*c'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *unit*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *has_derivative c c'* (*?net*::(*real*, *?'b*::*type*) *cart net*) ⟶ *has_derivative* (λ*x*::(*real*, *?'b*::*type*) *cart*. % (*HOL_Light_Import.drop* (*c x*)) *v*) (λ*x*::(*real*, *?'b*::*type*) *cart*. % (*HOL_Light_Import.drop* (*c' x*)) *v*) *?net*

thm HAS_DERIVATIVE_LIFT_DOT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *has_derivative f f'* (*?net*::(*real*, *?'b*::*type*) *cart net*) ⟶

has_derivative (λ*x*::(*real*, *?'b*::*type*) *cart*. *lift* (*dot* (*?v*::(*real*, *?'a*::*type*) *cart*) (*f* *x*))) (λ*t*::(*real*, *?'b*::*type*) *cart*. *lift* (*dot* *?v* (*f'* *t*))) *?net*

thm HAS_DERIVATIVE_TRANSFORM_WITHIN:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *d*::*real*. (*0*::*real*) < *d* ∧ *IN* *x* *s* ∧ (∀ *x'*::(*real*, *?'b*::*type*) *cart*. *IN* *x'* *s* ∧ *distance* (*x'*, *x*) < *d* ⟶ *f* *x'* = *g* *x'*) ∧ *has_derivative* *f* *f'* (*within* (*at* *x*) *s*) ⟶ *has_derivative* *g* *f'* (*within* (*at* *x*) *s*)

thm HAS_DERIVATIVE_TRANSFORM_AT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) *d*::*real*. (*0*::*real*) < *d* ∧ (∀ *x'*::(*real*, *?'b*::*type*) *cart*. *distance* (*x'*, *x*) < *d* ⟶ *f* *x'* = *g* *x'*) ∧ *has_derivative* *f* *f'* (*at* *x*) ⟶ *has_derivative* *g* *f'* (*at* *x*)

thm HAS_DERIVATIVE_TRANSFORM_WITHIN_OPEN:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'b*::*type*) *cart*. *HOL_Light_Import.open* *s* ∧ *IN* *x* *s* ∧ (∀ *y*::(*real*, *?'b*::*type*) *cart*. *IN* *y* *s* ⟶ *f* *y* = *g* *y*) ∧ *has_derivative* *f* (*?f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*at* *x*) ⟶ *has_derivative* *g* *?f'* (*at* *x*)

thm DEF_differentiable:

differentiable = (λ(*_1519169*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *_1519170*::(*real*, *?'b*::*type*) *cart* *net*. ∃ *f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *has_derivative* *_1519169* *f'* *_1519170*)

thm differentiable:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart* *net*. *differentiable* *f* *net* = (∃ *f'*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *has_derivative* *f* *f'* *net*)

thm DEF_differentiable_on:

differentiable_on = (λ(*_1519181*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *_1519182*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. ∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN* *x* *_1519182* ⟶ *differentiable* *_1519181* (*within* (*at* *x*) *_1519182*))

thm differentiable_on:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *differentiable_on* *f* *s* = (∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN* *x* *s* ⟶ *differentiable* *f* (*within* (*at* *x*) *s*))

thm HAS_DERIVATIVE_IMP_DIFFERENTIABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ has\_derivative\ f\ f'\ net \longrightarrow differentiable\ f\ net$

thm DIFFERENTIABLE_AT_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ differentiable\ f\ (at\ x) \longrightarrow differentiable\ f\ (within\ (at\ x)\ s)$

thm DIFFERENTIABLE_WITHIN_OPEN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ a\ s \wedge HOL\_Light\_Import.open\ s \longrightarrow differentiable\ f\ (within\ (at\ a)\ s) = differentiable\ f\ (at\ a)$

thm DIFFERENTIABLE_AT_IMP_DIFFERENTIABLE_ON:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow differentiable\ f\ (at\ x)) \longrightarrow differentiable\_on\ f\ s$

thm DIFFERENTIABLE_ON_EQ_DIFFERENTIABLE_AT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow differentiable\_on\ f\ s = (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow differentiable\ f\ (at\ x))$

thm DIFFERENTIABLE_TRANSFORM_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ x\ s \wedge (\forall x'::(real,\ ?'b::type)\ cart.\ IN\ x'\ s \wedge distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge differentiable\ f\ (within\ (at\ x)\ s) \longrightarrow differentiable\ g\ (within\ (at\ x)\ s)$

thm DIFFERENTIABLE_TRANSFORM_AT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ d::real.\ (0::real) < d \wedge (\forall x'::(real,\ ?'b::type)\ cart.\ distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge differentiable\ f\ (at\ x) \longrightarrow differentiable\ g\ (at\ x)$

thm DEF_frechet_derivative:

$frechet\_derivative = (\lambda(\_1519634::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ \_1519635::(real,\ ?'b::type)\ cart\ net.\ SOME\ f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ has\_derivative\ \_1519634\ f'\ \_1519635)$

thm frechet_derivative:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ frechet\_derivative\ f\ net = (SOME\ f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ has\_derivative\ f\ f'\ net)$

thm FRECHET_DERIVATIVE_WORKS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ differentiable\ f\ net = has\_derivative\ f\ (frechet\_derivative\ f\ net)\ net$

thm LINEAR_FRECHET_DERIVATIVE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ differentiable\ f\ net \longrightarrow linear\ (frechet\_derivative\ f\ net)$

thm DEF_jacobian:

$jacobian = (\lambda(\_1519646::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ \_1519647::(real,\ ?'b::type)\ cart\ net.\ matrix\ (frechet\_derivative\ \_1519646\ \_1519647))$

thm jacobian:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ jacobian\ f\ net = matrix\ (frechet\_derivative\ f\ net)$

thm JACOBIAN_WORKS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ net::(real,\ ?'b::type)\ cart\ net.\ differentiable\ f\ net = has\_derivative\ f\ (matrix\_vector\_mul\ (jacobian\ f\ net))\ net$

thm LIM_MUL_NORM_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ {-}{-}{>}\ f\ (vec\ (0::nat))\ (within\ (at\ a)\ s) \longrightarrow {-}{-}{>}\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ (vector\_norm\ (vector\_sub\ x\ a))\ (f\ x))\ (vec\ (0::nat))\ (within\ (at\ a)\ s)$

thm DIFFERENTIABLE_IMP_CONTINUOUS_WITHIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ differentiable\ f\ (within\ (at\ (?x::(real,\ ?'b::type)\ cart))\ s) \longrightarrow continuous\ f\ (within\ (at\ ?x)\ s)$

thm DIFFERENTIABLE_IMP_CONTINUOUS_AT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ differentiable\ f\ (at\ x) \longrightarrow continuous\ f\ (at\ x)$

thm DIFFERENTIABLE_IMP_CONTINUOUS_ON:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ differentiable\_on\ f\ s \longrightarrow continuous\_on\ f\ s$

thm HAS_DERIVATIVE_WITHIN_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ has\_derivative\ f\ (?f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (within\ (at\ x)\ s) \land SUBSET\ t\ s \longrightarrow has\_derivative\ f\ ?f'\ (within\ (at\ x)\ t)$

thm DIFFERENTIABLE_WITHIN_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ differentiable\ f\ (within\ (at\ (?x::(real,\ ?'b::type)\ cart))\ t) \wedge SUBSET\ s\ t \longrightarrow differentiable\ f\ (within\ (at\ ?x)\ s)$

thm DIFFERENTIABLE_ON_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ differentiable\_on\ f\ t \wedge SUBSET\ s\ t \longrightarrow differentiable\_on\ f\ s$

thm DIFFERENTIABLE_ON_EMPTY:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ differentiable\_on\ f\ EMPTY$

thm HAS_DERIVATIVE_WITHIN_ALT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ has\_derivative\ f\ f'\ (within\ (at\ x)\ s) = (linear\ f' \wedge (\forall e>0::real.\ \exists d>0::real.\ \forall y::(real,\ ?'b::type)\ cart.\ IN\ y\ s \wedge vector\_norm\ (vector\_sub\ y\ x) < d \longrightarrow vector\_norm\ (vector\_sub\ (vector\_sub\ (f\ y)\ (f\ x))\ (f'\ (vector\_sub\ y\ x))) \leq e * vector\_norm\ (vector\_sub\ y\ x)))$

thm HAS_DERIVATIVE_AT_ALT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ has\_derivative\ f\ f'\ (at\ x) = (linear\ f' \wedge (\forall e>0::real.\ \exists d>0::real.\ \forall y::(real,\ ?'b::type)\ cart.\ vector\_norm\ (vector\_sub\ y\ x) < d \longrightarrow vector\_norm\ (vector\_sub\ (vector\_sub\ (f\ y)\ (f\ x))\ (f'\ (vector\_sub\ y\ x))) \leq e * vector\_norm\ (vector\_sub\ y\ x)))$

thm DIFF_CHAIN_WITHIN:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'c::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ has\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge has\_derivative\ g\ g'\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow has\_derivative\ (g \circ f)\ (g' \circ f')\ (within\ (at\ x)\ s)$

thm DIFF_CHAIN_AT:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x::(real,\ ?'c::type)\ cart.\ has\_derivative\ f\ f'\ (at\ x) \wedge has\_derivative\ g\ g'\ (at\ (f\ x)) \longrightarrow has\_derivative\ (g \circ f)\ (g' \circ f')\ (at\ x)$

thm DIFFERENTIABLE_CONST:

$\forall (c::(real,\ ?'b::type)\ cart)\ net::(real,\ ?'a::type)\ cart\ net.\ differentiable\ (\lambda z::(real,\ ?'a::type)\ cart.\ c)\ net$

thm DIFFERENTIABLE_ID:

$\forall$ *net*::(*real*, *?'a*::*type*) *cart net*. *differentiable* (λz::(*real*, *?'a*::*type*) *cart*. *z*) *net*

thm DIFFERENTIABLE_CMUL:

$\forall$ (*net*::(*real*, *?'b*::*type*) *cart net*) (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *c*::*real*. *differentiable f net* $\longrightarrow$ *differentiable* (λx::(*real*, *?'b*::*type*) *cart*. % *c* (*f x*)) *net*

thm DIFFERENTIABLE_NEG:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *differentiable f net* $\longrightarrow$ *differentiable* (λz::(*real*, *?'b*::*type*) *cart*. *vector_neg* (*f z*)) *net*

thm DIFFERENTIABLE_ADD:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *differentiable f net* $\wedge$ *differentiable g net* $\longrightarrow$ *differentiable* (λz::(*real*, *?'b*::*type*) *cart*. *vector_add* (*f z*) (*g z*)) *net*

thm DIFFERENTIABLE_SUB:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *net*::(*real*, *?'b*::*type*) *cart net*. *differentiable f net* $\wedge$ *differentiable g net* $\longrightarrow$ *differentiable* (λz::(*real*, *?'b*::*type*) *cart*. *vector_sub* (*f z*) (*g z*)) *net*

thm DIFFERENTIABLE_VSUM:

$\forall$ (*f*::*?'c*::*type* $\Rightarrow$ (*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*net*::(*real*, *?'b*::*type*) *cart net*) *s*::*?'c*::*type* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ ($\forall$ *a*::*?'c*::*type*. *IN a s* $\longrightarrow$ *differentiable* (*f a*) *net*) $\longrightarrow$ *differentiable* (λx::(*real*, *?'b*::*type*) *cart*. *vsum s* (λa::*?'c*::*type*. *f a x*)) *net*

thm DIFFERENTIABLE_VSUM_NUMSEG:

$\forall$ (*f*::*nat* $\Rightarrow$ (*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*net*::(*real*, *?'c*::*type*) *cart net*) (*m*::*nat*) *n*::*nat*. *FINITE* (*?s*::*?'a*::*type* $\Rightarrow$ *bool*) $\wedge$ ($\forall$ *i*::*nat*. $m \leq i \wedge i \leq n$ $\longrightarrow$ *differentiable* (*f i*) *net*) $\longrightarrow$ *differentiable* (λx::(*real*, *?'c*::*type*) *cart*. *vsum* (*dotdot m n*) (λa::*nat*. *f a x*)) *net*

thm DIFFERENTIABLE_CHAIN_AT:

$\forall$ (*f*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'c*::*type*) *cart*. *differentiable f* (*at x*) $\wedge$ *differentiable g* (*at* (*f x*)) $\longrightarrow$ *differentiable* ($g \circ f$) (*at x*)

thm DIFFERENTIABLE_CHAIN_WITHIN:

$\forall$ (*f*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'c*::*type*) *cart*) *s*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ *bool*.

differentiable f (within (at x) s) ∧ differentiable g (within (at (f x)) (IMAGE f s)) ⟶ differentiable (g ∘ f) (within (at x) s)

thm DIFFERENTIABLE_ON_CONST:

∀ (s::(real, ?'b::type) cart ⇒ bool) c::(real, ?'a::type) cart. differentiable_on (λz::(real, ?'b::type) cart. c) s

thm DIFFERENTIABLE_ON_ID:

∀ s::(real, ?'a::type) cart ⇒ bool. differentiable_on (λz::(real, ?'a::type) cart. z) s

thm DIFFERENTIABLE_ON_COMPOSE:

∀ (f::(real, ?'c::type) cart ⇒ (real, ?'b::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'c::type) cart ⇒ bool. differentiable_on f s ∧ differentiable_on g (IMAGE f s) ⟶ differentiable_on (g ∘ f) s

thm DIFFERENTIABLE_ON_NEG:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. differentiable_on f s ⟶ differentiable_on (λz::(real, ?'b::type) cart. vector_neg (f z)) s

thm DIFFERENTIABLE_ON_ADD:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. differentiable_on f s ∧ differentiable_on g s ⟶ differentiable_on (λz::(real, ?'b::type) cart. vector_add (f z) (g z)) s

thm DIFFERENTIABLE_ON_SUB:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. differentiable_on f s ∧ differentiable_on g s ⟶ differentiable_on (λz::(real, ?'b::type) cart. vector_sub (f z) (g z)) s

thm FRECHET_DERIVATIVE_UNIQUE_WITHIN:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f'::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f''::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (x::(real, ?'b::type) cart) s::(real, ?'b::type) cart ⇒ bool. has_derivative f f' (within (at x) s) ∧ has_derivative f f'' (within (at x) s) ∧ (∀ (i::nat) e::real. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (0::real) < e ⟶ (∃ d::real. (0::real) < |d| ∧ |d| < e ∧ IN (vector_add x (% d (basis i))) s)) ⟶ f' = f''

thm FRECHET_DERIVATIVE_UNIQUE_AT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f'::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (f''::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart)

x::(*real*, ?'*b*::*type*) *cart*. *has_derivative* *f* *f'* (*at* *x*) ∧ *has_derivative* *f* *f''* (*at* *x*) ⟶ *f'* = *f''*

thm HAS_FRECHET_DERIVATIVE_UNIQUE_AT:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) *x*::(*real*, ?'*b*::*type*) *cart*. *has_derivative* *f* *f'* (*at* *x*) ⟶ *frechet_derivative* *f* (*at* *x*) = *f'*

thm FRECHET_DERIVATIVE_CONST_AT:

∀ (*c*::(*real*, ?'*b*::*type*) *cart*) *a*::(*real*, ?'*a*::*type*) *cart*. *frechet_derivative* (λ*x*::(*real*, ?'*a*::*type*) *cart*. *c*) (*at* *a*) = (λ*h*::(*real*, ?'*a*::*type*) *cart*. *vec* (*0*::*nat*))

thm FRECHET_DERIVATIVE_UNIQUE_WITHIN_CLOSED_INTERVAL:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f''*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*x*::(*real*, ?'*b*::*type*) *cart*) (*a*::(*real*, ?'*b*::*type*) *cart*) *b*::(*real*, ?'*b*::*type*) *cart*. (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex* *HOL_Light_Import.UNIV* ⟶ $ *a* *i* < $ *b* *i*) ∧ *IN* *x* (*closed_interval* [(*a*, *b*)]) ∧ *has_derivative* *f* *f'* (*within* (*at* *x*) (*closed_interval* [(*a*, *b*)])) ∧ *has_derivative* *f* *f''* (*within* (*at* *x*) (*closed_interval* [(*a*, *b*)])) ⟶ *f'* = *f''*

thm FRECHET_DERIVATIVE_UNIQUE_WITHIN_OPEN_INTERVAL:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f''*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*x*::(*real*, ?'*b*::*type*) *cart*) (*a*::(*real*, ?'*b*::*type*) *cart*) *b*::(*real*, ?'*b*::*type*) *cart*. *IN* *x* (*open_interval* (*a*, *b*)) ∧ *has_derivative* *f* *f'* (*within* (*at* *x*) (*open_interval* (*a*, *b*))) ∧ *has_derivative* *f* *f''* (*within* (*at* *x*) (*open_interval* (*a*, *b*))) ⟶ *f'* = *f''*

thm FRECHET_DERIVATIVE_AT:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) *x*::(*real*, ?'*b*::*type*) *cart*. *has_derivative* *f* *f'* (*at* *x*) ⟶ *f'* = *frechet_derivative* *f* (*at* *x*)

thm FRECHET_DERIVATIVE_WITHIN_CLOSED_INTERVAL:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*x*::(*real*, ?'*b*::*type*) *cart*) (*a*::(*real*, ?'*b*::*type*) *cart*) *b*::(*real*, ?'*b*::*type*) *cart*. (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex* *HOL_Light_Import.UNIV* ⟶ $ *a* *i* < $ *b* *i*) ∧ *IN* *x* (*closed_interval* [(*a*, *b*)]) ∧ *has_derivative* *f* *f'* (*within* (*at* *x*) (*closed_interval* [(*a*, *b*)])) ⟶ *frechet_derivative* *f* (*within* (*at* *x*) (*closed_interval* [(*a*, *b*)])) = *f'*

thm DIFFERENTIAL_COMPONENT_POS_AT_MINIMUM:

∀ (*f*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*f'*::(*real*, ?'*b*::*type*) *cart* ⇒ (*real*, ?'*a*::*type*) *cart*) (*x*::(*real*, ?'*b*::*type*) *cart*) (*s*::(*real*, ?'*b*::*type*) *cart* ⇒ *bool*) (*k*::*nat*) *e*::*real*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex* *HOL_Light_Import.UNIV* ∧ *IN* *x* *s* ∧ *convex* *s* ∧ *has_derivative* *f* *f'* (*within* (*at* *x*) *s*) ∧ (*0*::*real*) < *e* ∧

$(\forall\, w{::}(real,\ ?'b{::}type)\ cart.\ IN\ w\ (HOL\_Light\_Import.INTER\ s\ (ball\ (x,\ e))) \longrightarrow \$\ (f\, x)\ k \le \$\ (f\, w)\ k) \longrightarrow (\forall\, y{::}(real,\ ?'b{::}type)\ cart.\ IN\ y\ s \longrightarrow (0{::}real) \le \$\ (f'\ (vector\_sub\ y\ x))\ k)$

thm DIFFERENTIAL_COMPONENT_NEG_AT_MAXIMUM:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ (k{::}nat)\ e{::}real.\ (1{::}nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \wedge IN\ x\ s \wedge convex\ s \wedge has\_derivative\ f\, f'\ (within\ (at\ x)\ s) \wedge (0{::}real) < e \wedge (\forall\, w{::}(real,\ ?'b{::}type)\ cart.\ IN\ w\ (HOL\_Light\_Import.INTER\ s\ (ball\ (x,\ e))) \longrightarrow \$\ (f\ w)\ k \le \$\ (f\ x)\ k) \longrightarrow (\forall\, y{::}(real,\ ?'b{::}type)\ cart.\ IN\ y\ s \longrightarrow \$\ (f'\ (vector\_sub\ y\ x))\ k \le (0{::}real))$

thm DROP_DIFFERENTIAL_POS_AT_MINIMUM:

$\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ unit)\ cart)\ (x{::}(real,\ ?'a{::}type)\ cart)\ (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ e{::}real.\ IN\ x\ s \wedge convex\ s \wedge has\_derivative\ f\, f'\ (within\ (at\ x)\ s) \wedge (0{::}real) < e \wedge (\forall\, w{::}(real,\ ?'a{::}type)\ cart.\ IN\ w\ (HOL\_Light\_Import.INTER\ s\ (ball\ (x,\ e))) \longrightarrow HOL\_Light\_Import.drop\ (f\, x) \le HOL\_Light\_Import.drop\ (f\, w)) \longrightarrow (\forall\, y{::}(real,\ ?'a{::}type)\ cart.\ IN\ y\ s \longrightarrow (0{::}real) \le HOL\_Light\_Import.drop\ (f'\ (vector\_sub\ y\ x)))$

thm DROP_DIFFERENTIAL_NEG_AT_MAXIMUM:

$\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ unit)\ cart)\ (x{::}(real,\ ?'a{::}type)\ cart)\ (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ e{::}real.\ IN\ x\ s \wedge convex\ s \wedge has\_derivative\ f\, f'\ (within\ (at\ x)\ s) \wedge (0{::}real) < e \wedge (\forall\, w{::}(real,\ ?'a{::}type)\ cart.\ IN\ w\ (HOL\_Light\_Import.INTER\ s\ (ball\ (x,\ e))) \longrightarrow HOL\_Light\_Import.drop\ (f\, w) \le HOL\_Light\_Import.drop\ (f\, x)) \longrightarrow (\forall\, y{::}(real,\ ?'a{::}type)\ cart.\ IN\ y\ s \longrightarrow HOL\_Light\_Import.drop\ (f'\ (vector\_sub\ y\ x)) \le (0{::}real))$

thm DIFFERENTIAL_COMPONENT_ZERO_AT_MAXMIN:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ k{::}nat.\ (1{::}nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \wedge IN\ x\ s \wedge HOL\_Light\_Import.open\ s \wedge has\_derivative\ f\ f'\ (at\ x) \wedge ((\forall\, w{::}(real,\ ?'b{::}type)\ cart.\ IN\ w\ s \longrightarrow \$\ (f\, w)\ k \le \$\ (f\, x)\ k) \vee (\forall\, w{::}(real,\ ?'b{::}type)\ cart.\ IN\ w\ s \longrightarrow \$\ (f\ x)\ k \le \$\ (f\ w)\ k)) \longrightarrow (\forall\, h{::}(real,\ ?'b{::}type)\ cart.\ \$\ (f'\ h)\ k = (0{::}real))$

thm DIFFERENTIAL_ZERO_MAXMIN_COMPONENT:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (x{::}(real,\ ?'b{::}type)\ cart)\ (e{::}real)\ k{::}nat.\ (1{::}nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \wedge (0{::}real) < e \wedge ((\forall\, y{::}(real,\ ?'b{::}type)\ cart.\ IN\ y\ (ball\ (x,\ e)) \longrightarrow \$\ (f\ y)\ k \le \$\ (f\, x)\ k) \vee (\forall\, y{::}(real,\ ?'b{::}type)\ cart.\ IN\ y\ (ball\ (x,\ e)) \longrightarrow \$\ (f\, x)\ k \le \$\ (f\ y)\ k)) \wedge differentiable\ f\ (at\ x) \longrightarrow \$\ (jacobian\ f\ (at\ x))\ k = vec\ (0{::}nat)$

thm DIFFERENTIAL_ZERO_MAXMIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ s \wedge HOL\_Light\_Import.open\ s \wedge has\_derivative\ f\ f'\ (at\ x) \wedge ((\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ y) \leq HOL\_Light\_Import.drop\ (f\ x)) \vee (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (f\ y))) \longrightarrow f' = (\lambda v::(real,\ ?'a::type)\ cart.\ vec\ (0::nat))$

thm ROLLE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b \wedge f\ a = f\ b \wedge continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \longrightarrow has\_derivative\ f\ (f'\ x)\ (at\ x)) \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \wedge f'\ x = (\lambda v::(real,\ unit)\ cart.\ vec\ (0::nat)))$

thm MVT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b \wedge continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \longrightarrow has\_derivative\ f\ (f'\ x)\ (at\ x)) \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \wedge vector\_sub\ (f\ b)\ (f\ a) = f'\ x\ (vector\_sub\ b\ a))$

thm MVT_SIMPLE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow has\_derivative\ f\ (f'\ x)\ (within\ (at\ x)\ (closed\_interval\ [(a,\ b)]))) \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \wedge vector\_sub\ (f\ b)\ (f\ a) = f'\ x\ (vector\_sub\ b\ a))$

thm MVT_VERY_SIMPLE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow has\_derivative\ f\ (f'\ x)\ (within\ (at\ x)\ (closed\_interval\ [(a,\ b)]))) \longrightarrow (\exists x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge vector\_sub\ (f\ b)\ (f\ a) = f'\ x\ (vector\_sub\ b\ a))$

thm MVT_GENERAL:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b \wedge continuous\_on\ f\ (closed\_interval$

[(a, b)]) ∧ (∀ x::(real, unit) cart. IN x (open_interval (a, b)) ⟶ has_derivative f (f′ x) (at x)) ⟶ (∃ x::(real, unit) cart. IN x (open_interval (a, b)) ∧ vector_norm (vector_sub (f b) (f a)) ≤ vector_norm (f′ x (vector_sub b a)))

thm DIFFERENTIABLE_BOUND:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (f′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (s::(real, ?′b::type) cart ⇒ bool) B::real. convex s ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (f′ x) (within (at x) s)) ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ onorm (f′ x) ≤ B) ⟶ (∀ (x::(real, ?′b::type) cart) y::(real, ?′b::type) cart. IN x s ∧ IN y s ⟶ vector_norm (vector_sub (f x) (f y)) ≤ B ∗ vector_norm (vector_sub x y))

thm HAS_DERIVATIVE_ZERO_CONSTANT:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) s::(real, ?′b::type) cart ⇒ bool. convex s ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (λh::(real, ?′b::type) cart. vec (0::nat)) (within (at x) s)) ⟶ (∃ c::(real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ f x = c)

thm HAS_DERIVATIVE_ZERO_UNIQUE:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (s::(real, ?′b::type) cart ⇒ bool) (a::(real, ?′b::type) cart) c::(real, ?′a::type) cart. convex s ∧ IN a s ∧ f a = c ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (λh::(real, ?′b::type) cart. vec (0::nat)) (within (at x) s)) ⟶ (∀ x::(real, ?′b::type) cart. IN x s ⟶ f x = c)

thm HAS_DERIVATIVE_ZERO_CONNECTED_CONSTANT:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) s::(real, ?′b::type) cart ⇒ bool. HOL_Light_Import.open s ∧ connected s ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (λh::(real, ?′b::type) cart. vec (0::nat)) (at x)) ⟶ (∃ c::(real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ f x = c)

thm HAS_DERIVATIVE_ZERO_CONNECTED_UNIQUE:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (s::(real, ?′b::type) cart ⇒ bool) (a::(real, ?′b::type) cart) c::(real, ?′a::type) cart. HOL_Light_Import.open s ∧ connected s ∧ IN a s ∧ f a = c ∧ (∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (λh::(real, ?′b::type) cart. vec (0::nat)) (at x)) ⟶ (∀ x::(real, ?′b::type) cart. IN x s ⟶ f x = c)

thm HAS_DERIVATIVE_INVERSE_BASIC:

∀ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (g::(real, ?′a::type) cart ⇒ (real, ?′b::type) cart) (f′::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (g′::(real, ?′a::type) cart ⇒ (real, ?′b::type) cart) (t::(real, ?′a::type) cart ⇒ bool) y::(real, ?′a::type) cart. has_derivative f f′ (at (g y)) ∧ linear g′ ∧ g′ ∘ f′ = id ∧ continuous g (at y) ∧ HOL_Light_Import.open t ∧ IN y t ∧ (∀ z::(real, ?′a::type) cart. IN z t ⟶ f (g z) = z) ⟶ has_derivative g g′ (at y)

thm HAS_DERIVATIVE_INVERSE_BASIC_X:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ has\_derivative\ f\ f'\ (at\ x) \wedge linear\ g' \wedge g' \circ f' = id \wedge continuous\ g\ (at\ (f\ x)) \wedge g\ (f\ x) = x \wedge HOL\_Light\_Import.open\ t \wedge IN\ (f\ x)\ t \wedge (\forall y::(real,\ ?'a::type)\ cart.\ IN\ y\ t \longrightarrow f\ (g\ y) = y) \longrightarrow has\_derivative\ g\ g'\ (at\ (f\ x))$

thm HAS_DERIVATIVE_INVERSE_DIEUDONNE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge HOL\_Light\_Import.open\ (IMAGE\ f\ s) \wedge continuous\_on\ f\ s \wedge continuous\_on\ g\ (IMAGE\ f\ s) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \longrightarrow (\forall (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge has\_derivative\ f\ f'\ (at\ x) \wedge linear\ g' \wedge g' \circ f' = id \longrightarrow has\_derivative\ g\ g'\ (at\ (f\ x)))$

thm HAS_DERIVATIVE_INVERSE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ compact\ s \wedge IN\ x\ s \wedge IN\ (f\ x)\ (interior\ (IMAGE\ f\ s)) \wedge continuous\_on\ f\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \wedge has\_derivative\ f\ f'\ (at\ x) \wedge linear\ g' \wedge g' \circ f' = id \longrightarrow has\_derivative\ g\ g'\ (at\ (f\ x))$

thm BROUWER_SURJECTIVE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ t \wedge convex\ t \wedge t \neq EMPTY \wedge continuous\_on\ f\ t \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ t \longrightarrow IN\ (vector\_add\ x\ (vector\_sub\ y\ (f\ y)))\ t) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists y::(real,\ ?'a::type)\ cart.\ IN\ y\ t \wedge f\ y = x))$

thm BROUWER_SURJECTIVE_CBALL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \wedge continuous\_on\ f\ (cball\ (a,\ e)) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ (cball\ (a,\ e)) \longrightarrow IN\ (vector\_add\ x\ (vector\_sub\ y\ (f\ y)))\ (cball\ (a,\ e))) \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (\exists y::(real,\ ?'a::type)\ cart.\ IN\ y\ (cball\ (a,\ e)) \wedge f\ y = x))$

thm SUSSMANN_OPEN_MAPPING:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ x::(real,\ ?'b::type)\ cart.\ HOL\_Light\_Import.open\ s \wedge continuous\_on\ f\ s \wedge IN\ x\ s \wedge has\_derivative\ f\ f'\ (at\ x) \wedge linear\ g' \wedge f' \circ$

$g' = id \longrightarrow (\forall\, t{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ SUBSET\ t\ s \wedge IN\ x\ (interior\ t) \longrightarrow IN\ (f\,x)\ (interior\ (IMAGE\ f\ t)))$

thm HAS_DERIVATIVE_INVERSE_STRONG:

$\forall\,(f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ x{::}(real,\ ?'a{::}type)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ x\ s \wedge continuous\_on\ f\ s \wedge (\forall\, x{::}(real,\ ?'a{::}type)\ cart.\ IN\ x\ s \longrightarrow g\ (f\,x) = x) \wedge has\_derivative\ f\ f'\ (at\ x) \wedge f' \circ g' = id \longrightarrow has\_derivative\ g\ g'\ (at\ (f\,x))$

thm HAS_DERIVATIVE_INVERSE_STRONG_X:

$\forall\,(f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool)\ y{::}(real,\ ?'a{::}type)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ (g\ y)\ s \wedge continuous\_on\ f\ s \wedge (\forall\, x{::}(real,\ ?'a{::}type)\ cart.\ IN\ x\ s \longrightarrow g\ (f\,x) = x) \wedge has\_derivative\ f\ f'\ (at\ (g\ y)) \wedge f' \circ g' = id \wedge f\ (g\ y) = y \longrightarrow has\_derivative\ g\ g'\ (at\ y)$

thm HAS_DERIVATIVE_INVERSE_ON:

$\forall\,(f{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge (\forall\, x{::}(real,\ ?'a{::}type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ f\ ((?f'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ x)\ (at\ x) \wedge (?g{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f\ x) = x \wedge ?f'\ x \circ (?g'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ x = id) \longrightarrow (\forall\, x{::}(real,\ ?'a{::}type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ ?g\ (?g'\ x)\ (at\ (f\,x)))$

thm HAS_DERIVATIVE_LOCALLY_INJECTIVE:

$\forall\,(f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (g'{::}(real,\ ?'a{::}type)\ cart \Rightarrow (real,\ ?'b{::}type)\ cart)\ (s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ a{::}(real,\ ?'b{::}type)\ cart.\ IN\ a\ s \wedge HOL\_Light\_Import.open\ s \wedge linear\ g' \wedge g' \circ f'\ a = id \wedge (\forall\, x{::}(real,\ ?'b{::}type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ f\ (f'\ x)\ (at\ x)) \wedge (\forall\, e{>}0{::}real.\ \exists\, d{>}0{::}real.\ \forall\, x{::}(real,\ ?'b{::}type)\ cart.\ distance\ (a,\ x) < d \longrightarrow onorm\ (\lambda v{::}(real,\ ?'b{::}type)\ cart.\ vector\_sub\ (f'\ x\ v)\ (f'\ a\ v)) < e) \longrightarrow (\exists\, t{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool.\ IN\ a\ t \wedge HOL\_Light\_Import.open\ t \wedge (\forall\,(x{::}(real,\ ?'b{::}type)\ cart)\ x'{::}(real,\ ?'b{::}type)\ cart.\ IN\ x\ t \wedge IN\ x'\ t \wedge f\ x' = f\ x \longrightarrow x' = x))$

thm HAS_DERIVATIVE_SEQUENCE_LIPSCHITZ:

$\forall\,(s{::}(real,\ ?'b{::}type)\ cart \Rightarrow bool)\ (f{::}nat \Rightarrow (real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (f'{::}nat \Rightarrow (real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ g'{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart.\ convex\ s \wedge (\forall\,(n{::}nat)\ x{::}(real,\ ?'b{::}type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ (f\,n)\ (f'\ n\ x)\ (within\ (at\ x)\ s)) \wedge (\forall\, e{>}0{::}real.\ \exists\, N{::}nat.\ \forall\,(n{::}nat)\ (x{::}(real,\ ?'b{::}type)\ cart)\ h{::}(real,\ ?'b{::}type)\ cart.\ N \leq n \wedge IN\ x\ s \longrightarrow vector\_norm$

(vector_sub (f′ n x h) (g′ x h)) ≤ e ∗ vector_norm h) ⟶ (∀ e>0::real. ∃ N::nat. ∀ (m::nat) (n::nat) (x::(real, ?′b::type) cart) y::(real, ?′b::type) cart. N ≤ m ∧ N ≤ n ∧ IN x s ∧ IN y s ⟶ vector_norm (vector_sub (vector_sub (f m x) (f n x)) (vector_sub (f m y) (f n y))) ≤ e ∗ vector_norm (vector_sub x y))

thm HAS_DERIVATIVE_SEQUENCE:

∀ (s::(real, ?′b::type) cart ⇒ bool) (f::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (f′::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) g′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart. convex s ∧ (∀ (n::nat) x::(real, ?′b::type) cart. IN x s ⟶ has_derivative (f n) (f′ n x) (within (at x) s)) ∧ (∀ e>0::real. ∃ N::nat. ∀ (n::nat) (x::(real, ?′b::type) cart) h::(real, ?′b::type) cart. N ≤ n ∧ IN x s ⟶ vector_norm (vector_sub (f′ n x h) (g′ x h)) ≤ e ∗ vector_norm h) ∧ (∃ (x::(real, ?′b::type) cart) l::(real, ?′a::type) cart. IN x s ∧ --> (λn::nat. f n x) l sequentially) ⟶ (∃ g::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ --> (λn::nat. f n x) (g x) sequentially ∧ has_derivative g (g′ x) (within (at x) s))

thm HAS_ANTIDERIVATIVE_SEQUENCE:

∀ (s::(real, ?′b::type) cart ⇒ bool) (f::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (f′::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) g′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart. convex s ∧ (∀ (n::nat) x::(real, ?′b::type) cart. IN x s ⟶ has_derivative (f n) (f′ n x) (within (at x) s)) ∧ (∀ e>0::real. ∃ N::nat. ∀ (n::nat) (x::(real, ?′b::type) cart) h::(real, ?′b::type) cart. N ≤ n ∧ IN x s ⟶ vector_norm (vector_sub (f′ n x h) (g′ x h)) ≤ e ∗ vector_norm h) ⟶ (∃ g::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative g (g′ x) (within (at x) s))

thm HAS_ANTIDERIVATIVE_LIMIT:

∀ (s::(real, ?′b::type) cart ⇒ bool) g′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart. convex s ∧ (∀ e>0::real. ∃ (f::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart) f′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative f (f′ x) (within (at x) s) ∧ (∀ h::(real, ?′b::type) cart. vector_norm (vector_sub (f′ x h) (g′ x h)) ≤ e ∗ vector_norm h)) ⟶ (∃ g::(real, ?′b::type) cart ⇒ (real, ?′a::type) cart. ∀ x::(real, ?′b::type) cart. IN x s ⟶ has_derivative g (g′ x) (within (at x) s))

thm HAS_DERIVATIVE_SERIES:

∀ (s::(real, ?′b::type) cart ⇒ bool) (f::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (f′::nat ⇒ (real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) (g′::(real, ?′b::type) cart ⇒ (real, ?′b::type) cart ⇒ (real, ?′a::type) cart) k::nat ⇒ bool. convex s ∧ (∀ (n::nat) x::(real, ?′b::type) cart. IN x s ⟶ has_derivative (f n) (f′ n x) (within (at x) s)) ∧ (∀ e>0::real. ∃ N::nat. ∀ (n::nat) (x::(real, ?′b::type) cart) h::(real, ?′b::type) cart. N ≤ n

$\wedge$ *IN x s* $\longrightarrow$ *vector_norm (vector_sub (vsum (HOL_Light_Import.INTER k (dotdot (0::nat) n)) (λi::nat. f′ i x h)) (g′ x h))* $\leq$ *e* $*$ *vector_norm h)* $\wedge$ (∃ *(x::(real, ?′b::type) cart) l::(real, ?′a::type) cart. IN x s* $\wedge$ *sums (λn::nat. f n x) l k)* $\longrightarrow$ (∃ *g::(real, ?′b::type) cart* ⇒ *(real, ?′a::type) cart.* ∀ *x::(real, ?′b::type) cart. IN x s* $\longrightarrow$ *sums (λn::nat. f n x) (g x) k* $\wedge$ *has_derivative g (g′ x) (within (at x) s))*

thm HAS_DERIVATIVE_BILINEAR_WITHIN:

∀ *(h::(real, ?′d::type) cart* ⇒ *(real, ?′c::type) cart* ⇒ *(real, ?′b::type) cart) (f::(real, ?′a::type) cart* ⇒ *(real, ?′d::type) cart) (g::(real, ?′a::type) cart* ⇒ *(real, ?′c::type) cart) (f′::(real, ?′a::type) cart* ⇒ *(real, ?′d::type) cart) (g′::(real, ?′a::type) cart* ⇒ *(real, ?′c::type) cart) (x::(real, ?′a::type) cart) s::(real, ?′a::type) cart* ⇒ *bool. has_derivative f f′ (within (at x) s)* $\wedge$ *has_derivative g g′ (within (at x) s)* $\wedge$ *bilinear h* $\longrightarrow$ *has_derivative (λx::(real, ?′a::type) cart. h (f x) (g x)) (λd::(real, ?′a::type) cart. vector_add (h (f x) (g′ d)) (h (f′ d) (g x))) (within (at x) s)*

thm HAS_DERIVATIVE_BILINEAR_AT:

∀ *(h::(real, ?′d::type) cart* ⇒ *(real, ?′c::type) cart* ⇒ *(real, ?′b::type) cart) (f::(real, ?′a::type) cart* ⇒ *(real, ?′d::type) cart) (g::(real, ?′a::type) cart* ⇒ *(real, ?′c::type) cart) (f′::(real, ?′a::type) cart* ⇒ *(real, ?′d::type) cart) (g′::(real, ?′a::type) cart* ⇒ *(real, ?′c::type) cart) x::(real, ?′a::type) cart. has_derivative f f′ (at x)* $\wedge$ *has_derivative g g′ (at x)* $\wedge$ *bilinear h* $\longrightarrow$ *has_derivative (λx::(real, ?′a::type) cart. h (f x) (g x)) (λd::(real, ?′a::type) cart. vector_add (h (f x) (g′ d)) (h (f′ d) (g x))) (at x)*

thm DEF_has_vector_derivative:

has_vector_derivative = (λ(_1540601::(real, unit) cart ⇒ *(real, ?′a::type) cart) _1540602::(real, ?′a::type) cart. has_derivative _1540601 (λx::(real, unit) cart. % (HOL_Light_Import.drop x) _1540602))*

thm has_vector_derivative:

∀ *(f::(real, unit) cart* ⇒ *(real, ?′a::type) cart) (f′::(real, ?′a::type) cart) net::(real, unit) cart net. has_vector_derivative f f′ net = has_derivative f (λx::(real, unit) cart. % (HOL_Light_Import.drop x) f′) net*

thm DEF_vector_derivative:

vector_derivative = (λ(_1540622::(real, unit) cart ⇒ *(real, ?′a::type) cart) _1540623::(real, unit) cart net. SOME f′::(real, ?′a::type) cart. has_vector_derivative _1540622 f′ _1540623)*

thm vector_derivative:

∀ *(f::(real, unit) cart* ⇒ *(real, ?′a::type) cart) net::(real, unit) cart net. vector_derivative f net = (SOME f′::(real, ?′a::type) cart. has_vector_derivative f f′ net)*

thm VECTOR_DERIVATIVE_WORKS:

$\forall$ (*net*::(*real*, *unit*) *cart net*) *f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *differentiable f net* = *has_vector_derivative f* (*vector_derivative f net*) *net*

thm VECTOR_DERIVATIVE_UNIQUE_AT:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *unit*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) *f''*::(*real*, *?'a*::*type*) *cart*. *has_vector_derivative f f'* (*at x*) $\wedge$ *has_vector_derivative f f''* (*at x*) $\longrightarrow$ *f'* = *f''*

thm HAS_VECTOR_DERIVATIVE_UNIQUE_AT:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *unit*) *cart*. *has_vector_derivative f f'* (*at x*) $\longrightarrow$ *vector_derivative f* (*at x*) = *f'*

thm VECTOR_DERIVATIVE_UNIQUE_WITHIN_CLOSED_INTERVAL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) (*x*::(*real*, *unit*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) *f''*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.drop a* < *HOL_Light_Import.drop b* $\wedge$ *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *has_vector_derivative f f'* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) $\wedge$ *has_vector_derivative f f''* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) $\longrightarrow$ *f'* = *f''*

thm VECTOR_DERIVATIVE_AT:

has_vector_derivative (*?f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*?f'*::(*real*, *?'a*::*type*) *cart*) (*at* (*?x*::(*real*, *unit*) *cart*)) $\longrightarrow$ *vector_derivative ?f* (*at ?x*) = *?f'*

thm VECTOR_DERIVATIVE_WITHIN_CLOSED_INTERVAL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *unit*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *HOL_Light_Import.drop a* < *HOL_Light_Import.drop b* $\wedge$ *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *has_vector_derivative f f'* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) $\longrightarrow$ *vector_derivative f* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) = *f'*

thm HAS_VECTOR_DERIVATIVE_WITHIN_SUBSET:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*) *x*::(*real*, *unit*) *cart*. *has_vector_derivative f* (*?f'*::(*real*, *?'a*::*type*) *cart*) (*within* (*at x*) *s*) $\wedge$ *SUBSET t s* $\longrightarrow$ *has_vector_derivative f ?f'* (*within* (*at x*) *t*)

thm HAS_VECTOR_DERIVATIVE_CONST:

$\forall$ (*c*::(*real*, *?'a*::*type*) *cart*) *net*::(*real*, *unit*) *cart net*. *has_vector_derivative* (λ*x*::(*real*, *unit*) *cart*. *c*) (*vec* (*0*::*nat*)) *net*

thm VECTOR_DERIVATIVE_CONST_AT:

$\forall$ (*c*::(*real*, *?'a*::*type*) *cart*) *a*::(*real*, *unit*) *cart*. *vector_derivative* (λ*x*::(*real*, *unit*) *cart*. *c*) (*at a*) = *vec* (*0*::*nat*)

thm HAS_VECTOR_DERIVATIVE_ID:

$\forall$ *net*::(*real*, *unit*) *cart net*. *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *x*) (*vec* (*1*::*nat*)) *net*

thm HAS_VECTOR_DERIVATIVE_CMUL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) (*net*::(*real*, *unit*) *cart net*) *c*::*real*. *has_vector_derivative f f' net* $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. % *c* (*f x*)) (% *c f'*) *net*

thm HAS_VECTOR_DERIVATIVE_CMUL_EQ:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) (*net*::(*real*, *unit*) *cart net*) *c*::*real*. *c* $\neq$ (*0*::*real*) $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. % *c* (*f x*)) (% *c f'*) *net* = *has_vector_derivative f f' net*

thm HAS_VECTOR_DERIVATIVE_NEG:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) *net*::(*real*, *unit*) *cart net*. *has_vector_derivative f f' net* $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *vector_neg* (*f x*)) (*vector_neg f'*) *net*

thm HAS_VECTOR_DERIVATIVE_NEG_EQ:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) *net*::(*real*, *unit*) *cart net*. *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *vector_neg* (*f x*)) (*vector_neg f'*) *net* = *has_vector_derivative f f' net*

thm HAS_VECTOR_DERIVATIVE_ADD:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g'*::(*real*, *?'a*::*type*) *cart*) *net*::(*real*, *unit*) *cart net*. *has_vector_derivative f f' net* $\wedge$ *has_vector_derivative g g' net* $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *vector_add* (*f x*) (*g x*)) (*vector_add f' g'*) *net*

thm HAS_VECTOR_DERIVATIVE_SUB:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f'*::(*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g'*::(*real*, *?'a*::*type*) *cart*) *net*::(*real*, *unit*) *cart net*. *has_vector_derivative f f' net* $\wedge$ *has_vector_derivative g g' net* $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *vector_sub* (*f x*) (*g x*)) (*vector_sub f' g'*) *net*

thm HAS_VECTOR_DERIVATIVE_BILINEAR_WITHIN:

$\forall$ (*h*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'c*::*type*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*f'*::(*real*, *?'c*::*type*) *cart*) (*g'*::(*real*, *?'b*::*type*) *cart*) (*x*::(*real*, *unit*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_vector_derivative f f'* (*within* (*at x*) *s*) $\wedge$ *has_vector_derivative g g'* (*within* (*at x*) *s*) $\wedge$ *bilinear h* $\longrightarrow$ *has_vector_derivative* (λx::(*real*, *unit*) *cart*. *h* (*f x*) (*g x*)) (*vector_add* (*h* (*f x*) *g'*) (*h f'* (*g x*))) (*within* (*at x*) *s*)

thm HAS_VECTOR_DERIVATIVE_BILINEAR_AT:

$\forall (h::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'c::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (f'::(real,\ ?'c::type)\ cart)\ (g'::(real,\ ?'b::type)\ cart)\ x::(real,\ unit)\ cart.\ has\_vector\_derivative\ f\ f'\ (at\ x) \wedge has\_vector\_derivative\ g\ g'\ (at\ x) \wedge bilinear\ h \longrightarrow has\_vector\_derivative\ (\lambda x::(real,\ unit)\ cart.\ h\ (f\ x)\ (g\ x))\ (vector\_add\ (h\ (f\ x)\ g')\ (h\ f'\ (g\ x)))\ (at\ x)$

thm HAS_VECTOR_DERIVATIVE_AT_WITHIN:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ unit)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_vector\_derivative\ f\ (?f'::(real,\ ?'a::type)\ cart)\ (at\ x) \longrightarrow has\_vector\_derivative\ f\ ?f'\ (within\ (at\ x)\ s)$

thm HAS_VECTOR_DERIVATIVE_TRANSFORM_WITHIN:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ unit)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ x\ s \wedge (\forall x'::(real,\ unit)\ cart.\ IN\ x'\ s \wedge distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge has\_vector\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_vector\_derivative\ g\ f'\ (within\ (at\ x)\ s)$

thm HAS_VECTOR_DERIVATIVE_TRANSFORM_AT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ unit)\ cart)\ d::real.\ (0::real) < d \wedge (\forall x'::(real,\ unit)\ cart.\ distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge has\_vector\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_vector\_derivative\ g\ f'\ (at\ x)$

thm HAS_VECTOR_DERIVATIVE_TRANSFORM_WITHIN_OPEN:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)\ x::(real,\ unit)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ x\ s \wedge (\forall y::(real,\ unit)\ cart.\ IN\ y\ s \longrightarrow f\ y = g\ y) \wedge has\_vector\_derivative\ f\ (?f'::(real,\ ?'a::type)\ cart)\ (at\ x) \longrightarrow has\_vector\_derivative\ g\ ?f'\ (at\ x)$

thm VECTOR_DIFF_CHAIN_AT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ unit)\ cart)\ (g'::(real,\ ?'a::type)\ cart)\ x::(real,\ unit)\ cart.\ has\_vector\_derivative\ f\ f'\ (at\ x) \wedge has\_vector\_derivative\ g\ g'\ (at\ (f\ x)) \longrightarrow has\_vector\_derivative\ (g \circ f)\ (\%\ (HOL\_Light\_Import.drop\ f')\ g')\ (at\ x)$

thm VECTOR_DIFF_CHAIN_WITHIN:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ unit)\ cart)\ (g'::(real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)\ x::(real,\ unit)\ cart.\ has\_vector\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge has\_vector\_derivative\ g\ g'\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow has\_vector\_derivative\ (g \circ f)\ (\%\ (HOL\_Light\_Import.drop\ f')\ g')\ (within\ (at\ x)\ s)$

thm CONVEX_ON_DERIVATIVE_SECANT_IMP:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\_on$

$f\ s \wedge SUBSET\ (closed\_segment\ [(x,\ y)])\ s \wedge has\_derivative\ (lift \circ f)\ (lift \circ f')\ (within\ (at\ x)\ s) \longrightarrow f'\ (vector\_sub\ y\ x) \leq f\ y - f\ x$

thm CONVEX_ON_SECANT_DERIVATIVE_IMP:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\_on\ f\ s \wedge SUBSET\ (closed\_segment\ [(x,\ y)])\ s \wedge has\_derivative\ (lift \circ f)\ (lift \circ f')\ (within\ (at\ y)\ s) \longrightarrow f\ y - f\ x \leq f'\ (vector\_sub\ y\ x)$

thm CONVEX_ON_DERIVATIVES_IMP:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'x::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'y::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ convex\_on\ f\ s \wedge SUBSET\ (closed\_segment\ [(x,\ y)])\ s \wedge has\_derivative\ (lift \circ f)\ (lift \circ f'x)\ (within\ (at\ x)\ s) \wedge has\_derivative\ (lift \circ f)\ (lift \circ f'y)\ (within\ (at\ y)\ s) \longrightarrow f'x\ (vector\_sub\ y\ x) \leq f'y\ (vector\_sub\ y\ x)$

thm CONVEX_ON_DERIVATIVES:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ (lift \circ f)\ (lift \circ f'\ x)\ (within\ (at\ x)\ s)) \longrightarrow convex\_on\ f\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f'\ x\ (vector\_sub\ y\ x) \leq f'\ y\ (vector\_sub\ y\ x))$

thm CONVEX_ON_DERIVATIVE_SECANT:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ (lift \circ f)\ (lift \circ f'\ x)\ (within\ (at\ x)\ s)) \longrightarrow convex\_on\ f\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f'\ x\ (vector\_sub\ y\ x) \leq f\ y - f\ x)$

thm CONVEX_ON_SECANT_DERIVATIVE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow has\_derivative\ (lift \circ f)\ (lift \circ f'\ x)\ (within\ (at\ x)\ s)) \longrightarrow convex\_on\ f\ s = (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f\ y - f\ x \leq f'\ y\ (vector\_sub\ y\ x))$

thm INTERIOR_SUBSET_UNION_INTERVALS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart \Rightarrow bool)\ j::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ i = closed\_interval\ [(a,\ b)]) \wedge (\exists (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ j = closed\_interval\ [(c,\ d)]) \wedge interior\ j \neq EMPTY \wedge SUBSET\ i\ (HOL\_Light\_Import.UNION\ j\ s) \wedge HOL\_Light\_Import.INTER\ (interior\ i)\ (interior\ j) = EMPTY \longrightarrow SUBSET\ (interior\ i)\ (interior\ s)$

thm INTER_INTERIOR_UNIONS_INTERVALS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge HOL\_Light\_Import.open\ s \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ f \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ t = closed\_interval\ [(a,\ b)])) \wedge (\forall t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ t\ f \longrightarrow HOL\_Light\_Import.INTER\ s\ (interior\ t) = EMPTY) \longrightarrow HOL\_Light\_Import.INTER\ s\ (interior\ (UNIONS\ f)) = EMPTY$

thm ITERATE_NONZERO_IMAGE_LEMMA:

$\forall (op::?'c::type \Rightarrow ?'c::type \Rightarrow ?'c::type)\ (s::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow ?'c::type)\ a::?'a::type.\ monoidal\ op \wedge FINITE\ s \wedge g\ a = neutral\ op \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \wedge x \neq y \longrightarrow g\ (f\ x) = neutral\ op) \longrightarrow iterate\ op\ (GSPEC\ (\lambda GEN\%PVAR\%1622::?'a::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%1622\ (IN\ x\ s \wedge f\ x \neq a)\ (f\ x)))\ g = iterate\ op\ s\ (g \circ f)$

thm DEF_interval_upperbound:

$interval\_upperbound = (\lambda\_1542959::(real,\ ?'a::type)\ cart \Rightarrow bool.\ lambda\ (\lambda i::nat.\ HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%1623::real.\ \exists a::real.\ SETSPEC\ GEN\%PVAR\%1623\ (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_1542959 \wedge \$\ x\ i = a)\ a))))$

thm interval_upperbound:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ interval\_upperbound\ s = lambda\ (\lambda i::nat.\ HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%1623::real.\ \exists a::real.\ SETSPEC\ GEN\%PVAR\%1623\ (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge \$\ x\ i = a)\ a)))$

thm DEF_interval_lowerbound:

$interval\_lowerbound = (\lambda\_1542964::(real,\ ?'a::type)\ cart \Rightarrow bool.\ lambda\ (\lambda i::nat.\ HOL\_Light\_Import.inf\ (GSPEC\ (\lambda GEN\%PVAR\%1624::real.\ \exists a::real.\ SETSPEC\ GEN\%PVAR\%1624\ (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ \_1542964 \wedge \$\ x\ i = a)\ a))))$

thm interval_lowerbound:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ interval\_lowerbound\ s = lambda\ (\lambda i::nat.\ HOL\_Light\_Import.inf\ (GSPEC\ (\lambda GEN\%PVAR\%1624::real.\ \exists a::real.\ SETSPEC\ GEN\%PVAR\%1624\ (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge \$\ x\ i = a)\ a)))$

thm INTERVAL_UPPERBOUND:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ b\ i) \longrightarrow interval\_upperbound\ (closed\_interval\ [(a,\ b)]) = b$

thm INTERVAL_LOWERBOUND:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ b\ i) \longrightarrow interval\_lowerbound\ (closed\_interval\ [(a,\ b)]) = a$

thm INTERVAL_UPPERBOUND_1:

$\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \longrightarrow interval\_upperbound\ (closed\_interval\ [(a,\ b)]) = b$

thm INTERVAL_LOWERBOUND_1:

$\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \longrightarrow interval\_lowerbound\ (closed\_interval\ [(a,\ b)]) = a$

thm DEF_content:

$content = (\lambda\_1543071::(real, ?'a::type)\ cart \Rightarrow bool.\ if\ \_1543071 = EMPTY\ then\ 0::real\ else\ product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (interval\_upperbound\ \_1543071)\ i - \$\ (interval\_lowerbound\ \_1543071)\ i))$

thm content:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ content\ s = (if\ s = EMPTY\ then\ 0::real\ else\ product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ (interval\_upperbound\ s)\ i - \$\ (interval\_lowerbound\ s)\ i))$

thm CONTENT_CLOSED_INTERVAL:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i \leq \$\ b\ i) \longrightarrow content\ (closed\_interval\ [(a,\ b)]) = product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\lambda i::nat.\ \$\ b\ i - \$\ a\ i)$

thm CONTENT_1:

$\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ b \longrightarrow content\ (closed\_interval\ [(a,\ b)]) = HOL\_Light\_Import.drop\ b - HOL\_Light\_Import.drop\ a$

thm CONTENT_UNIT:

$content\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) = (1::real)$

thm CONTENT_UNIT_1:

$content\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]) = (1::real)$

thm CONTENT_POS_LE:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ (0::real) \leq content\ (closed\_interval\ [(a,\ b)])$

thm CONTENT_POS_LT:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ a i < $ b i) ⟶ (0::real) < content (closed_interval [(a, b)])

thm CONTENT_POS_LT_1:

∀ (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.drop a < HOL_Light_Import.drop b ⟶ (0::real) < content (closed_interval [(a, b)])

thm CONTENT_EQ_0_GEN:

∀ s::(real, ?'a::type) cart ⇒ bool. bounded s ⟶ (content s = (0::real)) = (∃ (i::nat) a::real. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ $ x i = a))

thm CONTENT_EQ_0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (content (closed_interval [(a, b)]) = (0::real)) = (∃ i≥1::nat. i ≤ dimindex HOL_Light_Import.UNIV ∧ $ b i ≤ $ a i)

thm CONTENT_0_SUBSET_GEN:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. SUBSET s t ∧ bounded t ∧ content t = (0::real) ⟶ content s = (0::real)

thm CONTENT_0_SUBSET:

∀ (s::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. SUBSET s (closed_interval [(a, b)]) ∧ content (closed_interval [(a, b)]) = (0::real) ⟶ content s = (0::real)

thm CONTENT_CLOSED_INTERVAL_CASES:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. content (closed_interval [(a, b)]) = (if ∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ a i ≤ $ b i then product (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) (λi::nat. $ b i − $ a i) else (0::real))

thm CONTENT_EQ_0_INTERIOR:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (content (closed_interval [(a, b)]) = (0::real)) = (interior (closed_interval [(a, b)]) = EMPTY)

thm CONTENT_EQ_0_1:

∀ (a::(real, unit) cart) b::(real, unit) cart. (content (closed_interval [(a, b)]) = (0::real)) = (HOL_Light_Import.drop b ≤ HOL_Light_Import.drop a)

thm CONTENT_POS_LT_EQ:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. ((0::real) < content (closed_interval [(a, b)])) = (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ $ a i < $ b i)

thm CONTENT_EMPTY:

content EMPTY = (*0*::*real*)

thm CONTENT_SUBSET:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) ⟶ *content* (*closed_interval* [(*a*, *b*)]) ≤ *content* (*closed_interval* [(*c*, *d*)])

thm CONTENT_LT_NZ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ((*0*::*real*) < *content* (*closed_interval* [(*a*, *b*)])) = (*content* (*closed_interval* [(*a*, *b*)]) ≠ (*0*::*real*))

thm INTERVAL_BOUNDS_NULL_1:

∀ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *interval_upperbound* (*closed_interval* [(*a*, *b*)]) = *interval_lowerbound* (*closed_interval* [(*a*, *b*)])

thm INTERVAL_BOUNDS_EMPTY_1:

interval_upperbound EMPTY = *interval_lowerbound EMPTY*

thm CONTENT_PASTECART:

∀ (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *content* (*closed_interval* [(*pastecart a c*, *pastecart b d*)]) = *content* (*closed_interval* [(*a*, *b*)]) ∗ *content* (*closed_interval* [(*c*, *d*)])

thm DEF_gauge:

gauge = (λ*_1543905*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*_1543905 x*) ∧ *HOL_Light_Import.open* (*_1543905 x*))

thm gauge:

∀ *d*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. *gauge d* = (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*d x*) ∧ *HOL_Light_Import.open* (*d x*))

thm GAUGE_BALL_DEPENDENT:

∀ *e*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *e x*) ⟶ *gauge* (λ*x*::(*real*, *?'a*::*type*) *cart*. *ball* (*x*, *e x*))

thm GAUGE_BALL:

∀ *e*>*0*::*real*. *gauge* (λ*x*::(*real*, *?'a*::*type*) *cart*. *ball* (*x*, *e*))

thm GAUGE_TRIVIAL:

gauge (λ*x*::(*real*, *?'a*::*type*) *cart*. *ball* (*x*, *1*::*real*))

thm GAUGE_INTER:

∀ (*d1*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*) *d2*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. *gauge d1* ∧ *gauge d2* ⟶ *gauge* (λ*x*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.INTER* (*d1 x*) (*d2 x*))

thm GAUGE_INTERS:

$\forall\, s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall\, d::?'b::type.\ IN\ d\ s \longrightarrow gauge\ ((?f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool)\ d)) \longrightarrow gauge\ (\lambda x::(real,\ ?'a::type)\ cart.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%1626::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, d::?'b::type.\ SETSPEC\ GEN\%PVAR\%1626\ (IN\ d\ s)\ (?f\ d\ x))))$

thm GAUGE_EXISTENCE_LEMMA:

$(\forall\, x::?'a::type.\ \exists\, d::real.\ (?p::?'a::type \Rightarrow bool)\ x \longrightarrow (0::real) < d \wedge (?q::real \Rightarrow ?'a::type \Rightarrow bool)\ d\ x) = (\forall\, x::?'a::type.\ \exists\, d{>}0::real.\ ?p\ x \longrightarrow ?q\ d\ x)$

thm DEF_division_of:

$division\_of = (\lambda(\_1543944::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ \_1543945::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ \_1543944 \wedge (\forall\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ \_1543944 \longrightarrow SUBSET\ k\ \_1543945 \wedge k \neq EMPTY \wedge (\exists\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \wedge (\forall\, (k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k1\ \_1543944 \wedge IN\ k2\ \_1543944 \wedge k1 \neq k2 \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k1)\ (interior\ k2) = EMPTY) \wedge UNIONS\ \_1543944 = \_1543945)$

thm division_of:

$\forall\, (s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ s\ i = (FINITE\ s \wedge (\forall\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ s \longrightarrow SUBSET\ k\ i \wedge k \neq EMPTY \wedge (\exists\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \wedge (\forall\, (k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k1\ s \wedge IN\ k2\ s \wedge k1 \neq k2 \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k1)\ (interior\ k2) = EMPTY) \wedge UNIONS\ s = i)$

thm DIVISION_OF:

$division\_of\ (?s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (?i::(real,\ ?'a::type)\ cart \Rightarrow bool) = (FINITE\ ?s \wedge (\forall\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ ?s \longrightarrow k \neq EMPTY \wedge (\exists\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \wedge (\forall\, (k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k1\ ?s \wedge IN\ k2\ ?s \wedge k1 \neq k2 \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k1)\ (interior\ k2) = EMPTY) \wedge UNIONS\ ?s = ?i)$

thm DIVISION_OF_FINITE:

$\forall\, (s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ s\ i \longrightarrow FINITE\ s$

thm DIVISION_OF_SELF:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow division\_of\ (INSERT\ (closed\_interval\ [(a,\ b)])\ EMPTY)\ (closed\_interval\ [(a,\ b)])$

thm DIVISION_OF_TRIVIAL:

$\forall\, s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ s\ EMPTY = (s = EMPTY)$

thm EMPTY_DIVISION_OF:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ EMPTY\ s = (s = EMPTY)$

thm DIVISION_OF_SING:

$\forall\, (s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ division\_of\ s\ (closed\_interval\ [(a,\ a)]) = (s = INSERT\ (closed\_interval\ [(a,\ a)])\ EMPTY)$

thm ELEMENTARY_EMPTY:

$\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ EMPTY$

thm ELEMENTARY_INTERVAL:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ \exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (closed\_interval\ [(a,\ b)])$

thm DIVISION_CONTAINS:

$\forall\, (s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ s\ i \longrightarrow (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ i \longrightarrow (\exists\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ k \wedge IN\ k\ s))$

thm FORALL_IN_DIVISION:

$\forall\, (P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ d\ i \longrightarrow (\forall\, x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ d \longrightarrow P\ x) = (\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ (closed\_interval\ [(a,\ b)])\ d \longrightarrow P\ (closed\_interval\ [(a,\ b)]))$

thm FORALL_IN_DIVISION_NONEMPTY:

$\forall\, (P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ d\ i \longrightarrow (\forall\, x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ d \longrightarrow P\ x) = (\forall\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ IN\ (closed\_interval\ [(a,\ b)])\ d \wedge closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow P\ (closed\_interval\ [(a,\ b)]))$

thm DIVISION_OF_SUBSET:

$\forall\, (p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ q::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (UNIONS\ p) \wedge SUBSET\ q\ p \longrightarrow division\_of\ q\ (UNIONS\ q)$

thm DIVISION_OF_UNION_SELF:

$\forall\, (p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ p\ s \longrightarrow division\_of\ p\ (UNIONS\ p)$

thm DIVISION_OF_CONTENT_0:

$\forall\, (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \wedge division\_of\ d$

$(closed\_interval\ [(a,\ b)]) \longrightarrow (\forall\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow content\ k = (0::real))$

thm DIVISION_INTER:

$\forall\,(s1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (s2::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (p1::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ p2::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p1\ s1 \wedge division\_of\ p2\ s2 \longrightarrow division\_of\ (GSPEC\ (\lambda GEN\%PVAR\%1630::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\,(k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1630\ (IN\ k1\ p1 \wedge IN\ k2\ p2 \wedge HOL\_Light\_Import.INTER\ k1\ k2 \neq EMPTY)\ (HOL\_Light\_Import.INTER\ k1\ k2)))\ (HOL\_Light\_Import.INTER\ s1\ s2)$

thm DIVISION_INTER_1:

$\forall\,(d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ division\_of\ d\ i \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ i \longrightarrow division\_of\ (GSPEC\ (\lambda GEN\%PVAR\%1632::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1632\ (IN\ k\ d \wedge HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ k \neq EMPTY)\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ k)))\ (closed\_interval\ [(a,\ b)])$

thm ELEMENTARY_INTER:

$\forall\,(s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ s) \wedge (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ t) \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (HOL\_Light\_Import.INTER\ s\ t))$

thm ELEMENTARY_INTERS:

$\forall\, f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\, f \wedge f \neq EMPTY \wedge (\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ s)) \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (INTERS\ f))$

thm DIVISION_DISJOINT_UNION:

$\forall\,(s1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (s2::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (p1::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ p2::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p1\ s1 \wedge division\_of\ p2\ s2 \wedge HOL\_Light\_Import.INTER\ (interior\ s1)\ (interior\ s2) = EMPTY \longrightarrow division\_of\ (HOL\_Light\_Import.UNION\ p1\ p2)\ (HOL\_Light\_Import.UNION\ s1\ s2)$

thm PARTIAL_DIVISION_EXTEND_1:

$\forall\,(a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_interval\ [(c,\ d)])\ (closed\_interval\ [(a,\ b)]) \wedge closed\_interval\ [(c,\ d)] \neq EMPTY \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge IN\ (closed\_interval\ [(c,\ d)])\ p)$

thm PARTIAL_DIVISION_EXTEND_INTERVAL:

$\forall$ (*p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *division_of p* (*UNIONS p*) $\wedge$ *SUBSET* (*UNIONS p*) (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\exists$ *q*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *SUBSET p q* $\wedge$ *division_of q* (*closed_interval* [(*a*, *b*)]))

thm ELEMENTARY_BOUNDED:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\exists$ *p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *division_of p s*) $\longrightarrow$ *bounded s*

thm ELEMENTARY_SUBSET_INTERVAL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\exists$ *p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *division_of p s*) $\longrightarrow$ ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *SUBSET s* (*closed_interval* [(*a*, *b*)]))

thm DIVISION_UNION_INTERVALS_EXISTS:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *closed_interval* [(*a*, *b*)] $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *division_of* (*INSERT* (*closed_interval* [(*a*, *b*)]) *p*) (*HOL_Light_Import.UNION* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)])))

thm DIVISION_OF_UNIONS:

$\forall$ *f*::(((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FINITE f* $\wedge$ ($\forall$ *p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *IN p f* $\longrightarrow$ *division_of p* (*UNIONS p*)) $\wedge$ ($\forall$ (*k1*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *k2*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN k1* (*UNIONS f*) $\wedge$ *IN k2* (*UNIONS f*) $\wedge$ *k1* $\neq$ *k2* $\longrightarrow$ *HOL_Light_Import.INTER* (*interior k1*) (*interior k2*) = *EMPTY*) $\longrightarrow$ *division_of* (*UNIONS f*) (*UNIONS* (*UNIONS f*))

thm ELEMENTARY_UNION_INTERVAL_STRONG:

$\forall$ (*p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *division_of p* (*UNIONS p*) $\longrightarrow$ ($\exists$ *q*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *SUBSET p q* $\wedge$ *division_of q* (*HOL_Light_Import.UNION* (*closed_interval* [(*a*, *b*)]) (*UNIONS p*)))

thm ELEMENTARY_UNION_INTERVAL:

$\forall$ (*p*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *division_of p* (*UNIONS p*) $\longrightarrow$ ($\exists$ *q*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *division_of q* (*HOL_Light_Import.UNION* (*closed_interval* [(*a*, *b*)]) (*UNIONS p*)))

thm ELEMENTARY_UNIONS_INTERVALS:

$\forall$ *f*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FINITE f* $\wedge$ ($\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN s f* $\longrightarrow$ ($\exists$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*.

$s = closed\_interval\ [(a,\ b)])) \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (UNIONS\ f))$

thm ELEMENTARY_UNION:

$\forall\,(s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ s) \wedge (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ t) \longrightarrow (\exists\, p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ p\ (HOL\_Light\_Import.UNION\ s\ t))$

thm PARTIAL_DIVISION_EXTEND:

$\forall\,(p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (q::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ p\ s \wedge division\_of\ q\ t \wedge SUBSET\ s\ t \longrightarrow (\exists\, r::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SUBSET\ p\ r \wedge division\_of\ r\ t)$

thm INTERVAL_SUBDIVISION:

$\forall\,(a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow division\_of\ (IMAGE\ (\lambda s::nat \Rightarrow bool.\ closed\_interval\ [(lambda\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ \$\ c\ i\ else\ \$\ a\ i),\ lambda\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ \$\ b\ i\ else\ \$\ c\ i))])\ (GSPEC\ (\lambda GEN\%PVAR\%1643::nat \Rightarrow bool.\ \exists\, s::nat \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1643\ (SUBSET\ s\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)))\ s)))\ (closed\_interval\ [(a,\ b)])$

thm DIVISION_OF_NONTRIVIAL:

$\forall\,(s::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ division\_of\ s\ (closed\_interval\ [(a,\ b)]) \wedge content\ (closed\_interval\ [(a,\ b)]) \neq (0::real) \longrightarrow division\_of\ (GSPEC\ (\lambda GEN\%PVAR\%1645::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1645\ (IN\ k\ s \wedge content\ k \neq (0::real))\ k))\ (closed\_interval\ [(a,\ b)])$

thm DIVISION_OF_AFFINITY:

$\forall\,(d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::real)\ c::(real,\ ?'a::type)\ cart.\ division\_of\ (IMAGE\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c))\ d)\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c)\ s) = (if\ m = (0::real)\ then\ if\ s = EMPTY\ then\ d = EMPTY\ else\ d \neq EMPTY \wedge (\forall\, k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow k \neq EMPTY)\ else\ division\_of\ d\ s)$

thm DIVISION_OF_TRANSLATION:

$\forall\,(d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ (IMAGE\ (IMAGE\ (vector\_add\ (?a::(real,\ ?'a::type)\ cart)))\ d)\ (IMAGE\ (vector\_add\ ?a)\ s) = division\_of\ d\ s$

thm DIVISION_OF_REFLECT:

$\forall\,(d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ (IMAGE\ (IMAGE\ vector\_neg)\ d)\ (IMAGE\ vector\_neg\ s) = division\_of\ d\ s$

thm ELEMENTARY_COMPACT:

∀ s::(real, ?'a::type) cart ⇒ bool. (∃ d::((real, ?'a::type) cart ⇒ bool) ⇒ bool. division_of d s) ⟶ compact s

thm DEF_tagged_partial_division_of:

tagged_partial_division_of = (λ(_1574234::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool) _1574235::(real, ?'a::type) cart ⇒ bool. FINITE _1574234 ∧ (∀ (x::(real, ?'a::type) cart) k::(real, ?'a::type) cart ⇒ bool. IN (x, k) _1574234 ⟶ IN x k ∧ SUBSET k _1574235 ∧ (∃ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. k = closed_interval [(a, b)])) ∧ (∀ (x1::(real, ?'a::type) cart) (k1::(real, ?'a::type) cart ⇒ bool) (x2::(real, ?'a::type) cart) k2::(real, ?'a::type) cart ⇒ bool. IN (x1, k1) _1574234 ∧ IN (x2, k2) _1574234 ∧ (x1, k1) ≠ (x2, k2) ⟶ HOL_Light_Import.INTER (interior k1) (interior k2) = EMPTY))

thm tagged_partial_division_of:

∀ (i::(real, ?'a::type) cart ⇒ bool) s::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool. tagged_partial_division_of s i = (FINITE s ∧ (∀ (x::(real, ?'a::type) cart) k::(real, ?'a::type) cart ⇒ bool. IN (x, k) s ⟶ IN x k ∧ SUBSET k i ∧ (∃ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. k = closed_interval [(a, b)])) ∧ (∀ (x1::(real, ?'a::type) cart) (k1::(real, ?'a::type) cart ⇒ bool) (x2::(real, ?'a::type) cart) k2::(real, ?'a::type) cart ⇒ bool. IN (x1, k1) s ∧ IN (x2, k2) s ∧ (x1, k1) ≠ (x2, k2) ⟶ HOL_Light_Import.INTER (interior k1) (interior k2) = EMPTY))

thm DEF_tagged_division_of:

tagged_division_of = (λ(_1574246::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool) _1574247::(real, ?'a::type) cart ⇒ bool. tagged_partial_division_of _1574246 _1574247 ∧ UNIONS (GSPEC (λGEN%PVAR%1646::(real, ?'a::type) cart ⇒ bool. ∃ k::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1646 (∃ x::(real, ?'a::type) cart. IN (x, k) _1574246) k)) = _1574247)

thm tagged_division_of:

∀ (s::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool) i::(real, ?'a::type) cart ⇒ bool. tagged_division_of s i = (tagged_partial_division_of s i ∧ UNIONS (GSPEC (λGEN%PVAR%1646::(real, ?'a::type) cart ⇒ bool. ∃ k::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%1646 (∃ x::(real, ?'a::type) cart. IN (x, k) s) k)) = i)

thm TAGGED_DIVISION_OF_FINITE:

∀ (s::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool) i::(real, ?'a::type) cart ⇒ bool. tagged_division_of s i ⟶ FINITE s

thm TAGGED_DIVISION_OF:

tagged_division_of (?s::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) ⇒ bool) (?i::(real, ?'a::type) cart ⇒ bool) = (FINITE ?s ∧ (∀ (x::(real, ?'a::type) cart) k::(real, ?'a::type) cart ⇒ bool. IN (x, k) ?s ⟶ IN x k ∧ SUBSET k

?i ∧ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *k* = *closed_interval* [(*a*, *b*)])) ∧ (∀ (*x1*::(*real*, *?'a*::*type*) *cart*) (*k1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*x2*::(*real*, *?'a*::*type*) *cart*) *k2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* (*x1*, *k1*) *?s* ∧ *IN* (*x2*, *k2*) *?s* ∧ (*x1*, *k1*) ≠ (*x2*, *k2*) ⟶ *HOL_Light_Import.INTER* (*interior k1*) (*interior k2*) = *EMPTY*) ∧ *UNIONS* (*GSPEC* (λ*GEN%PVAR%1647*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%1647* (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN* (*x*, *k*) *?s*) *k*)) = *?i*)

thm DIVISION_OF_TAGGED_DIVISION:

∀ (*s*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_division_of s i* ⟶ *division_of* (*IMAGE snd s*) *i*

thm PARTIAL_DIVISION_OF_TAGGED_DIVISION:

∀ (*s*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of s i* ⟶ *division_of* (*IMAGE snd s*) (*UNIONS* (*IMAGE snd s*))

thm TAGGED_PARTIAL_DIVISION_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of s i* ∧ *SUBSET t s* ⟶ *tagged_partial_division_of t i*

thm VSUM_OVER_TAGGED_PARTIAL_DIVISION_LEMMA:

∀ (*d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of p i* ∧ (∀ (*u*::(*real*, *?'b*::*type*) *cart*) *v*::(*real*, *?'b*::*type*) *cart*. *closed_interval* [(*u*, *v*)] ≠ *EMPTY* ∧ *content* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*) ⟶ *d* (*closed_interval* [(*u*, *v*)]) = *vec* (*0*::*nat*)) ⟶ *vsum p* (*GABS* (λ*f*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (*d k*))) = *vsum* (*IMAGE snd p*) *d*

thm VSUM_OVER_TAGGED_DIVISION_LEMMA:

∀ (*d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *tagged_division_of p i* ∧ (∀ (*u*::(*real*, *?'b*::*type*) *cart*) *v*::(*real*, *?'b*::*type*) *cart*. *closed_interval* [(*u*, *v*)] ≠ *EMPTY* ∧ *content* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*) ⟶ *d* (*closed_interval* [(*u*, *v*)]) = *vec* (*0*::*nat*)) ⟶ *vsum p* (*GABS* (λ*f*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (*d k*))) = *vsum* (*IMAGE snd p*) *d*

thm SUM_OVER_TAGGED_PARTIAL_DIVISION_LEMMA:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *real*) (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of* *p i* ∧ (∀(*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *closed_interval* [(*u*, *v*)] ≠ *EMPTY* ∧ *content* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*) ⟶ *d* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*)) ⟶ *sum p* (*GABS* (λ*f*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *real*. ∀(*x*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (*d k*))) = *sum* (*IMAGE snd p*) *d*

thm SUM_OVER_TAGGED_DIVISION_LEMMA:

∀(*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *real*) (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_division_of p i* ∧ (∀(*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *closed_interval* [(*u*, *v*)] ≠ *EMPTY* ∧ *content* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*) ⟶ *d* (*closed_interval* [(*u*, *v*)]) = (*0*::*real*)) ⟶ *sum p* (*GABS* (λ*f*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *real*. ∀(*x*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (*d k*))) = *sum* (*IMAGE snd p*) *d*

thm TAG_IN_INTERVAL:

∀(*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_division_of p i* ∧ *IN* (*?x*::(*real*, *?'a*::*type*) *cart*, *k*) *p* ⟶ *IN ?x i*

thm TAGGED_DIVISION_OF_EMPTY:

tagged_division_of EMPTY EMPTY

thm TAGGED_PARTIAL_DIVISION_OF_TRIVIAL:

∀*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_partial_division_of p EMPTY* = (*p* = *EMPTY*)

thm TAGGED_DIVISION_OF_TRIVIAL:

∀*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p EMPTY* = (*p* = *EMPTY*)

thm TAGGED_DIVISION_OF_SELF:

∀(*x*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ *tagged_division_of* (*INSERT* (*x*, *closed_interval* [(*a*, *b*)]) *EMPTY*) (*closed_interval* [(*a*, *b*)])

thm TAGGED_DIVISION_UNION:

∀(*s1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p1*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *p2*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p1 s1* ∧ *tagged_division_of p2 s2* ∧ *HOL_Light_Import.INTER* (*interior s1*) (*interior s2*) = *EMPTY* ⟶ *tagged_division_of* (*HOL_Light_Import.UNION p1 p2*) (*HOL_Light_Import.UNION s1 s2*)

thm TAGGED_DIVISION_UNIONS:

∀(*iset*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *pfn*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE* *iset* ∧ (∀ *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN i iset* ⟶ *tagged_division_of* (*pfn* *i*) *i*) ∧ (∀ (*i1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *i2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN i1 iset* ∧ *IN i2 iset* ∧ *i1* ≠ *i2* ⟶ *HOL_Light_Import.INTER* (*interior* *i1*) (*interior i2*) = *EMPTY*) ⟶ *tagged_division_of* (*UNIONS* (*IMAGE pfn* *iset*)) (*UNIONS iset*)

thm TAGGED_PARTIAL_DIVISION_OF_UNION_SELF:

∀ (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of p s* ⟶ *tagged_division_of p* (*UNIONS* (*IMAGE snd p*))

thm TAGGED_DIVISION_OF_UNION_SELF:

∀ (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_division_of p s* ⟶ *tagged_division_of p* (*UNIONS* (*IMAGE snd p*))

thm TAGGED_DIVISION_OF_ALT:

∀ (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_division_of p s* = (*tagged_partial_division_of p* *s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (∃ (*t*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* (*t*, *k*) *p* ∧ *IN x k*)))

thm TAGGED_DIVISION_OF_ANOTHER:

∀ (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of* *p s'* ∧ (∀ (*t*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* (*t*, *k*) *p* ⟶ *SUBSET k s*) ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ (∃ (*t*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN* (*t*, *k*) *p* ∧ *IN x k*)) ⟶ *tagged_division_of p s*

thm TAGGED_PARTIAL_DIVISION_OF_SUBSET:

∀ (*p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *tagged_partial_division_of* *p s* ∧ *SUBSET s t* ⟶ *tagged_partial_division_of p t*

thm TAGGED_DIVISION_OF_NONTRIVIAL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *tagged_division_of s* (*closed_interval* [(*a*, *b*)]) ∧ *content* (*closed_interval* [(*a*, *b*)]) ≠ (*0*::*real*) ⟶ *tagged_division_of* (*GSPEC* (λ*GEN%PVAR%1652*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*). ∃ (*x*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%1652* (*IN* (*x*, *k*) *s* ∧ *content k* ≠ (*0*::*real*)) (*x*, *k*))) (*closed_interval* [(*a*, *b*)])

thm DEF_fine:

fine = (λ(*_1581004*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) *_1581005*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*. ∀(*x*::*?'b*::*type*) *k*::*?'a*::*type* ⇒ *bool*. *IN* (*x*, *k*) *_1581005* ⟶ *SUBSET k* (*_1581004 x*))

thm fine:

∀(*s*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *d*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *fine d s* = (∀(*x*::*?'b*::*type*) *k*::*?'a*::*type* ⇒ *bool*. *IN* (*x*, *k*) *s* ⟶ *SUBSET k* (*d x*))

thm FINE_INTER:

∀(*p*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*d1*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) *d2*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *fine* (λ*x*::*?'b*::*type*. *HOL_Light_Import.INTER* (*d1 x*) (*d2 x*)) *p* = (*fine d1 p* ∧ *fine d2 p*)

thm FINE_INTERS:

∀(*f*::*?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) (*s*::*?'c*::*type* ⇒ *bool*) *p*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *fine* (λ*x*::*?'b*::*type*. *INTERS* (*GSPEC* (λ*GEN%PVAR%1653*::*?'a*::*type* ⇒ *bool*. ∃*d*::*?'c*::*type*. *SETSPEC GEN%PVAR%1653* (*IN d s*) (*f d x*)))) *p* = (∀*d*::*?'c*::*type*. *IN d s* ⟶ *fine* (*f d*) *p*)

thm FINE_UNION:

∀(*d*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) (*p1*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *p2*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *fine d p1* ∧ *fine d p2* ⟶ *fine d* (*HOL_Light_Import.UNION p1 p2*)

thm FINE_UNIONS:

∀(*d*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) *ps*::(*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*) ⇒ *bool*. (∀*p*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *IN p ps* ⟶ *fine d p*) ⟶ *fine d* (*UNIONS ps*)

thm FINE_SUBSET:

∀(*d*::*?'b*::*type* ⇒ *?'a*::*type* ⇒ *bool*) (*p*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *q*::*?'b*::*type* × (*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *SUBSET p q* ∧ *fine d q* ⟶ *fine d p*

thm DEF_has_integral_compact_interval:

has_integral_compact_interval = (λ(*_1581092*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*_1581093*::(*real*, *?'a*::*type*) *cart*) *_1581094*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. ∀*e*>*0*::*real*. ∃*d*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ *bool*. *gauge d* ∧ (∀*p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p _1581094* ∧ *fine d p* ⟶ *vector_norm* (*vector_sub* (*vsum p* (*GABS* (λ*f*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀(*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (% (*content k*) (*_1581092 x*))))) *_1581093*) < *e*))

thm has_integral_compact_interval:

∀(*i*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *has_integral_compact_interval f y i* = (∀*e*>*0*::*real*.

∃ d::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart ⇒ bool. gauge d ∧ (∀ p::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ bool. tagged_division_of p i ∧ fine d p ⟶ vector_norm (vector_sub (vsum p (GABS (λfa::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (fa (x, k)) (% (content k) (f x))))) y) < e))

thm DEF_has_integral:

has_integral = (λ(_1581113::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (_1581114::(real, ?'a::type) cart) _1581115::(real, ?'b::type) cart ⇒ bool. if ∃ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. _1581115 = closed_interval [(a, b)] then has_integral_compact_interval _1581113 _1581114 _1581115 else ∀ e>0::real. ∃ B>0::real. ∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. SUBSET (ball (vec (0::nat), B)) (closed_interval [(a, b)]) ⟶ (∃ z::(real, ?'a::type) cart. has_integral_compact_interval (λx::(real, ?'b::type) cart. if IN x _1581115 then _1581113 x else vec (0::nat)) z (closed_interval [(a, b)]) ∧ vector_norm (vector_sub z _1581114) < e))

thm has_integral_def:

∀ (i::(real, ?'b::type) cart ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) y::(real, ?'a::type) cart. has_integral f y i = (if ∃ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. i = closed_interval [(a, b)] then has_integral_compact_interval f y i else ∀ e>0::real. ∃ B>0::real. ∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. SUBSET (ball (vec (0::nat), B)) (closed_interval [(a, b)]) ⟶ (∃ z::(real, ?'a::type) cart. has_integral_compact_interval (λx::(real, ?'b::type) cart. if IN x i then f x else vec (0::nat)) z (closed_interval [(a, b)]) ∧ vector_norm (vector_sub z y) < e))

thm has_integral:

has_integral (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (?y::(real, ?'a::type) cart) (closed_interval [(?a::(real, ?'b::type) cart, ?b::(real, ?'b::type) cart)]) = (∀ e>0::real. ∃ d::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart ⇒ bool. gauge d ∧ (∀ p::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ bool. tagged_division_of p (closed_interval [(?a, ?b)]) ∧ fine d p ⟶ vector_norm (vector_sub (vsum p (GABS (λf::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (f (x, k)) (% (content k) (?f x))))) ?y) < e))

thm has_integral_alt:

has_integral (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (?y::(real, ?'a::type) cart) (?i::(real, ?'b::type) cart ⇒ bool) = (if ∃ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. ?i = closed_interval [(a, b)] then has_integral ?f ?y ?i else ∀ e>0::real. ∃ B>0::real. ∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. SUBSET (ball (vec (0::nat), B)) (closed_interval [(a, b)]) ⟶ (∃ z::(real, ?'a::type) cart. has_integral (λx::(real, ?'b::type) cart. if IN x ?i then ?f x else vec (0::nat)) z (closed_interval [(a, b)]) ∧ vector_norm (vector_sub z ?y) < e))

thm DEF_integrable_on:

$integrable\_on = (\lambda(\_1581595::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ \_1581596::(real, ?'b::type)\ cart \Rightarrow bool.\ \exists\, y::(real, ?'a::type)\ cart.\ has\_integral\ \_1581595\ y\ \_1581596)$

thm integrable_on:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ i::(real, ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ i = (\exists\, y::(real, ?'a::type)\ cart.\ has\_integral\ f\ y\ i)$

thm DEF_integral:

$integral = (\lambda(\_1581607::(real, ?'b::type)\ cart \Rightarrow bool)\ \_1581608::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ SOME\ y::(real, ?'a::type)\ cart.\ has\_integral\ \_1581608\ y\ \_1581607)$

thm integral:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ i::(real, ?'b::type)\ cart \Rightarrow bool.\ integral\ i\ f = (SOME\ y::(real, ?'a::type)\ cart.\ has\_integral\ f\ y\ i)$

thm INTEGRABLE_INTEGRAL:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ i::(real, ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ i \longrightarrow has\_integral\ f\ (integral\ i\ f)\ i$

thm HAS_INTEGRAL_INTEGRABLE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (i::(real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ i\ s \longrightarrow integrable\_on\ f\ s$

thm HAS_INTEGRAL_INTEGRAL:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s = has\_integral\ f\ (integral\ s\ f)\ s$

thm VSUM_CONTENT_NULL:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, ?'b::type)\ cart)\ (b::(real, ?'b::type)\ cart)\ p::(real, ?'b::type)\ cart \times ((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ content\ (closed\_interval\ [(a, b)]) = (0::real) \land tagged\_division\_of\ p\ (closed\_interval\ [(a, b)]) \longrightarrow vsum\ p\ (GABS\ (\lambda fa::(real, ?'b::type)\ cart \times ((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart.\ \forall (x::(real, ?'b::type)\ cart)\ k::(real, ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (fa\ (x, k))\ (\%\ (content\ k)\ (f\ x)))) = vec\ (0::nat)$

thm TAGGED_DIVISION_UNIONS_EXISTS:

$\forall (d::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool)\ (iset::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ iset \land (\forall\, i::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ i\ iset \longrightarrow (\exists\, p::(real, ?'a::type)\ cart \times ((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ tagged\_division\_of\ p\ i \land fine\ d\ p)) \land (\forall (i1::(real, ?'a::type)\ cart \Rightarrow bool)\ i2::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ i1\ iset \land IN\ i2\ iset \land i1 \neq$

i2 ⟶ *HOL_Light_Import.INTER* (*interior i1*) (*interior i2*) = *EMPTY*) ∧ *UNIONS iset* = *i* ⟶ (∃ *p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p i* ∧ *fine d p*)

thm DIVISION_OF_CLOSED:

∀ (*s*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *division_of s i* ⟶ *HOL_Light_Import.closed i*

thm INTERVAL_BISECTION_STEP:

∀ *P*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *P EMPTY* ∧ (∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *P s* ∧ *P t* ∧ *HOL_Light_Import.INTER* (*interior s*) (*interior t*) = *EMPTY* ⟶ *P* (*HOL_Light_Import.UNION s t*)) ⟶ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *P* (*closed_interval* [(*a*, *b*)]) ⟶ (∃ (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. ¬ *P* (*closed_interval* [(*c*, *d*)]) ∧ (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ $ *a i* ≤ $ *c i* ∧ $ *c i* ≤ $ *d i* ∧ $ *d i* ≤ $ *b i* ∧ *real_of_nat* (*2*::*nat*) ∗ ($ *d i* − $ *c i*) ≤ $ *b i* − $ *a i*)))

thm INTERVAL_BISECTION:

∀ *P*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *P EMPTY* ∧ (∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *P s* ∧ *P t* ∧ *HOL_Light_Import.INTER* (*interior s*) (*interior t*) = *EMPTY* ⟶ *P* (*HOL_Light_Import.UNION s t*)) ⟶ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *P* (*closed_interval* [(*a*, *b*)]) ⟶ (∃ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ∧ (∀ *e*>*0*::*real*. ∃ (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*c*, *d*)]) ∧ *SUBSET* (*closed_interval* [(*c*, *d*)]) (*ball* (*x*, *e*)) ∧ *SUBSET* (*closed_interval* [(*c*, *d*)]) (*closed_interval* [(*a*, *b*)]) ∧ ¬ *P* (*closed_interval* [(*c*, *d*)]))))

thm FINE_DIVISION_EXISTS:

∀ (*g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *gauge g* ⟶ (∃ *p*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) ∧ *fine g p*)

thm HAS_INTEGRAL_UNIQUE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*i*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*k1*::(*real*, *?'a*::*type*) *cart*) *k2*::(*real*, *?'a*::*type*) *cart*. *has_integral f k1 i* ∧ *has_integral f k2 i* ⟶ *k1* = *k2*

thm INTEGRAL_UNIQUE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *has_integral f y k* ⟶ *integral k f* = *y*

thm HAS_INTEGRAL_INTEGRABLE_INTEGRAL:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*i*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *has_integral f i s* = (*integrable_on f s* ∧ *integral s f* = *i*)

thm INTEGRAL_EQ_HAS_INTEGRAL:

$\forall (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ integrable\_on\ f\ s \longrightarrow (integral\ s\ f = y) = has\_integral\ f\ y\ s$

thm HAS_INTEGRAL_IS_0:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real, ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow has\_integral\ f\ (vec\ (0::nat))\ s$

thm HAS_INTEGRAL_0:

$\forall s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ (\lambda x::(real, ?'b::type)\ cart.\ vec\ (0::nat))\ (vec\ (0::nat))\ s$

thm HAS_INTEGRAL_0_EQ:

$\forall (i::(real, ?'b::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ has\_integral\ (\lambda x::(real, ?'a::type)\ cart.\ vec\ (0::nat))\ i\ s = (i = vec\ (0::nat))$

thm HAS_INTEGRAL_LINEAR:

$\forall (f::(real, ?'c::type)\ cart \Rightarrow (real, ?'b::type)\ cart)\ (y::(real, ?'b::type)\ cart)\ (s::(real, ?'c::type)\ cart \Rightarrow bool)\ h::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ has\_integral\ f\ y\ s \wedge linear\ h \longrightarrow has\_integral\ (h \circ f)\ (h\ y)\ s$

thm HAS_INTEGRAL_CMUL:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (k::(real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ c::real.\ has\_integral\ f\ k\ s \longrightarrow has\_integral\ (\lambda x::(real, ?'b::type)\ cart.\ \%\ c\ (f\ x))\ (\%\ c\ k)\ s$

thm HAS_INTEGRAL_NEG:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (k::(real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ k\ s \longrightarrow has\_integral\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_neg\ (f\ x))\ (vector\_neg\ k)\ s$

thm HAS_INTEGRAL_ADD:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ (?k::(real, ?'a::type)\ cart)\ s \wedge has\_integral\ g\ (?l::(real, ?'a::type)\ cart)\ s \longrightarrow has\_integral\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x))\ (vector\_add\ ?k\ ?l)\ s$

thm HAS_INTEGRAL_SUB:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ (?k::(real, ?'a::type)\ cart)\ s \wedge has\_integral\ g\ (?l::(real, ?'a::type)\ cart)\ s \longrightarrow has\_integral\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ (vector\_sub\ ?k\ ?l)\ s$

thm INTEGRAL_0:

$\forall s::(real, ?'b::type) cart \Rightarrow bool. integral s (\lambda x::(real, ?'b::type) cart. vec (0::nat)) = vec (0::nat)$

thm INTEGRAL_ADD:

$\forall (f::(real, ?'d::type) cart \Rightarrow (real, ?'c::type) cart) (g::(real, ?'d::type) cart \Rightarrow (real, ?'c::type) cart) (k::?'b::type) (l::?'a::type) s::(real, ?'d::type) cart \Rightarrow bool. integrable\_on f s \wedge integrable\_on g s \longrightarrow integral s (\lambda x::(real, ?'d::type) cart. vector\_add (f x) (g x)) = vector\_add (integral s f) (integral s g)$

thm INTEGRAL_CMUL:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) (c::real) s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on f s \longrightarrow integral s (\lambda x::(real, ?'b::type) cart. \% c (f x)) = \% c (integral s f)$

thm INTEGRAL_NEG:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on f s \longrightarrow integral s (\lambda x::(real, ?'b::type) cart. vector\_neg (f x)) = vector\_neg (integral s f)$

thm INTEGRAL_SUB:

$\forall (f::(real, ?'d::type) cart \Rightarrow (real, ?'c::type) cart) (g::(real, ?'d::type) cart \Rightarrow (real, ?'c::type) cart) (k::?'b::type) (l::?'a::type) s::(real, ?'d::type) cart \Rightarrow bool. integrable\_on f s \wedge integrable\_on g s \longrightarrow integral s (\lambda x::(real, ?'d::type) cart. vector\_sub (f x) (g x)) = vector\_sub (integral s f) (integral s g)$

thm INTEGRABLE_0:

$\forall s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on (\lambda x::(real, ?'b::type) cart. vec (0::nat)) s$

thm INTEGRABLE_ADD:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) (g::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on f s \wedge integrable\_on g s \longrightarrow integrable\_on (\lambda x::(real, ?'b::type) cart. vector\_add (f x) (g x)) s$

thm INTEGRABLE_CMUL:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) (c::real) s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on f s \longrightarrow integrable\_on (\lambda x::(real, ?'b::type) cart. \% c (f x)) s$

thm INTEGRABLE_NEG:

$\forall (f::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. integrable\_on f s \longrightarrow integrable\_on (\lambda x::(real, ?'b::type) cart. vector\_neg (f x)) s$

thm INTEGRABLE_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge integrable\_on\ g\ s \longrightarrow integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm INTEGRABLE_LINEAR:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge linear\ h \longrightarrow integrable\_on\ (h \circ f)\ s$

thm INTEGRAL_LINEAR:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::(real,\ ?'c::type)\ cart \Rightarrow bool)\ h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ integrable\_on\ f\ s \wedge linear\ h \longrightarrow integral\ s\ (h \circ f) = h\ (integral\ s\ f)$

thm HAS_INTEGRAL_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::?'c::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'c::type.\ IN\ a\ t \longrightarrow has\_integral\ (f\ a)\ ((?i::?'c::type \Rightarrow (real,\ ?'a::type)\ cart)\ a)\ s) \longrightarrow has\_integral\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ t\ (\lambda a::?'c::type.\ f\ a\ x))\ (vsum\ t\ ?i)\ s$

thm INTEGRAL_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::?'c::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'c::type.\ IN\ a\ t \longrightarrow integrable\_on\ (f\ a)\ s) \longrightarrow integral\ s\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ t\ (\lambda a::?'c::type.\ f\ a\ x)) = vsum\ t\ (\lambda a::?'c::type.\ integral\ s\ (f\ a))$

thm INTEGRABLE_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::?'c::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'c::type.\ IN\ a\ t \longrightarrow integrable\_on\ (f\ a)\ s) \longrightarrow integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ t\ (\lambda a::?'c::type.\ f\ a\ x))\ s$

thm HAS_INTEGRAL_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (k::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge has\_integral\ f\ k\ s \longrightarrow has\_integral\ g\ k\ s$

thm INTEGRABLE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge integrable\_on\ f\ s \longrightarrow integrable\_on\ g\ s$

thm HAS_INTEGRAL_EQ_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (k::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.$

$(\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow has\_integral\ f\ k\ s = has\_integral\ g\ k\ s$

thm HAS_INTEGRAL_NULL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow has\_integral\ f\ (vec\ (0::nat))\ (closed\_interval\ [(a,\ b)])$

thm HAS_INTEGRAL_NULL_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ i::(real,\ ?'a::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow has\_integral\ f\ i\ (closed\_interval\ [(a,\ b)]) = (i = vec\ (0::nat))$

thm INTEGRAL_NULL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow integral\ (closed\_interval\ [(a,\ b)])\ f = vec\ (0::nat)$

thm INTEGRABLE_ON_NULL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm HAS_INTEGRAL_EMPTY:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ has\_integral\ f\ (vec\ (0::nat))\ EMPTY$

thm HAS_INTEGRAL_EMPTY_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ i::(real,\ ?'a::type)\ cart.\ has\_integral\ f\ i\ EMPTY = (i = vec\ (0::nat))$

thm INTEGRABLE_ON_EMPTY:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ integrable\_on\ f\ EMPTY$

thm INTEGRAL_EMPTY:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ integral\ EMPTY\ f = vec\ (0::nat)$

thm HAS_INTEGRAL_REFL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ has\_integral\ f\ (vec\ (0::nat))\ (closed\_interval\ [(a,\ a)])$

thm INTEGRABLE_ON_REFL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ a::(real,\ ?'b::type)\ cart.\ integrable\_on\ f\ (closed\_interval\ [(a,\ a)])$

thm INTEGRAL_REFL:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *a*::(*real*, *?'b*::*type*) *cart*. *integral* (*closed_interval* [(*a*, *a*)]) *f* = *vec* (*0*::*nat*)

thm INTEGRABLE_CAUCHY:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) = (∀ *e*>*0*::*real*. ∃ *d*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ *bool*. *gauge d* ∧ (∀ (*p1*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *p2*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p1* (*closed_interval* [(*a*, *b*)]) ∧ *fine d p1* ∧ *tagged_division_of p2* (*closed_interval* [(*a*, *b*)]) ∧ *fine d p2* ⟶ *vector_norm* (*vector_sub* (*vsum p1* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*))))) (*vsum p2* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*)))))) < *e*))

thm INTERVAL_SPLIT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1658*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1658* ($ *x k* ≤ *c*) *x*)) = *closed_interval* [(*a*, *lambda* (λ*i*::*nat*. *if i* = *k then min* ($ *b k*) *c else* $ *b i*))] ∧ *HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1659*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1659* (*c* ≤ $ *x k*) *x*)) = *closed_interval* [(*lambda* (λ*i*::*nat*. *if i* = *k then max* ($ *a k*) *c else* $ *a i*), *b*)]

thm CONTENT_SPLIT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *content* (*closed_interval* [(*a*, *b*)]) = *content* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1660*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1660* ($ *x k* ≤ (*?c*::*real*)) *x*))) + *content* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1661*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1661* (*?c* ≤ $ *x k*) *x*)))

thm DIVISION_SPLIT_RIGHT_INJ:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*i*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*k1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*k2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*k*::*nat*) *c*::*real*. *division_of d i* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *IN k1 d* ∧ *IN k2 d* ∧ *k1* ≠ *k2* ∧ *HOL_Light_Import.INTER k1* (*GSPEC* (λ*GEN%PVAR%1669*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1669* (*c* ≤ $ *x k*) *x*)) = *HOL_Light_Import.INTER k2* (*GSPEC* (λ*GEN%PVAR%1670*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1670* (*c* ≤ $ *x k*) *x*)) ⟶ *content* (*HOL_Light_Import.INTER*

$k1$ $(GSPEC$ $(\lambda GEN\%PVAR\%1671::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1671\ (c \le \$\ x\ k)\ x))) = (0::real)$

thm DIVISION_SPLIT_LEFT_INJ:

$\forall (d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart \Rightarrow bool)$ $(k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (k2::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (k::nat)$ $c::real.\ division\_of\ d\ i \wedge (1::nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV$ $\wedge\ IN\ k1\ d \wedge IN\ k2\ d \wedge k1 \ne k2 \wedge HOL\_Light\_Import.INTER\ k1\ (GSPEC$ $(\lambda GEN\%PVAR\%1666::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1666\ (\$\ x\ k \le c)\ x)) = HOL\_Light\_Import.INTER\ k2$ $(GSPEC\ (\lambda GEN\%PVAR\%1667::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.$ $SETSPEC\ GEN\%PVAR\%1667\ (\$\ x\ k \le c)\ x)) \longrightarrow content\ (HOL\_Light\_Import.INTER$ $k1\ (GSPEC\ (\lambda GEN\%PVAR\%1668::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1668\ (\$\ x\ k \le c)\ x))) = (0::real)$

thm TAGGED_DIVISION_SPLIT_LEFT_INJ:

$\forall (d::(real,\ ?'a::type)\ cart \times ((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (i::(real,$ $?'a::type)\ cart \Rightarrow bool)\ (x1::(real,\ ?'a::type)\ cart)\ (k1::(real,\ ?'a::type)\ cart$ $\Rightarrow bool)\ (x2::(real,\ ?'a::type)\ cart)\ (k2::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (k::nat)$ $c::real.\ tagged\_division\_of\ d\ i \wedge (1::nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV$ $\wedge\ IN\ (x1,\ k1)\ d \wedge IN\ (x2,\ k2)\ d \wedge k1 \ne k2 \wedge HOL\_Light\_Import.INTER$ $k1\ (GSPEC\ (\lambda GEN\%PVAR\%1672::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1672\ (\$\ x\ k \le c)\ x)) = HOL\_Light\_Import.INTER$ $k2\ (GSPEC\ (\lambda GEN\%PVAR\%1673::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1673\ (\$\ x\ k \le c)\ x)) \longrightarrow content\ (HOL\_Light\_Import.INTER$ $k1\ (GSPEC\ (\lambda GEN\%PVAR\%1674::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1674\ (\$\ x\ k \le c)\ x))) = (0::real)$

thm TAGGED_DIVISION_SPLIT_RIGHT_INJ:

$\forall (d::(real,\ ?'a::type)\ cart \times ((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (i::(real,$ $?'a::type)\ cart \Rightarrow bool)\ (x1::(real,\ ?'a::type)\ cart)\ (k1::(real,\ ?'a::type)\ cart$ $\Rightarrow bool)\ (x2::(real,\ ?'a::type)\ cart)\ (k2::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (k::nat)$ $c::real.\ tagged\_division\_of\ d\ i \wedge (1::nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV$ $\wedge\ IN\ (x1,\ k1)\ d \wedge IN\ (x2,\ k2)\ d \wedge k1 \ne k2 \wedge HOL\_Light\_Import.INTER$ $k1\ (GSPEC\ (\lambda GEN\%PVAR\%1675::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1675\ (c \le \$\ x\ k)\ x)) = HOL\_Light\_Import.INTER$ $k2\ (GSPEC\ (\lambda GEN\%PVAR\%1676::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1676\ (c \le \$\ x\ k)\ x)) \longrightarrow content\ (HOL\_Light\_Import.INTER$ $k1\ (GSPEC\ (\lambda GEN\%PVAR\%1677::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)$ $cart.\ SETSPEC\ GEN\%PVAR\%1677\ (c \le \$\ x\ k)\ x))) = (0::real)$

thm DIVISION_SPLIT:

$\forall (p::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,$ $?'a::type)\ cart)\ (k::nat)\ c::real.\ division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge (1::nat)$ $\le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow division\_of\ (GSPEC\ (\lambda GEN\%PVAR\%1685::(real,$ $?'a::type)\ cart \Rightarrow bool.\ \exists l::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1685$

(*IN l p* ∧ *HOL_Light_Import.INTER l* (*GSPEC* (λ*GEN%PVAR%1684*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1684* ($ *x* *k* ≤ *c*) *x*)) ≠ *EMPTY*) (*HOL_Light_Import.INTER l* (*GSPEC* (λ*GEN%PVAR%1683*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1683* ($ *x* *k* ≤ *c*) *x*))))) (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1686*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1686* ($ *x k* ≤ *c*) *x*))) ∧ *division_of* (*GSPEC* (λ*GEN%PVAR%1691*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *l*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%1691* (*IN l p* ∧ *HOL_Light_Import.INTER l* (*GSPEC* (λ*GEN%PVAR%1690*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1690* (*c* ≤ $ *x k*) *x*)) ≠ *EMPTY*) (*HOL_Light_Import.INTER l* (*GSPEC* (λ*GEN%PVAR%1689*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1689* (*c* ≤ $ *x k*) *x*))))) (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1692*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1692* (*c* ≤ $ *x k*) *x*)))

thm HAS_INTEGRAL_SPLIT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*k*::*nat*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *c*::*real*. *has_integral f* (*?i*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1710*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1710* ($ *x k* ≤ *c*) *x*))) ∧ *has_integral f* (*?j*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1711*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1711* (*c* ≤ $ *x k*) *x*))) ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *has_integral f* (*vector_add ?i ?j*) (*closed_interval* [(*a*, *b*)])

thm TAGGED_DIVISION_UNION_INTERVAL:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*p1*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*p2*::(*real*, *?'a*::*type*) *cart* × ((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*c*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *tagged_division_of p1* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1714*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1714* ($ *x k* ≤ *c*) *x*))) ∧ *tagged_division_of p2* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1715*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1715* (*c* ≤ $ *x k*) *x*))) ⟶ *tagged_division_of* (*HOL_Light_Import.UNION p1 p2*) (*closed_interval* [(*a*, *b*)])

thm HAS_INTEGRAL_SEPARATE_SIDES:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*i*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *k*::*nat*. *has_integral f i* (*closed_interval* [(*a*, *b*)]) ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ (∀ *e*>*0*::*real*. ∃ *d*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ *bool*. *gauge d* ∧ (∀ (*p1*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *p2*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p1* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1718*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1718* ($ *x k* ≤ (*?c*::*real*)) *x*))) ∧ *fine d p1* ∧ *tagged_division_of p2* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1719*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1719* (*?c* ≤ $ *x k*) *x*))) ∧ *fine d p2* ⟶ *vector_norm* (*vector_sub* (*vector_add* (*vsum p1* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*))))) (*vsum p2* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*)))))) *i*) < *e*))

thm INTEGRABLE_SPLIT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ (*1*::*nat*) ≤ (*?k*::*nat*) ∧ *?k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *integrable_on f* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1720*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1720* ($ *x ?k* ≤ (*?c*::*real*)) *x*))) ∧ *integrable_on f* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1721*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1721* (*?c* ≤ $ *x ?k*) *x*)))

thm DEF_operative:

operative = (λ(*_1598447*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'b*::*type*) *_1598448*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *?'b*::*type*. (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *_1598448* (*closed_interval* [(*a*, *b*)]) = *neutral _1598447*) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *_1598448* (*closed_interval* [(*a*, *b*)]) = *_1598447* (*_1598448* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1722*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1722* ($ *x k* ≤ *c*) *x*)))) (*_1598448* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1723*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1723* (*c* ≤ $ *x k*) *x*))))))

thm operative:

∀ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'b*::*type*) *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *?'b*::*type*. *operative op f* = ((∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *f* (*closed_interval* [(*a*, *b*)]) = *neutral op*) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *f* (*closed_interval* [(*a*, *b*)]) = *op* (*f* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1722*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1722* ($ *x k* ≤ *c*) *x*)))) (*f* (*HOL_Light_Import.INTER* (*closed_interval* [(*a*, *b*)]) (*GSPEC* (λ*GEN%PVAR%1723*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1723* (*c* ≤ $ *x k*) *x*))))))

thm OPERATIVE_TRIVIAL:

$\forall$ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'b*::*type*) (*f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *?'b*::*type*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *operative op f* ∧ *content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *f* (*closed_interval* [(*a*, *b*)]) = *neutral op*

thm PROPERTY_EMPTY_INTERVAL:

$\forall$ *P*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. ($\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *P* (*closed_interval* [(*a*, *b*)])) ⟶ *P EMPTY*

thm OPERATIVE_EMPTY:

$\forall$ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'b*::*type*) *f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *?'b*::*type*. *operative op f* ⟶ *f EMPTY* = *neutral op*

thm FORALL_OPTION:

($\forall$ *x*::*?'a*::*type HOL_Light_Import.option*. (*?P*::*?'a*::*type HOL_Light_Import.option* ⇒ *bool*) *x*) = (*?P NONE* ∧ ($\forall$ *x*::*?'a*::*type*. *?P* (*SOME x*)))

thm EXISTS_OPTION:

($\exists$ *x*::*?'a*::*type HOL_Light_Import.option*. (*?P*::*?'a*::*type HOL_Light_Import.option* ⇒ *bool*) *x*) = (*?P NONE* ∨ ($\exists$ *x*::*?'a*::*type*. *?P* (*SOME x*)))

thm DEF_lifted:

lifted = (*SOME lifted*::*nat* ⇒ (*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) ⇒ *?'b*::*type HOL_Light_Import.option* ⇒ *?'b*::*type HOL_Light_Import.option* ⇒ *?'a*::*type HOL_Light_Import.option*. $\forall$ *_1599769*::*nat*. ($\forall$ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) *a_*::*?'b*::*type HOL_Light_Import.option*. *lifted _1599769 op NONE a_* = *NONE*) ∧ ($\forall$ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) *a_*::*?'b*::*type HOL_Light_Import.option*. *lifted _1599769 op a_ NONE* = *NONE*) ∧ ($\forall$ (*op*::*?'b*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) (*x*::*?'b*::*type*) *y*::*?'b*::*type*. *lifted _1599769 op* (*SOME x*) (*SOME y*) = *SOME* (*op x y*))) (*53*::*nat*)

thm lifted:

lifted (*?op*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *?'b*::*type*) *NONE* (*?a_*::*?'a*::*type HOL_Light_Import.option*) = *NONE* ∧ *lifted ?op ?a_ NONE* = *NONE* ∧ *lifted ?op* (*SOME* (*?x*::*?'a*::*type*)) (*SOME* (*?y*::*?'a*::*type*)) = *SOME* (*?op ?x ?y*)

thm lifted_conjunct2:

lifted (*?op*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *?'b*::*type*) (*SOME* (*?x*::*?'a*::*type*)) (*SOME* (*?y*::*?'a*::*type*)) = *SOME* (*?op ?x ?y*)

thm lifted_conjunct1:

lifted (*?op*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *?'b*::*type*) (*?a_*::*?'a*::*type HOL_Light_Import.option*) *NONE* = *NONE*

thm lifted_conjunct0:

$lifted\ (?op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'b::type)\ NONE\ (?a\_::?'a::type\ HOL\_Light\_Import.option) = NONE$

thm NEUTRAL_LIFTED:

$\forall\, op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow neutral\ (lifted\ op) = SOME\ (neutral\ op)$

thm MONOIDAL_LIFTED:

$\forall\, op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow monoidal\ (lifted\ op)$

thm ITERATE_SOME:

$\forall\, op::?'b::type \Rightarrow ?'b::type \Rightarrow ?'b::type.\ monoidal\ op \longrightarrow (\forall\,(f::?'a::type \Rightarrow ?'b::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow iterate\ (lifted\ op)\ s\ (\lambda x::?'a::type.\ SOME\ (f\ x)) = SOME\ (iterate\ op\ s\ f))$

thm OPERATIVE_CONTENT:

$operative\ op + content$

thm OPERATIVE_INTEGRAL:

$\forall\, f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ operative\ (lifted\ vector\_add)\ (\lambda i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ if\ integrable\_on\ f\ i\ then\ SOME\ (integral\ i\ f)\ else\ NONE)$

thm DEF_division_points:

$division\_points = (\lambda(\_1600310::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1600311::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1724::nat \times real.\ \exists\,(j::nat)\ x::real.\ SETSPEC\ GEN\%PVAR\%1724\ ((1::nat) \le j \wedge j \le dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ (interval\_lowerbound\ \_1600310)\ j < x \wedge x < \$\ (interval\_upperbound\ \_1600310)\ j \wedge (\exists\, i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ i\ \_1600311 \wedge (\$\ (interval\_lowerbound\ i)\ j = x \vee \$\ (interval\_upperbound\ i)\ j = x)))\ (j,\ x)))$

thm division_points:

$\forall\,(k::(real,\ ?'a::type)\ cart \Rightarrow bool)\ d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_points\ k\ d = GSPEC\ (\lambda GEN\%PVAR\%1724::nat \times real.\ \exists\,(j::nat)\ x::real.\ SETSPEC\ GEN\%PVAR\%1724\ ((1::nat) \le j \wedge j \le dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ (interval\_lowerbound\ k)\ j < x \wedge x < \$\ (interval\_upperbound\ k)\ j \wedge (\exists\, i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ i\ d \wedge (\$\ (interval\_lowerbound\ i)\ j = x \vee \$\ (interval\_upperbound\ i)\ j = x)))\ (j,\ x))$

thm DIVISION_POINTS_FINITE:

$\forall\,(d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ division\_of\ d\ i \longrightarrow FINITE\ (division\_points\ i\ d)$

thm DIVISION_POINTS_SUBSET:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::real)\ (d::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ k::nat.\ division\_of\ d\ (closed\_interval\ [(a, b)]) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i < \$\ b\ i) \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ a\ k < c \wedge c < \$\ b\ k \longrightarrow SUBSET\ (division\_points\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a, b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1725::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1725\ (\$\ x\ k \leq c)\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%1730::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists l::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1730\ (IN\ l\ d \wedge HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1729::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1729\ (\$\ x\ k \leq c)\ x)) \neq EMPTY)\ (HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1728::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1728\ (\$\ x\ k \leq c)\ x))))))\ (division\_points\ (closed\_interval\ [(a, b)])\ d) \wedge SUBSET\ (division\_points\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a, b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1731::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1731\ (c \leq \$\ x\ k)\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%1736::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists l::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1736\ (IN\ l\ d \wedge HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1735::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1735\ (c \leq \$\ x\ k)\ x)) \neq EMPTY)\ (HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1734::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1734\ (c \leq \$\ x\ k)\ x))))))\ (division\_points\ (closed\_interval\ [(a, b)])\ d)$

thm DIVISION_POINTS_PSUBSET:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (c::real)\ (d::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ k::nat.\ division\_of\ d\ (closed\_interval\ [(a, b)]) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ a\ i < \$\ b\ i) \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ a\ k < c \wedge c < \$\ b\ k \wedge (\exists l::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ l\ d \wedge (\$\ (interval\_lowerbound\ l)\ k = c \vee \$\ (interval\_upperbound\ l)\ k = c)) \longrightarrow PSUBSET\ (division\_points\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a, b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1737::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1737\ (\$\ x\ k \leq c)\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%1742::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists l::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1742\ (IN\ l\ d \wedge HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1741::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1741\ (\$\ x\ k \leq c)\ x)) \neq EMPTY)\ (HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1740::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1740\ (\$\ x\ k \leq c)\ x))))))\ (division\_points\ (closed\_interval\ [(a, b)])\ d) \wedge PSUBSET\ (division\_points\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a, b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1743::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1743\ (c \leq \$\ x\ k)\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%1748::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists l::(real, ?'a::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1748\ (IN\ l\ d \wedge HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1747::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1747\ (c \leq \$\ x\ k)\ x)) \neq EMPTY)\ (HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1746::(real,$

?'a::type) cart. ∃ *x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1746* (*c* ≤ $ *x k*) *x*)))))) (*division_points* (*closed_interval* [(*a*, *b*)]) *d*)

thm OPERATIVE_DIVISION:

∀ (*op::?'b::type* ⇒ *?'b::type* ⇒ *?'b::type*) (*d*::((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a::type*) *cart*) (*b*::(*real*, *?'a::type*) *cart*) *f*::((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *?'b::type*. *monoidal op* ∧ *operative op f* ∧ *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *iterate op d f* = *f* (*closed_interval* [(*a*, *b*)])

thm OPERATIVE_TAGGED_DIVISION:

∀ (*op::?'b::type* ⇒ *?'b::type* ⇒ *?'b::type*) (*d*::(*real*, *?'a::type*) *cart* × ((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a::type*) *cart*) (*b*::(*real*, *?'a::type*) *cart*) *f*::((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *?'b::type*. *monoidal op* ∧ *operative op f* ∧ *tagged_division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *iterate op d* (*GABS* (λ*fa*::(*real*, *?'a::type*) *cart* × ((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *?'b::type*. ∀ (*x*::(*real*, *?'a::type*) *cart*) *l*::(*real*, *?'a::type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *l*)) (*f l*))) = *f* (*closed_interval* [(*a*, *b*)])

thm ADDITIVE_CONTENT_DIVISION:

∀ (*d*::((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a::type*) *cart*) *b*::(*real*, *?'a::type*) *cart*. *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d content* = *content* (*closed_interval* [(*a*, *b*)])

thm ADDITIVE_CONTENT_TAGGED_DIVISION:

∀ (*d*::(*real*, *?'a::type*) *cart* × ((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a::type*) *cart*) *b*::(*real*, *?'a::type*) *cart*. *tagged_division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d* (*GABS* (λ*f*::(*real*, *?'a::type*) *cart* × ((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *real*. ∀ (*x*::(*real*, *?'a::type*) *cart*) *l*::(*real*, *?'a::type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *l*)) (*content l*))) = *content* (*closed_interval* [(*a*, *b*)])

thm SUBADDITIVE_CONTENT_DIVISION:

∀ (*d*::((*real*, *?'a::type*) *cart* ⇒ *bool*) ⇒ *bool*) (*s*::(*real*, *?'a::type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a::type*) *cart*) *b*::(*real*, *?'a::type*) *cart*. *division_of d s* ∧ *SUBSET s* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d content* ≤ *content* (*closed_interval* [(*a*, *b*)])

thm HAS_INTEGRAL_CONST:

∀ (*a*::(*real*, *?'b::type*) *cart*) (*b*::(*real*, *?'b::type*) *cart*) *c*::(*real*, *?'a::type*) *cart*. *has_integral* (λ*x*::(*real*, *?'b::type*) *cart*. *c*) (% (*content* (*closed_interval* [(*a*, *b*)])) *c*) (*closed_interval* [(*a*, *b*)])

thm INTEGRABLE_CONST:

∀ (*a*::(*real*, *?'b::type*) *cart*) (*b*::(*real*, *?'b::type*) *cart*) *c*::(*real*, *?'a::type*) *cart*. *integrable_on* (λ*x*::(*real*, *?'b::type*) *cart*. *c*) (*closed_interval* [(*a*, *b*)])

thm INTEGRAL_CONST:

$\forall (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'b::type)\ cart.\ c) = \%\ (content\ (closed\_interval\ [(a,\ b)]))\ c$

thm INTEGRAL_PASTECART_CONST:

$\forall (a::(real,\ ?'c::type)\ cart)\ (b::(real,\ ?'c::type)\ cart)\ (c::(real,\ ?'b::type)\ cart)\ (d::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'a::type)\ cart.\ integral\ (closed\_interval\ [(pastecart\ a\ c,\ pastecart\ b\ d)])\ (\lambda x::(real,\ (?'c::type,\ ?'b::type)\ finite\_sum)\ cart.\ k) = integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'c::type)\ cart.\ integral\ (closed\_interval\ [(c,\ d)])\ (\lambda y::(real,\ ?'b::type)\ cart.\ k))$

thm DSUM_BOUND:

$\forall (p::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ e::real.\ division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge vector\_norm\ c \leq e \longrightarrow vector\_norm\ (vsum\ p\ (\lambda l::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \%\ (content\ l)\ c)) \leq e * content\ (closed\_interval\ [(a,\ b)])$

thm RSUM_BOUND:

$\forall (p::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ e::real.\ tagged\_division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (f\ x) \leq e) \longrightarrow vector\_norm\ (vsum\ p\ (GABS\ (\lambda fa::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (fa\ (x,\ k))\ (\%\ (content\ k)\ (f\ x))))) \leq e * content\ (closed\_interval\ [(a,\ b)])$

thm RSUM_DIFF_BOUND:

$\forall (p::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ tagged\_division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x)) \leq (?e::real)) \longrightarrow vector\_norm\ (vector\_sub\ (vsum\ p\ (GABS\ (\lambda fa::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (fa\ (x,\ k))\ (\%\ (content\ k)\ (f\ x)))))\ (vsum\ p\ (GABS\ (\lambda f::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (f\ (x,\ k))\ (\%\ (content\ k)\ (g\ x)))))) \leq ?e * content\ (closed\_interval\ [(a,\ b)])$

thm HAS_INTEGRAL_BOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (i::(real,\ ?'a::type)\ cart)\ B::real.\ (0::real) \leq B \wedge has\_integral\ f\ i\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (f\ x) \leq B) \longrightarrow vector\_norm\ i \leq B * content\ (closed\_interval\ [(a,\ b)])$

thm RSUM_COMPONENT_LE:

$\forall (p::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ tagged\_division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge (1::nat) \leq (?i::nat) \wedge ?i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow \$\ (f\ x)\ ?i \leq \$\ (g\ x)\ ?i) \longrightarrow \$\ (vsum\ p\ (GABS\ (\lambda fa::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (fa\ (x,\ k))\ (\%\ (content\ k)\ (f\ x)))))\ ?i \leq \$\ (vsum\ p\ (GABS\ (\lambda f::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ \forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ GEQ\ (f\ (x,\ k))\ (\%\ (content\ k)\ (g\ x)))))\ ?i$

thm HAS_INTEGRAL_COMPONENT_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart)\ (j::(real,\ ?'a::type)\ cart)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge has\_integral\ f\ i\ s \wedge has\_integral\ g\ j\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow \$\ (f\ x)\ k \leq \$\ (g\ x)\ k) \longrightarrow \$\ i\ k \leq \$\ j\ k$

thm INTEGRAL_COMPONENT_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge integrable\_on\ f\ s \wedge integrable\_on\ g\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow \$\ (f\ x)\ k \leq \$\ (g\ x)\ k) \longrightarrow \$\ (integral\ s\ f)\ k \leq \$\ (integral\ s\ g)\ k$

thm HAS_INTEGRAL_DROP_LE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (i::(real,\ unit)\ cart)\ j::(real,\ unit)\ cart.\ has\_integral\ f\ i\ s \wedge has\_integral\ g\ j\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \longrightarrow HOL\_Light\_Import.drop\ i \leq HOL\_Light\_Import.drop\ j$

thm INTEGRAL_DROP_LE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge integrable\_on\ g\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \longrightarrow HOL\_Light\_Import.drop\ (integral\ s\ f) \leq HOL\_Light\_Import.drop\ (integral\ s\ g)$

thm HAS_INTEGRAL_COMPONENT_POS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge has\_integral\ f\ i\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq \$\ (f\ x)\ k) \longrightarrow (0::real) \leq \$\ i\ k$

thm INTEGRAL_COMPONENT_POS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge integrable\_on\ f\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq \$\ (f\ x)\ k) \longrightarrow (0::real) \leq \$\ (integral\ s\ f)\ k$

thm HAS_INTEGRAL_DROP_POS:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ i::(real,\ unit)\ cart.\ has\_integral\ f\ i\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ i$

thm INTEGRAL_DROP_POS:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (integral\ s\ f)$

thm HAS_INTEGRAL_COMPONENT_NEG:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart)\ k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge has\_integral\ f\ i\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow \$\ (f\ x)\ k \leq (0::real)) \longrightarrow \$\ i\ k \leq (0::real)$

thm HAS_INTEGRAL_DROP_NEG:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ i::(real,\ unit)\ cart.\ has\_integral\ f\ i\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq (0::real)) \longrightarrow HOL\_Light\_Import.drop\ i \leq (0::real)$

thm HAS_INTEGRAL_COMPONENT_LBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (i::(real,\ ?'a::type)\ cart)\ k::nat.\ has\_integral\ f\ i\ (closed\_interval\ [(a,\ b)]) \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow (?B::real) \leq \$\ (f\ x)\ k) \longrightarrow ?B * content\ (closed\_interval\ [(a,\ b)]) \leq \$\ i\ k$

thm HAS_INTEGRAL_COMPONENT_UBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (i::(real,\ ?'a::type)\ cart)\ k::nat.\ has\_integral\ f\ i\ (closed\_interval\ [(a,\ b)]) \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow \$\ (f\ x)\ k \leq (?B::real)) \longrightarrow \$\ i\ k \leq ?B * content\ (closed\_interval\ [(a,\ b)])$

thm INTEGRAL_COMPONENT_LBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ k::nat.\ integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge$

$(1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow (?B::real) \leq \$\ (f\ x)\ k) \longrightarrow ?B * content\ (closed\_interval\ [(a,\ b)]) \leq \$\ (integral\ (closed\_interval\ [(a,\ b)])\ f)\ k$

thm INTEGRAL_COMPONENT_UBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ k::nat.\ integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow \$\ (f\ x)\ k \leq (?B::real)) \longrightarrow \$\ (integral\ (closed\_interval\ [(a,\ b)])\ f)\ k \leq ?B * content\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_UNIFORM_LIMIT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ (\forall e>0::real.\ \exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x)) \leq e) \wedge integrable\_on\ g\ (closed\_interval\ [(a,\ b)])) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm DEF_indicator:

$indicator = (\lambda(\_1618832::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ if\ IN\ x\ \_1618832\ then\ vec\ (1::nat)\ else\ vec\ (0::nat))$

thm indicator:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ indicator\ s = (\lambda x::(real,\ ?'a::type)\ cart.\ if\ IN\ x\ s\ then\ vec\ (1::nat)\ else\ vec\ (0::nat))$

thm DROP_INDICATOR:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.drop\ (indicator\ s\ x) = (if\ IN\ x\ s\ then\ 1::real\ else\ (0::real))$

thm DROP_INDICATOR_POS_LE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ (0::real) \leq HOL\_Light\_Import.drop\ (indicator\ s\ x)$

thm DROP_INDICATOR_LE_1:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.drop\ (indicator\ s\ x) \leq (1::real)$

thm DROP_INDICATOR_ABS_LE_1:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ |HOL\_Light\_Import.drop\ (indicator\ s\ x)| \leq (1::real)$

thm DEF_negligible:

$negligible = (\lambda\_1618847::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ has\_integral\ (indicator\ \_1618847)\ (vec\ (0::nat))\ (closed\_interval\ [(a,\ b)]))$

thm negligible:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ negligible\ s = (\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ b{::}(real,\ ?'a{::}type)\ cart.\ has\_integral\ (indicator\ s)\ (vec\ (0{::}nat))\ (closed\_interval\ [(a,\ b)]))$

thm VSUM_NONZERO_IMAGE_LEMMA:

$\forall\, (s{::}?'c{::}type \Rightarrow bool)\ (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ (g{::}?'b{::}type \Rightarrow (real,\ ?'a{::}type)\ cart)\ a{::}?'b{::}type.\ FINITE\ s \wedge g\ a = vec\ (0{::}nat) \wedge (\forall\, (x{::}?'c{::}type)\ y{::}?'c{::}type.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \wedge x \neq y \longrightarrow g\ (f\ x) = vec\ (0{::}nat)) \longrightarrow vsum\ (GSPEC\ (\lambda GEN\%PVAR\%1762{::}?'b{::}type.\ \exists\, x{::}?'c{::}type.\ SETSPEC\ GEN\%PVAR\%1762\ (IN\ x\ s \wedge f\ x \neq a)\ (f\ x)))\ g = vsum\ s\ (g \circ f)$

thm INTERVAL_DOUBLESPLIT:

$(1{::}nat) \leq (?k{::}nat) \wedge ?k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow HOL\_Light\_Import.INTER\ (closed\_interval\ [(?a{::}(real,\ ?'a{::}type)\ cart,\ ?b{::}(real,\ ?'a{::}type)\ cart)])\ (GSPEC\ (\lambda GEN\%PVAR\%1766{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1766\ (|\$\ x\ ?k - (?c{::}real)| \leq (?e{::}real))\ x)) = closed\_interval\ [(lambda\ (\lambda i{::}nat.\ if\ i = ?k\ then\ max\ (\$\ ?a\ ?k)\ (?c - ?e)\ else\ \$\ ?a\ i),\ lambda\ (\lambda i{::}nat.\ if\ i = ?k\ then\ min\ (\$\ ?b\ ?k)\ (?c + ?e)\ else\ \$\ ?b\ i))]$

thm DIVISION_DOUBLESPLIT:

$\forall\, (p{::}((real,\ ?'a{::}type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a{::}(real,\ ?'a{::}type)\ cart)\ (b{::}(real,\ ?'a{::}type)\ cart)\ (k{::}nat)\ (c{::}real)\ e{::}real.\ division\_of\ p\ (closed\_interval\ [(a,\ b)]) \wedge (1{::}nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow division\_of\ (GSPEC\ (\lambda GEN\%PVAR\%1771{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ \exists\, l{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1771\ (IN\ l\ p \wedge HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1770{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1770\ (|\$\ x\ k - c| \leq e)\ x)) \neq EMPTY)\ (HOL\_Light\_Import.INTER\ l\ (GSPEC\ (\lambda GEN\%PVAR\%1769{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1769\ (|\$\ x\ k - c| \leq e)\ x)))))\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1772{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1772\ (|\$\ x\ k - c| \leq e)\ x)))$

thm CONTENT_DOUBLESPLIT:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ (b{::}(real,\ ?'a{::}type)\ cart)\ (k{::}nat)\ (c{::}real)\ e{::}real.\ (0{::}real) < e \wedge (1{::}nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\exists\, d{>}0{::}real.\ content\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1773{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1773\ (|\$\ x\ k - c| \leq d)\ x))) < e)$

thm NEGLIGIBLE_STANDARD_HYPERPLANE:

$\forall\, (c{::}real)\ k{::}nat.\ (1{::}nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow negligible\ (GSPEC\ (\lambda GEN\%PVAR\%1780{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%1780\ (\$\ x\ k = c)\ x))$

thm TAGGED_DIVISION_FINER:

$\forall (p::(real,\ ?'a::type)\ cart \times ((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool.\ tagged\_division\_of\ p\ (closed\_interval\ [(a,\ b)]) \land gauge\ d \longrightarrow (\exists q::(real,\ ?'a::type)\ cart \times ((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ tagged\_division\_of\ q\ (closed\_interval\ [(a,\ b)]) \land fine\ d\ q \land (\forall (x::(real,\ ?'a::type)\ cart)\ k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ (x,\ k)\ p \land SUBSET\ k\ (d\ x) \longrightarrow IN\ (x,\ k)\ q))$

thm HAS_INTEGRAL_NEGLIGIBLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ negligible\ s \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow f\ x = vec\ (0::nat)) \longrightarrow has\_integral\ f\ (vec\ (0::nat))\ t$

thm HAS_INTEGRAL_SPIKE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ negligible\ s \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \land has\_integral\ f\ (?y::(real,\ ?'a::type)\ cart)\ t \longrightarrow has\_integral\ g\ ?y\ t$

thm HAS_INTEGRAL_SPIKE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ y::(real,\ ?'a::type)\ cart.\ negligible\ s \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow has\_integral\ f\ y\ t = has\_integral\ g\ y\ t$

thm INTEGRABLE_SPIKE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ negligible\ s \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow integrable\_on\ f\ t \longrightarrow integrable\_on\ g\ t$

thm INTEGRAL_SPIKE:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (s::(real,\ ?'c::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'c::type)\ cart \Rightarrow bool)\ y::?'a::type.\ negligible\ s \land (\forall x::(real,\ ?'c::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow integral\ t\ f = integral\ t\ g$

thm NEGLIGIBLE_SUBSET:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ s \land SUBSET\ t\ s \longrightarrow negligible\ t$

thm NEGLIGIBLE_DIFF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ s \longrightarrow negligible\ (DIFF\ s\ t)$

thm NEGLIGIBLE_INTER:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ s \lor negligible\ t \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ s\ t)$

thm NEGLIGIBLE_UNION:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* ∧ *negligible t* ⟶ *negligible* (*HOL_Light_Import.UNION s t*)

thm NEGLIGIBLE_UNION_EQ:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible* (*HOL_Light_Import.UNION s t*) = (*negligible s* ∧ *negligible t*)

thm NEGLIGIBLE_SING:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. *negligible* (*INSERT a EMPTY*)

thm NEGLIGIBLE_INSERT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible* (*INSERT a s*) = *negligible s*

thm NEGLIGIBLE_EMPTY:

negligible EMPTY

thm NEGLIGIBLE_FINITE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *negligible s*

thm NEGLIGIBLE_UNIONS:

$\forall$ *s*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE s* ∧ ($\forall$ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN t s* ⟶ *negligible t*) ⟶ *negligible* (*UNIONS s*)

thm NEGLIGIBLE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* = ($\forall$ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *has_integral* (*indicator s*) (*vec* (*0*::*nat*)) *t*)

thm HAS_INTEGRAL_SPIKE_FINITE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *y*::(*real*, *?'a*::*type*) *cart*. *FINITE s* ∧ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x* (*DIFF t s*) ⟶ *g x* = *f x*) ∧ *has_integral f y t* ⟶ *has_integral g y t*

thm HAS_INTEGRAL_SPIKE_FINITE_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *y*::(*real*, *?'a*::*type*) *cart*. *FINITE s* ∧ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x* (*DIFF* (*?t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *s*) ⟶ *g x* = *f x*) ⟶ *has_integral f y ?t* = *has_integral g y ?t*

thm INTEGRABLE_SPIKE_FINITE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x* (*DIFF* (*?t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *s*) ⟶ *g x* = *f x*) ⟶ *integrable_on f ?t* ⟶ *integrable_on g ?t*

thm INTEGRAL_EQ:

$\forall (f::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (g::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; s::(real, ?'b::type) \; cart \Rightarrow bool. \; (\forall x::(real, ?'b::type) \; cart. \; IN \; x \; s \longrightarrow f \; x = g \; x) \longrightarrow integral \; s \; f = integral \; s \; g$

thm INTEGRAL_EQ_0:

$\forall (f::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; s::(real, ?'b::type) \; cart \Rightarrow bool. \; (\forall x::(real, ?'b::type) \; cart. \; IN \; x \; s \longrightarrow f \; x = vec \; (0::nat)) \longrightarrow integral \; s \; f = vec \; (0::nat)$

thm NEGLIGIBLE_FRONTIER_INTERVAL:

$\forall (a::(real, ?'a::type) \; cart) \; b::(real, ?'a::type) \; cart. \; negligible \; (DIFF \; (closed\_interval \; [(a, b)]) \; (open\_interval \; (a, b)))$

thm HAS_INTEGRAL_SPIKE_INTERIOR:

$\forall (f::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (g::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (a::(real, ?'b::type) \; cart) \; (b::(real, ?'b::type) \; cart) \; y::(real, ?'a::type) \; cart. \; (\forall x::(real, ?'b::type) \; cart. \; IN \; x \; (open\_interval \; (a, b)) \longrightarrow g \; x = f \; x) \land has\_integral \; f \; y \; (closed\_interval \; [(a, b)]) \longrightarrow has\_integral \; g \; y \; (closed\_interval \; [(a, b)])$

thm HAS_INTEGRAL_SPIKE_INTERIOR_EQ:

$\forall (f::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (g::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (a::(real, ?'b::type) \; cart) \; (b::(real, ?'b::type) \; cart) \; y::(real, ?'a::type) \; cart. \; (\forall x::(real, ?'b::type) \; cart. \; IN \; x \; (open\_interval \; (a, b)) \longrightarrow g \; x = f \; x) \longrightarrow has\_integral \; f \; y \; (closed\_interval \; [(a, b)]) = has\_integral \; g \; y \; (closed\_interval \; [(a, b)])$

thm INTEGRABLE_SPIKE_INTERIOR:

$\forall (f::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (g::(real, ?'b::type) \; cart \Rightarrow (real, ?'a::type) \; cart) \; (a::(real, ?'b::type) \; cart) \; b::(real, ?'b::type) \; cart. \; (\forall x::(real, ?'b::type) \; cart. \; IN \; x \; (open\_interval \; (a, b)) \longrightarrow g \; x = f \; x) \longrightarrow integrable\_on \; f \; (closed\_interval \; [(a, b)]) \longrightarrow integrable\_on \; g \; (closed\_interval \; [(a, b)])$

thm NEUTRAL_AND:

$neutral \; op \land = True$

thm MONOIDAL_AND:

$monoidal \; op \land$

thm ITERATE_AND:

$\forall (p::?'a::type \Rightarrow bool) \; s::?'a::type \Rightarrow bool. \; FINITE \; s \longrightarrow iterate \; op \land \; s \; p = (\forall x::?'a::type. \; IN \; x \; s \longrightarrow p \; x)$

thm OPERATIVE_DIVISION_AND:

$\forall (P::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ operative\ op \wedge P \wedge division\_of\ d\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\forall i::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ i\ d \longrightarrow P\ i) = P\ (closed\_interval\ [(a,\ b)])$

thm OPERATIVE_APPROXIMABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ e::real.\ (0::real) \leq e \longrightarrow operative\ op \wedge (\lambda i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ i \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x)) \leq e) \wedge integrable\_on\ g\ i)$

thm APPROXIMABLE_ON_DIVISION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ (0::real) \leq (?e::real) \wedge division\_of\ d\ (closed\_interval\ [(a,\ b)]) \wedge (\forall i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ i\ d \longrightarrow (\exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ i \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x)) \leq ?e) \wedge integrable\_on\ g\ i)) \longrightarrow (\exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x)) \leq ?e) \wedge integrable\_on\ g\ (closed\_interval\ [(a,\ b)]))$

thm INTEGRABLE_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm OPERATIVE_1_LT:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow (\forall f::((real,\ unit)\ cart \Rightarrow bool) \Rightarrow ?'a::type.\ operative\ op\ f = ((\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ b \leq HOL\_Light\_Import.drop\ a \longrightarrow f\ (closed\_interval\ [(a,\ b)]) = neutral\ op) \wedge (\forall (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ c \wedge HOL\_Light\_Import.drop\ c < HOL\_Light\_Import.drop\ b \longrightarrow op\ (f\ (closed\_interval\ [(a,\ c)]))\ (f\ (closed\_interval\ [(c,\ b)])) = f\ (closed\_interval\ [(a,\ b)]))))$

thm OPERATIVE_1_LE:

$\forall op::?'a::type \Rightarrow ?'a::type \Rightarrow ?'a::type.\ monoidal\ op \longrightarrow (\forall f::((real,\ unit)\ cart \Rightarrow bool) \Rightarrow ?'a::type.\ operative\ op\ f = ((\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ b \leq HOL\_Light\_Import.drop\ a \longrightarrow f\ (closed\_interval\ [(a,\ b)]) = neutral\ op) \wedge (\forall (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop\ c \wedge HOL\_Light\_Import.drop\ c \leq HOL\_Light\_Import.drop\ b \longrightarrow op\ (f\ (closed\_interval\ [(a,\ c)]))\ (f\ (closed\_interval\ [(c,\ b)])) = f\ (closed\_interval\ [(a,\ b)]))))$

thm ADDITIVE_TAGGED_DIVISION_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (p::(real,\ unit)\ cart \times ((real,\ unit)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop$

a ≤ HOL_Light_Import.drop b ∧ tagged_division_of p (closed_interval [(a, b)]) ⟶ vsum p (GABS (λfa::(real, unit) cart × ((real, unit) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, unit) cart) k::(real, unit) cart ⇒ bool. GEQ (fa (x, k)) (vector_sub (f (interval_upperbound k)) (f (interval_lowerbound k))))) = vector_sub (f b) (f a)

thm HAS_INTEGRAL_FACTOR_CONTENT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (i::(real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. has_integral f i (closed_interval [(a, b)]) = (∀ e>0::real. ∃ d::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart ⇒ bool. gauge d ∧ (∀ p::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ bool. tagged_division_of p (closed_interval [(a, b)]) ∧ fine d p ⟶ vector_norm (vector_sub (vsum p (GABS (λfa::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (fa (x, k)) (% (content k) (f x))))) i) ≤ e ∗ content (closed_interval [(a, b)])))

thm GAUGE_MODIFY:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ s::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.open s ⟶ HOL_Light_Import.open (GSPEC (λGEN%PVAR%1794::(real, ?'b::type) cart. ∃ x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1794 (IN (f x) s) x))) ⟶ (∀ d::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ bool. gauge d ⟶ gauge (λ(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. d (f x) (f y)))

thm OPERATIVE_INTEGRABLE:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. operative op ∧ (integrable_on f)

thm INTEGRABLE_SUBINTERVAL:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (c::(real, ?'b::type) cart) d::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)]) ∧ SUBSET (closed_interval [(c, d)]) (closed_interval [(a, b)]) ⟶ integrable_on f (closed_interval [(c, d)])

thm HAS_INTEGRAL_COMBINE:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (i::(real, ?'a::type) cart) (j::(real, ?'a::type) cart) (a::(real, unit) cart) (b::(real, unit) cart) c::(real, unit) cart. HOL_Light_Import.drop a ≤ HOL_Light_Import.drop c ∧ HOL_Light_Import.drop c ≤ HOL_Light_Import.drop b ∧ has_integral f i (closed_interval [(a, c)]) ∧ has_integral f j (closed_interval [(c, b)]) ⟶ has_integral f (vector_add i j) (closed_interval [(a, b)])

thm INTEGRAL_COMBINE:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) (b::(real, unit) cart) c::(real, unit) cart. HOL_Light_Import.drop a ≤ HOL_Light_Import.drop

c ∧ *HOL_Light_Import.drop c* ≤ *HOL_Light_Import.drop b* ∧ *integrable_on f* (*closed_interval* [(*a*, *b*)]) ⟶ *vector_add* (*integral* (*closed_interval* [(*a*, *c*)]) *f*) (*integral* (*closed_interval* [(*c*, *b*)]) *f*) = *integral* (*closed_interval* [(*a*, *b*)]) *f*

thm INTEGRABLE_COMBINE:

∀(*f*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) *c*::(*real*, *unit*) *cart*. *HOL_Light_Import.drop a* ≤ *HOL_Light_Import.drop c* ∧ *HOL_Light_Import.drop c* ≤ *HOL_Light_Import.drop b* ∧ *integrable_on f* (*closed_interval* [(*a*, *c*)]) ∧ *integrable_on f* (*closed_interval* [(*c*, *b*)]) ⟶ *integrable_on f* (*closed_interval* [(*a*, *b*)])

thm INTEGRABLE_ON_LITTLE_SUBINTERVALS:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. (∀*x*::(*real*, *?'b*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ (∃*d*>*0*::*real*. ∀(*u*::(*real*, *?'b*::*type*) *cart*) *v*::(*real*, *?'b*::*type*) *cart*. *IN x* (*closed_interval* [(*u*, *v*)]) ∧ *SUBSET* (*closed_interval* [(*u*, *v*)]) (*ball* (*x*, *d*)) ∧ *SUBSET* (*closed_interval* [(*u*, *v*)]) (*closed_interval* [(*a*, *b*)]) ⟶ *integrable_on f* (*closed_interval* [(*u*, *v*)]))) ⟶ *integrable_on f* (*closed_interval* [(*a*, *b*)])

thm INTEGRAL_HAS_VECTOR_DERIVATIVE:

∀(*f*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *continuous_on f* (*closed_interval* [(*a*, *b*)]) ⟶ (∀*x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ *has_vector_derivative* (λ*u*::(*real*, *unit*) *cart*. *integral* (*closed_interval* [(*a*, *u*)]) *f*) (*f x*) (*within* (*at x*) (*closed_interval* [(*a*, *b*)])))

thm ANTIDERIVATIVE_CONTINUOUS:

∀(*f*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *continuous_on f* (*closed_interval* [(*a*, *b*)]) ⟶ (∃*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∀*x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ *has_vector_derivative g* (*f x*) (*within* (*at x*) (*closed_interval* [(*a*, *b*)])))

thm HAS_INTEGRAL_TWIDDLE:

∀(*f*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'c*::*type*) *cart*) (*h*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*r*::*real*) (*i*::(*real*, *?'b*::*type*) *cart*) (*a*::(*real*, *?'c*::*type*) *cart*) *b*::(*real*, *?'c*::*type*) *cart*. (*0*::*real*) < *r* ∧ (∀*x*::(*real*, *?'a*::*type*) *cart*. *h* (*g x*) = *x*) ∧ (∀*x*::(*real*, *?'c*::*type*) *cart*. *g* (*h x*) = *x*) ∧ (∀*x*::(*real*, *?'a*::*type*) *cart*. *continuous g* (*at x*)) ∧ (∀(*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. ∃(*w*::(*real*, *?'c*::*type*) *cart*) *z*::(*real*, *?'c*::*type*) *cart*. *IMAGE g* (*closed_interval* [(*u*, *v*)]) = *closed_interval* [(*w*, *z*)]) ∧ (∀(*u*::(*real*, *?'c*::*type*) *cart*) *v*::(*real*, *?'c*::*type*) *cart*. ∃(*w*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *IMAGE h* (*closed_interval* [(*u*, *v*)]) = *closed_interval* [(*w*, *z*)]) ∧ (∀(*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *content* (*IMAGE g* (*closed_interval* [(*u*, *v*)])) = *r* ∗ *content* (*closed_interval* [(*u*, *v*)])) ∧ *has_integral f i* (*closed_interval* [(*a*, *b*)]) ⟶ *has_integral* (λ*x*::(*real*,

?'a::type) cart. f (g x)) (% (inverse_class.inverse r) i) (IMAGE h (closed_interval [(a, b)]))

thm INTERVAL_IMAGE_AFFINITY_INTERVAL:

$\forall$ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (m::real) c::(real, ?'a::type) cart.* $\exists$ *(u::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IMAGE (*λ*x::(real, ?'a::type) cart. vector_add (% m x) c) (closed_interval [(a, b)]) = closed_interval [(u, v)]*

thm CONTENT_IMAGE_AFFINITY_INTERVAL:

$\forall$ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (m::real) c::(real, ?'a::type) cart. content (IMAGE (*λ*x::(real, ?'a::type) cart. vector_add (% m x) c) (closed_interval [(a, b)])) =* $|m|^{dimindex\ HOL\_Light\_Import.UNIV}$ *∗ content (closed_interval [(a, b)])*

thm HAS_INTEGRAL_AFFINITY:

$\forall$ *(f::(real, ?'b::type) cart* $\Rightarrow$ *(real, ?'a::type) cart) (i::(real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (m::real) c::(real, ?'b::type) cart. has_integral f i (closed_interval [(a, b)])* $\wedge$ *m* $\neq$ *(0::real)* $\longrightarrow$ *has_integral (*λ*x::(real, ?'b::type) cart. f (vector_add (% m x) c)) (% (inverse_class.inverse* $|m|^{dimindex\ HOL\_Light\_Import.UNIV}$*) i) (IMAGE (*$\lambda$*x::(real, ?'b::type) cart. vector_add (% (inverse_class.inverse m) x) (vector_neg (% (inverse_class.inverse m) c))) (closed_interval [(a, b)]))*

thm INTEGRABLE_AFFINITY:

$\forall$ *(f::(real, ?'b::type) cart* $\Rightarrow$ *(real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (m::real) c::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)])* $\wedge$ *m* $\neq$ *(0::real)* $\longrightarrow$ *integrable_on (*λ*x::(real, ?'b::type) cart. f (vector_add (% m x) c)) (IMAGE (*λ*x::(real, ?'b::type) cart. vector_add (% (inverse_class.inverse m) x) (vector_neg (% (inverse_class.inverse m) c))) (closed_interval [(a, b)]))*

thm CONTENT_IMAGE_STRETCH_INTERVAL:

$\forall$ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) m::nat* $\Rightarrow$ *real. content (IMAGE (*λ*x::(real, ?'a::type) cart. lambda (*λ*k::nat. m k ∗ $ x k)) (closed_interval [(a, b)])) = |product (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) m| ∗ content (closed_interval [(a, b)])*

thm HAS_INTEGRAL_STRETCH:

$\forall$ *(f::(real, ?'b::type) cart* $\Rightarrow$ *(real, ?'a::type) cart) (i::(real, ?'a::type) cart) (m::nat* $\Rightarrow$ *real) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. has_integral f i (closed_interval [(a, b)])* $\wedge$ *(*$\forall$ *k::nat. (1::nat)* $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *m k* $\neq$ *(0::real))* $\longrightarrow$ *has_integral (*λ*x::(real, ?'b::type) cart. f (lambda (*λ*k::nat. m k ∗ $ x k))) (% (inverse_class.inverse |product (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) m|) i) (IMAGE (*λ*x::(real, ?'b::type) cart. lambda (*λ*k::nat. inverse_class.inverse (m k) ∗ $ x k)) (closed_interval [(a, b)]))*

thm INTEGRABLE_STRETCH:

$\forall (f::(real, ?'b::type) \ cart \Rightarrow (real, ?'a::type) \ cart) \ (m::nat \Rightarrow real) \ (a::(real, ?'b::type) \ cart) \ b::(real, ?'b::type) \ cart. \ integrable\_on \ f \ (closed\_interval \ [(a, b)]) \wedge (\forall k::nat. \ (1::nat) \leq k \wedge k \leq dimindex \ HOL\_Light\_Import.UNIV \longrightarrow m \ k \neq (0::real)) \longrightarrow integrable\_on \ (\lambda x::(real, ?'b::type) \ cart. \ f \ (lambda \ (\lambda k::nat. \ m \ k * \$ \ x \ k))) \ (IMAGE \ (\lambda x::(real, ?'b::type) \ cart. \ lambda \ (\lambda k::nat. \ inverse\_class.inverse \ (m \ k) * \$ \ x \ k)) \ (closed\_interval \ [(a, b)]))$

thm HAS_INTEGRAL_REFLECT_LEMMA:

$\forall (f::(real, ?'b::type) \ cart \Rightarrow (real, ?'a::type) \ cart) \ (i::(real, ?'a::type) \ cart) \ (a::(real, ?'b::type) \ cart) \ b::(real, ?'b::type) \ cart. \ has\_integral \ f \ i \ (closed\_interval \ [(a, b)]) \longrightarrow has\_integral \ (\lambda x::(real, ?'b::type) \ cart. \ f \ (vector\_neg \ x)) \ i \ (closed\_interval \ [(vector\_neg \ b, vector\_neg \ a)])$

thm HAS_INTEGRAL_REFLECT:

$\forall (f::(real, ?'b::type) \ cart \Rightarrow (real, ?'a::type) \ cart) \ (i::(real, ?'a::type) \ cart) \ (a::(real, ?'b::type) \ cart) \ b::(real, ?'b::type) \ cart. \ has\_integral \ (\lambda x::(real, ?'b::type) \ cart. \ f \ (vector\_neg \ x)) \ i \ (closed\_interval \ [(vector\_neg \ b, vector\_neg \ a)]) = has\_integral \ f \ i \ (closed\_interval \ [(a, b)])$

thm INTEGRABLE_REFLECT:

$\forall (f::(real, ?'b::type) \ cart \Rightarrow (real, ?'a::type) \ cart) \ (a::(real, ?'b::type) \ cart) \ b::(real, ?'b::type) \ cart. \ integrable\_on \ (\lambda x::(real, ?'b::type) \ cart. \ f \ (vector\_neg \ x)) \ (closed\_interval \ [(vector\_neg \ b, vector\_neg \ a)]) = integrable\_on \ f \ (closed\_interval \ [(a, b)])$

thm INTEGRAL_REFLECT:

$\forall (f::(real, ?'b::type) \ cart \Rightarrow (real, ?'a::type) \ cart) \ (a::(real, ?'b::type) \ cart) \ b::(real, ?'b::type) \ cart. \ integral \ (closed\_interval \ [(vector\_neg \ b, vector\_neg \ a)]) \ (\lambda x::(real, ?'b::type) \ cart. \ f \ (vector\_neg \ x)) = integral \ (closed\_interval \ [(a, b)]) \ f$

thm DIVISION_COMMON_POINT_BOUND:

$\forall (d::((real, ?'a::type) \ cart \Rightarrow bool) \Rightarrow bool) \ (s::(real, ?'a::type) \ cart \Rightarrow bool) \ x::(real, ?'a::type) \ cart. \ division\_of \ d \ s \longrightarrow CARD \ (GSPEC \ (\lambda GEN\%PVAR\%1797::(real, ?'a::type) \ cart \Rightarrow bool. \ \exists k::(real, ?'a::type) \ cart \Rightarrow bool. \ SETSPEC \ GEN\%PVAR\%1797 \ (IN \ k \ d \wedge content \ k \neq (0::real) \wedge IN \ x \ k) \ k)) \leq (2::nat)^{dimindex \ HOL\_Light\_Import.UNIV}$

thm TAGGED_PARTIAL_DIVISION_COMMON_POINT_BOUND:

$\forall (p::(real, ?'a::type) \ cart \times ((real, ?'a::type) \ cart \Rightarrow bool) \Rightarrow bool) \ (s::(real, ?'a::type) \ cart \Rightarrow bool) \ y::(real, ?'a::type) \ cart. \ tagged\_partial\_division\_of \ p \ s \longrightarrow CARD \ (GSPEC \ (\lambda GEN\%PVAR\%1798::(real, ?'a::type) \ cart \times ((real, ?'a::type) \ cart \Rightarrow bool). \ \exists (x::(real, ?'a::type) \ cart) \ k::(real, ?'a::type) \ cart \Rightarrow bool. \ SETSPEC \ GEN\%PVAR\%1798 \ (IN \ (x, k) \ p \wedge IN \ y \ k \wedge content \ k \neq (0::real)) \ (x, k))) \leq (2::nat)^{dimindex \ HOL\_Light\_Import.UNIV}$

thm TAGGED_PARTIAL_DIVISION_COMMON_TAGS:

$\forall$ (*p*::(*real*, *?'a*::*type*) *cart* $\times$ ((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *tagged_partial_division_of p s* $\longrightarrow$ *CARD* (*GSPEC* (λ*GEN%PVAR%1799*::(*real*, *?'a*::*type*) *cart* $\times$ ((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*). $\exists$ *k*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%1799* (*IN* (*x*, *k*) *p* $\wedge$ *content k* $\neq$ (*0*::*real*)) (*x*, *k*))) $\leq$ (*2*::*nat*)$^{dimindex\ HOL\_Light\_Import.UNIV}$

thm HAS_INTEGRAL_RESTRICT_OPEN_SUBINTERVAL:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart*) *i*::(*real*, *?'a*::*type*) *cart*. *has_integral f i* (*closed_interval* [(*c*, *d*)]) $\wedge$ *SUBSET* (*closed_interval* [(*c*, *d*)]) (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x* (*open_interval* (*c*, *d*)) *then f x else vec* (*0*::*nat*)) *i* (*closed_interval* [(*a*, *b*)])

thm HAS_INTEGRAL_RESTRICT_CLOSED_SUBINTERVAL:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart*) *i*::(*real*, *?'a*::*type*) *cart*. *has_integral f i* (*closed_interval* [(*c*, *d*)]) $\wedge$ *SUBSET* (*closed_interval* [(*c*, *d*)]) (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x* (*closed_interval* [(*c*, *d*)]) *then f x else vec* (*0*::*nat*)) *i* (*closed_interval* [(*a*, *b*)])

thm HAS_INTEGRAL_RESTRICT_CLOSED_SUBINTERVALS_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart*) *i*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*closed_interval* [(*c*, *d*)]) (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x* (*closed_interval* [(*c*, *d*)]) *then f x else vec* (*0*::*nat*)) *i* (*closed_interval* [(*a*, *b*)]) = *has_integral f i* (*closed_interval* [(*c*, *d*)])

thm HAS_INTEGRAL:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*i*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *has_integral f i s* = ($\forall$ *e*>*0*::*real*. $\exists$ *B*>*0*::*real*. $\forall$ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *SUBSET* (*ball* (*vec* (*0*::*nat*), *B*)) (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\exists$ *z*::(*real*, *?'a*::*type*) *cart*. *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *z* (*closed_interval* [(*a*, *b*)]) $\wedge$ *vector_norm* (*vector_sub z i*) < *e*))

thm HAS_INTEGRAL_RESTRICT:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *i*::(*real*, *?'a*::*type*) *cart*. *SUBSET s t* $\longrightarrow$ *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *i t* = *has_integral f i s*

thm INTEGRAL_RESTRICT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ⟶ *integral t* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) = *integral s f*

thm INTEGRABLE_RESTRICT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ⟶ *integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *t* = *integrable_on f s*

thm HAS_INTEGRAL_RESTRICT_UNIV:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *i*::(*real*, *?'a*::*type*) *cart*. *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *i HOL_Light_Import.UNIV* = *has_integral f i s*

thm INTEGRAL_RESTRICT_UNIV:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *integral HOL_Light_Import.UNIV* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) = *integral s f*

thm INTEGRABLE_RESTRICT_UNIV:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *HOL_Light_Import.UNIV* = *integrable_on f s*

thm HAS_INTEGRAL_RESTRICT_INTER:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *has_integral* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) (*?i*::(*real*, *?'a*::*type*) *cart*) *t* = *has_integral f ?i* (*HOL_Light_Import.INTER s t*)

thm INTEGRAL_RESTRICT_INTER:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *integral t* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) = *integral* (*HOL_Light_Import.INTER s t*) *f*

thm INTEGRABLE_RESTRICT_INTER:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) *t* = *integrable_on f* (*HOL_Light_Import.INTER s t*)

thm HAS_INTEGRAL_ON_SUPERSET:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. (∀ *x*::(*real*, *?'b*::*type*) *cart*. ¬ *IN x s* ⟶ *f x* = *vec* (*0*::*nat*)) ∧ *SUBSET s t* ∧ *has_integral f* (*?i*::(*real*, *?'a*::*type*) *cart*) *s* ⟶ *has_integral f ?i t*

thm INTEGRABLE_ON_SUPERSET:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real, ?'b::type)\ cart.\ \neg\ IN\ x\ s \longrightarrow f\ x = vec\ (0::nat)) \wedge SUBSET\ s\ t \wedge integrable\_on\ f\ s \longrightarrow integrable\_on\ f\ t$

thm NEGLIGIBLE_ON_INTERVALS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ negligible\ s = (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ negligible\ (HOL\_Light\_Import.INTER\ s\ (closed\_interval\ [(a, b)])))$

thm HAS_INTEGRAL_SPIKE_SET_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (t::(real, ?'b::type)\ cart \Rightarrow bool)\ y::(real, ?'a::type)\ cart.\ negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow has\_integral\ f\ y\ s = has\_integral\ f\ y\ t$

thm HAS_INTEGRAL_SPIKE_SET:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ (t::(real, ?'b::type)\ cart \Rightarrow bool)\ y::(real, ?'a::type)\ cart.\ negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \wedge has\_integral\ f\ y\ s \longrightarrow has\_integral\ f\ y\ t$

thm INTEGRABLE_SPIKE_SET:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow integrable\_on\ f\ s \longrightarrow integrable\_on\ f\ t$

thm INTEGRABLE_SPIKE_SET_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow integrable\_on\ f\ s = integrable\_on\ f\ t$

thm INTEGRAL_SPIKE_SET:

$\forall (f::(real, ?'c::type)\ cart \Rightarrow (real, ?'b::type)\ cart)\ (g::?'a::type)\ (s::(real, ?'c::type)\ cart \Rightarrow bool)\ t::(real, ?'c::type)\ cart \Rightarrow bool.\ negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow integral\ s\ f = integral\ t\ f$

thm HAS_INTEGRAL_INTERIOR:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ negligible\ (frontier\ s) \longrightarrow has\_integral\ f\ y\ (interior\ s) = has\_integral\ f\ y\ s$

thm HAS_INTEGRAL_CLOSURE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ negligible\ (frontier\ s) \longrightarrow has\_integral\ f\ y\ (closure\ s) = has\_integral\ f\ y\ s$

thm HAS_INTEGRAL_SUBSET_COMPONENT_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (i::(real,\ ?'a::type)\ cart)\ (j::(real,\ ?'a::type)\ cart)\ k::nat.\ SUBSET\ s\ t \wedge has\_integral\ f\ i\ s \wedge has\_integral\ f\ j\ t \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ t \longrightarrow (0::real) \leq \$\ (f\ x)\ k) \longrightarrow \$\ i\ k \leq \$\ j\ k$

thm INTEGRAL_SUBSET_COMPONENT_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ k::nat.\ SUBSET\ s\ t \wedge integrable\_on\ f\ s \wedge integrable\_on\ f\ t \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ t \longrightarrow (0::real) \leq \$\ (f\ x)\ k) \longrightarrow \$\ (integral\ s\ f)\ k \leq \$\ (integral\ t\ f)\ k$

thm HAS_INTEGRAL_SUBSET_DROP_LE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (i::(real,\ unit)\ cart)\ j::(real,\ unit)\ cart.\ SUBSET\ s\ t \wedge has\_integral\ f\ i\ s \wedge has\_integral\ f\ j\ t \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ t \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow HOL\_Light\_Import.drop\ i \leq HOL\_Light\_Import.drop\ j$

thm INTEGRAL_SUBSET_DROP_LE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge integrable\_on\ f\ s \wedge integrable\_on\ f\ t \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ t \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow HOL\_Light\_Import.drop\ (integral\ s\ f) \leq HOL\_Light\_Import.drop\ (integral\ t\ f)$

thm HAS_INTEGRAL_ALT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ i::(real,\ ?'a::type)\ cart.\ has\_integral\ f\ i\ s = ((\forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ if\ IN\ x\ s\ then\ f\ x\ else\ vec\ (0::nat))\ (closed\_interval\ [(a,\ b)])) \wedge (\forall e>0::real.\ \exists B>0::real.\ \forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ SUBSET\ (ball\ (vec\ (0::nat),\ B))\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ (integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'b::type)\ cart.\ if\ IN\ x\ s\ then\ f\ x\ else\ vec\ (0::nat)))\ i) < e))$

thm INTEGRABLE_ALT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s = ((\forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ if\ IN\ x\ s\ then\ f\ x\ else\ vec\ (0::nat))\ (closed\_interval\ [(a,\ b)])) \wedge (\forall e>0::real.\ \exists B>0::real.\ \forall (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (c::(real,\ ?'b::type)\ cart)\ d::(real,\ ?'b::type)\ cart.\ SUBSET\ (ball\ (vec\ (0::nat),\ B))\ (closed\_interval\ [(a,\ b)]) \wedge SUBSET\ (ball\ (vec\ (0::nat),\ B))\ (closed\_interval\ [(c,\ d)]) \longrightarrow vector\_norm\ (vector\_sub\ (integral$

(*closed_interval* [(*a*, *b*)]) (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*))) (*integral* (*closed_interval* [(*c*, *d*)]) (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)))) < *e*))

thm INTEGRABLE_ALT_SUBSET:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *integrable_on f s* = ((∀ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) (*closed_interval* [(*a*, *b*)])) ∧ (∀ *e*>*0*::*real*. ∃ *B*>*0*::*real*. ∀ (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'b*::*type*) *cart*) *d*::(*real*, *?'b*::*type*) *cart*. *SUBSET* (*ball* (*vec* (*0*::*nat*), *B*)) (*closed_interval* [(*a*, *b*)]) ∧ *SUBSET* (*closed_interval* [(*a*, *b*)]) (*closed_interval* [(*c*, *d*)]) ⟶ *vector_norm* (*vector_sub* (*integral* (*closed_interval* [(*a*, *b*)]) (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*))) (*integral* (*closed_interval* [(*c*, *d*)]) (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)))) < *e*))

thm INTEGRABLE_ON_SUBINTERVAL:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f s* ∧ *SUBSET* (*closed_interval* [(*a*, *b*)]) *s* ⟶ *integrable_on f* (*closed_interval* [(*a*, *b*)])

thm INTEGRAL_SPLIT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*t*::*real*) *k*::*nat*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *integral* (*closed_interval* [(*a*, *b*)]) *f* = *vector_add* (*integral* (*closed_interval* [(*a*, *lambda* (λ*i*::*nat*. *if i* = *k then min* ($ *b k*) *t else* $ *b i*))]) *f*) (*integral* (*closed_interval* [(*lambda* (λ*i*::*nat*. *if i* = *k then max* ($ *a k*) *t else* $ *a i*), *b*)]) *f*)

thm INTEGRAL_SPLIT_SIGNED:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*t*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ $ *a k* ≤ *t* ∧ $ *a k* ≤ $ *b k* ∧ *integrable_on f* (*closed_interval* [(*a*, *lambda* (λ*i*::*nat*. *if i* = *k then max* ($ *b k*) *t else* $ *b i*))]) ⟶ *integral* (*closed_interval* [(*a*, *b*)]) *f* = *vector_add* (*integral* (*closed_interval* [(*a*, *lambda* (λ*i*::*nat*. *if i* = *k then t else* $ *b i*))]) *f*) (% (*if* $ *b k* < *t then* − (*1*::*real*) *else* (*1*::*real*)) (*integral* (*closed_interval* [(*lambda* (λ*i*::*nat*. *if i* = *k then min* ($ *b k*) *t else* $ *a i*), *lambda* (λ*i*::*nat*. *if i* = *k then max* ($ *b k*) *t else* $ *b i*))]) *f*))

thm POWERSET_CLAUSES_conjunct1:

∀ (*a*::*?'a*::*type*) *t*::*?'a*::*type* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%99*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%99* (*SUBSET s* (*INSERT a t*)) *s*) = *HOL_Light_Import.UNION* (*GSPEC* (λ*GEN%PVAR%100*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%100* (*SUBSET s t*) *s*)) (*IMAGE* (*INSERT a*) (*GSPEC* (λ*GEN%PVAR%101*::*?'a*::*type* ⇒ *bool*. ∃ *s*::*?'a*::*type* ⇒ *bool*. *SETSPEC GEN%PVAR%101* (*SUBSET s t*) *s*)))

thm INTEGRAL_INTERVALS_INCLUSION_EXCLUSION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (c::(real, ?'b::type) cart) d::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)]) ∧ IN c (closed_interval [(a, b)]) ∧ IN d (closed_interval [(a, b)]) ⟶ integral (closed_interval [(a, d)]) f = vsum (GSPEC (λGEN%PVAR%1811::nat ⇒ bool. ∃ s::nat ⇒ bool. SETSPEC GEN%PVAR%1811 (SUBSET s (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV))) s)) (λs::nat ⇒ bool. % (− (1::real))$^{CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1812::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%1812\ (IN\ i\ s\ \wedge\ \$\ d\ i\ <\ \$\ c\ i)\ i))}$ (integral (closed_interval [(lambda (λi::nat. if IN i s then min ($ c i) ($ d i) else $ a i), lambda (λi::nat. if IN i s then max ($ c i) ($ d i) else $ c i))]) f))

thm INTEGRAL_INTERVALS_DIFF_INCLUSION_EXCLUSION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (c::(real, ?'b::type) cart) d::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)]) ∧ IN c (closed_interval [(a, b)]) ∧ IN d (closed_interval [(a, b)]) ⟶ vector_sub (integral (closed_interval [(a, d)]) f) (integral (closed_interval [(a, c)]) f) = vsum (GSPEC (λGEN%PVAR%1815::nat ⇒ bool. ∃ s::nat ⇒ bool. SETSPEC GEN%PVAR%1815 (s ≠ EMPTY ∧ SUBSET s (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV))) s)) (λs::nat ⇒ bool. % (− (1::real))$^{CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1816}$ (integral (closed_interval [(lambda (λi::nat. if IN i s then min ($ c i) ($ d i) else $ a i), lambda (λi::nat. if IN i s then max ($ c i) ($ d i) else $ c i))]) f))

thm INTEGRAL_INTERVALS_INCLUSION_EXCLUSION_RIGHT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) c::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)]) ∧ IN c (closed_interval [(a, b)]) ⟶ integral (closed_interval [(a, c)]) f = vsum (GSPEC (λGEN%PVAR%1817::nat ⇒ bool. ∃ s::nat ⇒ bool. SETSPEC GEN%PVAR%1817 (SUBSET s (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV))) s)) (λs::nat ⇒ bool. % (− (1::real))$^{CARD\ s}$ (integral (closed_interval [(lambda (λi::nat. if IN i s then $ c i else $ a i), b)]) f))

thm INTEGRAL_INTERVALS_INCLUSION_EXCLUSION_LEFT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) c::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)]) ∧ IN c (closed_interval [(a, b)]) ⟶ integral (closed_interval [(c, b)]) f = vsum (GSPEC (λGEN%PVAR%1818::nat ⇒ bool. ∃ s::nat ⇒ bool. SETSPEC GEN%PVAR%1818 (SUBSET s (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV))) s)) (λs::nat ⇒ bool. % (− (1::real))$^{CARD\ s}$ (integral (closed_interval [(a, lambda (λi::nat. if IN i s then $ c i else $ b i))]) f))

thm INTEGRABLE_STRADDLE_INTERVAL:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (∀ e>0::real. ∃ (g::(real, ?'a::type) cart ⇒ (real, unit) cart) (h::(real, ?'a::type) cart ⇒ (real, unit) cart) (i::(real, unit) cart) j::(real, unit) cart. has_integral g i (closed_interval [(a, b)]) ∧ has_integral h j (closed_interval

$[(a,\ b)]) \wedge vector\_norm\ (vector\_sub\ i\ j) < e \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow HOL\_Light\_Import.drop\ (g\ x) \leq HOL\_Light\_Import.drop\ (f\ x) \wedge HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (h\ x))) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_STRADDLE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall e>0::real.\ \exists (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (i::(real,\ unit)\ cart)\ j::(real,\ unit)\ cart.\ has\_integral\ g\ i\ s \wedge has\_integral\ h\ j\ s \wedge vector\_norm\ (vector\_sub\ i\ j) < e \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (g\ x) \leq HOL\_Light\_Import.drop\ (f\ x) \wedge HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (h\ x))) \longrightarrow integrable\_on\ f\ s$

thm HAS_INTEGRAL_STRADDLE_NULL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq HOL\_Light\_Import.drop\ (f\ x) \wedge HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \wedge has\_integral\ g\ (vec\ (0::nat))\ s \longrightarrow has\_integral\ f\ (vec\ (0::nat))\ s$

thm HAS_INTEGRAL_UNION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (i::(real,\ ?'a::type)\ cart)\ (j::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ i\ s \wedge has\_integral\ f\ j\ t \wedge negligible\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow has\_integral\ f\ (vector\_add\ i\ j)\ (HOL\_Light\_Import.UNION\ s\ t)$

thm INTEGRAL_UNION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge integrable\_on\ f\ t \wedge negligible\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow integral\ (HOL\_Light\_Import.UNION\ s\ t)\ f = vector\_add\ (integral\ s\ f)\ (integral\ t\ f)$

thm HAS_INTEGRAL_UNIONS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (i::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ t::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ t \wedge (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ s\ t \longrightarrow has\_integral\ f\ (i\ s)\ s) \wedge (\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ s'::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ s\ t \wedge IN\ s'\ t \wedge s \neq s' \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ s\ s')) \longrightarrow has\_integral\ f\ (vsum\ t\ i)\ (UNIONS\ t)$

thm HAS_INTEGRAL_DIFF:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (i::(real,\ ?'a::type)\ cart)\ (j::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_integral\ f\ i\ s \wedge has\_integral\ f\ j\ t \wedge negligible\ (DIFF\ t\ s) \longrightarrow has\_integral\ f\ (vector\_sub\ i\ j)\ (DIFF\ s\ t)$

thm INTEGRAL_DIFF:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge integrable\_on\ f\ t \wedge negligible\ (DIFF\ t\ s) \longrightarrow integral\ (DIFF\ s\ t)\ f = vector\_sub\ (integral\ s\ f)\ (integral\ t\ f)$

thm HAS_INTEGRAL_COMBINE_DIVISION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ division\_of\ d\ s \wedge (\forall k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow has\_integral\ f\ (i\ k)\ k) \longrightarrow has\_integral\ f\ (vsum\ d\ i)\ s$

thm INTEGRAL_COMBINE_DIVISION_BOTTOMUP:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ s \wedge (\forall k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow integrable\_on\ f\ k) \longrightarrow integral\ s\ f = vsum\ d\ (\lambda i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integral\ i\ f)$

thm HAS_INTEGRAL_COMBINE_DIVISION_TOPDOWN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge division\_of\ d\ k \wedge SUBSET\ k\ s \longrightarrow has\_integral\ f\ (vsum\ d\ (\lambda i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integral\ i\ f))\ k$

thm INTEGRAL_COMBINE_DIVISION_TOPDOWN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \wedge division\_of\ d\ s \longrightarrow integral\ s\ f = vsum\ d\ (\lambda i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integral\ i\ f)$

thm INTEGRABLE_COMBINE_DIVISION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ s \wedge (\forall i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ i\ d \longrightarrow integrable\_on\ f\ i) \longrightarrow integrable\_on\ f\ s$

thm INTEGRABLE_ON_SUBDIVISION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ i \wedge integrable\_on\ f\ s \wedge SUBSET\ i\ s \longrightarrow integrable\_on\ f\ i$

thm HAS_INTEGRAL_COMBINE_TAGGED_DIVISION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (p::(real,\ ?'b::type)\ cart \times ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ i::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ tagged\_division\_of\ p\ s \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ (x,\ k)\ p \longrightarrow has\_integral\ f\ (i\ k)\ k) \longrightarrow has\_integral\ f\ (vsum\ p\ (GABS\ (\lambda f::(real,\ ?'b::type)$

cart × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*f* (*x*, *k*)) (*i k*)))) *s*

thm INTEGRAL_COMBINE_TAGGED_DIVISION_BOTTOMUP:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *IN* (*x*, *k*) *p* ⟶ *integrable_on f k*) ⟶ *integral* (*closed_interval* [(*a*, *b*)]) *f* = *vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (*integral k f*)))

thm HAS_INTEGRAL_COMBINE_TAGGED_DIVISION_TOPDOWN:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) ⟶ *has_integral f* (*vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (*integral k f*)))) (*closed_interval* [(*a*, *b*)])

thm INTEGRAL_COMBINE_TAGGED_DIVISION_TOPDOWN:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) ⟶ *integral* (*closed_interval* [(*a*, *b*)]) *f* = *vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (*integral k f*)))

thm HENSTOCK_LEMMA_PART1:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ *bool*) *e*::*real*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ (*0*::*real*) < *e* ∧ *gauge d* ∧ (∀ *p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) ∧ *fine d p* ⟶ *vector_norm* (*vector_sub* (*vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*))))) (*integral* (*closed_interval* [(*a*, *b*)]) *f*)) < *e*) ⟶ (∀ *p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *tagged_partial_division_of p* (*closed_interval* [(*a*, *b*)]) ∧ *fine d p* ⟶ *vector_norm* (*vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*. ∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *GEQ* (*fa* (*x*, *k*)) (*vector_sub* (% (*content k*) (*f x*)) (*integral k f*))))) ≤ *e*)

thm HENSTOCK_LEMMA_PART2:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *e*::*real*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) $\wedge$ (*0*::*real*) < *e* $\wedge$ *gauge d* $\wedge$ ($\forall$ *p*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *tagged_division_of p* (*closed_interval* [(*a*, *b*)]) $\wedge$ *fine d p* $\longrightarrow$ *vector_norm* (*vector_sub* (*vsum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. $\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *GEQ* (*fa* (*x*, *k*)) (% (*content k*) (*f x*))))) (*integral* (*closed_interval* [(*a*, *b*)]) *f*)) < *e*) $\longrightarrow$ ($\forall$ *p*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *tagged_partial_division_of p* (*closed_interval* [(*a*, *b*)]) $\wedge$ *fine d p* $\longrightarrow$ *sum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *real*. $\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *GEQ* (*fa* (*x*, *k*)) (*vector_norm* (*vector_sub* (% (*content k*) (*f x*)) (*integral k f*))))) $\leq$ *real_of_nat* (*2*::*nat*) $*$ (*real_of_nat* (*dimindex HOL_Light_Import.UNIV*) $*$ *e*))

thm HENSTOCK_LEMMA:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\forall$ *e*>*0*::*real*. $\exists$ *d*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *gauge d* $\wedge$ ($\forall$ *p*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *tagged_partial_division_of p* (*closed_interval* [(*a*, *b*)]) $\wedge$ *fine d p* $\longrightarrow$ *sum p* (*GABS* (λ*fa*::(*real*, *?'b*::*type*) *cart* $\times$ ((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *real*. $\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *GEQ* (*fa* (*x*, *k*)) (*vector_norm* (*vector_sub* (% (*content k*) (*f x*)) (*integral k f*))))) < *e*))

thm MONOTONE_CONVERGENCE_INTERVAL:

$\forall$ (*f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ($\forall$ *k*::*nat*. *integrable_on* (*f k*) (*closed_interval* [(*a*, *b*)])) $\wedge$ ($\forall$ (*k*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *HOL_Light_Import.drop* (*f k x*) $\leq$ *HOL_Light_Import.drop* (*f* (*Suc k*) *x*)) $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *-->* (λ*k*::*nat*. *f k x*) (*g x*) *sequentially*) $\wedge$ *bounded* (*GSPEC* (λ*GEN%PVAR%1828*::(*real*, *unit*) *cart*. $\exists$ *k*::*nat*. *SETSPEC GEN%PVAR%1828* (*IN k HOL_Light_Import.UNIV*) (*integral* (*closed_interval* [(*a*, *b*)]) (*f k*)))) $\longrightarrow$ *integrable_on g* (*closed_interval* [(*a*, *b*)]) $\wedge$ *-->* (λ*k*::*nat*. *integral* (*closed_interval* [(*a*, *b*)]) (*f k*)) (*integral* (*closed_interval* [(*a*, *b*)]) *g*) *sequentially*

thm MONOTONE_CONVERGENCE_INCREASING:

$\forall$ (*f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *k*::*nat*. *integrable_on* (*f k*) *s*) $\wedge$ ($\forall$ (*k*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ *HOL_Light_Import.drop* (*f k x*) $\leq$ *HOL_Light_Import.drop* (*f* (*Suc k*) *x*)) $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ *-->* (λ*k*::*nat*. *f k x*) (*g x*) *sequentially*) $\wedge$ *bounded* (*GSPEC* (λ*GEN%PVAR%1830*::(*real*, *unit*) *cart*. $\exists$ *k*::*nat*. *SETSPEC GEN%PVAR%1830* (*IN k HOL_Light_Import.UNIV*) (*integral s* (*f k*)))) $\longrightarrow$ *integrable_on g s* $\wedge$ *-->* (λ*k*::*nat*. *integral s* (*f k*)) (*integral s g*) *sequentially*

thm MONOTONE_CONVERGENCE_DECREASING:

$\forall$ (*f*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *k*::*nat*. *integrable_on* (*f k*) *s*) $\wedge$ ($\forall$ (*k*::*nat*) *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ *HOL_Light_Import.drop* (*f* (*Suc k*) *x*) $\leq$ *HOL_Light_Import.drop* (*f k x*)) $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ *-->* (λ*k*::*nat*. *f k x*) (*g x*) *sequentially*) $\wedge$ *bounded* (*GSPEC* (λ*GEN%PVAR%1832*::(*real*, *unit*) *cart*. $\exists$ *k*::*nat*. *SETSPEC GEN%PVAR%1832* (*IN k HOL_Light_Import.UNIV*) (*integral s* (*f k*)))) $\longrightarrow$ *integrable_on g s* $\wedge$ *-->* (λ*k*::*nat*. *integral s* (*f k*)) (*integral s g*) *sequentially*

thm INTEGRAL_NORM_BOUND_INTEGRAL:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *integrable_on f s* $\wedge$ *integrable_on g s* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* $\longrightarrow$ *vector_norm* (*f x*) $\leq$ *HOL_Light_Import.drop* (*g x*)) $\longrightarrow$ *vector_norm* (*integral s f*) $\leq$ *HOL_Light_Import.drop* (*integral s g*)

thm INTEGRAL_NORM_BOUND_INTEGRAL_COMPONENT:

$\forall$ (*f*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ *bool*) *k*::*nat*. (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *integrable_on f s* $\wedge$ *integrable_on g s* $\wedge$ ($\forall$ *x*::(*real*, *?'c*::*type*) *cart*. *IN x s* $\longrightarrow$ *vector_norm* (*f x*) $\leq$ \$ (*g x*) *k*) $\longrightarrow$ *vector_norm* (*integral s f*) $\leq$ \$ (*integral s g*) *k*

thm HAS_INTEGRAL_NORM_BOUND_INTEGRAL_COMPONENT:

$\forall$ (*f*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ *bool*) (*i*::(*real*, *?'b*::*type*) *cart*) (*j*::(*real*, *?'a*::*type*) *cart*) *k*::*nat*. (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *has_integral f i s* $\wedge$ *has_integral g j s* $\wedge$ ($\forall$ *x*::(*real*, *?'c*::*type*) *cart*. *IN x s* $\longrightarrow$ *vector_norm* (*f x*) $\leq$ \$ (*g x*) *k*) $\longrightarrow$ *vector_norm i* $\leq$ \$ *j k*

thm INTEGRABLE_ON_ALL_INTERVALS_INTEGRABLE_BOUND:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *if IN x s then f x else vec* (*0*::*nat*)) (*closed_interval* [(*a*, *b*)])) $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* $\longrightarrow$ *vector_norm* (*f x*) $\leq$ *HOL_Light_Import.drop* (*g x*)) $\wedge$ *integrable_on g s* $\longrightarrow$ *integrable_on f s*

thm DEF_set_variation:

set_variation = (λ(*_1664647*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *_1664648*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.sup* (*GSPEC* (λ*GEN%PVAR%1833*::*real*. $\exists$ *d*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%1833* ($\exists$ *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *division_of d t* $\wedge$ *SUBSET t _1664647*) (*sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *vector_norm* (*_1664648 k*))))))

thm set_variation:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ set\_variation\ s\ f = HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%1833::real.\ \exists\ d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1833\ (\exists\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s)\ (sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)))))$

thm DEF_has_bounded_setvariation_on:

$has\_bounded\_setvariation\_on = (\lambda(\_1664659::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ \_1664660::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \exists\ B::real.\ \forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ \_1664660 \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (\_1664659\ k)) \leq B)$

thm has_bounded_setvariation_on:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ has\_bounded\_setvariation\_on\ f\ s = (\exists\ B::real.\ \forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B)$

thm HAS_BOUNDED_SETVARIATION_ON:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s = (\exists\ B{>}0::real.\ \forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B)$

thm HAS_BOUNDED_SETVARIATION_ON_EQ:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (g::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ s \longrightarrow f\ (closed\_interval\ [(a,\ b)]) = g\ (closed\_interval\ [(a,\ b)])) \wedge has\_bounded\_setvariation\_on\ f\ s \longrightarrow has\_bounded\_setvariation\_on\ g\ s$

thm SET_VARIATION_EQ:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (g::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ s \longrightarrow f\ (closed\_interval\ [(a,\ b)]) = g\ (closed\_interval\ [(a,\ b)])) \longrightarrow set\_variation\ s\ f = set\_variation\ s\ g$

thm HAS_BOUNDED_SETVARIATION_ON_COMPONENTWISE:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s = (\forall\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow has\_bounded\_setvariation\_on\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ lift\ (\$\ (f\ k)\ i))\ s)$

thm SETVARIATION_EQUAL_LEMMA:

$\forall (mf::(((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow ((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (ms::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow bool)\ ms'::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ ms'\ (ms\ s) = s \wedge ms\ (ms'\ s) = s) \wedge (\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow mf\ f\ (ms\ (closed\_interval\ [(a,\ b)])) = f\ (closed\_interval\ [(a,\ b)]) \wedge (\exists (a'::(real,\ ?'b::type)\ cart)\ b'::(real,\ ?'b::type)\ cart.\ closed\_interval\ [(a',\ b')] \neq EMPTY \wedge ms'\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(a',\ b')])) \wedge (\forall (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ u::(real,\ ?'b::type)\ cart \Rightarrow bool.\ SUBSET\ t\ u \longrightarrow SUBSET\ (ms\ t)\ (ms\ u) \wedge SUBSET\ (ms'\ t)\ (ms'\ u)) \wedge (\forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \longrightarrow division\_of\ (IMAGE\ ms\ d)\ (ms\ t) \wedge division\_of\ (IMAGE\ ms'\ d)\ (ms'\ t)) \longrightarrow (\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ (mf\ f)\ (ms\ s) = has\_bounded\_setvariation\_on\ f\ s) \wedge (\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ set\_variation\ (ms\ s)\ (mf\ f) = set\_variation\ s\ f)$

thm HAS_BOUNDED_SETVARIATION_ON_ELEMENTARY:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\exists\ d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ s) \longrightarrow has\_bounded\_setvariation\_on\ f\ s = (\exists\ B::real.\ \forall\ d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B)$

thm HAS_BOUNDED_SETVARIATION_ON_INTERVAL:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ has\_bounded\_setvariation\_on\ f\ (closed\_interval\ [(a,\ b)]) = (\exists\ B::real.\ \forall\ d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ (closed\_interval\ [(a,\ b)]) \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B)$

thm HAS_BOUNDED_SETVARIATION_ON_UNIV:

$\forall f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart.\ has\_bounded\_setvariation\_on\ f\ HOL\_Light\_Import.UNIV = (\exists\ B::real.\ \forall\ d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ (UNIONS\ d) \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B)$

thm HAS_BOUNDED_SETVARIATION_ON_SUBSET:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) (s::(real, ?'b::type) cart \Rightarrow bool) t::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \wedge SUBSET t s \longrightarrow has\_bounded\_setvariation\_on f t$

thm HAS_BOUNDED_SETVARIATION_ON_IMP_BOUNDED_ON_SUBINTERVALS:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \longrightarrow bounded (GSPEC (\lambda GEN\%PVAR\%1836::(real, ?'a::type) cart. \exists (c::(real, ?'b::type) cart) d::(real, ?'b::type) cart. SETSPEC GEN\%PVAR\%1836 (SUBSET (closed\_interval [(c, d)]) s) (f (closed\_interval [(c, d)]))))$

thm HAS_BOUNDED_SETVARIATION_ON_NORM:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \longrightarrow has\_bounded\_setvariation\_on (\lambda x::(real, ?'b::type) cart \Rightarrow bool. lift (vector\_norm (f x))) s$

thm HAS_BOUNDED_SETVARIATION_ON_COMPOSE_LINEAR:

$\forall (f::((real, ?'c::type) cart \Rightarrow bool) \Rightarrow (real, ?'b::type) cart) (g::(real, ?'b::type) cart \Rightarrow (real, ?'a::type) cart) s::(real, ?'c::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \wedge linear g \longrightarrow has\_bounded\_setvariation\_on (g \circ f) s$

thm HAS_BOUNDED_SETVARIATION_ON_0:

$\forall s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on (\lambda x::(real, ?'b::type) cart \Rightarrow bool. vec (0::nat)) s$

thm SET_VARIATION_0:

$\forall s::(real, ?'b::type) cart \Rightarrow bool. set\_variation s (\lambda x::(real, ?'b::type) cart \Rightarrow bool. vec (0::nat)) = (0::real)$

thm HAS_BOUNDED_SETVARIATION_ON_CMUL:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) (c::real) s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \longrightarrow has\_bounded\_setvariation\_on (\lambda x::(real, ?'b::type) cart \Rightarrow bool. \% c (f x)) s$

thm HAS_BOUNDED_SETVARIATION_ON_NEG:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \longrightarrow has\_bounded\_setvariation\_on (\lambda x::(real, ?'b::type) cart \Rightarrow bool. vector\_neg (f x)) s$

thm HAS_BOUNDED_SETVARIATION_ON_ADD:

$\forall (f::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) (g::((real, ?'b::type) cart \Rightarrow bool) \Rightarrow (real, ?'a::type) cart) s::(real, ?'b::type) cart \Rightarrow bool. has\_bounded\_setvariation\_on f s \wedge has\_bounded\_setvariation\_on g s \longrightarrow has\_bounded\_setvariation\_on (\lambda x::(real, ?'b::type) cart \Rightarrow bool. vector\_add (f x) (g x)) s$

thm HAS_BOUNDED_SETVARIATION_ON_SUB:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (g::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s \wedge has\_bounded\_setvariation\_on\ g\ s \longrightarrow has\_bounded\_setvariation\_on\ (\lambda x::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm HAS_BOUNDED_SETVARIATION_ON_NULL:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow f\ (closed\_interval\ [(a,\ b)]) = vec\ (0::nat)) \wedge content\ s = (0::real) \wedge bounded\ s \longrightarrow has\_bounded\_setvariation\_on\ f\ s$

thm SET_VARIATION_ELEMENTARY_LEMMA:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\exists\, d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ s) \longrightarrow (\forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq (?b::real)) = (\forall\, d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq ?b)$

thm SET_VARIATION_ON_ELEMENTARY:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\exists\, d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ s) \longrightarrow set\_variation\ s\ f = HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%1837::real.\ \exists\, d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1837\ (division\_of\ d\ s)\ (sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)))))$

thm SET_VARIATION_ON_INTERVAL:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ set\_variation\ (closed\_interval\ [(a,\ b)])\ f = HOL\_Light\_Import.sup\ (GSPEC\ (\lambda GEN\%PVAR\%1838::real.\ \exists\, d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1838\ (division\_of\ d\ (closed\_interval\ [(a,\ b)]))\ (sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)))))$

thm HAS_BOUNDED_SETVARIATION_WORKS:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s \longrightarrow (\forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq set\_variation\ s\ f) \wedge (\forall\, B::real.\ (\forall (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ division\_of\ d\ t \wedge SUBSET\ t\ s \longrightarrow sum\ d\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ vector\_norm\ (f\ k)) \leq B) \longrightarrow set\_variation\ s\ f \leq B)$

thm HAS_BOUNDED_SETVARIATION_WORKS_ON_ELEMENTARY:

$\forall (f::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s \wedge (\exists\, d::((real,\ ?'b::type)\ cart \Rightarrow$

bool) ⇒ *bool*. *division_of d s*) ⟶ (∀ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d s* ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *set_variation s f*) ∧ (∀ *B*::*real*. (∀ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d s* ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *B*) ⟶ *set_variation s f* ≤ *B*)

thm HAS_BOUNDED_SETVARIATION_WORKS_ON_INTERVAL:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *has_bounded_setvariation_on f* (*closed_interval* [(*a*, *b*)]) ⟶ (∀ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *set_variation* (*closed_interval* [(*a*, *b*)]) *f*) ∧ (∀ *B*::*real*. (∀ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *B*) ⟶ *set_variation* (*closed_interval* [(*a*, *b*)]) *f* ≤ *B*)

thm SET_VARIATION_UBOUND:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *B*::*real*. *has_bounded_setvariation_on f s* ∧ (∀ (*d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *division_of d t* ∧ *SUBSET t s* ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *B*) ⟶ *set_variation s f* ≤ *B*

thm SET_VARIATION_UBOUND_ON_INTERVAL:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *B*::*real*. *has_bounded_setvariation_on f* (*closed_interval* [(*a*, *b*)]) ∧ (∀ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*)) ≤ *B*) ⟶ *set_variation* (*closed_interval* [(*a*, *b*)]) *f* ≤ *B*

thm SET_VARIATION_LBOUND:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *B*::*real*. *has_bounded_setvariation_on f s* ∧ (∃ (*d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *division_of d t* ∧ *SUBSET t s* ∧ *B* ≤ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*))) ⟶ *B* ≤ *set_variation s f*

thm SET_VARIATION_LBOUND_ON_INTERVAL:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *B*::*real*. *has_bounded_setvariation_on f* (*closed_interval* [(*a*, *b*)]) ∧ (∃ *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *division_of d* (*closed_interval* [(*a*, *b*)]) ∧ *B* ≤ *sum d* (λ*k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_norm* (*f k*))) ⟶ *B* ≤ *set_variation* (*closed_interval* [(*a*, *b*)]) *f*

thm SET_VARIATION:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *t*::(*real*, *?'b*::*type*) *cart*

⇒ *bool. has_bounded_setvariation_on f s* ∧ *division_of d t* ∧ *SUBSET t s* ⟶ *sum d* (*λk*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. vector_norm* (*f k*)) ≤ *set_variation s f*

thm SET_VARIATION_WORKS_ON_INTERVAL:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool. has_bounded_setvariation_on f* (*closed_interval* [(*a*, *b*)]) ∧ *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ *sum d* (*λk*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. vector_norm* (*f k*)) ≤ *set_variation* (*closed_interval* [(*a*, *b*)]) *f*

thm SET_VARIATION_POS_LE:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. has_bounded_setvariation_on f s* ⟶ (*0*::*real*) ≤ *set_variation s f*

thm SET_VARIATION_GE_FUNCTION:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart. has_bounded_setvariation_on f s* ∧ *SUBSET* (*closed_interval* [(*a*, *b*)]) *s* ∧ *closed_interval* [(*a*, *b*)] ≠ *EMPTY* ⟶ *vector_norm* (*f* (*closed_interval* [(*a*, *b*)])) ≤ *set_variation s f*

thm SET_VARIATION_ON_NULL:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. (∀ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart. content* (*closed_interval* [(*a*, *b*)]) = (*0*::*real*) ⟶ *f* (*closed_interval* [(*a*, *b*)]) = *vec* (*0*::*nat*)) ∧ *content s* = (*0*::*real*) ∧ *bounded s* ⟶ *set_variation s f* = (*0*::*real*)

thm SET_VARIATION_TRIANGLE:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. has_bounded_setvariation_on f s* ∧ *has_bounded_setvariation_on g s* ⟶ *set_variation s* (*λx*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. vector_add* (*f x*) (*g x*)) ≤ *set_variation s f* + *set_variation s g*

thm OPERATIVE_LIFTED_SETVARIATION:

∀ *f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart. operative vector_add f* ⟶ *operative* (*lifted op* +) (*λi*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. if has_bounded_setvariation_on f i then SOME* (*set_variation i f*) *else NONE*)

thm HAS_BOUNDED_SETVARIATION_ON_DIVISION:

∀ (*f*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *d*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool. operative vector_add f* ∧ *division_of d* (*closed_interval* [(*a*, *b*)]) ⟶ (∀ *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool. IN k d* ⟶ *has_bounded_setvariation_on f k*) = *has_bounded_setvariation_on f* (*closed_interval* [(*a*, *b*)])

thm SET_VARIATION_ON_DIVISION:

$\forall (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, ?'b::type)\ cart)\ (b::(real, ?'b::type)\ cart)\ d::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ operative\ vector\_add\ f \wedge division\_of\ d\ (closed\_interval\ [(a, b)]) \wedge has\_bounded\_setvariation\_on\ f\ (closed\_interval\ [(a, b)]) \longrightarrow sum\ d\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ set\_variation\ k\ f) = set\_variation\ (closed\_interval\ [(a, b)])\ f$

thm SET_VARIATION_MONOTONE:

$\forall (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow set\_variation\ t\ f \leq set\_variation\ s\ f$

thm SET_VARIATION_REFLECT2:

$\forall (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ set\_variation\ (IMAGE\ vector\_neg\ s)\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ f\ (IMAGE\ vector\_neg\ k)) = set\_variation\ s\ f$

thm HAS_BOUNDED_SETVARIATION_REFLECT2_EQ:

$\forall (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ f\ (IMAGE\ vector\_neg\ k))\ (IMAGE\ vector\_neg\ s) = has\_bounded\_setvariation\_on\ f\ s$

thm SET_VARIATION_TRANSLATION2:

$\forall (a::(real, ?'b::type)\ cart)\ (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ set\_variation\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s)\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ f\ (IMAGE\ (vector\_add\ a)\ k)) = set\_variation\ s\ f$

thm HAS_BOUNDED_SETVARIATION_TRANSLATION2_EQ:

$\forall (a::(real, ?'b::type)\ cart)\ (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ f\ (IMAGE\ (vector\_add\ a)\ k))\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s) = has\_bounded\_setvariation\_on\ f\ s$

thm HAS_BOUNDED_SETVARIATION_TRANSLATION:

$\forall (f::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ a::(real, ?'b::type)\ cart.\ has\_bounded\_setvariation\_on\ f\ s \longrightarrow has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ f\ (IMAGE\ (vector\_add\ a)\ k))\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s)$

thm DEF_absolutely_integrable_on:

$absolutely\_integrable\_on = (\lambda(\_1669605::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ \_1669606::(real, ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ \_1669605\ \_1669606 \wedge integrable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ lift\ (vector\_norm\ (\_1669605\ x)))\ \_1669606)$

thm absolutely_integrable_on:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\, f\, s = (integrable\_on\, f\, s \wedge integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\, x)))\ s)$

thm ABSOLUTELY_INTEGRABLE_IMP_INTEGRABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \longrightarrow integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_LE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\, f\, s \longrightarrow vector\_norm\ (integral\, s\, f) \le HOL\_Light\_Import.drop\ (integral\ s\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\, x))))$

thm ABSOLUTELY_INTEGRABLE_0:

$\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vec\ (0::nat))\ s$

thm ABSOLUTELY_INTEGRABLE_CMUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ c::real.\ absolutely\_integrable\_on\, f\, s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ c\ (f\, x))\ s$

thm ABSOLUTELY_INTEGRABLE_NEG:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\, f\, s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_neg\ (f\, x))\ s$

thm ABSOLUTELY_INTEGRABLE_NORM:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\, f\, s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\, x)))\ s$

thm ABSOLUTELY_INTEGRABLE_ABS_1:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ |HOL\_Light\_Import.drop\ (f\, x)|)\ s$

thm ABSOLUTELY_INTEGRABLE_ON_SUBINTERVAL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ absolutely\_integrable\_on\ f\, s \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ s \longrightarrow absolutely\_integrable\_on\, f\ (closed\_interval\ [(a,\ b)])$

thm ABSOLUTELY_INTEGRABLE_BOUNDED_SETVARIATION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \longrightarrow has\_bounded\_setvariation\_on\ (\lambda k::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integral\ k\ f)\ s$

thm lemma:

$\forall (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (g::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ (s::?'b::type \Rightarrow bool)\ e::real.\ sum\ s\ (\lambda x::?'b::type.\ vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x))) < e \longrightarrow FINITE\ s \longrightarrow |sum\ s\ (\lambda x::?'b::type.\ vector\_norm\ (f\ x)) - sum\ s\ (\lambda x::?'b::type.\ vector\_norm\ (g\ x))| < e$

thm BOUNDED_SETVARIATION_ABSOLUTELY_INTEGRABLE_INTERVAL:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, ?'b::type)\ cart)\ b::(real, ?'b::type)\ cart.\ integrable\_on\ f\ (closed\_interval\ [(a, b)]) \wedge has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ integral\ k\ f)\ (closed\_interval\ [(a, b)]) \longrightarrow absolutely\_integrable\_on\ f\ (closed\_interval\ [(a, b)])$

thm BOUNDED_SETVARIATION_ABSOLUTELY_INTEGRABLE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ integrable\_on\ f\ HOL\_Light\_Import.UNIV \wedge has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ integral\ k\ f)\ HOL\_Light\_Import.UNIV \longrightarrow absolutely\_integrable\_on\ f\ HOL\_Light\_Import.UNIV$

thm ABSOLUTELY_INTEGRABLE_BOUNDED_SETVARIATION_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, ?'b::type)\ cart)\ b::(real, ?'b::type)\ cart.\ absolutely\_integrable\_on\ f\ (closed\_interval\ [(a, b)]) = (integrable\_on\ f\ (closed\_interval\ [(a, b)]) \wedge has\_bounded\_setvariation\_on\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ integral\ k\ f)\ (closed\_interval\ [(a, b)]))$

thm ABSOLUTELY_INTEGRABLE_SET_VARIATION:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, ?'b::type)\ cart)\ b::(real, ?'b::type)\ cart.\ absolutely\_integrable\_on\ f\ (closed\_interval\ [(a, b)]) \longrightarrow set\_variation\ (closed\_interval\ [(a, b)])\ (\lambda k::(real, ?'b::type)\ cart \Rightarrow bool.\ integral\ k\ f) = HOL\_Light\_Import.drop\ (integral\ (closed\_interval\ [(a, b)])\ (\lambda x::(real, ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\ x))))$

thm ABSOLUTELY_INTEGRABLE_RESTRICT_UNIV:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ if\ IN\ x\ s\ then\ f\ x\ else\ vec\ (0::nat))\ HOL\_Light\_Import.UNIV = absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_CONST:

$\forall (a::(real, ?'b::type)\ cart)\ (b::(real, ?'b::type)\ cart)\ c::(real, ?'a::type)\ cart.\ absolutely\_integrable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ c)\ (closed\_interval\ [(a, b)])$

thm ABSOLUTELY_INTEGRABLE_ADD:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ vector\_add\ (f\ x)\ (g\ x))\ s$

thm ABSOLUTELY_INTEGRABLE_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm ABSOLUTELY_INTEGRABLE_LINEAR:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (h::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \wedge linear\ h \longrightarrow absolutely\_integrable\_on\ (h \circ f)\ s$

thm ABSOLUTELY_INTEGRABLE_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::?'c::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'c::type.\ IN\ a\ t \longrightarrow absolutely\_integrable\_on\ (f\ a)\ s) \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ t\ (\lambda a::?'c::type.\ f\ a\ x))\ s$

thm ABSOLUTELY_INTEGRABLE_ABS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ |\$\ (f\ x)\ i|))\ s$

thm ABSOLUTELY_INTEGRABLE_MAX:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ max\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm ABSOLUTELY_INTEGRABLE_MAX_1:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (f\ x))\ s \wedge absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (g\ x))\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (max\ (f\ x)\ (g\ x)))\ s$

thm ABSOLUTELY_INTEGRABLE_MIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ min\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm ABSOLUTELY_INTEGRABLE_MIN_1:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (f\ x))\ s \wedge absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (g\ x))\ s \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (min\ (f\ x)\ (g\ x)))\ s$

thm ABSOLUTELY_INTEGRABLE_ABS_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s = (integrable\_on\ f\ s \wedge integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ |\$\ (f\ x)\ i|))\ s)$

thm NONNEGATIVE_ABSOLUTELY_INTEGRABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (x::(real,\ ?'b::type)\ cart)\ i::nat.\ IN\ x\ s \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (0::real) \leq \$\ (f\ x)\ i) \wedge integrable\_on\ f\ s \longrightarrow absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_INTEGRABLE_BOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \wedge integrable\_on\ f\ s \wedge integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_ABSOLUTELY_INTEGRABLE_BOUND:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'c::type)\ cart.\ IN\ x\ s \longrightarrow vector\_norm\ (f\ x) \leq vector\_norm\ (g\ x)) \wedge integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_INF_1:

$\forall (fs::(real,\ ?'b::type)\ cart \Rightarrow ?'a::type \Rightarrow real)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge k \neq EMPTY \wedge (\forall i::?'a::type.\ IN\ i\ k \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (fs\ x\ i))\ s) \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (HOL\_Light\_Import.inf\ (IMAGE\ (fs\ x)\ k)))\ s$

thm ABSOLUTELY_INTEGRABLE_SUP_1:

$\forall (fs::(real,\ ?'b::type)\ cart \Rightarrow ?'a::type \Rightarrow real)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge k \neq EMPTY \wedge (\forall i::?'a::type.\ IN\ i\ k \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (fs\ x\ i))\ s) \longrightarrow absolutely\_integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (HOL\_Light\_Import.sup\ (IMAGE\ (fs\ x)\ k)))\ s$

thm ABSOLUTELY_INTEGRABLE_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ continuous\_on\ f\ (closed\_interval\ [(a,\ b)]) \longrightarrow absolutely\_integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_MIN_CONST_1:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::real.\ (0::real) \leq t \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \wedge integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (f\ x))\ s \longrightarrow integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (min\ (f\ x)\ t))\ s$

thm ABSOLUTELY_INTEGRABLE_ABSOLUTELY_INTEGRABLE_COMPONENT_UBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (x::(real,\ ?'b::type)\ cart)\ i::nat.\ IN\ x\ s \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (f\ x)\ i \leq \$\ (g\ x)\ i) \wedge integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_ABSOLUTELY_INTEGRABLE_COMPONENT_LBOUND:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ (\forall (x::(real,\ ?'b::type)\ cart)\ i::nat.\ IN\ x\ s \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (f\ x)\ i \leq \$\ (g\ x)\ i) \wedge absolutely\_integrable\_on\ f\ s \wedge integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ g\ s$

thm ABSOLUTELY_INTEGRABLE_ABSOLUTELY_INTEGRABLE_DROP_UBOUND:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \wedge integrable\_on\ f\ s \wedge absolutely\_integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_ABSOLUTELY_INTEGRABLE_DROP_LBOUND:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ unit)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (g\ x)) \wedge absolutely\_integrable\_on\ f\ s \wedge integrable\_on\ g\ s \longrightarrow absolutely\_integrable\_on\ g\ s$

thm HAS_INTEGRAL_COMPONENTWISE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ y::(real,\ ?'a::type)\ cart.\ has\_integral\ f\ y\ s = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow has\_integral\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (\$\ (f\ x)\ i))\ (lift\ (\$\ y\ i))\ s)$

thm INTEGRABLE_COMPONENTWISE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (\$\ (f\ x)\ i))\ s)$

thm LIFT_INTEGRAL_COMPONENT:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ integrable\_on\ f\ (?s::(real,\ ?'b::type)\ cart \Rightarrow bool) \longrightarrow lift\ (\$\ (integral\ ?s\ f)\ (?k::nat)) = integral\ ?s\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (\$\ (f\ x)\ ?k))$

thm INTEGRAL_COMPONENT:

$\forall f$::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *integrable_on* *f* (*?s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⟶ $ (*integral* *?s* *f*) (*?k*::*nat*) = *HOL_Light_Import.drop* (*integral* *?s* (λ*x*::(*real*, *?'b*::*type*) *cart*. *lift* ($ (*f* *x*) *?k*)))

thm ABSOLUTELY_INTEGRABLE_COMPONENTWISE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *absolutely_integrable_on* *f* *s* = (∀ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex* *HOL_Light_Import.UNIV* ⟶ *absolutely_integrable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *lift* ($ (*f* *x*) *i*)) *s*)

thm DOMINATED_CONVERGENCE:

∀ (*f*::*nat* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *unit*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. (∀ *k*::*nat*. *integrable_on* (*f* *k*) *s*) ∧ *integrable_on* *h* *s* ∧ (∀ (*k*::*nat*) *x*::(*real*, *?'b*::*type*) *cart*. *IN* *x* *s* ⟶ *vector_norm* (*f* *k* *x*) ≤ *HOL_Light_Import.drop* (*h* *x*)) ∧ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN* *x* *s* ⟶ *−−>* (λ*k*::*nat*. *f* *k* *x*) (*g* *x*) *sequentially*) ⟶ *integrable_on* *g* *s* ∧ *−−>* (λ*k*::*nat*. *integral* *s* (*f* *k*)) (*integral* *s* *g*) *sequentially*

thm DOMINATED_CONVERGENCE_INTEGRABLE:

∀ (*f*::*nat* ⇒ (*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*h*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *unit*) *cart*) *s*::(*real*, *?'c*::*type*) *cart* ⇒ *bool*. (∀ *k*::*nat*. *absolutely_integrable_on* (*f* *k*) *s*) ∧ *integrable_on* *h* *s* ∧ (∀ (*k*::*?'a*::*type*) *x*::(*real*, *?'c*::*type*) *cart*. *IN* *x* *s* ⟶ *vector_norm* (*g* *x*) ≤ *HOL_Light_Import.drop* (*h* *x*)) ∧ (∀ *x*::(*real*, *?'c*::*type*) *cart*. *IN* *x* *s* ⟶ *−−>* (λ*k*::*nat*. *f* *k* *x*) (*g* *x*) *sequentially*) ⟶ *integrable_on* *g* *s*

thm DOMINATED_CONVERGENCE_ABSOLUTELY_INTEGRABLE:

∀ (*f*::*nat* ⇒ (*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*h*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *unit*) *cart*) *s*::(*real*, *?'c*::*type*) *cart* ⇒ *bool*. (∀ *k*::*nat*. *absolutely_integrable_on* (*f* *k*) *s*) ∧ *integrable_on* *h* *s* ∧ (∀ (*k*::*?'a*::*type*) *x*::(*real*, *?'c*::*type*) *cart*. *IN* *x* *s* ⟶ *vector_norm* (*g* *x*) ≤ *HOL_Light_Import.drop* (*h* *x*)) ∧ (∀ *x*::(*real*, *?'c*::*type*) *cart*. *IN* *x* *s* ⟶ *−−>* (λ*k*::*nat*. *f* *k* *x*) (*g* *x*) *sequentially*) ⟶ *absolutely_integrable_on* *g* *s*

thm NEGLIGIBLE_ON_UNIV:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible* *s* = *has_integral* (*indicator* *s*) (*vec* (*0*::*nat*)) *HOL_Light_Import.UNIV*

thm NEGLIGIBLE_COUNTABLE_UNIONS:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *negligible* (*s* *n*)) ⟶ *negligible* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%1924*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC* *GEN%PVAR%1924* (*IN* *n* *HOL_Light_Import.UNIV*) (*s* *n*))))

thm HAS_INTEGRAL_NEGLIGIBLE_EQ:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀ (x::(real, ?'b::type) cart) i::nat. IN x s ∧ (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ (0::real) ≤ $ (f x) i) ⟶ has_integral f (vec (0::nat)) s = negligible (GSPEC (λGEN%PVAR%1933::(real, ?'b::type) cart. ∃ x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%1933 (IN x s ∧ f x ≠ vec (0::nat)) x))

thm NEGLIGIBLE_COUNTABLE:

∀ s::(real, ?'a::type) cart ⇒ bool. COUNTABLE s ⟶ negligible s

thm BEPPO_LEVI_INCREASING:

∀ (f::nat ⇒ (real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. (∀ k::nat. integrable_on (f k) s) ∧ (∀ (k::nat) x::(real, ?'a::type) cart. IN x s ⟶ HOL_Light_Import.drop (f k x) ≤ HOL_Light_Import.drop (f (Suc k) x)) ∧ bounded (GSPEC (λGEN%PVAR%1937::(real, unit) cart. ∃ k::nat. SETSPEC GEN%PVAR%1937 (IN k HOL_Light_Import.UNIV) (integral s (f k)))) ⟶ (∃ (g::(real, ?'a::type) cart ⇒ (real, unit) cart) k::(real, ?'a::type) cart ⇒ bool. negligible k ∧ (∀ x::(real, ?'a::type) cart. IN x (DIFF s k) ⟶ −−> (λk::nat. f k x) (g x) sequentially))

thm BEPPO_LEVI_DECREASING:

∀ (f::nat ⇒ (real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. (∀ k::nat. integrable_on (f k) s) ∧ (∀ (k::nat) x::(real, ?'a::type) cart. IN x s ⟶ HOL_Light_Import.drop (f (Suc k) x) ≤ HOL_Light_Import.drop (f k x)) ∧ bounded (GSPEC (λGEN%PVAR%1938::(real, unit) cart. ∃ k::nat. SETSPEC GEN%PVAR%1938 (IN k HOL_Light_Import.UNIV) (integral s (f k)))) ⟶ (∃ (g::(real, ?'a::type) cart ⇒ (real, unit) cart) k::(real, ?'a::type) cart ⇒ bool. negligible k ∧ (∀ x::(real, ?'a::type) cart. IN x (DIFF s k) ⟶ −−> (λk::nat. f k x) (g x) sequentially))

thm BEPPO_LEVI_MONOTONE_CONVERGENCE_INCREASING:

∀ (f::nat ⇒ (real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. (∀ k::nat. integrable_on (f k) s) ∧ (∀ (k::nat) x::(real, ?'a::type) cart. IN x s ⟶ HOL_Light_Import.drop (f k x) ≤ HOL_Light_Import.drop (f (Suc k) x)) ∧ bounded (GSPEC (λGEN%PVAR%1939::(real, unit) cart. ∃ k::nat. SETSPEC GEN%PVAR%1939 (IN k HOL_Light_Import.UNIV) (integral s (f k)))) ⟶ (∃ (g::(real, ?'a::type) cart ⇒ (real, unit) cart) k::(real, ?'a::type) cart ⇒ bool. negligible k ∧ (∀ x::(real, ?'a::type) cart. IN x (DIFF s k) ⟶ −−> (λk::nat. f k x) (g x) sequentially) ∧ integrable_on g s ∧ −−> (λk::nat. integral s (f k)) (integral s g) sequentially)

thm BEPPO_LEVI_MONOTONE_CONVERGENCE_DECREASING:

∀ (f::nat ⇒ (real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. (∀ k::nat. integrable_on (f k) s) ∧ (∀ (k::nat) x::(real, ?'a::type) cart. IN x s ⟶ HOL_Light_Import.drop (f (Suc k) x) ≤ HOL_Light_Import.drop (f k x)) ∧ bounded (GSPEC (λGEN%PVAR%1940::(real, unit) cart. ∃ k::nat.

SETSPEC GEN%PVAR%1940 (IN k HOL_Light_Import.UNIV) (integral s (f k)))) ⟶ (∃ (g::(real, ?'a::type) cart ⇒ (real, unit) cart) k::(real, ?'a::type) cart ⇒ bool. negligible k ∧ (∀ x::(real, ?'a::type) cart. IN x (DIFF s k) ⟶ --> (λk::nat. f k x) (g x) sequentially) ∧ integrable_on g s ∧ --> (λk::nat. integral s (f k)) (integral s g) sequentially)

thm FUNDAMENTAL_THEOREM_OF_CALCULUS_STRONG:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (f'::(real, unit) cart ⇒ (real, ?'a::type) cart) (s::(real, unit) cart ⇒ bool) (a::(real, unit) cart) b::(real, unit) cart. COUNTABLE s ∧ HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b ∧ continuous_on f (closed_interval [(a, b)]) ∧ (∀ x::(real, unit) cart. IN x (DIFF (closed_interval [(a, b)]) s) ⟶ has_vector_derivative f (f' x) (within (at x) (closed_interval [(a, b)]))) ⟶ has_integral f' (vector_sub (f b) (f a)) (closed_interval [(a, b)])

thm FUNDAMENTAL_THEOREM_OF_CALCULUS_INTERIOR_STRONG:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (f'::(real, unit) cart ⇒ (real, ?'a::type) cart) (s::(real, unit) cart ⇒ bool) (a::(real, unit) cart) b::(real, unit) cart. COUNTABLE s ∧ HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b ∧ continuous_on f (closed_interval [(a, b)]) ∧ (∀ x::(real, unit) cart. IN x (DIFF (open_interval (a, b)) s) ⟶ has_vector_derivative f (f' x) (at x)) ⟶ has_integral f' (vector_sub (f b) (f a)) (closed_interval [(a, b)])

thm FUNDAMENTAL_THEOREM_OF_CALCULUS:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (f'::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b ∧ (∀ x::(real, unit) cart. IN x (closed_interval [(a, b)]) ⟶ has_vector_derivative f (f' x) (within (at x) (closed_interval [(a, b)]))) ⟶ has_integral f' (vector_sub (f b) (f a)) (closed_interval [(a, b)])

thm FUNDAMENTAL_THEOREM_OF_CALCULUS_INTERIOR:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (f'::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b ∧ continuous_on f (closed_interval [(a, b)]) ∧ (∀ x::(real, unit) cart. IN x (open_interval (a, b)) ⟶ has_vector_derivative f (f' x) (at x)) ⟶ has_integral f' (vector_sub (f b) (f a)) (closed_interval [(a, b)])

thm ANTIDERIVATIVE_INTEGRAL_CONTINUOUS:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) b::(real, unit) cart. continuous_on f (closed_interval [(a, b)]) ⟶ (∃ g::(real, unit) cart ⇒ (real, ?'a::type) cart. ∀ (u::(real, unit) cart) v::(real, unit) cart. IN u (closed_interval [(a, b)]) ∧ IN v (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop u ≤ HOL_Light_Import.drop v ⟶ has_integral f (vector_sub (g v) (g u)) (closed_interval [(u, v)]))

thm HAS_DERIVATIVE_ZERO_UNIQUE_STRONG_INTERVAL:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) (b::(real, unit) cart) (k::(real, unit) cart ⇒ bool) y::(real, ?'a::type) cart. COUNTABLE k ∧ continuous_on f (closed_interval [(a, b)]) ∧ f a = y ∧ (∀ x::(real, unit) cart. IN x (DIFF (closed_interval [(a, b)]) k) ⟶ has_derivative f (λh::(real, unit) cart. vec (0::nat)) (within (at x) (closed_interval [(a, b)]))) ⟶ (∀ x::(real, unit) cart. IN x (closed_interval [(a, b)]) ⟶ f x = y)

thm HAS_DERIVATIVE_ZERO_UNIQUE_STRONG_CONVEX:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (k::(real, ?'b::type) cart ⇒ bool) (c::(real, ?'b::type) cart) y::(real, ?'a::type) cart. convex s ∧ COUNTABLE k ∧ continuous_on f s ∧ IN c s ∧ f c = y ∧ (∀ x::(real, ?'b::type) cart. IN x (DIFF s k) ⟶ has_derivative f (λh::(real, ?'b::type) cart. vec (0::nat)) (within (at x) s)) ⟶ (∀ x::(real, ?'b::type) cart. IN x s ⟶ f x = y)

thm HAS_DERIVATIVE_ZERO_UNIQUE_STRONG_CONNECTED:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) (k::(real, ?'b::type) cart ⇒ bool) (c::(real, ?'b::type) cart) y::(real, ?'a::type) cart. connected s ∧ HOL_Light_Import.open s ∧ COUNTABLE k ∧ continuous_on f s ∧ IN c s ∧ f c = y ∧ (∀ x::(real, ?'b::type) cart. IN x (DIFF s k) ⟶ has_derivative f (λh::(real, ?'b::type) cart. vec (0::nat)) (within (at x) s)) ⟶ (∀ x::(real, ?'b::type) cart. IN x s ⟶ f x = y)

thm DEF_equiintegrable_on:

equiintegrable_on = (λ(_1715589::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) _1715590::(real, ?'b::type) cart ⇒ bool. (∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. IN f _1715589 ⟶ integrable_on f _1715590) ∧ (∀ e>0::real. ∃ d::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart ⇒ bool. gauge d ∧ (∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) p::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ bool. IN f _1715589 ∧ tagged_division_of p _1715590 ∧ fine d p ⟶ vector_norm (vector_sub (vsum p (GABS (λfa::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (fa (x, k)) (% (content k) (f x))))) (integral _1715590 f)) < e)))

thm equiintegrable_on:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) i::(real, ?'b::type) cart ⇒ bool. equiintegrable_on fs i = ((∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. IN f fs ⟶ integrable_on f i) ∧ (∀ e>0::real. ∃ d::(real, ?'b::type) cart ⇒ (real, ?'b::type) cart ⇒ bool. gauge d ∧ (∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) p::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ bool. IN f fs ∧ tagged_division_of p i ∧ fine d p ⟶ vector_norm (vector_sub (vsum p (GABS (λfa::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ (real, ?'a::type) cart. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (fa (x, k)) (% (content k) (f x))))) (integral i f)) < e)))

thm EQUIINTEGRABLE_ON_SING:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ equiintegrable\_on\ (INSERT\ f\ EMPTY)\ (closed\_interval\ [(a,\ b)]) = integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm EQUIINTEGRABLE_ON_NULL:

$\forall (fs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ content\ (closed\_interval\ [(a,\ b)]) = (0::real) \longrightarrow equiintegrable\_on\ fs\ (closed\_interval\ [(a,\ b)])$

thm EQUIINTEGRABLE_ON_SPLIT:

$\forall (fs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (k::nat)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ c::real.\ equiintegrable\_on\ fs\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1972::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1972\ (\$\ x\ k \le c)\ x))) \land equiintegrable\_on\ fs\ (HOL\_Light\_Import.INTER\ (closed\_interval\ [(a,\ b)])\ (GSPEC\ (\lambda GEN\%PVAR\%1973::(real,\ ?'b::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1973\ (c \le \$\ x\ k)\ x))) \land (1::nat) \le k \land k \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow equiintegrable\_on\ fs\ (closed\_interval\ [(a,\ b)])$

thm EQUIINTEGRABLE_DIVISION:

$\forall (fs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (d::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ division\_of\ d\ (closed\_interval\ [(a,\ b)]) \longrightarrow equiintegrable\_on\ fs\ (closed\_interval\ [(a,\ b)]) = (\forall i::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ i\ d \longrightarrow equiintegrable\_on\ fs\ i)$

thm EQUIINTEGRABLE_LIMIT:

$\forall (f::nat \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ equiintegrable\_on\ (GSPEC\ (\lambda GEN\%PVAR\%1974::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1974\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (f\ n)))\ (closed\_interval\ [(a,\ b)]) \land (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow --> (\lambda n::nat.\ f\ n\ x)\ (g\ x)\ sequentially) \longrightarrow integrable\_on\ g\ (closed\_interval\ [(a,\ b)]) \land --> (\lambda n::nat.\ integral\ (closed\_interval\ [(a,\ b)])\ (f\ n))\ (integral\ (closed\_interval\ [(a,\ b)])\ g)\ sequentially$

thm EQUIINTEGRABLE_SUBSET:

$\forall (fs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (gs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ equiintegrable\_on\ fs\ s \land SUBSET\ gs\ fs \longrightarrow equiintegrable\_on\ gs\ s$

thm EQUIINTEGRABLE_UNION:

$\forall (fs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ (gs::((real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ equiintegrable\_on\ fs\ s \land equiintegrable\_on\ gs\ s \longrightarrow equiintegrable\_on\ (HOL\_Light\_Import.UNION\ fs\ gs)\ s$

thm EQUIINTEGRABLE_EQ:

$\forall$ (*fs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) (*gs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *equiintegrable_on* *fs s* $\wedge$ ($\forall$ *g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *IN g gs* $\longrightarrow$ ($\exists$ *f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *IN f fs* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN* *x s* $\longrightarrow$ *f x* = *g x*))) $\longrightarrow$ *equiintegrable_on gs s*

thm EQUIINTEGRABLE_CMUL:

$\forall$ (*fs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *k*::*real*. *equiintegrable_on fs s* $\longrightarrow$ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1975*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. $\exists$ (*c*::*real*) *f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1975* ($|c| \leq k$ $\wedge$ *IN f fs*) (λ*x*::(*real*, *?'b*::*type*) *cart*. % *c* (*f x*)))) *s*

thm EQUIINTEGRABLE_ADD:

$\forall$ (*fs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) (*gs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *equiintegrable_on* *fs s* $\wedge$ *equiintegrable_on gs s* $\longrightarrow$ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1976*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. $\exists$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1976* (*IN f fs* $\wedge$ *IN g gs*) (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_add* (*f x*) (*g x*)))) *s*

thm EQUIINTEGRABLE_NEG:

$\forall$ (*fs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *equiintegrable_on fs s* $\longrightarrow$ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1977*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. $\exists$ *f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1977* (*IN f fs*) (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_neg* (*f x*)))) *s*

thm EQUIINTEGRABLE_SUB:

$\forall$ (*fs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) (*gs*::((*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) $\Rightarrow$ *bool*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *equiintegrable_on* *fs s* $\wedge$ *equiintegrable_on gs s* $\longrightarrow$ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1978*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. $\exists$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *g*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%1978* (*IN f fs* $\wedge$ *IN g gs*) (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_sub* (*f x*) (*g x*)))) *s*

thm EQUIINTEGRABLE_SUM:

$\forall$ (*fs*::((*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*) $\Rightarrow$ *bool*) (*a*::(*real*, *?'c*::*type*) *cart*) *b*::(*real*, *?'c*::*type*) *cart*. *equiintegrable_on fs* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1979*::(*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*. $\exists$ (*t*::*?'a*::*type* $\Rightarrow$ *bool*) (*c*::*?'a*::*type* $\Rightarrow$ *real*) *f*::*?'a*::*type* $\Rightarrow$ (*real*, *?'c*::*type*) *cart* $\Rightarrow$ (*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%1979* (*FINITE t* $\wedge$ ($\forall$ *i*::*?'a*::*type*. *IN i t* $\longrightarrow$ (*0*::*real*) $\leq$ *c i* $\wedge$ *IN* (*f i*) *fs*) $\wedge$ *sum t c*

= (*1*::*real*)) (λ*x*::(*real*, *?'c*::*type*) *cart*. *vsum t* (λ*i*::*?'a*::*type*. % (*c i*) (*f i x*))))) (*closed_interval* [(*a*, *b*)])

thm EQUIINTEGRABLE_UNIFORM_LIMIT:

∀ (*fs*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *equiintegrable_on fs* (*closed_interval* [(*a*, *b*)]) ⟶ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1980*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∃ *g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1980* (∀ *e*>*0*::*real*. ∃ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *IN f fs* ∧ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ *vector_norm* (*vector_sub* (*g x*) (*f x*)) < *e*)) *g*)) (*closed_interval* [(*a*, *b*)])

thm EQUIINTEGRABLE_REFLECT:

∀ (*fs*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *equiintegrable_on fs* (*closed_interval* [(*a*, *b*)]) ⟶ *equiintegrable_on* (*GSPEC* (λ*GEN%PVAR%1981*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∃ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *SET-SPEC GEN%PVAR%1981* (*IN f fs*) (λ*x*::(*real*, *?'b*::*type*) *cart*. *f* (*vector_neg x*)))) (*closed_interval* [(*vector_neg b*, *vector_neg a*)])

thm SUM_CONTENT_AREA_OVER_THIN_DIVISION:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*i*::*nat*) *c*::*real*. *division_of d s* ∧ *SUBSET s* (*closed_interval* [(*a*, *b*)]) ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ $ *a i* ≤ *c* ∧ *c* ≤ $ *b i* ∧ (∀ *k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN k d* ⟶ *HOL_Light_Import.INTER k* (*GSPEC* (λ*GEN%PVAR%2026*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2026* ($ *x i* = *c*) *x*)) ≠ *EMPTY*) ⟶ ($ *b i* − $ *a i*) ∗ *sum d* (λ*k*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *content k* / ($ (*interval_upperbound k*) *i* − $ (*interval_lowerbound k*) *i*)) ≤ *real_of_nat* (*2*::*nat*) ∗ *content* (*closed_interval* [(*a*, *b*)])

thm BOUNDED_EQUIINTEGRAL_OVER_THIN_TAGGED_PARTIAL_DIVISION:

∀ (*fs*::((*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) ⇒ *bool*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *e*::*real*. *equiintegrable_on fs* (*closed_interval* [(*a*, *b*)]) ∧ *IN f fs* ∧ (∀ (*h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'b*::*type*) *cart*. *IN h fs* ∧ *IN x* (*closed_interval* [(*a*, *b*)]) ⟶ *vector_norm* (*h x*) ≤ *vector_norm* (*f x*)) ∧ (*0*::*real*) < *e* ⟶ (∃ *d*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart* ⇒ *bool*. *gauge d* ∧ (∀ (*c*::(*real*, *?'b*::*type*) *cart*) (*i*::*nat*) (*p*::(*real*, *?'b*::*type*) *cart* × ((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *IN c* (*closed_interval* [(*a*, *b*)]) ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *tagged_partial_division_of p* (*closed_interval* [(*a*, *b*)]) ∧ *fine d p* ∧ *IN h fs* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *k*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *IN* (*x*, *k*) *p* ⟶ *HOL_Light_Import.INTER k* (*GSPEC* (λ*GEN%PVAR%2027*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2027* ($ *x i*

= $ c i) x)) ≠ EMPTY) ⟶ sum p (GABS (λf::(real, ?'b::type) cart × ((real, ?'b::type) cart ⇒ bool) ⇒ real. ∀ (x::(real, ?'b::type) cart) k::(real, ?'b::type) cart ⇒ bool. GEQ (f (x, k)) (vector_norm (integral k h)))) < e))

thm EQUIINTEGRABLE_HALFSPACE_RESTRICTIONS_LE:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. equiintegrable_on fs (closed_interval [(a, b)]) ∧ IN f fs ∧ (∀ (h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x::(real, ?'b::type) cart. IN h fs ∧ IN x (closed_interval [(a, b)]) ⟶ vector_norm (h x) ≤ vector_norm (f x)) ⟶ equiintegrable_on (GSPEC (λGEN%PVAR%2044::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. ∃ (i::nat) (c::real) h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. SETSPEC GEN%PVAR%2044 (IN i (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) ∧ IN c HOL_Light_Import.UNIV ∧ IN h fs) (λx::(real, ?'b::type) cart. if $ x i ≤ c then h x else vec (0::nat)))) (closed_interval [(a, b)])

thm EQUIINTEGRABLE_HALFSPACE_RESTRICTIONS_GE:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. equiintegrable_on fs (closed_interval [(a, b)]) ∧ IN f fs ∧ (∀ (h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x::(real, ?'b::type) cart. IN h fs ∧ IN x (closed_interval [(a, b)]) ⟶ vector_norm (h x) ≤ vector_norm (f x)) ⟶ equiintegrable_on (GSPEC (λGEN%PVAR%2046::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. ∃ (i::nat) (c::real) h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. SETSPEC GEN%PVAR%2046 (IN i (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) ∧ IN c HOL_Light_Import.UNIV ∧ IN h fs) (λx::(real, ?'b::type) cart. if c ≤ $ x i then h x else vec (0::nat)))) (closed_interval [(a, b)])

thm EQUIINTEGRABLE_HALFSPACE_RESTRICTIONS_LT:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. equiintegrable_on fs (closed_interval [(a, b)]) ∧ IN f fs ∧ (∀ (h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x::(real, ?'b::type) cart. IN h fs ∧ IN x (closed_interval [(a, b)]) ⟶ vector_norm (h x) ≤ vector_norm (f x)) ⟶ equiintegrable_on (GSPEC (λGEN%PVAR%2047::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. ∃ (i::nat) (c::real) h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. SETSPEC GEN%PVAR%2047 (IN i (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV)) ∧ IN c HOL_Light_Import.UNIV ∧ IN h fs) (λx::(real, ?'b::type) cart. if $ x i < c then h x else vec (0::nat)))) (closed_interval [(a, b)])

thm EQUIINTEGRABLE_HALFSPACE_RESTRICTIONS_GT:

∀ (fs::((real, ?'b::type) cart ⇒ (real, ?'a::type) cart) ⇒ bool) (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. equiintegrable_on fs (closed_interval [(a, b)]) ∧ IN f fs ∧ (∀ (h::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x::(real, ?'b::type) cart. IN h fs ∧ IN x (closed_interval [(a, b)]) ⟶ vector_norm (h x) ≤ vector_norm (f x)) ⟶ equiintegrable_on

(*GSPEC* (*λGEN%PVAR%2048*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∃(*i*::*nat*) (*c*::*real*) *h*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%2048* (*IN i* (*dotdot* (*1*::*nat*) (*dimindex HOL_Light_Import.UNIV*)) ∧ *IN c HOL_Light_Import.UNIV* ∧ *IN h fs*) (*λx*::(*real*, *?'b*::*type*) *cart*. *if c* < $ *x i then h x else vec* (*0*::*nat*)))) (*closed_interval* [(*a*, *b*)])

thm EQUIINTEGRABLE_OPEN_INTERVAL_RESTRICTIONS:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ⟶ *equiintegrable_on* (*GSPEC* (*λGEN%PVAR%2051*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∃(*c*::(*real*, *?'b*::*type*) *cart*) *d*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2051* (*IN c HOL_Light_Import.UNIV* ∧ *IN d HOL_Light_Import.UNIV*) (*λx*::(*real*, *?'b*::*type*) *cart*. *if IN x* (*open_interval* (*c*, *d*)) *then f x else vec* (*0*::*nat*)))) (*closed_interval* [(*a*, *b*)])

thm EQUIINTEGRABLE_CLOSED_INTERVAL_RESTRICTIONS:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ⟶ *equiintegrable_on* (*GSPEC* (*λGEN%PVAR%2054*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. ∃(*c*::(*real*, *?'b*::*type*) *cart*) *d*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2054* (*IN c HOL_Light_Import.UNIV* ∧ *IN d HOL_Light_Import.UNIV*) (*λx*::(*real*, *?'b*::*type*) *cart*. *if IN x* (*closed_interval* [(*c*, *d*)]) *then f x else vec* (*0*::*nat*)))) (*closed_interval* [(*a*, *b*)])

thm INDEFINITE_INTEGRAL_CONTINUOUS:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) (*c*::(*real*, *?'b*::*type*) *cart*) (*d*::(*real*, *?'b*::*type*) *cart*) *e*::*real*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ∧ *IN c* (*closed_interval* [(*a*, *b*)]) ∧ *IN d* (*closed_interval* [(*a*, *b*)]) ∧ (*0*::*real*) < *e* ⟶ (∃*k*>*0*::*real*. ∀(*c'*::(*real*, *?'b*::*type*) *cart*) *d'*::(*real*, *?'b*::*type*) *cart*. *IN c'* (*closed_interval* [(*a*, *b*)]) ∧ *IN d'* (*closed_interval* [(*a*, *b*)]) ∧ *vector_norm* (*vector_sub c' c*) ≤ *k* ∧ *vector_norm* (*vector_sub d' d*) ≤ *k* ⟶ *vector_norm* (*vector_sub* (*integral* (*closed_interval* [(*c'*, *d'*)]) *f*) (*integral* (*closed_interval* [(*c*, *d*)]) *f*)) < *e*)

thm INDEFINITE_INTEGRAL_CONTINUOUS_RIGHT:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ⟶ *continuous_on* (*λx*::(*real*, *?'b*::*type*) *cart*. *integral* (*closed_interval* [(*a*, *x*)]) *f*) (*closed_interval* [(*a*, *b*)])

thm INDEFINITE_INTEGRAL_CONTINUOUS_LEFT:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *integrable_on f* (*closed_interval* [(*a*, *b*)]) ⟶ *continuous_on* (*λx*::(*real*, *?'b*::*type*) *cart*. *integral* (*closed_interval* [(*x*, *b*)]) *f*) (*closed_interval* [(*a*, *b*)])

thm INDEFINITE_INTEGRAL_UNIFORMLY_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)$ $b::(real,\ ?'b::type)\ cart.\ integrable\_on\, f\ (closed\_interval\ [(a,\ b)]) \longrightarrow uniformly\_continuous\_on$ $(\lambda y::(real,\ (?'b::type,\ ?'b::type)\ finite\_sum)\ cart.\ integral\ (closed\_interval\ [(fstcart$ $y,\ sndcart\ y)])\ f)\ (closed\_interval\ [(pastecart\ a\ a,\ pastecart\ b\ b)])$

thm INDEFINITE_INTEGRAL_UNIFORMLY_CONTINUOUS_EXPLICIT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)$ $(b::(real,\ ?'b::type)\ cart)\ e::real.\ integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge$ $(0::real) < e \longrightarrow (\exists k>0::real.\ \forall (c::(real,\ ?'b::type)\ cart)\ (d::(real,\ ?'b::type)$ $cart)\ (c'::(real,\ ?'b::type)\ cart)\ d'::(real,\ ?'b::type)\ cart.\ IN\ c\ (closed\_interval$ $[(a,\ b)]) \wedge IN\ d\ (closed\_interval\ [(a,\ b)]) \wedge IN\ c'\ (closed\_interval\ [(a,\ b)]) \wedge IN$ $d'\ (closed\_interval\ [(a,\ b)]) \wedge vector\_norm\ (vector\_sub\ c'\ c) \leq k \wedge vector\_norm$ $(vector\_sub\ d'\ d) \leq k \longrightarrow vector\_norm\ (vector\_sub\ (integral\ (closed\_interval$ $[(c',\ d')])\ f)\ (integral\ (closed\_interval\ [(c,\ d)])\ f)) < e)$

thm SECOND_MEAN_VALUE_THEOREM_FULL:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,$ $unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge integrable\_on$ $f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x$ $(closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop$ $x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN$ $c\ (closed\_interval\ [(a,\ b)]) \wedge has\_integral\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))$ $(vector\_add\ (\%\ (g\ a)\ (integral\ (closed\_interval\ [(a,\ c)])\ f))\ (\%\ (g\ b)\ (integral$ $(closed\_interval\ [(c,\ b)])\ f)))\ (closed\_interval\ [(a,\ b)]))$

thm SECOND_MEAN_VALUE_THEOREM:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,$ $unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge integrable\_on$ $f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x$ $(closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop$ $x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN$ $c\ (closed\_interval\ [(a,\ b)]) \wedge integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ unit)$ $cart.\ \%\ (g\ x)\ (f\ x)) = vector\_add\ (\%\ (g\ a)\ (integral\ (closed\_interval\ [(a,\ c)])$ $f))\ (\%\ (g\ b)\ (integral\ (closed\_interval\ [(c,\ b)])\ f)))$

thm SECOND_MEAN_VALUE_THEOREM_GEN_FULL:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,$ $unit)\ cart)\ (b::(real,\ unit)\ cart)\ (u::real)\ v::real.\ closed\_interval\ [(a,\ b)] \neq$ $EMPTY \wedge integrable\_on\, f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.$ $IN\ x\ (open\_interval\ (a,\ b)) \longrightarrow u \leq g\ x \wedge g\ x \leq v) \wedge (\forall (x::(real,\ unit)$ $cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval$ $[(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq$ $g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN\ c\ (closed\_interval\ [(a,\ b)]) \wedge has\_integral$ $(\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ (vector\_add\ (\%\ u\ (integral\ (closed\_interval$ $[(a,\ c)])\ f))\ (\%\ v\ (integral\ (closed\_interval\ [(c,\ b)])\ f)))\ (closed\_interval\ [(a,$ $b)]))$

thm SECOND_MEAN_VALUE_THEOREM_GEN:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ (u::real)\ v::real.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (open\_interval\ (a,\ b)) \longrightarrow u \leq g\ x \wedge g\ x \leq v) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN\ c\ (closed\_interval\ [(a,\ b)]) \wedge integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x)) = vector\_add\ (\%\ u\ (integral\ (closed\_interval\ [(a,\ c)])\ f))\ (\%\ v\ (integral\ (closed\_interval\ [(c,\ b)])\ f)))$

thm SECOND_MEAN_VALUE_THEOREM_BONNET_FULL:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow (0::real) \leq g\ x) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN\ c\ (closed\_interval\ [(a,\ b)]) \wedge has\_integral\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ (\%\ (g\ b)\ (integral\ (closed\_interval\ [(c,\ b)])\ f))\ (closed\_interval\ [(a,\ b)]))$

thm SECOND_MEAN_VALUE_THEOREM_BONNET:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \longrightarrow (0::real) \leq g\ x) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::(real,\ unit)\ cart.\ IN\ c\ (closed\_interval\ [(a,\ b)]) \wedge integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x)) = \%\ (g\ b)\ (integral\ (closed\_interval\ [(c,\ b)])\ f))$

thm INTEGRABLE_INCREASING_PRODUCT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \longrightarrow integrable\_on\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_INCREASING_PRODUCT_UNIV:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ B::real.\ integrable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ x \leq g\ y) \wedge (\forall x::(real,\ unit)\ cart.\ |g\ x| \leq B) \longrightarrow integrable\_on\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ HOL\_Light\_Import.UNIV$

thm INTEGRABLE_INCREASING:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ (y::(real,\ unit)\ cart)\ i::nat.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (f\ x)\ i \leq \$\ (f\ y)\ i) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_INCREASING_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop\ (f\ x) \leq HOL\_Light\_Import.drop\ (f\ y)) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_DECREASING_PRODUCT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ integrable\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ y \leq g\ x) \longrightarrow integrable\_on\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_DECREASING_PRODUCT_UNIV:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow real)\ B::real.\ integrable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow g\ y \leq g\ x) \wedge (\forall x::(real,\ unit)\ cart.\ |g\ x| \leq B) \longrightarrow integrable\_on\ (\lambda x::(real,\ unit)\ cart.\ \%\ (g\ x)\ (f\ x))\ HOL\_Light\_Import.UNIV$

thm INTEGRABLE_DECREASING:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ (y::(real,\ unit)\ cart)\ i::nat.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ (f\ y)\ i \leq \$\ (f\ x)\ i) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_DECREASING_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop\ (f\ y) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm DEF_has_bounded_variation_on:

$has\_bounded\_variation\_on = (\lambda\_1775666::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ has\_bounded\_setvariation\_on\ (\lambda k::(real,\ unit)\ cart \Rightarrow bool.\ vector\_sub\ (\_1775666\ (interval\_upperbound\ k))\ (\_1775666\ (interval\_lowerbound\ k))))$

thm has_bounded_variation_on:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s = has\_bounded\_setvariation\_on\ (\lambda k::(real,\ unit)\ cart \Rightarrow bool.\ vector\_sub\ (f\ (interval\_upperbound\ k))\ (f\ (interval\_lowerbound\ k)))\ s$

thm DEF_vector_variation:

$vector\_variation = (\lambda(\_1775678::(real,\ unit)\ cart \Rightarrow bool)\ \_1775679::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ set\_variation\ \_1775678\ (\lambda k::(real,\ unit)\ cart \Rightarrow bool.\ vector\_sub\ (\_1775679\ (interval\_upperbound\ k))\ (\_1775679\ (interval\_lowerbound\ k))))$

thm vector_variation:

$\forall (s::(real,\ unit)\ cart \Rightarrow bool)\ f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ vector\_variation\ s\ f = set\_variation\ s\ (\lambda k::(real,\ unit)\ cart \Rightarrow bool.\ vector\_sub\ (f\ (interval\_upperbound\ k))\ (f\ (interval\_lowerbound\ k)))$

thm HAS_BOUNDED_VARIATION_ON_EQ:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ (\forall x::(real,\ unit)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge has\_bounded\_variation\_on\ f\ s \longrightarrow has\_bounded\_variation\_on\ g\ s$

thm VECTOR_VARIATION_EQ:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ (\forall x::(real,\ unit)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow vector\_variation\ s\ f = vector\_variation\ s\ g$

thm HAS_BOUNDED_VARIATION_ON_COMPONENTWISE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real,\ unit)\ cart.\ lift\ (\$\ (f\ x)\ i))\ s)$

thm VARIATION_EQUAL_LEMMA:

$\forall (ms::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ ms'::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart.\ (\forall s::(real,\ unit)\ cart.\ ms'\ (ms\ s) = s \wedge ms\ (ms'\ s) = s) \wedge (\forall (d::((real,\ unit)\ cart \Rightarrow bool) \Rightarrow bool)\ t::(real,\ unit)\ cart \Rightarrow bool.\ division\_of\ d\ t \longrightarrow division\_of\ (IMAGE\ (IMAGE\ ms)\ d)\ (IMAGE\ ms\ t) \wedge division\_of\ (IMAGE\ (IMAGE\ ms')\ d)\ (IMAGE\ ms'\ t)) \wedge (\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow IMAGE\ ms'\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(ms'\ a,\ ms'\ b)] \vee IMAGE\ ms'\ (closed\_interval\ [(a,\ b)]) = closed\_interval\ [(ms'\ b,\ ms'\ a)]) \longrightarrow (\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real,\ unit)\ cart.\ f\ (ms'\ x))\ (IMAGE\ ms\ s) = has\_bounded\_variation\_on\ f\ s) \wedge (\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ vector\_variation\ (IMAGE\ ms\ s)\ (\lambda x::(real,\ unit)\ cart.\ f\ (ms'\ x)) = vector\_variation\ s\ f)$

thm HAS_BOUNDED_VARIATION_ON_SUBSET:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, unit)\ cart \Rightarrow bool)\ t::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow has\_bounded\_variation\_on\ f\ t$

thm HAS_BOUNDED_VARIATION_ON_CONST:

$\forall (s::(real, unit)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart.\ has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ c)\ s$

thm VECTOR_VARIATION_CONST:

$\forall (s::(real, unit)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart.\ vector\_variation\ s\ (\lambda x::(real, unit)\ cart.\ c) = (0::real)$

thm HAS_BOUNDED_VARIATION_ON_CMUL:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (c::real)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ \%\ c\ (f\ x))\ s$

thm HAS_BOUNDED_VARIATION_ON_NEG:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ vector\_neg\ (f\ x))\ s$

thm HAS_BOUNDED_VARIATION_ON_ADD:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge has\_bounded\_variation\_on\ g\ s \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ vector\_add\ (f\ x)\ (g\ x))\ s$

thm HAS_BOUNDED_VARIATION_ON_SUB:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge has\_bounded\_variation\_on\ g\ s \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm HAS_BOUNDED_VARIATION_ON_COMPOSE_LINEAR:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'b::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge linear\ g \longrightarrow has\_bounded\_variation\_on\ (g \circ f)\ s$

thm HAS_BOUNDED_VARIATION_ON_NULL:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ content\ s = (0::real) \wedge bounded\ s \longrightarrow has\_bounded\_variation\_on\ f\ s$

thm HAS_BOUNDED_VARIATION_ON_EMPTY:

$\forall f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ has\_bounded\_variation\_on\ f\ EMPTY$

thm HAS_BOUNDED_VARIATION_ON_NORM:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_bounded_variation_on f s* $\longrightarrow$ *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. *lift* (*vector_norm* (*f x*))) *s*

thm HAS_BOUNDED_VARIATION_ON_MAX:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_bounded_variation_on f s* $\wedge$ *has_bounded_variation_on g s* $\longrightarrow$ *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. *lift* (*max* (*HOL_Light_Import.drop* (*f x*)) (*HOL_Light_Import.drop* (*g x*)))) *s*

thm HAS_BOUNDED_VARIATION_ON_MIN:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_bounded_variation_on f s* $\wedge$ *has_bounded_variation_on g s* $\longrightarrow$ *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. *lift* (*min* (*HOL_Light_Import.drop* (*f x*)) (*HOL_Light_Import.drop* (*g x*)))) *s*

thm HAS_BOUNDED_VARIATION_ON_IMP_BOUNDED_ON_SUBINTERVALS:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_bounded_variation_on f s* $\longrightarrow$ *bounded* (*GSPEC* (λ*GEN%PVAR%2077*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*d*::(*real*, *unit*) *cart*) *c*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%2077* (*SUBSET* (*closed_interval* [(*c*, *d*)]) *s* $\wedge$ *closed_interval* [(*c*, *d*)] $\neq$ *EMPTY*) (*vector_sub* (*f d*) (*f c*))))

thm HAS_BOUNDED_VARIATION_ON_IMP_BOUNDED_ON_INTERVAL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *bounded* (*IMAGE f* (*closed_interval* [(*a*, *b*)]))

thm HAS_BOUNDED_VARIATION_ON_MUL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f* (*closed_interval* [(*a*, *b*)]) $\wedge$ *has_bounded_variation_on g* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. % (*HOL_Light_Import.drop* (*f x*)) (*g x*)) (*closed_interval* [(*a*, *b*)])

thm VECTOR_VARIATION_POS_LE:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*. *has_bounded_variation_on f s* $\longrightarrow$ (*0*::*real*) $\leq$ *vector_variation s f*

thm VECTOR_VARIATION_GE_NORM_FUNCTION:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f s* $\wedge$ *SUBSET* (*closed_interval* [(*a*, *b*)]) *s* $\wedge$ *closed_interval* [(*a*, *b*)] $\neq$ *EMPTY* $\longrightarrow$ *vector_norm* (*vector_sub* (*f b*) (*f a*)) $\leq$ *vector_variation s f*

thm VECTOR_VARIATION_GE_DROP_FUNCTION:

$\forall (f::(real, unit)\ cart \Rightarrow (real, unit)\ cart)\ (s::(real, unit)\ cart \Rightarrow bool)\ (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ has\_bounded\_variation\_on\ f\ s \wedge SUBSET\ (closed\_interval\ [(a,\ b)])\ s \wedge closed\_interval\ [(a,\ b)] \neq EMPTY \longrightarrow HOL\_Light\_Import.drop\ (f\ b) - HOL\_Light\_Import.drop\ (f\ a) \leq vector\_variation\ s\ f$

thm VECTOR_VARIATION_MONOTONE:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, unit)\ cart \Rightarrow bool)\ t::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow vector\_variation\ t\ f \leq vector\_variation\ s\ f$

thm VECTOR_VARIATION_NEG:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ vector\_variation\ s\ (\lambda x::(real, unit)\ cart.\ vector\_neg\ (f\ x)) = vector\_variation\ s\ f$

thm VECTOR_VARIATION_TRIANGLE:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ f\ s \wedge has\_bounded\_variation\_on\ g\ s \longrightarrow vector\_variation\ s\ (\lambda x::(real, unit)\ cart.\ vector\_add\ (f\ x)\ (g\ x)) \leq vector\_variation\ s\ f + vector\_variation\ s\ g$

thm OPERATIVE_FUNCTION_ENDPOINT_DIFF:

$\forall f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ operative\ vector\_add\ (\lambda k::(real, unit)\ cart \Rightarrow bool.\ vector\_sub\ (f\ (interval\_upperbound\ k))\ (f\ (interval\_lowerbound\ k)))$

thm OPERATIVE_REAL_FUNCTION_ENDPOINT_DIFF:

$\forall f::(real, unit)\ cart \Rightarrow real.\ operative\ op\ +\ (\lambda k::(real, unit)\ cart \Rightarrow bool.\ f\ (interval\_upperbound\ k) - f\ (interval\_lowerbound\ k))$

thm OPERATIVE_LIFTED_VECTOR_VARIATION:

$\forall f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart.\ operative\ (lifted\ op\ +)\ (\lambda i::(real, unit)\ cart \Rightarrow bool.\ if\ has\_bounded\_variation\_on\ f\ i\ then\ SOME\ (vector\_variation\ i\ f)\ else\ NONE)$

thm HAS_BOUNDED_VARIATION_ON_DIVISION:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, unit)\ cart)\ (b::(real, unit)\ cart)\ d::((real, unit)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\forall k::(real, unit)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow has\_bounded\_variation\_on\ f\ k) = has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm VECTOR_VARIATION_ON_DIVISION:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (a::(real, unit)\ cart)\ (b::(real, unit)\ cart)\ d::((real, unit)\ cart \Rightarrow bool) \Rightarrow bool.\ division\_of\ d\ (closed\_interval$

$[(a,\ b)])\ \wedge\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)])\ \longrightarrow\ sum\ d$
$(\lambda k::(real,\ unit)\ cart \Rightarrow bool.\ vector\_variation\ k\ f) = vector\_variation\ (closed\_interval$
$[(a,\ b)])\ f$

thm HAS_BOUNDED_VARIATION_ON_COMBINE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,$
$unit)\ cart)\ c::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop$
$c \wedge HOL\_Light\_Import.drop\ c \leq HOL\_Light\_Import.drop\ b \longrightarrow has\_bounded\_variation\_on$
$f\ (closed\_interval\ [(a,\ b)]) = (has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,$
$c)])\ \wedge\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(c,\ b)]))$

thm VECTOR_VARIATION_COMBINE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,$
$unit)\ cart)\ c::(real,\ unit)\ cart.\ HOL\_Light\_Import.drop\ a \leq HOL\_Light\_Import.drop$
$c \wedge HOL\_Light\_Import.drop\ c \leq HOL\_Light\_Import.drop\ b \wedge has\_bounded\_variation\_on$
$f\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_variation\ (closed\_interval\ [(a,\ c)])\ f\ +$
$vector\_variation\ (closed\_interval\ [(c,\ b)])\ f = vector\_variation\ (closed\_interval$
$[(a,\ b)])\ f$

thm VECTOR_VARIATION_MINUS_FUNCTION_MONOTONE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)$
$cart)\ (c::(real,\ unit)\ cart)\ d::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval$
$[(a,\ b)]) \wedge SUBSET\ (closed\_interval\ [(c,\ d)])\ (closed\_interval\ [(a,\ b)]) \wedge closed\_interval$
$[(c,\ d)] \neq EMPTY \longrightarrow vector\_variation\ (closed\_interval\ [(c,\ d)])\ f - HOL\_Light\_Import.drop$
$(vector\_sub\ (f\ d)\ (f\ c)) \leq vector\_variation\ (closed\_interval\ [(a,\ b)])\ f - HOL\_Light\_Import.drop$
$(vector\_sub\ (f\ b)\ (f\ a))$

thm INCREASING_BOUNDED_VARIATION:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)$
$cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)])$
$\wedge\ IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop$
$y\ \longrightarrow\ HOL\_Light\_Import.drop\ (f\ x)\ \leq\ HOL\_Light\_Import.drop\ (f\ y))\ \longrightarrow$
$has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm DECREASING_BOUNDED_VARIATION:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)$
$cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)])$
$\wedge\ IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop$
$y\ \longrightarrow\ HOL\_Light\_Import.drop\ (f\ y)\ \leq\ HOL\_Light\_Import.drop\ (f\ x))\ \longrightarrow$
$has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INCREASING_VECTOR_VARIATION:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)$
$cart.\ closed\_interval\ [(a,\ b)]\ \neq\ EMPTY\ \wedge\ (\forall (x::(real,\ unit)\ cart)\ y::(real,$
$unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_interval\ [(a,\ b)])\ \wedge$
$HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop$

$(f\ x) \leq HOL\_Light\_Import.drop\ (f\ y)) \longrightarrow vector\_variation\ (closed\_interval\ [(a,\ b)])\ f = HOL\_Light\_Import.drop\ (f\ b) - HOL\_Light\_Import.drop\ (f\ a)$

thm DECREASING_VECTOR_VARIATION:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_interval\ [(a,\ b)] \neq EMPTY \wedge (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop\ (f\ y) \leq HOL\_Light\_Import.drop\ (f\ x)) \longrightarrow vector\_variation\ (closed\_interval\ [(a,\ b)])\ f = HOL\_Light\_Import.drop\ (f\ a) - HOL\_Light\_Import.drop\ (f\ b)$

thm VECTOR_VARIATION_TRANSLATION2:

$\forall (a::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ vector\_variation\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s)\ (\lambda x::(real,\ unit)\ cart.\ f\ (vector\_add\ a\ x)) = vector\_variation\ s\ f$

thm HAS_BOUNDED_VARIATION_TRANSLATION2_EQ:

$\forall (a::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real,\ unit)\ cart.\ f\ (vector\_add\ a\ x))\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s) = has\_bounded\_variation\_on\ f\ s$

thm VECTOR_VARIATION_AFFINITY2:

$\forall (m::real)\ (c::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ vector\_variation\ (IMAGE\ (\lambda x::(real,\ unit)\ cart.\ vector\_add\ (\%\ (inverse\_class.inverse\ m)\ x)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c)))\ s)\ (\lambda x::(real,\ unit)\ cart.\ f\ (vector\_add\ (\%\ m\ x)\ c)) = (if\ m = (0::real)\ then\ 0::real\ else\ vector\_variation\ s\ f)$

thm HAS_BOUNDED_VARIATION_AFFINITY2_EQ:

$\forall (m::real)\ (c::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real,\ unit)\ cart.\ f\ (vector\_add\ (\%\ m\ x)\ c))\ (IMAGE\ (\lambda x::(real,\ unit)\ cart.\ vector\_add\ (\%\ (inverse\_class.inverse\ m)\ x)\ (vector\_neg\ (\%\ (inverse\_class.inverse\ m)\ c)))\ s) = (m = (0::real) \vee has\_bounded\_variation\_on\ f\ s)$

thm VECTOR_VARIATION_AFFINITY:

$\forall (m::real)\ (c::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ vector\_variation\ s\ (\lambda x::(real,\ unit)\ cart.\ f\ (vector\_add\ (\%\ m\ x)\ c)) = (if\ m = (0::real)\ then\ 0::real\ else\ vector\_variation\ (IMAGE\ (\lambda x::(real,\ unit)\ cart.\ vector\_add\ (\%\ m\ x)\ c)\ s)\ f)$

thm HAS_BOUNDED_VARIATION_AFFINITY_EQ:

$\forall (m::real)\ (c::(real,\ unit)\ cart)\ (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real,\ unit)\ cart.\ f$

$(vector\_add\ (\%\ m\ x)\ c))\ s = (m = (0::real) \vee has\_bounded\_variation\_on\ f\ (IMAGE\ (\lambda x::(real, unit)\ cart.\ vector\_add\ (\%\ m\ x)\ c)\ s))$

thm VECTOR_VARIATION_TRANSLATION:

$\forall (a::(real, unit)\ cart)\ (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ vector\_variation\ s\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_add\ a\ x)) = vector\_variation\ (IMAGE\ (vector\_add\ a)\ s)\ f$

thm HAS_BOUNDED_VARIATION_TRANSLATION_EQ:

$\forall (a::(real, unit)\ cart)\ (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_add\ a\ x))\ s = has\_bounded\_variation\_on\ f\ (IMAGE\ (vector\_add\ a)\ s)$

thm VECTOR_VARIATION_TRANSLATION_INTERVAL:

$\forall (a::(real, unit)\ cart)\ (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ vector\_variation\ (closed\_interval\ [(u, v)])\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_add\ a\ x)) = vector\_variation\ (closed\_interval\ [(vector\_add\ a\ u, vector\_add\ a\ v)])\ f$

thm HAS_BOUNDED_VARIATION_TRANSLATION_EQ_INTERVAL:

$\forall (a::(real, unit)\ cart)\ (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (u::(real, unit)\ cart)\ v::(real, unit)\ cart.\ has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_add\ a\ x))\ (closed\_interval\ [(u, v)]) = has\_bounded\_variation\_on\ f\ (closed\_interval\ [(vector\_add\ a\ u, vector\_add\ a\ v)])$

thm HAS_BOUNDED_VARIATION_TRANSLATION:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, unit)\ cart \Rightarrow bool)\ a::(real, unit)\ cart.\ has\_bounded\_variation\_on\ f\ s \longrightarrow has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_add\ a\ x))\ (IMAGE\ (vector\_add\ (vector\_neg\ a))\ s)$

thm VECTOR_VARIATION_REFLECT2:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ vector\_variation\ (IMAGE\ vector\_neg\ s)\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_neg\ x)) = vector\_variation\ s\ f$

thm HAS_BOUNDED_VARIATION_REFLECT2_EQ:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ has\_bounded\_variation\_on\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_neg\ x))\ (IMAGE\ vector\_neg\ s) = has\_bounded\_variation\_on\ f\ s$

thm VECTOR_VARIATION_REFLECT:

$\forall (f::(real, unit)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, unit)\ cart \Rightarrow bool.\ vector\_variation\ s\ (\lambda x::(real, unit)\ cart.\ f\ (vector\_neg\ x)) = vector\_variation\ (IMAGE\ vector\_neg\ s)\ f$

thm HAS_BOUNDED_VARIATION_REFLECT_EQ:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, unit) cart ⇒ bool. has_bounded_variation_on (λx::(real, unit) cart. f (vector_neg x)) s = has_bounded_variation_on f (IMAGE vector_neg s)

thm HAS_BOUNDED_VARIATION_REFLECT_EQ_INTERVAL:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (u::(real, unit) cart) v::(real, unit) cart. has_bounded_variation_on (λx::(real, unit) cart. f (vector_neg x)) (closed_interval [(u, v)]) = has_bounded_variation_on f (closed_interval [(vector_neg v, vector_neg u)])

thm HAS_BOUNDED_VARIATION_DARBOUX:

∀ (f::(real, unit) cart ⇒ (real, unit) cart) (a::(real, unit) cart) b::(real, unit) cart. has_bounded_variation_on f (closed_interval [(a, b)]) = (∃ (g::(real, unit) cart ⇒ (real, unit) cart) h::(real, unit) cart ⇒ (real, unit) cart. (∀ (x::(real, unit) cart) y::(real, unit) cart. IN x (closed_interval [(a, b)]) ∧ IN y (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop x ≤ HOL_Light_Import.drop y ⟶ HOL_Light_Import.drop (g x) ≤ HOL_Light_Import.drop (g y)) ∧ (∀ (x::(real, unit) cart) y::(real, unit) cart. IN x (closed_interval [(a, b)]) ∧ IN y (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop x ≤ HOL_Light_Import.drop y ⟶ HOL_Light_Import.drop (h x) ≤ HOL_Light_Import.drop (h y)) ∧ (∀ x::(real, unit) cart. f x = vector_sub (g x) (h x)))

thm HAS_BOUNDED_VARIATION_DARBOUX_STRICT:

∀ (f::(real, unit) cart ⇒ (real, unit) cart) (a::(real, unit) cart) b::(real, unit) cart. has_bounded_variation_on f (closed_interval [(a, b)]) = (∃ (g::(real, unit) cart ⇒ (real, unit) cart) h::(real, unit) cart ⇒ (real, unit) cart. (∀ (x::(real, unit) cart) y::(real, unit) cart. IN x (closed_interval [(a, b)]) ∧ IN y (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop x < HOL_Light_Import.drop y ⟶ HOL_Light_Import.drop (g x) < HOL_Light_Import.drop (g y)) ∧ (∀ (x::(real, unit) cart) y::(real, unit) cart. IN x (closed_interval [(a, b)]) ∧ IN y (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop x < HOL_Light_Import.drop y ⟶ HOL_Light_Import.drop (h x) < HOL_Light_Import.drop (h y)) ∧ (∀ x::(real, unit) cart. f x = vector_sub (g x) (h x)))

thm HAS_BOUNDED_VARIATION_COMPOSE_INCREASING:

∀ (f::(real, unit) cart ⇒ (real, unit) cart) (g::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) b::(real, unit) cart. (∀ (x::(real, unit) cart) y::(real, unit) cart. IN x (closed_interval [(a, b)]) ∧ IN y (closed_interval [(a, b)]) ∧ HOL_Light_Import.drop x ≤ HOL_Light_Import.drop y ⟶ HOL_Light_Import.drop (f x) ≤ HOL_Light_Import.drop (f y)) ∧ has_bounded_variation_on g (closed_interval [(f a, f b)]) ⟶ has_bounded_variation_on (g ∘ f) (closed_interval [(a, b)])

thm HAS_BOUNDED_VARIATION_ON_REFLECT:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) s::(real, unit) cart ⇒ bool. has_bounded_variation_on f (IMAGE vector_neg s) ⟶ has_bounded_variation_on (λx::(real, unit) cart. f (vector_neg x)) s

thm HAS_BOUNDED_VARIATION_ON_REFLECT_INTERVAL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f* (*closed_interval* [(*vector_neg b*, *vector_neg a*)]) $\longrightarrow$ *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. *f* (*vector_neg x*)) (*closed_interval* [(*a*, *b*)])

thm VECTOR_VARIATION_REFLECT_INTERVAL:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *vector_variation* (*closed_interval* [(*a*, *b*)]) (λ*x*::(*real*, *unit*) *cart*. *f* (*vector_neg x*)) = *vector_variation* (*closed_interval* [(*vector_neg b*, *vector_neg a*)]) *f*

thm HAS_BOUNDED_VARIATION_COMPOSE_DECREASING:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import.drop x* $\leq$ *HOL_Light_Import.drop y* $\longrightarrow$ *HOL_Light_Import.drop* (*f y*) $\leq$ *HOL_Light_Import.drop* (*f x*)) $\wedge$ *has_bounded_variation_on g* (*closed_interval* [(*f b*, *f a*)]) $\longrightarrow$ *has_bounded_variation_on* (*g* $\circ$ *f*) (*closed_interval* [(*a*, *b*)])

thm HAS_BOUNDED_VARIATION_ON_ID:

$\forall$ (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on* (λ*x*::(*real*, *unit*) *cart*. *x*) (*closed_interval* [(*a*, *b*)])

thm HAS_BOUNDED_VARIATION_ON_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *linear f* $\wedge$ *has_bounded_variation_on g* (*IMAGE f* (*closed_interval* [(*a*, *b*)])) $\longrightarrow$ *has_bounded_variation_on* (*g* $\circ$ *f*) (*closed_interval* [(*a*, *b*)])

thm INCREASING_LEFT_LIMIT_1:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) *c*::(*real*, *unit*) *cart*. ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import.drop x* $\leq$ *HOL_Light_Import.drop y* $\longrightarrow$ *HOL_Light_Import.drop* (*f x*) $\leq$ *HOL_Light_Import.drop* (*f y*)) $\wedge$ *IN c* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\exists$ *l*::(*real*, *unit*) *cart*. *-->* *f l* (*within* (*at c*) (*closed_interval* [(*a*, *c*)])))

thm DECREASING_LEFT_LIMIT_1:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*a*::(*real*, *unit*) *cart*) (*b*::(*real*, *unit*) *cart*) *c*::(*real*, *unit*) *cart*. ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import.drop x* $\leq$ *HOL_Light_Import.drop y* $\longrightarrow$ *HOL_Light_Import.drop* (*f y*) $\leq$ *HOL_Light_Import.drop* (*f x*)) $\wedge$ *IN c* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\exists$ *l*::(*real*, *unit*) *cart*. *-->* *f l* (*within* (*at c*) (*closed_interval* [(*a*, *c*)])))

thm INCREASING_RIGHT_LIMIT_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop\ (f\,x) \leq HOL\_Light\_Import.drop\ (f\ y)) \wedge\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\exists\,l::(real,\ unit)\ cart.\ -->\ f\ l\ (within\ (at\ c)\ (closed\_interval\ [(c,\ b)])))$

thm DECREASING_RIGHT_LIMIT_1:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ (\forall (x::(real,\ unit)\ cart)\ y::(real,\ unit)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \wedge HOL\_Light\_Import.drop\ x \leq HOL\_Light\_Import.drop\ y \longrightarrow HOL\_Light\_Import.drop\ (f\,y) \leq HOL\_Light\_Import.drop\ (f\ x)) \wedge\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\exists\,l::(real,\ unit)\ cart.\ -->\ f\ l\ (within\ (at\ c)\ (closed\_interval\ [(c,\ b)])))$

thm HAS_BOUNDED_VECTOR_VARIATION_LEFT_LIMIT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\exists\,l::(real,\ unit)\ cart.\ -->\ f\,l\ (within\ (at\ c)\ (closed\_interval\ [(a,\ c)])))$

thm HAS_BOUNDED_VECTOR_VARIATION_RIGHT_LIMIT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow (\exists\,l::(real,\ unit)\ cart.\ -->\ f\,l\ (within\ (at\ c)\ (closed\_interval\ [(c,\ b)])))$

thm VECTOR_VARIATION_CONTINUOUS_LEFT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow continuous\ (\lambda x::(real,\ unit)\ cart.\ lift\ (vector\_variation\ (closed\_interval\ [(a,\ x)])\ f))\ (within\ (at\ c)\ (closed\_interval\ [(a,\ c)])) = continuous\,f\ (within\ (at\ c)\ (closed\_interval\ [(a,\ c)]))$

thm VECTOR_VARIATION_CONTINUOUS_RIGHT:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow continuous\ (\lambda x::(real,\ unit)\ cart.\ lift\ (vector\_variation\ (closed\_interval\ [(a,\ x)])\ f))\ (within\ (at\ c)\ (closed\_interval\ [(c,\ b)])) = continuous\,f\ (within\ (at\ c)\ (closed\_interval\ [(c,\ b)]))$

thm VECTOR_VARIATION_CONTINUOUS:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ unit)\ cart)\ (a::(real,\ unit)\ cart)\ (b::(real,\ unit)\ cart)\ c::(real,\ unit)\ cart.\ has\_bounded\_variation\_on\ f\ (closed\_interval\ [(a,\ b)]) \wedge\ IN\ c\ (closed\_interval\ [(a,\ b)]) \longrightarrow continuous\ (\lambda x::(real,\ unit)\ cart.\ lift$

(*vector_variation* (*closed_interval* [(*a*, *x*)]) *f*)) (*within* (*at c*) (*closed_interval* [(*a*, *b*)])) = *continuous f* (*within* (*at c*) (*closed_interval* [(*a*, *b*)]))

thm HAS_BOUNDED_VARIATION_DARBOUX_STRONG:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ ($\exists$ (*g*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) *h*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*. ($\forall$ *x*::(*real*, *unit*) *cart*. *f x* = *vector_sub* (*g x*) (*h x*)) $\wedge$ ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import*.*drop x* $\leq$ *HOL_Light_Import*.*drop y* $\longrightarrow$ *HOL_Light_Import*.*drop* (*g x*) $\leq$ *HOL_Light_Import*.*drop* (*g y*)) $\wedge$ ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import*.*drop x* $\leq$ *HOL_Light_Import*.*drop y* $\longrightarrow$ *HOL_Light_Import*.*drop* (*h x*) $\leq$ *HOL_Light_Import*.*drop* (*h y*)) $\wedge$ ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import*.*drop x* < *HOL_Light_Import*.*drop y* $\longrightarrow$ *HOL_Light_Import*.*drop* (*g x*) < *HOL_Light_Import*.*drop* (*g y*)) $\wedge$ ($\forall$ (*x*::(*real*, *unit*) *cart*) *y*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *IN y* (*closed_interval* [(*a*, *b*)]) $\wedge$ *HOL_Light_Import*.*drop x* < *HOL_Light_Import*.*drop y* $\longrightarrow$ *HOL_Light_Import*.*drop* (*h x*) < *HOL_Light_Import*.*drop* (*h y*)) $\wedge$ ($\forall$ *x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *continuous f* (*within* (*at x*) (*closed_interval* [(*a*, *x*)])) $\longrightarrow$ *continuous g* (*within* (*at x*) (*closed_interval* [(*a*, *x*)])) $\wedge$ *continuous h* (*within* (*at x*) (*closed_interval* [(*a*, *x*)]))) $\wedge$ ($\forall$ *x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *continuous f* (*within* (*at x*) (*closed_interval* [(*x*, *b*)])) $\longrightarrow$ *continuous g* (*within* (*at x*) (*closed_interval* [(*x*, *b*)])) $\wedge$ *continuous h* (*within* (*at x*) (*closed_interval* [(*x*, *b*)]))) $\wedge$ ($\forall$ *x*::(*real*, *unit*) *cart*. *IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ *continuous f* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) $\longrightarrow$ *continuous g* (*within* (*at x*) (*closed_interval* [(*a*, *b*)])) $\wedge$ *continuous h* (*within* (*at x*) (*closed_interval* [(*a*, *b*)]))))

thm HAS_BOUNDED_VARIATION_COUNTABLE_DISCONTINUITIES:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *unit*) *cart*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *has_bounded_variation_on f* (*closed_interval* [(*a*, *b*)]) $\longrightarrow$ *COUNTABLE* (*GSPEC* (λ*GEN%PVAR%2092*::(*real*, *unit*) *cart*. $\exists$ *x*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%2092* (*IN x* (*closed_interval* [(*a*, *b*)]) $\wedge$ $\neg$ *continuous f* (*at x*)) *x*))

thm HAS_BOUNDED_VARIATION_ABSOLUTELY_INTEGRABLE_DERIVATIVE:

$\forall$ (*f*::(*real*, *unit*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *unit*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *unit*) *cart*) *b*::(*real*, *unit*) *cart*. *COUNTABLE s* $\wedge$ *continuous_on f* (*closed_interval* [(*a*, *b*)]) $\wedge$ ($\forall$ *x*::(*real*, *unit*) *cart*. *IN x* (*DIFF* (*closed_interval* [(*a*, *b*)]) *s*) $\longrightarrow$ *differentiable f* (*at x*)) $\longrightarrow$ *has_bounded_variation_on f* (*closed_interval* [(*a*, *b*)]) = *absolutely_integrable_on* (λ*x*::(*real*, *unit*) *cart*. *vector_derivative f* (*at x*)) (*closed_interval* [(*a*, *b*)])

thm HAS_BOUNDED_VARIATION_INTEGRABLE_NORM_DERIVATIVE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)$ $(a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ COUNTABLE\ s \wedge continuous\_on\ f$ $(closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (DIFF\ (closed\_interval$ $[(a,\ b)])\ s) \longrightarrow differentiable\ f\ (at\ x)) \longrightarrow has\_bounded\_variation\_on\ f\ (closed\_interval$ $[(a,\ b)]) = integrable\_on\ (\lambda x::(real,\ unit)\ cart.\ lift\ (vector\_norm\ (vector\_derivative$ $f\ (at\ x))))\ (closed\_interval\ [(a,\ b)])$

thm VECTOR_VARIATION_INTEGRAL_NORM_DERIVATIVE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)$ $(a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ COUNTABLE\ s \wedge continuous\_on\ f$ $(closed\_interval\ [(a,\ b)]) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (DIFF\ (closed\_interval$ $[(a,\ b)])\ s) \longrightarrow differentiable\ f\ (at\ x)) \wedge has\_bounded\_variation\_on\ f\ (closed\_interval$ $[(a,\ b)]) \longrightarrow vector\_variation\ (closed\_interval\ [(a,\ b)])\ f = HOL\_Light\_Import.drop$ $(integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ unit)\ cart.\ lift\ (vector\_norm\ (vector\_derivative$ $f\ (at\ x)))))$

thm INTEGRAL_PASTECART_CONTINUOUS:

$\forall (f::(real,\ (?'c::type,\ ?'b::type)\ finite\_sum)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,$ $?'c::type)\ cart)\ (b::(real,\ ?'c::type)\ cart)\ (c::(real,\ ?'b::type)\ cart)\ d::(real,$ $?'b::type)\ cart.\ continuous\_on\ f\ (closed\_interval\ [(pastecart\ a\ c,\ pastecart\ b$ $d)]) \longrightarrow integral\ (closed\_interval\ [(pastecart\ a\ c,\ pastecart\ b\ d)])\ f = integral$ $(closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'c::type)\ cart.\ integral\ (closed\_interval\ [(c,$ $d)])\ (\lambda y::(real,\ ?'b::type)\ cart.\ f\ (pastecart\ x\ y)))$

thm INTEGRAL_SWAP_CONTINUOUS:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)$ $(a::(real,\ ?'c::type)\ cart)\ (b::(real,\ ?'c::type)\ cart)\ (c::(real,\ ?'b::type)\ cart)$ $d::(real,\ ?'b::type)\ cart.\ continuous\_on\ (\lambda z::(real,\ (?'c::type,\ ?'b::type)\ finite\_sum)$ $cart.\ f\ (fstcart\ z)\ (sndcart\ z))\ (closed\_interval\ [(pastecart\ a\ c,\ pastecart\ b\ d)])$ $\longrightarrow integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'c::type)\ cart.\ integral\ (closed\_interval$ $[(c,\ d)])\ (f\ x)) = integral\ (closed\_interval\ [(c,\ d)])\ (\lambda y::(real,\ ?'b::type)\ cart.$ $integral\ (closed\_interval\ [(a,\ b)])\ (\lambda x::(real,\ ?'c::type)\ cart.\ f\ x\ y))$

thm DEF_rectifiable_path:

$rectifiable\_path = (\lambda\_1783303::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ path$ $\_1783303 \wedge has\_bounded\_variation\_on\ \_1783303\ (closed\_interval\ [(vec\ (0::nat),$ $vec\ (1::nat))]))$

thm rectifiable_path:

$\forall g::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ rectifiable\_path\ g = (path\ g \wedge$ $has\_bounded\_variation\_on\ g\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))]))$

thm DEF_path_length:

$path\_length = vector\_variation\ (closed\_interval\ [(vec\ (0::nat),\ vec\ (1::nat))])$

thm path_length:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path_length g* = *vector_variation* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) *g*

thm BOUNDED_RECTIFIABLE_PATH_IMAGE:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g* ⟶ *bounded* (*path_image g*)

thm RECTIFIABLE_PATH_IMP_PATH:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g* ⟶ *path g*

thm RECTIFIABLE_PATH_LINEPATH:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *rectifiable_path* (*linepath* (*a*, *b*))

thm RECTIFIABLE_PATH_REVERSEPATH:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path* (*reversepath g*) = *rectifiable_path g*

thm PATH_LENGTH_REVERSEPATH:

∀ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *path_length* (*reversepath g*) = *path_length g*

thm RECTIFIABLE_PATH_SUBPATH:

∀ (*u*::(*real*, *unit*) *cart*) (*v*::(*real*, *unit*) *cart*) *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g* ∧ *IN u* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ∧ *IN v* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) ⟶ *rectifiable_path* (*subpath u v g*)

thm RECTIFIABLE_PATH_JOIN:

∀ (*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *pathfinish g1* = *pathstart g2* ⟶ *rectifiable_path* (++ *g1 g2*) = (*rectifiable_path g1* ∧ *rectifiable_path g2*)

thm RECTIFIABLE_PATH_JOIN_IMP:

∀ (*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g1* ∧ *rectifiable_path g2* ∧ *pathfinish g1* = *pathstart g2* ⟶ *rectifiable_path* (++ *g1 g2*)

thm RECTIFIABLE_PATH_JOIN_EQ:

∀ (*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g1* ∧ *rectifiable_path g2* ⟶ *rectifiable_path* (++ *g1 g2*) = (*pathfinish g1* = *pathstart g2*)

thm PATH_LENGTH_JOIN:

∀ (*g1*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *g2*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *rectifiable_path g1* ∧ *rectifiable_path g2* ∧ *pathfinish g1* = *pathstart g2* ⟶ *path_length* (++ *g1 g2*) = *path_length g1* + *path_length g2*

thm RECTIFIABLE_PATH_DIFFERENTIABLE:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *unit*) *cart* ⇒ *bool*. *COUNTABLE s* ∧ *path g* ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*DIFF* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) *s*) ⟶ *differentiable g* (*at t*)) ⟶ *rectifiable_path g* = *absolutely_integrable_on* (λ*t*::(*real*, *unit*) *cart*. *vector_derivative g* (*at t*)) (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))])

thm PATH_LENGTH_DIFFERENTIABLE:

∀ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *unit*) *cart* ⇒ *bool*. *COUNTABLE s* ∧ *rectifiable_path g* ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*DIFF* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) *s*) ⟶ *differentiable g* (*at t*)) ⟶ *path_length g* = *HOL_Light_Import.drop* (*integral* (*closed_interval* [(*vec* (*0*::*nat*), *vec* (*1*::*nat*))]) (λ*t*::(*real*, *unit*) *cart*. *lift* (*vector_norm* (*vector_derivative g* (*at t*)))))

thm DEF_has_measure:

has_measure = (λ(*_1783492*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *_1783493*::*real*. *has_integral* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vec* (*1*::*nat*)) (*lift _1783493*) *_1783492*)

thm has_measure:

∀ (*m*::*real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *has_measure s m* = *has_integral* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vec* (*1*::*nat*)) (*lift m*) *s*

thm DEF_measurable:

measurable = (λ*_1783504*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *m*::*real*. *has_measure _1783504 m*)

thm measurable:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* = (∃ *m*::*real*. *has_measure s m*)

thm DEF_measure:

HOL_Light_Import.measure = (λ*_1783509*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SOME m*::*real*. *has_measure _1783509 m*)

thm measure:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.measure s* = (*SOME m*::*real*. *has_measure s m*)

thm HAS_MEASURE_MEASURE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* = *has_measure s* (*HOL_Light_Import.measure s*)

thm HAS_MEASURE_UNIQUE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*m1*::*real*) *m2*::*real*. *has_measure s m1* ∧ *has_measure s m2* ⟶ *m1* = *m2*

thm MEASURE_UNIQUE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ m::real.\ has\_measure\ s\ m \longrightarrow HOL\_Light\_Import.measure\ s = m$

thm HAS_MEASURE_MEASURABLE_MEASURE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ m::real.\ has\_measure\ s\ m = (measurable\ s \wedge HOL\_Light\_Import.measure\ s = m)$

thm HAS_MEASURE_IMP_MEASURABLE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ m::real.\ has\_measure\ s\ m \longrightarrow measurable\ s$

thm HAS_MEASURE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ m::real.\ has\_measure\ s\ m = has\_integral\ (\lambda x::(real, ?'a::type)\ cart.\ if\ IN\ x\ s\ then\ vec\ (1::nat)\ else\ vec\ (0::nat))\ (lift\ m)\ HOL\_Light\_Import.UNIV$

thm MEASURABLE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ s = integrable\_on\ (\lambda x::(real, ?'a::type)\ cart.\ vec\ (1::nat))\ s$

thm MEASURABLE_INTEGRABLE:

$measurable\ (?s::(real, ?'a::type)\ cart \Rightarrow bool) = integrable\_on\ (\lambda x::(real, ?'a::type)\ cart.\ if\ IN\ x\ ?s\ then\ vec\ (1::nat)\ else\ vec\ (0::nat))\ HOL\_Light\_Import.UNIV$

thm MEASURE_INTEGRAL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow HOL\_Light\_Import.measure\ s = HOL\_Light\_Import.drop\ (integral\ s\ (\lambda x::(real, ?'a::type)\ cart.\ vec\ (1::nat)))$

thm MEASURE_INTEGRAL_UNIV:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow HOL\_Light\_Import.measure\ s = HOL\_Light\_Import.drop\ (integral\ HOL\_Light\_Import.UNIV\ (\lambda x::(real, ?'a::type)\ cart.\ if\ IN\ x\ s\ then\ vec\ (1::nat)\ else\ vec\ (0::nat)))$

thm INTEGRAL_MEASURE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow integral\ s\ (\lambda x::(real, ?'a::type)\ cart.\ vec\ (1::nat)) = lift\ (HOL\_Light\_Import.measure\ s)$

thm INTEGRAL_MEASURE_UNIV:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow integral\ HOL\_Light\_Import.UNIV\ (\lambda x::(real, ?'a::type)\ cart.\ if\ IN\ x\ s\ then\ vec\ (1::nat)\ else\ vec\ (0::nat)) = lift\ (HOL\_Light\_Import.measure\ s)$

thm HAS_MEASURE_INTERVAL_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ has\_measure\ (closed\_interval\ [(a, b)])\ (content\ (closed\_interval\ [(a, b)]))$

thm HAS_MEASURE_INTERVAL:

(∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. has_measure (closed_interval [(a, b)]) (content (closed_interval [(a, b)]))) ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. has_measure (open_interval (a, b)) (content (closed_interval [(a, b)])))

thm MEASURABLE_INTERVAL:

(∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. measurable (closed_interval [(a, b)])) ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. measurable (open_interval (a, b)))

thm HAS_MEASURE_INTERVAL_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. has_measure (open_interval (a, b)) (content (closed_interval [(a, b)]))

thm MEASURE_INTERVAL_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. HOL_Light_Import.measure (open_interval (a, b)) = content (closed_interval [(a, b)])

thm MEASURE_INTERVAL_conjunct0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. HOL_Light_Import.measure (closed_interval [(a, b)]) = content (closed_interval [(a, b)])

thm MEASURE_INTERVAL:

(∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. HOL_Light_Import.measure (closed_interval [(a, b)]) = content (closed_interval [(a, b)])) ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. HOL_Light_Import.measure (open_interval (a, b)) = content (closed_interval [(a, b)]))

thm MEASURE_INTERVAL_1:

(∀ (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.measure (closed_interval [(a, b)]) = (if HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b then HOL_Light_Import.drop b − HOL_Light_Import.drop a else (0::real))) ∧ (∀ (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.measure (open_interval (a, b)) = (if HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b then HOL_Light_Import.drop b − HOL_Light_Import.drop a else (0::real)))

thm MEASURE_INTERVAL_1_conjunct1:

∀ (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.measure (open_interval (a, b)) = (if HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b then HOL_Light_Import.drop b − HOL_Light_Import.drop a else (0::real))

thm MEASURE_INTERVAL_1_conjunct0:

∀ (a::(real, unit) cart) b::(real, unit) cart. HOL_Light_Import.measure (closed_interval [(a, b)]) = (if HOL_Light_Import.drop a ≤ HOL_Light_Import.drop b then HOL_Light_Import.drop b − HOL_Light_Import.drop a else (0::real))

thm MEASURE_INTERVAL_1_ALT:

$(\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) = (if\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b\ then\ HOL\_Light\_Import.drop\ b - HOL\_Light\_Import.drop\ a\ else\ (0::real))) \wedge (\forall (a::(real, unit)\ cart)\ b::(real, unit)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ HOL\_Light\_Import.drop\ a < HOL\_Light\_Import.drop\ b\ then\ HOL\_Light\_Import.drop\ b - HOL\_Light\_Import.drop\ a\ else\ (0::real)))$

thm MEASURE_INTERVAL_2:

$(\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real))) \wedge (\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real)))$

thm MEASURE_INTERVAL_2_conjunct1:

$\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real))$

thm MEASURE_INTERVAL_2_conjunct0:

$\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real))$

thm MEASURE_INTERVAL_2_ALT:

$(\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) = (if\ \$\ a\ (1::nat) < \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) < \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real))) \wedge (\forall (a::(real, 2)\ cart)\ b::(real, 2)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ \$\ a\ (1::nat) < \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) < \$\ b\ (2::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * (\$\ b\ (2::nat) - \$\ a\ (2::nat))\ else\ (0::real)))$

thm MEASURE_INTERVAL_3:

$(\forall (a::(real, 3)\ cart)\ b::(real, 3)\ cart.\ HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat) \wedge \$\ a\ (3::nat) \leq \$\ b\ (3::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * ((\$\ b\ (2::nat) - \$\ a\ (2::nat)) * (\$\ b\ (3::nat) - \$\ a\ (3::nat)))\ else\ (0::real))) \wedge (\forall (a::(real, 3)\ cart)\ b::(real, 3)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat) \wedge \$\ a\ (3::nat) \leq \$\ b\ (3::nat)\ then\ (\$\ b\ (1::nat) - \$\ a\ (1::nat)) * ((\$\ b\ (2::nat) - \$\ a\ (2::nat)) * (\$\ b\ (3::nat) - \$\ a\ (3::nat)))\ else\ (0::real)))$

thm MEASURE_INTERVAL_3_conjunct1:

$\forall (a::(real, 3)\ cart)\ b::(real, 3)\ cart.\ HOL\_Light\_Import.measure\ (open\_interval\ (a,\ b)) = (if\ \$\ a\ (1::nat) \leq \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) \leq \$\ b\ (2::nat) \wedge \$\ a$

$(3::nat) \leq \$ \ b \ (3::nat)$ *then* $(\$ \ b \ (1::nat) - \$ \ a \ (1::nat)) * ((\$ \ b \ (2::nat) - \$ \ a \ (2::nat)) * (\$ \ b \ (3::nat) - \$ \ a \ (3::nat)))$ *else* $(0::real))$

thm MEASURE_INTERVAL_3_conjunct0:

$\forall (a::(real, 3) \ cart) \ b::(real, 3) \ cart. \ HOL\_Light\_Import.measure \ (closed\_interval \ [(a, b)]) = (if \ \$ \ a \ (1::nat) \leq \$ \ b \ (1::nat) \land \$ \ a \ (2::nat) \leq \$ \ b \ (2::nat) \land \$ \ a \ (3::nat) \leq \$ \ b \ (3::nat) \ then \ (\$ \ b \ (1::nat) - \$ \ a \ (1::nat)) * ((\$ \ b \ (2::nat) - \$ \ a \ (2::nat)) * (\$ \ b \ (3::nat) - \$ \ a \ (3::nat))) \ else \ (0::real))$

thm MEASURE_INTERVAL_3_ALT:

$(\forall (a::(real, 3) \ cart) \ b::(real, 3) \ cart. \ HOL\_Light\_Import.measure \ (closed\_interval \ [(a, b)]) = (if \ \$ \ a \ (1::nat) < \$ \ b \ (1::nat) \land \$ \ a \ (2::nat) < \$ \ b \ (2::nat) \land \$ \ a \ (3::nat) < \$ \ b \ (3::nat) \ then \ (\$ \ b \ (1::nat) - \$ \ a \ (1::nat)) * ((\$ \ b \ (2::nat) - \$ \ a \ (2::nat)) * (\$ \ b \ (3::nat) - \$ \ a \ (3::nat))) \ else \ (0::real))) \land (\forall (a::(real, 3) \ cart) \ b::(real, 3) \ cart. \ HOL\_Light\_Import.measure \ (open\_interval \ (a, b)) = (if \ \$ \ a \ (1::nat) < \$ \ b \ (1::nat) \land \$ \ a \ (2::nat) < \$ \ b \ (2::nat) \land \$ \ a \ (3::nat) < \$ \ b \ (3::nat) \ then \ (\$ \ b \ (1::nat) - \$ \ a \ (1::nat)) * ((\$ \ b \ (2::nat) - \$ \ a \ (2::nat)) * (\$ \ b \ (3::nat) - \$ \ a \ (3::nat))) \ else \ (0::real)))$

thm MEASURABLE_INTER:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ t::(real, ?'a::type) \ cart \Rightarrow bool. \ measurable \ s \land measurable \ t \longrightarrow measurable \ (HOL\_Light\_Import.INTER \ s \ t)$

thm MEASURABLE_UNION:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ t::(real, ?'a::type) \ cart \Rightarrow bool. \ measurable \ s \land measurable \ t \longrightarrow measurable \ (HOL\_Light\_Import.UNION \ s \ t)$

thm HAS_MEASURE_DISJOINT_UNION:

$\forall (s1::(real, ?'a::type) \ cart \Rightarrow bool) \ (s2::(real, ?'a::type) \ cart \Rightarrow bool) \ (m1::real) \ m2::real. \ has\_measure \ s1 \ m1 \land has\_measure \ s2 \ m2 \land DISJOINT \ s1 \ s2 \longrightarrow has\_measure \ (HOL\_Light\_Import.UNION \ s1 \ s2) \ (m1 + m2)$

thm MEASURE_DISJOINT_UNION:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ t::(real, ?'a::type) \ cart \Rightarrow bool. \ measurable \ s \land measurable \ t \land DISJOINT \ s \ t \longrightarrow HOL\_Light\_Import.measure \ (HOL\_Light\_Import.UNION \ s \ t) = HOL\_Light\_Import.measure \ s + HOL\_Light\_Import.measure \ t$

thm MEASURE_DISJOINT_UNION_EQ:

$\forall (s::(real, ?'a::type) \ cart \Rightarrow bool) \ (t::(real, ?'a::type) \ cart \Rightarrow bool) \ u::(real, ?'a::type) \ cart \Rightarrow bool. \ measurable \ s \land measurable \ t \land HOL\_Light\_Import.UNION \ s \ t = u \land DISJOINT \ s \ t \longrightarrow HOL\_Light\_Import.measure \ s + HOL\_Light\_Import.measure \ t = HOL\_Light\_Import.measure \ u$

thm HAS_MEASURE_POS_LE:

$\forall (m::real) \ s::(real, ?'a::type) \ cart \Rightarrow bool. \ has\_measure \ s \ m \longrightarrow (0::real) \leq m$

thm MEASURE_POS_LE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ⟶ (*0*::*real*) ≤ *HOL_Light_Import.measure s*

thm HAS_MEASURE_SUBSET:

∀ (*s1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*m1*::*real*) *m2*::*real*. *has_measure s1 m1* ∧ *has_measure s2 m2* ∧ *SUBSET s1 s2* ⟶ *m1* ≤ *m2*

thm MEASURE_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ∧ *measurable t* ∧ *SUBSET s t* ⟶ *HOL_Light_Import.measure s* ≤ *HOL_Light_Import.measure t*

thm HAS_MEASURE_0:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *has_measure s* (*0*::*real*) = *negligible s*

thm MEASURE_EQ_0:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* ⟶ *HOL_Light_Import.measure s* = (*0*::*real*)

thm NEGLIGIBLE_IMP_MEASURABLE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* ⟶ *measurable s*

thm HAS_MEASURE_EMPTY:

has_measure EMPTY (*0*::*real*)

thm MEASURE_EMPTY:

HOL_Light_Import.measure EMPTY = (*0*::*real*)

thm MEASURABLE_EMPTY:

measurable EMPTY

thm MEASURABLE_MEASURE_EQ_0:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ⟶ (*HOL_Light_Import.measure s* = (*0*::*real*)) = *negligible s*

thm MEASURABLE_MEASURE_POS_LT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ⟶ ((*0*::*real*) < *HOL_Light_Import.measure s*) = (¬ *negligible s*)

thm NEGLIGIBLE_INTERVAL:

(∀ (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *negligible* (*closed_interval* [(*a*, *b*)]) = (*open_interval* (*a*, *b*) = *EMPTY*)) ∧ (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *negligible* (*open_interval* (*a*, *b*)) = (*open_interval* (*a*, *b*) = *EMPTY*))

thm MEASURABLE_UNIONS:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE f* ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *measurable s*) ⟶ *measurable* (*UNIONS f*)

thm HAS_MEASURE_DIFF_SUBSET:

∀ (*s1*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s2*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*m1*::*real*) *m2*::*real*. *has_measure s1 m1* ∧ *has_measure s2 m2* ∧ *SUBSET s2 s1* ⟶ *has_measure* (*DIFF s1 s2*) (*m1* − *m2*)

thm MEASURABLE_DIFF:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ∧ *measurable t* ⟶ *measurable* (*DIFF s t*)

thm MEASURE_DIFF_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ∧ *measurable t* ∧ *SUBSET t s* ⟶ *HOL_Light_Import.measure* (*DIFF s t*) = *HOL_Light_Import.measure s* − *HOL_Light_Import.measure t*

thm HAS_MEASURE_UNION_NEGLIGIBLE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *has_measure s m* ∧ *negligible t* ⟶ *has_measure* (*HOL_Light_Import.UNION s t*) *m*

thm HAS_MEASURE_DIFF_NEGLIGIBLE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *has_measure s m* ∧ *negligible t* ⟶ *has_measure* (*DIFF s t*) *m*

thm HAS_MEASURE_UNION_NEGLIGIBLE_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *negligible t* ⟶ *has_measure* (*HOL_Light_Import.UNION s t*) *m* = *has_measure s m*

thm HAS_MEASURE_DIFF_NEGLIGIBLE_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *negligible t* ⟶ *has_measure* (*DIFF s t*) *m* = *has_measure s m*

thm HAS_MEASURE_ALMOST:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *has_measure s m* ∧ *negligible t* ∧ *HOL_Light_Import.UNION s t* = *HOL_Light_Import.UNION s' t* ⟶ *has_measure s' m*

thm HAS_MEASURE_ALMOST_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible t* ∧ *HOL_Light_Import.UNION s t* = *HOL_Light_Import.UNION s' t* ⟶ *has_measure s* (*?m*::*real*) = *has_measure s' ?m*

thm MEASURABLE_ALMOST:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) (s'::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. measurable s \wedge negligible t \wedge HOL\_Light\_Import.UNION s t = HOL\_Light\_Import.UNION s' t \longrightarrow measurable s'$

thm HAS_MEASURE_NEGLIGIBLE_UNION:

$\forall (s1::(real, ?'a::type) cart \Rightarrow bool) (s2::(real, ?'a::type) cart \Rightarrow bool) (m1::real) m2::real. has\_measure s1 m1 \wedge has\_measure s2 m2 \wedge negligible (HOL\_Light\_Import.INTER s1 s2) \longrightarrow has\_measure (HOL\_Light\_Import.UNION s1 s2) (m1 + m2)$

thm MEASURE_NEGLIGIBLE_UNION:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. measurable s \wedge measurable t \wedge negligible (HOL\_Light\_Import.INTER s t) \longrightarrow HOL\_Light\_Import.measure (HOL\_Light\_Import.UNION s t) = HOL\_Light\_Import.measure s + HOL\_Light\_Import.measure t$

thm MEASURE_NEGLIGIBLE_UNION_EQ:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) (t::(real, ?'a::type) cart \Rightarrow bool) u::(real, ?'a::type) cart \Rightarrow bool. measurable s \wedge measurable t \wedge HOL\_Light\_Import.UNION s t = u \wedge negligible (HOL\_Light\_Import.INTER s t) \longrightarrow HOL\_Light\_Import.measure s + HOL\_Light\_Import.measure t = HOL\_Light\_Import.measure u$

thm HAS_MEASURE_NEGLIGIBLE_SYMDIFF:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) (t::(real, ?'a::type) cart \Rightarrow bool) m::real. has\_measure s m \wedge negligible (HOL\_Light\_Import.UNION (DIFF s t) (DIFF t s)) \longrightarrow has\_measure t m$

thm MEASURABLE_NEGLIGIBLE_SYMDIFF:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. measurable s \wedge negligible (HOL\_Light\_Import.UNION (DIFF s t) (DIFF t s)) \longrightarrow measurable t$

thm MEASURABLE_NEGLIGIBLE_SYMDIFF_EQ:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. negligible (HOL\_Light\_Import.UNION (DIFF s t) (DIFF t s)) \longrightarrow measurable s = measurable t$

thm MEASURE_NEGLIGIBLE_SYMDIFF:

$\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool. negligible (HOL\_Light\_Import.UNION (DIFF s t) (DIFF t s)) \longrightarrow HOL\_Light\_Import.measure s = HOL\_Light\_Import.measure t$

thm HAS_MEASURE_NEGLIGIBLE_UNIONS:

$\forall (m::((real, ?'a::type) cart \Rightarrow bool) \Rightarrow real) f::((real, ?'a::type) cart \Rightarrow bool) \Rightarrow bool. FINITE f \wedge (\forall s::(real, ?'a::type) cart \Rightarrow bool. IN s f \longrightarrow has\_measure s (m s)) \wedge (\forall (s::(real, ?'a::type) cart \Rightarrow bool) t::(real, ?'a::type) cart \Rightarrow bool.$

IN s f ∧ IN t f ∧ s ≠ t ⟶ negligible (HOL_Light_Import.INTER s t)) ⟶ has_measure (UNIONS f) (sum f m)

thm MEASURE_NEGLIGIBLE_UNIONS:

∀(m::((real, ?'a::type) cart ⇒ bool) ⇒ real) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ has_measure s (m s)) ∧ (∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. IN s f ∧ IN t f ∧ s ≠ t ⟶ negligible (HOL_Light_Import.INTER s t)) ⟶ HOL_Light_Import.measure (UNIONS f) = sum f m

thm HAS_MEASURE_DISJOINT_UNIONS:

∀(m::((real, ?'a::type) cart ⇒ bool) ⇒ real) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ has_measure s (m s)) ∧ (∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. IN s f ∧ IN t f ∧ s ≠ t ⟶ DISJOINT s t) ⟶ has_measure (UNIONS f) (sum f m)

thm MEASURE_DISJOINT_UNIONS:

∀(m::((real, ?'a::type) cart ⇒ bool) ⇒ real) f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE f ∧ (∀s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ has_measure s (m s)) ∧ (∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. IN s f ∧ IN t f ∧ s ≠ t ⟶ DISJOINT s t) ⟶ HOL_Light_Import.measure (UNIONS f) = sum f m

thm HAS_MEASURE_NEGLIGIBLE_UNIONS_IMAGE:

∀(f::?'b::type ⇒ (real, ?'a::type) cart ⇒ bool) s::?'b::type ⇒ bool. FINITE s ∧ (∀x::?'b::type. IN x s ⟶ measurable (f x)) ∧ (∀(x::?'b::type) y::?'b::type. IN x s ∧ IN y s ∧ x ≠ y ⟶ negligible (HOL_Light_Import.INTER (f x) (f y))) ⟶ has_measure (UNIONS (IMAGE f s)) (sum s (λx::?'b::type. HOL_Light_Import.measure (f x)))

thm MEASURE_NEGLIGIBLE_UNIONS_IMAGE:

∀(f::?'b::type ⇒ (real, ?'a::type) cart ⇒ bool) s::?'b::type ⇒ bool. FINITE s ∧ (∀x::?'b::type. IN x s ⟶ measurable (f x)) ∧ (∀(x::?'b::type) y::?'b::type. IN x s ∧ IN y s ∧ x ≠ y ⟶ negligible (HOL_Light_Import.INTER (f x) (f y))) ⟶ HOL_Light_Import.measure (UNIONS (IMAGE f s)) = sum s (λx::?'b::type. HOL_Light_Import.measure (f x))

thm HAS_MEASURE_DISJOINT_UNIONS_IMAGE:

∀(f::?'b::type ⇒ (real, ?'a::type) cart ⇒ bool) s::?'b::type ⇒ bool. FINITE s ∧ (∀x::?'b::type. IN x s ⟶ measurable (f x)) ∧ (∀(x::?'b::type) y::?'b::type. IN x s ∧ IN y s ∧ x ≠ y ⟶ DISJOINT (f x) (f y)) ⟶ has_measure (UNIONS (IMAGE f s)) (sum s (λx::?'b::type. HOL_Light_Import.measure (f x)))

thm MEASURE_DISJOINT_UNIONS_IMAGE:

∀(f::?'b::type ⇒ (real, ?'a::type) cart ⇒ bool) s::?'b::type ⇒ bool. FINITE s ∧ (∀x::?'b::type. IN x s ⟶ measurable (f x)) ∧ (∀(x::?'b::type) y::?'b::type. IN x

$s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (f\,x)\ (f\,y)) \longrightarrow HOL\_Light\_Import.measure\ (UNIONS\ (IMAGE\ f\ s)) = sum\ s\ (\lambda x::?'b::type.\ HOL\_Light\_Import.measure\ (f\,x))$

thm HAS_MEASURE_NEGLIGIBLE_UNIONS_IMAGE_STRONG:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2104::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%2104\ (IN\ x\ s \wedge f\ x \neq EMPTY)\ x)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow measurable\ (f\ x)) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ (f\,x)\ (f\,y))) \longrightarrow has\_measure\ (UNIONS\ (IMAGE\ f\,s))\ (sum\ s\ (\lambda x::?'b::type.\ HOL\_Light\_Import.measure\ (f\,x)))$

thm MEASURE_NEGLIGIBLE_UNIONS_IMAGE_STRONG:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2105::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%2105\ (IN\ x\ s \wedge f\ x \neq EMPTY)\ x)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow measurable\ (f\ x)) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ (f\ x)\ (f\ y))) \longrightarrow HOL\_Light\_Import.measure\ (UNIONS\ (IMAGE\ f\ s)) = sum\ s\ (\lambda x::?'b::type.\ HOL\_Light\_Import.measure\ (f\,x))$

thm HAS_MEASURE_DISJOINT_UNIONS_IMAGE_STRONG:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2106::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%2106\ (IN\ x\ s \wedge f\ x \neq EMPTY)\ x)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow measurable\ (f\ x)) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (f\ x)\ (f\ y)) \longrightarrow has\_measure\ (UNIONS\ (IMAGE\ f\ s))\ (sum\ s\ (\lambda x::?'b::type.\ HOL\_Light\_Import.measure\ (f\,x)))$

thm MEASURE_DISJOINT_UNIONS_IMAGE_STRONG:

$\forall (f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool)\ s::?'b::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2107::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%2107\ (IN\ x\ s \wedge f\ x \neq EMPTY)\ x)) \wedge (\forall x::?'b::type.\ IN\ x\ s \longrightarrow measurable\ (f\ x)) \wedge (\forall (x::?'b::type)\ y::?'b::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (f\ x)\ (f\ y)) \longrightarrow HOL\_Light\_Import.measure\ (UNIONS\ (IMAGE\ f\ s)) = sum\ s\ (\lambda x::?'b::type.\ HOL\_Light\_Import.measure\ (f\,x))$

thm MEASURE_UNION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \wedge measurable\ t \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.measure\ s + (HOL\_Light\_Import.measure\ t - HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ s\ t))$

thm MEASURE_UNION_LE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \wedge measurable\ t \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.UNION\ s\ t) \leq HOL\_Light\_Import.measure\ s + HOL\_Light\_Import.measure\ t$

thm MEASURE_UNIONS_LE:

$\forall f$::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE f* ∧ (∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN s f* ⟶ *measurable s*) ⟶ *HOL_Light_Import.measure* (*UNIONS f*) ≤ *sum f HOL_Light_Import.measure*

thm MEASURE_UNIONS_LE_IMAGE:

∀ (*f*::*?'b*::*type* ⇒ *bool*) *s*::*?'b*::*type* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE f* ∧ (∀ *a*::*?'b*::*type*. *IN a f* ⟶ *measurable* (*s a*)) ⟶ *HOL_Light_Import.measure* (*UNIONS* (*IMAGE s f*)) ≤ *sum f* (λ*a*::*?'b*::*type*. *HOL_Light_Import.measure* (*s a*))

thm MEASURABLE_INNER_OUTER:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* = (∀ *e*>*0*::*real*. ∃ (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *u*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *SUBSET s u* ∧ *measurable t* ∧ *measurable u* ∧ |*HOL_Light_Import.measure t* − *HOL_Light_Import.measure u*| < *e*)

thm HAS_MEASURE_INNER_OUTER:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *has_measure s m* = ((∀ *e*>*0*::*real*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *measurable t* ∧ *m* − *e* < *HOL_Light_Import.measure t*) ∧ (∀ *e*>*0*::*real*. ∃ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s u* ∧ *measurable u* ∧ *HOL_Light_Import.measure u* < *m* + *e*))

thm HAS_MEASURE_INNER_OUTER_LE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *m*::*real*. *has_measure s m* = ((∀ *e*>*0*::*real*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t s* ∧ *measurable t* ∧ *m* − *e* ≤ *HOL_Light_Import.measure t*) ∧ (∀ *e*>*0*::*real*. ∃ *u*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s u* ∧ *measurable u* ∧ *HOL_Light_Import.measure u* ≤ *m* + *e*))

thm NEGLIGIBLE_OUTER:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* = (∀ *e*>*0*::*real*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ∧ *measurable t* ∧ *HOL_Light_Import.measure t* < *e*)

thm NEGLIGIBLE_OUTER_LE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *negligible s* = (∀ *e*>*0*::*real*. ∃ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s t* ∧ *measurable t* ∧ *HOL_Light_Import.measure t* ≤ *e*)

thm HAS_MEASURE_LIMIT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *has_measure s* (*?m*::*real*) = (∀ *e*>*0*::*real*. ∃ *B*>*0*::*real*. ∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*ball* (*vec* (*0*::*nat*), *B*)) (*closed_interval* [(*a*, *b*)]) ⟶ (∃ *z*::*real*. *has_measure* (*HOL_Light_Import.INTER s* (*closed_interval* [(*a*, *b*)])) *z* ∧ |*z* − *?m*| < *e*))

thm MEASURE_LIMIT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge (0::real) < e \longrightarrow (\exists B>0::real. \forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ SUBSET\ (ball\ (vec\ (0::nat),\ B))\ (closed\_interval\ [(a,\ b)]) \longrightarrow |HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ s\ (closed\_interval\ [(a,\ b)])) - HOL\_Light\_Import.measure\ s| < e)$

thm INTEGRABLE_ON_CONST:

$\forall c::(real,\ ?'b::type)\ cart.\ integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ c)\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) = (c = vec\ (0::nat) \vee measurable\ ?s)$

thm ABSOLUTELY_INTEGRABLE_ON_CONST:

$\forall c::(real,\ ?'b::type)\ cart.\ absolutely\_integrable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ c)\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool) = (c = vec\ (0::nat) \vee measurable\ ?s)$

thm NEGLIGIBLE_INTERVAL_conjunct1:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ negligible\ (open\_interval\ (a,\ b)) = (open\_interval\ (a,\ b) = EMPTY)$

thm NEGLIGIBLE_INTERVAL_conjunct0:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ negligible\ (closed\_interval\ [(a,\ b)]) = (open\_interval\ (a,\ b) = EMPTY)$

thm OPEN_NOT_NEGLIGIBLE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge s \neq EMPTY \longrightarrow \neg\ negligible\ s$

thm HAS_MEASURE_AFFINITY:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::real)\ (c::(real,\ ?'a::type)\ cart)\ y::real.\ has\_measure\ s\ y \longrightarrow has\_measure\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ vector\_add\ (\%\ m\ x)\ c)\ s)\ (|m|^{dimindex\ HOL\_Light\_Import.UNIV} * y)$

thm STRETCH_GALOIS:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ m::nat \Rightarrow real.\ (\forall k::nat.\ (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow m\ k \neq (0::real)) \longrightarrow (y = lambda\ (\lambda k::nat.\ m\ k * \$\ x\ k)) = (lambda\ (\lambda k::nat.\ inverse\_class.inverse\ (m\ k) * \$\ y\ k) = x)$

thm HAS_MEASURE_STRETCH:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::nat \Rightarrow real)\ y::real.\ has\_measure\ s\ y \longrightarrow has\_measure\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda k::nat.\ m\ k * \$\ x\ k))\ s)\ (|product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ m| * y)$

thm HAS_MEASURE_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::real)\ a::(real,\ ?'a::type)\ cart.\ has\_measure\ s\ m \longrightarrow has\_measure\ (IMAGE\ (vector\_add\ a)\ s)\ m$

thm NEGLIGIBLE_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ negligible\ s \longrightarrow negligible\ (IMAGE\ (vector\_add\ a)\ s)$

thm HAS_MEASURE_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ m::real.\ has\_measure\ (IMAGE\ (vector\_add\ a)\ s)\ m = has\_measure\ s\ m$

thm MEASURE_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.measure\ (IMAGE\ (vector\_add\ a)\ s) = HOL\_Light\_Import.measure\ s$

thm NEGLIGIBLE_TRANSLATION_REV:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ negligible\ (IMAGE\ (vector\_add\ a)\ s) \longrightarrow negligible\ s$

thm NEGLIGIBLE_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ (IMAGE\ (vector\_add\ a)\ s) = negligible\ s$

thm MEASURABLE_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ measurable\ (IMAGE\ (vector\_add\ a)\ s) = measurable\ s$

thm MEASURABLE_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ measurable\ s \longrightarrow measurable\ (IMAGE\ (vector\_add\ a)\ s)$

thm HAS_MEASURE_SCALING:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::real)\ c::real.\ has\_measure\ s\ m \longrightarrow has\_measure\ (IMAGE\ (\%\ c)\ s)\ (|c|^{dimindex\ HOL\_Light\_Import.UNIV} * m)$

thm HAS_MEASURE_SCALING_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (m::real)\ c::real.\ c \neq (0::real) \longrightarrow has\_measure\ (IMAGE\ (\%\ c)\ s)\ (|c|^{dimindex\ HOL\_Light\_Import.UNIV} * m) = has\_measure\ s\ m$

thm MEASURABLE_SCALING:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ measurable\ s \longrightarrow measurable\ (IMAGE\ (\%\ c)\ s)$

thm MEASURABLE_SCALING_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ c::real.\ c \neq (0::real) \longrightarrow measurable\ (IMAGE\ (\%\ c)\ s) = measurable\ s$

thm MEASURE_SCALING:

∀ s::(real, ?'a::type) cart ⇒ bool. measurable s ⟶ HOL_Light_Import.measure (IMAGE (% (?c::real)) s) = |?c|$^{dimindex\ HOL\_Light\_Import.UNIV}$ ∗ HOL_Light_Import.measure s

thm HAS_MEASURE_NESTED_UNIONS:

∀ (s::nat ⇒ (real, ?'a::type) cart ⇒ bool) B::real. (∀ n::nat. measurable (s n)) ∧ (∀ n::nat. HOL_Light_Import.measure (s n) ≤ B) ∧ (∀ n::nat. SUBSET (s n) (s (Suc n))) ⟶ measurable (UNIONS (GSPEC (λGEN%PVAR%2109::(real, ?'a::type) cart ⇒ bool. ∃ n::nat. SETSPEC GEN%PVAR%2109 (IN n HOL_Light_Import.UNIV) (s n)))) ∧ −−> (λn::nat. lift (HOL_Light_Import.measure (s n))) (lift (HOL_Light_Import.measure (UNIONS (GSPEC (λGEN%PVAR%2110::(real, ?'a::type) cart ⇒ bool. ∃ n::nat. SETSPEC GEN%PVAR%2110 (IN n HOL_Light_Import.UNIV) (s n)))))) sequentially

thm MEASURABLE_NESTED_UNIONS:

∀ (s::nat ⇒ (real, ?'a::type) cart ⇒ bool) B::real. (∀ n::nat. measurable (s n)) ∧ (∀ n::nat. HOL_Light_Import.measure (s n) ≤ B) ∧ (∀ n::nat. SUBSET (s n) (s (Suc n))) ⟶ measurable (UNIONS (GSPEC (λGEN%PVAR%2111::(real, ?'a::type) cart ⇒ bool. ∃ n::nat. SETSPEC GEN%PVAR%2111 (IN n HOL_Light_Import.UNIV) (s n))))

thm HAS_MEASURE_COUNTABLE_NEGLIGIBLE_UNIONS:

∀ (s::nat ⇒ (real, ?'a::type) cart ⇒ bool) B::real. (∀ n::nat. measurable (s n)) ∧ (∀ (m::nat) n::nat. m ≠ n ⟶ negligible (HOL_Light_Import.INTER (s m) (s n))) ∧ (∀ n::nat. sum (dotdot (0::nat) n) (λk::nat. HOL_Light_Import.measure (s k)) ≤ B) ⟶ measurable (UNIONS (GSPEC (λGEN%PVAR%2113::(real, ?'a::type) cart ⇒ bool. ∃ n::nat. SETSPEC GEN%PVAR%2113 (IN n HOL_Light_Import.UNIV) (s n)))) ∧ sums (λn::nat. lift (HOL_Light_Import.measure (s n))) (lift (HOL_Light_Import.measure (UNIONS (GSPEC (λGEN%PVAR%2114::(real, ?'a::type) cart ⇒ bool. ∃ n::nat. SETSPEC GEN%PVAR%2114 (IN n HOL_Light_Import.UNIV) (s n)))))) (from (0::nat))

thm NEGLIGIBLE_COUNTABLE_UNIONS_GEN:

∀ f::((real, ?'a::type) cart ⇒ bool) ⇒ bool. COUNTABLE f ∧ (∀ s::(real, ?'a::type) cart ⇒ bool. IN s f ⟶ negligible s) ⟶ negligible (UNIONS f)

thm MEASURABLE_INTERVAL_conjunct1:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. measurable (open_interval (a, b))

thm MEASURABLE_INTERVAL_conjunct0:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. measurable (closed_interval [(a, b)])

thm HAS_MEASURE_COUNTABLE_NEGLIGIBLE_UNIONS_BOUNDED:

∀ s::nat ⇒ (real, ?'a::type) cart ⇒ bool. (∀ n::nat. measurable (s n)) ∧ (∀ (m::nat) n::nat. m ≠ n ⟶ negligible (HOL_Light_Import.INTER (s m) (s n))) ∧

bounded (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2115*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2115* (*IN n HOL_Light_Import.UNIV*) (*s n*)))) $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2116*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2116* (*IN n HOL_Light_Import.UNIV*) (*s n*)))) $\wedge$ *sums* (λ*n*::*nat*. *lift* (*HOL_Light_Import.measure* (*s n*))) (*lift* (*HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2117*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2117* (*IN n HOL_Light_Import.UNIV*) (*s n*)))))) (*from* (*0*::*nat*))

thm MEASURABLE_COUNTABLE_UNIONS_BOUNDED:

$\forall$ *s*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *n*::*nat*. *measurable* (*s n*)) $\wedge$ *bounded* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2122*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2122* (*IN n HOL_Light_Import.UNIV*) (*s n*)))) $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2123*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2123* (*IN n HOL_Light_Import.UNIV*) (*s n*))))

thm MEASURE_COUNTABLE_UNIONS_LE_STRONG:

$\forall$ (*d*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *B*::*real*. ($\forall$ *n*::*nat*. *measurable* (*d n*)) $\wedge$ ($\forall$ *n*::*nat*. *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2130*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *k*::*nat*. *SETSPEC GEN%PVAR%2130* ($k \leq n$) (*d k*)))) $\leq$ *B*) $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2131*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2131* (*IN n HOL_Light_Import.UNIV*) (*d n*)))) $\wedge$ *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2132*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2132* (*IN n HOL_Light_Import.UNIV*) (*d n*)))) $\leq$ *B*

thm MEASURE_COUNTABLE_UNIONS_LE:

$\forall$ (*d*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *B*::*real*. ($\forall$ *n*::*nat*. *measurable* (*d n*)) $\wedge$ ($\forall$ *n*::*nat*. *sum* (*dotdot* (*0*::*nat*) *n*) (λ*k*::*nat*. *HOL_Light_Import.measure* (*d k*)) $\leq$ *B*) $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2133*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2133* (*IN n HOL_Light_Import.UNIV*) (*d n*)))) $\wedge$ *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2134*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2134* (*IN n HOL_Light_Import.UNIV*) (*d n*)))) $\leq$ *B*

thm MEASURABLE_COUNTABLE_UNIONS_STRONG:

$\forall$ (*s*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *B*::*real*. ($\forall$ *n*::*nat*. *measurable* (*s n*)) $\wedge$ ($\forall$ *n*::*nat*. *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2135*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *k*::*nat*. *SETSPEC GEN%PVAR%2135* ($k \leq n$) (*s k*)))) $\leq$ *B*) $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2136*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *n*::*nat*. *SETSPEC GEN%PVAR%2136* (*IN n HOL_Light_Import.UNIV*) (*s n*))))

thm MEASURABLE_COUNTABLE_UNIONS:

$\forall$ (*s*::*nat* $\Rightarrow$ (*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *B*::*real*. ($\forall$ *n*::*nat*. *measurable* (*s n*)) $\wedge$ ($\forall$ *n*::*nat*. *sum* (*dotdot* (*0*::*nat*) *n*) (λ*k*::*nat*. *HOL_Light_Import.measure* (*s* *k*)) ≤ *B*) ⟶ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2137*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%2137* (*IN n HOL_Light_Import*.*UNIV*) (*s n*))))

thm MEASURE_COUNTABLE_UNIONS_LE_STRONG_GEN:

∀ (*D*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *B*::*real*. *COUNTABLE D* ∧ (∀ *d*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN d D* ⟶ *measurable d*) ∧ (∀ *D'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET D' D* ∧ *FINITE D'* ⟶ *HOL_Light_Import*.*measure* (*UNIONS D'*) ≤ *B*) ⟶ *measurable* (*UNIONS D*) ∧ *HOL_Light_Import*.*measure* (*UNIONS D*) ≤ *B*

thm MEASURE_COUNTABLE_UNIONS_LE_GEN:

∀ (*D*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *B*::*real*. *COUNTABLE D* ∧ (∀ *d*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN d D* ⟶ *measurable d*) ∧ (∀ *D'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET D' D* ∧ *FINITE D'* ⟶ *sum D' HOL_Light_Import*.*measure* ≤ *B*) ⟶ *measurable* (*UNIONS D*) ∧ *HOL_Light_Import*.*measure* (*UNIONS D*) ≤ *B*

thm MEASURABLE_COUNTABLE_INTERS:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *measurable* (*s n*)) ⟶ *measurable* (*INTERS* (*GSPEC* (λ*GEN%PVAR%2141*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%2141* (*IN n HOL_Light_Import*.*UNIV*) (*s n*))))

thm MEASURE_COUNTABLE_UNIONS_APPROACHABLE:

∀ (*D*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*B*::*real*) *e*::*real*. *COUNTABLE D* ∧ (∀ *d*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN d D* ⟶ *measurable d*) ∧ (∀ *D'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET D' D* ∧ *FINITE D'* ⟶ *HOL_Light_Import*.*measure* (*UNIONS D'*) ≤ *B*) ∧ (*0*::*real*) < *e* ⟶ (∃ *D'*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *SUBSET D' D* ∧ *FINITE D'* ∧ *HOL_Light_Import*.*measure* (*UNIONS D*) − *e* < *HOL_Light_Import*.*measure* (*UNIONS D'*))

thm HAS_MEASURE_NESTED_INTERS:

∀ *s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *n*::*nat*. *measurable* (*s n*)) ∧ (∀ *n*::*nat*. *SUBSET* (*s* (*Suc n*)) (*s n*)) ⟶ *measurable* (*INTERS* (*GSPEC* (λ*GEN%PVAR%2150*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%2150* (*IN n HOL_Light_Import*.*UNIV*) (*s n*)))) ∧ −−> (λ*n*::*nat*. *lift* (*HOL_Light_Import*.*measure* (*s n*))) (*lift* (*HOL_Light_Import*.*measure* (*INTERS* (*GSPEC* (λ*GEN%PVAR%2151*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%2151* (*IN n HOL_Light_Import*.*UNIV*) (*s n*)))))) *sequentially*

thm MEASURABLE_COMPACT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ⟶ *measurable s*

thm MEASURABLE_OPEN:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *HOL_Light_Import*.*open s* ⟶ *measurable s*

thm MEASURE_OPEN_POS_LT:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *bounded s* ∧ *s* ≠ *EMPTY* ⟶ (*0*::*real*) < *HOL_Light_Import.measure s*

thm MEASURABLE_CLOSURE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *measurable* (*closure s*)

thm MEASURABLE_INTERIOR:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *measurable* (*interior s*)

thm MEASURABLE_FRONTIER:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *measurable* (*frontier s*)

thm MEASURE_FRONTIER:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ⟶ *HOL_Light_Import.measure* (*frontier s*) = *HOL_Light_Import.measure* (*closure s*) − *HOL_Light_Import.measure* (*interior s*)

thm MEASURE_CLOSURE:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *negligible* (*frontier s*) ⟶ *HOL_Light_Import.measure* (*closure s*) = *HOL_Light_Import.measure s*

thm MEASURE_INTERIOR:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *negligible* (*frontier s*) ⟶ *HOL_Light_Import.measure* (*interior s*) = *HOL_Light_Import.measure s*

thm MEASURABLE_JORDAN:

$\forall s$::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *bounded s* ∧ *negligible* (*frontier s*) ⟶ *measurable s*

thm HAS_MEASURE_ELEMENTARY:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *division_of d s* ⟶ *has_measure s* (*sum d content*)

thm MEASURABLE_ELEMENTARY:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *division_of d s* ⟶ *measurable s*

thm MEASURE_ELEMENTARY:

∀ (*d*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *division_of d s* ⟶ *HOL_Light_Import.measure s* = *sum d content*

thm MEASURABLE_INTER_INTERVAL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *measurable s* ⟶ *measurable* (*HOL_Light_Import.INTER s* (*closed_interval* [(*a*, *b*)]))

thm MEASURABLE_INSIDE:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. compact s ⟶ measurable (inside s)*

thm STARLIKE_NEGLIGIBLE_BOUNDED_MEASURABLE:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. measurable s ∧ bounded s ∧ (∀ (c::real) x::(real, ?'a::type) cart. (0::real) ≤ c ∧ IN x s ∧ IN (% c x) s ⟶ c = (1::real)) ⟶ negligible s*

thm STARLIKE_NEGLIGIBLE_LEMMA:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. compact s ∧ (∀ (c::real) x::(real, ?'a::type) cart. (0::real) ≤ c ∧ IN x s ∧ IN (% c x) s ⟶ c = (1::real)) ⟶ negligible s*

thm STARLIKE_NEGLIGIBLE:

$\forall$ *(s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. HOL_Light_Import.closed s ∧ (∀ (c::real) x::(real, ?'a::type) cart. (0::real) ≤ c ∧ IN (vector_add a x) s ∧ IN (vector_add a (% c x)) s ⟶ c = (1::real)) ⟶ negligible s*

thm STARLIKE_NEGLIGIBLE_STRONG:

$\forall$ *(s::(real, ?'a::type) cart ⇒ bool) a::(real, ?'a::type) cart. HOL_Light_Import.closed s ∧ (∀ (c::real) x::(real, ?'a::type) cart. (0::real) ≤ c ∧ c < (1::real) ∧ IN (vector_add a x) s ⟶ ¬ IN (vector_add a (% c x)) s) ⟶ negligible s*

thm NEGLIGIBLE_HYPERPLANE:

$\forall$ *(a::(real, ?'a::type) cart) b::real. ¬ (a = vec (0::nat) ∧ b = (0::real)) ⟶ negligible (GSPEC (λGEN%PVAR%2158::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2158 (dot a x = b) x))*

thm NEGLIGIBLE_LOWDIM:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. dim s < dimindex HOL_Light_Import.UNIV ⟶ negligible s*

thm NEGLIGIBLE_AFFINE_HULL:

$\forall$ *s::(real, ?'a::type) cart ⇒ bool. FINITE s ∧ CARD s ≤ dimindex HOL_Light_Import.UNIV ⟶ negligible (hull affine s)*

thm NEGLIGIBLE_AFFINE_HULL_1:

$\forall$ *a::(real, unit) cart. negligible (hull affine (INSERT a EMPTY))*

thm NEGLIGIBLE_AFFINE_HULL_2:

$\forall$ *(a::(real, 2) cart) b::(real, 2) cart. negligible (hull affine (INSERT a (INSERT b EMPTY)))*

thm NEGLIGIBLE_AFFINE_HULL_3:

$\forall$ *(a::(real, 3) cart) (b::(real, 3) cart) c::(real, 3) cart. negligible (hull affine (INSERT a (INSERT b (INSERT c EMPTY))))*

thm NEGLIGIBLE_CONVEX_HULL:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ FINITE\ s \wedge CARD\ s \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow negligible\ (hull\ convex\ s)$

thm NEGLIGIBLE_CONVEX_HULL_1:

$\forall\, a{::}(real,\ unit)\ cart.\ negligible\ (hull\ convex\ (INSERT\ a\ EMPTY))$

thm NEGLIGIBLE_CONVEX_HULL_2:

$\forall\, (a{::}(real,\ 2)\ cart)\ b{::}(real,\ 2)\ cart.\ negligible\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ EMPTY)))$

thm NEGLIGIBLE_CONVEX_HULL_3:

$\forall\, (a{::}(real,\ 3)\ cart)\ (b{::}(real,\ 3)\ cart)\ c{::}(real,\ 3)\ cart.\ negligible\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))))$

thm NEGLIGIBLE_CONVEX_FRONTIER:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow negligible\ (frontier\ s)$

thm MEASURABLE_CONVEX:

$\forall\, s{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ convex\ s \wedge bounded\ s \longrightarrow measurable\ s$

thm NEGLIGIBLE_SPHERE:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ r{::}real.\ negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2160{::}(real,\ ?'a{::}type)\ cart.\ \exists\, x{::}(real,\ ?'a{::}type)\ cart.\ SETSPEC\ GEN\%PVAR\%2160\ (distance\ (a,\ x) = r)\ x))$

thm MEASURABLE_BALL:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ r{::}real.\ measurable\ (ball\ (a,\ r))$

thm MEASURABLE_CBALL:

$\forall\, (a{::}(real,\ ?'a{::}type)\ cart)\ r{::}real.\ measurable\ (cball\ (a,\ r))$

thm MEASURE_BALL_POS:

$\forall\, (x{::}(real,\ ?'a{::}type)\ cart)\ e{::}real.\ (0{::}real) < e \longrightarrow (0{::}real) < HOL\_Light\_Import.measure\ (ball\ (x,\ e))$

thm MEASURE_CBALL_POS:

$\forall\, (x{::}(real,\ ?'a{::}type)\ cart)\ e{::}real.\ (0{::}real) < e \longrightarrow (0{::}real) < HOL\_Light\_Import.measure\ (cball\ (x,\ e))$

thm HAS_INTEGRAL_OPEN_INTERVAL:

$\forall\, (f{::}(real,\ ?'b{::}type)\ cart \Rightarrow (real,\ ?'a{::}type)\ cart)\ (a{::}(real,\ ?'b{::}type)\ cart)\ (b{::}(real,\ ?'b{::}type)\ cart)\ y{::}(real,\ ?'a{::}type)\ cart.\ has\_integral\ f\ y\ (open\_interval\ (a,\ b)) = has\_integral\ f\ y\ (closed\_interval\ [(a,\ b)])$

thm INTEGRABLE_ON_OPEN_INTERVAL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ integrable\_on\ f\ (open\_interval\ (a,\ b)) = integrable\_on\ f\ (closed\_interval\ [(a,\ b)])$

thm INTEGRAL_OPEN_INTERVAL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ integral\ (open\_interval\ (a,\ b))\ f = integral\ (closed\_interval\ [(a,\ b)])\ f$

thm NEGLIGIBLE_LINEAR_SINGULAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge \neg (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow negligible\ (IMAGE\ f\ s)$

thm COVERING_LEMMA:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ (closed\_interval\ [(a,\ b)]) \wedge open\_interval\ (a,\ b) \neq EMPTY \wedge gauge\ g \longrightarrow (\exists d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow SUBSET\ k\ (closed\_interval\ [(a,\ b)]) \wedge interior\ k \neq EMPTY \wedge (\exists (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(c,\ d)])) \wedge (\forall (k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k1\ d \wedge IN\ k2\ d \wedge k1 \neq k2 \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k1)\ (interior\ k2) = EMPTY) \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow (\exists x::(real,\ ?'a::type)\ cart.\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ k) \wedge SUBSET\ k\ (g\ x))) \wedge (\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ IN\ (closed\_interval\ [(u,\ v)])\ d \longrightarrow (\exists n::nat.\ \forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ v\ i - \$\ u\ i = (\$\ b\ i - \$\ a\ i)\ /\ (real\_of\_nat\ (2::nat))^{n})) \wedge SUBSET\ s\ (UNIONS\ d))$

thm Hypermap.LT_PLUS:

$\forall n::nat.\ n < Suc\ n$

thm COUNTABLE_ELEMENTARY_DIVISION:

$\forall d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \longrightarrow (\exists d'::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d' \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d' \longrightarrow k \neq EMPTY \wedge (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \wedge (\forall (k::(real,\ ?'a::type)\ cart \Rightarrow bool)\ l::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d' \wedge IN\ l\ d' \wedge k \neq l \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k)\ (interior\ l) = EMPTY) \wedge UNIONS\ d' = UNIONS\ d)$

thm EXPAND_CLOSED_OPEN_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ e::real.\ (0::real) < e \longrightarrow (\exists (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ SUBSET\ (closed\_interval$

$[(a,\ b)])$ $(open\_interval\ (c,\ d)) \wedge HOL\_Light\_Import.measure\ (open\_interval\ (c,\ d)) \leq HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]) + e)$

thm MEASURABLE_OUTER_INTERVALS_BOUNDED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ e::real.\ measurable\ s \wedge SUBSET\ s\ (closed\_interval\ [(a,\ b)]) \wedge (0::real) < e \longrightarrow (\exists d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow SUBSET\ k\ (closed\_interval\ [(a,\ b)]) \wedge k \neq EMPTY \wedge (\exists (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(c,\ d)])) \wedge (\forall (k1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ k2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k1\ d \wedge IN\ k2\ d \wedge k1 \neq k2 \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k1)\ (interior\ k2) = EMPTY) \wedge (\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ IN\ (closed\_interval\ [(u,\ v)])\ d \longrightarrow (\exists n::nat.\ \forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ v\ i - \$\ u\ i = (\$\ b\ i - \$\ a\ i)\ /\ (real\_of\_nat\ (2::nat))^n)) \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \wedge open\_interval\ (a,\ b) \neq EMPTY \longrightarrow interior\ k \neq EMPTY) \wedge SUBSET\ s\ (UNIONS\ d) \wedge measurable\ (UNIONS\ d) \wedge HOL\_Light\_Import.measure\ (UNIONS\ d) \leq HOL\_Light\_Import.measure\ s + e)$

thm MEASURABLE_OUTER_CLOSED_INTERVALS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge (0::real) < e \longrightarrow (\exists d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow k \neq EMPTY \wedge (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = closed\_interval\ [(a,\ b)])) \wedge (\forall (k::(real,\ ?'a::type)\ cart \Rightarrow bool)\ l::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \wedge IN\ l\ d \wedge k \neq l \longrightarrow HOL\_Light\_Import.INTER\ (interior\ k)\ (interior\ l) = EMPTY) \wedge SUBSET\ s\ (UNIONS\ d) \wedge measurable\ (UNIONS\ d) \wedge HOL\_Light\_Import.measure\ (UNIONS\ d) \leq HOL\_Light\_Import.measure\ s + e)$

thm MEASURABLE_OUTER_OPEN_INTERVALS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge (0::real) < e \longrightarrow (\exists d::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ d \wedge (\forall k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ k\ d \longrightarrow k \neq EMPTY \wedge (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ k = open\_interval\ (a,\ b))) \wedge SUBSET\ s\ (UNIONS\ d) \wedge measurable\ (UNIONS\ d) \wedge HOL\_Light\_Import.measure\ (UNIONS\ d) \leq HOL\_Light\_Import.measure\ s + e)$

thm MEASURABLE_OUTER_OPEN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge (0::real) < e \longrightarrow (\exists t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge SUBSET\ s\ t \wedge measurable\ t \wedge HOL\_Light\_Import.measure\ t < HOL\_Light\_Import.measure\ s + e)$

thm MEASURABLE_INNER_COMPACT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge (0::real) < e \longrightarrow (\exists t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ t \wedge SUBSET\ t\ s \wedge measurable\ t \wedge HOL\_Light\_Import.measure\ s < HOL\_Light\_Import.measure\ t + e)$

thm OPEN_MEASURABLE_INNER_DIVISION:

∀ (s::(real, ?'a::type) cart ⇒ bool) e::real. HOL_Light_Import.open s ∧ measurable s ∧ (0::real) < e ⟶ (∃ D::((real, ?'a::type) cart ⇒ bool) ⇒ bool. division_of D (UNIONS D) ∧ SUBSET (UNIONS D) s ∧ HOL_Light_Import.measure s < HOL_Light_Import.measure (UNIONS D) + e)

thm MEASURABLE_LINEAR_IMAGE_INTERVAL:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. linear f ⟶ measurable (IMAGE f (closed_interval [(a, b)]))

thm HAS_MEASURE_LINEAR_SUFFICIENT:

∀ (f::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart) m::real. linear f ∧ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. has_measure (IMAGE f (closed_interval [(a, b)])) (m ∗ HOL_Light_Import.measure (closed_interval [(a, b)]))) ⟶ (∀ s::(real, ?'a::type) cart ⇒ bool. measurable s ⟶ has_measure (IMAGE f s) (m ∗ HOL_Light_Import.measure s))

thm INDUCT_MATRIX_ROW_OPERATIONS:

∀ P::((real, ?'a::type) cart, ?'a::type) cart ⇒ bool. (∀ (A::((real, ?'a::type) cart, ?'a::type) cart) i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ row i A = vec (0::nat) ⟶ P A) ∧ (∀ A::((real, ?'a::type) cart, ?'a::type) cart. (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ i ≠ j ⟶ $ ($ A i) j = (0::real)) ⟶ P A) ∧ (∀ (A::((real, ?'a::type) cart, ?'a::type) cart) (m::nat) n::nat. P A ∧ (1::nat) ≤ m ∧ m ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ n ∧ n ≤ dimindex HOL_Light_Import.UNIV ∧ m ≠ n ⟶ P (lambda (λi::nat. lambda (λj::nat. $ ($ A i) (swap (m, n) j))))) ∧ (∀ (A::((real, ?'a::type) cart, ?'a::type) cart) (m::nat) (n::nat) c::real. P A ∧ (1::nat) ≤ m ∧ m ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ n ∧ n ≤ dimindex HOL_Light_Import.UNIV ∧ m ≠ n ⟶ P (lambda (λi::nat. if i = m then vector_add (row m A) (% c (row n A)) else row i A))) ⟶ (∀ A::((real, ?'a::type) cart, ?'a::type) cart. P A)

thm INDUCT_MATRIX_ELEMENTARY:

∀ P::((real, ?'a::type) cart, ?'a::type) cart ⇒ bool. (∀ (A::((real, ?'a::type) cart, ?'a::type) cart) B::((real, ?'a::type) cart, ?'a::type) cart. P A ∧ P B ⟶ P (matrix_mul A B)) ∧ (∀ (A::((real, ?'a::type) cart, ?'a::type) cart) i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ row i A = vec (0::nat) ⟶ P A) ∧ (∀ A::((real, ?'a::type) cart, ?'a::type) cart. (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ i ≠ j ⟶ $ ($ A i) j = (0::real)) ⟶ P A) ∧ (∀ (m::nat) n::nat. (1::nat) ≤ m ∧ m ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ n ∧ n ≤ dimindex HOL_Light_Import.UNIV ∧ m ≠ n ⟶ P (lambda (λi::nat. lambda (λj::nat. $ ($ (mat (1::nat)) i) (swap

$(m,\ n)\ j))))) \wedge (\forall (m::nat)\ (n::nat)\ c::real.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \longrightarrow P\ (lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ if\ i = m \wedge j = n\ then\ c\ else\ if\ i = j\ then\ 1::real\ else\ (0::real))))) \longrightarrow (\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ P\ A)$

thm INDUCT_MATRIX_ELEMENTARY_ALT:

$\forall P::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ B::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ P\ A \wedge P\ B \longrightarrow P\ (matrix\_mul\ A\ B)) \wedge (\forall (A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ A = vec\ (0::nat) \longrightarrow P\ A) \wedge (\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge i \neq j \longrightarrow \$\ (\$\ A\ i)\ j = (0::real)) \longrightarrow P\ A) \wedge (\forall (m::nat)\ n::nat.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \longrightarrow P\ (lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ \$\ (\$\ (mat\ (1::nat))\ i)\ (swap\ (m,\ n)\ j))))) \wedge (\forall (m::nat)\ n::nat.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \longrightarrow P\ (lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ if\ i = m \wedge j = n \vee i = j\ then\ 1::real\ else\ (0::real))))) \longrightarrow (\forall A::((real,\ ?'a::type)\ cart,\ ?'a::type)\ cart.\ P\ A)$

thm INDUCT_LINEAR_ELEMENTARY:

$\forall P::((real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart) \Rightarrow bool.\ (\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \wedge linear\ g \wedge P\ f \wedge P\ g \longrightarrow P\ (f \circ g)) \wedge (\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ i::nat.\ linear\ f \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'a::type)\ cart.\ \$\ (f\ x)\ i = (0::real)) \longrightarrow P\ f) \wedge (\forall c::nat \Rightarrow real.\ P\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ c\ i * \$\ x\ i))) \wedge (\forall (m::nat)\ n::nat.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \longrightarrow P\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ \$\ x\ (swap\ (m,\ n)\ i)))) \wedge (\forall (m::nat)\ n::nat.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \longrightarrow P\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ if\ i = m\ then\ \$\ x\ m + \$\ x\ n\ else\ \$\ x\ i))) \longrightarrow (\forall f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ linear\ f \longrightarrow P\ f)$

thm LAMBDA_SWAP_GALOIS:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (1::nat) \leq (?m::nat) \wedge ?m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?n::nat) \wedge ?n \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (x = lambda\ (\lambda i::nat.\ \$\ y\ (swap\ (?m,\ ?n)\ i))) = (lambda\ (\lambda i::nat.\ \$\ x\ (swap\ (?m,\ ?n)\ i)) = y)$

thm LAMBDA_ADD_GALOIS:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (1::nat) \leq (?m::nat) \wedge ?m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?n::nat) \wedge ?n \leq dimindex$

$HOL\_Light\_Import.UNIV \wedge ?m \neq ?n \longrightarrow (x = lambda\ (\lambda i::nat.\ if\ i = ?m\ then\ \$\ y\ ?m + \$\ y\ ?n\ else\ \$\ y\ i)) = (lambda\ (\lambda i::nat.\ if\ i = ?m\ then\ \$\ x\ ?m - \$\ x\ ?n\ else\ \$\ x\ i) = y)$

thm HAS_MEASURE_SHEAR_INTERVAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (m::nat)\ n::nat.\ (1::nat) \leq m \wedge m \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq n \wedge n \leq dimindex\ HOL\_Light\_Import.UNIV \wedge m \neq n \wedge closed\_interval\ [(a,\ b)] \neq EMPTY \wedge (0::real) \leq \$\ a\ n \longrightarrow has\_measure\ (IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ if\ i = m\ then\ \$\ x\ m + \$\ x\ n\ else\ \$\ x\ i))\ (closed\_interval\ [(a,\ b)]))\ (HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]))$

thm HAS_MEASURE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge measurable\ s \longrightarrow has\_measure\ (IMAGE\ f\ s)\ (|det\ (matrix\ f)| * HOL\_Light\_Import.measure\ s)$

thm MEASURABLE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge measurable\ s \longrightarrow measurable\ (IMAGE\ f\ s)$

thm MEASURE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge measurable\ s \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ f\ s) = |det\ (matrix\ f)| * HOL\_Light\_Import.measure\ s$

thm HAS_MEASURE_LINEAR_IMAGE_ALT:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ m::real.\ linear\ f \wedge has\_measure\ s\ m \longrightarrow has\_measure\ (IMAGE\ f\ s)\ (|det\ (matrix\ f)| * m)$

thm HAS_MEASURE_LINEAR_IMAGE_SAME:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge measurable\ s \wedge |det\ (matrix\ f)| = (1::real) \longrightarrow has\_measure\ (IMAGE\ f\ s)\ (HOL\_Light\_Import.measure\ s)$

thm MEASURE_LINEAR_IMAGE_SAME:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge measurable\ s \wedge |det\ (matrix\ f)| = (1::real) \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ f\ s) = HOL\_Light\_Import.measure\ s$

thm MEASURABLE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow measurable\ (IMAGE\ f\ s) = measurable\ s$

thm NEGLIGIBLE_LINEAR_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge negligible\ s \longrightarrow negligible\ (IMAGE\ f\ s)$

thm NEGLIGIBLE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow negligible\ (IMAGE\ f\ s) = negligible\ s$

thm HAS_MEASURE_ORTHOGONAL_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ m::real.\ orthogonal\_transformation\ f \wedge has\_measure\ s\ m \longrightarrow has\_measure\ (IMAGE\ f\ s)\ m$

thm HAS_MEASURE_ORTHOGONAL_IMAGE_EQ:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ m::real.\ orthogonal\_transformation\ f \longrightarrow has\_measure\ (IMAGE\ f\ s)\ m = has\_measure\ s\ m$

thm MEASURE_ORTHOGONAL_IMAGE_EQ:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ orthogonal\_transformation\ f \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ f\ s) = HOL\_Light\_Import.measure\ s$

thm CONGRUENT_IMAGE_STD_SIMPLEX:

$\forall p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%2194::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2194\ ((0::real) \leq \$\ x\ (p\ (1::nat)) \wedge \$\ x\ (p\ (dimindex\ HOL\_Light\_Import.UNIV)) \leq (1::real) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ (p\ i) \leq \$\ x\ (p\ (i + (1::nat)))))\ x) = IMAGE\ (\lambda x::(real,\ ?'a::type)\ cart.\ lambda\ (\lambda i::nat.\ sum\ (dotdot\ (1::nat)\ (HOL\_Light\_Import.inverse\ p\ i))\ (\$\ x)))\ (GSPEC\ (\lambda GEN\%PVAR\%2195::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2195\ ((\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (0::real) \leq \$\ x\ i) \wedge sum\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV))\ (\$\ x) \leq (1::real))\ x))$

thm HAS_MEASURE_IMAGE_STD_SIMPLEX:

$\forall p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV)) \longrightarrow has\_measure\ (GSPEC\ (\lambda GEN\%PVAR\%2196::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2196\ ((0::real) \leq \$\ x\ (p\ (1::nat)) \wedge \$\ x\ (p\ (dimindex\ HOL\_Light\_Import.UNIV)) \leq (1::real) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i < dimindex\ HOL\_Light\_Import.UNIV \longrightarrow \$\ x\ (p\ i) \leq \$\ x\ (p\ (i + (1::nat)))))\ x))\ (HOL\_Light\_Import.measure\ (hull\ convex\ (INSERT\ (vec\ (0::nat))\ (GSPEC\ (\lambda GEN\%PVAR\%2197::(real,\ ?'a::type)\ cart.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%2197\ ((1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV)\ (basis\ i))))))$

thm HAS_MEASURE_STD_SIMPLEX:

has_measure (hull convex (INSERT (vec (0::nat)) (GSPEC (λGEN%PVAR%2201::(real, ?'a::type) cart. ∃ i::nat. SETSPEC GEN%PVAR%2201 ((1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV) (basis i))))) (inverse_class.inverse (real_of_nat (fact (dimindex HOL_Light_Import.UNIV))))

thm HAS_MEASURE_SIMPLEX_0:

∀ l::(real, ?'a::type) cart list. length l = dimindex HOL_Light_Import.UNIV ⟶ has_measure (hull convex (INSERT (vec (0::nat)) (set_of_list l))) (|det (vector l)| / real_of_nat (fact (dimindex HOL_Light_Import.UNIV)))

thm HAS_MEASURE_SIMPLEX:

∀ (a::(real, ?'a::type) cart) l::(real, ?'a::type) cart list. length l = dimindex HOL_Light_Import.UNIV ⟶ has_measure (hull convex (set_of_list (a # l))) (|det (vector (map (λx::(real, ?'a::type) cart. vector_sub x a) l))| / real_of_nat (fact (dimindex HOL_Light_Import.UNIV)))

thm MEASURABLE_CONVEX_HULL:

∀ s::(real, ?'a::type) cart ⇒ bool. bounded s ⟶ measurable (hull convex s)

thm MEASURABLE_SIMPLEX:

∀ l::(real, ?'a::type) cart list. measurable (hull convex (set_of_list l))

thm MEASURE_SIMPLEX:

∀ (a::(real, ?'a::type) cart) l::(real, ?'a::type) cart list. length l = dimindex HOL_Light_Import.UNIV ⟶ HOL_Light_Import.measure (hull convex (set_of_list (a # l))) = |det (vector (map (λx::(real, ?'a::type) cart. vector_sub x a) l))| / real_of_nat (fact (dimindex HOL_Light_Import.UNIV))

thm HAS_MEASURE_TRIANGLE:

∀ (a::(real, 2) cart) (b::(real, 2) cart) c::(real, 2) cart. has_measure (hull convex (INSERT a (INSERT b (INSERT c EMPTY)))) (|($ b (1::nat) − $ a (1::nat)) ∗ ($ c (2::nat) − $ a (2::nat)) − ($ b (2::nat) − $ a (2::nat)) ∗ ($ c (1::nat) − $ a (1::nat))| / real_of_nat (2::nat))

thm MEASURABLE_TRIANGLE:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) c::(real, ?'a::type) cart. measurable (hull convex (INSERT a (INSERT b (INSERT c EMPTY))))

thm MEASURE_TRIANGLE:

∀ (a::(real, 2) cart) (b::(real, 2) cart) c::(real, 2) cart. HOL_Light_Import.measure (hull convex (INSERT a (INSERT b (INSERT c EMPTY)))) = |($ b (1::nat) − $ a (1::nat)) ∗ ($ c (2::nat) − $ a (2::nat)) − ($ b (2::nat) − $ a (2::nat)) ∗ ($ c (1::nat) − $ a (1::nat))| / real_of_nat (2::nat)

thm HAS_MEASURE_TETRAHEDRON:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. has_measure (hull convex (INSERT a (INSERT b (INSERT c (INSERT d EMPTY))))) (|($ b (1::nat) − $ a (1::nat)) ∗ (($ c (2::nat) − $ a (2::nat)) ∗ ($ d (3::nat) − $ a (3::nat))) + (($ b (2::nat) − $ a (2::nat)) ∗ (($ c (3::nat) − $ a (3::nat)) ∗ ($ d (1::nat) − $ a (1::nat))) + (($ b (3::nat) − $ a (3::nat)) ∗ (($ c (1::nat) − $ a (1::nat)) ∗ ($ d (2::nat) − $ a (2::nat))) − ($ b (1::nat) − $ a (1::nat)) ∗ (($ c (3::nat) − $ a (3::nat)) ∗ ($ d (2::nat) − $ a (2::nat))) − ($ b (2::nat) − $ a (2::nat)) ∗ (($ c (1::nat) − $ a (1::nat)) ∗ ($ d (3::nat) − $ a (3::nat))) − ($ b (3::nat) − $ a (3::nat)) ∗ (($ c (2::nat) − $ a (2::nat)) ∗ ($ d (1::nat) − $ a (1::nat)))))| / real_of_nat (6::nat))

thm MEASURABLE_TETRAHEDRON:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (c::(real, ?'a::type) cart) d::(real, ?'a::type) cart. measurable (hull convex (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))))

thm MEASURE_TETRAHEDRON:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. HOL_Light_Import.measure (hull convex (INSERT a (INSERT b (INSERT c (INSERT d EMPTY))))) = |($ b (1::nat) − $ a (1::nat)) ∗ (($ c (2::nat) − $ a (2::nat)) ∗ ($ d (3::nat) − $ a (3::nat))) + (($ b (2::nat) − $ a (2::nat)) ∗ (($ c (3::nat) − $ a (3::nat)) ∗ ($ d (1::nat) − $ a (1::nat))) + (($ b (3::nat) − $ a (3::nat)) ∗ (($ c (1::nat) − $ a (1::nat)) ∗ ($ d (2::nat) − $ a (2::nat))) − ($ b (1::nat) − $ a (1::nat)) ∗ (($ c (3::nat) − $ a (3::nat)) ∗ ($ d (2::nat) − $ a (2::nat))) − ($ b (2::nat) − $ a (2::nat)) ∗ (($ c (1::nat) − $ a (1::nat)) ∗ ($ d (3::nat) − $ a (3::nat))) − ($ b (3::nat) − $ a (3::nat)) ∗ (($ c (2::nat) − $ a (2::nat)) ∗ ($ d (1::nat) − $ a (1::nat)))))| / real_of_nat (6::nat)

thm STEINHAUS:

∀ s::(real, ?'a::type) cart ⇒ bool. measurable s ∧ (0::real) < HOL_Light_Import.measure s ⟶ (∃ d>0::real. SUBSET (ball (vec (0::nat), d)) (GSPEC (λGEN%PVAR%2203::(real, ?'a::type) cart. ∃ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2203 (IN x s ∧ IN y s) (vector_sub x y))))

thm MEASURABLE_NONNEGLIGIBLE_IMP_LARGE:

∀ s::(real, ?'a::type) cart ⇒ bool. measurable s ∧ (0::real) < HOL_Light_Import.measure s ⟶ =_c s HOL_Light_Import.UNIV

thm MEASURABLE_SMALL_IMP_NEGLIGIBLE:

∀ s::(real, ?'a::type) cart ⇒ bool. measurable s ∧ <_c s HOL_Light_Import.UNIV ⟶ negligible s

thm AUSTIN_LEMMA:

∀ D::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FINITE D ∧ (∀ d::(real, ?'a::type) cart ⇒ bool. IN d D ⟶ (∃ (k::real) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. d = closed_interval [(a, b)] ∧ (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex

HOL_Light_Import.UNIV ⟶ $ b i − $ a i = k))) ⟶ (∃ D'::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET D' D ∧ pairwise DISJOINT D' ∧ HOL_Light_Import.measure (UNIONS D) / (real_of_nat (3::nat))^(dimindex HOL_Light_Import.UNIV) ≤ HOL_Light_Import.measure (UNIONS D'))

thm INTEGRABLE_CCONTINUOUS_EXPLICIT:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)])) ⟶ (∃ k::(real, ?'b::type) cart ⇒ bool. negligible k ∧ (∀ (x::(real, ?'b::type) cart) e::real. ¬ IN x k ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ h::real. (0::real) < h ∧ h < d ⟶ vector_norm (vector_sub (% (inverse_class.inverse (content (closed_interval [(x, vector_add x (% h (vec (1::nat))))]))) (integral (closed_interval [(x, vector_add x (% h (vec (1::nat))))]) f)) (f x)) < e)))

thm INTEGRABLE_CCONTINUOUS_EXPLICIT_SYMMETRIC:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)])) ⟶ (∃ k::(real, ?'b::type) cart ⇒ bool. negligible k ∧ (∀ (x::(real, ?'b::type) cart) e::real. ¬ IN x k ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ h::real. (0::real) < h ∧ h < d ⟶ vector_norm (vector_sub (% (inverse_class.inverse (content (closed_interval [(vector_sub x (% h (vec (1::nat))), vector_add x (% h (vec (1::nat))))]))) (integral (closed_interval [(vector_sub x (% h (vec (1::nat))), vector_add x (% h (vec (1::nat))))]) f)) (f x)) < e)))

thm HAS_VECTOR_DERIVATIVE_INDEFINITE_INTEGRAL:

∀ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::(real, unit) cart) b::(real, unit) cart. integrable_on f (closed_interval [(a, b)]) ⟶ (∃ k::(real, unit) cart ⇒ bool. negligible k ∧ (∀ x::(real, unit) cart. IN x (DIFF (closed_interval [(a, b)]) k) ⟶ has_vector_derivative (λx::(real, unit) cart. integral (closed_interval [(a, x)]) f) (f x) (within (at x) (closed_interval [(a, b)]))))

thm ABSOLUTELY_INTEGRABLE_LEBESGUE_POINTS:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. absolutely_integrable_on f (closed_interval [(a, b)])) ⟶ (∃ k::(real, ?'b::type) cart ⇒ bool. negligible k ∧ (∀ (x::(real, ?'b::type) cart) e::real. ¬ IN x k ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ h::real. (0::real) < h ∧ h < d ⟶ vector_norm (% (inverse_class.inverse (content (closed_interval [(vector_sub x (% h (vec (1::nat))), vector_add x (% h (vec (1::nat))))]))) (integral (closed_interval [(vector_sub x (% h (vec (1::nat))), vector_add x (% h (vec (1::nat))))]) (λt::(real, ?'b::type) cart. lift (vector_norm (vector_sub (f t) (f x)))))) < e)))

thm DEF_measurable_on:

measurable_on = (λ(_1856634::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) _1856635::(real, ?'b::type) cart ⇒ bool. ∃ (k::(real, ?'b::type) cart ⇒ bool) g::nat ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart. negligible k ∧ (∀ n::nat.

continuous_on (g n) HOL_Light_Import.UNIV) ∧ (∀ x::(real, ?'b::type) cart. ¬ IN x k ⟶ --> (λn::nat. g n x) (if IN x _1856635 then _1856634 x else vec (0::nat)) sequentially))

thm measurable_on:

∀ (s::(real, ?'b::type) cart ⇒ bool) f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. measurable_on f s = (∃ (k::(real, ?'b::type) cart ⇒ bool) g::nat ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart. negligible k ∧ (∀ n::nat. continuous_on (g n) HOL_Light_Import.UNIV) ∧ (∀ x::(real, ?'b::type) cart. ¬ IN x k ⟶ --> (λn::nat. g n x) (if IN x s then f x else vec (0::nat)) sequentially))

thm MEASURABLE_ON_UNIV:

measurable_on (λx::(real, ?'b::type) cart. if IN x (?s::(real, ?'b::type) cart ⇒ bool) then (?f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) x else vec (0::nat)) HOL_Light_Import.UNIV = measurable_on ?f ?s

thm DEF_lebesgue_measurable:

lebesgue_measurable = (λ_1856646::(real, ?'a::type) cart ⇒ bool. measurable_on (indicator _1856646) HOL_Light_Import.UNIV)

thm lebesgue_measurable:

∀ s::(real, ?'a::type) cart ⇒ bool. lebesgue_measurable s = measurable_on (indicator s) HOL_Light_Import.UNIV

thm MEASURABLE_BOUNDED_BY_INTEGRABLE_IMP_INTEGRABLE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, unit) cart) s::(real, ?'b::type) cart ⇒ bool. measurable_on f s ∧ integrable_on g s ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ vector_norm (f x) ≤ HOL_Light_Import.drop (g x)) ⟶ integrable_on f s

thm MEASURABLE_BOUNDED_BY_INTEGRABLE_IMP_ABSOLUTELY_INTEGRABLE:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, unit) cart) s::(real, ?'b::type) cart ⇒ bool. measurable_on f s ∧ integrable_on g s ∧ (∀ x::(real, ?'b::type) cart. IN x s ⟶ vector_norm (f x) ≤ HOL_Light_Import.drop (g x)) ⟶ absolutely_integrable_on f s

thm INTEGRAL_DROP_LE_MEASURABLE:

∀ (f::(real, ?'a::type) cart ⇒ (real, unit) cart) (g::(real, ?'a::type) cart ⇒ (real, unit) cart) s::(real, ?'a::type) cart ⇒ bool. measurable_on f s ∧ integrable_on g s ∧ (∀ x::(real, ?'a::type) cart. IN x s ⟶ (0::real) ≤ HOL_Light_Import.drop (f x) ∧ HOL_Light_Import.drop (f x) ≤ HOL_Light_Import.drop (g x)) ⟶ HOL_Light_Import.drop (integral s f) ≤ HOL_Light_Import.drop (integral s g)

thm INTEGRABLE_SUBINTERVALS_IMP_MEASURABLE:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. (∀ (a::(real, ?'b::type) cart) b::(real, ?'b::type) cart. integrable_on f (closed_interval [(a, b)])) ⟶ measurable_on f HOL_Light_Import.UNIV

thm INTEGRABLE_IMP_MEASURABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ integrable\_on\ f\ s \longrightarrow measurable\_on\ f\ s$

thm ABSOLUTELY_INTEGRABLE_MEASURABLE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ absolutely\_integrable\_on\ f\ s = (measurable\_on\ f\ s \wedge integrable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (vector\_norm\ (f\ x)))\ s)$

thm MEASURABLE_ON_COMPOSE_CONTINUOUS:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge continuous\_on\ g\ HOL\_Light\_Import.UNIV \longrightarrow measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_ON_COMPOSE_CONTINUOUS_0:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'c::type)\ cart \Rightarrow bool.\ measurable\_on\ f\ s \wedge continuous\_on\ g\ HOL\_Light\_Import.UNIV \wedge g\ (vec\ (0::nat)) = vec\ (0::nat) \longrightarrow measurable\_on\ (g \circ f)\ s$

thm MEASURABLE_ON_COMPOSE_CONTINUOUS_OPEN_INTERVAL:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'c::type)\ cart.\ IN\ (f\ x)\ (open\_interval\ (a,\ b))) \wedge continuous\_on\ g\ (open\_interval\ (a,\ b)) \longrightarrow measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_ON_COMPOSE_CONTINUOUS_CLOSED_SET:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall x::(real,\ ?'c::type)\ cart.\ IN\ (f\ x)\ s) \wedge continuous\_on\ g\ s \longrightarrow measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_ON_COMPOSE_CONTINUOUS_CLOSED_SET_0:

$\forall (f::(real,\ ?'c::type)\ cart \Rightarrow (real,\ ?'b::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'c::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge measurable\_on\ f\ t \wedge (\forall x::(real,\ ?'c::type)\ cart.\ IN\ (f\ x)\ s) \wedge continuous\_on\ g\ s \wedge IN\ (vec\ (0::nat))\ s \wedge g\ (vec\ (0::nat)) = vec\ (0::nat) \longrightarrow measurable\_on\ (g \circ f)\ t$

thm CONTINUOUS_IMP_MEASURABLE_ON:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\_on\ f\ HOL\_Light\_Import.UNIV \longrightarrow measurable\_on\ f\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_ON_CONST:

$\forall k::(real,\ ?'b::type)\ cart.\ measurable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ k)\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_ON_0:

$\forall s$::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vec* (*0*::*nat*)) *s*

thm MEASURABLE_ON_CMUL:

∀ (*c*::*real*) (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ⟶ *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. % *c* (*f x*)) *s*

thm MEASURABLE_ON_NEG:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ⟶ *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_neg* (*f x*)) *s*

thm MEASURABLE_ON_NEG_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_neg* (*f x*)) *s* = *measurable_on f s*

thm MEASURABLE_ON_NORM:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ⟶ *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *lift* (*vector_norm* (*f x*))) *s*

thm MEASURABLE_ON_PASTECART:

∀ (*f*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*g*::(*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'c*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ∧ *measurable_on g s* ⟶ *measurable_on* (λ*x*::(*real*, *?'c*::*type*) *cart*. *pastecart* (*f x*) (*g x*)) *s*

thm MEASURABLE_ON_COMBINE:

∀ (*h*::(*real*, *?'d*::*type*) *cart* ⇒ (*real*, *?'c*::*type*) *cart* ⇒ (*real*, *?'b*::*type*) *cart*) (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'d*::*type*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'c*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ∧ *measurable_on g s* ∧ *continuous_on* (λ*x*::(*real*, (*?'d*::*type*, *?'c*::*type*) *finite_sum*) *cart*. *h* (*fstcart x*) (*sndcart x*)) *HOL_Light_Import.UNIV* ∧ *h* (*vec* (*0*::*nat*)) (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*) ⟶ *measurable_on* (λ*x*::(*real*, *?'a*::*type*) *cart*. *h* (*f x*) (*g x*)) *s*

thm MEASURABLE_ON_ADD:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *measurable_on f s* ∧ *measurable_on g s* ⟶ *measurable_on* (λ*x*::(*real*, *?'b*::*type*) *cart*. *vector_add* (*f x*) (*g x*)) *s*

thm MEASURABLE_ON_SUB:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ measurable\_on\ f\ s \wedge measurable\_on\ g\ s \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ s$

thm MEASURABLE_ON_MAX:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ measurable\_on\ f\ s \wedge measurable\_on\ g\ s \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ max\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm MEASURABLE_ON_MIN:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ measurable\_on\ f\ s \wedge measurable\_on\ g\ s \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ min\ (\$\ (f\ x)\ i)\ (\$\ (g\ x)\ i)))\ s$

thm MEASURABLE_ON_DROP_MUL:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ unit)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ measurable\_on\ f\ s \wedge measurable\_on\ g\ s \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ \%\ (HOL\_Light\_Import.drop\ (f\ x))\ (g\ x))\ s$

thm MEASURABLE_ON_LIFT_MUL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ measurable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (f\ x))\ s \wedge measurable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (g\ x))\ s \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'a::type)\ cart.\ lift\ (f\ x * g\ x))\ s$

thm MEASURABLE_ON_VSUM:

$\forall (f::?'c::type \Rightarrow (real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ t::?'c::type \Rightarrow bool.\ FINITE\ t \wedge (\forall i::?'c::type.\ IN\ i\ t \longrightarrow measurable\_on\ (f\ i)\ (?s::(real,\ ?'b::type)\ cart \Rightarrow bool)) \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ vsum\ t\ (\lambda i::?'c::type.\ f\ i\ x))\ ?s$

thm MEASURABLE_ON_COMPONENTWISE:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow measurable\_on\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (\$\ (f\ x)\ i))\ HOL\_Light\_Import.UNIV)$

thm MEASURABLE_ON_SPIKE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ negligible\ s \wedge (\forall x::(real,\ ?'b::type)\ cart.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow measurable\_on\ f\ t \longrightarrow measurable\_on\ g\ t$

thm MEASURABLE_ON_SPIKE_SET:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. negligible (HOL_Light_Import.UNION (DIFF s t) (DIFF t s)) ⟶ measurable_on f s ⟶ measurable_on f t

thm MEASURABLE_ON_RESTRICT:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. measurable_on f HOL_Light_Import.UNIV ∧ lebesgue_measurable s ⟶ measurable_on (λx::(real, ?'b::type) cart. if IN x s then f x else vec (0::nat)) HOL_Light_Import.UNIV

thm MEASURABLE_ON_LEBESGUE_MEASURABLE_SUBSET:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. SUBSET s t ∧ measurable_on f t ∧ lebesgue_measurable s ⟶ measurable_on f s

thm MEASURABLE_ON_LIMIT:

∀ (f::nat ⇒ (real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) k::(real, ?'b::type) cart ⇒ bool. (∀ n::nat. measurable_on (f n) s) ∧ negligible k ∧ (∀ x::(real, ?'b::type) cart. IN x (DIFF s k) ⟶ --> (λn::nat. f n x) (g x) sequentially) ⟶ measurable_on g s

thm MEASURABLE_ON_EMPTY:

∀ f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. measurable_on f EMPTY

thm MEASURABLE_ON_INTER:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. measurable_on f s ∧ measurable_on f t ⟶ measurable_on f (HOL_Light_Import.INTER s t)

thm MEASURABLE_ON_DIFF:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. measurable_on f s ∧ measurable_on f t ⟶ measurable_on f (DIFF s t)

thm MEASURABLE_ON_UNION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. measurable_on f s ∧ measurable_on f t ⟶ measurable_on f (HOL_Light_Import.UNION s t)

thm MEASURABLE_ON_UNIONS:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) k::((real, ?'b::type) cart ⇒ bool) ⇒ bool. FINITE k ∧ (∀ s::(real, ?'b::type) cart ⇒ bool. IN s k ⟶ measurable_on f s) ⟶ measurable_on f (UNIONS k)

thm MEASURABLE_ON_COUNTABLE_UNIONS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ k::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ k \wedge (\forall s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ s\ k \longrightarrow measurable\_on\ f\ s) \longrightarrow measurable\_on\ f\ (UNIONS\ k)$

thm Hypermap.LT0_LE1:

$\forall n::nat.\ ((0::nat) < n) = ((1::nat) \leq n)$

thm NEGLIGIBLE_LIPSCHITZ_IMAGE_BOUNDED:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ B::real.\ negligible\ s \wedge SUBSET\ s\ (closed\_interval\ [(a,\ b)]) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ (closed\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_interval\ [(a,\ b)]) \longrightarrow vector\_norm\ (vector\_sub\ (f\,x)\ (f\,y)) \leq B * vector\_norm\ (vector\_sub\ x\ y)) \longrightarrow negligible\ (IMAGE\ f\ s)$

thm NEGLIGIBLE_LOCALLY_LIPSCHITZ_IMAGE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ s \wedge (\forall a::(real,\ ?'a::type)\ cart.\ IN\ a\ s \longrightarrow (\exists (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ B::real.\ IN\ a\ t \wedge HOL\_Light\_Import.open\ t \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ t \wedge IN\ y\ t \longrightarrow vector\_norm\ (vector\_sub\ (f\ x)\ (f\,y)) \leq B * vector\_norm\ (vector\_sub\ x\ y)))) \longrightarrow negligible\ (IMAGE\ f\ s)$

thm NEGLIGIBLE_LIPSCHITZ_IMAGE_UNIV:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ B::real.\ negligible\ s \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_norm\ (vector\_sub\ (f\,x)\ (f\,y)) \leq B * vector\_norm\ (vector\_sub\ x\ y)) \longrightarrow negligible\ (IMAGE\ f\ s)$

thm MEASURABLE_IMP_LEBESGUE_MEASURABLE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow lebesgue\_measurable\ s$

thm NEGLIGIBLE_IMP_LEBESGUE_MEASURABLE:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ s \longrightarrow lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_EMPTY:

$lebesgue\_measurable\ EMPTY$

thm LEBESGUE_MEASURABLE_UNIV:

$lebesgue\_measurable\ HOL\_Light\_Import.UNIV$

thm LEBESGUE_MEASURABLE_COMPACT:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_INTERVAL:

$(\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ lebesgue\_measurable\ (closed\_interval\ [(a,\ b)])) \wedge (\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ lebesgue\_measurable\ (open\_interval\ (a,\ b)))$

thm LEBESGUE_MEASURABLE_INTER:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ s \wedge lebesgue\_measurable\ t \longrightarrow lebesgue\_measurable\ (HOL\_Light\_Import.INTER\ s\ t)$

thm LEBESGUE_MEASURABLE_UNION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ s \wedge lebesgue\_measurable\ t \longrightarrow lebesgue\_measurable\ (HOL\_Light\_Import.UNION\ s\ t)$

thm LEBESGUE_MEASURABLE_DIFF:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ s \wedge lebesgue\_measurable\ t \longrightarrow lebesgue\_measurable\ (DIFF\ s\ t)$

thm LEBESGUE_MEASURABLE_COMPL:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ (DIFF\ HOL\_Light\_Import.UNIV\ s) = lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_INTERVAL_conjunct1:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ lebesgue\_measurable\ (open\_interval\ (a, b))$

thm LEBESGUE_MEASURABLE_INTERVAL_conjunct0:

$\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ lebesgue\_measurable\ (closed\_interval\ [(a, b)])$

thm LEBESGUE_MEASURABLE_ON_SUBINTERVALS:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ s = (\forall (a::(real, ?'a::type)\ cart)\ b::(real, ?'a::type)\ cart.\ lebesgue\_measurable\ (HOL\_Light\_Import.INTER\ s\ (closed\_interval\ [(a, b)])))$

thm LEBESGUE_MEASURABLE_CLOSED:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_OPEN:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_UNIONS:

$\forall f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow lebesgue\_measurable\ s) \longrightarrow lebesgue\_measurable\ (UNIONS\ f)$

thm LEBESGUE_MEASURABLE_COUNTABLE_UNIONS:

$\forall f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ f \wedge (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow lebesgue\_measurable\ s) \longrightarrow lebesgue\_measurable\ (UNIONS\ f)$

thm LEBESGUE_MEASURABLE_COUNTABLE_UNIONS_EXPLICIT:

$\forall s::nat \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool.\ (\forall n::nat.\ lebesgue\_measurable\ (s\ n)) \longrightarrow lebesgue\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2224::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2224\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm LEBESGUE_MEASURABLE_COUNTABLE_INTERS:

$\forall f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ f \wedge (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow lebesgue\_measurable\ s) \longrightarrow lebesgue\_measurable\ (INTERS\ f)$

thm LEBESGUE_MEASURABLE_COUNTABLE_INTERS_EXPLICIT:

$\forall s::nat \Rightarrow (real, ?'a::type)\ cart \Rightarrow bool.\ (\forall n::nat.\ lebesgue\_measurable\ (s\ n)) \longrightarrow lebesgue\_measurable\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%2225::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2225\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm LEBESGUE_MEASURABLE_INTERS:

$\forall f::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow lebesgue\_measurable\ s) \longrightarrow lebesgue\_measurable\ (INTERS\ f)$

thm LEBESGUE_MEASURABLE_IFF_MEASURABLE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ bounded\ s \longrightarrow lebesgue\_measurable\ s = measurable\ s$

thm MEASURABLE_ON_MEASURABLE_SUBSET:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ t::(real, ?'b::type)\ cart \Rightarrow bool.\ SUBSET\ s\ t \wedge measurable\_on\ f\ t \wedge measurable\ s \longrightarrow measurable\_on\ f\ s$

thm MEASURABLE_ON_CASES:

$\forall (P::(real, ?'b::type)\ cart \Rightarrow bool)\ (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2228::(real, ?'b::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2228\ (P\ x)\ x)) \wedge measurable\_on\ f\ s \wedge measurable\_on\ g\ s \longrightarrow measurable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ if\ P\ x\ then\ f\ x\ else\ g\ x)\ s$

thm LEBESGUE_MEASURABLE_JORDAN:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ negligible\ (frontier\ s) \longrightarrow lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_CONVEX:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow lebesgue\_measurable\ s$

thm MEASURABLE_ON_TRANSLATION:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ a::(real, ?'b::type)\ cart.\ measurable\_on\ f\ (IMAGE\ (vector\_add\ a)\ s) \longrightarrow measurable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ f\ (vector\_add\ a\ x))\ s$

thm MEASURABLE_ON_TRANSLATION_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (s::(real, ?'b::type)\ cart \Rightarrow bool)\ a::(real, ?'b::type)\ cart.\ measurable\_on\ (\lambda x::(real, ?'b::type)\ cart.\ f\ (vector\_add\ a\ x))\ s = measurable\_on\ f\ (IMAGE\ (vector\_add\ a)\ s)$

thm MEASURABLE_ON_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'b::type)\ cart)\ (h::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow measurable\_on\ (h \circ f)\ s = measurable\_on\ h\ (IMAGE\ f\ s)$

thm LEBESGUE_MEASURABLE_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ (IMAGE\ (vector\_add\ a)\ s) = lebesgue\_measurable\ s$

thm LEBESGUE_MEASURABLE_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow lebesgue\_measurable\ (IMAGE\ f\ s) = lebesgue\_measurable\ s$

thm MEASURABLE_ON_SIMPLE_FUNCTION_LIMIT:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\exists g::nat \Rightarrow (real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ (\forall n::nat.\ measurable\_on\ (g\ n)\ HOL\_Light\_Import.UNIV) \wedge (\forall n::nat.\ FINITE\ (IMAGE\ (g\ n)\ HOL\_Light\_Import.UNIV)) \wedge (\forall x::(real, ?'b::type)\ cart.\ -->\ (\lambda n::nat.\ g\ n\ x)\ (f\ x)\ sequentially))$

thm MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_COMPONENT_LT:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall (a::real)\ k::nat.\ (1::nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2241::(real, ?'b::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2241\ (\$\ (f\ x)\ k < a)\ x)))$

thm MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_COMPONENT_GE:

$\forall f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart.\ measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall (a::real)\ k::nat.\ (1::nat) \le k \wedge k \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2244::(real, ?'b::type)\ cart.\ \exists x::(real, ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2244\ (a \le \$\ (f\ x)\ k)\ x)))$

thm MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_COMPONENT_GT:

$\forall f$::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ (*a*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2245*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2245* (*a* < $ (*f x*) *k*) *x*)))

thm MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_COMPONENT_LE:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ (*a*::*real*) *k*::*nat*. (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2248*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2248* ($ (*f x*) *k* ≤ *a*) *x*)))

thm MEASURABLE_ON_PREIMAGE_OPEN:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.open t* ⟶ *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2264*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2264* (*IN* (*f x*) *t*) *x*)))

thm MEASURABLE_ON_PREIMAGE_OPEN_INTERVAL:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2263*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2263* (*IN* (*f x*) (*open_interval* (*a*, *b*))) *x*)))

thm MEASURABLE_ON_PREIMAGE_CLOSED:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.closed t* ⟶ *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2267*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2267* (*IN* (*f x*) *t*) *x*)))

thm MEASURABLE_ON_PREIMAGE_CLOSED_INTERVAL:

∀ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *measurable_on f HOL_Light_Import.UNIV* = (∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2272*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2272* (*IN* (*f x*) (*closed_interval* [(*a*, *b*)])) *x*)))

thm LEBESGUE_MEASURABLE_PREIMAGE_OPEN:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable_on f HOL_Light_Import.UNIV* ∧ *HOL_Light_Import.open t* ⟶ *lebesgue_measurable* (*GSPEC* (λ*GEN%PVAR%2273*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%2273* (*IN* (*f x*) *t*) *x*))

thm LEBESGUE_MEASURABLE_PREIMAGE_CLOSED:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable_on f HOL_Light_Import.UNIV* ∧ *HOL_Light_Import.closed*

$t \longrightarrow$ *lebesgue_measurable (GSPEC (λGEN%PVAR%2274::(real, ?'b::type) cart. ∃x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%2274 (IN (f x) t) x))*

thm MEASURABLE_LEGESGUE_MEASURABLE_SUBSET:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. lebesgue_measurable s ∧ measurable t ∧ SUBSET s t ⟶ measurable s

thm MEASURABLE_LEGESGUE_MEASURABLE_INTER_MEASURABLE:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. lebesgue_measurable s ∧ measurable t ⟶ measurable (HOL_Light_Import.INTER s t)

thm MEASURABLE_MEASURABLE_INTER_LEGESGUE_MEASURABLE:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. measurable s ∧ lebesgue_measurable t ⟶ measurable (HOL_Light_Import.INTER s t)

thm MEASURABLE_INTER_HALFSPACE_LE:

∀(s::(real, ?'a::type) cart ⇒ bool) (a::real) i::nat. measurable s ⟶ measurable (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%2275::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2275 ($ x i ≤ a) x)))

thm MEASURABLE_INTER_HALFSPACE_GE:

∀(s::(real, ?'a::type) cart ⇒ bool) (a::real) i::nat. measurable s ⟶ measurable (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%2276::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2276 (a ≤ $ x i) x)))

thm MEASURABLE_MEASURABLE_DIFF_LEGESGUE_MEASURABLE:

∀(s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. measurable s ∧ lebesgue_measurable t ⟶ measurable (DIFF s t)

thm LEBESGUE_MEASURABLE_OUTER_OPEN:

∀(s::(real, ?'a::type) cart ⇒ bool) e::real. lebesgue_measurable s ∧ (0::real) < e ⟶ (∃t::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.open t ∧ SUBSET s t ∧ measurable (DIFF t s) ∧ HOL_Light_Import.measure (DIFF t s) < e)

thm LEBESGUE_MEASURABLE_INNER_CLOSED:

∀(s::(real, ?'a::type) cart ⇒ bool) e::real. lebesgue_measurable s ∧ (0::real) < e ⟶ (∃t::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.closed t ∧ SUBSET t s ∧ measurable (DIFF s t) ∧ HOL_Light_Import.measure (DIFF s t) < e)

thm STEINHAUS_LEBESGUE:

∀s::(real, ?'a::type) cart ⇒ bool. lebesgue_measurable s ∧ ¬ negligible s ⟶ (∃d>0::real. SUBSET (ball (vec (0::nat), d)) (GSPEC (λGEN%PVAR%2279::(real, ?'a::type) cart. ∃(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2279 (IN x s ∧ IN y s) (vector_sub x y))))

thm CONTINUOUS_IMP_MEASURABLE_ON_LEBESGUE_MEASURABLE_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge lebesgue\_measurable\ s \longrightarrow measurable\_on\ f\ s$

thm CONTINUOUS_IMP_MEASURABLE_ON_CLOSED_SUBSET:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.closed\ s \longrightarrow measurable\_on\ f\ s$

thm MEASURABLE_ON_VECTOR_DERIVATIVE:

$\forall (f::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (f'::(real,\ unit)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ unit)\ cart \Rightarrow bool)\ k::(real,\ unit)\ cart \Rightarrow bool.\ negligible\ k \wedge negligible\ (frontier\ s) \wedge (\forall x::(real,\ unit)\ cart.\ IN\ x\ (DIFF\ s\ k) \longrightarrow has\_vector\_derivative\ f\ (f'\ x)\ (at\ x)) \longrightarrow measurable\_on\ f'\ s$

thm ABSOLUTELY_INTEGRABLE_APPROXIMATE_CONTINUOUS:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ e::real.\ measurable\ s \wedge absolutely\_integrable\_on\ f\ s \wedge (0::real) < e \longrightarrow (\exists g::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ absolutely\_integrable\_on\ g\ s \wedge continuous\_on\ g\ HOL\_Light\_Import.UNIV \wedge bounded\ (IMAGE\ g\ HOL\_Light\_Import.UNIV) \wedge vector\_norm\ (integral\ s\ (\lambda x::(real,\ ?'b::type)\ cart.\ lift\ (vector\_norm\ (vector\_sub\ (f\ x)\ (g\ x))))) < e)$

thm DEF_ITER:

$ITER = (SOME\ ITER::nat \Rightarrow nat \Rightarrow (?'a::type \Rightarrow ?'a::type) \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall \_1867420::nat.\ (\forall (x::?'a::type)\ f::?'a::type \Rightarrow ?'a::type.\ ITER\ \_1867420\ (0::nat)\ f\ x = x) \wedge (\forall (x::?'a::type)\ (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ ITER\ \_1867420\ (Suc\ n)\ f\ x = f\ (ITER\ \_1867420\ n\ f\ x)))\ (54::nat)$

thm ITER:

$(\forall f::?'a::type \Rightarrow ?'a::type.\ ITER\ (0::nat)\ f\ (?x::?'a::type) = ?x) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ ITER\ (Suc\ n)\ f\ ?x = f\ (ITER\ n\ f\ ?x))$

thm ITER_conjunct1:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ ITER\ (Suc\ n)\ f\ (?x::?'a::type) = f\ (ITER\ n\ f\ ?x)$

thm ITER_conjunct0:

$\forall f::?'a::type \Rightarrow ?'a::type.\ ITER\ (0::nat)\ f\ (?x::?'a::type) = ?x$

thm ITER_POINTLESS:

$(\forall f::?'b::type \Rightarrow ?'b::type.\ ITER\ (0::nat)\ f = id) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ ITER\ (Suc\ n)\ f = f \circ ITER\ n\ f)$

thm ITER_ALT:

$(\forall (f::?'b::type \Rightarrow ?'b::type)\ x::?'b::type.\ ITER\ (0::nat)\ f\ x = x) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ ITER\ (Suc\ n)\ f\ x = ITER\ n\ f\ (f\ x))$

thm ITER_ALT_conjunct1:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ ITER\ (Suc\ n)\ f\ x = ITER\ n\ f\ (f\ x)$

thm ITER_ALT_conjunct0:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ ITER\ (0::nat)\ f\ x = x$

thm ITER_ALT_POINTLESS:

$(\forall f::?'b::type \Rightarrow ?'b::type.\ ITER\ (0::nat)\ f = id) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ ITER\ (Suc\ n)\ f = ITER\ n\ f \circ f)$

thm ITER_ADD:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ (m::nat)\ x::?'a::type.\ ITER\ n\ f\ (ITER\ m\ f\ x) = ITER\ (n + m)\ f\ x$

thm ITER_MUL:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ (m::nat)\ x::?'a::type.\ ITER\ n\ (ITER\ m\ f)\ x = ITER\ (n * m)\ f\ x$

thm ITER_FIXPOINT:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ f\ x = x \longrightarrow ITER\ n\ f\ x = x$

thm DEF_Re:

$Re = (\lambda\_1867421::(real,\ 2)\ cart.\ \$\ \_1867421\ (1::nat))$

thm RE_DEF:

$\forall z::(real,\ 2)\ cart.\ Re\ z = \$\ z\ (1::nat)$

thm DEF_Im:

$Im = (\lambda\_1867426::(real,\ 2)\ cart.\ \$\ \_1867426\ (2::nat))$

thm IM_DEF:

$\forall z::(real,\ 2)\ cart.\ Im\ z = \$\ z\ (2::nat)$

thm DEF_complex:

$complex = (\lambda\_1867431::real \times real.\ vector\ [fst\ \_1867431,\ snd\ \_1867431])$

thm complex:

$\forall (x::real)\ y::real.\ complex\ (x,\ y) = vector\ [x,\ y]$

thm DEF_Cx:

$Cx = (\lambda\_1867440::real.\ complex\ (\_1867440,\ 0::real))$

thm CX_DEF:

$\forall a::real.\ Cx\ a = complex\ (a,\ 0::real)$

thm ii:

$ii = complex\ (0::real,\ 1::real)$

thm DEF_complex_mul:

$complex\_mul = (\lambda(\_1867445::(real,\ 2)\ cart)\ \_1867446::(real,\ 2)\ cart.\ complex\ (Re\ \_1867445 * Re\ \_1867446 - Im\ \_1867445 * Im\ \_1867446,\ Re\ \_1867445 * Im\ \_1867446 + Im\ \_1867445 * Re\ \_1867446))$

thm complex_mul:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_mul\ w\ z = complex\ (Re\ w * Re\ z - Im\ w * Im\ z,\ Re\ w * Im\ z + Im\ w * Re\ z)$

thm DEF_complex_inv:

$complex\_inv = (\lambda\_1867457::(real,\ 2)\ cart.\ complex\ (Re\ \_1867457\ /\ ((Re\ \_1867457)^2 + (Im\ \_1867457)^2),\ -\ Im\ \_1867457\ /\ ((Re\ \_1867457)^2 + (Im\ \_1867457)^2)))$

thm complex_inv:

$\forall z::(real,\ 2)\ cart.\ complex\_inv\ z = complex\ (Re\ z\ /\ ((Re\ z)^2 + (Im\ z)^2),\ -\ Im\ z\ /\ ((Re\ z)^2 + (Im\ z)^2))$

thm DEF_complex_div:

$complex\_div = (\lambda(\_1867462::(real,\ 2)\ cart)\ \_1867463::(real,\ 2)\ cart.\ complex\_mul\ \_1867462\ (complex\_inv\ \_1867463))$

thm complex_div:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_div\ w\ z = complex\_mul\ w\ (complex\_inv\ z)$

thm DEF_complex_pow:

$complex\_pow = (SOME\ complex\_pow::nat \Rightarrow (real,\ 2)\ cart \Rightarrow nat \Rightarrow (real,\ 2)\ cart.\ \forall \_1867858::nat.\ (\forall x::(real,\ 2)\ cart.\ complex\_pow\ \_1867858\ x\ (0::nat) = Cx\ (1::real)) \wedge (\forall (x::(real,\ 2)\ cart)\ n::nat.\ complex\_pow\ \_1867858\ x\ (Suc\ n) = complex\_mul\ x\ (complex\_pow\ \_1867858\ x\ n)))\ (55::nat)$

thm complex_pow:

$complex\_pow\ (?x::(real,\ 2)\ cart)\ (0::nat) = Cx\ (1::real) \wedge (\forall n::nat.\ complex\_pow\ ?x\ (Suc\ n) = complex\_mul\ ?x\ (complex\_pow\ ?x\ n))$

thm RE:

$Re\ (complex\ (?x::real,\ ?y::real)) = ?x$

thm IM:

$Im\ (complex\ (?x::real,\ ?y::real)) = ?y$

thm COMPLEX_EQ:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ (w = z) = (Re\ w = Re\ z \wedge Im\ w = Im\ z)$

thm COMPLEX:

$\forall z{::}(real,\ 2)\ cart.\ complex\ (Re\ z,\ Im\ z) = z$

thm COMPLEX_EQ_0:

$((?z{::}(real,\ 2)\ cart) = Cx\ (0{::}real)) = ((Re\ ?z)^2 + (Im\ ?z)^2 = (0{::}real))$

thm FORALL_COMPLEX:

$(\forall z{::}(real,\ 2)\ cart.\ (?P{::}(real,\ 2)\ cart \Rightarrow bool)\ z) = (\forall (x{::}real)\ y{::}real.\ ?P\ (complex\ (x,\ y)))$

thm EXISTS_COMPLEX:

$(\exists z{::}(real,\ 2)\ cart.\ (?P{::}(real,\ 2)\ cart \Rightarrow bool)\ z) = (\exists (x{::}real)\ y{::}real.\ ?P\ (complex\ (x,\ y)))$

thm complex_neg:

$vector\_neg\ (?z{::}(real,\ 2)\ cart) = complex\ (-\ Re\ ?z,\ -\ Im\ ?z)$

thm complex_add:

$vector\_add\ (?w{::}(real,\ 2)\ cart)\ (?z{::}(real,\ 2)\ cart) = complex\ (Re\ ?w + Re\ ?z,\ Im\ ?w + Im\ ?z)$

thm complex_sub:

$vector\_sub\ (?w{::}(real,\ 2)\ cart)\ (?z{::}(real,\ 2)\ cart) = vector\_add\ ?w\ (vector\_neg\ ?z)$

thm complex_norm:

$vector\_norm\ (?z{::}(real,\ 2)\ cart) = sqrt\ ((Re\ ?z)^2 + (Im\ ?z)^2)$

thm COMPLEX_SQNORM:

$(vector\_norm\ (?z{::}(real,\ 2)\ cart))^2 = (Re\ ?z)^2 + (Im\ ?z)^2$

thm COMPLEX_ADD_SYM:

$\forall (x{::}(real,\ 2)\ cart)\ y{::}(real,\ 2)\ cart.\ vector\_add\ x\ y = vector\_add\ y\ x$

thm COMPLEX_ADD_ASSOC:

$\forall (x{::}(real,\ 2)\ cart)\ (y{::}(real,\ 2)\ cart)\ z{::}(real,\ 2)\ cart.\ vector\_add\ x\ (vector\_add\ y\ z) = vector\_add\ (vector\_add\ x\ y)\ z$

thm COMPLEX_ADD_LID:

$\forall x{::}(real,\ 2)\ cart.\ vector\_add\ (Cx\ (0{::}real))\ x = x$

thm COMPLEX_ADD_LINV:

$\forall x{::}(real,\ 2)\ cart.\ vector\_add\ (vector\_neg\ x)\ x = Cx\ (0{::}real)$

thm COMPLEX_MUL_SYM:

$\forall (x{::}(real,\ 2)\ cart)\ y{::}(real,\ 2)\ cart.\ complex\_mul\ x\ y = complex\_mul\ y\ x$

thm COMPLEX_MUL_ASSOC:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\, x\ (complex\_mul\ y\ z) = complex\_mul\ (complex\_mul\ x\ y)\ z$

thm COMPLEX_MUL_LID:

$\forall x::(real, 2)\ cart.\ complex\_mul\ (Cx\ (1::real))\ x = x$

thm COMPLEX_ADD_LDISTRIB:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\, x\ (vector\_add\ y\ z) = vector\_add\ (complex\_mul\ x\ y)\ (complex\_mul\ x\ z)$

thm COMPLEX_ADD_AC:

$vector\_add\ (?m::(real, 2)\ cart)\ (?n::(real, 2)\ cart) = vector\_add\ ?n\ ?m \wedge vector\_add\ (vector\_add\ ?m\ ?n)\ (?p::(real, 2)\ cart) = vector\_add\ ?m\ (vector\_add\ ?n\ ?p) \wedge vector\_add\ ?m\ (vector\_add\ ?n\ ?p) = vector\_add\ ?n\ (vector\_add\ ?m\ ?p)$

thm COMPLEX_MUL_AC:

$complex\_mul\ (?m::(real, 2)\ cart)\ (?n::(real, 2)\ cart) = complex\_mul\ ?n\ ?m \wedge complex\_mul\ (complex\_mul\ ?m\ ?n)\ (?p::(real, 2)\ cart) = complex\_mul\ ?m\ (complex\_mul\ ?n\ ?p) \wedge complex\_mul\ ?m\ (complex\_mul\ ?n\ ?p) = complex\_mul\ ?n\ (complex\_mul\ ?m\ ?p)$

thm COMPLEX_ADD_RID:

$\forall x::(real, 2)\ cart.\ vector\_add\ x\ (Cx\ (0::real)) = x$

thm COMPLEX_MUL_RID:

$\forall x::(real, 2)\ cart.\ complex\_mul\ x\ (Cx\ (1::real)) = x$

thm COMPLEX_ADD_RINV:

$\forall x::(real, 2)\ cart.\ vector\_add\ x\ (vector\_neg\ x) = Cx\ (0::real)$

thm COMPLEX_ADD_RDISTRIB:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\ (vector\_add\ x\ y)\ z = vector\_add\ (complex\_mul\ x\ z)\ (complex\_mul\ y\ z)$

thm COMPLEX_EQ_ADD_LCANCEL:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ (vector\_add\ x\ y = vector\_add\ x\ z) = (y = z)$

thm COMPLEX_EQ_ADD_RCANCEL:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ (vector\_add\ x\ z = vector\_add\ y\ z) = (x = y)$

thm COMPLEX_MUL_RZERO:

$\forall x::(real, 2)\ cart.\ complex\_mul\ x\ (Cx\ (0::real)) = Cx\ (0::real)$

thm COMPLEX_MUL_LZERO:

$\forall x$::(*real*, *2*) *cart*. *complex_mul* (*Cx* (*0*::*real*)) *x* = *Cx* (*0*::*real*)

thm COMPLEX_NEG_NEG:

$\forall x$::(*real*, *2*) *cart*. *vector_neg* (*vector_neg x*) = *x*

thm COMPLEX_MUL_RNEG:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *complex_mul x* (*vector_neg y*) = *vector_neg* (*complex_mul x y*)

thm COMPLEX_MUL_LNEG:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *complex_mul* (*vector_neg x*) *y* = *vector_neg* (*complex_mul x y*)

thm COMPLEX_NEG_ADD:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *vector_neg* (*vector_add x y*) = *vector_add* (*vector_neg x*) (*vector_neg y*)

thm COMPLEX_NEG_0:

vector_neg (*Cx* (*0*::*real*)) = *Cx* (*0*::*real*)

thm COMPLEX_EQ_ADD_LCANCEL_0:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. (*vector_add x y* = *x*) = (*y* = *Cx* (*0*::*real*))

thm COMPLEX_EQ_ADD_RCANCEL_0:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. (*vector_add x y* = *y*) = (*x* = *Cx* (*0*::*real*))

thm COMPLEX_LNEG_UNIQ:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. (*vector_add x y* = *Cx* (*0*::*real*)) = (*x* = *vector_neg y*)

thm COMPLEX_RNEG_UNIQ:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. (*vector_add x y* = *Cx* (*0*::*real*)) = (*y* = *vector_neg x*)

thm COMPLEX_NEG_LMUL:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *vector_neg* (*complex_mul x y*) = *complex_mul* (*vector_neg x*) *y*

thm COMPLEX_NEG_RMUL:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *vector_neg* (*complex_mul x y*) = *complex_mul* *x* (*vector_neg y*)

thm COMPLEX_NEG_MUL2:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *complex_mul* (*vector_neg x*) (*vector_neg* *y*) = *complex_mul x y*

thm COMPLEX_SUB_ADD:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_add\ (vector\_sub\ x\ y)\ y = x$

thm COMPLEX_SUB_ADD2:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_add\ y\ (vector\_sub\ x\ y) = x$

thm COMPLEX_SUB_REFL:

$\forall x::(real, 2)\ cart.\ vector\_sub\ x\ x = Cx\ (0::real)$

thm COMPLEX_SUB_0:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ (vector\_sub\ x\ y = Cx\ (0::real)) = (x = y)$

thm COMPLEX_NEG_EQ_0:

$\forall x::(real, 2)\ cart.\ (vector\_neg\ x = Cx\ (0::real)) = (x = Cx\ (0::real))$

thm COMPLEX_NEG_SUB:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_neg\ (vector\_sub\ x\ y) = vector\_sub\ y\ x$

thm COMPLEX_ADD_SUB:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_sub\ (vector\_add\ x\ y)\ x = y$

thm COMPLEX_NEG_EQ:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ (vector\_neg\ x = y) = (x = vector\_neg\ y)$

thm COMPLEX_NEG_MINUS1:

$\forall x::(real, 2)\ cart.\ vector\_neg\ x = complex\_mul\ (vector\_neg\ (Cx\ (1::real)))\ x$

thm COMPLEX_SUB_SUB:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_sub\ (vector\_sub\ x\ y)\ x = vector\_neg\ y$

thm COMPLEX_ADD2_SUB2:

$\forall (a::(real, 2)\ cart)\ (b::(real, 2)\ cart)\ (c::(real, 2)\ cart)\ d::(real, 2)\ cart.\ vector\_sub\ (vector\_add\ a\ b)\ (vector\_add\ c\ d) = vector\_add\ (vector\_sub\ a\ c)\ (vector\_sub\ b\ d)$

thm COMPLEX_SUB_LZERO:

$\forall x::(real, 2)\ cart.\ vector\_sub\ (Cx\ (0::real))\ x = vector\_neg\ x$

thm COMPLEX_SUB_RZERO:

$\forall x::(real, 2)\ cart.\ vector\_sub\ x\ (Cx\ (0::real)) = x$

thm COMPLEX_SUB_LNEG:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ vector\_sub\ (vector\_neg\ x)\ y = vector\_neg\ (vector\_add\ x\ y)$

thm COMPLEX_SUB_RNEG:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ vector\_sub\ x\ (vector\_neg\ y) = vector\_add\ x\ y$

thm COMPLEX_SUB_NEG2:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ vector\_sub\ (vector\_neg\ x)\ (vector\_neg\ y) = vector\_sub\ y\ x$

thm COMPLEX_SUB_TRIANGLE:

$\forall (a::(real,\ 2)\ cart)\ (b::(real,\ 2)\ cart)\ c::(real,\ 2)\ cart.\ vector\_add\ (vector\_sub\ a\ b)\ (vector\_sub\ b\ c) = vector\_sub\ a\ c$

thm COMPLEX_EQ_SUB_LADD:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ (x = vector\_sub\ y\ z) = (vector\_add\ x\ z = y)$

thm COMPLEX_EQ_SUB_RADD:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ (vector\_sub\ x\ y = z) = (x = vector\_add\ z\ y)$

thm COMPLEX_SUB_SUB2:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ vector\_sub\ x\ (vector\_sub\ x\ y) = y$

thm COMPLEX_ADD_SUB2:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ vector\_sub\ x\ (vector\_add\ x\ y) = vector\_neg\ y$

thm COMPLEX_DIFFSQ:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ complex\_mul\ (vector\_add\ x\ y)\ (vector\_sub\ x\ y) = vector\_sub\ (complex\_mul\ x\ x)\ (complex\_mul\ y\ y)$

thm COMPLEX_EQ_NEG2:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ (vector\_neg\ x = vector\_neg\ y) = (x = y)$

thm COMPLEX_SUB_LDISTRIB:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_mul\ x\ (vector\_sub\ y\ z) = vector\_sub\ (complex\_mul\ x\ y)\ (complex\_mul\ x\ z)$

thm COMPLEX_SUB_RDISTRIB:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_mul\ (vector\_sub\ x\ y)\ z = vector\_sub\ (complex\_mul\ x\ z)\ (complex\_mul\ y\ z)$

thm COMPLEX_MUL_2:

$\forall x::(real,\ 2)\ cart.\ complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ x = vector\_add\ x\ x$

thm II_NZ:

$ii \neq Cx\ (0::real)$

thm COMPLEX_MUL_LINV:

$\forall z::(real,\ 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow complex\_mul\ (complex\_inv\ z)\ z = Cx\ (1::real)$

thm COMPLEX_ENTIRE:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ (complex\_mul\ x\ y = Cx\ (0::real)) = (x = Cx\ (0::real) \lor y = Cx\ (0::real))$

thm COMPLEX_MUL_RINV:

$\forall z::(real,\ 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow complex\_mul\ z\ (complex\_inv\ z) = Cx\ (1::real)$

thm COMPLEX_DIV_REFL:

$\forall x::(real,\ 2)\ cart.\ x \neq Cx\ (0::real) \longrightarrow complex\_div\ x\ x = Cx\ (1::real)$

thm CX_INJ:

$\forall (x::real)\ y::real.\ (Cx\ x = Cx\ y) = (x = y)$

thm CX_NEG:

$\forall x::real.\ Cx\ (-\ x) = vector\_neg\ (Cx\ x)$

thm CX_ADD:

$\forall (x::real)\ y::real.\ Cx\ (x + y) = vector\_add\ (Cx\ x)\ (Cx\ y)$

thm CX_SUB:

$\forall (x::real)\ y::real.\ Cx\ (x - y) = vector\_sub\ (Cx\ x)\ (Cx\ y)$

thm CX_INV:

$\forall x::real.\ Cx\ (inverse\_class.inverse\ x) = complex\_inv\ (Cx\ x)$

thm CX_MUL:

$\forall (x::real)\ y::real.\ Cx\ (x * y) = complex\_mul\ (Cx\ x)\ (Cx\ y)$

thm complex_pow_conjunct1:

$\forall n::nat.\ complex\_pow\ (?x::(real,\ 2)\ cart)\ (Suc\ n) = complex\_mul\ ?x\ (complex\_pow\ ?x\ n)$

thm complex_pow_conjunct0:

$complex\_pow\ (?x::(real,\ 2)\ cart)\ (0::nat) = Cx\ (1::real)$

thm CX_POW:

$\forall (x::real)\ n::nat.\ Cx\ x^n = complex\_pow\ (Cx\ x)\ n$

thm CX_DIV:

$\forall (x::real)\ y::real.\ Cx\ (x\ /\ y) = complex\_div\ (Cx\ x)\ (Cx\ y)$

thm CX_ABS:

$\forall x::real.\ Cx\ |x| = Cx\ (vector\_norm\ (Cx\ x))$

thm COMPLEX_NORM_CX:

$\forall x::real.\ vector\_norm\ (Cx\ x) = |x|$

thm RE_CX:

$\forall x::real.\ Re\ (Cx\ x) = x$

thm RE_NEG:

$\forall x::(real,\ 2)\ cart.\ Re\ (vector\_neg\ x) = -\ Re\ x$

thm RE_ADD:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ Re\ (vector\_add\ x\ y) = Re\ x + Re\ y$

thm RE_SUB:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ Re\ (vector\_sub\ x\ y) = Re\ x - Re\ y$

thm IM_CX:

$\forall x::real.\ Im\ (Cx\ x) = (0::real)$

thm IM_NEG:

$\forall x::(real,\ 2)\ cart.\ Im\ (vector\_neg\ x) = -\ Im\ x$

thm IM_ADD:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ Im\ (vector\_add\ x\ y) = Im\ x + Im\ y$

thm IM_SUB:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ Im\ (vector\_sub\ x\ y) = Im\ x - Im\ y$

thm COMPLEX_EXPAND:

$\forall z::(real,\ 2)\ cart.\ z = vector\_add\ (Cx\ (Re\ z))\ (complex\_mul\ ii\ (Cx\ (Im\ z)))$

thm COMPLEX_TRAD:

$\forall (x::real)\ y::real.\ complex\ (x,\ y) = vector\_add\ (Cx\ x)\ (complex\_mul\ ii\ (Cx\ y))$

thm RE_II:

$Re\ ii = (0::real)$

thm IM_II:

$Im\ ii = (1::real)$

thm RE_MUL_II:

$\forall z::(real, 2)\ cart.\ Re\ (complex\_mul\ z\ ii) = -\ Im\ z \wedge Re\ (complex\_mul\ ii\ z) = -\ Im\ z$

thm IM_MUL_II:

$\forall z::(real, 2)\ cart.\ Im\ (complex\_mul\ z\ ii) = Re\ z \wedge Im\ (complex\_mul\ ii\ z) = Re\ z$

thm COMPLEX_NORM_II:

$vector\_norm\ ii = (1::real)$

thm RE_CMUL:

$\forall (a::real)\ z::(real, 2)\ cart.\ Re\ (\%\ a\ z) = a * Re\ z$

thm IM_CMUL:

$\forall (a::real)\ z::(real, 2)\ cart.\ Im\ (\%\ a\ z) = a * Im\ z$

thm RE_MUL_CX:

$\forall (x::real)\ z::(real, 2)\ cart.\ Re\ (complex\_mul\ (Cx\ x)\ z) = x * Re\ z \wedge Re\ (complex\_mul\ z\ (Cx\ x)) = Re\ z * x$

thm IM_MUL_CX:

$\forall (x::real)\ z::(real, 2)\ cart.\ Im\ (complex\_mul\ (Cx\ x)\ z) = x * Im\ z \wedge Im\ (complex\_mul\ z\ (Cx\ x)) = Im\ z * x$

thm RE_DIV_CX:

$\forall (z::(real, 2)\ cart)\ x::real.\ Re\ (complex\_div\ z\ (Cx\ x)) = Re\ z\ /\ x$

thm IM_DIV_CX:

$\forall (z::(real, 2)\ cart)\ x::real.\ Im\ (complex\_div\ z\ (Cx\ x)) = Im\ z\ /\ x$

thm COMPLEX_POLY_CLAUSES:

$(\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ vector\_add\ x\ (vector\_add\ y\ z) = vector\_add\ (vector\_add\ x\ y)\ z) \wedge (\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_add\ x\ y = vector\_add\ y\ x) \wedge (\forall x::(real, 2)\ cart.\ vector\_add\ (Cx\ (0::real))\ x = x) \wedge (\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\ x\ (complex\_mul\ y\ z) = complex\_mul\ (complex\_mul\ x\ y)\ z) \wedge (\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ complex\_mul\ x\ y = complex\_mul\ y\ x) \wedge (\forall x::(real, 2)\ cart.\ complex\_mul\ (Cx\ (1::real))\ x = x) \wedge (\forall x::(real, 2)\ cart.\ complex\_mul\ (Cx\ (0::real))\ x = Cx\ (0::real)) \wedge (\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\ x\ (vector\_add\ y\ z) = vector\_add\ (complex\_mul\ x\ y)\ (complex\_mul\ x\ z)) \wedge (\forall x::(real, 2)\ cart.\ complex\_pow\ x\ (0::nat) = Cx\ (1::real)) \wedge (\forall (x::(real, 2)\ cart)\ n::nat.\ complex\_pow\ x\ (Suc\ n) = complex\_mul\ x\ (complex\_pow\ x\ n))$

thm COMPLEX_POLY_NEG_CLAUSES:

$(\forall x::(real, 2)\ cart.\ vector\_neg\ x = complex\_mul\ (Cx\ (-\ (1::real)))\ x) \wedge (\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_sub\ x\ y = vector\_add\ x\ (complex\_mul\ (Cx\ (-\ (1::real)))\ y))$

thm COMPLEX_POLY_NEG_CLAUSES_conjunct0:

$\forall x::(real, 2)\ cart.\ vector\_neg\ x = complex\_mul\ (Cx\ (-\ (1::real)))\ x$

thm COMPLEX_POLY_NEG_CLAUSES_conjunct1:

$\forall (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ vector\_sub\ x\ y = vector\_add\ x\ (complex\_mul\ (Cx\ (-\ (1::real)))\ y)$

thm COMPLEX_INV_0:

$complex\_inv\ (Cx\ (0::real)) = Cx\ (0::real)$

thm COMPLEX_INV_1:

$complex\_inv\ (Cx\ (1::real)) = Cx\ (1::real)$

thm COMPLEX_INV_MUL:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_inv\ (complex\_mul\ w\ z) = complex\_mul\ (complex\_inv\ w)\ (complex\_inv\ z)$

thm COMPLEX_POW_INV:

$\forall (x::(real, 2)\ cart)\ n::nat.\ complex\_pow\ (complex\_inv\ x)\ n = complex\_inv\ (complex\_pow\ x\ n)$

thm COMPLEX_INV_INV:

$\forall x::(real, 2)\ cart.\ complex\_inv\ (complex\_inv\ x) = x$

thm COMPLEX_MUL_AC_conjunct2:

$complex\_mul\ (?m::(real, 2)\ cart)\ (complex\_mul\ (?n::(real, 2)\ cart)\ (?p::(real, 2)\ cart)) = complex\_mul\ ?n\ (complex\_mul\ ?m\ ?p)$

thm COMPLEX_MUL_AC_conjunct1:

$complex\_mul\ (complex\_mul\ (?m::(real, 2)\ cart)\ (?n::(real, 2)\ cart))\ (?p::(real, 2)\ cart) = complex\_mul\ ?m\ (complex\_mul\ ?n\ ?p)$

thm COMPLEX_MUL_AC_conjunct0:

$complex\_mul\ (?m::(real, 2)\ cart)\ (?n::(real, 2)\ cart) = complex\_mul\ ?n\ ?m$

thm COMPLEX_INV_DIV:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_inv\ (complex\_div\ w\ z) = complex\_div\ z\ w$

thm COMPLEX_EQ_MUL_LCANCEL:

$\forall (x::(real, 2)\ cart)\ (y::(real, 2)\ cart)\ z::(real, 2)\ cart.\ (complex\_mul\ x\ y = complex\_mul\ x\ z) = (x = Cx\ (0::real) \vee y = z)$

thm COMPLEX_EQ_MUL_RCANCEL:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ (complex\_mul\ x\ z = complex\_mul\ y\ z) = (x = y \vee z = Cx\ (0::real))$

thm COMPLEX_DIV_1:

$\forall z::(real,\ 2)\ cart.\ complex\_div\ z\ (Cx\ (1::real)) = z$

thm COMPLEX_DIV_LMUL:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ y \neq Cx\ (0::real) \longrightarrow complex\_mul\ y\ (complex\_div\ x\ y) = x$

thm COMPLEX_DIV_RMUL:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ y \neq Cx\ (0::real) \longrightarrow complex\_mul\ (complex\_div\ x\ y)\ y = x$

thm COMPLEX_INV_EQ_0:

$\forall x::(real,\ 2)\ cart.\ (complex\_inv\ x = Cx\ (0::real)) = (x = Cx\ (0::real))$

thm COMPLEX_INV_NEG:

$\forall x::(real,\ 2)\ cart.\ complex\_inv\ (vector\_neg\ x) = vector\_neg\ (complex\_inv\ x)$

thm COMPLEX_NEG_INV:

$\forall x::(real,\ 2)\ cart.\ vector\_neg\ (complex\_inv\ x) = complex\_inv\ (vector\_neg\ x)$

thm COMPLEX_INV_EQ_1:

$\forall x::(real,\ 2)\ cart.\ (complex\_inv\ x = Cx\ (1::real)) = (x = Cx\ (1::real))$

thm COMPLEX_POW_ADD:

$\forall (x::(real,\ 2)\ cart)\ (m::nat)\ n::nat.\ complex\_pow\ x\ (m + n) = complex\_mul\ (complex\_pow\ x\ m)\ (complex\_pow\ x\ n)$

thm COMPLEX_POW_POW:

$\forall (x::(real,\ 2)\ cart)\ (m::nat)\ n::nat.\ complex\_pow\ (complex\_pow\ x\ m)\ n = complex\_pow\ x\ (m * n)$

thm COMPLEX_POW_1:

$\forall x::(real,\ 2)\ cart.\ complex\_pow\ x\ (1::nat) = x$

thm COMPLEX_POW_2:

$\forall x::(real,\ 2)\ cart.\ complex\_pow\ x\ (2::nat) = complex\_mul\ x\ x$

thm COMPLEX_POW_NEG:

$\forall (x::(real,\ 2)\ cart)\ n::nat.\ complex\_pow\ (vector\_neg\ x)\ n = (if\ even\ n\ then\ complex\_pow\ x\ n\ else\ vector\_neg\ (complex\_pow\ x\ n))$

thm COMPLEX_POW_ONE:

$\forall n::nat.\ complex\_pow\ (Cx\ (1::real))\ n = Cx\ (1::real)$

thm COMPLEX_POW_MUL:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ n::nat.\ complex\_pow\ (complex\_mul\ x\ y)\ n = complex\_mul\ (complex\_pow\ x\ n)\ (complex\_pow\ y\ n)$

thm COMPLEX_POW_DIV:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ n::nat.\ complex\_pow\ (complex\_div\ x\ y)\ n = complex\_div\ (complex\_pow\ x\ n)\ (complex\_pow\ y\ n)$

thm COMPLEX_POW_II_2:

$complex\_pow\ ii\ (2::nat) = vector\_neg\ (Cx\ (1::real))$

thm COMPLEX_POW_EQ_0:

$\forall (x::(real,\ 2)\ cart)\ n::nat.\ (complex\_pow\ x\ n = Cx\ (0::real)) = (x = Cx\ (0::real) \wedge n \neq (0::nat))$

thm COMPLEX_POW_ZERO:

$\forall n::nat.\ complex\_pow\ (Cx\ (0::real))\ n = (if\ n = (0::nat)\ then\ Cx\ (1::real)\ else\ Cx\ (0::real))$

thm COMPLEX_INV_II:

$complex\_inv\ ii = vector\_neg\ ii$

thm COMPLEX_DIV_POW:

$\forall (x::(real,\ 2)\ cart)\ (n::nat)\ k::nat.\ x \neq Cx\ (0::real) \wedge k \leq n \wedge k \neq (0::nat) \longrightarrow complex\_pow\ x\ (n - k) = complex\_div\ (complex\_pow\ x\ n)\ (complex\_pow\ x\ k)$

thm COMPLEX_VEC_0:

$vec\ (0::nat) = Cx\ (0::real)$

thm COMPLEX_NORM_ZERO:

$\forall z::(real,\ 2)\ cart.\ (vector\_norm\ z = (0::real)) = (z = Cx\ (0::real))$

thm COMPLEX_NORM_NUM:

$\forall n::nat.\ vector\_norm\ (Cx\ (real\_of\_nat\ n)) = real\_of\_nat\ n$

thm COMPLEX_NORM_0:

$vector\_norm\ (Cx\ (0::real)) = (0::real)$

thm COMPLEX_NORM_NZ:

$\forall z::(real,\ 2)\ cart.\ ((0::real) < vector\_norm\ z) = (z \neq Cx\ (0::real))$

thm COMPLEX_NORM_MUL:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ vector\_norm\ (complex\_mul\ w\ z) = vector\_norm\ w * vector\_norm\ z$

thm COMPLEX_NORM_POW:

$\forall (z::(real,\ 2)\ cart)\ n::nat.\ vector\_norm\ (complex\_pow\ z\ n) = (vector\_norm\ z)^n$

thm COMPLEX_NORM_INV:

$\forall z::(real,\ 2)\ cart.\ vector\_norm\ (complex\_inv\ z) = inverse\_class.inverse\ (vector\_norm\ z)$

thm COMPLEX_NORM_DIV:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ vector\_norm\ (complex\_div\ w\ z) = vector\_norm\ w\ /\ vector\_norm\ z$

thm COMPLEX_NORM_TRIANGLE_SUB:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ vector\_norm\ w \leq vector\_norm\ (vector\_add\ w\ z) + vector\_norm\ z$

thm COMPLEX_NORM_ABS_NORM:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ |vector\_norm\ w - vector\_norm\ z| \leq vector\_norm\ (vector\_sub\ w\ z)$

thm COMPLEX_POW_EQ_1:

$\forall (z::(real,\ 2)\ cart)\ n::nat.\ complex\_pow\ z\ n = Cx\ (1::real) \longrightarrow vector\_norm\ z = (1::real) \lor n = (0::nat)$

thm DEF_cnj:

$cnj = (\lambda\_1868149::(real,\ 2)\ cart.\ complex\ (Re\ \_1868149,\ -\ Im\ \_1868149))$

thm cnj:

$\forall z::(real,\ 2)\ cart.\ cnj\ z = complex\ (Re\ z,\ -\ Im\ z)$

thm CNJ_INJ:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ (cnj\ w = cnj\ z) = (w = z)$

thm CNJ_CNJ:

$\forall z::(real,\ 2)\ cart.\ cnj\ (cnj\ z) = z$

thm CNJ_CX:

$\forall x::real.\ cnj\ (Cx\ x) = Cx\ x$

thm COMPLEX_NORM_CNJ:

$\forall z::(real,\ 2)\ cart.\ vector\_norm\ (cnj\ z) = vector\_norm\ z$

thm CNJ_NEG:

$\forall z::(real,\ 2)\ cart.\ cnj\ (vector\_neg\ z) = vector\_neg\ (cnj\ z)$

thm CNJ_INV:

$\forall z$::(*real*, *2*) *cart*. *cnj* (*complex_inv* *z*) = *complex_inv* (*cnj* *z*)

thm CNJ_ADD:

$\forall$(*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *cnj* (*vector_add* *w* *z*) = *vector_add* (*cnj* *w*) (*cnj* *z*)

thm CNJ_SUB:

$\forall$(*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *cnj* (*vector_sub* *w* *z*) = *vector_sub* (*cnj* *w*) (*cnj* *z*)

thm CNJ_MUL:

$\forall$(*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *cnj* (*complex_mul* *w* *z*) = *complex_mul* (*cnj* *w*) (*cnj* *z*)

thm CNJ_DIV:

$\forall$(*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *cnj* (*complex_div* *w* *z*) = *complex_div* (*cnj* *w*) (*cnj* *z*)

thm CNJ_POW:

$\forall$(*z*::(*real*, *2*) *cart*) *n*::*nat*. *cnj* (*complex_pow* *z* *n*) = *complex_pow* (*cnj* *z*) *n*

thm RE_CNJ:

$\forall z$::(*real*, *2*) *cart*. *Re* (*cnj* *z*) = *Re* *z*

thm IM_CNJ:

$\forall z$::(*real*, *2*) *cart*. *Im* (*cnj* *z*) = − *Im* *z*

thm CNJ_EQ_CX:

$\forall$(*x*::*real*) *z*::(*real*, *2*) *cart*. (*cnj* *z* = *Cx* *x*) = (*z* = *Cx* *x*)

thm CNJ_EQ_0:

$\forall z$::(*real*, *2*) *cart*. (*cnj* *z* = *Cx* (*0*::*real*)) = (*z* = *Cx* (*0*::*real*))

thm COMPLEX_ADD_CNJ:

($\forall z$::(*real*, *2*) *cart*. *vector_add* *z* (*cnj* *z*) = *Cx* (*real_of_nat* (*2*::*nat*) ∗ *Re* *z*)) ∧
($\forall z$::(*real*, *2*) *cart*. *vector_add* (*cnj* *z*) *z* = *Cx* (*real_of_nat* (*2*::*nat*) ∗ *Re* *z*))

thm CNJ_II:

cnj *ii* = *vector_neg* *ii*

thm CX_RE_CNJ:

$\forall z$::(*real*, *2*) *cart*. *Cx* (*Re* *z*) = *complex_div* (*vector_add* *z* (*cnj* *z*)) (*Cx* (*real_of_nat* (*2*::*nat*)))

thm CX_IM_CNJ:

$\forall z$::(*real*, *2*) *cart*. *Cx* (*Im* *z*) = *complex_mul* (*vector_neg* *ii*) (*complex_div* (*vector_sub* *z* (*cnj* *z*)) (*Cx* (*real_of_nat* (*2*::*nat*))))

thm COMPLEX_NORM_POW_2:

∀ *z*::(*real*, *2*) *cart*. *complex_pow* (*Cx* (*vector_norm z*)) (*2*::*nat*) = *complex_mul z* (*cnj z*)

thm COMPLEX_MUL_CNJ:

∀ *z*::(*real*, *2*) *cart*. *complex_mul* (*cnj z*) *z* = *complex_pow* (*Cx* (*vector_norm z*)) (*2*::*nat*) ∧ *complex_mul z* (*cnj z*) = *complex_pow* (*Cx* (*vector_norm z*)) (*2*::*nat*)

thm COMPLEX_INV_CNJ:

∀ *z*::(*real*, *2*) *cart*. *complex_inv z* = *complex_div* (*cnj z*) (*complex_pow* (*Cx* (*vector_norm z*)) (*2*::*nat*))

thm COMPLEX_DIV_CNJ:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. *complex_div a b* = *complex_div* (*complex_mul a* (*cnj b*)) (*complex_pow* (*Cx* (*vector_norm b*)) (*2*::*nat*))

thm RE_COMPLEX_DIV_EQ_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. (*Re* (*complex_div a b*) = (*0*::*real*)) = (*Re* (*complex_mul a* (*cnj b*)) = (*0*::*real*))

thm IM_COMPLEX_DIV_EQ_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. (*Im* (*complex_div a b*) = (*0*::*real*)) = (*Im* (*complex_mul a* (*cnj b*)) = (*0*::*real*))

thm RE_COMPLEX_DIV_GT_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. ((*0*::*real*) < *Re* (*complex_div a b*)) = ((*0*::*real*) < *Re* (*complex_mul a* (*cnj b*)))

thm IM_COMPLEX_DIV_GT_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. ((*0*::*real*) < *Im* (*complex_div a b*)) = ((*0*::*real*) < *Im* (*complex_mul a* (*cnj b*)))

thm RE_COMPLEX_DIV_GE_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. ((*0*::*real*) ≤ *Re* (*complex_div a b*)) = ((*0*::*real*) ≤ *Re* (*complex_mul a* (*cnj b*)))

thm IM_COMPLEX_DIV_GE_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. ((*0*::*real*) ≤ *Im* (*complex_div a b*)) = ((*0*::*real*) ≤ *Im* (*complex_mul a* (*cnj b*)))

thm RE_COMPLEX_DIV_LE_0:

∀ (*a*::(*real*, *2*) *cart*) *b*::(*real*, *2*) *cart*. (*Re* (*complex_div a b*) ≤ (*0*::*real*)) = (*Re* (*complex_mul a* (*cnj b*)) ≤ (*0*::*real*))

thm IM_COMPLEX_DIV_LE_0:

$\forall (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ (Im\ (complex\_div\ a\ b) \le (0::real)) = (Im\ (complex\_mul\ a\ (cnj\ b)) \le (0::real))$

thm RE_COMPLEX_DIV_LT_0:

$\forall (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ (Re\ (complex\_div\ a\ b) < (0::real)) = (Re\ (complex\_mul\ a\ (cnj\ b)) < (0::real))$

thm IM_COMPLEX_DIV_LT_0:

$\forall (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ (Im\ (complex\_div\ a\ b) < (0::real)) = (Im\ (complex\_mul\ a\ (cnj\ b)) < (0::real))$

thm IM_COMPLEX_INV_GE_0:

$\forall z::(real,\ 2)\ cart.\ ((0::real) \le Im\ (complex\_inv\ z)) = (Im\ z \le (0::real))$

thm IM_COMPLEX_INV_LE_0:

$\forall z::(real,\ 2)\ cart.\ (Im\ (complex\_inv\ z) \le (0::real)) = ((0::real) \le Im\ z)$

thm IM_COMPLEX_INV_GT_0:

$\forall z::(real,\ 2)\ cart.\ ((0::real) < Im\ (complex\_inv\ z)) = (Im\ z < (0::real))$

thm IM_COMPLEX_INV_LT_0:

$\forall z::(real,\ 2)\ cart.\ (Im\ (complex\_inv\ z) < (0::real)) = ((0::real) < Im\ z)$

thm IM_COMPLEX_INV_EQ_0:

$\forall z::(real,\ 2)\ cart.\ (Im\ (complex\_inv\ z) = (0::real)) = (Im\ z = (0::real))$

thm REAL_SGN_IM_COMPLEX_DIV:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ sgn\ (Im\ (complex\_div\ w\ z)) = sgn\ (Im\ (complex\_mul\ w\ (cnj\ z)))$

thm COMPLEX_NORM_GE_RE_IM:

$\forall z::(real,\ 2)\ cart.\ |Re\ z| \le vector\_norm\ z \land |Im\ z| \le vector\_norm\ z$

thm COMPLEX_NORM_LE_RE_IM:

$\forall z::(real,\ 2)\ cart.\ vector\_norm\ z \le |Re\ z| + |Im\ z|$

thm DEF_csqrt:

$csqrt = (\lambda\_1868184::(real,\ 2)\ cart.\ if\ Im\ \_1868184 = (0::real)\ then\ if\ (0::real) \le Re\ \_1868184\ then\ complex\ (sqrt\ (Re\ \_1868184),\ 0::real)\ else\ complex\ (0::real,\ sqrt\ (-\ Re\ \_1868184))\ else\ complex\ (sqrt\ ((vector\_norm\ \_1868184 + Re\ \_1868184)\ /\ real\_of\_nat\ (2::nat)),\ Im\ \_1868184\ /\ |Im\ \_1868184| * sqrt\ ((vector\_norm\ \_1868184 - Re\ \_1868184)\ /\ real\_of\_nat\ (2::nat))))$

thm csqrt:

$\forall z::(real,\ 2)\ cart.\ csqrt\ z = (if\ Im\ z = (0::real)\ then\ if\ (0::real) \le Re\ z\ then\ complex\ (sqrt\ (Re\ z),\ 0::real)\ else\ complex\ (0::real,\ sqrt\ (-\ Re\ z))\ else\ complex$

(*sqrt* ((*vector_norm* *z* + *Re* *z*) / *real_of_nat* (*2*::*nat*)), *Im* *z* / |*Im* *z*| ∗ *sqrt* ((*vector_norm* *z* − *Re* *z*) / *real_of_nat* (*2*::*nat*))))

thm CSQRT:

∀ *z*::(*real*, *2*) *cart*. *complex_pow* (*csqrt* *z*) (*2*::*nat*) = *z*

thm CX_SQRT:

∀ *x*≥*0*::*real*. *Cx* (*sqrt* *x*) = *csqrt* (*Cx* *x*)

thm CSQRT_CX:

∀ *x*≥*0*::*real*. *csqrt* (*Cx* *x*) = *Cx* (*sqrt* *x*)

thm CSQRT_0:

csqrt (*Cx* (*0*::*real*)) = *Cx* (*0*::*real*)

thm CSQRT_1:

csqrt (*Cx* (*1*::*real*)) = *Cx* (*1*::*real*)

thm CSQRT_PRINCIPAL:

∀ *z*::(*real*, *2*) *cart*. (*0*::*real*) < *Re* (*csqrt* *z*) ∨ *Re* (*csqrt* *z*) = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im* (*csqrt* *z*)

thm RE_CSQRT:

∀ *z*::(*real*, *2*) *cart*. (*0*::*real*) ≤ *Re* (*csqrt* *z*)

thm CSQRT_UNIQUE:

∀ (*s*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *complex_pow* *s* (*2*::*nat*) = *z* ∧ ((*0*::*real*) < *Re* *s* ∨ *Re* *s* = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im* *s*) ⟶ *csqrt* *z* = *s*

thm POW_2_CSQRT:

∀ *z*::(*real*, *2*) *cart*. (*0*::*real*) < *Re* *z* ∨ *Re* *z* = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im* *z* ⟶ *csqrt* (*complex_pow* *z* (*2*::*nat*)) = *z*

thm CSQRT_EQ_0:

∀ *z*::(*real*, *2*) *cart*. (*csqrt* *z* = *Cx* (*0*::*real*)) = (*z* = *Cx* (*0*::*real*))

thm COMPLEX_CMUL:

∀ (*c*::*real*) *x*::(*real*, *2*) *cart*. % *c* *x* = *complex_mul* (*Cx* *c*) *x*

thm LINEAR_COMPLEX_MUL:

∀ *c*::(*real*, *2*) *cart*. *linear* (*complex_mul* *c*)

thm BILINEAR_COMPLEX_MUL:

bilinear *complex_mul*

thm RE_VSUM:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow Re\ (vsum\ s\ f) = sum\ s\ (\lambda x::?'a::type.\ Re\ (f\ x))$

thm IM_VSUM:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow Im\ (vsum\ s\ f) = sum\ s\ (\lambda x::?'a::type.\ Im\ (f\ x))$

thm VSUM_COMPLEX_LMUL:

$\forall (c::(real,\ 2)\ cart)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'a::type.\ complex\_mul\ c\ (f\ x)) = complex\_mul\ c\ (vsum\ s\ f)$

thm VSUM_COMPLEX_RMUL:

$\forall (c::(real,\ 2)\ cart)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda x::?'a::type.\ complex\_mul\ (f\ x)\ c) = complex\_mul\ (vsum\ s\ f)\ c$

thm VSUM_CX:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ s\ (\lambda a::?'a::type.\ Cx\ (f\ a)) = Cx\ (sum\ s\ f)$

thm CNJ_VSUM:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cnj\ (vsum\ s\ f) = vsum\ s\ (\lambda x::?'a::type.\ cnj\ (f\ x))$

thm VSUM_CX_NUMSEG:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ vsum\ (dotdot\ m\ n)\ (\lambda a::nat.\ Cx\ (f\ a)) = Cx\ (sum\ (dotdot\ m\ n)\ f)$

thm COMPLEX_SUB_POW:

$\forall (x::(real,\ 2)\ cart)\ (y::(real,\ 2)\ cart)\ n::nat.\ (1::nat) \le n \longrightarrow vector\_sub\ (complex\_pow\ x\ n)\ (complex\_pow\ y\ n) = complex\_mul\ (vector\_sub\ x\ y)\ (vsum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ complex\_mul\ (complex\_pow\ x\ i)\ (complex\_pow\ y\ (n - (1::nat) - i))))$

thm COMPLEX_SUB_POW_R1:

$\forall (x::(real,\ 2)\ cart)\ n::nat.\ (1::nat) \le n \longrightarrow vector\_sub\ (complex\_pow\ x\ n)\ (Cx\ (1::real)) = complex\_mul\ (vector\_sub\ x\ (Cx\ (1::real)))\ (vsum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (complex\_pow\ x))$

thm COMPLEX_SUB_POW_L1:

$\forall (x::(real,\ 2)\ cart)\ n::nat.\ (1::nat) \le n \longrightarrow vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ x\ n) = complex\_mul\ (vector\_sub\ (Cx\ (1::real))\ x)\ (vsum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (complex\_pow\ x))$

thm DEF_real:

$HOL\_Light\_Import.real = (\lambda\_1868306::(real,\ 2)\ cart.\ Im\ \_1868306 = (0::real))$

thm real:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z = (Im\ z = (0::real))$

thm REAL:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z = (Cx\ (Re\ z) = z)$

thm REAL_CNJ:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z = (cnj\ z = z)$

thm REAL_EXISTS:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z = (\exists x::real.\ z = Cx\ x)$

thm FORALL_REAL:

$(\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow (?P::(real,\ 2)\ cart \Rightarrow bool)\ z) = (\forall x::real.\ ?P\ (Cx\ x))$

thm EXISTS_REAL:

$(\exists z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \wedge (?P::(real,\ 2)\ cart \Rightarrow bool)\ z) = (\exists x::real.\ ?P\ (Cx\ x))$

thm REAL_CX:

$\forall x::real.\ HOL\_Light\_Import.real\ (Cx\ x)$

thm REAL_MUL_CX:

$\forall (x::real)\ z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ (complex\_mul\ (Cx\ x)\ z) = (x = (0::real) \vee HOL\_Light\_Import.real\ z)$

thm REAL_ADD:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ w \wedge HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (vector\_add\ w\ z)$

thm REAL_NEG:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (vector\_neg\ z)$

thm REAL_SUB:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ w \wedge HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (vector\_sub\ w\ z)$

thm REAL_MUL:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ w \wedge HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (complex\_mul\ w\ z)$

thm REAL_POW:

$\forall (z::(real,\ 2)\ cart)\ n::nat.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (complex\_pow\ z\ n)$

thm REAL_INV:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (complex\_inv\ z)$

thm REAL_INV_EQ:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ (complex\_inv\ z) = HOL\_Light\_Import.real\ z$

thm REAL_DIV:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ w \wedge HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (complex\_div\ w\ z)$

thm REAL_VSUM:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow HOL\_Light\_Import.real\ (f\ a)) \longrightarrow HOL\_Light\_Import.real\ (vsum\ s\ f)$

thm REAL_SEGMENT:

$\forall (a::(real, 2)\ cart)\ (b::(real, 2)\ cart)\ x::(real, 2)\ cart.\ IN\ x\ (closed\_segment\ [(a, b)]) \wedge HOL\_Light\_Import.real\ a \wedge HOL\_Light\_Import.real\ b \longrightarrow HOL\_Light\_Import.real\ x$

thm IN_SEGMENT_CX:

$\forall (a::real)\ (b::real)\ x::real.\ IN\ (Cx\ x)\ (closed\_segment\ [(Cx\ a, Cx\ b)]) = (a \leq x \wedge x \leq b \vee b \leq x \wedge x \leq a)$

thm IN_SEGMENT_CX_GEN:

$\forall (a::real)\ (b::real)\ x::(real, 2)\ cart.\ IN\ x\ (closed\_segment\ [(Cx\ a, Cx\ b)]) = (Im\ x = (0::real) \wedge (a \leq Re\ x \wedge Re\ x \leq b \vee b \leq Re\ x \wedge Re\ x \leq a))$

thm RE_POS_SEGMENT:

$\forall (a::(real, 2)\ cart)\ (b::(real, 2)\ cart)\ x::(real, 2)\ cart.\ IN\ x\ (closed\_segment\ [(a, b)]) \wedge (0::real) < Re\ a \wedge (0::real) < Re\ b \longrightarrow (0::real) < Re\ x$

thm CONVEX_REAL:

$convex\ HOL\_Light\_Import.real$

thm IMAGE_CX:

$\forall s::real \Rightarrow bool.\ IMAGE\ Cx\ s = GSPEC\ (\lambda GEN\%PVAR\%2284::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2284\ (HOL\_Light\_Import.real\ z \wedge IN\ (Re\ z)\ s)\ z)$

thm REAL_NORM:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow vector\_norm\ z = |Re\ z|$

thm REAL_NORM_POS:

$\forall\, z{::}(real,\, 2)\ cart.\ HOL\_Light\_Import.real\ z \wedge (0{::}real) \leq Re\ z \longrightarrow vector\_norm\ z = Re\ z$

thm COMPLEX_NORM_VSUM_SUM_RE:

$\forall\, (f{::}?'a{::}type \Rightarrow (real,\, 2)\ cart)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge (\forall\, x{::}?'a{::}type.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.real\ (f\ x) \wedge (0{::}real) \leq Re\ (f\ x)) \longrightarrow vector\_norm\ (vsum\ s\ f) = sum\ s\ (\lambda x{::}?'a{::}type.\ Re\ (f\ x))$

thm COMPLEX_NORM_VSUM_BOUND:

$\forall\, (s{::}?'a{::}type \Rightarrow bool)\ (f{::}?'a{::}type \Rightarrow (real,\, 2)\ cart)\ g{::}?'a{::}type \Rightarrow (real,\, 2)\ cart.\ FINITE\ s \wedge (\forall\, x{::}?'a{::}type.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.real\ (g\ x) \wedge vector\_norm\ (f\ x) \leq Re\ (g\ x)) \longrightarrow vector\_norm\ (vsum\ s\ f) \leq vector\_norm\ (vsum\ s\ g)$

thm COMPLEX_NORM_VSUM_BOUND_SUBSET:

$\forall\, (f{::}?'a{::}type \Rightarrow (real,\, 2)\ cart)\ (g{::}?'a{::}type \Rightarrow (real,\, 2)\ cart)\ (s{::}?'a{::}type \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ t\ s \wedge (\forall\, x{::}?'a{::}type.\ IN\ x\ s \longrightarrow HOL\_Light\_Import.real\ (g\ x) \wedge vector\_norm\ (f\ x) \leq Re\ (g\ x)) \longrightarrow vector\_norm\ (vsum\ t\ f) \leq vector\_norm\ (vsum\ s\ g)$

thm VSUM_GP_BASIC:

$\forall\, (x{::}(real,\, 2)\ cart)\ n{::}nat.\ complex\_mul\ (vector\_sub\ (Cx\ (1{::}real))\ x)\ (vsum\ (dotdot\ (0{::}nat)\ n)\ (complex\_pow\ x)) = vector\_sub\ (Cx\ (1{::}real))\ (complex\_pow\ x\ (Suc\ n))$

thm VSUM_GP_MULTIPLIED:

$\forall\, (x{::}(real,\, 2)\ cart)\ (m{::}nat)\ n{::}nat.\ m \leq n \longrightarrow complex\_mul\ (vector\_sub\ (Cx\ (1{::}real))\ x)\ (vsum\ (dotdot\ m\ n)\ (complex\_pow\ x)) = vector\_sub\ (complex\_pow\ x\ m)\ (complex\_pow\ x\ (Suc\ n))$

thm VSUM_GP:

$\forall\, (x{::}(real,\, 2)\ cart)\ (m{::}nat)\ n{::}nat.\ vsum\ (dotdot\ m\ n)\ (complex\_pow\ x) = (if\ n < m\ then\ Cx\ (0{::}real)\ else\ if\ x = Cx\ (1{::}real)\ then\ Cx\ (real\_of\_nat\ (n + (1{::}nat) - m))\ else\ complex\_div\ (vector\_sub\ (complex\_pow\ x\ m)\ (complex\_pow\ x\ (Suc\ n)))\ (vector\_sub\ (Cx\ (1{::}real))\ x))$

thm VSUM_GP_OFFSET:

$\forall\, (x{::}(real,\, 2)\ cart)\ (m{::}nat)\ n{::}nat.\ vsum\ (dotdot\ m\ (m + n))\ (complex\_pow\ x) = (if\ x = Cx\ (1{::}real)\ then\ vector\_add\ (Cx\ (real\_of\_nat\ n))\ (Cx\ (1{::}real))\ else\ complex\_mul\ (complex\_pow\ x\ m)\ (complex\_div\ (vector\_sub\ (Cx\ (1{::}real))\ (complex\_pow\ x\ (Suc\ n)))\ (vector\_sub\ (Cx\ (1{::}real))\ x)))$

thm COMPLEX_SUB_POLYFUN:

$\forall\, (a{::}nat \Rightarrow (real,\, 2)\ cart)\ (x{::}(real,\, 2)\ cart)\ (y{::}(real,\, 2)\ cart)\ n{::}nat.\ (1{::}nat) \leq n \longrightarrow vector\_sub\ (vsum\ (dotdot\ (0{::}nat)\ n)\ (\lambda i{::}nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ x\ i)))\ (vsum\ (dotdot\ (0{::}nat)\ n)\ (\lambda i{::}nat.\ complex\_mul\ (a\ i)$

(*complex_pow y i*))) = *complex_mul* (*vector_sub x y*) (*vsum* (*dotdot* (*0*::*nat*) (*n* − (*1*::*nat*))) (λ*j*::*nat*. *complex_mul* (*vsum* (*dotdot* (*j* + (*1*::*nat*)) *n*) (λ*i*::*nat*. *complex_mul* (*a i*) (*complex_pow y* (*i* − *j* − (*1*::*nat*))))) (*complex_pow x j*)))

thm COMPLEX_SUB_POLYFUN_ALT:

∀ (*a*::*nat* ⇒ (*real*, *2*) *cart*) (*x*::(*real*, *2*) *cart*) (*y*::(*real*, *2*) *cart*) *n*::*nat*. (*1*::*nat*) ≤ *n* ⟶ *vector_sub* (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*a i*) (*complex_pow x i*))) (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*a i*) (*complex_pow y i*))) = *complex_mul* (*vector_sub x y*) (*vsum* (*dotdot* (*0*::*nat*) (*n* − (*1*::*nat*))) (λ*j*::*nat*. *complex_mul* (*vsum* (*dotdot* (*0*::*nat*) (*n* − *j* − (*1*::*nat*))) (λ*k*::*nat*. *complex_mul* (*a* (*j* + (*k* + (*1*::*nat*)))) (*complex_pow y k*))) (*complex_pow x j*)))

thm COMPLEX_POLYFUN_LINEAR_FACTOR:

∀ (*a*::(*real*, *2*) *cart*) (*c*::*nat* ⇒ (*real*, *2*) *cart*) *n*::*nat*. ∃ *b*::*nat* ⇒ (*real*, *2*) *cart*. ∀ *z*::(*real*, *2*) *cart*. *vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*)) = *vector_add* (*complex_mul* (*vector_sub z a*) (*vsum* (*dotdot* (*0*::*nat*) (*n* − (*1*::*nat*))) (λ*i*::*nat*. *complex_mul* (*b i*) (*complex_pow z i*)))) (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow a i*)))

thm COMPLEX_POLYFUN_LINEAR_FACTOR_ROOT:

∀ (*a*::(*real*, *2*) *cart*) (*c*::*nat* ⇒ (*real*, *2*) *cart*) *n*::*nat*. *vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow a i*)) = *Cx* (*0*::*real*) ⟶ (∃ *b*::*nat* ⇒ (*real*, *2*) *cart*. ∀ *z*::(*real*, *2*) *cart*. *vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*)) = *complex_mul* (*vector_sub z a*) (*vsum* (*dotdot* (*0*::*nat*) (*n* − (*1*::*nat*))) (λ*i*::*nat*. *complex_mul* (*b i*) (*complex_pow z i*))))

thm COMPLEX_POLYFUN_EXTREMAL_LEMMA:

∀ (*c*::*nat* ⇒ (*real*, *2*) *cart*) (*n*::*nat*) *e*::*real*. (*0*::*real*) < *e* ⟶ (∃ *M*::*real*. ∀ *z*::(*real*, *2*) *cart*. *M* ≤ *vector_norm z* ⟶ *vector_norm* (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*))) ≤ *e* ∗ (*vector_norm z*)$^{n\ +\ (1::nat)}$)

thm COMPLEX_POLYFUN_EXTREMAL:

∀ (*c*::*nat* ⇒ (*real*, *2*) *cart*) *n*::*nat*. (∀ *k*::*nat*. *IN k* (*dotdot* (*1*::*nat*) *n*) ⟶ *c k* = *Cx* (*0*::*real*)) ∨ (∀ *B*::*real*. *eventually* (λ*z*::(*real*, *2*) *cart*. *B* ≤ *vector_norm* (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*)))) *at_infinity*)

thm COMPLEX_POLYFUN_ROOTBOUND:

∀ (*n*::*nat*) *c*::*nat* ⇒ (*real*, *2*) *cart*. ¬ (∀ *i*::*nat*. *IN i* (*dotdot* (*0*::*nat*) *n*) ⟶ *c i* = *Cx* (*0*::*real*)) ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%2288*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2288* (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*)) = *Cx* (*0*::*real*)) *z*)) ∧ *CARD* (*GSPEC* (λ*GEN%PVAR%2289*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2289* (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*c i*) (*complex_pow z i*)) = *Cx* (*0*::*real*)) *z*)) ≤ *n*

thm COMPLEX_POLYFUN_FINITE_ROOTS:

$\forall (n::nat)\ c::nat \Rightarrow (real,\ 2)\ cart.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2291::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2291\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ complex\_mul\ (c\ i)\ (complex\_pow\ x\ i)) = Cx\ (0::real))\ x)) = (\exists i::nat.\ IN\ i\ (dotdot\ (0::nat)\ n) \wedge c\ i \neq Cx\ (0::real))$

thm COMPLEX_POLYFUN_EQ_0:

$\forall (n::nat)\ c::nat \Rightarrow (real,\ 2)\ cart.\ (\forall z::(real,\ 2)\ cart.\ vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ complex\_mul\ (c\ i)\ (complex\_pow\ z\ i)) = Cx\ (0::real)) = (\forall i::nat.\ IN\ i\ (dotdot\ (0::nat)\ n) \longrightarrow c\ i = Cx\ (0::real))$

thm COMPLEX_POLYFUN_EQ_CONST:

$\forall (n::nat)\ (c::nat \Rightarrow (real,\ 2)\ cart)\ k::(real,\ 2)\ cart.\ (\forall z::(real,\ 2)\ cart.\ vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ complex\_mul\ (c\ i)\ (complex\_pow\ z\ i)) = k) = (c\ (0::nat) = k \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow c\ i = Cx\ (0::real)))$

thm cproduct:

$cproduct = iterate\ complex\_mul$

thm NEUTRAL_COMPLEX_MUL:

$neutral\ complex\_mul = Cx\ (1::real)$

thm MONOIDAL_COMPLEX_MUL:

$monoidal\ complex\_mul$

thm CPRODUCT_CLAUSES:

$(\forall f::?'b::type \Rightarrow (real,\ 2)\ cart.\ cproduct\ EMPTY\ f = Cx\ (1::real)) \wedge (\forall (x::?'a::type)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cproduct\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ cproduct\ s\ f\ else\ complex\_mul\ (f\ x)\ (cproduct\ s\ f)))$

thm CPRODUCT_CLAUSES_conjunct1:

$\forall (x::?'a::type)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cproduct\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ cproduct\ s\ f\ else\ complex\_mul\ (f\ x)\ (cproduct\ s\ f))$

thm CPRODUCT_CLAUSES_conjunct0:

$\forall f::?'a::type \Rightarrow (real,\ 2)\ cart.\ cproduct\ EMPTY\ f = Cx\ (1::real)$

thm CPRODUCT_EQ_0:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow (cproduct\ s\ f = Cx\ (0::real)) = (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x = Cx\ (0::real))$

thm CPRODUCT_INV:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cproduct\ s\ (\lambda x::?'a::type.\ complex\_inv\ (f\ x)) = complex\_inv\ (cproduct\ s\ f)$

thm CPRODUCT_MUL:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ (g::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cproduct\ s\ (\lambda x::?'a::type.\ complex\_mul\ (f\ x)\ (g\ x)) = complex\_mul\ (cproduct\ s\ f)\ (cproduct\ s\ g)$

thm CPRODUCT_EQ_1:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = Cx\ (1::real)) \longrightarrow cproduct\ s\ f = Cx\ (1::real)$

thm CPRODUCT_1:

$\forall s::?'a::type \Rightarrow bool.\ cproduct\ s\ (\lambda n::?'a::type.\ Cx\ (1::real)) = Cx\ (1::real)$

thm CPRODUCT_POW:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ (s::?'a::type \Rightarrow bool)\ n::nat.\ FINITE\ s \longrightarrow cproduct\ s\ (\lambda x::?'a::type.\ complex\_pow\ (f\ x)\ n) = complex\_pow\ (cproduct\ s\ f)\ n$

thm NORM_CPRODUCT:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow vector\_norm\ (cproduct\ s\ f) = product\ s\ (\lambda x::?'a::type.\ vector\_norm\ (f\ x))$

thm CPRODUCT_EQ:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ (g::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow cproduct\ s\ f = cproduct\ s\ g$

thm CLOSED_HALFSPACE_RE_GE:

$\forall b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%2293::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2293\ (b \le Re\ z)\ z))$

thm CLOSED_HALFSPACE_RE_LE:

$\forall b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%2294::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2294\ (Re\ z \le b)\ z))$

thm CLOSED_HALFSPACE_RE_EQ:

$\forall b::real.\ HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%2298::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2298\ (Re\ z = b)\ z))$

thm OPEN_HALFSPACE_RE_GT:

$\forall b::real.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%2301::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2301\ (b < Re\ z)\ z))$

thm OPEN_HALFSPACE_RE_LT:

$\forall b::real.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%2304::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2304\ (Re\ z < b)\ z))$

thm CLOSED_HALFSPACE_IM_GE:

∀ *b*::*real*. *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%2305*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2305* (*b* ≤ *Im z*) *z*))

thm CLOSED_HALFSPACE_IM_LE:

∀ *b*::*real*. *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%2306*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2306* (*Im z* ≤ *b*) *z*))

thm CLOSED_HALFSPACE_IM_EQ:

∀ *b*::*real*. *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%2310*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2310* (*Im z* = *b*) *z*))

thm OPEN_HALFSPACE_IM_GT:

∀ *b*::*real*. *HOL_Light_Import.open* (*GSPEC* (λ*GEN%PVAR%2313*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2313* (*b* < *Im z*) *z*))

thm OPEN_HALFSPACE_IM_LT:

∀ *b*::*real*. *HOL_Light_Import.open* (*GSPEC* (λ*GEN%PVAR%2316*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2316* (*Im z* < *b*) *z*))

thm CONVEX_HALFSPACE_RE_GE:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2317*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2317* (*b* ≤ *Re z*) *z*))

thm CONVEX_HALFSPACE_RE_GT:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2318*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2318* (*b* < *Re z*) *z*))

thm CONVEX_HALFSPACE_RE_LE:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2319*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2319* (*Re z* ≤ *b*) *z*))

thm CONVEX_HALFSPACE_RE_LT:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2320*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2320* (*Re z* < *b*) *z*))

thm CONVEX_HALFSPACE_IM_GE:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2321*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2321* (*b* ≤ *Im z*) *z*))

thm CONVEX_HALFSPACE_IM_GT:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2322*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2322* (*b* < *Im z*) *z*))

thm CONVEX_HALFSPACE_IM_LE:

∀ *b*::*real*. *convex* (*GSPEC* (λ*GEN%PVAR%2323*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2323* (*Im z* ≤ *b*) *z*))

thm CONVEX_HALFSPACE_IM_LT:

$\forall$ *b::real. convex (GSPEC (λGEN%PVAR%2324::(real, 2) cart. ∃ z::(real, 2) cart. SETSPEC GEN%PVAR%2324 (Im z < b) z))*

thm COMPLEX_IN_BALL_0:

$\forall$ *(v::(real, 2) cart) r::real. IN v (ball (Cx (0::real), r)) = (vector_norm v < r)*

thm COMPLEX_IN_CBALL_0:

$\forall$ *(v::(real, 2) cart) r::real. IN v (cball (Cx (0::real), r)) = (vector_norm v ≤ r)*

thm IN_BALL_RE:

$\forall$ *(x::(real, 2) cart) (z::(real, 2) cart) e::real. IN x (ball (z, e)) ⟶ |Re x − Re z| < e*

thm IN_BALL_IM:

$\forall$ *(x::(real, 2) cart) (z::(real, 2) cart) e::real. IN x (ball (z, e)) ⟶ |Im x − Im z| < e*

thm IN_CBALL_RE:

$\forall$ *(x::(real, 2) cart) (z::(real, 2) cart) e::real. IN x (cball (z, e)) ⟶ |Re x − Re z| ≤ e*

thm IN_CBALL_IM:

$\forall$ *(x::(real, 2) cart) (z::(real, 2) cart) e::real. IN x (cball (z, e)) ⟶ |Im x − Im z| ≤ e*

thm CLOSED_REAL_SET:

HOL_Light_Import.closed (GSPEC (λGEN%PVAR%2325::(real, 2) cart. ∃ z::(real, 2) cart. SETSPEC GEN%PVAR%2325 (HOL_Light_Import.real z) z))

thm CLOSED_REAL:

HOL_Light_Import.closed HOL_Light_Import.real

thm UNIFORM_LIM_COMPLEX_MUL:

$\forall$ *(net::?'b::type net) (P::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'b::type ⇒ (real, 2) cart) (g::?'a::type ⇒ ?'b::type ⇒ (real, 2) cart) (l::?'a::type ⇒ (real, 2) cart) (m::?'a::type ⇒ (real, 2) cart) (b1::real) b2::real. eventually (λx::?'b::type. ∀ n::?'a::type. P n ⟶ vector_norm (l n) ≤ b1) net ∧ eventually (λx::?'b::type. ∀ n::?'a::type. P n ⟶ vector_norm (m n) ≤ b2) net ∧ (∀ e>0::real. eventually (λx::?'b::type. ∀ n::?'a::type. P n ⟶ vector_norm (vector_sub (f n x) (l n)) < e) net) ∧ (∀ e>0::real. eventually (λx::?'b::type. ∀ n::?'a::type. P n ⟶ vector_norm (vector_sub (g n x) (m n)) < e) net) ⟶ (∀ e>0::real. eventually (λx::?'b::type. ∀ n::?'a::type. P n ⟶ vector_norm (vector_sub (complex_mul (f n x) (g n x)) (complex_mul (l n) (m n))) < e) net)*

thm UNIFORM_LIM_COMPLEX_INV:

$\forall$ (*net::?'b::type net*) (*P::?'a::type* $\Rightarrow$ *bool*) (*f::?'a::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) *b::real*. ($\forall$ *e*>*0::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*f n x*) (*l n*)) < *e*) *net*) $\wedge$ (*0::real*) < *b* $\wedge$ *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *b* $\leq$ *vector_norm* (*l n*)) *net* $\longrightarrow$ ($\forall$ *e*>*0::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*complex_inv* (*f n x*)) (*complex_inv* (*l n*))) < *e*) *net*)

thm UNIFORM_LIM_COMPLEX_DIV:

$\forall$ (*net::?'b::type net*) (*P::?'a::type* $\Rightarrow$ *bool*) (*f::?'a::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ (*real*, *2*) *cart*) (*g::?'a::type* $\Rightarrow$ *?'b::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*m::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*b1::real*) *b2::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*l n*) $\leq$ *b1*) *net* $\wedge$ (*0::real*) < *b2* $\wedge$ *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *b2* $\leq$ *vector_norm* (*m n*)) *net* $\wedge$ ($\forall$ *e*>*0::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*f n x*) (*l n*)) < *e*) *net*) $\wedge$ ($\forall$ *e*>*0::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*g n x*) (*m n*)) < *e*) *net*) $\longrightarrow$ ($\forall$ *e*>*0::real*. *eventually* (λ*x::?'b::type*. $\forall$ *n::?'a::type*. *P n* $\longrightarrow$ *vector_norm* (*vector_sub* (*complex_div* (*f n x*) (*g n x*)) (*complex_div* (*l n*) (*m n*))) < *e*) *net*)

thm LIM_COMPLEX_MUL:

$\forall$ (*net::?'a::type net*) (*f::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*g::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::*(*real*, *2*) *cart*) *m::*(*real*, *2*) *cart*. *-->* *f l net* $\wedge$ *-->* *g m net* $\longrightarrow$ *-->* (λ*x::?'a::type*. *complex_mul* (*f x*) (*g x*)) (*complex_mul l m*) *net*

thm LIM_COMPLEX_INV:

$\forall$ (*net::?'c::type net*) (*f::?'c::type* $\Rightarrow$ (*real*, *2*) *cart*) (*g::?'b::type*) (*l::*(*real*, *2*) *cart*) *m::?'a::type*. *-->* *f l net* $\wedge$ *l* $\neq$ *Cx* (*0::real*) $\longrightarrow$ *-->* (λ*x::?'c::type*. *complex_inv* (*f x*)) (*complex_inv l*) *net*

thm LIM_COMPLEX_DIV:

$\forall$ (*net::?'a::type net*) (*f::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*g::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::*(*real*, *2*) *cart*) *m::*(*real*, *2*) *cart*. *-->* *f l net* $\wedge$ *-->* *g m net* $\wedge$ *m* $\neq$ *Cx* (*0::real*) $\longrightarrow$ *-->* (λ*x::?'a::type*. *complex_div* (*f x*) (*g x*)) (*complex_div l m*) *net*

thm LIM_COMPLEX_POW:

$\forall$ (*net::?'a::type net*) (*f::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::*(*real*, *2*) *cart*) *n::nat*. *-->* *f l net* $\longrightarrow$ *-->* (λ*x::?'a::type*. *complex_pow* (*f x*) *n*) (*complex_pow l n*) *net*

thm LIM_COMPLEX_LMUL:

$\forall$ (*f::?'a::type* $\Rightarrow$ (*real*, *2*) *cart*) (*l::*(*real*, *2*) *cart*) *c::*(*real*, *2*) *cart*. *-->* *f l* (*?net::?'a::type net*) $\longrightarrow$ *-->* (λ*x::?'a::type*. *complex_mul c* (*f x*)) (*complex_mul c l*) *?net*

thm LIM_COMPLEX_RMUL:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ c::(real,\ 2)\ cart.\ -->\ f\ l\ (?net::?'a::type\ net) \longrightarrow -->\ (\lambda x::?'a::type.\ complex\_mul\ (f\ x)\ c)\ (complex\_mul\ l\ c)\ ?net$

thm LIM_NULL_COMPLEX_NEG:

$\forall (net::?'a::type\ net)\ f::?'a::type \Rightarrow (real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ vector\_neg\ (f\ x))\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_ADD:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ g::?'a::type \Rightarrow (real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \wedge -->\ g\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ vector\_add\ (f\ x)\ (g\ x))\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_SUB:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ g::?'a::type \Rightarrow (real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \wedge -->\ g\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ vector\_sub\ (f\ x)\ (g\ x))\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_MUL:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ g::?'a::type \Rightarrow (real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \wedge -->\ g\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ complex\_mul\ (f\ x)\ (g\ x))\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_LMUL:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ c::(real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ complex\_mul\ c\ (f\ x))\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_RMUL:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ c::(real,\ 2)\ cart.\ -->\ f\ (Cx\ (0::real))\ net \longrightarrow -->\ (\lambda x::?'a::type.\ complex\_mul\ (f\ x)\ c)\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_POW:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ n::nat.\ -->\ f\ (Cx\ (0::real))\ net \wedge n \neq (0::nat) \longrightarrow -->\ (\lambda x::?'a::type.\ complex\_pow\ (f\ x)\ n)\ (Cx\ (0::real))\ net$

thm LIM_NULL_COMPLEX_BOUND:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ g::?'a::type \Rightarrow (real,\ 2)\ cart.\ eventually\ (\lambda n::?'a::type.\ vector\_norm\ (f\ n) \leq vector\_norm\ (g\ n))\ (?net::?'a::type\ net) \wedge -->\ g\ (Cx\ (0::real))\ ?net \longrightarrow -->\ f\ (Cx\ (0::real))\ ?net$

thm SUMS_COMPLEX_0:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ (\forall n::nat.\ IN\ n\ s \longrightarrow f\ n = Cx\ (0::real)) \longrightarrow sums\ f\ (Cx\ (0::real))\ s$

thm LIM_RE_UBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ b::real.\ \neg\ trivial\_limit\ net \wedge --> f\ l\ net \wedge eventually\ (\lambda x::?'a::type.\ Re\ (f\ x) \leq b)\ net \longrightarrow Re\ l \leq b$

thm LIM_RE_LBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ b::real.\ \neg\ trivial\_limit\ net \wedge --> f\ l\ net \wedge eventually\ (\lambda x::?'a::type.\ b \leq Re\ (f\ x))\ net \longrightarrow b \leq Re\ l$

thm LIM_IM_UBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ b::real.\ \neg\ trivial\_limit\ net \wedge --> f\ l\ net \wedge eventually\ (\lambda x::?'a::type.\ Im\ (f\ x) \leq b)\ net \longrightarrow Im\ l \leq b$

thm LIM_IM_LBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ b::real.\ \neg\ trivial\_limit\ net \wedge --> f\ l\ net \wedge eventually\ (\lambda x::?'a::type.\ b \leq Im\ (f\ x))\ net \longrightarrow b \leq Im\ l$

thm SERIES_COMPLEX_LMUL:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow sums\ (\lambda x::nat.\ complex\_mul\ c\ (f\ x))\ (complex\_mul\ c\ l)\ s$

thm SERIES_COMPLEX_RMUL:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow sums\ (\lambda x::nat.\ complex\_mul\ (f\ x)\ c)\ (complex\_mul\ l\ c)\ s$

thm SERIES_COMPLEX_DIV:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow sums\ (\lambda x::nat.\ complex\_div\ (f\ x)\ c)\ (complex\_div\ l\ c)\ s$

thm SUMMABLE_COMPLEX_LMUL:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ f \longrightarrow summable\ s\ (\lambda x::nat.\ complex\_mul\ c\ (f\ x))$

thm SUMMABLE_COMPLEX_RMUL:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ f \longrightarrow summable\ s\ (\lambda x::nat.\ complex\_mul\ (f\ x)\ c)$

thm SUMMABLE_COMPLEX_DIV:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ summable\ s\ f \longrightarrow summable\ s\ (\lambda x::nat.\ complex\_div\ (f\ x)\ c)$

thm CONTINUOUS_COMPLEX_MUL:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*) *g*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*. *continuous f net* $\wedge$ *continuous g net* $\longrightarrow$ *continuous* (λ*x*::*?'a*::*type*. *complex_mul* (*f x*) (*g x*)) *net*

thm CONTINUOUS_COMPLEX_INV:

$\forall$ (*net*::*?'a*::*type net*) *f*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*. *continuous f net* $\wedge$ *f* (*netlimit net*) $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::*?'a*::*type*. *complex_inv* (*f x*)) *net*

thm CONTINUOUS_COMPLEX_DIV:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*) *g*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*. *continuous f net* $\wedge$ *continuous g net* $\wedge$ *g* (*netlimit net*) $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::*?'a*::*type*. *complex_div* (*f x*) (*g x*)) *net*

thm CONTINUOUS_COMPLEX_POW:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ (*real*, *2*) *cart*) *n*::*nat*. *continuous f net* $\longrightarrow$ *continuous* (λ*x*::*?'a*::*type*. *complex_pow* (*f x*) *n*) *net*

thm CONTINUOUS_COMPLEX_INV_WITHIN:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous f* (*within* (*at a*) *s*) $\wedge$ *f a* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_inv* (*f x*)) (*within* (*at a*) *s*)

thm CONTINUOUS_COMPLEX_INV_AT:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous f* (*at a*) $\wedge$ *f a* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_inv* (*f x*)) (*at a*)

thm CONTINUOUS_COMPLEX_DIV_WITHIN:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous f* (*within* (*at a*) *s*) $\wedge$ *continuous g* (*within* (*at a*) *s*) $\wedge$ *g a* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_div* (*f x*) (*g x*)) (*within* (*at a*) *s*)

thm CONTINUOUS_COMPLEX_DIV_AT:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous f* (*at a*) $\wedge$ *continuous g* (*at a*) $\wedge$ *g a* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_div* (*f x*) (*g x*)) (*at a*)

thm CONTINUOUS_ON_COMPLEX_MUL:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*g*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *continuous_on f s* $\wedge$ *continuous_on g s* $\longrightarrow$ *continuous_on* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_mul* (*f x*) (*g x*)) *s*

thm CONTINUOUS_ON_COMPLEX_DIV:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, ?'a::type)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge continuous\_on\ g\ s \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow g\ x \neq Cx\ (0::real)) \longrightarrow continuous\_on\ (\lambda x::(real, ?'a::type)\ cart.\ complex\_div\ (f\ x)\ (g\ x))\ s$

thm CONTINUOUS_ON_COMPLEX_POW:

$\forall (f::(real, ?'a::type)\ cart \Rightarrow (real, 2)\ cart)\ (n::nat)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \longrightarrow continuous\_on\ (\lambda x::(real, ?'a::type)\ cart.\ complex\_pow\ (f\ x)\ n)\ s$

thm LIM_CONTINUOUS:

$\forall (net::?'b::type\ net)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ l::(real, ?'a::type)\ cart.\ continuous\ f\ net \wedge f\ (netlimit\ net) = l \longrightarrow --> f\ l\ net$

thm CONTINUOUS_AT_CX_NORM:

$\forall z::(real, ?'a::type)\ cart.\ continuous\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (vector\_norm\ z))\ (at\ z)$

thm CONTINUOUS_WITHIN_CX_NORM:

$\forall (z::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (vector\_norm\ z))\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CX_NORM:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (vector\_norm\ z))\ s$

thm CONTINUOUS_AT_CX_DOT:

$\forall (c::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ continuous\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (dot\ c\ z))\ (at\ z)$

thm CONTINUOUS_WITHIN_CX_DOT:

$\forall (c::(real, ?'a::type)\ cart)\ (z::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ continuous\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (dot\ c\ z))\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CX_DOT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ c::(real, ?'a::type)\ cart.\ continuous\_on\ (\lambda z::(real, ?'a::type)\ cart.\ Cx\ (dot\ c\ z))\ s$

thm LINEAR_CX_RE:

$linear\ (Cx \circ Re)$

thm CONTINUOUS_AT_CX_RE:

$\forall z::(real, 2)\ cart.\ continuous\ (Cx \circ Re)\ (at\ z)$

thm CONTINUOUS_ON_CX_RE:

$\forall\, s{::}(real,\ 2)\ cart \Rightarrow bool.\ continuous\_on\ (Cx \circ Re)\ s$

thm LINEAR_CX_IM:

$linear\ (Cx \circ Im)$

thm CONTINUOUS_AT_CX_IM:

$\forall\, z{::}(real,\ 2)\ cart.\ continuous\ (Cx \circ Im)\ (at\ z)$

thm CONTINUOUS_ON_CX_IM:

$\forall\, s{::}(real,\ 2)\ cart \Rightarrow bool.\ continuous\_on\ (Cx \circ Im)\ s$

thm DEF_has_complex_derivative:

$has\_complex\_derivative = (\lambda(\_1870770{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ \_1870771{::}(real,\ 2)\ cart.\ has\_derivative\ \_1870770\ (complex\_mul\ \_1870771))$

thm has_complex_derivative:

$\forall\, (f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'{::}(real,\ 2)\ cart)\ net{::}(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ f\ f'\ net = has\_derivative\ f\ (complex\_mul\ f')\ net$

thm DEF_complex_differentiable:

$complex\_differentiable = (\lambda(\_1870791{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ \_1870792{::}(real,\ 2)\ cart\ net.\ \exists\, f'{::}(real,\ 2)\ cart.\ has\_complex\_derivative\ \_1870791\ f'\ \_1870792)$

thm complex_differentiable:

$\forall\, (f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ net{::}(real,\ 2)\ cart\ net.\ complex\_differentiable\ f\ net = (\exists\, f'{::}(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ net)$

thm DEF_complex_derivative:

$complex\_derivative = (\lambda(\_1870803{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ \_1870804{::}(real,\ 2)\ cart.\ SOME\ f'{::}(real,\ 2)\ cart.\ has\_complex\_derivative\ \_1870803\ f'\ (at\ \_1870804))$

thm complex_derivative:

$\forall\, (f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ x{::}(real,\ 2)\ cart.\ complex\_derivative\ f\ x = (SOME\ f'{::}(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x))$

thm DEF_higher_complex_derivative:

$higher\_complex\_derivative = (SOME\ higher\_complex\_derivative{::}nat \Rightarrow nat \Rightarrow ((real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart) \Rightarrow (real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ \forall\, \_1871199{::}nat.\ (\forall\, f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ higher\_complex\_derivative\ \_1871199\ (0{::}nat)\ f = f) \land (\forall\, (f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ n{::}nat.\ higher\_complex\_derivative\ \_1871199\ (Suc\ n)\ f = complex\_derivative\ (higher\_complex\_derivative\ \_1871199\ n\ f)))\ (56{::}nat)$

thm higher_complex_derivative:

$higher\_complex\_derivative\ (0{::}nat)\ (?f{::}(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart) = ?f \land (\forall\, n{::}nat.\ higher\_complex\_derivative\ (Suc\ n)\ ?f = complex\_derivative\ (higher\_complex\_derivative\ n\ ?f))$

thm DEF_holomorphic_on:

$holomorphic\_on = (\lambda(\_1871200::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ \_1871201::(real, 2)\ cart \Rightarrow bool.\ \forall x::(real, 2)\ cart.\ IN\ x\ \_1871201 \longrightarrow (\exists f'::(real, 2)\ cart.\ has\_complex\_derivative\ \_1871200\ f'\ (within\ (at\ x)\ \_1871201)))$

thm holomorphic_on:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s = (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow (\exists f'::(real, 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s)))$

thm HOLOMORPHIC_ON_DIFFERENTIABLE:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s = (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow complex\_differentiable\ f\ (within\ (at\ x)\ s))$

thm HOLOMORPHIC_ON_OPEN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \longrightarrow holomorphic\_on\ f\ s = (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow (\exists f'::(real, 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x)))$

thm HOLOMORPHIC_ON_IMP_DIFFERENTIABLE_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ x::(real, 2)\ cart.\ holomorphic\_on\ f\ s \wedge IN\ x\ s \longrightarrow complex\_differentiable\ f\ (within\ (at\ x)\ s)$

thm HOLOMORPHIC_ON_IMP_DIFFERENTIABLE_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ x::(real, 2)\ cart.\ holomorphic\_on\ f\ s \wedge HOL\_Light\_Import.open\ s \wedge IN\ x\ s \longrightarrow complex\_differentiable\ f\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_IMP_CONTINUOUS_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ x::(real, 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow continuous\ f\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_IMP_CONTINUOUS_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (x::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow continuous\ f\ (within\ (at\ x)\ s)$

thm COMPLEX_DIFFERENTIABLE_IMP_CONTINUOUS_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ x::(real, 2)\ cart.\ complex\_differentiable\ f\ (at\ x) \longrightarrow continuous\ f\ (at\ x)$

thm HOLOMORPHIC_ON_IMP_CONTINUOUS_ON:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \longrightarrow continuous\_on\ f\ s$

thm HOLOMORPHIC_ON_SUBSET:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ t::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow holomorphic\_on\ f\ t$

thm HAS_COMPLEX_DERIVATIVE_WITHIN_SUBSET:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ (t::(real, 2)\ cart \Rightarrow bool)\ x::(real, 2)\ cart.\ has\_complex\_derivative\ f\ (?f'::(real, 2)\ cart)\ (within\ (at\ x)\ s) \wedge SUBSET\ t\ s \longrightarrow has\_complex\_derivative\ f\ ?f'\ (within\ (at\ x)\ t)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_SUBSET:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ t::(real, 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ (?x::(real, 2)\ cart))\ s) \wedge SUBSET\ t\ s \longrightarrow complex\_differentiable\ f\ (within\ (at\ ?x)\ t)$

thm HAS_COMPLEX_DERIVATIVE_AT_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (x::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_WITHIN_OPEN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (a::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ IN\ a\ s \wedge HOL\_Light\_Import.open\ s \longrightarrow has\_complex\_derivative\ f\ f'\ (within\ (at\ a)\ s) = has\_complex\_derivative\ f\ f'\ (at\ a)$

thm COMPLEX_DIFFERENTIABLE_AT_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \longrightarrow complex\_differentiable\ f\ (within\ (at\ z)\ s)$

thm HAS_COMPLEX_DERIVATIVE_TRANSFORM_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (x::(real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ x\ s \wedge (\forall x'::(real, 2)\ cart.\ IN\ x'\ s \wedge distance\ (x', x) < d \longrightarrow f\ x' = g\ x') \wedge has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ g\ f'\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_TRANSFORM_WITHIN_OPEN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ z\ s \wedge (\forall w::(real, 2)\ cart.\ IN\ w\ s \longrightarrow f\ w = g\ w) \wedge has\_complex\_derivative\ f\ f'\ (at\ z) \longrightarrow has\_complex\_derivative\ g\ f'\ (at\ z)$

thm HAS_COMPLEX_DERIVATIVE_TRANSFORM_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (x::(real, 2)\ cart)\ d::real.\ (0::real) < d \wedge (\forall x'::(real, 2)\ cart.\ distance\ (x', x) < d \longrightarrow f\ x' = g\ x') \wedge has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ g\ f'\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_ZERO_CONSTANT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ convex\ s \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow has\_complex\_derivative\ f\ (Cx\ (0::real))\ (within\ (at\ x)\ s)) \longrightarrow (\exists c::(real,\ 2)\ cart.\ \forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = c)$

thm HAS_COMPLEX_DERIVATIVE_ZERO_UNIQUE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ (c::(real,\ 2)\ cart)\ a::(real,\ 2)\ cart.\ convex\ s \wedge IN\ a\ s \wedge f\ a = c \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow has\_complex\_derivative\ f\ (Cx\ (0::real))\ (within\ (at\ x)\ s)) \longrightarrow (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = c)$

thm HAS_COMPLEX_DERIVATIVE_ZERO_CONNECTED_CONSTANT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge connected\ s \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow has\_complex\_derivative\ f\ (Cx\ (0::real))\ (at\ x)) \longrightarrow (\exists c::(real,\ 2)\ cart.\ \forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = c)$

thm HAS_COMPLEX_DERIVATIVE_ZERO_CONNECTED_UNIQUE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ (c::(real,\ 2)\ cart)\ a::(real,\ 2)\ cart.\ HOL\_Light\_Import.open\ s \wedge connected\ s \wedge IN\ a\ s \wedge f\ a = c \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow has\_complex\_derivative\ f\ (Cx\ (0::real))\ (at\ x)) \longrightarrow (\forall x::(real,\ 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = c)$

thm COMPLEX_DIFF_CHAIN_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge has\_complex\_derivative\ g\ g'\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow has\_complex\_derivative\ (g \circ f)\ (complex\_mul\ g'\ f')\ (within\ (at\ x)\ s)$

thm COMPLEX_DIFF_CHAIN_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \wedge has\_complex\_derivative\ g\ g'\ (at\ (f\ x)) \longrightarrow has\_complex\_derivative\ (g \circ f)\ (complex\_mul\ g'\ f')\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_CHAIN:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ (\forall x::(real,\ 2)\ cart.\ P\ x \longrightarrow has\_complex\_derivative\ g\ ((?g'::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ x)\ (at\ x)) \longrightarrow (\forall (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ (?f'::(real,\ 2)\ cart)\ (within\ (at\ x)\ s) \wedge P\ (f\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ g\ (f\ x))\ (complex\_mul\ ?f'\ (?g'\ (f\ x)))\ (within\ (at\ x)\ s)) \wedge (\forall x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ ?f'\ (at\ x) \wedge P\ (f\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ g\ (f\ x))\ (complex\_mul\ ?f'\ (?g'\ (f\ x)))\ (at\ x))$

thm HAS_COMPLEX_DERIVATIVE_CHAIN_UNIV:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ (\forall x::(real,\ 2)\ cart.\ has\_complex\_derivative\ g\ ((?g'::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ x)\ (at$

$x)) \longrightarrow (\forall (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ (?f'::(real,\ 2)\ cart)\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ g\ (f\ x))\ (complex\_mul\ ?f'\ (?g'\ (f\ x)))\ (within\ (at\ x)\ s)) \wedge (\forall x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ ?f'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ g\ (f\ x))\ (complex\_mul\ ?f'\ (?g'\ (f\ x)))\ (at\ x))$

thm COMPLEX_DERIVATIVE_UNIQUE_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (z::(real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ f''::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ z) \wedge has\_complex\_derivative\ f\ f''\ (at\ z) \longrightarrow f' = f''$

thm higher_complex_derivative_conjunct1:

$\forall n::nat.\ higher\_complex\_derivative\ (Suc\ n)\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart) = complex\_derivative\ (higher\_complex\_derivative\ n\ ?f)$

thm higher_complex_derivative_conjunct0:

$higher\_complex\_derivative\ (0::nat)\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart) = ?f$

thm HIGHER_COMPLEX_DERIVATIVE_1:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ higher\_complex\_derivative\ (1::nat)\ f\ z = complex\_derivative\ f\ z$

thm HAS_COMPLEX_DERIVATIVE_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ a::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ (?f'::(real,\ 2)\ cart)\ (within\ (at\ a)\ s) = --> (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (vector\_sub\ (f\ x)\ (f\ a))\ (vector\_sub\ x\ a))\ ?f'\ (within\ (at\ a)\ s)$

thm HAS_COMPLEX_DERIVATIVE_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ a::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ (?f'::(real,\ 2)\ cart)\ (at\ a) = --> (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (vector\_sub\ (f\ x)\ (f\ a))\ (vector\_sub\ x\ a))\ ?f'\ (at\ a)$

thm HAS_DERIVATIVE_COMPLEX_CMUL:

$\forall (net::(real,\ 2)\ cart\ net)\ c::(real,\ 2)\ cart.\ has\_derivative\ (complex\_mul\ c)\ (complex\_mul\ c)\ net$

thm HAS_COMPLEX_DERIVATIVE_LINEAR:

$\forall (net::(real,\ 2)\ cart\ net)\ c::(real,\ 2)\ cart.\ has\_complex\_derivative\ (complex\_mul\ c)\ c\ net$

thm HAS_COMPLEX_DERIVATIVE_LMUL_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ c\ (f\ x))\ (complex\_mul\ c\ f')\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_LMUL_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ c\ (f\ x))\ (complex\_mul\ c\ f')\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_RMUL_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ (f\ x)\ c)\ (complex\_mul\ f'\ c)\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_RMUL_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ (f\ x)\ c)\ (complex\_mul\ f'\ c)\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_CDIV_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (f\ x)\ c)\ (complex\_div\ f'\ c)\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_CDIV_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (c::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::?'a::type.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (f\ x)\ c)\ (complex\_div\ f'\ c)\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_ID:

$\forall net::(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ x)\ (Cx\ (1::real))\ net$

thm HAS_COMPLEX_DERIVATIVE_CONST:

$\forall (c::(real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ c)\ (Cx\ (0::real))\ net$

thm HAS_COMPLEX_DERIVATIVE_NEG:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ f\ f'\ net \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ vector\_neg\ (f\ x))\ (vector\_neg\ f')\ net$

thm HAS_COMPLEX_DERIVATIVE_ADD:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ f\ f'\ net \wedge has\_complex\_derivative\ g\ g'\ net \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ vector\_add\ (f\ x)\ (g\ x))\ (vector\_add\ f'\ g')\ net$

thm HAS_COMPLEX_DERIVATIVE_SUB:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ has\_complex\_derivative\ f\ f'\ net \wedge has\_complex\_derivative\ g\ g'\ net \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ vector\_sub\ (f\ x)\ (g\ x))\ (vector\_sub\ f'\ g')\ net$

thm HAS_COMPLEX_DERIVATIVE_MUL_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge has\_complex\_derivative\ g\ g'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ (f\ x)\ (g\ x))\ (vector\_add\ (complex\_mul\ (f\ x)\ g')\ (complex\_mul\ f'\ (g\ x)))\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_MUL_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \wedge has\_complex\_derivative\ g\ g'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_mul\ (f\ x)\ (g\ x))\ (vector\_add\ (complex\_mul\ (f\ x)\ g')\ (complex\_mul\ f'\ (g\ x)))\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_POW_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ n::nat.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_pow\ (f\ x)\ n)\ (complex\_mul\ (Cx\ (real\_of\_nat\ n))\ (complex\_mul\ (complex\_pow\ (f\ x)\ (n - (1::nat)))\ f'))\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_POW_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ n::nat.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_pow\ (f\ x)\ n)\ (complex\_mul\ (Cx\ (real\_of\_nat\ n))\ (complex\_mul\ (complex\_pow\ (f\ x)\ (n - (1::nat)))\ f'))\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_INV_BASIC:

$\forall x::(real,\ 2)\ cart.\ x \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ complex\_inv\ (vector\_neg\ (complex\_inv\ (complex\_pow\ x\ (2::nat))))\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_INV_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge f\ x \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_inv\ (f\ x))\ (complex\_div\ (vector\_neg\ f')\ (complex\_pow\ (f\ x)\ (2::nat)))\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_INV_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \wedge f\ x \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_inv\ (f\ x))\ (complex\_div\ (vector\_neg\ f')\ (complex\_pow\ (f\ x)\ (2::nat)))\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_DIV_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ has\_complex\_derivative\ f\ f'\ (within\ (at\ x)\ s) \wedge has\_complex\_derivative\ g\ g'\ (within\ (at\ x)\ s) \wedge g\ x \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (f\ x)\ (g\ x))\ (complex\_div\ (vector\_sub\ (complex\_mul\ f'\ (g\ x))\ (complex\_mul\ (f\ x)\ g'))\ (complex\_pow\ (g\ x)\ (2::nat)))\ (within\ (at\ x)\ s)$

thm HAS_COMPLEX_DERIVATIVE_DIV_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g'::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \wedge has\_complex\_derivative\ g\ g'\ (at\ x) \wedge g\ x \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ complex\_div\ (f\ x)\ (g\ x))\ (complex\_div\ (vector\_sub\ (complex\_mul\ f'\ (g\ x))\ (complex\_mul\ (f\ x)\ g'))\ (complex\_pow\ (g\ x)\ (2::nat)))\ (at\ x)$

thm HAS_COMPLEX_DERIVATIVE_VSUM:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (net::(real,\ 2)\ cart\ net)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow has\_complex\_derivative\ (f\ a)\ ((?f'::?'a::type \Rightarrow (real,\ 2)\ cart)\ a)\ net) \longrightarrow has\_complex\_derivative\ (\lambda x::(real,\ 2)\ cart.\ vsum\ s\ (\lambda a::?'a::type.\ f\ a\ x))\ (vsum\ s\ ?f')\ net$

thm COMPLEX_DIFFERENTIABLE_LINEAR:

$complex\_differentiable\ (complex\_mul\ (?c::(real,\ 2)\ cart))\ (?p::(real,\ 2)\ cart\ net)$

thm COMPLEX_DIFFERENTIABLE_CONST:

$\forall (c::(real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ c)\ net$

thm COMPLEX_DIFFERENTIABLE_ID:

$\forall\ net::(real,\ 2)\ cart\ net.\ complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ z)\ net$

thm COMPLEX_DIFFERENTIABLE_NEG:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ complex\_differentiable\ f\ net \longrightarrow complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ vector\_neg\ (f\ z))\ net$

thm COMPLEX_DIFFERENTIABLE_ADD:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ net::(real,\ 2)\ cart\ net.\ complex\_differentiable\ f\ net \wedge complex\_differentiable\ g\ net \longrightarrow complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ vector\_add\ (f\ z)\ (g\ z))\ net$

thm COMPLEX_DIFFERENTIABLE_SUB:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ net::(real, 2)\ cart\ net.\ complex\_differentiable\ f\ net \land complex\_differentiable\ g\ net \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ vector\_sub\ (f\ z)\ (g\ z))\ net$

thm COMPLEX_DIFFERENTIABLE_INV_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (z::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ z)\ s) \land f\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ complex\_inv\ (f\ z))\ (within\ (at\ z)\ s)$

thm COMPLEX_DIFFERENTIABLE_MUL_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (z::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ z)\ s) \land complex\_differentiable\ g\ (within\ (at\ z)\ s) \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ complex\_mul\ (f\ z)\ (g\ z))\ (within\ (at\ z)\ s)$

thm COMPLEX_DIFFERENTIABLE_DIV_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (z::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ z)\ s) \land complex\_differentiable\ g\ (within\ (at\ z)\ s) \land g\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ complex\_div\ (f\ z)\ (g\ z))\ (within\ (at\ z)\ s)$

thm COMPLEX_DIFFERENTIABLE_POW_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (n::nat)\ (z::(real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ z)\ s) \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ complex\_pow\ (f\ z)\ n)\ (within\ (at\ z)\ s)$

thm COMPLEX_DIFFERENTIABLE_TRANSFORM_WITHIN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (x::(real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ d::real.\ (0::real) < d \land IN\ x\ s \land (\forall x'::(real, 2)\ cart.\ IN\ x'\ s \land distance\ (x', x) < d \longrightarrow f\ x' = g\ x') \land complex\_differentiable\ f\ (within\ (at\ x)\ s) \longrightarrow complex\_differentiable\ g\ (within\ (at\ x)\ s)$

thm HOLOMORPHIC_TRANSFORM:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \land holomorphic\_on\ f\ s \longrightarrow holomorphic\_on\ g\ s$

thm HOLOMORPHIC_EQ:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow holomorphic\_on\ f\ s = holomorphic\_on\ g\ s$

thm COMPLEX_DIFFERENTIABLE_INV_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \land f\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ (\lambda z::(real, 2)\ cart.\ complex\_inv\ (f\ z))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_MUL_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ z) \longrightarrow complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ complex\_mul\ (f\ z)\ (g\ z))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_DIV_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ z) \wedge g\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ complex\_div\ (f\ z)\ (g\ z))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_POW_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (n::nat)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \longrightarrow complex\_differentiable\ (\lambda z::(real,\ 2)\ cart.\ complex\_pow\ (f\ z)\ n)\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_TRANSFORM_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ d::real.\ (0::real) < d \wedge (\forall x'::(real,\ 2)\ cart.\ distance\ (x',\ x) < d \longrightarrow f\ x' = g\ x') \wedge complex\_differentiable\ f\ (at\ x) \longrightarrow complex\_differentiable\ g\ (at\ x)$

thm COMPLEX_DIFFERENTIABLE_COMPOSE_WITHIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ complex\_differentiable\ f\ (within\ (at\ x)\ s) \wedge complex\_differentiable\ g\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow complex\_differentiable\ (g \circ f)\ (within\ (at\ x)\ s)$

thm COMPLEX_DIFFERENTIABLE_COMPOSE_AT:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (x::(real,\ 2)\ cart)\ s::?'a::type.\ complex\_differentiable\ f\ (at\ x) \wedge complex\_differentiable\ g\ (at\ (f\ x)) \longrightarrow complex\_differentiable\ (g \circ f)\ (at\ x)$

thm HOLOMORPHIC_ON_LINEAR:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ c::(real,\ 2)\ cart.\ holomorphic\_on\ (complex\_mul\ c)\ s$

thm HOLOMORPHIC_ON_CONST:

$\forall (c::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ c)\ s$

thm HOLOMORPHIC_ON_ID:

$\forall s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ z)\ s$

thm HOLOMORPHIC_ON_COMPOSE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ (IMAGE\ f\ s) \longrightarrow holomorphic\_on\ (g \circ f)\ s$

thm HOLOMORPHIC_ON_NEG:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ vector\_neg\ (f\ z))\ s$

thm HOLOMORPHIC_ON_ADD:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ s \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ vector\_add\ (f\ z)\ (g\ z))\ s$

thm HOLOMORPHIC_ON_SUB:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ s \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ vector\_sub\ (f\ z)\ (g\ z))\ s$

thm HOLOMORPHIC_ON_MUL:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ s \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ complex\_mul\ (f\ z)\ (g\ z))\ s$

thm HOLOMORPHIC_ON_INV:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \longrightarrow f\ z \neq Cx\ (0::real)) \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ complex\_inv\ (f\ z))\ s$

thm HOLOMORPHIC_ON_DIV:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ s \wedge (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \longrightarrow g\ z \neq Cx\ (0::real)) \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ complex\_div\ (f\ z)\ (g\ z))\ s$

thm HOLOMORPHIC_ON_POW:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ n::nat.\ holomorphic\_on\ f\ s \longrightarrow holomorphic\_on\ (\lambda z::(real,\ 2)\ cart.\ complex\_pow\ (f\ z)\ n)\ s$

thm HOLOMORPHIC_ON_VSUM:

$\forall (f::?'a::type \Rightarrow (real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge (\forall a::?'a::type.\ IN\ a\ k \longrightarrow holomorphic\_on\ (f\ a)\ s) \longrightarrow holomorphic\_on\ (\lambda x::(real,\ 2)\ cart.\ vsum\ k\ (\lambda a::?'a::type.\ f\ a\ x))\ s$

thm HOLOMORPHIC_ON_COMPOSE_GEN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ t::(real,\ 2)\ cart \Rightarrow bool.\ holomorphic\_on\ f\ s \wedge holomorphic\_on\ g\ t \wedge (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \longrightarrow IN\ (f\ z)\ t) \longrightarrow holomorphic\_on\ (g \circ f)\ s$

thm HAS_COMPLEX_DERIVATIVE_DERIVATIVE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (f'::(real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ f'\ (at\ x) \longrightarrow complex\_derivative\ f\ x = f'$

thm HAS_COMPLEX_DERIVATIVE_DIFFERENTIABLE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ x::(real,\ 2)\ cart.\ has\_complex\_derivative\ f\ (complex\_derivative\ f\ x)\ (at\ x) = complex\_differentiable\ f\ (at\ x)$

thm COMPLEX_DIFFERENTIABLE_COMPOSE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ (f\ z)) \longrightarrow complex\_differentiable\ (g \circ f)\ (at\ z)$

thm COMPLEX_DERIVATIVE_CHAIN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ (f\ z)) \longrightarrow complex\_derivative\ (g \circ f)\ z = complex\_mul\ (complex\_derivative\ g\ (f\ z))\ (complex\_derivative\ f\ z)$

thm COMPLEX_DERIVATIVE_LINEAR:

$\forall c::(real,\ 2)\ cart.\ complex\_derivative\ (complex\_mul\ c) = (\lambda z::(real,\ 2)\ cart.\ c)$

thm COMPLEX_DERIVATIVE_ID:

$complex\_derivative\ (\lambda w::(real,\ 2)\ cart.\ w) = (\lambda z::(real,\ 2)\ cart.\ Cx\ (1::real))$

thm COMPLEX_DERIVATIVE_CONST:

$\forall c::(real,\ 2)\ cart.\ complex\_derivative\ (\lambda w::(real,\ 2)\ cart.\ c) = (\lambda z::(real,\ 2)\ cart.\ Cx\ (0::real))$

thm COMPLEX_DERIVATIVE_ADD:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ z) \longrightarrow complex\_derivative\ (\lambda w::(real,\ 2)\ cart.\ vector\_add\ (f\ w)\ (g\ w))\ z = vector\_add\ (complex\_derivative\ f\ z)\ (complex\_derivative\ g\ z)$

thm COMPLEX_DERIVATIVE_SUB:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \wedge complex\_differentiable\ g\ (at\ z) \longrightarrow complex\_derivative\ (\lambda w::(real,\ 2)\ cart.\ vector\_sub\ (f\ w)\ (g\ w))\ z = vector\_sub\ (complex\_derivative\ f\ z)\ (complex\_derivative\ g\ z)$

thm COMPLEX_DERIVATIVE_MUL:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) z::(real, 2) cart. complex_differentiable f (at z) ∧ complex_differentiable g (at z) ⟶ complex_derivative (λw::(real, 2) cart. complex_mul (f w) (g w)) z = vector_add (complex_mul (f z) (complex_derivative g z)) (complex_mul (complex_derivative f z) (g z))

thm COMPLEX_DERIVATIVE_LMUL:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (c::(real, 2) cart) z::(real, 2) cart. complex_differentiable f (at z) ⟶ complex_derivative (λw::(real, 2) cart. complex_mul c (f w)) z = complex_mul c (complex_derivative f z)

thm COMPLEX_DERIVATIVE_RMUL:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (c::(real, 2) cart) z::(real, 2) cart. complex_differentiable f (at z) ⟶ complex_derivative (λw::(real, 2) cart. complex_mul (f w) c) z = complex_mul (complex_derivative f z) c

thm COMPLEX_DERIVATIVE_TRANSFORM_WITHIN_OPEN:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) s::(real, 2) cart ⇒ bool. HOL_Light_Import.open s ∧ holomorphic_on f s ∧ holomorphic_on g s ∧ IN (?z::(real, 2) cart) s ∧ (∀w::(real, 2) cart. IN w s ⟶ f w = g w) ⟶ complex_derivative f ?z = complex_derivative g ?z

thm COMPLEX_DERIVATIVE_COMPOSE_LINEAR:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (c::(real, 2) cart) z::(real, 2) cart. complex_differentiable f (at (complex_mul c z)) ⟶ complex_derivative (λw::(real, 2) cart. f (complex_mul c w)) z = complex_mul c (complex_derivative f (complex_mul c z))

thm DEF_analytic_on:

analytic_on = (λ(_1873215::(real, 2) cart ⇒ (real, 2) cart) _1873216::(real, 2) cart ⇒ bool. ∀x::(real, 2) cart. IN x _1873216 ⟶ (∃e>0::real. holomorphic_on _1873215 (ball (x, e))))

thm analytic_on:

∀(s::(real, 2) cart ⇒ bool) f::(real, 2) cart ⇒ (real, 2) cart. analytic_on f s = (∀x::(real, 2) cart. IN x s ⟶ (∃e>0::real. holomorphic_on f (ball (x, e))))

thm ANALYTIC_IMP_HOLOMORPHIC:

∀(f::(real, 2) cart ⇒ (real, 2) cart) s::(real, 2) cart ⇒ bool. analytic_on f s ⟶ holomorphic_on f s

thm ANALYTIC_ON_OPEN:

∀(f::(real, 2) cart ⇒ (real, 2) cart) s::(real, 2) cart ⇒ bool. HOL_Light_Import.open s ⟶ analytic_on f s = holomorphic_on f s

thm ANALYTIC_ON_IMP_DIFFERENTIABLE_AT:

∀(f::(real, 2) cart ⇒ (real, 2) cart) (s::(real, 2) cart ⇒ bool) x::(real, 2) cart. analytic_on f s ∧ IN x s ⟶ complex_differentiable f (at x)

thm ANALYTIC_ON_SUBSET:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ t::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow analytic\_on\ f\ t$

thm ANALYTIC_ON_UNION:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ t::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ (HOL\_Light\_Import.UNION\ s\ t) = (analytic\_on\ f\ s \wedge analytic\_on\ f\ t)$

thm ANALYTIC_ON_UNIONS:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::((real,\ 2)\ cart \Rightarrow bool) \Rightarrow bool.\ analytic\_on\ f\ (UNIONS\ s) = (\forall t::(real,\ 2)\ cart \Rightarrow bool.\ IN\ t\ s \longrightarrow analytic\_on\ f\ t)$

thm ANALYTIC_ON_HOLOMORPHIC:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s = (\exists t::(real,\ 2)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ t \wedge SUBSET\ s\ t \wedge holomorphic\_on\ f\ t)$

thm ANALYTIC_ON_LINEAR:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ c::(real,\ 2)\ cart.\ analytic\_on\ (complex\_mul\ c)\ s$

thm ANALYTIC_ON_CONST:

$\forall (c::(real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ (\lambda z::(real,\ 2)\ cart.\ c)\ s$

thm ANALYTIC_ON_ID:

$\forall s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ (\lambda z::(real,\ 2)\ cart.\ z)\ s$

thm ANALYTIC_ON_COMPOSE:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ (IMAGE\ f\ s) \longrightarrow analytic\_on\ (g \circ f)\ s$

thm ANALYTIC_ON_COMPOSE_GEN:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::(real,\ 2)\ cart \Rightarrow bool)\ t::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ t \wedge (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \longrightarrow IN\ (f\ z)\ t) \longrightarrow analytic\_on\ (g \circ f)\ s$

thm ANALYTIC_ON_NEG:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \longrightarrow analytic\_on\ (\lambda z::(real,\ 2)\ cart.\ vector\_neg\ (f\ z))\ s$

thm ANALYTIC_ON_ADD:

$\forall (f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (g::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ s \longrightarrow analytic\_on\ (\lambda z::(real,\ 2)\ cart.\ vector\_add\ (f\ z)\ (g\ z))\ s$

thm ANALYTIC_ON_SUB:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ s \longrightarrow analytic\_on\ (\lambda z::(real, 2)\ cart.\ vector\_sub\ (f\ z)\ (g\ z))\ s$

thm ANALYTIC_ON_MUL:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ s \longrightarrow analytic\_on\ (\lambda z::(real, 2)\ cart.\ complex\_mul\ (f\ z)\ (g\ z))\ s$

thm ANALYTIC_ON_INV:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge (\forall z::(real, 2)\ cart.\ IN\ z\ s \longrightarrow f\ z \neq Cx\ (0::real)) \longrightarrow analytic\_on\ (\lambda z::(real, 2)\ cart.\ complex\_inv\ (f\ z))\ s$

thm ANALYTIC_ON_DIV:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s \wedge analytic\_on\ g\ s \wedge (\forall z::(real, 2)\ cart.\ IN\ z\ s \longrightarrow g\ z \neq Cx\ (0::real)) \longrightarrow analytic\_on\ (\lambda z::(real, 2)\ cart.\ complex\_div\ (f\ z)\ (g\ z))\ s$

thm ANALYTIC_ON_POW:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ n::nat.\ analytic\_on\ f\ s \longrightarrow analytic\_on\ (\lambda z::(real, 2)\ cart.\ complex\_pow\ (f\ z)\ n)\ s$

thm ANALYTIC_ON_VSUM:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge (\forall a::?'a::type.\ IN\ a\ k \longrightarrow analytic\_on\ (f\ a)\ s) \longrightarrow analytic\_on\ (\lambda x::(real, 2)\ cart.\ vsum\ k\ (\lambda a::?'a::type.\ f\ a\ x))\ s$

thm ANALYTIC_AT_BALL:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ analytic\_on\ f\ (INSERT\ z\ EMPTY) = (\exists e>0::real.\ holomorphic\_on\ f\ (ball\ (z,\ e)))$

thm ANALYTIC_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ analytic\_on\ f\ (INSERT\ z\ EMPTY) = (\exists s::(real, 2)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ s \wedge IN\ z\ s \wedge holomorphic\_on\ f\ s)$

thm ANALYTIC_ON_ANALYTIC_AT:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ s::(real, 2)\ cart \Rightarrow bool.\ analytic\_on\ f\ s = (\forall z::(real, 2)\ cart.\ IN\ z\ s \longrightarrow analytic\_on\ f\ (INSERT\ z\ EMPTY))$

thm ANALYTIC_AT_TWO:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ (analytic\_on\ f\ (INSERT\ z\ EMPTY) \wedge analytic\_on\ g\ (INSERT\ z$

EMPTY)) = (∃ *s*::(*real*, *2*) *cart* ⇒ *bool*. *HOL_Light_Import.open s* ∧ *IN z s* ∧ *holomorphic_on f s* ∧ *holomorphic_on g s*)

thm ANALYTIC_AT_ADD:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *analytic_on* (λ*w*::(*real*, *2*) *cart*. *vector_add* (*f w*) (*g w*)) (*INSERT z EMPTY*)

thm ANALYTIC_AT_SUB:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *analytic_on* (λ*w*::(*real*, *2*) *cart*. *vector_sub* (*f w*) (*g w*)) (*INSERT z EMPTY*)

thm ANALYTIC_AT_MUL:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *analytic_on* (λ*w*::(*real*, *2*) *cart*. *complex_mul* (*f w*) (*g w*)) (*INSERT z EMPTY*)

thm ANALYTIC_AT_POW:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*n*::*nat*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ⟶ *analytic_on* (λ*w*::(*real*, *2*) *cart*. *complex_pow* (*f w*) *n*) (*INSERT z EMPTY*)

thm COMPLEX_DERIVATIVE_ADD_AT:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *complex_derivative* (λ*w*::(*real*, *2*) *cart*. *vector_add* (*f w*) (*g w*)) *z* = *vector_add* (*complex_derivative f z*) (*complex_derivative g z*)

thm COMPLEX_DERIVATIVE_SUB_AT:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *complex_derivative* (λ*w*::(*real*, *2*) *cart*. *vector_sub* (*f w*) (*g w*)) *z* = *vector_sub* (*complex_derivative f z*) (*complex_derivative g z*)

thm COMPLEX_DERIVATIVE_MUL_AT:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ∧ *analytic_on g* (*INSERT z EMPTY*) ⟶ *complex_derivative* (λ*w*::(*real*, *2*) *cart*. *complex_mul* (*f w*) (*g w*)) *z* = *vector_add* (*complex_mul* (*f z*) (*complex_derivative g z*)) (*complex_mul* (*complex_derivative f z*) (*g z*))

thm COMPLEX_DERIVATIVE_LMUL_AT:

∀ (*f*::(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*c*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *analytic_on f* (*INSERT z EMPTY*) ⟶ *complex_derivative* (λ*w*::(*real*, *2*) *cart*. *complex_mul c* (*f w*)) *z* = *complex_mul c* (*complex_derivative f z*)

thm COMPLEX_DERIVATIVE_RMUL_AT:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (c::(real, 2) cart) z::(real, 2) cart. analytic_on f (INSERT z EMPTY) ⟶ complex_derivative (λw::(real, 2) cart. complex_mul (f w) c) z = complex_mul (complex_derivative f z) c

thm HAS_VECTOR_DERIVATIVE_REAL_COMPLEX:

has_complex_derivative (?f::(real, 2) cart ⇒ (real, 2) cart) (?f′::(real, 2) cart) (at (Cx (HOL_Light_Import.drop (?a::(real, unit) cart)))) ⟶ has_vector_derivative (λx::(real, unit) cart. ?f (Cx (HOL_Light_Import.drop x))) ?f′ (at ?a)

thm HAS_COMPLEX_DERIVATIVE_SEQUENCE:

∀ (s::(real, 2) cart ⇒ bool) (f::nat ⇒ (real, 2) cart ⇒ (real, 2) cart) (f′::nat ⇒ (real, 2) cart ⇒ (real, 2) cart) g′::(real, 2) cart ⇒ (real, 2) cart. convex s ∧ (∀ (n::nat) x::(real, 2) cart. IN x s ⟶ has_complex_derivative (f n) (f′ n x) (within (at x) s)) ∧ (∀ e>0::real. ∃ N::nat. ∀ (n::nat) x::(real, 2) cart. N ≤ n ∧ IN x s ⟶ vector_norm (vector_sub (f′ n x) (g′ x)) ≤ e) ∧ (∃ (x::(real, 2) cart) l::(real, 2) cart. IN x s ∧ −−> (λn::nat. f n x) l sequentially) ⟶ (∃ g::(real, 2) cart ⇒ (real, 2) cart. ∀ x::(real, 2) cart. IN x s ⟶ −−> (λn::nat. f n x) (g x) sequentially ∧ has_complex_derivative g (g′ x) (within (at x) s))

thm HAS_COMPLEX_DERIVATIVE_SERIES:

∀ (s::(real, 2) cart ⇒ bool) (f::nat ⇒ (real, 2) cart ⇒ (real, 2) cart) (f′::nat ⇒ (real, 2) cart ⇒ (real, 2) cart) (g′::(real, 2) cart ⇒ (real, 2) cart) k::nat ⇒ bool. convex s ∧ (∀ (n::nat) x::(real, 2) cart. IN x s ⟶ has_complex_derivative (f n) (f′ n x) (within (at x) s)) ∧ (∀ e>0::real. ∃ N::nat. ∀ (n::nat) x::(real, 2) cart. N ≤ n ∧ IN x s ⟶ vector_norm (vector_sub (vsum (HOL_Light_Import.INTER k (dotdot (0::nat) n)) (λi::nat. f′ i x)) (g′ x)) ≤ e) ∧ (∃ (x::(real, 2) cart) l::(real, 2) cart. IN x s ∧ sums (λn::nat. f n x) l k) ⟶ (∃ g::(real, 2) cart ⇒ (real, 2) cart. ∀ x::(real, 2) cart. IN x s ⟶ sums (λn::nat. f n x) (g x) k ∧ has_complex_derivative g (g′ x) (within (at x) s))

thm COMPLEX_DIFFERENTIABLE_BOUND:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (f′::(real, 2) cart ⇒ (real, 2) cart) (s::(real, 2) cart ⇒ bool) B::real. convex s ∧ (∀ x::(real, 2) cart. IN x s ⟶ has_complex_derivative f (f′ x) (within (at x) s) ∧ vector_norm (f′ x) ≤ B) ⟶ (∀ (x::(real, 2) cart) y::(real, 2) cart. IN x s ∧ IN y s ⟶ vector_norm (vector_sub (f x) (f y)) ≤ B ∗ vector_norm (vector_sub x y))

thm HAS_COMPLEX_DERIVATIVE_INVERSE_BASIC:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (f′::(real, 2) cart) (t::(real, 2) cart ⇒ bool) y::(real, 2) cart. has_complex_derivative f f′ (at (g y)) ∧ f′ ≠ Cx (0::real) ∧ continuous g (at y) ∧ HOL_Light_Import.open t ∧ IN y t ∧ (∀ z::(real, 2) cart. IN z t ⟶ f (g z) = z) ⟶ has_complex_derivative g (complex_inv f′) (at y)

thm HAS_COMPLEX_DERIVATIVE_INVERSE_STRONG:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ x::(real, 2)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ x\ s \wedge continuous\_on\ f\ s \wedge (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \wedge has\_complex\_derivative\ f\ f'\ (at\ x) \wedge f' \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ g\ (complex\_inv\ f')\ (at\ (f\ x))$

thm HAS_COMPLEX_DERIVATIVE_INVERSE_STRONG_X:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (g::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (f'::(real, 2)\ cart)\ (s::(real, 2)\ cart \Rightarrow bool)\ y::(real, 2)\ cart.\ HOL\_Light\_Import.open\ s \wedge IN\ (g\ y)\ s \wedge continuous\_on\ f\ s \wedge (\forall x::(real, 2)\ cart.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \wedge has\_complex\_derivative\ f\ f'\ (at\ (g\ y)) \wedge f' \neq Cx\ (0::real) \wedge f\ (g\ y) = y \longrightarrow has\_complex\_derivative\ g\ (complex\_inv\ f')\ (at\ y)$

thm COMPLEX_BASIS:

$basis\ (1::nat) = Cx\ (1::real) \wedge basis\ (2::nat) = ii$

thm COMPLEX_DIFFERENTIABLE_IMP_DIFFERENTIABLE:

$\forall (net::(real, 2)\ cart\ net)\ f::(real, 2)\ cart \Rightarrow (real, 2)\ cart.\ complex\_differentiable\ f\ net \longrightarrow differentiable\ f\ net$

thm COMPLEX_BASIS_conjunct1:

$basis\ (2::nat) = ii$

thm COMPLEX_BASIS_conjunct0:

$basis\ (1::nat) = Cx\ (1::real)$

thm CAUCHY_RIEMANN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_differentiable\ f\ (at\ z) = (differentiable\ f\ (at\ z) \wedge \$\ (\$\ (jacobian\ f\ (at\ z))\ (1::nat))\ (1::nat) = \$\ (\$\ (jacobian\ f\ (at\ z))\ (2::nat))\ (2::nat) \wedge \$\ (\$\ (jacobian\ f\ (at\ z))\ (1::nat))\ (2::nat) = -\ \$\ (\$\ (jacobian\ f\ (at\ z))\ (2::nat))\ (1::nat))$

thm COMPLEX_DERIVATIVE_JACOBIAN:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_differentiable\ f\ (at\ z) \longrightarrow complex\_derivative\ f\ z = complex\ (\$\ (\$\ (jacobian\ f\ (at\ z))\ (1::nat))\ (1::nat), \$\ (\$\ (jacobian\ f\ (at\ z))\ (2::nat))\ (1::nat))$

thm COMPLEX_DIFFERENTIABLE_EQ_CONFORMAL:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ z::(real, 2)\ cart.\ (complex\_differentiable\ f\ (at\ z) \wedge complex\_derivative\ f\ z \neq Cx\ (0::real)) = (differentiable\ f\ (at\ z) \wedge (\exists a::real.\ a \neq (0::real) \wedge rotation\_matrix\ (\%\%\ a\ (jacobian\ f\ (at\ z)))))$

thm COMPLEX_TAYLOR:

$\forall (f::nat \Rightarrow (real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (n::nat)\ (s::(real, 2)\ cart \Rightarrow bool)\ B::real.\ convex\ s \wedge (\forall (i::nat)\ x::(real, 2)\ cart.\ IN\ x\ s \wedge i \leq n \longrightarrow has\_complex\_derivative\ (f\ i)\ (f\ (i + (1::nat))\ x)\ (within\ (at\ x)\ s)) \wedge (\forall x::(real,$

2) cart. IN x s ⟶ vector_norm (f (n + (1::nat)) x) ≤ B) ⟶ (∀ (w::(real, 2) cart) z::(real, 2) cart. IN w s ∧ IN z s ⟶ vector_norm (vector_sub (f (0::nat) z) (vsum (dotdot (0::nat) n) (λi::nat. complex_mul (f i w) (complex_div (complex_pow (vector_sub z w) i) (Cx (real_of_nat (fact i))))))) ≤ B ∗ ((vector_norm (vector_sub z w))$^{n + (1::nat)}$ */ real_of_nat (fact n)))*

thm COMPLEX_MVT:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (f′::(real, 2) cart ⇒ (real, 2) cart) (s::(real, 2) cart ⇒ bool) B::real. convex s ∧ (∀ z::(real, 2) cart. IN z s ⟶ has_complex_derivative f (f′ z) (within (at z) s)) ∧ (∀ z::(real, 2) cart. IN z s ⟶ vector_norm (f′ z) ≤ B) ⟶ (∀ (w::(real, 2) cart) z::(real, 2) cart. IN w s ∧ IN z s ⟶ vector_norm (vector_sub (f z) (f w)) ≤ B ∗ vector_norm (vector_sub z w))

thm COMPLEX_MVT_LINE:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (f′::(real, 2) cart ⇒ (real, 2) cart) (w::(real, 2) cart) z::(real, 2) cart. (∀ u::(real, 2) cart. IN u (closed_segment [(w, z)]) ⟶ has_complex_derivative f (f′ u) (at u)) ⟶ (∃ u::(real, 2) cart. IN u (closed_segment [(w, z)]) ∧ Re (f z) − Re (f w) = Re (complex_mul (f′ u) (vector_sub z w)))

thm COMPLEX_TAYLOR_MVT:

∀ (f::nat ⇒ (real, 2) cart ⇒ (real, 2) cart) (w::(real, 2) cart) (z::(real, 2) cart) n::nat. (∀ (i::nat) x::(real, 2) cart. IN x (closed_segment [(w, z)]) ∧ i ≤ n ⟶ has_complex_derivative (f i) (f (i + (1::nat)) x) (at x)) ⟶ (∃ u::(real, 2) cart. IN u (closed_segment [(w, z)]) ∧ Re (f (0::nat) z) = Re (vector_add (vsum (dotdot (0::nat) n) (λi::nat. complex_mul (f i w) (complex_div (complex_pow (vector_sub z w) i) (Cx (real_of_nat (fact i)))))) (complex_mul (complex_mul (f (n + (1::nat)) u) (complex_div (complex_pow (vector_sub z u) n) (Cx (real_of_nat (fact n))))) (vector_sub z w))))

thm LIM_CNJ:

∀ (net::?′a::type net) (f::?′a::type ⇒ (real, 2) cart) l::(real, 2) cart. −−> (λx::?′a::type. cnj (f x)) (cnj l) net = −−> f l net

thm SUMS_CNJ:

∀ (net::nat ⇒ bool) (f::nat ⇒ (real, 2) cart) l::(real, 2) cart. sums (λx::nat. cnj (f x)) (cnj l) net = sums f l net

thm REAL_LIM:

∀ (net::?′a::type net) (f::?′a::type ⇒ (real, 2) cart) l::(real, 2) cart. −−> f l net ∧ ¬ trivial_limit net ∧ (∃ b::?′a::type. (∃ a::?′a::type. netord net a b) ∧ (∀ a::?′a::type. netord net a b ⟶ HOL_Light_Import.real (f a))) ⟶ HOL_Light_Import.real l

thm REAL_LIM_SEQUENTIALLY:

∀ (f::nat ⇒ (real, 2) cart) l::(real, 2) cart. −−> f l sequentially ∧ (∃ N::nat. ∀ n≥N. HOL_Light_Import.real (f n)) ⟶ HOL_Light_Import.real l

thm REAL_SERIES:

∀(f::nat ⇒ (real, 2) cart) (l::(real, 2) cart) s::nat ⇒ bool. sums f l s ∧ (∀n::nat. HOL_Light_Import.real (f n)) ⟶ HOL_Light_Import.real l

thm LIM_NULL_COMPARISON_COMPLEX:

∀(net::?'a::type net) (f::?'a::type ⇒ (real, 2) cart) g::?'a::type ⇒ (real, 2) cart. eventually (λx::?'a::type. vector_norm (f x) ≤ vector_norm (g x)) net ∧ --> g (Cx (0::real)) net ⟶ --> f (Cx (0::real)) net

thm LIM_NULL_COMPARISON_COMPLEX_RE:

∀(net::?'a::type net) (f::?'a::type ⇒ (real, 2) cart) g::?'a::type ⇒ (real, 2) cart. eventually (λx::?'a::type. vector_norm (f x) ≤ Re (g x)) net ∧ --> g (Cx (0::real)) net ⟶ --> f (Cx (0::real)) net

thm SERIES_COMPARISON_COMPLEX:

∀(f::nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, 2) cart) s::nat ⇒ bool. summable s g ∧ (∀n::nat. IN n s ⟶ HOL_Light_Import.real (g n) ∧ (0::real) ≤ Re (g n)) ∧ (∃N::nat. ∀n::nat. N ≤ n ∧ IN n s ⟶ vector_norm (f n) ≤ vector_norm (g n)) ⟶ summable s f

thm SERIES_COMPARISON_UNIFORM_COMPLEX:

∀(f::?'b::type ⇒ nat ⇒ (real, ?'a::type) cart) (g::nat ⇒ (real, 2) cart) (P::?'b::type ⇒ bool) s::nat ⇒ bool. summable s g ∧ (∀n::nat. IN n s ⟶ HOL_Light_Import.real (g n) ∧ (0::real) ≤ Re (g n)) ∧ (∃N::nat. ∀(n::nat) x::?'b::type. N ≤ n ∧ IN n s ∧ P x ⟶ vector_norm (f x n) ≤ vector_norm (g n)) ⟶ (∃l::?'b::type ⇒ (real, ?'a::type) cart. ∀e>0::real. ∃N::nat. ∀(n::nat) x::?'b::type. N ≤ n ∧ P x ⟶ distance (vsum (HOL_Light_Import.INTER s (dotdot (0::nat) n)) (f x), l x) < e)

thm SUMMABLE_SUBSET_COMPLEX:

∀(x::nat ⇒ (real, 2) cart) (s::nat ⇒ bool) t::nat ⇒ bool. (∀n::nat. IN n s ⟶ HOL_Light_Import.real (x n) ∧ (0::real) ≤ Re (x n)) ∧ summable s x ∧ SUBSET t s ⟶ summable t x

thm SERIES_ABSCONV_IMP_CONV:

∀(x::nat ⇒ (real, ?'a::type) cart) k::nat ⇒ bool. summable k (λn::nat. Cx (vector_norm (x n))) ⟶ summable k x

thm SUMS_GP:

∀(n::nat) z::(real, 2) cart. vector_norm z < (1::real) ⟶ sums (complex_pow z) (complex_div (complex_pow z n) (vector_sub (Cx (1::real)) z)) (from n)

thm SUMMABLE_GP:

∀(z::(real, 2) cart) k::nat ⇒ bool. vector_norm z < (1::real) ⟶ summable k (complex_pow z)

thm SERIES_DIRICHLET_COMPLEX_GEN:

∀ (f::nat ⇒ (real, 2) cart) (g::nat ⇒ (real, 2) cart) (N::?'a::type) (k::nat) (m::nat) (p::nat) l::(real, 2) cart. bounded (GSPEC (λGEN%PVAR%2328::(real, 2) cart. ∃ n::nat. SETSPEC GEN%PVAR%2328 (IN n HOL_Light_Import.UNIV) (vsum (dotdot m n) f))) ∧ summable (from p) (λn::nat. Cx (vector_norm (vector_sub (g (n + (1::nat))) (g n)))) ∧ --> (λn::nat. complex_mul (vsum (dotdot (1::nat) n) f) (g (n + (1::nat)))) l sequentially ⟶ summable (from k) (λn::nat. complex_mul (f n) (g n))

thm SERIES_DIRICHLET_COMPLEX:

∀ (f::nat ⇒ (real, 2) cart) (g::nat ⇒ (real, 2) cart) (N::nat) (k::nat) m::nat. bounded (GSPEC (λGEN%PVAR%2329::(real, 2) cart. ∃ n::nat. SETSPEC GEN%PVAR%2329 (IN n HOL_Light_Import.UNIV) (vsum (dotdot m n) f))) ∧ (∀ n::nat. HOL_Light_Import.real (g n)) ∧ (∀ n≥N. Re (g (n + (1::nat))) ≤ Re (g n)) ∧ --> g (Cx (0::real)) sequentially ⟶ summable (from k) (λn::nat. complex_mul (f n) (g n))

thm SERIES_DIRICHLET_COMPLEX_VERY_EXPLICIT:

∀ (f::nat ⇒ (real, 2) cart) (g::nat ⇒ (real, 2) cart) (B::real) p::nat. (0::real) < B ∧ (1::nat) ≤ p ∧ (∀ (m::nat) n::nat. p ≤ m ⟶ vector_norm (vsum (dotdot m n) f) ≤ B) ∧ (∀ n≥p. HOL_Light_Import.real (g n) ∧ (0::real) ≤ Re (g n)) ∧ (∀ n≥p. Re (g (n + (1::nat))) ≤ Re (g n)) ⟶ (∀ (m::nat) n::nat. p ≤ m ⟶ vector_norm (vsum (dotdot m n) (λk::nat. complex_mul (f k) (g k))) ≤ real_of_nat (2::nat) * (B * vector_norm (g m)))

thm SERIES_DIRICHLET_COMPLEX_EXPLICIT:

∀ (f::nat ⇒ (real, 2) cart) (g::nat ⇒ (real, 2) cart) (p::nat) q::nat. (1::nat) ≤ p ∧ bounded (GSPEC (λGEN%PVAR%2330::(real, 2) cart. ∃ n::nat. SETSPEC GEN%PVAR%2330 (IN n HOL_Light_Import.UNIV) (vsum (dotdot q n) f))) ∧ (∀ n≥p. HOL_Light_Import.real (g n) ∧ (0::real) ≤ Re (g n)) ∧ (∀ n≥p. Re (g (n + (1::nat))) ≤ Re (g n)) ⟶ (∃ B>0::real. ∀ (m::nat) n::nat. p ≤ m ⟶ vector_norm (vsum (dotdot m n) (λk::nat. complex_mul (f k) (g k))) ≤ B * vector_norm (g m))

thm ABEL_LEMMA:

∀ (a::nat ⇒ (real, ?'a::type) cart) (M::real) (r::real) r0::real. (0::real) ≤ r ∧ r < r0 ∧ (∀ n::nat. IN n (?k::nat ⇒ bool) ⟶ vector_norm (a n) * $r0^{n}$ ≤ M) ⟶ summable ?k (λn::nat. Cx (vector_norm (a n) * r^{n}))

thm POWER_SERIES_CONV_IMP_ABSCONV:

∀ (a::nat ⇒ (real, 2) cart) (k::nat ⇒ bool) (w::(real, 2) cart) z::(real, 2) cart. summable k (λn::nat. complex_mul (a n) (complex_pow z n)) ∧ vector_norm w < vector_norm z ⟶ summable k (λn::nat. Cx (vector_norm (complex_mul (a n) (complex_pow w n))))

thm POWER_SERIES_CONV_IMP_ABSCONV_WEAK:

∀ (a::nat ⇒ (real, 2) cart) (k::nat ⇒ bool) (w::(real, 2) cart) z::(real, 2) cart. summable k (λn::nat. complex_mul (a n) (complex_pow z n)) ∧ vector_norm w < vector_norm z ⟶ summable k (λn::nat. complex_mul (Cx (vector_norm (a n))) (complex_pow w n))

thm SUM_INTEGRAL_UBOUND_INCREASING:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (m::nat) n::nat. m ≤ n ∧ (∀ x::(real, 2) cart. IN x (closed_segment [(Cx (real_of_nat m), Cx (real_of_nat n + (1::real)))]) ⟶ has_complex_derivative g (f x) (at x)) ∧ (∀ (x::real) y::real. real_of_nat m ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n + (1::real) ⟶ Re (f (Cx x)) ≤ Re (f (Cx y))) ⟶ sum (dotdot m n) (λk::nat. Re (f (Cx (real_of_nat k)))) ≤ Re (vector_sub (g (Cx (real_of_nat n + (1::real)))) (g (Cx (real_of_nat m))))

thm SUM_INTEGRAL_UBOUND_DECREASING:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (m::nat) n::nat. m ≤ n ∧ (∀ x::(real, 2) cart. IN x (closed_segment [(Cx (real_of_nat m − (1::real)), Cx (real_of_nat n))]) ⟶ has_complex_derivative g (f x) (at x)) ∧ (∀ (x::real) y::real. real_of_nat m − (1::real) ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n ⟶ Re (f (Cx y)) ≤ Re (f (Cx x))) ⟶ sum (dotdot m n) (λk::nat. Re (f (Cx (real_of_nat k)))) ≤ Re (vector_sub (g (Cx (real_of_nat n))) (g (Cx (real_of_nat m − (1::real)))))

thm SUM_INTEGRAL_LBOUND_INCREASING:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (m::nat) n::nat. m ≤ n ∧ (∀ x::(real, 2) cart. IN x (closed_segment [(Cx (real_of_nat m − (1::real)), Cx (real_of_nat n))]) ⟶ has_complex_derivative g (f x) (at x)) ∧ (∀ (x::real) y::real. real_of_nat m − (1::real) ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n ⟶ Re (f (Cx x)) ≤ Re (f (Cx y))) ⟶ Re (vector_sub (g (Cx (real_of_nat n))) (g (Cx (real_of_nat m − (1::real))))) ≤ sum (dotdot m n) (λk::nat. Re (f (Cx (real_of_nat k))))

thm SUM_INTEGRAL_LBOUND_DECREASING:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (m::nat) n::nat. m ≤ n ∧ (∀ x::(real, 2) cart. IN x (closed_segment [(Cx (real_of_nat m), Cx (real_of_nat n + (1::real)))]) ⟶ has_complex_derivative g (f x) (at x)) ∧ (∀ (x::real) y::real. real_of_nat m ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n + (1::real) ⟶ Re (f (Cx y)) ≤ Re (f (Cx x))) ⟶ Re (vector_sub (g (Cx (real_of_nat n + (1::real)))) (g (Cx (real_of_nat m)))) ≤ sum (dotdot m n) (λk::nat. Re (f (Cx (real_of_nat k))))

thm SUM_INTEGRAL_BOUNDS_INCREASING:

∀ (f::(real, 2) cart ⇒ (real, 2) cart) (g::(real, 2) cart ⇒ (real, 2) cart) (m::nat) n::nat. m ≤ n ∧ (∀ x::(real, 2) cart. IN x (closed_segment [(Cx (real_of_nat m − (1::real)), Cx (real_of_nat n + (1::real)))]) ⟶ has_complex_derivative g (f x) (at x)) ∧ (∀ (x::real) y::real. real_of_nat m − (1::real) ≤ x ∧ x ≤ y

$\wedge$ $y \leq$ *real_of_nat n* + (*1::real*) $\longrightarrow$ *Re* (*f* (*Cx x*)) $\leq$ *Re* (*f* (*Cx y*))) $\longrightarrow$ *Re* (*vector_sub* (*g* (*Cx* (*real_of_nat n*))) (*g* (*Cx* (*real_of_nat m* − (*1::real*))))) $\leq$ *sum* (*dotdot m n*) (λ*k::nat*. *Re* (*f* (*Cx* (*real_of_nat k*)))) ∧ *sum* (*dotdot m n*) (λ*k::nat*. *Re* (*f* (*Cx* (*real_of_nat k*)))) $\leq$ *Re* (*vector_sub* (*g* (*Cx* (*real_of_nat n* + (*1::real*)))) (*g* (*Cx* (*real_of_nat m*))))

thm SUM_INTEGRAL_BOUNDS_DECREASING:

∀(*f::*(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*g::*(*real*, *2*) *cart* ⇒ (*real*, *2*) *cart*) (*m::nat*) *n::nat*. *m* ≤ *n* ∧ (∀*x::*(*real*, *2*) *cart*. *IN x* (*closed_segment* [(*Cx* (*real_of_nat m* − (*1::real*)), *Cx* (*real_of_nat n* + (*1::real*)))]) ⟶ *has_complex_derivative g* (*f x*) (*at x*)) ∧ (∀(*x::real*) *y::real*. *real_of_nat m* − (*1::real*) ≤ *x* ∧ *x* ≤ *y* ∧ *y* ≤ *real_of_nat n* + (*1::real*) ⟶ *Re* (*f* (*Cx y*)) ≤ *Re* (*f* (*Cx x*))) ⟶ *Re* (*vector_sub* (*g* (*Cx* (*real_of_nat n* + (*1::real*)))) (*g* (*Cx* (*real_of_nat m*)))) ≤ *sum* (*dotdot m n*) (λ*k::nat*. *Re* (*f* (*Cx* (*real_of_nat k*)))) ∧ *sum* (*dotdot m n*) (λ*k::nat*. *Re* (*f* (*Cx* (*real_of_nat k*)))) ≤ *Re* (*vector_sub* (*g* (*Cx* (*real_of_nat n*))) (*g* (*Cx* (*real_of_nat m* − (*1::real*)))))

thm LIM_INFINITY_SEQUENTIALLY_COMPLEX:

∀(*f::*(*real*, *2*) *cart* ⇒ (*real*, *?'a::type*) *cart*) *l::*(*real*, *?'a::type*) *cart*. −−> *f l at_infinity* ⟶ −−> (λ*n::nat*. *f* (*Cx* (*real_of_nat n*))) *l sequentially*

thm LIM_ZERO_INFINITY_COMPLEX:

∀(*f::*(*real*, *2*) *cart* ⇒ (*real*, *?'a::type*) *cart*) *l::*(*real*, *?'a::type*) *cart*. −−> (λ*x::*(*real*, *2*) *cart*. *f* (*complex_div* (*Cx* (*1::real*)) *x*)) *l* (*at* (*Cx* (*0::real*))) ⟶ −−> *f l at_infinity*

thm LIM_COMPLEX_REAL:

∀(*f::nat* ⇒ *real*) (*g::nat* ⇒ (*real*, *2*) *cart*) (*l::real*) *m::*(*real*, *2*) *cart*. *eventually* (λ*n::nat*. *Re* (*g n*) = *f n*) *sequentially* ∧ *Re m* = *l* ∧ −−> *g m sequentially* ⟶ (∀*e*>*0::real*. ∃*N::nat*. ∀*n*≥*N*. |*f n* − *l*| < *e*)

thm LIM_COMPLEX_REAL_0:

∀(*f::nat* ⇒ *real*) *g::nat* ⇒ (*real*, *2*) *cart*. *eventually* (λ*n::nat*. *Re* (*g n*) = *f n*) *sequentially* ∧ −−> *g* (*Cx* (*0::real*)) *sequentially* ⟶ (∀*e*>*0::real*. ∃*N::nat*. ∀*n*≥*N*. |*f n*| < *e*)

thm POWER_SERIES_UNIFORM_CONVERGENCE_STOLZ_1:

∀(*M::real*) (*a::nat* ⇒ (*real*, *2*) *cart*) (*s::nat* ⇒ *bool*) *e::real*. *summable s a* ∧ (*0::real*) < *M* ∧ (*0::real*) < *e* ⟶ *eventually* (λ*n::nat*. ∀*z::*(*real*, *2*) *cart*. *vector_norm* (*vector_sub* (*Cx* (*1::real*)) *z*) ≤ *M* ∗ ((*1::real*) − *vector_norm z*) ⟶ *vector_norm* (*vector_sub* (*vsum* (*HOL_Light_Import.INTER s* (*dotdot* (*0::nat*) *n*)) (λ*i::nat*. *complex_mul* (*a i*) (*complex_pow z i*))) (*infsum s* (λ*i::nat*. *complex_mul* (*a i*) (*complex_pow z i*)))) < *e*) *sequentially*

thm POWER_SERIES_UNIFORM_CONVERGENCE_STOLZ:

∀(*M::real*) (*a::nat* ⇒ (*real*, *2*) *cart*) (*w::*(*real*, *2*) *cart*) (*s::nat* ⇒ *bool*) *e::real*. *summable s* (λ*i::nat*. *complex_mul* (*a i*) (*complex_pow w i*)) ∧ (*0::real*) <

$M \wedge (0::real) < e \longrightarrow eventually\ (\lambda n::nat.\ \forall z::(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ w\ z) \leq M * (vector\_norm\ w - vector\_norm\ z) \longrightarrow vector\_norm\ (vector\_sub\ (vsum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ (0::nat)\ n))\ (\lambda i::nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ z\ i)))\ (infsum\ s\ (\lambda i::nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ z\ i)))) < e)\ sequentially$

thm ABEL_POWER_SERIES_CONTINUOUS:

$\forall (M::real)\ (s::nat \Rightarrow bool)\ a::nat \Rightarrow (real,\ 2)\ cart.\ summable\ s\ a \wedge (0::real) < M \longrightarrow continuous\_on\ (\lambda z::(real,\ 2)\ cart.\ infsum\ s\ (\lambda i::nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ z\ i)))\ (GSPEC\ (\lambda GEN\%PVAR\%2332::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2332\ (vector\_norm\ (vector\_sub\ (Cx\ (1::real))\ z) \leq M * ((1::real) - vector\_norm\ z))\ z))$

thm ABEL_LIMIT_THEOREM:

$\forall (M::real)\ (s::nat \Rightarrow bool)\ a::nat \Rightarrow (real,\ 2)\ cart.\ summable\ s\ a \wedge (0::real) < M \longrightarrow (\forall z::(real,\ 2)\ cart.\ vector\_norm\ z < (1::real) \longrightarrow summable\ s\ (\lambda i::nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ z\ i))) \wedge --> (\lambda z::(real,\ 2)\ cart.\ infsum\ s\ (\lambda i::nat.\ complex\_mul\ (a\ i)\ (complex\_pow\ z\ i)))\ (infsum\ s\ a)\ (within\ (at\ (Cx\ (1::real)))\ (GSPEC\ (\lambda GEN\%PVAR\%2334::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2334\ (vector\_norm\ (vector\_sub\ (Cx\ (1::real))\ z) \leq M * ((1::real) - vector\_norm\ z))\ z)))$

thm DEF_cexp:

$cexp = (\lambda\_1875609::(real,\ 2)\ cart.\ infsum\ (from\ (0::nat))\ (\lambda n::nat.\ complex\_div\ (complex\_pow\ \_1875609\ n)\ (Cx\ (real\_of\_nat\ (fact\ n)))))$

thm cexp:

$\forall z::(real,\ 2)\ cart.\ cexp\ z = infsum\ (from\ (0::nat))\ (\lambda n::nat.\ complex\_div\ (complex\_pow\ z\ n)\ (Cx\ (real\_of\_nat\ (fact\ n))))$

thm CEXP_0:

$cexp\ (Cx\ (0::real)) = Cx\ (1::real)$

thm CEXP_CONVERGES_UNIFORMLY_CAUCHY:

$\forall (R::real)\ e::real.\ (0::real) < e \wedge (0::real) < R \longrightarrow (\exists N::nat.\ \forall (m::nat)\ (n::nat)\ z::(real,\ 2)\ cart.\ N \leq m \wedge vector\_norm\ z \leq R \longrightarrow vector\_norm\ (vsum\ (dotdot\ m\ n)\ (\lambda i::nat.\ complex\_div\ (complex\_pow\ z\ i)\ (Cx\ (real\_of\_nat\ (fact\ i))))) < e)$

thm CEXP_CONVERGES:

$\forall z::(real,\ 2)\ cart.\ sums\ (\lambda n::nat.\ complex\_div\ (complex\_pow\ z\ n)\ (Cx\ (real\_of\_nat\ (fact\ n))))\ (cexp\ z)\ (from\ (0::nat))$

thm CEXP_CONVERGES_UNIQUE:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ sums\ (\lambda n::nat.\ complex\_div\ (complex\_pow\ z\ n)\ (Cx\ (real\_of\_nat\ (fact\ n))))\ w\ (from\ (0::nat)) = (w = cexp\ z)$

thm CEXP_CONVERGES_UNIFORMLY:

$\forall (R::real)\ e::real.\ (0::real) < R \wedge (0::real) < e \longrightarrow (\exists N::nat.\ \forall (n::nat)\ z::(real, 2)\ cart.\ N \leq n \wedge vector\_norm\ z < R \longrightarrow vector\_norm\ (vector\_sub\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ complex\_div\ (complex\_pow\ z\ i)\ (Cx\ (real\_of\_nat\ (fact\ i)))))\ (cexp\ z)) \leq e)$

thm HAS_COMPLEX_DERIVATIVE_CEXP:

$\forall z::(real, 2)\ cart.\ has\_complex\_derivative\ cexp\ (cexp\ z)\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CEXP:

$\forall z::(real, 2)\ cart.\ complex\_differentiable\ cexp\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CEXP:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ complex\_differentiable\ cexp\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CEXP:

$\forall z::(real, 2)\ cart.\ continuous\ cexp\ (at\ z)$

thm CONTINUOUS_WITHIN_CEXP:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ continuous\ cexp\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CEXP:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ continuous\_on\ cexp\ s$

thm HOLOMORPHIC_ON_CEXP:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ cexp\ s$

thm CEXP_ADD_MUL:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ complex\_mul\ (cexp\ (vector\_add\ w\ z))\ (cexp\ (vector\_neg\ z)) = cexp\ w$

thm CEXP_NEG_RMUL:

$\forall z::(real, 2)\ cart.\ complex\_mul\ (cexp\ z)\ (cexp\ (vector\_neg\ z)) = Cx\ (1::real)$

thm CEXP_NEG_LMUL:

$\forall z::(real, 2)\ cart.\ complex\_mul\ (cexp\ (vector\_neg\ z))\ (cexp\ z) = Cx\ (1::real)$

thm CEXP_NEG:

$\forall z::(real, 2)\ cart.\ cexp\ (vector\_neg\ z) = complex\_inv\ (cexp\ z)$

thm CEXP_ADD:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ cexp\ (vector\_add\ w\ z) = complex\_mul\ (cexp\ w)\ (cexp\ z)$

thm CEXP_SUB:

$\forall (w::(real, 2)\ cart)\ z::(real, 2)\ cart.\ cexp\ (vector\_sub\ w\ z) = complex\_div\ (cexp\ w)\ (cexp\ z)$

thm CEXP_NZ:

$\forall z::(real, 2)\ cart.\ cexp\ z \neq Cx\ (0::real)$

thm CEXP_N:

$\forall (n::nat)\ x::(real, 2)\ cart.\ cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ n))\ x) = complex\_pow\ (cexp\ x)\ n$

thm CEXP_VSUM:

$\forall (f::?'a::type \Rightarrow (real, 2)\ cart)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow cexp\ (vsum\ s\ f) = cproduct\ s\ (\lambda x::?'a::type.\ cexp\ (f\ x))$

thm CEXP_BOUND_BLEMMA:

$\forall B::real.\ (\forall z::(real, 2)\ cart.\ vector\_norm\ z \leq (1::real)\ /\ real\_of\_nat\ (2::nat) \longrightarrow vector\_norm\ (cexp\ z) \leq B) \longrightarrow (\forall z::(real, 2)\ cart.\ vector\_norm\ z \leq (1::real)\ /\ real\_of\_nat\ (2::nat) \longrightarrow vector\_norm\ (cexp\ z) \leq (1::real) + B\ /\ real\_of\_nat\ (2::nat))$

thm CEXP_BOUND_HALF:

$\forall z::(real, 2)\ cart.\ vector\_norm\ z \leq (1::real)\ /\ real\_of\_nat\ (2::nat) \longrightarrow vector\_norm\ (cexp\ z) \leq real\_of\_nat\ (2::nat)$

thm CEXP_BOUND_LEMMA:

$\forall z::(real, 2)\ cart.\ vector\_norm\ z \leq (1::real)\ /\ real\_of\_nat\ (2::nat) \longrightarrow vector\_norm\ (cexp\ z) \leq (1::real) + real\_of\_nat\ (2::nat) * vector\_norm\ z$

thm DEF_ccos:

$ccos = (\lambda\_1875777::(real, 2)\ cart.\ complex\_div\ (vector\_add\ (cexp\ (complex\_mul\ ii\ \_1875777))\ (cexp\ (complex\_mul\ (vector\_neg\ ii)\ \_1875777)))\ (Cx\ (real\_of\_nat\ (2::nat))))$

thm ccos:

$\forall z::(real, 2)\ cart.\ ccos\ z = complex\_div\ (vector\_add\ (cexp\ (complex\_mul\ ii\ z))\ (cexp\ (complex\_mul\ (vector\_neg\ ii)\ z)))\ (Cx\ (real\_of\_nat\ (2::nat)))$

thm DEF_csin:

$csin = (\lambda\_1875782::(real, 2)\ cart.\ complex\_div\ (vector\_sub\ (cexp\ (complex\_mul\ ii\ \_1875782))\ (cexp\ (complex\_mul\ (vector\_neg\ ii)\ \_1875782)))\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ ii))$

thm csin:

$\forall z::(real, 2)\ cart.\ csin\ z = complex\_div\ (vector\_sub\ (cexp\ (complex\_mul\ ii\ z))\ (cexp\ (complex\_mul\ (vector\_neg\ ii)\ z)))\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ ii)$

thm CSIN_0:

csin (*Cx* (*0*::*real*)) = *Cx* (*0*::*real*)

thm CCOS_0:

ccos (*Cx* (*0*::*real*)) = *Cx* (*1*::*real*)

thm CSIN_CIRCLE:

∀ *z*::(*real*, *2*) *cart*. *vector_add* (*complex_pow* (*csin z*) (*2*::*nat*)) (*complex_pow* (*ccos z*) (*2*::*nat*)) = *Cx* (*1*::*real*)

thm CSIN_ADD:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *csin* (*vector_add w z*) = *vector_add* (*complex_mul* (*csin w*) (*ccos z*)) (*complex_mul* (*ccos w*) (*csin z*))

thm CCOS_ADD:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos* (*vector_add w z*) = *vector_sub* (*complex_mul* (*ccos w*) (*ccos z*)) (*complex_mul* (*csin w*) (*csin z*))

thm CSIN_NEG:

∀ *z*::(*real*, *2*) *cart*. *csin* (*vector_neg z*) = *vector_neg* (*csin z*)

thm CCOS_NEG:

∀ *z*::(*real*, *2*) *cart*. *ccos* (*vector_neg z*) = *ccos z*

thm CSIN_DOUBLE:

∀ *z*::(*real*, *2*) *cart*. *csin* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) *z*) = *complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*complex_mul* (*csin z*) (*ccos z*))

thm CCOS_DOUBLE:

∀ *z*::(*real*, *2*) *cart*. *ccos* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) *z*) = *vector_sub* (*complex_pow* (*ccos z*) (*2*::*nat*)) (*complex_pow* (*csin z*) (*2*::*nat*))

thm CSIN_SUB:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *csin* (*vector_sub w z*) = *vector_sub* (*complex_mul* (*csin w*) (*ccos z*)) (*complex_mul* (*ccos w*) (*csin z*))

thm CCOS_SUB:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos* (*vector_sub w z*) = *vector_add* (*complex_mul* (*ccos w*) (*ccos z*)) (*complex_mul* (*csin w*) (*csin z*))

thm COMPLEX_MUL_CSIN_CSIN:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *complex_mul* (*csin w*) (*csin z*) = *complex_div* (*vector_sub* (*ccos* (*vector_sub w z*)) (*ccos* (*vector_add w z*))) (*Cx* (*real_of_nat* (*2*::*nat*)))

thm COMPLEX_MUL_CSIN_CCOS:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *complex_mul* (*csin w*) (*ccos z*) = *complex_div* (*vector_add* (*csin* (*vector_add w z*)) (*csin* (*vector_sub w z*))) (*Cx* (*real_of_nat* (*2*::*nat*)))

thm COMPLEX_MUL_CCOS_CSIN:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *complex_mul* (*ccos w*) (*csin z*) = *complex_div* (*vector_sub* (*csin* (*vector_add w z*)) (*csin* (*vector_sub w z*))) (*Cx* (*real_of_nat* (*2*::*nat*)))

thm COMPLEX_MUL_CCOS_CCOS:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *complex_mul* (*ccos w*) (*ccos z*) = *complex_div* (*vector_add* (*ccos* (*vector_sub w z*)) (*ccos* (*vector_add w z*))) (*Cx* (*real_of_nat* (*2*::*nat*)))

thm COMPLEX_ADD_CSIN:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *vector_add* (*csin w*) (*csin z*) = *complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*complex_mul* (*csin* (*complex_div* (*vector_add w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))) (*ccos* (*complex_div* (*vector_sub w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))))

thm COMPLEX_SUB_CSIN:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *vector_sub* (*csin w*) (*csin z*) = *complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*complex_mul* (*csin* (*complex_div* (*vector_sub w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))) (*ccos* (*complex_div* (*vector_add w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))))

thm COMPLEX_ADD_CCOS:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *vector_add* (*ccos w*) (*ccos z*) = *complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*complex_mul* (*ccos* (*complex_div* (*vector_add w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))) (*ccos* (*complex_div* (*vector_sub w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))))

thm COMPLEX_SUB_CCOS:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *vector_sub* (*ccos w*) (*ccos z*) = *complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*complex_mul* (*csin* (*complex_div* (*vector_add w z*) (*Cx* (*real_of_nat* (*2*::*nat*))))) (*csin* (*complex_div* (*vector_sub z w*) (*Cx* (*real_of_nat* (*2*::*nat*))))))

thm CEXP_EULER:

$\forall$ *z*::(*real*, *2*) *cart*. *cexp* (*complex_mul ii z*) = *vector_add* (*ccos z*) (*complex_mul ii* (*csin z*))

thm DEMOIVRE:

$\forall$ (*z*::(*real*, *2*) *cart*) *n*::*nat*. *complex_pow* (*vector_add* (*ccos z*) (*complex_mul ii* (*csin z*))) *n* = *vector_add* (*ccos* (*complex_mul* (*Cx* (*real_of_nat n*)) *z*)) (*complex_mul ii* (*csin* (*complex_mul* (*Cx* (*real_of_nat n*)) *z*)))

thm DEF_exp:

$exp = (\lambda\_1875787::real.\ Re\ (cexp\ (Cx\ \_1875787)))$

thm exp:

$\forall x::real.\ exp\ x = Re\ (cexp\ (Cx\ x))$

thm CNJ_CEXP:

$\forall z::(real,\ 2)\ cart.\ cnj\ (cexp\ z) = cexp\ (cnj\ z)$

thm REAL_EXP:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (cexp\ z)$

thm CX_EXP:

$\forall x::real.\ Cx\ (exp\ x) = cexp\ (Cx\ x)$

thm REAL_EXP_ADD:

$\forall (x::real)\ y::real.\ exp\ (x + y) = exp\ x * exp\ y$

thm REAL_EXP_0:

$exp\ (0::real) = (1::real)$

thm REAL_EXP_ADD_MUL:

$\forall (x::real)\ y::real.\ exp\ (x + y) * exp\ (-\ x) = exp\ y$

thm REAL_EXP_NEG_MUL:

$\forall x::real.\ exp\ x * exp\ (-\ x) = (1::real)$

thm REAL_EXP_NEG_MUL2:

$\forall x::real.\ exp\ (-\ x) * exp\ x = (1::real)$

thm REAL_EXP_NEG:

$\forall x::real.\ exp\ (-\ x) = inverse\_class.inverse\ (exp\ x)$

thm REAL_EXP_N:

$\forall (n::nat)\ x::real.\ exp\ (real\_of\_nat\ n * x) = (exp\ x)^n$

thm REAL_EXP_SUB:

$\forall (x::real)\ y::real.\ exp\ (x - y) = exp\ x\ /\ exp\ y$

thm REAL_EXP_NZ:

$\forall x::real.\ exp\ x \neq (0::real)$

thm REAL_EXP_POS_LE:

$\forall x::real.\ (0::real) \leq exp\ x$

thm REAL_EXP_POS_LT:

$\forall x::real.\ (0::real) < exp\ x$

thm REAL_EXP_LE_X:

$\forall x \geq 0::real.\ (1::real) + x \leq exp\ x$

thm REAL_EXP_LT_1:

$\forall x > 0::real.\ (1::real) < exp\ x$

thm REAL_EXP_MONO_IMP:

$\forall (x::real)\ y::real.\ x < y \longrightarrow exp\ x < exp\ y$

thm REAL_EXP_MONO_LT:

$\forall (x::real)\ y::real.\ (exp\ x < exp\ y) = (x < y)$

thm REAL_EXP_MONO_LE:

$\forall (x::real)\ y::real.\ (exp\ x \leq exp\ y) = (x \leq y)$

thm REAL_EXP_INJ:

$\forall (x::real)\ y::real.\ (exp\ x = exp\ y) = (x = y)$

thm REAL_EXP_EQ_1:

$\forall x::real.\ (exp\ x = (1::real)) = (x = (0::real))$

thm REAL_ABS_EXP:

$\forall x::real.\ |exp\ x| = exp\ x$

thm REAL_EXP_SUM:

$\forall (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow exp\ (sum\ s\ f) = product\ s\ (\lambda x::?'a::type.\ exp\ (f\ x))$

thm REAL_EXP_BOUND_LEMMA:

$\forall x::real.\ (0::real) \leq x \wedge x \leq inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) \longrightarrow exp\ x \leq (1::real) + real\_of\_nat\ (2::nat) * x$

thm DEF_sin:

$sin = (\lambda\_1875818::real.\ Re\ (csin\ (Cx\ \_1875818)))$

thm sin:

$\forall x::real.\ sin\ x = Re\ (csin\ (Cx\ x))$

thm DEF_cos:

$cos = (\lambda\_1875823::real.\ Re\ (ccos\ (Cx\ \_1875823)))$

thm cos:

$\forall x::real.\ cos\ x = Re\ (ccos\ (Cx\ x))$

thm CNJ_CSIN:

$\forall z::(real,\ 2)\ cart.\ cnj\ (csin\ z) = csin\ (cnj\ z)$

thm COMPLEX_ADD_AC_conjunct2:

$vector\_add\ (?m::(real,\ 2)\ cart)\ (vector\_add\ (?n::(real,\ 2)\ cart)\ (?p::(real,\ 2)\ cart)) = vector\_add\ ?n\ (vector\_add\ ?m\ ?p)$

thm COMPLEX_ADD_AC_conjunct1:

$vector\_add\ (vector\_add\ (?m::(real,\ 2)\ cart)\ (?n::(real,\ 2)\ cart))\ (?p::(real,\ 2)\ cart) = vector\_add\ ?m\ (vector\_add\ ?n\ ?p)$

thm COMPLEX_ADD_AC_conjunct0:

$vector\_add\ (?m::(real,\ 2)\ cart)\ (?n::(real,\ 2)\ cart) = vector\_add\ ?n\ ?m$

thm CNJ_CCOS:

$\forall z::(real,\ 2)\ cart.\ cnj\ (ccos\ z) = ccos\ (cnj\ z)$

thm REAL_SIN:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (csin\ z)$

thm REAL_COS:

$\forall z::(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (ccos\ z)$

thm CX_SIN:

$\forall x::real.\ Cx\ (sin\ x) = csin\ (Cx\ x)$

thm CX_COS:

$\forall x::real.\ Cx\ (cos\ x) = ccos\ (Cx\ x)$

thm HAS_COMPLEX_DERIVATIVE_CSIN:

$\forall z::(real,\ 2)\ cart.\ has\_complex\_derivative\ csin\ (ccos\ z)\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CSIN:

$\forall z::(real,\ 2)\ cart.\ complex\_differentiable\ csin\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CSIN:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ complex\_differentiable\ csin\ (within\ (at\ z)\ s)$

thm HAS_COMPLEX_DERIVATIVE_CCOS:

$\forall z::(real,\ 2)\ cart.\ has\_complex\_derivative\ ccos\ (vector\_neg\ (csin\ z))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CCOS:

$\forall z::(real,\ 2)\ cart.\ complex\_differentiable\ ccos\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CCOS:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ complex\_differentiable\ ccos\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CSIN:

$\forall z::(real, 2)\ cart.\ continuous\ csin\ (at\ z)$

thm CONTINUOUS_WITHIN_CSIN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ continuous\ csin\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CSIN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ continuous\_on\ csin\ s$

thm HOLOMORPHIC_ON_CSIN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ csin\ s$

thm CONTINUOUS_AT_CCOS:

$\forall z::(real, 2)\ cart.\ continuous\ ccos\ (at\ z)$

thm CONTINUOUS_WITHIN_CCOS:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ continuous\ ccos\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CCOS:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ continuous\_on\ ccos\ s$

thm HOLOMORPHIC_ON_CCOS:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ holomorphic\_on\ ccos\ s$

thm SIN_0:

$sin\ (0::real) = (0::real)$

thm COS_0:

$cos\ (0::real) = (1::real)$

thm SIN_CIRCLE:

$\forall x::real.\ (sin\ x)^2 + (cos\ x)^2 = (1::real)$

thm SIN_ADD:

$\forall (x::real)\ y::real.\ sin\ (x + y) = sin\ x * cos\ y + cos\ x * sin\ y$

thm COS_ADD:

$\forall (x::real)\ y::real.\ cos\ (x + y) = cos\ x * cos\ y - sin\ x * sin\ y$

thm SIN_NEG:

$\forall x::real.\ sin\ (-\ x) = -\ sin\ x$

thm COS_NEG:

$\forall x{::}real.\ cos\ (-\ x) = cos\ x$

thm SIN_DOUBLE:

$\forall x{::}real.\ sin\ (real\_of\_nat\ (2{::}nat) * x) = real\_of\_nat\ (2{::}nat) * (sin\ x * cos\ x)$

thm COS_DOUBLE:

$\forall x{::}real.\ cos\ (real\_of\_nat\ (2{::}nat) * x) = (cos\ x)^2 - (sin\ x)^2$

thm COS_DOUBLE_COS:

$\forall x{::}real.\ cos\ (real\_of\_nat\ (2{::}nat) * x) = real\_of\_nat\ (2{::}nat) * (cos\ x)^2 - (1{::}real)$

thm COS_BOUND:

$\forall x{::}real.\ |cos\ x| \leq (1{::}real)$

thm SIN_BOUND:

$\forall x{::}real.\ |sin\ x| \leq (1{::}real)$

thm SIN_BOUNDS:

$\forall x{::}real.\ -\ (1{::}real) \leq sin\ x \wedge sin\ x \leq (1{::}real)$

thm COS_BOUNDS:

$\forall x{::}real.\ -\ (1{::}real) \leq cos\ x \wedge cos\ x \leq (1{::}real)$

thm COS_ABS:

$\forall x{::}real.\ cos\ |x| = cos\ x$

thm SIN_SUB:

$\forall (w{::}real)\ z{::}real.\ sin\ (w - z) = sin\ w * cos\ z - cos\ w * sin\ z$

thm COS_SUB:

$\forall (w{::}real)\ z{::}real.\ cos\ (w - z) = cos\ w * cos\ z + sin\ w * sin\ z$

thm REAL_MUL_SIN_SIN:

$\forall (x{::}real)\ y{::}real.\ sin\ x * sin\ y = (cos\ (x - y) - cos\ (x + y))\ /\ real\_of\_nat\ (2{::}nat)$

thm REAL_MUL_SIN_COS:

$\forall (x{::}real)\ y{::}real.\ sin\ x * cos\ y = (sin\ (x + y) + sin\ (x - y))\ /\ real\_of\_nat\ (2{::}nat)$

thm REAL_MUL_COS_SIN:

$\forall (x{::}real)\ y{::}real.\ cos\ x * sin\ y = (sin\ (x + y) - sin\ (x - y))\ /\ real\_of\_nat\ (2{::}nat)$

thm REAL_MUL_COS_COS:

$\forall (x::real)\ y::real.\ cos\ x * cos\ y = (cos\ (x - y) + cos\ (x + y)) / real\_of\_nat\ (2::nat)$

thm REAL_ADD_SIN:

$\forall (x::real)\ y::real.\ sin\ x + sin\ y = real\_of\_nat\ (2::nat) * (sin\ ((x + y) / real\_of\_nat\ (2::nat)) * cos\ ((x - y) / real\_of\_nat\ (2::nat)))$

thm REAL_SUB_SIN:

$\forall (x::real)\ y::real.\ sin\ x - sin\ y = real\_of\_nat\ (2::nat) * (sin\ ((x - y) / real\_of\_nat\ (2::nat)) * cos\ ((x + y) / real\_of\_nat\ (2::nat)))$

thm REAL_ADD_COS:

$\forall (x::real)\ y::real.\ cos\ x + cos\ y = real\_of\_nat\ (2::nat) * (cos\ ((x + y) / real\_of\_nat\ (2::nat)) * cos\ ((x - y) / real\_of\_nat\ (2::nat)))$

thm REAL_SUB_COS:

$\forall (x::real)\ y::real.\ cos\ x - cos\ y = real\_of\_nat\ (2::nat) * (sin\ ((x + y) / real\_of\_nat\ (2::nat)) * sin\ ((y - x) / real\_of\_nat\ (2::nat)))$

thm EULER:

$\forall z::(real,\ 2)\ cart.\ cexp\ z = complex\_mul\ (Cx\ (exp\ (Re\ z)))\ (vector\_add\ (Cx\ (cos\ (Im\ z)))\ (complex\_mul\ ii\ (Cx\ (sin\ (Im\ z)))))$

thm RE_CEXP:

$\forall z::(real,\ 2)\ cart.\ Re\ (cexp\ z) = exp\ (Re\ z) * cos\ (Im\ z)$

thm IM_CEXP:

$\forall z::(real,\ 2)\ cart.\ Im\ (cexp\ z) = exp\ (Re\ z) * sin\ (Im\ z)$

thm RE_CSIN:

$\forall z::(real,\ 2)\ cart.\ Re\ (csin\ z) = (exp\ (Im\ z) + exp\ (-\ Im\ z)) / real\_of\_nat\ (2::nat) * sin\ (Re\ z)$

thm IM_CSIN:

$\forall z::(real,\ 2)\ cart.\ Im\ (csin\ z) = (exp\ (Im\ z) - exp\ (-\ Im\ z)) / real\_of\_nat\ (2::nat) * cos\ (Re\ z)$

thm RE_CCOS:

$\forall z::(real,\ 2)\ cart.\ Re\ (ccos\ z) = (exp\ (Im\ z) + exp\ (-\ Im\ z)) / real\_of\_nat\ (2::nat) * cos\ (Re\ z)$

thm IM_CCOS:

$\forall z::(real,\ 2)\ cart.\ Im\ (ccos\ z) = (exp\ (-\ Im\ z) - exp\ (Im\ z)) / real\_of\_nat\ (2::nat) * sin\ (Re\ z)$

thm IVT_INCREASING_RE:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (a::real)\ (b::real)\ y::real.\ a \leq b \wedge (\forall x::real.\ a \leq x \wedge x \leq b \longrightarrow continuous\ f\ (at\ (Cx\ x))) \wedge Re\ (f\ (Cx\ a)) \leq y \wedge y \leq Re\ (f\ (Cx\ b)) \longrightarrow (\exists x \geq a.\ x \leq b \wedge Re\ (f\ (Cx\ x)) = y)$

thm IVT_DECREASING_RE:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (a::real)\ (b::real)\ y::real.\ a \leq b \wedge (\forall x::real.\ a \leq x \wedge x \leq b \longrightarrow continuous\ f\ (at\ (Cx\ x))) \wedge Re\ (f\ (Cx\ b)) \leq y \wedge y \leq Re\ (f\ (Cx\ a)) \longrightarrow (\exists x \geq a.\ x \leq b \wedge Re\ (f\ (Cx\ x)) = y)$

thm IVT_INCREASING_IM:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (a::real)\ (b::real)\ y::real.\ a \leq b \wedge (\forall x::real.\ a \leq x \wedge x \leq b \longrightarrow continuous\ f\ (at\ (Cx\ x))) \wedge Im\ (f\ (Cx\ a)) \leq y \wedge y \leq Im\ (f\ (Cx\ b)) \longrightarrow (\exists x \geq a.\ x \leq b \wedge Im\ (f\ (Cx\ x)) = y)$

thm IVT_DECREASING_IM:

$\forall (f::(real, 2)\ cart \Rightarrow (real, 2)\ cart)\ (a::real)\ (b::real)\ y::real.\ a \leq b \wedge (\forall x::real.\ a \leq x \wedge x \leq b \longrightarrow continuous\ f\ (at\ (Cx\ x))) \wedge Im\ (f\ (Cx\ b)) \leq y \wedge y \leq Im\ (f\ (Cx\ a)) \longrightarrow (\exists x \geq a.\ x \leq b \wedge Im\ (f\ (Cx\ x)) = y)$

thm DEF_log:

$log = (\lambda\_1875924::real.\ SOME\ x::real.\ exp\ x = \_1875924)$

thm log_def:

$\forall y::real.\ log\ y = (SOME\ x::real.\ exp\ x = y)$

thm EXP_LOG:

$\forall x>0::real.\ exp\ (log\ x) = x$

thm LOG_EXP:

$\forall x::real.\ log\ (exp\ x) = x$

thm REAL_EXP_LOG:

$\forall x::real.\ (exp\ (log\ x) = x) = ((0::real) < x)$

thm LOG_MUL:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow log\ (x * y) = log\ x + log\ y$

thm LOG_INJ:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow (log\ x = log\ y) = (x = y)$

thm LOG_1:

$log\ (1::real) = (0::real)$

thm LOG_INV:

$\forall x>0::real.\ log\ (inverse\_class.inverse\ x) = -\ log\ x$

thm LOG_DIV:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow log\ (x\ /\ y) = log\ x - log\ y$

thm LOG_MONO_LT:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow (log\ x < log\ y) = (x < y)$

thm LOG_MONO_LT_IMP:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x < y \longrightarrow log\ x < log\ y$

thm LOG_MONO_LE:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) < y \longrightarrow (log\ x \leq log\ y) = (x \leq y)$

thm LOG_MONO_LE_IMP:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge x \leq y \longrightarrow log\ x \leq log\ y$

thm LOG_POW:

$\forall (n::nat)\ x::real.\ (0::real) < x \longrightarrow log\ x^n = real\_of\_nat\ n * log\ x$

thm LOG_LE:

$\forall x \geq 0::real.\ log\ ((1::real) + x) \leq x$

thm LOG_LT_X:

$\forall x > 0::real.\ log\ x < x$

thm LOG_POS:

$\forall x \geq 1::real.\ (0::real) \leq log\ x$

thm LOG_POS_LT:

$\forall x > 1::real.\ (0::real) < log\ x$

thm SIN_NEARZERO:

$\exists x > 0::real.\ \forall y::real.\ (0::real) < y \wedge y \leq x \longrightarrow (0::real) < sin\ y$

thm SIN_NONTRIVIAL:

$\exists x > 0::real.\ sin\ x \neq (0::real)$

thm COS_NONTRIVIAL:

$\exists x > 0::real.\ cos\ x \neq (1::real)$

thm COS_DOUBLE_BOUND:

$\forall x::real.\ (0::real) \leq cos\ x \longrightarrow real\_of\_nat\ (2::nat) * ((1::real) - cos\ x) \leq (1::real) - cos\ (real\_of\_nat\ (2::nat) * x)$

thm COS_GOESNEGATIVE_LEMMA:

$\forall x::real.\ cos\ x < (1::real) \longrightarrow (\exists n::nat.\ cos\ ((real\_of\_nat\ (2::nat))^n * x) < (0::real))$

thm COS_GOESNEGATIVE:

$\exists\, x{>}0::real.\ cos\ x < (0::real)$

thm COS_HASZERO:

$\exists\, x{>}0::real.\ cos\ x = (0::real)$

thm SIN_HASZERO:

$\exists\, x{>}0::real.\ sin\ x = (0::real)$

thm SIN_HASZERO_MINIMAL:

$\exists\, p{>}0::real.\ sin\ p = (0::real) \wedge (\forall\, x::real.\ (0::real) < x \wedge x < p \longrightarrow sin\ x \neq (0::real))$

thm pi:

$pi = (SOME\ p::real.\ (0::real) < p \wedge sin\ p = (0::real) \wedge (\forall\, x::real.\ (0::real) < x \wedge x < p \longrightarrow sin\ x \neq (0::real)))$

thm PI_WORKS:

$(0::real) < pi \wedge sin\ pi = (0::real) \wedge (\forall\, x::real.\ (0::real) < x \wedge x < pi \longrightarrow sin\ x \neq (0::real))$

thm PI_WORKS_conjunct2:

$\forall\, x::real.\ (0::real) < x \wedge x < pi \longrightarrow sin\ x \neq (0::real)$

thm SIN_PI:

$sin\ pi = (0::real)$

thm PI_POS:

$(0::real) < pi$

thm PI_POS_LE:

$(0::real) \leq pi$

thm PI_NZ:

$pi \neq (0::real)$

thm REAL_ABS_PI:

$|pi| = pi$

thm SIN_POS_PI:

$\forall\, x::real.\ (0::real) < x \wedge x < pi \longrightarrow (0::real) < sin\ x$

thm COS_PI2:

$cos\ (pi\ /\ real\_of\_nat\ (2::nat)) = (0::real)$

thm COS_PI:

$cos\ pi = -\ (1::real)$

thm SIN_PI2:

$sin\ (pi\ /\ real\_of\_nat\ (2::nat)) = (1::real)$

thm SIN_COS:

$\forall x::real.\ sin\ x = cos\ (pi\ /\ real\_of\_nat\ (2::nat) - x)$

thm COS_SIN:

$\forall x::real.\ cos\ x = sin\ (pi\ /\ real\_of\_nat\ (2::nat) - x)$

thm SIN_PERIODIC_PI:

$\forall x::real.\ sin\ (x + pi) = -\ sin\ x$

thm COS_PERIODIC_PI:

$\forall x::real.\ cos\ (x + pi) = -\ cos\ x$

thm SIN_PERIODIC:

$\forall x::real.\ sin\ (x + real\_of\_nat\ (2::nat) * pi) = sin\ x$

thm COS_PERIODIC:

$\forall x::real.\ cos\ (x + real\_of\_nat\ (2::nat) * pi) = cos\ x$

thm SIN_NPI:

$\forall n::nat.\ sin\ (real\_of\_nat\ n * pi) = (0::real)$

thm COS_NPI:

$\forall n::nat.\ cos\ (real\_of\_nat\ n * pi) = (-\ (1::real))^{n}$

thm COS_POS_PI2:

$\forall x::real.\ (0::real) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) < cos\ x$

thm SIN_POS_PI2:

$\forall x::real.\ (0::real) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) < sin\ x$

thm COS_POS_PI:

$\forall x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) < cos\ x$

thm COS_POS_PI_LE:

$\forall x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) \leq x \wedge x \leq pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) \leq cos\ x$

thm SIN_POS_PI_LE:

$\forall x::real.\ (0::real) \leq x \wedge x \leq pi \longrightarrow (0::real) \leq sin\ x$

thm SIN_PIMUL_EQ_0:

$\forall n{::}real.\ (sin\ (n * pi) = (0{::}real)) = integer\ n$

thm SIN_EQ_0:

$\forall x{::}real.\ (sin\ x = (0{::}real)) = (\exists n{::}real.\ integer\ n \wedge x = n * pi)$

thm COS_EQ_0:

$\forall x{::}real.\ (cos\ x = (0{::}real)) = (\exists n{::}real.\ integer\ n \wedge x = (n + (1{::}real) / real\_of\_nat\ (2{::}nat)) * pi)$

thm SIN_ZERO_PI:

$\forall x{::}real.\ (sin\ x = (0{::}real)) = ((\exists n{::}nat.\ x = real\_of\_nat\ n * pi) \vee (\exists n{::}nat.\ x = -\ (real\_of\_nat\ n * pi)))$

thm COS_ZERO_PI:

$\forall x{::}real.\ (cos\ x = (0{::}real)) = ((\exists n{::}nat.\ x = (real\_of\_nat\ n + (1{::}real) / real\_of\_nat\ (2{::}nat)) * pi) \vee (\exists n{::}nat.\ x = -\ ((real\_of\_nat\ n + (1{::}real) / real\_of\_nat\ (2{::}nat)) * pi)))$

thm SIN_ZERO:

$\forall x{::}real.\ (sin\ x = (0{::}real)) = ((\exists n{::}nat.\ even\ n \wedge x = real\_of\_nat\ n * (pi / real\_of\_nat\ (2{::}nat))) \vee (\exists n{::}nat.\ even\ n \wedge x = -\ (real\_of\_nat\ n * (pi / real\_of\_nat\ (2{::}nat)))))$

thm COS_ZERO:

$\forall x{::}real.\ (cos\ x = (0{::}real)) = ((\exists n{::}nat.\ odd\ n \wedge x = real\_of\_nat\ n * (pi / real\_of\_nat\ (2{::}nat))) \vee (\exists n{::}nat.\ odd\ n \wedge x = -\ (real\_of\_nat\ n * (pi / real\_of\_nat\ (2{::}nat)))))$

thm COS_ONE_2PI:

$\forall x{::}real.\ (cos\ x = (1{::}real)) = ((\exists n{::}nat.\ x = real\_of\_nat\ n * (real\_of\_nat\ (2{::}nat) * pi)) \vee (\exists n{::}nat.\ x = -\ (real\_of\_nat\ n * (real\_of\_nat\ (2{::}nat) * pi))))$

thm SIN_COS_SQRT:

$\forall x{::}real.\ (0{::}real) \leq sin\ x \longrightarrow sin\ x = sqrt\ ((1{::}real) - (cos\ x)^2)$

thm SIN_EQ_0_PI:

$\forall x{::}real.\ -\ pi < x \wedge x < pi \wedge sin\ x = (0{::}real) \longrightarrow x = (0{::}real)$

thm COS_TREBLE_COS:

$\forall x{::}real.\ cos\ (real\_of\_nat\ (3{::}nat) * x) = real\_of\_nat\ (4{::}nat) * (cos\ x)^{3{::}nat} - real\_of\_nat\ (3{::}nat) * cos\ x$

thm COS_PI6:

$cos\ (pi\ /\ real\_of\_nat\ (6{::}nat)) = sqrt\ (real\_of\_nat\ (3{::}nat))\ /\ real\_of\_nat\ (2{::}nat)$

thm SIN_PI6:

$sin\ (pi\ /\ real\_of\_nat\ (6::nat)) = (1::real)\ /\ real\_of\_nat\ (2::nat)$

thm SIN_POS_PI_REV:

$\forall x::real.\ (0::real) \leq x \wedge x \leq real\_of\_nat\ (2::nat) * pi \wedge (0::real) < sin\ x \longrightarrow (0::real) < x \wedge x < pi$

thm SIN_TOTAL_POS:

$\forall y::real.\ (0::real) \leq y \wedge y \leq (1::real) \longrightarrow (\exists x\geq 0::real.\ x \leq pi\ /\ real\_of\_nat\ (2::nat) \wedge sin\ x = y)$

thm SINCOS_TOTAL_PI2:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x^2 + y^2 = (1::real) \longrightarrow (\exists t\geq 0::real.\ t \leq pi\ /\ real\_of\_nat\ (2::nat) \wedge x = cos\ t \wedge y = sin\ t)$

thm SINCOS_TOTAL_PI:

$\forall (x::real)\ y::real.\ (0::real) \leq y \wedge x^2 + y^2 = (1::real) \longrightarrow (\exists t\geq 0::real.\ t \leq pi \wedge x = cos\ t \wedge y = sin\ t)$

thm SINCOS_TOTAL_2PI:

$\forall (x::real)\ y::real.\ x^2 + y^2 = (1::real) \longrightarrow (\exists t\geq 0::real.\ t < real\_of\_nat\ (2::nat) * pi \wedge x = cos\ t \wedge y = sin\ t)$

thm CIRCLE_SINCOS:

$\forall (x::real)\ y::real.\ x^2 + y^2 = (1::real) \longrightarrow (\exists t::real.\ x = cos\ t \wedge y = sin\ t)$

thm CX_PI_NZ:

$Cx\ pi \neq Cx\ (0::real)$

thm Functional_equation.sqrt_sqrt:

$\forall x\geq 0::real.\ sqrt\ x * sqrt\ x = x$

thm COMPLEX_UNIMODULAR_POLAR:

$\forall z::(real,\ 2)\ cart.\ vector\_norm\ z = (1::real) \longrightarrow (\exists x::real.\ z = complex\ (cos\ x,\ sin\ x))$

thm SIN_INTEGER_2PI:

$\forall n::real.\ integer\ n \longrightarrow sin\ (real\_of\_nat\ (2::nat) * pi * n) = (0::real)$

thm COS_INTEGER_2PI:

$\forall n::real.\ integer\ n \longrightarrow cos\ (real\_of\_nat\ (2::nat) * pi * n) = (1::real)$

thm Trigonometry2.TWO_PI_POS:

$(0::real) < real\_of\_nat\ (2::nat) * pi$

thm SINCOS_PRINCIPAL_VALUE:

$\forall x{::}real.\ \exists y{::}real.\ (-\ pi < y \wedge y \leq pi) \wedge sin\ y = sin\ x \wedge cos\ y = cos\ x$

thm CEXP_COMPLEX:

$\forall (r{::}real)\ t{::}real.\ cexp\ (complex\ (r,\ t)) = complex\_mul\ (Cx\ (exp\ r))\ (complex\ (cos\ t,\ sin\ t))$

thm NORM_COSSIN:

$\forall t{::}real.\ vector\_norm\ (complex\ (cos\ t,\ sin\ t)) = (1{::}real)$

thm NORM_CEXP:

$\forall z{::}(real,\ 2)\ cart.\ vector\_norm\ (cexp\ z) = exp\ (Re\ z)$

thm NORM_CEXP_II:

$\forall t{::}real.\ vector\_norm\ (cexp\ (complex\_mul\ ii\ (Cx\ t))) = (1{::}real)$

thm NORM_CEXP_IMAGINARY:

$\forall z{::}(real,\ 2)\ cart.\ vector\_norm\ (cexp\ z) = (1{::}real) \longrightarrow Re\ z = (0{::}real)$

thm CEXP_EQ_1:

$\forall z{::}(real,\ 2)\ cart.\ (cexp\ z = Cx\ (1{::}real)) = (Re\ z = (0{::}real) \wedge (\exists n{::}real.\ integer\ n \wedge Im\ z = real\_of\_nat\ (2{::}nat) * (n * pi)))$

thm CEXP_EQ:

$\forall (w{::}(real,\ 2)\ cart)\ z{::}(real,\ 2)\ cart.\ (cexp\ w = cexp\ z) = (\exists n{::}real.\ integer\ n \wedge w = vector\_add\ z\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2{::}nat) * (n * pi)))\ ii))$

thm COMPLEX_EQ_CEXP:

$\forall (w{::}(real,\ 2)\ cart)\ z{::}(real,\ 2)\ cart.\ |Im\ w - Im\ z| < real\_of\_nat\ (2{::}nat) * pi \wedge cexp\ w = cexp\ z \longrightarrow w = z$

thm SIN_COS_EQ:

$\forall (x{::}real)\ y{::}real.\ (sin\ y = sin\ x \wedge cos\ y = cos\ x) = (\exists n{::}real.\ integer\ n \wedge y = x + real\_of\_nat\ (2{::}nat) * (n * pi))$

thm SIN_COS_INJ:

$\forall (x{::}real)\ y{::}real.\ sin\ x = sin\ y \wedge cos\ x = cos\ y \wedge |x - y| < real\_of\_nat\ (2{::}nat) * pi \longrightarrow x = y$

thm CEXP_II_NE_1:

$\forall x{::}real.\ (0{::}real) < x \wedge x < real\_of\_nat\ (2{::}nat) * pi \longrightarrow cexp\ (complex\_mul\ ii\ (Cx\ x)) \neq Cx\ (1{::}real)$

thm TAYLOR_CEXP:

$\forall (n{::}nat)\ z{::}(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ (cexp\ z)\ (vsum\ (dotdot\ (0{::}nat)\ n)\ (\lambda k{::}nat.\ complex\_div\ (complex\_pow\ z\ k)\ (Cx\ (real\_of\_nat\ (fact\ k)))))) \leq exp\ |Re\ z| * ((vector\_norm\ z)^{n\ +\ (1{::}nat)}\ /\ real\_of\_nat\ (fact\ n))$

thm E_APPROX_32:

$|exp\ (1::real) - real\_of\_nat\ (11674931555::nat)\ /\ real\_of\_nat\ (4294967296::nat)| \leq inverse\_class.inverse\ (real\_of\_nat\ (2::nat))^{32::nat}$

thm TAYLOR_CSIN_RAW:

$\forall (n::nat)\ z::(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ (csin\ z)\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ if\ ODD\ k\ then\ complex\_mul\ (vector\_neg\ ii)\ (complex\_div\ (complex\_pow\ (complex\_mul\ ii\ z)\ k)\ (Cx\ (real\_of\_nat\ (fact\ k))))\ else\ Cx\ (0::real)))) \leq exp\ |Im\ z| * ((vector\_norm\ z)^{n\ +\ (1::nat)}\ /\ real\_of\_nat\ (fact\ n))$

thm TAYLOR_CSIN:

$\forall (n::nat)\ z::(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ (csin\ z)\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ complex\_mul\ (complex\_pow\ (vector\_neg\ (Cx\ (1::real)))\ k)\ (complex\_div\ (complex\_pow\ z\ ((2::nat) * k + (1::nat)))\ (Cx\ (real\_of\_nat\ (fact\ ((2::nat) * k + (1::nat))))))))) \leq exp\ |Im\ z| * ((vector\_norm\ z)^{(2::nat)\ *\ n\ +\ (3::nat)}\ /\ real\_of\_nat\ (fact\ ((2::nat) * n + (2::nat))))$

thm TAYLOR_CCOS_RAW:

$\forall (n::nat)\ z::(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ (ccos\ z)\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ if\ even\ k\ then\ complex\_div\ (complex\_pow\ (complex\_mul\ ii\ z)\ k)\ (Cx\ (real\_of\_nat\ (fact\ k)))\ else\ Cx\ (0::real)))) \leq exp\ |Im\ z| * ((vector\_norm\ z)^{n\ +\ (1::nat)}\ /\ real\_of\_nat\ (fact\ n))$

thm TAYLOR_CCOS:

$\forall (n::nat)\ z::(real,\ 2)\ cart.\ vector\_norm\ (vector\_sub\ (ccos\ z)\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ complex\_mul\ (complex\_pow\ (vector\_neg\ (Cx\ (1::real)))\ k)\ (complex\_div\ (complex\_pow\ z\ ((2::nat) * k))\ (Cx\ (real\_of\_nat\ (fact\ ((2::nat) * k)))))))) \leq exp\ |Im\ z| * ((vector\_norm\ z)^{(2::nat)\ *\ n\ +\ (2::nat)}\ /\ real\_of\_nat\ (fact\ ((2::nat) * n + (1::nat))))$

thm DEF_Arg:

$Arg = (\lambda\_1876907::(real,\ 2)\ cart.\ if\ \_1876907 = Cx\ (0::real)\ then\ 0::real\ else\ SOME\ t::real.\ (0::real) \leq t \wedge t < real\_of\_nat\ (2::nat) * pi \wedge \_1876907 = complex\_mul\ (Cx\ (vector\_norm\ \_1876907))\ (cexp\ (complex\_mul\ ii\ (Cx\ t))))$

thm Arg_DEF:

$\forall z::(real,\ 2)\ cart.\ Arg\ z = (if\ z = Cx\ (0::real)\ then\ 0::real\ else\ SOME\ t::real.\ (0::real) \leq t \wedge t < real\_of\_nat\ (2::nat) * pi \wedge z = complex\_mul\ (Cx\ (vector\_norm\ z))\ (cexp\ (complex\_mul\ ii\ (Cx\ t))))$

thm ARG_0:

$Arg\ (Cx\ (0::real)) = (0::real)$

thm ARG:

$\forall z::(real,\ 2)\ cart.\ (0::real) \leq Arg\ z \wedge Arg\ z < real\_of\_nat\ (2::nat) * pi \wedge z = complex\_mul\ (Cx\ (vector\_norm\ z))\ (cexp\ (complex\_mul\ ii\ (Cx\ (Arg\ z))))$

thm COMPLEX_NORM_EQ_1_CEXP:

$\forall z::(real, 2)\ cart.\ (vector\_norm\ z = (1::real)) = (\exists t::real.\ z = cexp\ (complex\_mul\ ii\ (Cx\ t)))$

thm ARG_UNIQUE:

$\forall (a::real)\ (r::real)\ z::(real, 2)\ cart.\ (0::real) < r \wedge complex\_mul\ (Cx\ r)\ (cexp\ (complex\_mul\ ii\ (Cx\ a))) = z \wedge (0::real) \leq a \wedge a < real\_of\_nat\ (2::nat) * pi \longrightarrow Arg\ z = a$

thm ARG_MUL_CX:

$\forall (r::real)\ z::(real, 2)\ cart.\ (0::real) < r \longrightarrow Arg\ (complex\_mul\ (Cx\ r)\ z) = Arg\ z$

thm ARG_DIV_CX:

$\forall (r::real)\ z::(real, 2)\ cart.\ (0::real) < r \longrightarrow Arg\ (complex\_div\ z\ (Cx\ r)) = Arg\ z$

thm ARG_LT_NZ:

$\forall z::(real, 2)\ cart.\ ((0::real) < Arg\ z) = (Arg\ z \neq (0::real))$

thm ARG_LE_PI:

$\forall z::(real, 2)\ cart.\ (Arg\ z \leq pi) = ((0::real) \leq Im\ z)$

thm ARG_LT_PI:

$\forall z::(real, 2)\ cart.\ ((0::real) < Arg\ z \wedge Arg\ z < pi) = ((0::real) < Im\ z)$

thm ARG_EQ_0:

$\forall z::(real, 2)\ cart.\ (Arg\ z = (0::real)) = (HOL\_Light\_Import.real\ z \wedge (0::real) \leq Re\ z)$

thm ARG_NUM:

$\forall n::nat.\ Arg\ (Cx\ (real\_of\_nat\ n)) = (0::real)$

thm ARG_EQ_PI:

$\forall z::(real, 2)\ cart.\ (Arg\ z = pi) = (HOL\_Light\_Import.real\ z \wedge Re\ z < (0::real))$

thm ARG_EQ_0_PI:

$\forall z::(real, 2)\ cart.\ (Arg\ z = (0::real) \vee Arg\ z = pi) = HOL\_Light\_Import.real\ z$

thm ARG_INV:

$\forall z::(real, 2)\ cart.\ \neg\ (HOL\_Light\_Import.real\ z \wedge (0::real) \leq Re\ z) \longrightarrow Arg\ (complex\_inv\ z) = real\_of\_nat\ (2::nat) * pi - Arg\ z$

thm ARG_EQ:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow (Arg\ w = Arg\ z) = (\exists x{>}0::real.\ w = complex\_mul\ (Cx\ x)\ z)$

thm ARG_INV_EQ_0:

$\forall z::(real,\ 2)\ cart.\ (Arg\ (complex\_inv\ z) = (0::real)) = (Arg\ z = (0::real))$

thm ARG_LE_DIV_SUM:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \wedge Arg\ w \leq Arg\ z \longrightarrow Arg\ z = Arg\ w + Arg\ (complex\_div\ z\ w)$

thm ARG_LE_DIV_SUM_EQ:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow (Arg\ w \leq Arg\ z) = (Arg\ z = Arg\ w + Arg\ (complex\_div\ z\ w))$

thm REAL_SUB_ARG:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow Arg\ w - Arg\ z = (if\ Arg\ z \leq Arg\ w\ then\ Arg\ (complex\_div\ w\ z)\ else\ Arg\ (complex\_div\ w\ z) - real\_of\_nat\ (2::nat) * pi)$

thm REAL_ADD_ARG:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow Arg\ w + Arg\ z = (if\ Arg\ w + Arg\ z < real\_of\_nat\ (2::nat) * pi\ then\ Arg\ (complex\_mul\ w\ z)\ else\ Arg\ (complex\_mul\ w\ z) + real\_of\_nat\ (2::nat) * pi)$

thm ARG_MUL:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow Arg\ (complex\_mul\ w\ z) = (if\ Arg\ w + Arg\ z < real\_of\_nat\ (2::nat) * pi\ then\ Arg\ w + Arg\ z\ else\ Arg\ w + Arg\ z - real\_of\_nat\ (2::nat) * pi)$

thm DEF_rotate2d:

$rotate2d = (\lambda(\_1877163::real)\ \_1877164::(real,\ 2)\ cart.\ vector\ [\$\ \_1877164\ (1::nat) * cos\ \_1877163 - \$\ \_1877164\ (2::nat) * sin\ \_1877163,\ \$\ \_1877164\ (1::nat) * sin\ \_1877163 + \$\ \_1877164\ (2::nat) * cos\ \_1877163])$

thm rotate2d:

$\forall (x::(real,\ 2)\ cart)\ t::real.\ rotate2d\ t\ x = vector\ [\$\ x\ (1::nat) * cos\ t - \$\ x\ (2::nat) * sin\ t,\ \$\ x\ (1::nat) * sin\ t + \$\ x\ (2::nat) * cos\ t]$

thm LINEAR_ROTATE2D:

$\forall t::real.\ linear\ (rotate2d\ t)$

thm ROTATE2D_SUB:

$\forall (t::real)\ (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ rotate2d\ t\ (vector\_sub\ w\ z) = vector\_sub\ (rotate2d\ t\ w)\ (rotate2d\ t\ z)$

thm NORM_ROTATE2D:

$\forall (t::real)\ z::(real,\ 2)\ cart.\ vector\_norm\ (rotate2d\ t\ z) = vector\_norm\ z$

thm ROTATE2D_0:

$\forall t::real.\ rotate2d\ t\ (Cx\ (0::real)) = Cx\ (0::real)$

thm ROTATE2D_EQ_0:

$\forall (t::real)\ z::(real,\ 2)\ cart.\ (rotate2d\ t\ z = Cx\ (0::real)) = (z = Cx\ (0::real))$

thm ROTATE2D_ZERO:

$\forall z::(real,\ 2)\ cart.\ rotate2d\ (0::real)\ z = z$

thm ORTHOGONAL_TRANSFORMATION_ROTATE2D:

$\forall t::real.\ orthogonal\_transformation\ (rotate2d\ t)$

thm ROTATE2D_POLAR:

$\forall (r::real)\ (t::real)\ s::real.\ rotate2d\ t\ (vector\ [r * cos\ s,\ r * sin\ s]) = vector\ [r * cos\ (t + s),\ r * sin\ (t + s)]$

thm MATRIX_ROTATE2D:

$\forall t::real.\ matrix\ (rotate2d\ t) = vector\ [vector\ [cos\ t,\ -\ sin\ t],\ vector\ [sin\ t,\ cos\ t]]$

thm DET_MATRIX_ROTATE2D:

$\forall t::real.\ det\ (matrix\ (rotate2d\ t)) = (1::real)$

thm ROTATION_ROTATE2D:

$\forall f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ orthogonal\_transformation\ f \wedge det\ (matrix\ f) = (1::real) \longrightarrow (\exists t \geq 0::real.\ t < real\_of\_nat\ (2::nat) * pi \wedge f = rotate2d\ t)$

thm ROTATE2D_ADD:

$\forall (s::real)\ (t::real)\ x::(real,\ 2)\ cart.\ rotate2d\ (s + t)\ x = rotate2d\ s\ (rotate2d\ t\ x)$

thm ROTATE2D_COMPLEX:

$\forall (t::real)\ z::(real,\ 2)\ cart.\ rotate2d\ t\ z = complex\_mul\ (cexp\ (complex\_mul\ ii\ (Cx\ t)))\ z$

thm ROTATE2D_PI2:

$\forall z::(real,\ 2)\ cart.\ rotate2d\ (pi\ /\ real\_of\_nat\ (2::nat))\ z = complex\_mul\ ii\ z$

thm ROTATE2D_PI:

$\forall z::(real,\ 2)\ cart.\ rotate2d\ pi\ z = vector\_neg\ z$

thm ROTATE2D_NPI:

$\forall (n::nat)\ z::(real,\ 2)\ cart.\ rotate2d\ (real\_of\_nat\ n * pi)\ z = complex\_mul\ (complex\_pow\ (vector\_neg\ (Cx\ (1::real)))\ n)\ z$

thm ROTATE2D_2PI:

$\forall\, z$::(*real*, *2*) *cart*. *rotate2d* (*real_of_nat* (*2*::*nat*) $*$ *pi*) *z* = *z*

thm ARG_ROTATE2D:

$\forall$ (*t*::*real*) *z*::(*real*, *2*) *cart*. *z* $\neq$ *Cx* (*0*::*real*) $\wedge$ (*0*::*real*) $\leq$ *t* + *Arg z* $\wedge$ *t* + *Arg* *z* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\longrightarrow$ *Arg* (*rotate2d t z*) = *t* + *Arg z*

thm ARG_ROTATE2D_UNIQUE:

$\forall$ (*t*::*real*) (*a*::*real*) *z*::(*real*, *2*) *cart*. *z* $\neq$ *Cx* (*0*::*real*) $\wedge$ *Arg* (*rotate2d t z*) = *a* $\longrightarrow$ ($\exists$ *n*::*real*. *integer n* $\wedge$ *t* = *real_of_nat* (*2*::*nat*) $*$ (*n* $*$ *pi*) + (*a* $-$ *Arg z*))

thm ARG_ROTATE2D_UNIQUE_2PI:

$\forall$ (*s*::*real*) (*t*::*real*) *z*::(*real*, *2*) *cart*. *z* $\neq$ *Cx* (*0*::*real*) $\wedge$ (*0*::*real*) $\leq$ *s* $\wedge$ *s* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\wedge$ (*0*::*real*) $\leq$ *t* $\wedge$ *t* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\wedge$ *Arg* (*rotate2d s z*) = *Arg* (*rotate2d t z*) $\longrightarrow$ *s* = *t*

thm COMPLEX_DIV_ROTATION:

$\forall$ (*f*::(*real*, *2*) *cart* $\Rightarrow$ (*real*, *2*) *cart*) (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *orthogonal_transformation* *f* $\wedge$ *det* (*matrix f*) = (*1*::*real*) $\longrightarrow$ *complex_div* (*f w*) (*f z*) = *complex_div w z*

thm ROTATION_ROTATE2D_EXISTS_GEN:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. $\exists$ *t*$\geq$*0*::*real*. *t* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\wedge$ % (*vector_norm y*) (*rotate2d t x*) = % (*vector_norm x*) *y*

thm ROTATION_ROTATE2D_EXISTS:

$\forall$ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *vector_norm x* = *vector_norm y* $\longrightarrow$ ($\exists$ *t*$\geq$*0*::*real*. *t* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\wedge$ *rotate2d t x* = *y*)

thm ROTATION_ROTATE2D_EXISTS_ORTHOGONAL:

$\forall$ (*e1*::(*real*, *2*) *cart*) *e2*::(*real*, *2*) *cart*. *vector_norm e1* = (*1*::*real*) $\wedge$ *vector_norm* *e2* = (*1*::*real*) $\wedge$ *orthogonal e1 e2* $\longrightarrow$ *e1* = *rotate2d* (*pi* / *real_of_nat* (*2*::*nat*)) *e2* $\vee$ *e2* = *rotate2d* (*pi* / *real_of_nat* (*2*::*nat*)) *e1*

thm ROTATION_ROTATE2D_EXISTS_ORTHOGONAL_ORIENTED:

$\forall$ (*e1*::(*real*, *2*) *cart*) *e2*::(*real*, *2*) *cart*. *vector_norm e1* = (*1*::*real*) $\wedge$ *vector_norm* *e2* = (*1*::*real*) $\wedge$ *orthogonal e1 e2* $\wedge$ (*0*::*real*) < $ *e1* (*1*::*nat*) $*$ $ *e2* (*2*::*nat*) $-$ $ *e1* (*2*::*nat*) $*$ $ *e2* (*1*::*nat*) $\longrightarrow$ *e2* = *rotate2d* (*pi* / *real_of_nat* (*2*::*nat*)) *e1*

thm ROTATE2D_EQ:

$\forall$ (*t*::*real*) (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. (*rotate2d t x* = *rotate2d t y*) = (*x* = *y*)

thm ROTATE2D_SUB_ARG:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *z* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *rotate2d* (*Arg w* $-$ *Arg z*) = *rotate2d* (*Arg* (*complex_div w z*))

thm ROTATION_MATRIX_ROTATE2D:

$\forall$ *t*::*real*. *rotation_matrix* (*matrix* (*rotate2d t*))

thm ROTATION_MATRIX_ROTATE2D_EQ:

$\forall$ *A*::((*real*, *2*) *cart*, *2*) *cart*. *rotation_matrix A* = ($\exists$ *t*::*real*. *A* = *matrix* (*rotate2d* *t*))

thm DEF_ctan:

ctan = (λ*_1877408*::(*real*, *2*) *cart*. *complex_div* (*csin _1877408*) (*ccos _1877408*))

thm ctan:

$\forall$ *z*::(*real*, *2*) *cart*. *ctan z* = *complex_div* (*csin z*) (*ccos z*)

thm CTAN_0:

ctan (*Cx* (*0*::*real*)) = *Cx* (*0*::*real*)

thm CTAN_NEG:

$\forall$ *z*::(*real*, *2*) *cart*. *ctan* (*vector_neg z*) = *vector_neg* (*ctan z*)

thm CTAN_ADD:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos z* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos* (*vector_add w z*) $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *ctan* (*vector_add w z*) = *complex_div* (*vector_add* (*ctan w*) (*ctan z*)) (*vector_sub* (*Cx* (*1*::*real*)) (*complex_mul* (*ctan w*) (*ctan z*)))

thm CTAN_DOUBLE:

$\forall$ *z*::(*real*, *2*) *cart*. *ccos z* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) *z*) $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *ctan* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) *z*) = *complex_div* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*ctan z*)) (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow* (*ctan z*) (*2*::*nat*)))

thm CTAN_SUB:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos z* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos* (*vector_sub w z*) $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *ctan* (*vector_sub w z*) = *complex_div* (*vector_sub* (*ctan w*) (*ctan z*)) (*vector_add* (*Cx* (*1*::*real*)) (*complex_mul* (*ctan w*) (*ctan z*)))

thm COMPLEX_ADD_CTAN:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos z* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *vector_add* (*ctan w*) (*ctan z*) = *complex_div* (*csin* (*vector_add w z*)) (*complex_mul* (*ccos w*) (*ccos z*))

thm COMPLEX_SUB_CTAN:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *ccos w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *ccos z* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *vector_sub* (*ctan w*) (*ctan z*) = *complex_div* (*csin* (*vector_sub w z*)) (*complex_mul* (*ccos w*) (*ccos z*))

thm HAS_COMPLEX_DERIVATIVE_CTAN:

$\forall z::(real, 2)\ cart.\ ccos\ z \neq Cx\ (0::real) \longrightarrow has\_complex\_derivative\ ctan\ (complex\_inv\ (complex\_pow\ (ccos\ z)\ (2::nat)))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CTAN:

$\forall z::(real, 2)\ cart.\ ccos\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ ctan\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CTAN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ ccos\ z \neq Cx\ (0::real) \longrightarrow complex\_differentiable\ ctan\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CTAN:

$\forall z::(real, 2)\ cart.\ ccos\ z \neq Cx\ (0::real) \longrightarrow continuous\ ctan\ (at\ z)$

thm CONTINUOUS_WITHIN_CTAN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ ccos\ z \neq Cx\ (0::real) \longrightarrow continuous\ ctan\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CTAN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \longrightarrow ccos\ z \neq Cx\ (0::real)) \longrightarrow continuous\_on\ ctan\ s$

thm HOLOMORPHIC_ON_CTAN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \longrightarrow ccos\ z \neq Cx\ (0::real)) \longrightarrow holomorphic\_on\ ctan\ s$

thm DEF_tan:

$tan = (\lambda\_1877611::real.\ Re\ (ctan\ (Cx\ \_1877611)))$

thm tan_def:

$\forall x::real.\ tan\ x = Re\ (ctan\ (Cx\ x))$

thm CNJ_CTAN:

$\forall z::(real, 2)\ cart.\ cnj\ (ctan\ z) = ctan\ (cnj\ z)$

thm REAL_TAN:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ z \longrightarrow HOL\_Light\_Import.real\ (ctan\ z)$

thm CX_TAN:

$\forall x::real.\ Cx\ (tan\ x) = ctan\ (Cx\ x)$

thm tan:

$\forall x::real.\ tan\ x = sin\ x\ /\ cos\ x$

thm TAN_0:

$tan\ (0::real) = (0::real)$

thm TAN_PI:

$tan\ pi = (0::real)$

thm TAN_NPI:

$\forall\, n::nat.\ tan\ (real\_of\_nat\ n * pi) = (0::real)$

thm TAN_NEG:

$\forall\, x::real.\ tan\ (-\ x) = -\ tan\ x$

thm TAN_PERIODIC_PI:

$\forall\, x::real.\ tan\ (x + pi) = tan\ x$

thm TAN_PERIODIC_NPI:

$\forall\, (x::real)\ n::nat.\ tan\ (x + real\_of\_nat\ n * pi) = tan\ x$

thm TAN_ADD:

$\forall\, (x::real)\ y::real.\ cos\ x \neq (0::real) \wedge cos\ y \neq (0::real) \wedge cos\ (x + y) \neq (0::real) \longrightarrow tan\ (x + y) = (tan\ x + tan\ y)\ /\ ((1::real) - tan\ x * tan\ y)$

thm TAN_SUB:

$\forall\, (x::real)\ y::real.\ cos\ x \neq (0::real) \wedge cos\ y \neq (0::real) \wedge cos\ (x - y) \neq (0::real) \longrightarrow tan\ (x - y) = (tan\ x - tan\ y)\ /\ ((1::real) + tan\ x * tan\ y)$

thm TAN_DOUBLE:

$\forall\, x::real.\ cos\ x \neq (0::real) \wedge cos\ (real\_of\_nat\ (2::nat) * x) \neq (0::real) \longrightarrow tan\ (real\_of\_nat\ (2::nat) * x) = real\_of\_nat\ (2::nat) * tan\ x\ /\ ((1::real) - (tan\ x)^2)$

thm REAL_ADD_TAN:

$\forall\, (x::real)\ y::real.\ cos\ x \neq (0::real) \wedge cos\ y \neq (0::real) \longrightarrow tan\ x + tan\ y = sin\ (x + y)\ /\ (cos\ x * cos\ y)$

thm REAL_SUB_TAN:

$\forall\, (x::real)\ y::real.\ cos\ x \neq (0::real) \wedge cos\ y \neq (0::real) \longrightarrow tan\ x - tan\ y = sin\ (x - y)\ /\ (cos\ x * cos\ y)$

thm TAN_PI4:

$tan\ (pi\ /\ real\_of\_nat\ (4::nat)) = (1::real)$

thm TAN_POS_PI2:

$\forall\, x::real.\ (0::real) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) < tan\ x$

thm TAN_POS_PI2_LE:

$\forall\, x::real.\ (0::real) \leq x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) \leq tan\ x$

thm COS_TAN:

$\forall x::real.\ |x| < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow cos\ x = (1::real)\ /\ sqrt\ ((1::real) + (tan\ x)^2)$

thm SIN_TAN:

$\forall x::real.\ |x| < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow sin\ x = tan\ x\ /\ sqrt\ ((1::real) + (tan\ x)^2)$

thm SIN_MONO_LT:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) \le x \wedge x < y \wedge y \le pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow sin\ x < sin\ y$

thm SIN_MONO_LE:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) \le x \wedge x \le y \wedge y \le pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow sin\ x \le sin\ y$

thm SIN_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) \le x \wedge x \le pi\ /\ real\_of\_nat\ (2::nat) \wedge - (pi\ /\ real\_of\_nat\ (2::nat)) \le y \wedge y \le pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (sin\ x < sin\ y) = (x < y)$

thm SIN_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) \le x \wedge x \le pi\ /\ real\_of\_nat\ (2::nat) \wedge - (pi\ /\ real\_of\_nat\ (2::nat)) \le y \wedge y \le pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (sin\ x \le sin\ y) = (x \le y)$

thm SIN_INJ_PI:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) \le x \wedge x \le pi\ /\ real\_of\_nat\ (2::nat) \wedge - (pi\ /\ real\_of\_nat\ (2::nat)) \le y \wedge y \le pi\ /\ real\_of\_nat\ (2::nat) \wedge sin\ x = sin\ y \longrightarrow x = y$

thm COS_MONO_LT:

$\forall (x::real)\ y::real.\ (0::real) \le x \wedge x < y \wedge y \le pi \longrightarrow cos\ y < cos\ x$

thm COS_MONO_LE:

$\forall (x::real)\ y::real.\ (0::real) \le x \wedge x \le y \wedge y \le pi \longrightarrow cos\ y \le cos\ x$

thm COS_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ (0::real) \le x \wedge x \le pi \wedge (0::real) \le y \wedge y \le pi \longrightarrow (cos\ x < cos\ y) = (y < x)$

thm COS_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ (0::real) \le x \wedge x \le pi \wedge (0::real) \le y \wedge y \le pi \longrightarrow (cos\ x \le cos\ y) = (y \le x)$

thm COS_INJ_PI:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge x \leq pi \wedge (0::real) \leq y \wedge y \leq pi \wedge cos\ x = cos\ y \longrightarrow x = y$

thm TAN_MONO_LT:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < y \wedge y < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow tan\ x < tan\ y$

thm TAN_MONO_LE:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x \leq y \wedge y < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow tan\ x \leq tan\ y$

thm TAN_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \wedge - (pi\ /\ real\_of\_nat\ (2::nat)) < y \wedge y < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (tan\ x < tan\ y) = (x < y)$

thm TAN_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ - (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \wedge - (pi\ /\ real\_of\_nat\ (2::nat)) < y \wedge y < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (tan\ x \leq tan\ y) = (x \leq y)$

thm TAN_BOUND_PI2:

$\forall x::real.\ |x| < pi\ /\ real\_of\_nat\ (4::nat) \longrightarrow |tan\ x| < (1::real)$

thm TAN_COT:

$\forall x::real.\ tan\ (pi\ /\ real\_of\_nat\ (2::nat) - x) = inverse\_class.inverse\ (tan\ x)$

thm SIN_PI6_STRADDLE:

$\forall (a::real)\ b::real.\ (0::real) \leq a \wedge a \leq b \wedge b \leq real\_of\_nat\ (4::nat) \wedge sin\ (a\ /\ real\_of\_nat\ (6::nat)) \leq (1::real)\ /\ real\_of\_nat\ (2::nat) \wedge (1::real)\ /\ real\_of\_nat\ (2::nat) \leq sin\ (b\ /\ real\_of\_nat\ (6::nat)) \longrightarrow a \leq pi \wedge pi \leq b$

thm PI_APPROX_32:

$|pi - real\_of\_nat\ (13493037705::nat)\ /\ real\_of\_nat\ (4294967296::nat)| \leq inverse\_class.inverse\ (real\_of\_nat\ (2::nat))^{32::nat}$

thm PI2_BOUNDS:

$(0::real) < pi\ /\ real\_of\_nat\ (2::nat) \wedge pi\ /\ real\_of\_nat\ (2::nat) < real\_of\_nat\ (2::nat)$

thm DEF_clog:

$clog = (\lambda\_1877898::(real,\ 2)\ cart.\ SOME\ w::(real,\ 2)\ cart.\ cexp\ w = \_1877898 \wedge - pi < Im\ w \wedge Im\ w \leq pi)$

thm clog:

$\forall z::(real,\ 2)\ cart.\ clog\ z = (SOME\ w::(real,\ 2)\ cart.\ cexp\ w = z \wedge - pi < Im\ w \wedge Im\ w \leq pi)$

thm EXISTS_COMPLEX′:

$\forall P::real \Rightarrow real \Rightarrow bool.\ (\exists z::(real,\ 2)\ cart.\ P\ (Re\ z)\ (Im\ z)) = (\exists (x::real)\ y::real.\ P\ x\ y)$

thm CLOG_WORKS:

$\forall z::(real,\ 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow cexp\ (clog\ z) = z \wedge -\ pi < Im\ (clog\ z) \wedge Im\ (clog\ z) \leq pi$

thm CEXP_CLOG:

$\forall z::(real,\ 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow cexp\ (clog\ z) = z$

thm CLOG_CEXP:

$\forall z::(real,\ 2)\ cart.\ -\ pi < Im\ z \wedge Im\ z \leq pi \longrightarrow clog\ (cexp\ z) = z$

thm CLOG_EQ:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \neq Cx\ (0::real) \wedge z \neq Cx\ (0::real) \longrightarrow (clog\ w = clog\ z) = (w = z)$

thm CLOG_UNIQUE:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ -\ pi < Im\ z \wedge Im\ z \leq pi \wedge cexp\ z = w \longrightarrow clog\ w = z$

thm RE_CLOG:

$\forall z::(real,\ 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow Re\ (clog\ z) = log\ (vector\_norm\ z)$

thm HAS_COMPLEX_DERIVATIVE_CLOG:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow has\_complex\_derivative\ clog\ (complex\_inv\ z)\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CLOG:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow complex\_differentiable\ clog\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CLOG:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow complex\_differentiable\ clog\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CLOG:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow continuous\ clog\ (at\ z)$

thm CONTINUOUS_WITHIN_CLOG:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow continuous\ clog\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CLOG:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \land Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow continuous\_on\ clog\ s$

thm HOLOMORPHIC_ON_CLOG:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \land Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow holomorphic\_on\ clog\ s$

thm CX_LOG:

$\forall z{>}0::real.\ Cx\ (log\ z) = clog\ (Cx\ z)$

thm RE_CLOG_POS_LT:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow (|Im\ (clog\ z)| < pi\ /\ real\_of\_nat\ (2::nat)) = ((0::real) < Re\ z)$

thm RE_CLOG_POS_LE:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow (|Im\ (clog\ z)| \leq pi\ /\ real\_of\_nat\ (2::nat)) = ((0::real) \leq Re\ z)$

thm IM_CLOG_POS_LT:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow ((0::real) < Im\ (clog\ z) \land Im\ (clog\ z) < pi) = ((0::real) < Im\ z)$

thm IM_CLOG_POS_LE:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow ((0::real) \leq Im\ (clog\ z)) = ((0::real) \leq Im\ z)$

thm RE_CLOG_POS_LT_IMP:

$\forall z::(real, 2)\ cart.\ (0::real) < Re\ z \longrightarrow |Im\ (clog\ z)| < pi\ /\ real\_of\_nat\ (2::nat)$

thm IM_CLOG_POS_LT_IMP:

$\forall z::(real, 2)\ cart.\ (0::real) < Im\ z \longrightarrow (0::real) < Im\ (clog\ z) \land Im\ (clog\ z) < pi$

thm IM_CLOG_EQ_0:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow (Im\ (clog\ z) = (0::real)) = ((0::real) < Re\ z \land Im\ z = (0::real))$

thm IM_CLOG_EQ_PI:

$\forall z::(real, 2)\ cart.\ z \neq Cx\ (0::real) \longrightarrow (Im\ (clog\ z) = pi) = (Re\ z < (0::real) \land Im\ z = (0::real))$

thm CNJ_CLOG:

$\forall z::(real, 2)\ cart.\ (Im\ z = (0::real) \longrightarrow (0::real) < Re\ z) \longrightarrow cnj\ (clog\ z) = clog\ (cnj\ z)$

thm CLOG_INV:

$\forall z$::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *clog* (*complex_inv z*) = *vector_neg* (*clog z*)

thm CLOG_1:

clog (*Cx* (*1*::*real*)) = *Cx* (*0*::*real*)

thm CLOG_NEG_1:

clog (*vector_neg* (*Cx* (*1*::*real*))) = *complex_mul ii* (*Cx pi*)

thm CLOG_II:

clog ii = *complex_mul ii* (*Cx* (*pi* / *real_of_nat* (*2*::*nat*)))

thm CLOG_NEG_II:

clog (*vector_neg ii*) = *complex_mul* (*vector_neg ii*) (*Cx* (*pi* / *real_of_nat* (*2*::*nat*)))

thm CSQRT_CEXP_CLOG:

$\forall z$::(*real*, *2*) *cart*. *z* ≠ *Cx* (*0*::*real*) ⟶ *csqrt z* = *cexp* (*complex_div* (*clog z*) (*Cx* (*real_of_nat* (*2*::*nat*))))

thm CNJ_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) ≤ *Re z*) ⟶ *cnj* (*csqrt z*) = *csqrt* (*cnj z*)

thm HAS_COMPLEX_DERIVATIVE_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *has_complex_derivative csqrt* (*complex_inv* (*complex_mul* (*Cx* (*real_of_nat* (*2*::*nat*))) (*csqrt z*))) (*at z*)

thm COMPLEX_DIFFERENTIABLE_AT_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *complex_differentiable csqrt* (*at z*)

thm COMPLEX_DIFFERENTIABLE_WITHIN_CSQRT:

$\forall$ (*s*::(*real*, *2*) *cart* ⇒ *bool*) *z*::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *complex_differentiable csqrt* (*within* (*at z*) *s*)

thm CONTINUOUS_AT_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *continuous csqrt* (*at z*)

thm CONTINUOUS_WITHIN_CSQRT:

$\forall$ (*s*::(*real*, *2*) *cart* ⇒ *bool*) *z*::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *continuous csqrt* (*within* (*at z*) *s*)

thm CONTINUOUS_ON_CSQRT:

$\forall s$::(*real*, *2*) *cart* ⇒ *bool*. ($\forall z$::(*real*, *2*) *cart*. *IN z s* ∧ *Im z* = (*0*::*real*) ⟶ (*0*::*real*) < *Re z*) ⟶ *continuous_on csqrt s*

thm HOLOMORPHIC_ON_CSQRT:

$\forall$ *s::(real, 2) cart ⇒ bool. (∀ z::(real, 2) cart. IN z s ∧ Im z = (0::real) ⟶ (0::real) < Re z) ⟶ holomorphic_on csqrt s*

thm CONTINUOUS_WITHIN_CSQRT_POSREAL:

$\forall$ *z::(real, 2) cart. continuous csqrt (within (at z) (GSPEC (λGEN%PVAR%2347::(real, 2) cart. ∃ w::(real, 2) cart. SETSPEC GEN%PVAR%2347 (HOL_Light_Import.real w ∧ (0::real) ≤ Re w) w)))*

thm DEF_cpow:

cpow = (λ(_1878899::(real, 2) cart) _1878900::(real, 2) cart. if _1878899 = Cx (0::real) then Cx (0::real) else cexp (complex_mul _1878900 (clog _1878899)))

thm cpow:

$\forall$ *(z::(real, 2) cart) w::(real, 2) cart. cpow w z = (if w = Cx (0::real) then Cx (0::real) else cexp (complex_mul z (clog w)))*

thm CPOW_0:

$\forall$ *z::(real, 2) cart. cpow (Cx (0::real)) z = Cx (0::real)*

thm CPOW_N:

$\forall$ *z::(real, 2) cart. cpow z (Cx (real_of_nat (?n::nat))) = (if z = Cx (0::real) then Cx (0::real) else complex_pow z ?n)*

thm CPOW_1:

$\forall$ *z::(real, 2) cart. cpow (Cx (1::real)) z = Cx (1::real)*

thm CPOW_ADD:

$\forall$ *(w::(real, 2) cart) (z1::(real, 2) cart) z2::(real, 2) cart. cpow w (vector_add z1 z2) = complex_mul (cpow w z1) (cpow w z2)*

thm CPOW_NEG:

$\forall$ *(w::(real, 2) cart) z::(real, 2) cart. cpow w (vector_neg z) = complex_inv (cpow w z)*

thm CPOW_SUB:

$\forall$ *(w::(real, 2) cart) (z1::(real, 2) cart) z2::(real, 2) cart. cpow w (vector_sub z1 z2) = complex_div (cpow w z1) (cpow w z2)*

thm CPOW_EQ_0:

$\forall$ *(w::(real, 2) cart) z::(real, 2) cart. (cpow w z = Cx (0::real)) = (w = Cx (0::real))*

thm NORM_CPOW_REAL:

$\forall$ *(w::(real, 2) cart) z::(real, 2) cart. HOL_Light_Import.real w ∧ (0::real) < Re w ⟶ vector_norm (cpow w z) = exp (Re z * log (Re w))*

thm CPOW_REAL_REAL:

$\forall (w::(real, 2) \ cart) \ z::(real, 2) \ cart. \ HOL\_Light\_Import.real \ w \land HOL\_Light\_Import.real \ z \land (0::real) < Re \ w \longrightarrow cpow \ w \ z = Cx \ (exp \ (Re \ z * log \ (Re \ w)))$

thm NORM_CPOW_REAL_MONO:

$\forall (w::(real, 2) \ cart) \ (z1::(real, 2) \ cart) \ z2::(real, 2) \ cart. \ HOL\_Light\_Import.real \ w \land (1::real) < Re \ w \longrightarrow (vector\_norm \ (cpow \ w \ z1) \le vector\_norm \ (cpow \ w \ z2)) = (Re \ z1 \le Re \ z2)$

thm CPOW_MUL_REAL:

$\forall (x::(real, 2) \ cart) \ (y::(real, 2) \ cart) \ z::(real, 2) \ cart. \ HOL\_Light\_Import.real \ x \land HOL\_Light\_Import.real \ y \land (0::real) \le Re \ x \land (0::real) \le Re \ y \longrightarrow cpow \ (complex\_mul \ x \ y) \ z = complex\_mul \ (cpow \ x \ z) \ (cpow \ y \ z)$

thm HAS_COMPLEX_DERIVATIVE_CPOW:

$\forall (s::(real, 2) \ cart) \ z::(real, 2) \ cart. \ (Im \ z = (0::real) \longrightarrow (0::real) < Re \ z) \longrightarrow has\_complex\_derivative \ (\lambda z::(real, 2) \ cart. \ cpow \ z \ s) \ (complex\_mul \ s \ (cpow \ z \ (vector\_sub \ s \ (Cx \ (1::real))))) \ (at \ z)$

thm HAS_COMPLEX_DERIVATIVE_CPOW_RIGHT:

$\forall (w::(real, 2) \ cart) \ z::(real, 2) \ cart. \ w \neq Cx \ (0::real) \longrightarrow has\_complex\_derivative \ (cpow \ w) \ (complex\_mul \ (clog \ w) \ (cpow \ w \ z)) \ (at \ z)$

thm COMPLEX_DIFFERENTIABLE_CPOW_RIGHT:

$\forall (w::(real, 2) \ cart) \ z::(real, 2) \ cart. \ w \neq Cx \ (0::real) \longrightarrow complex\_differentiable \ (cpow \ w) \ (at \ z)$

thm HOLOMORPHIC_ON_CPOW_RIGHT:

$\forall (w::(real, 2) \ cart) \ (f::(real, 2) \ cart \Rightarrow (real, 2) \ cart) \ s::(real, 2) \ cart \Rightarrow bool. \ w \neq Cx \ (0::real) \land holomorphic\_on \ f \ s \longrightarrow holomorphic\_on \ (\lambda z::(real, 2) \ cart. \ cpow \ w \ (f \ z)) \ s$

thm CLOG_MUL:

$\forall (w::(real, 2) \ cart) \ z::(real, 2) \ cart. \ w \neq Cx \ (0::real) \land z \neq Cx \ (0::real) \longrightarrow clog \ (complex\_mul \ w \ z) = (if \ Im \ (vector\_add \ (clog \ w) \ (clog \ z)) \le - \ pi \ then \ vector\_add \ (vector\_add \ (clog \ w) \ (clog \ z)) \ (complex\_mul \ ii \ (Cx \ (real\_of\_nat \ (2::nat) * pi))) \ else \ if \ pi < Im \ (vector\_add \ (clog \ w) \ (clog \ z)) \ then \ vector\_sub \ (vector\_add \ (clog \ w) \ (clog \ z)) \ (complex\_mul \ ii \ (Cx \ (real\_of\_nat \ (2::nat) * pi))) \ else \ vector\_add \ (clog \ w) \ (clog \ z))$

thm CLOG_MUL_SIMPLE:

$\forall (w::(real, 2) \ cart) \ z::(real, 2) \ cart. \ w \neq Cx \ (0::real) \land z \neq Cx \ (0::real) \land - \ pi < Im \ (clog \ w) + Im \ (clog \ z) \land Im \ (clog \ w) + Im \ (clog \ z) \le pi \longrightarrow clog \ (complex\_mul \ w \ z) = vector\_add \ (clog \ w) \ (clog \ z)$

thm CLOG_NEG:

∀ z::(real, 2) cart. z ≠ Cx (0::real) ⟶ clog (vector_neg z) = (if Im z ≤ (0::real) ∧ ¬ (Re z < (0::real) ∧ Im z = (0::real)) then vector_add (clog z) (complex_mul ii (Cx pi)) else vector_sub (clog z) (complex_mul ii (Cx pi)))

thm CLOG_MUL_II:

∀ z::(real, 2) cart. z ≠ Cx (0::real) ⟶ clog (complex_mul ii z) = (if (0::real) ≤ Re z ∨ Im z < (0::real) then vector_add (clog z) (complex_mul ii (Cx (pi / real_of_nat (2::nat)))) else vector_sub (clog z) (complex_mul ii (Cx (real_of_nat (3::nat) ∗ (pi / real_of_nat (2::nat))))))

thm DEF_unwinding:

unwinding = (λ_1879039::(real, 2) cart. complex_div (vector_sub _1879039 (clog (cexp _1879039))) (complex_mul (Cx (real_of_nat (2::nat) ∗ pi)) ii))

thm unwinding:

∀ z::(real, 2) cart. unwinding z = complex_div (vector_sub z (clog (cexp z))) (complex_mul (Cx (real_of_nat (2::nat) ∗ pi)) ii)

thm UNWINDING_2PI:

complex_mul (Cx (real_of_nat (2::nat) ∗ pi)) (complex_mul ii (unwinding (?z::(real, 2) cart))) = vector_sub ?z (clog (cexp ?z))

thm CLOG_MUL_UNWINDING:

∀ (w::(real, 2) cart) z::(real, 2) cart. w ≠ Cx (0::real) ∧ z ≠ Cx (0::real) ⟶ clog (complex_mul w z) = vector_add (clog w) (vector_sub (clog z) (complex_mul (Cx (real_of_nat (2::nat) ∗ pi)) (complex_mul ii (unwinding (vector_add (clog w) (clog z))))))

thm DEF_catn:

catn = (λ_1879044::(real, 2) cart. complex_mul (complex_div ii (Cx (real_of_nat (2::nat)))) (clog (complex_div (vector_sub (Cx (1::real)) (complex_mul ii _1879044)) (vector_add (Cx (1::real)) (complex_mul ii _1879044)))))

thm catn:

∀ z::(real, 2) cart. catn z = complex_mul (complex_div ii (Cx (real_of_nat (2::nat)))) (clog (complex_div (vector_sub (Cx (1::real)) (complex_mul ii z)) (vector_add (Cx (1::real)) (complex_mul ii z))))

thm IM_COMPLEX_DIV_LEMMA:

∀ z::(real, 2) cart. (Im (complex_div (vector_sub (Cx (1::real)) (complex_mul ii z)) (vector_add (Cx (1::real)) (complex_mul ii z))) = (0::real)) = (Re z = (0::real))

thm RE_COMPLEX_DIV_LEMMA:

∀ z::(real, 2) cart. ((0::real) < Re (complex_div (vector_sub (Cx (1::real)) (complex_mul ii z)) (vector_add (Cx (1::real)) (complex_mul ii z)))) = (vector_norm z < (1::real))

thm CTAN_CATN:

$\forall z::(real, 2)\ cart.\ complex\_pow\ z\ (2::nat) \neq vector\_neg\ (Cx\ (1::real)) \longrightarrow ctan\ (catn\ z) = z$

thm CATN_CTAN:

$\forall z::(real, 2)\ cart.\ |Re\ z| < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow catn\ (ctan\ z) = z$

thm RE_CATN_BOUNDS:

$\forall z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow |Re\ (catn\ z)| < pi\ /\ real\_of\_nat\ (2::nat)$

thm HAS_COMPLEX_DERIVATIVE_CATN:

$\forall z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow has\_complex\_derivative\ catn\ (complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat))))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CATN:

$\forall z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow complex\_differentiable\ catn\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CATN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow complex\_differentiable\ catn\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CATN:

$\forall z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow continuous\ catn\ (at\ z)$

thm CONTINUOUS_WITHIN_CATN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow continuous\ catn\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CATN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \wedge Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow continuous\_on\ catn\ s$

thm HOLOMORPHIC_ON_CATN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \wedge Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow holomorphic\_on\ catn\ s$

thm DEF_atn:

$atn = (\lambda\_1879588::real.\ Re\ (catn\ (Cx\ \_1879588)))$

thm atn:

$\forall x::real.\ atn\ x = Re\ (catn\ (Cx\ x))$

thm CX_ATN:

$\forall x::real.\ Cx\ (atn\ x) = catn\ (Cx\ x)$

thm ATN_TAN:

$\forall y::real.\ tan\ (atn\ y) = y$

thm ATN_BOUND:

$\forall y::real.\ |atn\ y| < pi\ /\ real\_of\_nat\ (2::nat)$

thm ATN_BOUNDS:

$\forall y::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) < atn\ y \wedge atn\ y < pi\ /\ real\_of\_nat\ (2::nat)$

thm TAN_ATN:

$\forall x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow atn\ (tan\ x) = x$

thm ATN_0:

$atn\ (0::real) = (0::real)$

thm ATN_1:

$atn\ (1::real) = pi\ /\ real\_of\_nat\ (4::nat)$

thm ATN_NEG:

$\forall x::real.\ atn\ (-\ x) = -\ atn\ x$

thm ATN_MONO_LT:

$\forall (x::real)\ y::real.\ x < y \longrightarrow atn\ x < atn\ y$

thm ATN_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ (atn\ x < atn\ y) = (x < y)$

thm ATN_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ (atn\ x \leq atn\ y) = (x \leq y)$

thm ATN_INJ:

$\forall (x::real)\ y::real.\ (atn\ x = atn\ y) = (x = y)$

thm ATN_POS_LT:

$((0::real) < atn\ (?x::real)) = ((0::real) < ?x)$

thm ATN_POS_LE:

$((0::real) \leq atn\ (?x::real)) = ((0::real) \leq ?x)$

thm ATN_LT_PI4_POS:

$\forall x<1::real.\ atn\ x < pi\ /\ real\_of\_nat\ (4::nat)$

thm ATN_LT_PI4_NEG:

$\forall x{>}-(1::real).\ -(pi\ /\ real\_of\_nat\ (4::nat)) < atn\ x$

thm ATN_LT_PI4:

$\forall x::real.\ |x| < (1::real) \longrightarrow |atn\ x| < pi\ /\ real\_of\_nat\ (4::nat)$

thm ATN_LE_PI4:

$\forall x::real.\ |x| \leq (1::real) \longrightarrow |atn\ x| \leq pi\ /\ real\_of\_nat\ (4::nat)$

thm COS_ATN_NZ:

$\forall x::real.\ cos\ (atn\ x) \neq (0::real)$

thm TAN_SEC:

$\forall x::real.\ cos\ x \neq (0::real) \longrightarrow (1::real) + (tan\ x)^2 = (inverse\_class.inverse\ (cos\ x))^2$

thm COS_ATN:

$\forall x::real.\ cos\ (atn\ x) = (1::real)\ /\ sqrt\ ((1::real) + x^2)$

thm SIN_ATN:

$\forall x::real.\ sin\ (atn\ x) = x\ /\ sqrt\ ((1::real) + x^2)$

thm ATN_ABS_LE_X:

$\forall x::real.\ |atn\ x| \leq |x|$

thm ATN_LE_X:

$\forall x{\geq}0::real.\ atn\ x \leq x$

thm TAN_ABS_GE_X:

$\forall x::real.\ |x| < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow |x| \leq |tan\ x|$

thm TAN_TOTAL:

$\forall y::real.\ \exists!x::real.\ -(pi\ /\ real\_of\_nat\ (2::nat)) < x \land x < pi\ /\ real\_of\_nat\ (2::nat) \land tan\ x = y$

thm TAN_TOTAL_POS:

$\forall y{\geq}0::real.\ \exists x{\geq}0::real.\ x < pi\ /\ real\_of\_nat\ (2::nat) \land tan\ x = y$

thm TAN_TOTAL_LEMMA:

$\forall y{>}0::real.\ \exists x{>}0::real.\ x < pi\ /\ real\_of\_nat\ (2::nat) \land y < tan\ x$

thm RE_POW_2:

$Re\ (complex\_pow\ (?z::(real,\ 2)\ cart)\ (2::nat)) = (Re\ ?z)^2 - (Im\ ?z)^2$

thm IM_POW_2:

Im (*complex_pow* (*?z*::(*real*, *2*) *cart*) (*2*::*nat*)) = *real_of_nat* (*2*::*nat*) ∗ (*Re ?z* ∗ *Im ?z*)

thm ABS_SQUARE_LT_1:

∀ *x*::*real*. (x^2 < (*1*::*real*)) = (|*x*| < (*1*::*real*))

thm ABS_SQUARE_LE_1:

∀ *x*::*real*. (x^2 ≤ (*1*::*real*)) = (|*x*| ≤ (*1*::*real*))

thm DEF_casn:

casn = (λ*_1879931*::(*real*, *2*) *cart*. *complex_mul* (*vector_neg ii*) (*clog* (*vector_add* (*complex_mul ii _1879931*) (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow _1879931* (*2*::*nat*)))))))

thm casn:

∀ *z*::(*real*, *2*) *cart*. *casn z* = *complex_mul* (*vector_neg ii*) (*clog* (*vector_add* (*complex_mul ii z*) (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*))))))

thm CASN_BODY_LEMMA:

∀ *z*::(*real*, *2*) *cart*. *vector_add* (*complex_mul ii z*) (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*)))) ≠ *Cx* (*0*::*real*)

thm CSIN_CASN:

∀ *z*::(*real*, *2*) *cart*. *csin* (*casn z*) = *z*

thm CASN_CSIN:

∀ *z*::(*real*, *2*) *cart*. |*Re z*| < *pi* / *real_of_nat* (*2*::*nat*) ∨ |*Re z*| = *pi* / *real_of_nat* (*2*::*nat*) ∧ *Im z* = (*0*::*real*) ⟶ *casn* (*csin z*) = *z*

thm CASN_UNIQUE:

∀ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *csin z* = *w* ∧ (|*Re z*| < *pi* / *real_of_nat* (*2*::*nat*) ∨ |*Re z*| = *pi* / *real_of_nat* (*2*::*nat*) ∧ *Im z* = (*0*::*real*)) ⟶ *casn w* = *z*

thm CASN_0:

casn (*Cx* (*0*::*real*)) = *Cx* (*0*::*real*)

thm CASN_1:

casn (*Cx* (*1*::*real*)) = *Cx* (*pi* / *real_of_nat* (*2*::*nat*))

thm CASN_NEG_1:

casn (*vector_neg* (*Cx* (*1*::*real*))) = *vector_neg* (*Cx* (*pi* / *real_of_nat* (*2*::*nat*)))

thm HAS_COMPLEX_DERIVATIVE_CASN:

∀ *z*::(*real*, *2*) *cart*. (*Im z* = (*0*::*real*) ⟶ |*Re z*| < (*1*::*real*)) ⟶ *has_complex_derivative casn* (*complex_inv* (*ccos* (*casn z*))) (*at z*)

thm COMPLEX_DIFFERENTIABLE_AT_CASN:

$\forall z::(real, 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow complex\_differentiable\ casn\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CASN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow complex\_differentiable\ casn\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CASN:

$\forall z::(real, 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\ casn\ (at\ z)$

thm CONTINUOUS_WITHIN_CASN:

$\forall (s::(real, 2)\ cart \Rightarrow bool)\ z::(real, 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\ casn\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CASN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \wedge Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\_on\ casn\ s$

thm HOLOMORPHIC_ON_CASN:

$\forall s::(real, 2)\ cart \Rightarrow bool.\ (\forall z::(real, 2)\ cart.\ IN\ z\ s \wedge Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow holomorphic\_on\ casn\ s$

thm DEF_cacs:

$cacs = (\lambda\_1880425::(real, 2)\ cart.\ complex\_mul\ (vector\_neg\ ii)\ (clog\ (vector\_add\ \_1880425\ (complex\_mul\ ii\ (csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ \_1880425\ (2::nat))))))))$

thm cacs:

$\forall z::(real, 2)\ cart.\ cacs\ z = complex\_mul\ (vector\_neg\ ii)\ (clog\ (vector\_add\ z\ (complex\_mul\ ii\ (csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat)))))))$

thm CACS_BODY_LEMMA:

$\forall z::(real, 2)\ cart.\ vector\_add\ z\ (complex\_mul\ ii\ (csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat))))) \neq Cx\ (0::real)$

thm CCOS_CACS:

$\forall z::(real, 2)\ cart.\ ccos\ (cacs\ z) = z$

thm CACS_CCOS:

$\forall z::(real, 2)\ cart.\ (0::real) < Re\ z \wedge Re\ z < pi \vee Re\ z = (0::real) \wedge (0::real) \leq Im\ z \vee Re\ z = pi \wedge Im\ z \leq (0::real) \longrightarrow cacs\ (ccos\ z) = z$

thm CACS_UNIQUE:

$\forall (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ ccos\ z = w \wedge ((0::real) < Re\ z \wedge Re\ z < pi \vee Re\ z = (0::real) \wedge (0::real) \leq Im\ z \vee Re\ z = pi \wedge Im\ z \leq (0::real)) \longrightarrow cacs\ w = z$

thm CACS_0:

$cacs\ (Cx\ (0::real)) = Cx\ (pi\ /\ real\_of\_nat\ (2::nat))$

thm CACS_1:

$cacs\ (Cx\ (1::real)) = Cx\ (0::real)$

thm CACS_NEG_1:

$cacs\ (vector\_neg\ (Cx\ (1::real))) = Cx\ pi$

thm HAS_COMPLEX_DERIVATIVE_CACS:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow has\_complex\_derivative\ cacs\ (vector\_neg\ (complex\_inv\ (csin\ (cacs\ z))))\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_AT_CACS:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow complex\_differentiable\ cacs\ (at\ z)$

thm COMPLEX_DIFFERENTIABLE_WITHIN_CACS:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow complex\_differentiable\ cacs\ (within\ (at\ z)\ s)$

thm CONTINUOUS_AT_CACS:

$\forall z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\ cacs\ (at\ z)$

thm CONTINUOUS_WITHIN_CACS:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ (Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\ cacs\ (within\ (at\ z)\ s)$

thm CONTINUOUS_ON_CACS:

$\forall s::(real,\ 2)\ cart \Rightarrow bool.\ (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \wedge Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow continuous\_on\ cacs\ s$

thm HOLOMORPHIC_ON_CACS:

$\forall s::(real,\ 2)\ cart \Rightarrow bool.\ (\forall z::(real,\ 2)\ cart.\ IN\ z\ s \wedge Im\ z = (0::real) \longrightarrow |Re\ z| < (1::real)) \longrightarrow holomorphic\_on\ cacs\ s$

thm CASN_RANGE_LEMMA:

$\forall z::(real,\ 2)\ cart.\ |Re\ z| < (1::real) \longrightarrow (0::real) < Re\ (vector\_add\ (complex\_mul\ ii\ z)\ (csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat)))))$

thm CACS_RANGE_LEMMA:

$\forall z$::(*real*, *2*) *cart*. |*Re z*| < (*1*::*real*) ⟶ (*0*::*real*) < *Im* (*vector_add z* (*complex_mul ii* (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*))))))

thm RE_CASN:

$\forall z$::(*real*, *2*) *cart*. *Re* (*casn z*) = *Im* (*clog* (*vector_add* (*complex_mul ii z*) (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*))))))

thm RE_CACS:

$\forall z$::(*real*, *2*) *cart*. *Re* (*cacs z*) = *Im* (*clog* (*vector_add z* (*complex_mul ii* (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*)))))))

thm CASN_BOUNDS:

$\forall z$::(*real*, *2*) *cart*. |*Re z*| < (*1*::*real*) ⟶ |*Re* (*casn z*)| < *pi* / *real_of_nat* (*2*::*nat*)

thm CACS_BOUNDS:

$\forall z$::(*real*, *2*) *cart*. |*Re z*| < (*1*::*real*) ⟶ (*0*::*real*) < *Re* (*cacs z*) ∧ *Re* (*cacs z*) < *pi*

thm CCOS_CASN_NZ:

$\forall z$::(*real*, *2*) *cart*. *complex_pow z* (*2*::*nat*) ≠ *Cx* (*1*::*real*) ⟶ *ccos* (*casn z*) ≠ *Cx* (*0*::*real*)

thm CSIN_CACS_NZ:

$\forall z$::(*real*, *2*) *cart*. *complex_pow z* (*2*::*nat*) ≠ *Cx* (*1*::*real*) ⟶ *csin* (*cacs z*) ≠ *Cx* (*0*::*real*)

thm CCOS_CSIN_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*0*::*real*) < *cos* (*Re z*) ∨ *cos* (*Re z*) = (*0*::*real*) ∧ *Im z* * *sin* (*Re z*) ≤ (*0*::*real*) ⟶ *ccos z* = *csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow* (*csin z*) (*2*::*nat*)))

thm CSIN_CCOS_CSQRT:

$\forall z$::(*real*, *2*) *cart*. (*0*::*real*) < *sin* (*Re z*) ∨ *sin* (*Re z*) = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im z* * *cos* (*Re z*) ⟶ *csin z* = *csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow* (*ccos z*) (*2*::*nat*)))

thm CASN_CACS_SQRT_POS:

$\forall z$::(*real*, *2*) *cart*. (*0*::*real*) < *Re z* ∨ *Re z* = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im z* ⟶ *casn z* = *cacs* (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*))))

thm CACS_CASN_SQRT_POS:

$\forall z$::(*real*, *2*) *cart*. (*0*::*real*) < *Re z* ∨ *Re z* = (*0*::*real*) ∧ (*0*::*real*) ≤ *Im z* ⟶ *cacs z* = *casn* (*csqrt* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_pow z* (*2*::*nat*))))

thm CSIN_CACS:

$\forall z::(real, 2)\ cart.\ (0::real) < Re\ z \vee Re\ z = (0::real) \wedge (0::real) \leq Im\ z \longrightarrow csin\ (cacs\ z) = csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat)))$

thm CCOS_CASN:

$\forall z::(real, 2)\ cart.\ (0::real) < Re\ z \vee Re\ z = (0::real) \wedge (0::real) \leq Im\ z \longrightarrow ccos\ (casn\ z) = csqrt\ (vector\_sub\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat)))$

thm DEF_asn:

$asn = (\lambda\_1880956::real.\ Re\ (casn\ (Cx\ \_1880956)))$

thm asn:

$\forall x::real.\ asn\ x = Re\ (casn\ (Cx\ x))$

thm REAL_ASN:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ z \wedge |Re\ z| \leq (1::real) \longrightarrow HOL\_Light\_Import.real\ (casn\ z)$

thm CX_ASN:

$\forall x::real.\ |x| \leq (1::real) \longrightarrow Cx\ (asn\ x) = casn\ (Cx\ x)$

thm SIN_ASN:

$\forall y::real.\ -\ (1::real) \leq y \wedge y \leq (1::real) \longrightarrow sin\ (asn\ y) = y$

thm ASN_SIN:

$\forall x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) \leq x \wedge x \leq pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow asn\ (sin\ x) = x$

thm ASN_BOUNDS_LT:

$\forall y::real.\ -\ (1::real) < y \wedge y < (1::real) \longrightarrow -\ (pi\ /\ real\_of\_nat\ (2::nat)) < asn\ y \wedge asn\ y < pi\ /\ real\_of\_nat\ (2::nat)$

thm ASN_0:

$asn\ (0::real) = (0::real)$

thm ASN_1:

$asn\ (1::real) = pi\ /\ real\_of\_nat\ (2::nat)$

thm ASN_NEG_1:

$asn\ (-\ (1::real)) = -\ (pi\ /\ real\_of\_nat\ (2::nat))$

thm ASN_BOUNDS:

$\forall y::real.\ -\ (1::real) \leq y \wedge y \leq (1::real) \longrightarrow -\ (pi\ /\ real\_of\_nat\ (2::nat)) \leq asn\ y \wedge asn\ y \leq pi\ /\ real\_of\_nat\ (2::nat)$

thm ASN_NEG:

$\forall x::real.\ -\ (1::real) \leq x \wedge x \leq (1::real) \longrightarrow asn\ (-\ x) = -\ asn\ x$

thm COS_ASN_NZ:

$\forall x{::}real.\ -\ (1{::}real) < x \wedge x < (1{::}real) \longrightarrow cos\ (asn\ x) \neq (0{::}real)$

thm ASN_MONO_LT_EQ:

$\forall (x{::}real)\ y{::}real.\ |x| \leq (1{::}real) \wedge |y| \leq (1{::}real) \longrightarrow (asn\ x < asn\ y) = (x < y)$

thm ASN_MONO_LE_EQ:

$\forall (x{::}real)\ y{::}real.\ |x| \leq (1{::}real) \wedge |y| \leq (1{::}real) \longrightarrow (asn\ x \leq asn\ y) = (x \leq y)$

thm ASN_MONO_LT:

$\forall (x{::}real)\ y{::}real.\ -\ (1{::}real) \leq x \wedge x < y \wedge y \leq (1{::}real) \longrightarrow asn\ x < asn\ y$

thm ASN_MONO_LE:

$\forall (x{::}real)\ y{::}real.\ -\ (1{::}real) \leq x \wedge x \leq y \wedge y \leq (1{::}real) \longrightarrow asn\ x \leq asn\ y$

thm COS_ASN:

$\forall x{::}real.\ -\ (1{::}real) \leq x \wedge x \leq (1{::}real) \longrightarrow cos\ (asn\ x) = sqrt\ ((1{::}real) - x^2)$

thm DEF_acs:

$acs = (\lambda\_1881041{::}real.\ Re\ (cacs\ (Cx\ \_1881041)))$

thm acs:

$\forall x{::}real.\ acs\ x = Re\ (cacs\ (Cx\ x))$

thm REAL_ACS:

$\forall z{::}(real,\ 2)\ cart.\ HOL\_Light\_Import.real\ z \wedge |Re\ z| \leq (1{::}real) \longrightarrow HOL\_Light\_Import.real\ (cacs\ z)$

thm CX_ACS:

$\forall x{::}real.\ |x| \leq (1{::}real) \longrightarrow Cx\ (acs\ x) = cacs\ (Cx\ x)$

thm COS_ACS:

$\forall y{::}real.\ -\ (1{::}real) \leq y \wedge y \leq (1{::}real) \longrightarrow cos\ (acs\ y) = y$

thm ACS_COS:

$\forall x{::}real.\ (0{::}real) \leq x \wedge x \leq pi \longrightarrow acs\ (cos\ x) = x$

thm ACS_BOUNDS_LT:

$\forall y{::}real.\ -\ (1{::}real) < y \wedge y < (1{::}real) \longrightarrow (0{::}real) < acs\ y \wedge acs\ y < pi$

thm ACS_0:

$acs\ (0{::}real) = pi\ /\ real\_of\_nat\ (2{::}nat)$

thm ACS_1:

$acs\ (1::real) = (0::real)$

thm ACS_NEG_1:

$acs\ (-\ (1::real)) = pi$

thm ACS_BOUNDS:

$\forall y::real.\ -\ (1::real) \leq y \wedge y \leq (1::real) \longrightarrow (0::real) \leq acs\ y \wedge acs\ y \leq pi$

thm ACS_NEG:

$\forall x::real.\ -\ (1::real) \leq x \wedge x \leq (1::real) \longrightarrow acs\ (-\ x) = pi - acs\ x$

thm SIN_ACS_NZ:

$\forall x::real.\ -\ (1::real) < x \wedge x < (1::real) \longrightarrow sin\ (acs\ x) \neq (0::real)$

thm ACS_MONO_LT_EQ:

$\forall (x::real)\ y::real.\ |x| \leq (1::real) \wedge |y| \leq (1::real) \longrightarrow (acs\ x < acs\ y) = (y < x)$

thm ACS_MONO_LE_EQ:

$\forall (x::real)\ y::real.\ |x| \leq (1::real) \wedge |y| \leq (1::real) \longrightarrow (acs\ x \leq acs\ y) = (y \leq x)$

thm ACS_MONO_LT:

$\forall (x::real)\ y::real.\ -\ (1::real) \leq x \wedge x < y \wedge y \leq (1::real) \longrightarrow acs\ y < acs\ x$

thm ACS_MONO_LE:

$\forall (x::real)\ y::real.\ -\ (1::real) \leq x \wedge x \leq y \wedge y \leq (1::real) \longrightarrow acs\ y \leq acs\ x$

thm SIN_ACS:

$\forall x::real.\ -\ (1::real) \leq x \wedge x \leq (1::real) \longrightarrow sin\ (acs\ x) = sqrt\ ((1::real) - x^2)$

thm ACS_INJ:

$\forall (x::real)\ y::real.\ |x| \leq (1::real) \wedge |y| \leq (1::real) \longrightarrow (acs\ x = acs\ y) = (x = y)$

thm Trigonometry.SCEZKRH2:

$\forall x::real.\ cos\ (pi\ /\ real\_of\_nat\ (2::nat) - x) = sin\ x$

thm ACS_ATN:

$\forall x::real.\ -\ (1::real) < x \wedge x < (1::real) \longrightarrow acs\ x = pi\ /\ real\_of\_nat\ (2::nat) - atn\ (x\ /\ sqrt\ ((1::real) - x^2))$

thm ASN_PLUS_ACS:

$\forall x::real.\ - (1::real) \leq x \wedge x \leq (1::real) \longrightarrow asn\ x + acs\ x = pi\ /\ real\_of\_nat\ (2::nat)$

thm ASN_ACS:

$\forall x::real.\ - (1::real) \leq x \wedge x \leq (1::real) \longrightarrow asn\ x = pi\ /\ real\_of\_nat\ (2::nat) - acs\ x$

thm ACS_ASN:

$\forall x::real.\ - (1::real) \leq x \wedge x \leq (1::real) \longrightarrow acs\ x = pi\ /\ real\_of\_nat\ (2::nat) - asn\ x$

thm ASN_ATN:

$\forall x::real.\ - (1::real) < x \wedge x < (1::real) \longrightarrow asn\ x = atn\ (x\ /\ sqrt\ ((1::real) - x^2))$

thm ASN_ACS_SQRT_POS:

$\forall x::real.\ (0::real) \leq x \wedge x \leq (1::real) \longrightarrow asn\ x = acs\ (sqrt\ ((1::real) - x^2))$

thm ASN_ACS_SQRT_NEG:

$\forall x::real.\ - (1::real) \leq x \wedge x \leq (0::real) \longrightarrow asn\ x = - acs\ (sqrt\ ((1::real) - x^2))$

thm ACS_ASN_SQRT_POS:

$\forall x::real.\ (0::real) \leq x \wedge x \leq (1::real) \longrightarrow acs\ x = asn\ (sqrt\ ((1::real) - x^2))$

thm ACS_ASN_SQRT_NEG:

$\forall x::real.\ - (1::real) \leq x \wedge x \leq (0::real) \longrightarrow acs\ x = pi - asn\ (sqrt\ ((1::real) - x^2))$

thm CONTINUOUS_ON_CASN_REAL:

continuous_on casn (GSPEC (λGEN%PVAR%2352::(real, 2) cart. ∃ w::(real, 2) cart. SETSPEC GEN%PVAR%2352 (HOL_Light_Import.real w ∧ |Re w| ≤ (1::real)) w))

thm CONTINUOUS_WITHIN_CASN_REAL:

∀ z::(real, 2) cart. continuous casn (within (at z) (GSPEC (λGEN%PVAR%2357::(real, 2) cart. ∃ w::(real, 2) cart. SETSPEC GEN%PVAR%2357 (HOL_Light_Import.real w ∧ |Re w| ≤ (1::real)) w)))

thm CONTINUOUS_ON_CACS_REAL:

continuous_on cacs (GSPEC (λGEN%PVAR%2362::(real, 2) cart. ∃ w::(real, 2) cart. SETSPEC GEN%PVAR%2362 (HOL_Light_Import.real w ∧ |Re w| ≤ (1::real)) w))

thm CONTINUOUS_WITHIN_CACS_REAL:

$\forall z$::(*real*, *2*) *cart*. *continuous cacs* (*within* (*at z*) (*GSPEC* (λ*GEN%PVAR%2367*::(*real*, *2*) *cart*. $\exists$ *w*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2367* (*HOL_Light_Import.real* *w* $\wedge$ |*Re w*| $\leq$ (*1*::*real*)) *w*)))

thm LIM_LOG_OVER_POWER:

$\forall s$::(*real*, *2*) *cart*. (*0*::*real*) < *Re s* $\longrightarrow$ *−−>* (λ*n*::*nat*. *complex_div* (*clog* (*Cx* (*real_of_nat n*))) (*cpow* (*Cx* (*real_of_nat n*)) *s*)) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_LOG_OVER_N:

−−> (λ*n*::*nat*. *complex_div* (*clog* (*Cx* (*real_of_nat n*))) (*Cx* (*real_of_nat n*))) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_1_OVER_POWER:

$\forall s$::(*real*, *2*) *cart*. (*0*::*real*) < *Re s* $\longrightarrow$ *−−>* (λ*n*::*nat*. *complex_div* (*Cx* (*1*::*real*)) (*cpow* (*Cx* (*real_of_nat n*)) *s*)) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_1_OVER_N:

−−> (λ*n*::*nat*. *complex_div* (*Cx* (*1*::*real*)) (*Cx* (*real_of_nat n*))) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_INV_N:

−−> (λ*n*::*nat*. *complex_inv* (*Cx* (*real_of_nat n*))) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_1_OVER_LOG:

−−> (λ*n*::*nat*. *complex_div* (*Cx* (*1*::*real*)) (*clog* (*Cx* (*real_of_nat n*)))) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_N_TIMES_POWN:

$\forall z$::(*real*, *2*) *cart*. *vector_norm z* < (*1*::*real*) $\longrightarrow$ *−−>* (λ*n*::*nat*. *complex_mul* (*Cx* (*real_of_nat n*)) (*complex_pow z n*)) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_N_OVER_POWN:

$\forall z$::(*real*, *2*) *cart*. (*1*::*real*) < *vector_norm z* $\longrightarrow$ *−−>* (λ*n*::*nat*. *complex_div* (*Cx* (*real_of_nat n*)) (*complex_pow z n*)) (*Cx* (*0*::*real*)) *sequentially*

thm LIM_POWN:

$\forall z$::(*real*, *2*) *cart*. *vector_norm z* < (*1*::*real*) $\longrightarrow$ *−−>* (*complex_pow z*) (*Cx* (*0*::*real*)) *sequentially*

thm COMPLEX_ROOT_POLYFUN:

$\forall$ (*n*::*nat*) (*z*::(*real*, *2*) *cart*) *a*::(*real*, *2*) *cart*. (*1*::*nat*) $\leq$ *n* $\longrightarrow$ (*complex_pow z n* = *a*) = (*vsum* (*dotdot* (*0*::*nat*) *n*) (λ*i*::*nat*. *complex_mul* (*if i* = (*0*::*nat*) *then vector_neg a else if i* = *n then Cx* (*1*::*real*) *else Cx* (*0*::*real*)) (*complex_pow z i*)) = *Cx* (*0*::*real*))

thm COMPLEX_ROOT_UNITY:

$\forall (n::nat)\ j::nat.\ n \neq (0::nat) \longrightarrow complex\_pow\ (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n))))))\ n = Cx\ (1::real)$

thm COMPLEX_ROOT_UNITY_EQ:

$\forall (n::nat)\ (j::nat)\ k::nat.\ n \neq (0::nat) \longrightarrow (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n))))) = cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ k\ /\ real\_of\_nat\ n)))))) = HOL\_Light\_Import.==\ j\ k\ (num\_mod\ n)$

thm COMPLEX_ROOT_UNITY_EQ_1:

$\forall (n::nat)\ j::nat.\ n \neq (0::nat) \longrightarrow (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n))))) = Cx\ (1::real)) = num\_divides\ n\ j$

thm FINITE_CARD_COMPLEX_ROOTS_UNITY:

$\forall n \geq 1::nat.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2368::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2368\ (complex\_pow\ z\ n = Cx\ (1::real))\ z)) \wedge CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2369::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2369\ (complex\_pow\ z\ n = Cx\ (1::real))\ z)) \leq n$

thm FINITE_COMPLEX_ROOTS_UNITY:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2370::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2370\ (complex\_pow\ z\ n = Cx\ (1::real))\ z))$

thm FINITE_CARD_COMPLEX_ROOTS_UNITY_EXPLICIT:

$\forall n \geq 1::nat.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2371::(real,\ 2)\ cart.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%2371\ (j < n)\ (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n)))))))) \wedge CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2372::(real,\ 2)\ cart.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%2372\ (j < n)\ (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n)))))))) = n$

thm COMPLEX_ROOTS_UNITY:

$\forall n \geq 1::nat.\ GSPEC\ (\lambda GEN\%PVAR\%2373::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2373\ (complex\_pow\ z\ n = Cx\ (1::real))\ z) = GSPEC\ (\lambda GEN\%PVAR\%2374::(real,\ 2)\ cart.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%2374\ (j < n)\ (cexp\ (complex\_mul\ (Cx\ (real\_of\_nat\ (2::nat)))\ (complex\_mul\ (Cx\ pi)\ (complex\_mul\ ii\ (Cx\ (real\_of\_nat\ j\ /\ real\_of\_nat\ n)))))))$

thm CARD_COMPLEX_ROOTS_UNITY:

$\forall n \geq 1::nat.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2375::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2375\ (complex\_pow\ z\ n = Cx\ (1::real))\ z)) = n$

thm HAS_SIZE_COMPLEX_ROOTS_UNITY:

$\forall\, n \geq 1$::*nat*. *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%2376*::(*real*, *2*) *cart*. $\exists\, z$::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2376* (*complex_pow z n* = *Cx* (*1*::*real*)) *z*)) *n*

thm COMPLEX_NOT_ROOT_UNITY:

$\forall\, n \geq 1$::*nat*. $\exists\, u$::(*real*, *2*) *cart*. *vector_norm u* = (*1*::*real*) $\wedge$ *complex_pow u n* $\neq$ *Cx* (*1*::*real*)

thm ARG_CLOG:

$\forall\, z$::(*real*, *2*) *cart*. (*0*::*real*) < *Arg z* $\longrightarrow$ *Arg z* = *Im* (*clog* (*vector_neg z*)) + *pi*

thm CONTINUOUS_AT_ARG:

$\forall\, z$::(*real*, *2*) *cart*. $\neg$ (*HOL_Light_Import.real z* $\wedge$ (*0*::*real*) $\leq$ *Re z*) $\longrightarrow$ *continuous* (*Cx* $\circ$ *Arg*) (*at z*)

thm CONTINUOUS_WITHIN_UPPERHALF_ARG:

$\forall\, z$::(*real*, *2*) *cart*. *z* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *continuous* (*Cx* $\circ$ *Arg*) (*within* (*at z*) (*GSPEC* (λ*GEN%PVAR%2380*::(*real*, *2*) *cart*. $\exists\, z$::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2380* ((*0*::*real*) $\leq$ *Im z*) *z*)))

thm OPEN_ARG_LTT:

$\forall\,$(*s*::*real*) *t*::*real*. (*0*::*real*) $\leq$ *s* $\wedge$ *t* $\leq$ *real_of_nat* (*2*::*nat*) $*$ *pi* $\longrightarrow$ *HOL_Light_Import.open* (*GSPEC* (λ*GEN%PVAR%2386*::(*real*, *2*) *cart*. $\exists\, z$::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2386* (*s* < *Arg z* $\wedge$ *Arg z* < *t*) *z*))

thm OPEN_ARG_GT:

$\forall\, t$::*real*. *HOL_Light_Import.open* (*GSPEC* (λ*GEN%PVAR%2388*::(*real*, *2*) *cart*. $\exists\, z$::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2388* (*t* < *Arg z*) *z*))

thm CLOSED_ARG_LE:

$\forall\, t$::*real*. *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%2389*::(*real*, *2*) *cart*. $\exists\, z$::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2389* (*Arg z* $\leq$ *t*) *z*))

thm ARG_ATAN_UPPERHALF:

$\forall\, z$::(*real*, *2*) *cart*. (*0*::*real*) < *Im z* $\longrightarrow$ *Arg z* = *pi* / *real_of_nat* (*2*::*nat*) $-$ *atn* (*Re z* / *Im z*)

thm DEF_root:

root = (λ(*_1881279*::*nat*) *_1881280*::*real*. *SOME u*::*real*. ((*0*::*real*) $\leq$ *_1881280* $\longrightarrow$ (*0*::*real*) $\leq$ *u*) $\wedge$ $u^{-1881279}$ = *_1881280*)

thm root:

$\forall\,$(*n*::*nat*) *x*::*real*. *root n x* = (*SOME u*::*real*. ((*0*::*real*) $\leq$ *x* $\longrightarrow$ (*0*::*real*) $\leq$ *u*) $\wedge$ u^n = *x*)

thm ROOT_0:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow root\ n\ (0::real) = (0::real)$

thm ROOT_1:

$\forall n::nat.\ n \neq (0::nat) \longrightarrow root\ n\ (1::real) = (1::real)$

thm ROOT_2:

$\forall x::real.\ root\ (2::nat)\ x = sqrt\ x$

thm ROOT_WORKS:

$\forall (n::nat)\ x::real.\ ODD\ n \lor n \neq (0::nat) \land (0::real) \leq x \longrightarrow ((0::real) \leq x \longrightarrow (0::real) \leq root\ n\ x) \land (root\ n\ x)^n = x$

thm REAL_POW_ROOT:

$\forall (n::nat)\ x::real.\ ODD\ n \lor n \neq (0::nat) \land (0::real) \leq x \longrightarrow (root\ n\ x)^n = x$

thm ROOT_POS_LE:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \land (0::real) \leq x \longrightarrow (0::real) \leq root\ n\ x$

thm ROOT_POS_LT:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \land (0::real) < x \longrightarrow (0::real) < root\ n\ x$

thm REAL_ROOT_POW:

$\forall (n::nat)\ x::real.\ ODD\ n \lor n \neq (0::nat) \land (0::real) \leq x \longrightarrow root\ n\ x^n = x$

thm ROOT_UNIQUE:

$\forall (n::nat)\ (x::real)\ y::real.\ y^n = x \land (ODD\ n \lor n \neq (0::nat) \land (0::real) \leq y) \longrightarrow root\ n\ x = y$

thm REAL_ROOT_MUL:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \land (0::real) \leq x \land (0::real) \leq y \longrightarrow root\ n\ (x * y) = root\ n\ x * root\ n\ y$

thm REAL_ROOT_INV:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \land (0::real) \leq x \longrightarrow root\ n\ (inverse\_class.inverse\ x) = inverse\_class.inverse\ (root\ n\ x)$

thm REAL_ROOT_DIV:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \land (0::real) \leq x \land (0::real) \leq y \longrightarrow root\ n\ (x\ /\ y) = root\ n\ x\ /\ root\ n\ y$

thm ROOT_MONO_LT:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \land (0::real) \leq x \land x < y \longrightarrow root\ n\ x < root\ n\ y$

thm ROOT_MONO_LE:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge x \leq y \longrightarrow root\ n\ x \leq root\ n\ y$

thm ROOT_MONO_LT_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (root\ n\ x < root\ n\ y) = (x < y)$

thm ROOT_MONO_LE_EQ:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (root\ n\ x \leq root\ n\ y) = (x \leq y)$

thm ROOT_INJ:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (root\ n\ x = root\ n\ y) = (x = y)$

thm REAL_ROOT_LE:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (root\ n\ x \leq y) = (x \leq y^n)$

thm REAL_LE_ROOT:

$\forall (n::nat)\ (x::real)\ y::real.\ n \neq (0::nat) \wedge (0::real) \leq x \wedge (0::real) \leq y \longrightarrow (x \leq root\ n\ y) = (x^n \leq y)$

thm LOG_ROOT:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \wedge (0::real) < x \longrightarrow log\ (root\ n\ x) = log\ x\ /\ real\_of\_nat\ n$

thm ROOT_EXP_LOG:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \wedge (0::real) < x \longrightarrow root\ n\ x = exp\ (log\ x\ /\ real\_of\_nat\ n)$

thm DEF_rpow:

$rpow = (\lambda(\_1881556::real)\ \_1881557::real.\ if\ (0::real) < \_1881556\ then\ exp\ (\_1881557 * log\ \_1881556)\ else\ if\ \_1881556 = (0::real)\ then\ if\ \_1881557 = (0::real)\ then\ 1::real\ else\ (0::real)\ else\ if\ \exists (m::nat)\ n::nat.\ ODD\ m \wedge ODD\ n \wedge |\_1881557| = real\_of\_nat\ m\ /\ real\_of\_nat\ n\ then\ -\ exp\ (\_1881557 * log\ (-\ \_1881556))\ else\ exp\ (\_1881557 * log\ (-\ \_1881556)))$

thm rpow:

$\forall (y::real)\ x::real.\ rpow\ x\ y = (if\ (0::real) < x\ then\ exp\ (y * log\ x)\ else\ if\ x = (0::real)\ then\ if\ y = (0::real)\ then\ 1::real\ else\ (0::real)\ else\ if\ \exists (m::nat)\ n::nat.\ ODD\ m \wedge ODD\ n \wedge |y| = real\_of\_nat\ m\ /\ real\_of\_nat\ n\ then\ -\ exp\ (y * log\ (-\ x))\ else\ exp\ (y * log\ (-\ x)))$

thm RPOW_POW:

$\forall (x::real)\ n::nat.\ rpow\ x\ (real\_of\_nat\ n) = x^n$

thm RPOW_NEG:

$\forall (x::real)\ y::real.\ rpow\ x\ (-\ y) = inverse\_class.inverse\ (rpow\ x\ y)$

thm RPOW_ZERO:

$\forall y::real.\ rpow\ (0::real)\ y = (if\ y = (0::real)\ then\ 1::real\ else\ (0::real))$

thm RPOW_POS_LT:

$\forall (x::real)\ y::real.\ (0::real) < x \longrightarrow (0::real) < rpow\ x\ y$

thm RPOW_POS_LE:

$\forall (x::real)\ y::real.\ (0::real) \leq x \longrightarrow (0::real) \leq rpow\ x\ y$

thm RPOW_LT2:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge x < y \wedge (0::real) < z \longrightarrow rpow\ x\ z < rpow\ y\ z$

thm RPOW_LE2:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge x \leq y \wedge (0::real) \leq z \longrightarrow rpow\ x\ z \leq rpow\ y\ z$

thm REAL_ABS_RPOW:

$\forall (x::real)\ y::real.\ |rpow\ x\ y| = rpow\ |x|\ y$

thm RPOW_ONE:

$\forall z::real.\ rpow\ (1::real)\ z = (1::real)$

thm RPOW_RPOW:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \longrightarrow rpow\ (rpow\ x\ y)\ z = rpow\ x\ (y * z)$

thm RPOW_LNEG:

$\forall (x::real)\ y::real.\ rpow\ (-\ x)\ y = (if\ \exists (m::nat)\ n::nat.\ ODD\ m \wedge ODD\ n \wedge |y| = real\_of\_nat\ m\ /\ real\_of\_nat\ n\ then\ -\ rpow\ x\ y\ else\ rpow\ x\ y)$

thm RPOW_EQ_0:

$\forall (x::real)\ y::real.\ (rpow\ x\ y = (0::real)) = (x = (0::real) \wedge y \neq (0::real))$

thm RPOW_MUL:

$\forall (x::real)\ (y::real)\ z::real.\ rpow\ (x * y)\ z = rpow\ x\ z * rpow\ y\ z$

thm RPOW_INV:

$\forall (x::real)\ y::real.\ rpow\ (inverse\_class.inverse\ x)\ y = inverse\_class.inverse\ (rpow\ x\ y)$

thm REAL_INV_RPOW:

$\forall (x::real)\ y::real.\ inverse\_class.inverse\ (rpow\ x\ y) = rpow\ (inverse\_class.inverse\ x)\ y$

thm RPOW_ADD:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \longrightarrow rpow\ x\ (y + z) = rpow\ x\ y * rpow\ x\ z$

thm RPOW_ADD_ALT:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge (x = (0::real) \wedge y + z = (0::real) \longrightarrow y = (0::real) \vee z = (0::real)) \longrightarrow rpow\ x\ (y + z) = rpow\ x\ y * rpow\ x\ z$

thm RPOW_SQRT:

$\forall x\geq 0::real.\ rpow\ x\ ((1::real)\ /\ real\_of\_nat\ (2::nat)) = sqrt\ x$

thm RPOW_MONO:

$\forall (a::real)\ (b::real)\ x::real.\ (1::real) \leq x \wedge a \leq b \longrightarrow rpow\ x\ a \leq rpow\ x\ b$

thm RPOW_MONO_INV:

$\forall (a::real)\ (b::real)\ x::real.\ (0::real) < x \wedge x \leq (1::real) \wedge b \leq a \longrightarrow rpow\ x\ a \leq rpow\ x\ b$

thm RPOW_1_LE:

$\forall (a::real)\ x::real.\ (0::real) \leq x \wedge x \leq (1::real) \wedge (0::real) \leq a \longrightarrow rpow\ x\ a \leq (1::real)$

thm Local_lemmas1.INV_VNI:

$inverse\ (?x::real) = (1::real)\ /\ ?x$

thm REAL_ROOT_RPOW:

$\forall (n::nat)\ x::real.\ n \neq (0::nat) \wedge ((0::real) \leq x \vee ODD\ n) \longrightarrow root\ n\ x = rpow\ x\ (inverse\_class.inverse\ (real\_of\_nat\ n))$

thm CONTINUOUS_LOGARITHM_IMP_INESSENTIAL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ g\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = cexp\ (g\ x))) \longrightarrow (\exists a::(real,\ 2)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ True)\ (s,\ DIFF\ HOL\_Light\_Import.UNIV\ (INSERT\ (Cx\ (0::real))\ EMPTY))\ f\ (\lambda t::(real,\ ?'a::type)\ cart.\ a))$

thm INESSENTIAL_IMP_CONTINUOUS_LOGARITHM:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge (\exists a::(real,\ 2)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ True)\ (s,\ DIFF\ HOL\_Light\_Import.UNIV\ (INSERT\ (Cx\ (0::real))\ EMPTY))\ f\ (\lambda t::(real,\ ?'a::type)\ cart.\ a)) \longrightarrow (\exists g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ g\ s \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\ x = cexp\ (g\ x)))$

thm INESSENTIAL_EQ_CONTINUOUS_LOGARITHM:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ compact\ s \longrightarrow (\exists\, a::(real,\ 2)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ True)\ (s,\ DIFF\ HOL\_Light\_Import.UNIV\ (INSERT\ (Cx\ (0::real))\ EMPTY))\ f\ (\lambda t::(real,\ ?'a::type)\ cart.\ a)) = (\exists\, g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ g\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\, x = cexp\ (g\ x)))$

thm CONTINUOUS_LOGARITHM_ON_CONTRACTIBLE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge compact\ s \wedge contractible\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\, x \neq Cx\ (0::real)) \longrightarrow (\exists\, g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ g\ s \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow f\, x = cexp\ (g\ x)))$

thm CONTINUOUS_LOGARITHM_ON_CBALL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ r::real.\ continuous\_on\ f\ (cball\ (a,\ r)) \wedge (\forall\, z::(real,\ ?'a::type)\ cart.\ IN\ z\ (cball\ (a,\ r)) \longrightarrow f\ z \neq Cx\ (0::real)) \longrightarrow (\exists\, h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ h\ (cball\ (a,\ r)) \wedge (\forall\, z::(real,\ ?'a::type)\ cart.\ IN\ z\ (cball\ (a,\ r)) \longrightarrow f\ z = cexp\ (h\ z)))$

thm Hypermap.LE_PLUS:

$\forall\, n::nat.\ n \leq Suc\ n$

thm CONTINUOUS_LOGARITHM_ON_BALL:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ (a::(real,\ ?'a::type)\ cart)\ r::real.\ continuous\_on\, f\ (ball\ (a,\ r)) \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\, x\ (ball\ (a,\ r)) \longrightarrow f\, x \neq Cx\ (0::real)) \longrightarrow (\exists\, h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ h\ (ball\ (a,\ r)) \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ (ball\ (a,\ r)) \longrightarrow f\ x = cexp\ (h\ x)))$

thm INESSENTIAL_EQ_EXTENSIBLE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \longrightarrow (\exists\, a::(real,\ 2)\ cart.\ homotopic\_with\ (\lambda h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ True)\ (s,\ DIFF\ HOL\_Light\_Import.UNIV\ (INSERT\ (Cx\ (0::real))\ EMPTY))\, f\ (\lambda t::(real,\ ?'a::type)\ cart.\ a)) = (\exists\, g::(real,\ ?'a::type)\ cart \Rightarrow (real,\ 2)\ cart.\ continuous\_on\ g\ HOL\_Light\_Import.UNIV \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow g\ x = f\, x) \wedge (\forall\, x::(real,\ ?'a::type)\ cart.\ g\ x \neq Cx\ (0::real)))$

thm DEF_real_open:

$real\_open = (\lambda\_1884538::real \Rightarrow bool.\ \forall\, x::real.\ IN\, x\ \_1884538 \longrightarrow (\exists\, e{>}0::real.\ \forall\, x'::real.\ |x' - x| < e \longrightarrow IN\ x'\ \_1884538))$

thm real_open:

$\forall\, s{::}real \Rightarrow bool.\ real\_open\ s = (\forall\, x{::}real.\ IN\ x\ s \longrightarrow (\exists\, e{>}0{::}real.\ \forall\, x'{::}real.\ |x' - x| < e \longrightarrow IN\ x'\ s))$

thm DEF_real_closed:

$real\_closed = (\lambda\_1884543{::}real \Rightarrow bool.\ real\_open\ (DIFF\ HOL\_Light\_Import.UNIV\ \_1884543))$

thm real_closed:

$\forall\, s{::}real \Rightarrow bool.\ real\_closed\ s = real\_open\ (DIFF\ HOL\_Light\_Import.UNIV\ s)$

thm euclideanreal:

$euclideanreal = topology\ real\_open$

thm REAL_OPEN_EMPTY:

$real\_open\ EMPTY$

thm REAL_OPEN_UNIV:

$real\_open\ HOL\_Light\_Import.UNIV$

thm REAL_OPEN_INTER:

$\forall\, (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_open\ s \wedge real\_open\ t \longrightarrow real\_open\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_OPEN_UNIONS:

$(\forall\, s{::}real \Rightarrow bool.\ IN\ s\ (?f{::}(real \Rightarrow bool) \Rightarrow bool) \longrightarrow real\_open\ s) \longrightarrow real\_open\ (UNIONS\ ?f)$

thm REAL_OPEN_IN:

$\forall\, s{::}real \Rightarrow bool.\ real\_open\ s = open\_in\ euclideanreal\ s$

thm TOPSPACE_EUCLIDEANREAL:

$topspace\ euclideanreal = HOL\_Light\_Import.UNIV$

thm TOPSPACE_EUCLIDEANREAL_SUBTOPOLOGY:

$\forall\, s{::}real \Rightarrow bool.\ topspace\ (subtopology\ euclideanreal\ s) = s$

thm REAL_CLOSED_IN:

$\forall\, s{::}real \Rightarrow bool.\ real\_closed\ s = closed\_in\ euclideanreal\ s$

thm REAL_OPEN_UNION:

$\forall\, (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_open\ s \wedge real\_open\ t \longrightarrow real\_open\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_OPEN_SUBREAL_OPEN:

$\forall\, s{::}real \Rightarrow bool.\ real\_open\ s = (\forall\, x{::}real.\ IN\ x\ s \longrightarrow (\exists\, t{::}real \Rightarrow bool.\ real\_open\ t \wedge IN\ x\ t \wedge SUBSET\ t\ s))$

thm REAL_CLOSED_EMPTY:

real_closed EMPTY

thm REAL_CLOSED_UNIV:

real_closed HOL_Light_Import.UNIV

thm REAL_CLOSED_UNION:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *t*::*real* $\Rightarrow$ *bool*. *real_closed s* $\wedge$ *real_closed t* $\longrightarrow$ *real_closed* (*HOL_Light_Import.UNION s t*)

thm REAL_CLOSED_INTER:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *t*::*real* $\Rightarrow$ *bool*. *real_closed s* $\wedge$ *real_closed t* $\longrightarrow$ *real_closed* (*HOL_Light_Import.INTER s t*)

thm REAL_CLOSED_INTERS:

$\forall$ *f*::(*real* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. ($\forall$ *s*::*real* $\Rightarrow$ *bool*. *IN s f* $\longrightarrow$ *real_closed s*) $\longrightarrow$ *real_closed* (*INTERS f*)

thm REAL_OPEN_REAL_CLOSED:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_open s* = *real_closed* (*DIFF HOL_Light_Import.UNIV s*)

thm REAL_OPEN_DIFF:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *t*::*real* $\Rightarrow$ *bool*. *real_open s* $\wedge$ *real_closed t* $\longrightarrow$ *real_open* (*DIFF s t*)

thm REAL_CLOSED_DIFF:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *t*::*real* $\Rightarrow$ *bool*. *real_closed s* $\wedge$ *real_open t* $\longrightarrow$ *real_closed* (*DIFF s t*)

thm REAL_OPEN_INTERS:

$\forall$ *s*::(*real* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FINITE s* $\wedge$ ($\forall$ *t*::*real* $\Rightarrow$ *bool*. *IN t s* $\longrightarrow$ *real_open t*) $\longrightarrow$ *real_open* (*INTERS s*)

thm REAL_CLOSED_UNIONS:

$\forall$ *s*::(*real* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FINITE s* $\wedge$ ($\forall$ *t*::*real* $\Rightarrow$ *bool*. *IN t s* $\longrightarrow$ *real_closed t*) $\longrightarrow$ *real_closed* (*UNIONS s*)

thm REAL_OPEN:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_open s* = *HOL_Light_Import.open* (*IMAGE lift s*)

thm REAL_CLOSED:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_closed s* = *HOL_Light_Import.closed* (*IMAGE lift s*)

thm REAL_CLOSED_HALFSPACE_LE:

$\forall$ *a*::*real*. *real_closed* (*GSPEC* (λ*GEN%PVAR%2393*::*real*. $\exists$ *x*::*real*. *SETSPEC GEN%PVAR%2393* ($x \leq a$) *x*))

thm REAL_CLOSED_HALFSPACE_GE:

$\forall$ *a::real. real_closed (GSPEC (λGEN%PVAR%2395::real. ∃ x::real. SETSPEC GEN%PVAR%2395 (a ≤ x) x))*

thm REAL_OPEN_HALFSPACE_LT:

$\forall$ *a::real. real_open (GSPEC (λGEN%PVAR%2397::real. ∃ x::real. SETSPEC GEN%PVAR%2397 (x < a) x))*

thm REAL_OPEN_HALFSPACE_GT:

$\forall$ *a::real. real_open (GSPEC (λGEN%PVAR%2399::real. ∃ x::real. SETSPEC GEN%PVAR%2399 (a < x) x))*

thm DEF_real_bounded:

real_bounded = (λ_1884808::real ⇒ bool. ∃ B::real. ∀ x::real. IN x _1884808 ⟶ |x| ≤ B)

thm real_bounded:

∀ s::real ⇒ bool. real_bounded s = (∃ B::real. ∀ x::real. IN x s ⟶ |x| ≤ B)

thm REAL_BOUNDED:

real_bounded (?s::real ⇒ bool) = bounded (IMAGE lift ?s)

thm REAL_BOUNDED_POS:

∀ s::real ⇒ bool. real_bounded s = (∃ B>0::real. ∀ x::real. IN x s ⟶ |x| ≤ B)

thm REAL_BOUNDED_POS_LT:

∀ s::real ⇒ bool. real_bounded s = (∃ b>0::real. ∀ x::real. IN x s ⟶ |x| < b)

thm REAL_BOUNDED_SUBSET:

∀ (s::real ⇒ bool) t::real ⇒ bool. real_bounded t ∧ SUBSET s t ⟶ real_bounded s

thm REAL_BOUNDED_UNION:

∀ (s::real ⇒ bool) t::real ⇒ bool. real_bounded (HOL_Light_Import.UNION s t) = (real_bounded s ∧ real_bounded t)

thm DEF_real_compact:

real_compact = (λ_1884858::real ⇒ bool. compact (IMAGE lift _1884858))

thm real_compact:

∀ s::real ⇒ bool. real_compact s = compact (IMAGE lift s)

thm REAL_COMPACT_IMP_BOUNDED:

∀ s::real ⇒ bool. real_compact s ⟶ real_bounded s

thm REAL_COMPACT_IMP_CLOSED:

$\forall s::real \Rightarrow bool.\ real\_compact\ s \longrightarrow real\_closed\ s$

thm REAL_COMPACT_EQ_BOUNDED_CLOSED:

$\forall s::real \Rightarrow bool.\ real\_compact\ s = (real\_bounded\ s \wedge real\_closed\ s)$

thm REAL_COMPACT_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_compact\ s \wedge real\_compact\ t \longrightarrow real\_compact\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_COMPACT_ATTAINS_INF:

$\forall s::real \Rightarrow bool.\ real\_compact\ s \wedge s \neq EMPTY \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow x \leq y))$

thm REAL_COMPACT_ATTAINS_SUP:

$\forall s::real \Rightarrow bool.\ real\_compact\ s \wedge s \neq EMPTY \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow y \leq x))$

thm DEF_−−−>:

$--->\ = (\lambda(\_1884863::?'a::type \Rightarrow real)\ (\_1884864::real)\ \_1884865::?'a::type\ net.\ \forall e>0::real.\ eventually\ (\lambda x::?'a::type.\ |\_1884863\ x - \_1884864| < e)\ \_1884865)$

thm tendsto_real:

$\forall (f::?'a::type \Rightarrow real)\ (l::real)\ net::?'a::type\ net.\ --->\ f\ l\ net = (\forall e>0::real.\ eventually\ (\lambda x::?'a::type.\ |f\ x - l| < e)\ net)$

thm DEF_reallim:

$reallim = (\lambda(\_1884884::?'a::type\ net)\ \_1884885::?'a::type \Rightarrow real.\ SOME\ l::real.\ --->\ \_1884885\ l\ \_1884884)$

thm reallim:

$\forall (f::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ reallim\ net\ f = (SOME\ l::real.\ --->\ f\ l\ net)$

thm TENDSTO_REAL:

$--->\ (?s::?'a::type \Rightarrow real)\ (?l::real) = -->\ (lift \circ ?s)\ (lift\ ?l)$

thm REAL_TENDSTO:

$-->\ (?s::?'a::type \Rightarrow (real, unit)\ cart)\ (?l::(real, unit)\ cart) = --->\ (HOL\_Light\_Import.drop \circ ?s)\ (HOL\_Light\_Import.drop\ ?l)$

thm REALLIM_COMPLEX:

$--->\ (?s::?'a::type \Rightarrow real)\ (?l::real) = -->\ (Cx \circ ?s)\ (Cx\ ?l)$

thm REALLIM_UNIQUE:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (l::real)\ l'::real.\ \neg\ trivial\_limit\ net \wedge --->\ f\ l\ net \wedge --->\ f\ l'\ net \longrightarrow l = l'$

thm REALLIM_CONST:

$\forall (net::?'a::type\ net)\ a::real.\ ---> (\lambda x::?'a::type.\ a)\ a\ net$

thm REALLIM_LMUL:

$\forall (f::?'a::type \Rightarrow real)\ (l::real)\ c::real.\ ---> f\ l\ (?net::?'a::type\ net) \longrightarrow ---> (\lambda x::?'a::type.\ c * f\ x)\ (c * l)\ ?net$

thm REALLIM_RMUL:

$\forall (f::?'a::type \Rightarrow real)\ (l::real)\ c::real.\ ---> f\ l\ (?net::?'a::type\ net) \longrightarrow ---> (\lambda x::?'a::type.\ f\ x * c)\ (l * c)\ ?net$

thm REALLIM_LMUL_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (l::real)\ c::real.\ c \neq (0::real) \longrightarrow ---> (\lambda x::?'a::type.\ c * f\ x)\ (c * l)\ net = ---> f\ l\ net$

thm REALLIM_RMUL_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (l::real)\ c::real.\ c \neq (0::real) \longrightarrow ---> (\lambda x::?'a::type.\ f\ x * c)\ (l * c)\ net = ---> f\ l\ net$

thm REALLIM_NEG:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ l::real.\ ---> f\ l\ net \longrightarrow ---> (\lambda x::?'a::type.\ - f\ x)\ (- l)\ net$

thm REALLIM_NEG_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ l::real.\ ---> (\lambda x::?'a::type.\ - f\ x)\ (- l)\ net = ---> f\ l\ net$

thm REALLIM_ADD:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (l::real)\ m::real.\ ---> f\ l\ net \wedge ---> g\ m\ net \longrightarrow ---> (\lambda x::?'a::type.\ f\ x + g\ x)\ (l + m)\ net$

thm REALLIM_SUB:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (l::real)\ m::real.\ ---> f\ l\ net \wedge ---> g\ m\ net \longrightarrow ---> (\lambda x::?'a::type.\ f\ x - g\ x)\ (l - m)\ net$

thm REALLIM_MUL:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (l::real)\ m::real.\ ---> f\ l\ net \wedge ---> g\ m\ net \longrightarrow ---> (\lambda x::?'a::type.\ f\ x * g\ x)\ (l * m)\ net$

thm REALLIM_INV:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ l::real.\ ---> f\ l\ net \wedge l \neq (0::real) \longrightarrow ---> (\lambda x::?'a::type.\ inverse\_class.inverse\ (f\ x))\ (inverse\_class.inverse\ l)\ net$

thm REALLIM_DIV:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) (*g*::*?'a*::*type* $\Rightarrow$ *real*) (*l*::*real*) *m*::*real*. −−−> *f l net* $\wedge$ −−−> *g m net* $\wedge$ *m* $\neq$ (*0*::*real*) $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *f x* / *g x*) (*l* / *m*) *net*

thm REALLIM_ABS:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *l*::*real*. −−−> *f l net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. |*f x*|) |*l*| *net*

thm REALLIM_POW:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) (*l*::*real*) *n*::*nat*. −−−> *f l net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. $(f\ x)^n$) l^n *net*

thm REALLIM_MAX:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) (*g*::*?'a*::*type* $\Rightarrow$ *real*) (*l*::*real*) *m*::*real*. −−−> *f l net* $\wedge$ −−−> *g m net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *max* (*f x*) (*g x*)) (*max l m*) *net*

thm REALLIM_MIN:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) (*g*::*?'a*::*type* $\Rightarrow$ *real*) (*l*::*real*) *m*::*real*. −−−> *f l net* $\wedge$ −−−> *g m net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *min* (*f x*) (*g x*)) (*min l m*) *net*

thm REALLIM_NULL:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *l*::*real*. −−−> *f l net* = −−−> (λ*x*::*?'a*::*type*. *f x* − *l*) (*0*::*real*) *net*

thm REALLIM_NULL_ADD:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *g*::*?'a*::*type* $\Rightarrow$ *real*. −−−> *f* (*0*::*real*) *net* $\wedge$ −−−> *g* (*0*::*real*) *net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *f x* + *g x*) (*0*::*real*) *net*

thm REALLIM_NULL_LMUL:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *c*::*real*. −−−> *f* (*0*::*real*) *net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *c* $*$ *f x*) (*0*::*real*) *net*

thm REALLIM_NULL_RMUL:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *c*::*real*. −−−> *f* (*0*::*real*) *net* $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. *f x* $*$ *c*) (*0*::*real*) *net*

thm REALLIM_NULL_POW:

$\forall$ (*net*::*?'a*::*type net*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *n*::*nat*. −−−> *f* (*0*::*real*) *net* $\wedge$ *n* $\neq$ (*0*::*nat*) $\longrightarrow$ −−−> (λ*x*::*?'a*::*type*. $(f\ x)^n$) (*0*::*real*) *net*

thm REALLIM_NULL_LMUL_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ c::real.\ c \neq (0::real) \longrightarrow ---> (\lambda x::?'a::type.\ c * f\ x)\ (0::real)\ net = ---> f\ (0::real)\ net$

thm REALLIM_NULL_RMUL_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ c::real.\ c \neq (0::real) \longrightarrow ---> (\lambda x::?'a::type.\ f\ x * c)\ (0::real)\ net = ---> f\ (0::real)\ net$

thm REALLIM_RE:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ l::(real,\ 2)\ cart.\ --> f\ l\ net \longrightarrow ---> (Re \circ f)\ (Re\ l)\ net$

thm REALLIM_IM:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow (real,\ 2)\ cart)\ l::(real,\ 2)\ cart.\ --> f\ l\ net \longrightarrow ---> (Im \circ f)\ (Im\ l)\ net$

thm REALLIM_TRANSFORM_EVENTUALLY:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ l::real.\ eventually\ (\lambda x::?'a::type.\ f\ x = g\ x)\ net \wedge ---> f\ l\ net \longrightarrow ---> g\ l\ net$

thm REALLIM_TRANSFORM:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ l::real.\ ---> (\lambda x::?'a::type.\ f\ x - g\ x)\ (0::real)\ net \wedge ---> f\ l\ net \longrightarrow ---> g\ l\ net$

thm REALLIM_TRANSFORM_EQ:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ l::real.\ ---> (\lambda x::?'a::type.\ f\ x - g\ x)\ (0::real)\ net \longrightarrow ---> f\ l\ net = ---> g\ l\ net$

thm REAL_SEQ_OFFSET:

$\forall (f::nat \Rightarrow real)\ (l::real)\ k::nat.\ ---> f\ l\ sequentially \longrightarrow ---> (\lambda i::nat.\ f\ (i + k))\ l\ sequentially$

thm REAL_SEQ_OFFSET_REV:

$\forall (f::nat \Rightarrow real)\ (l::real)\ k::nat.\ ---> (\lambda i::nat.\ f\ (i + k))\ l\ sequentially \longrightarrow ---> f\ l\ sequentially$

thm REALLIM_TRANSFORM_STRADDLE:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (h::?'a::type \Rightarrow real)\ a::real.\ eventually\ (\lambda n::?'a::type.\ f\ n \leq g\ n)\ (?net::?'a::type\ net) \wedge ---> f\ a\ ?net \wedge eventually\ (\lambda n::?'a::type.\ g\ n \leq h\ n)\ ?net \wedge ---> h\ a\ ?net \longrightarrow ---> g\ a\ ?net$

thm REALLIM_TRANSFORM_BOUND:

$\forall (f::?'a::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ eventually\ (\lambda n::?'a::type.\ |f\ n| \leq g\ n)\ (?net::?'a::type\ net) \wedge ---> g\ (0::real)\ ?net \longrightarrow ---> f\ (0::real)\ ?net$

thm REAL_CONVERGENT_IMP_BOUNDED:

$\forall$ (*s::nat* $\Rightarrow$ *real*) *l::real*. *−−−>* *s l sequentially* $\longrightarrow$ *real_bounded* (*IMAGE s HOL_Light_Import.UNIV*)

thm REALLIM:

−−−> (*?f::?'a::type* $\Rightarrow$ *real*) (*?l::real*) (*?net::?'a::type net*) = (*trivial_limit ?net* $\vee$ ($\forall$ *e>0::real*. $\exists$ *y::?'a::type*. ($\exists$ *x::?'a::type*. *netord ?net x y*) $\wedge$ ($\forall$ *x::?'a::type*. *netord ?net x y* $\longrightarrow$ |*?f x* − *?l*| < *e*)))

thm REALLIM_NULL_ABS:

$\forall$ (*net::?'a::type net*) *f::?'a::type* $\Rightarrow$ *real*. *−−−>* (λ*x::?'a::type*. |*f x*|) (*0::real*) *net* = *−−−>* *f* (*0::real*) *net*

thm REALLIM_WITHIN_LE:

$\forall$ (*f::(real, ?'a::type) cart* $\Rightarrow$ *real*) (*l::real*) (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* $\Rightarrow$ *bool*. *−−−>* *f l* (*within* (*at a*) *s*) = ($\forall$ *e>0::real*. $\exists$ *d>0::real*. $\forall$ *x::(real, ?'a::type) cart*. *IN x s* $\wedge$ (*0::real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) $\leq$ *d* $\longrightarrow$ |*f x* − *l*| < *e*)

thm REALLIM_WITHIN:

$\forall$ (*f::(real, ?'a::type) cart* $\Rightarrow$ *real*) (*l::real*) (*a::(real, ?'a::type) cart*) *s::(real, ?'a::type) cart* $\Rightarrow$ *bool*. *−−−>* *f l* (*within* (*at a*) *s*) = ($\forall$ *e>0::real*. $\exists$ *d>0::real*. $\forall$ *x::(real, ?'a::type) cart*. *IN x s* $\wedge$ (*0::real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) < *d* $\longrightarrow$ |*f x* − *l*| < *e*)

thm REALLIM_AT:

$\forall$ (*f::(real, ?'a::type) cart* $\Rightarrow$ *real*) (*l::real*) *a::(real, ?'a::type) cart*. *−−−>* *f l* (*at a*) = ($\forall$ *e>0::real*. $\exists$ *d>0::real*. $\forall$ *x::(real, ?'a::type) cart*. (*0::real*) < *distance* (*x*, *a*) $\wedge$ *distance* (*x*, *a*) < *d* $\longrightarrow$ |*f x* − *l*| < *e*)

thm REALLIM_AT_INFINITY:

$\forall$ (*f::(real, ?'a::type) cart* $\Rightarrow$ *real*) *l::real*. *−−−>* *f l at_infinity* = ($\forall$ *e>0::real*. $\exists$ *b::real*. $\forall$ *x::(real, ?'a::type) cart*. *b* $\leq$ *vector_norm x* $\longrightarrow$ |*f x* − *l*| < *e*)

thm REALLIM_SEQUENTIALLY:

$\forall$ (*s::nat* $\Rightarrow$ *real*) *l::real*. *−−−>* *s l sequentially* = ($\forall$ *e>0::real*. $\exists$ *N::nat*. $\forall$ *n*$\geq$*N*. |*s n* − *l*| < *e*)

thm REALLIM_EVENTUALLY:

$\forall$ (*net::?'a::type net*) (*f::?'a::type* $\Rightarrow$ *real*) *l::real*. *eventually* (λ*x::?'a::type*. *f x* = *l*) *net* $\longrightarrow$ *−−−>* *f l net*

thm LIM_COMPONENTWISE:

$\forall$ (*net::?'b::type net*) *f::?'b::type* $\Rightarrow$ (*real*, *?'a::type*) *cart*. *−−>* *f* (*?l::(real, ?'a::type) cart*) *net* = ($\forall$ *i::nat*. (*1::nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *−−−>* (λ*x::?'b::type*. \$ (*f x*) *i*) (\$ *?l i*) *net*)

thm REALLIM_UBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (l::real)\ b::real.\ ---> f\ l\ net \wedge \neg$ $trivial\_limit\ net \wedge eventually\ (\lambda x::?'a::type.\ f\ x \leq b)\ net \longrightarrow l \leq b$

thm REALLIM_LBOUND:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (l::real)\ b::real.\ ---> f\ l\ net \wedge \neg$ $trivial\_limit\ net \wedge eventually\ (\lambda x::?'a::type.\ b \leq f\ x)\ net \longrightarrow b \leq l$

thm REALLIM_LE:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ (l::real)\ m::real.$ $---> f\ l\ net \wedge ---> g\ m\ net \wedge \neg\ trivial\_limit\ net \wedge eventually\ (\lambda x::?'a::type.$ $f\ x \leq g\ x)\ net \longrightarrow l \leq m$

thm REALLIM_CONST_EQ:

$\forall (net::?'a::type\ net)\ (c::real)\ d::real.\ ---> (\lambda x::?'a::type.\ c)\ d\ net = (trivial\_limit$ $net \vee c = d)$

thm REALLIM_SUM:

$\forall (f::?'b::type \Rightarrow ?'a::type \Rightarrow real)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \wedge (\forall i::?'b::type.$ $IN\ i\ s \longrightarrow ---> (f\ i)\ ((?l::?'b::type \Rightarrow real)\ i)\ (?net::?'a::type\ net)) \longrightarrow$ $---> (\lambda x::?'a::type.\ sum\ s\ (\lambda i::?'b::type.\ f\ i\ x))\ (sum\ s\ ?l)\ ?net$

thm REALLIM_NULL_COMPARISON:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ eventually\ (\lambda x::?'a::type.$ $|f\ x| \leq g\ x)\ net \wedge ---> g\ (0::real)\ net \longrightarrow ---> f\ (0::real)\ net$

thm DEF_real_sums:

$real\_sums = (\lambda(\_1885331::nat \Rightarrow real)\ (\_1885332::real)\ \_1885333::nat \Rightarrow bool.$ $---> (\lambda n::nat.\ sum\ (HOL\_Light\_Import.INTER\ \_1885333\ (dotdot\ (0::nat)$ $n))\ \_1885331)\ \_1885332\ sequentially)$

thm real_sums:

$\forall (s::nat \Rightarrow bool)\ (f::nat \Rightarrow real)\ l::real.\ real\_sums\ f\ l\ s = ---> (\lambda n::nat.$ $sum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ (0::nat)\ n))\ f)\ l\ sequentially$

thm DEF_real_infsum:

$real\_infsum = (\lambda(\_1885352::nat \Rightarrow bool)\ \_1885353::nat \Rightarrow real.\ SOME\ l::real.$ $real\_sums\ \_1885353\ l\ \_1885352)$

thm real_infsum:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_infsum\ s\ f = (SOME\ l::real.\ real\_sums\ f$ $l\ s)$

thm DEF_real_summable:

$real\_summable = (\lambda(\_1885364::nat \Rightarrow bool)\ \_1885365::nat \Rightarrow real.\ \exists l::real.$ $real\_sums\ \_1885365\ l\ \_1885364)$

thm real_summable:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ f = (\exists\, l::real.\ real\_sums\ f\ l\ s)$

thm REAL_SUMS:

$real\_sums\ (?f::nat \Rightarrow real)\ (?l::real) = sums\ (lift \circ ?f)\ (lift\ ?l)$

thm REAL_SUMS_RE:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow real\_sums\ (Re \circ f)\ (Re\ l)\ s$

thm REAL_SUMS_IM:

$\forall (f::nat \Rightarrow (real,\ 2)\ cart)\ (l::(real,\ 2)\ cart)\ s::nat \Rightarrow bool.\ sums\ f\ l\ s \longrightarrow real\_sums\ (Im \circ f)\ (Im\ l)\ s$

thm REAL_SUMS_COMPLEX:

$\forall (f::nat \Rightarrow real)\ (l::real)\ s::nat \Rightarrow bool.\ real\_sums\ f\ l\ s = sums\ (Cx \circ f)\ (Cx\ l)\ s$

thm REAL_SUMMABLE:

$real\_summable\ (?s::nat \Rightarrow bool)\ (?f::nat \Rightarrow real) = summable\ ?s\ (lift \circ ?f)$

thm REAL_SUMMABLE_COMPLEX:

$real\_summable\ (?s::nat \Rightarrow bool)\ (?f::nat \Rightarrow real) = summable\ ?s\ (Cx \circ ?f)$

thm REAL_SERIES_CAUCHY:

$(\exists\, l::real.\ real\_sums\ (?f::nat \Rightarrow real)\ l\ (?s::nat \Rightarrow bool)) = (\forall\, e>0::real.\ \exists\, N::nat.\ \forall (m::nat)\ n::nat.\ N \le m \longrightarrow |sum\ (HOL\_Light\_Import.INTER\ ?s\ (dotdot\ m\ n))\ ?f| < e)$

thm REAL_SUMS_EQ:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\forall\, x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \wedge real\_sums\ f\ (?l::real)\ k \longrightarrow real\_sums\ g\ ?l\ k$

thm REAL_SUMS_SUMMABLE:

$\forall (f::nat \Rightarrow real)\ (l::real)\ s::nat \Rightarrow bool.\ real\_sums\ f\ l\ s \longrightarrow real\_summable\ s\ f$

thm REAL_SUMS_INFSUM:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_sums\ f\ (real\_infsum\ s\ f)\ s = real\_summable\ s\ f$

thm REAL_INFSUM_COMPLEX:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ f \longrightarrow real\_infsum\ s\ f = Re\ (infsum\ s\ (Cx \circ f))$

thm REAL_SERIES_FROM:

$\forall (f::nat \Rightarrow real)\ (l::real)\ k::nat.\ real\_sums\ f\ l\ (from\ k) = {-}{-}{-}{>}\ (\lambda n::nat.\ sum\ (dotdot\ k\ n)\ f)\ l\ sequentially$

thm REAL_SERIES_UNIQUE:

$\forall (f::nat \Rightarrow real)\ (l::real)\ (l'::real)\ s::nat \Rightarrow bool.\ real\_sums\ f\ l\ s \land real\_sums\ f\ l'\ s \longrightarrow l = l'$

thm REAL_INFSUM_UNIQUE:

$\forall (f::nat \Rightarrow real)\ (l::real)\ s::nat \Rightarrow bool.\ real\_sums\ f\ l\ s \longrightarrow real\_infsum\ s\ f = l$

thm REAL_SERIES_FINITE:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ FINITE\ s \longrightarrow real\_sums\ f\ (sum\ s\ f)\ s$

thm REAL_SUMMABLE_IFF_EVENTUALLY:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\exists N::nat.\ \forall n::nat.\ N \leq n \land IN\ n\ k \longrightarrow f\ n = g\ n) \longrightarrow real\_summable\ k\ f = real\_summable\ k\ g$

thm REAL_SUMMABLE_EQ_EVENTUALLY:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\exists N::nat.\ \forall n::nat.\ N \leq n \land IN\ n\ k \longrightarrow f\ n = g\ n) \land real\_summable\ k\ f \longrightarrow real\_summable\ k\ g$

thm REAL_SUMMABLE_IFF_COFINITE:

$\forall (f::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_summable\ s\ f = real\_summable\ t\ f$

thm REAL_SUMMABLE_EQ_COFINITE:

$\forall (f::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \land real\_summable\ s\ f \longrightarrow real\_summable\ t\ f$

thm REAL_SUMMABLE_FROM_ELSEWHERE:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ real\_summable\ (from\ m)\ f \longrightarrow real\_summable\ (from\ n)\ f$

thm REAL_SERIES_GOESTOZERO:

$\forall (s::nat \Rightarrow bool)\ x::nat \Rightarrow real.\ real\_summable\ s\ x \longrightarrow (\forall e>0::real.\ eventually\ (\lambda n::nat.\ IN\ n\ s \longrightarrow |x\ n| < e)\ sequentially)$

thm REAL_SUMMABLE_IMP_TOZERO:

$\forall (f::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ real\_summable\ k\ f \longrightarrow --->\ (\lambda n::nat.\ if\ IN\ n\ k\ then\ f\ n\ else\ (0::real))\ (0::real)\ sequentially$

thm REAL_SUMMABLE_IMP_BOUNDED:

$\forall (f::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ real\_summable\ k\ f \longrightarrow real\_bounded\ (IMAGE\ f\ k)$

thm REAL_SUMMABLE_IMP_REAL_SUMS_BOUNDED:

$\forall (f::nat \Rightarrow real)\ k::nat.\ real\_summable\ (from\ k)\ f \longrightarrow real\_bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2400::real.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2400\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (sum\ (dotdot\ k\ n)\ f)))$

thm REAL_SERIES_0:

$\forall\, s::nat \Rightarrow bool.\ real\_sums\ (\lambda n::nat.\ 0::real)\ (0::real)\ s$

thm REAL_SERIES_ADD:

$\forall\, (x::nat \Rightarrow real)\ (x0::real)\ (y::nat \Rightarrow real)\ (y0::real)\ s::nat \Rightarrow bool.\ real\_sums\ x\ x0\ s \wedge real\_sums\ y\ y0\ s \longrightarrow real\_sums\ (\lambda n::nat.\ x\ n + y\ n)\ (x0 + y0)\ s$

thm REAL_SERIES_SUB:

$\forall\, (x::nat \Rightarrow real)\ (x0::real)\ (y::nat \Rightarrow real)\ (y0::real)\ s::nat \Rightarrow bool.\ real\_sums\ x\ x0\ s \wedge real\_sums\ y\ y0\ s \longrightarrow real\_sums\ (\lambda n::nat.\ x\ n - y\ n)\ (x0 - y0)\ s$

thm REAL_SERIES_LMUL:

$\forall\, (x::nat \Rightarrow real)\ (x0::real)\ (c::real)\ s::nat \Rightarrow bool.\ real\_sums\ x\ x0\ s \longrightarrow real\_sums\ (\lambda n::nat.\ c * x\ n)\ (c * x0)\ s$

thm REAL_SERIES_RMUL:

$\forall\, (x::nat \Rightarrow real)\ (x0::real)\ (c::real)\ s::nat \Rightarrow bool.\ real\_sums\ x\ x0\ s \longrightarrow real\_sums\ (\lambda n::nat.\ x\ n * c)\ (x0 * c)\ s$

thm REAL_SERIES_NEG:

$\forall\, (x::nat \Rightarrow real)\ (x0::real)\ s::nat \Rightarrow bool.\ real\_sums\ x\ x0\ s \longrightarrow real\_sums\ (\lambda n::nat.\ - x\ n)\ (- x0)\ s$

thm REAL_SUMS_IFF:

$\forall\, (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\forall\, x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \longrightarrow real\_sums\ f\ (?l::real)\ k = real\_sums\ g\ ?l\ k$

thm REAL_SERIES_FINITE_SUPPORT:

$\forall\, (f::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ k::nat \Rightarrow bool.\ FINITE\ (HOL\_Light\_Import.INTER\ s\ k) \wedge (\forall\, x::nat.\ \neg\ IN\ x\ (HOL\_Light\_Import.INTER\ s\ k) \longrightarrow f\ x = (0::real)) \longrightarrow real\_sums\ f\ (sum\ (HOL\_Light\_Import.INTER\ s\ k)\ f)\ k$

thm REAL_SERIES_DIFFS:

$\forall\, (f::nat \Rightarrow real)\ k::nat.\ --->\ f\ (0::real)\ sequentially \longrightarrow real\_sums\ (\lambda n::nat.\ f\ n - f\ (n + (1::nat)))\ (f\ k)\ (from\ k)$

thm REAL_SERIES_TRIVIAL:

$\forall\, f::nat \Rightarrow real.\ real\_sums\ f\ (0::real)\ EMPTY$

thm REAL_SERIES_RESTRICT:

$\forall\, (f::nat \Rightarrow real)\ (k::nat \Rightarrow bool)\ l::real.\ real\_sums\ (\lambda n::nat.\ if\ IN\ n\ k\ then\ f\ n\ else\ (0::real))\ l\ HOL\_Light\_Import.UNIV = real\_sums\ f\ l\ k$

thm REAL_SERIES_SUM:

$\forall\, (f::nat \Rightarrow real)\ (l::real)\ (k::nat \Rightarrow bool)\ s::nat \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ s\ k \wedge (\forall\, x::nat.\ \neg\ IN\ x\ s \longrightarrow f\ x = (0::real)) \wedge sum\ s\ f = l \longrightarrow real\_sums\ f\ l\ k$

thm REAL_SUMS_LE2:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ (y::real)\ z::real.\ real\_sums\ f\ y\ s \wedge real\_sums\ g\ z\ s \wedge (\forall i::nat.\ IN\ i\ s \longrightarrow f\ i \leq g\ i) \longrightarrow y \leq z$

thm REAL_SUMS_REINDEX:

$\forall (k::nat)\ (a::nat \Rightarrow real)\ (l::real)\ n::nat.\ real\_sums\ (\lambda x::nat.\ a\ (x + k))\ l\ (from\ n) = real\_sums\ a\ l\ (from\ (n + k))$

thm REAL_INFSUM:

$\forall (f::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ f \longrightarrow real\_infsum\ s\ f = HOL\_Light\_Import.drop\ (infsum\ s\ (lift \circ f))$

thm REAL_PARTIAL_SUMS_LE_INFSUM:

$\forall (f::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ n::nat.\ (\forall i::nat.\ IN\ i\ s \longrightarrow (0::real) \leq f\ i) \wedge real\_summable\ s\ f \longrightarrow sum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ (0::nat)\ n))\ f \leq real\_infsum\ s\ f$

thm REAL_SUMMABLE_0:

$\forall s::nat \Rightarrow bool.\ real\_summable\ s\ (\lambda n::nat.\ 0::real)$

thm REAL_SUMMABLE_ADD:

$\forall (x::nat \Rightarrow real)\ (y::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ x \wedge real\_summable\ s\ y \longrightarrow real\_summable\ s\ (\lambda n::nat.\ x\ n + y\ n)$

thm REAL_SUMMABLE_SUB:

$\forall (x::nat \Rightarrow real)\ (y::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ x \wedge real\_summable\ s\ y \longrightarrow real\_summable\ s\ (\lambda n::nat.\ x\ n - y\ n)$

thm REAL_SUMMABLE_LMUL:

$\forall (s::nat \Rightarrow bool)\ (x::nat \Rightarrow real)\ c::real.\ real\_summable\ s\ x \longrightarrow real\_summable\ s\ (\lambda n::nat.\ c * x\ n)$

thm REAL_SUMMABLE_RMUL:

$\forall (s::nat \Rightarrow bool)\ (x::nat \Rightarrow real)\ c::real.\ real\_summable\ s\ x \longrightarrow real\_summable\ s\ (\lambda n::nat.\ x\ n * c)$

thm REAL_SUMMABLE_NEG:

$\forall (x::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ x \longrightarrow real\_summable\ s\ (\lambda n::nat.\ - x\ n)$

thm REAL_SUMMABLE_IFF:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \longrightarrow real\_summable\ k\ f = real\_summable\ k\ g$

thm REAL_SUMMABLE_EQ:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \wedge real\_summable\ k\ f \longrightarrow real\_summable\ k\ g$

thm REAL_SERIES_SUBSET:

$\forall (x::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ (t::nat \Rightarrow bool)\ l::real.\ SUBSET\ s\ t \wedge real\_sums\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ x\ i\ else\ (0::real))\ l\ t \longrightarrow real\_sums\ x\ l\ s$

thm REAL_SUMMABLE_SUBSET:

$\forall (x::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ t::nat \Rightarrow bool.\ SUBSET\ s\ t \wedge real\_summable\ t\ (\lambda i::nat.\ if\ IN\ i\ s\ then\ x\ i\ else\ (0::real)) \longrightarrow real\_summable\ s\ x$

thm REAL_SUMMABLE_TRIVIAL:

$\forall f::nat \Rightarrow real.\ real\_summable\ EMPTY\ f$

thm REAL_SUMMABLE_RESTRICT:

$\forall (f::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ real\_summable\ HOL\_Light\_Import.UNIV\ (\lambda n::nat.\ if\ IN\ n\ k\ then\ f\ n\ else\ (0::real)) = real\_summable\ k\ f$

thm REAL_SUMS_FINITE_DIFF:

$\forall (f::nat \Rightarrow real)\ (t::nat \Rightarrow bool)\ (s::nat \Rightarrow bool)\ l::real.\ SUBSET\ t\ s \wedge FINITE\ t \wedge real\_sums\ f\ l\ s \longrightarrow real\_sums\ f\ (l - sum\ t\ f)\ (DIFF\ s\ t)$

thm REAL_SUMS_FINITE_UNION:

$\forall (f::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ (t::nat \Rightarrow bool)\ l::real.\ FINITE\ t \wedge real\_sums\ f\ l\ s \longrightarrow real\_sums\ f\ (l + sum\ (DIFF\ t\ s)\ f)\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_SUMS_OFFSET:

$\forall (f::nat \Rightarrow real)\ (l::real)\ (m::nat)\ n::nat.\ real\_sums\ f\ l\ (from\ m) \wedge m < n \longrightarrow real\_sums\ f\ (l - sum\ (dotdot\ m\ (n - (1::nat)))\ f)\ (from\ n)$

thm REAL_SUMS_OFFSET_REV:

$\forall (f::nat \Rightarrow real)\ (l::real)\ (m::nat)\ n::nat.\ real\_sums\ f\ l\ (from\ m) \wedge n < m \longrightarrow real\_sums\ f\ (l + sum\ (dotdot\ n\ (m - (1::nat)))\ f)\ (from\ n)$

thm REAL_INFSUM_0:

$real\_infsum\ (?s::nat \Rightarrow bool)\ (\lambda i::nat.\ 0::real) = (0::real)$

thm REAL_INFSUM_ADD:

$\forall (x::nat \Rightarrow real)\ (y::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ x \wedge real\_summable\ s\ y \longrightarrow real\_infsum\ s\ (\lambda i::nat.\ x\ i + y\ i) = real\_infsum\ s\ x + real\_infsum\ s\ y$

thm REAL_INFSUM_SUB:

$\forall (x::nat \Rightarrow real)\ (y::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ x \wedge real\_summable\ s\ y \longrightarrow real\_infsum\ s\ (\lambda i::nat.\ x\ i - y\ i) = real\_infsum\ s\ x - real\_infsum\ s\ y$

thm REAL_INFSUM_LMUL:

$\forall (s::nat \Rightarrow bool)\ (x::nat \Rightarrow real)\ c::real.\ real\_summable\ s\ x \longrightarrow real\_infsum\ s\ (\lambda n::nat.\ c * x\ n) = c * real\_infsum\ s\ x$

thm REAL_INFSUM_RMUL:

$\forall (s::nat \Rightarrow bool)\ (x::nat \Rightarrow real)\ c::real.\ real\_summable\ s\ x \longrightarrow real\_infsum\ s\ (\lambda n::nat.\ x\ n * c) = real\_infsum\ s\ x * c$

thm REAL_INFSUM_NEG:

$\forall (s::nat \Rightarrow bool)\ x::nat \Rightarrow real.\ real\_summable\ s\ x \longrightarrow real\_infsum\ s\ (\lambda n::nat.\ -\ x\ n) = -\ real\_infsum\ s\ x$

thm REAL_INFSUM_EQ:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ real\_summable\ k\ f \wedge real\_summable\ k\ g \wedge (\forall x::nat.\ IN\ x\ k \longrightarrow f\ x = g\ x) \longrightarrow real\_infsum\ k\ f = real\_infsum\ k\ g$

thm REAL_INFSUM_RESTRICT:

$\forall (k::nat \Rightarrow bool)\ a::nat \Rightarrow real.\ real\_infsum\ HOL\_Light\_Import.UNIV\ (\lambda n::nat.\ if\ IN\ n\ k\ then\ a\ n\ else\ (0::real)) = real\_infsum\ k\ a$

thm REAL_SERIES_CAUCHY_UNIFORM:

$\forall (P::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow nat \Rightarrow real)\ k::nat \Rightarrow bool.\ (\exists l::?'a::type \Rightarrow real.\ \forall e>0::real.\ \exists N::nat.\ \forall (n::nat)\ x::?'a::type.\ N \le n \wedge P\ x \longrightarrow |sum\ (HOL\_Light\_Import.INTER\ k\ (dotdot\ (0::nat)\ n))\ (f\ x) - l\ x| < e) = (\forall e>0::real.\ \exists N::nat.\ \forall (m::nat)\ (n::nat)\ x::?'a::type.\ N \le m \wedge P\ x \longrightarrow |sum\ (HOL\_Light\_Import.INTER\ k\ (dotdot\ m\ n))\ (f\ x)| < e)$

thm REAL_SERIES_COMPARISON:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ (\exists l::real.\ real\_sums\ g\ l\ s) \wedge (\exists N::nat.\ \forall n::nat.\ N \le n \wedge IN\ n\ s \longrightarrow |f\ n| \le g\ n) \longrightarrow (\exists l::real.\ real\_sums\ f\ l\ s)$

thm REAL_SUMMABLE_COMPARISON:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ s::nat \Rightarrow bool.\ real\_summable\ s\ g \wedge (\exists N::nat.\ \forall n::nat.\ N \le n \wedge IN\ n\ s \longrightarrow |f\ n| \le g\ n) \longrightarrow real\_summable\ s\ f$

thm REAL_SERIES_COMPARISON_UNIFORM:

$\forall (f::?'a::type \Rightarrow nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (P::?'a::type \Rightarrow bool)\ s::nat \Rightarrow bool.\ (\exists l::real.\ real\_sums\ g\ l\ s) \wedge (\exists N::nat.\ \forall (n::nat)\ x::?'a::type.\ N \le n \wedge IN\ n\ s \wedge P\ x \longrightarrow |f\ x\ n| \le g\ n) \longrightarrow (\exists l::?'a::type \Rightarrow real.\ \forall e>0::real.\ \exists N::nat.\ \forall (n::nat)\ x::?'a::type.\ N \le n \wedge P\ x \longrightarrow |sum\ (HOL\_Light\_Import.INTER\ s\ (dotdot\ (0::nat)\ n))\ (f\ x) - l\ x| < e)$

thm REAL_SERIES_RATIO:

$\forall (c::real)\ (a::nat \Rightarrow real)\ (s::nat \Rightarrow bool)\ N::nat.\ c < (1::real) \wedge (\forall n \ge N.\ |a\ (Suc\ n)| \le c * |a\ n|) \longrightarrow (\exists l::real.\ real\_sums\ a\ l\ s)$

thm BOUNDED_PARTIAL_REAL_SUMS:

$\forall (f::nat \Rightarrow real)\ k::nat.\ real\_bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2405::real.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2405\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (sum\ (dotdot\ k\ n)\ f))) \longrightarrow real\_bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2406::real.\ \exists (m::nat)\ n::nat.\ SETSPEC\ GEN\%PVAR\%2406\ (IN\ m\ HOL\_Light\_Import.UNIV \wedge IN\ n\ HOL\_Light\_Import.UNIV)\ (sum\ (dotdot\ m\ n)\ f)))$

thm REAL_SERIES_DIRICHLET:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (N::nat)\ (k::nat)\ m::nat.\ real\_bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2409::real.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2409\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (sum\ (dotdot\ m\ n)\ f))) \wedge (\forall n \geq N.\ g\ (n + (1::nat)) \leq g\ n) \wedge ---> g\ (0::real)\ sequentially \longrightarrow real\_summable\ (from\ k)\ (\lambda n::nat.\ g\ n * f\ n)$

thm REAL_SERIES_ABSCONV_IMP_CONV:

$\forall (x::nat \Rightarrow real)\ k::nat \Rightarrow bool.\ real\_summable\ k\ (\lambda n::nat.\ |x\ n|) \longrightarrow real\_summable\ k\ x$

thm REAL_SUMS_GP:

$\forall (n::nat)\ x::real.\ |x| < (1::real) \longrightarrow real\_sums\ (op\ \hat{}\ x)\ (x^n\ /\ ((1::real) - x))\ (from\ n)$

thm REAL_SUMMABLE_GP:

$\forall (x::real)\ k::nat \Rightarrow bool.\ |x| < (1::real) \longrightarrow real\_summable\ k\ (op\ \hat{}\ x)$

thm REAL_ABEL_LEMMA:

$\forall (a::nat \Rightarrow real)\ (M::real)\ (r::real)\ r0::real.\ (0::real) \leq r \wedge r < r0 \wedge (\forall n::nat.\ IN\ n\ (?k::nat \Rightarrow bool) \longrightarrow |a\ n| * r0^n \leq M) \longrightarrow real\_summable\ ?k\ (\lambda n::nat.\ |a\ n| * r^n)$

thm REAL_POWER_SERIES_CONV_IMP_ABSCONV:

$\forall (a::nat \Rightarrow real)\ (k::nat \Rightarrow bool)\ (w::real)\ z::real.\ real\_summable\ k\ (\lambda n::nat.\ a\ n * z^n) \wedge |w| < |z| \longrightarrow real\_summable\ k\ (\lambda n::nat.\ |a\ n * w^n|)$

thm POWER_REAL_SERIES_CONV_IMP_ABSCONV_WEAK:

$\forall (a::nat \Rightarrow real)\ (k::nat \Rightarrow bool)\ (w::real)\ z::real.\ real\_summable\ k\ (\lambda n::nat.\ a\ n * z^n) \wedge |w| < |z| \longrightarrow real\_summable\ k\ (\lambda n::nat.\ |a\ n| * w^n)$

thm REALLIM_1_OVER_N:

$---> (\lambda n::nat.\ inverse\_class.inverse\ (real\_of\_nat\ n))\ (0::real)\ sequentially$

thm REALLIM_LOG_OVER_N:

$---> (\lambda n::nat.\ log\ (real\_of\_nat\ n)\ /\ real\_of\_nat\ n)\ (0::real)\ sequentially$

thm REALLIM_1_OVER_LOG:

$---> (\lambda n::nat.\ inverse\_class.inverse\ (log\ (real\_of\_nat\ n)))\ (0::real)\ sequentially$

thm REALLIM_POWN:

$\forall z{::}real.\ |z| < (1{::}real) \longrightarrow ---> (op\ \hat{}\ z)\ (0{::}real)\ sequentially$

thm DEF_atreal:

$atreal = (\lambda\_1886739{::}real.\ mk\_net\ (\lambda(x{::}real)\ y{::}real.\ (0{::}real) < |x - \_1886739| \wedge |x - \_1886739| \leq |y - \_1886739|))$

thm atreal:

$\forall a{::}real.\ atreal\ a = mk\_net\ (\lambda(x{::}real)\ y{::}real.\ (0{::}real) < |x - a| \wedge |x - a| \leq |y - a|)$

thm at_posinfinity:

$at\_posinfinity = mk\_net\ (\lambda(x{::}real)\ y{::}real.\ y \leq x)$

thm at_neginfinity:

$at\_neginfinity = mk\_net\ op \leq$

thm ATREAL:

$\forall (a{::}real)\ (x{::}real)\ y{::}real.\ netord\ (atreal\ a)\ x\ y = ((0{::}real) < |x - a| \wedge |x - a| \leq |y - a|)$

thm AT_POSINFINITY:

$\forall (x{::}real)\ y{::}real.\ netord\ at\_posinfinity\ x\ y = (y \leq x)$

thm AT_NEGINFINITY:

$\forall (x{::}real)\ y{::}real.\ netord\ at\_neginfinity\ x\ y = (x \leq y)$

thm WITHINREAL_UNIV:

$\forall x{::}real.\ within\ (atreal\ x)\ HOL\_Light\_Import.UNIV = atreal\ x$

thm TRIVIAL_LIMIT_ATREAL:

$\forall a{::}real.\ \neg\ trivial\_limit\ (atreal\ a)$

thm TRIVIAL_LIMIT_AT_POSINFINITY:

$\neg\ trivial\_limit\ at\_posinfinity$

thm TRIVIAL_LIMIT_AT_NEGINFINITY:

$\neg\ trivial\_limit\ at\_neginfinity$

thm NETLIMIT_WITHINREAL:

$\forall (a{::}real)\ s{::}real \Rightarrow bool.\ \neg\ trivial\_limit\ (within\ (atreal\ a)\ s) \longrightarrow netlimit\ (within\ (atreal\ a)\ s) = a$

thm NETLIMIT_ATREAL:

$\forall a{::}real.\ netlimit\ (atreal\ a) = a$

thm EVENTUALLY_WITHINREAL_LE:

$\forall (s::real \Rightarrow bool)\ (a::real)\ p::real \Rightarrow bool.\ eventually\ p\ (within\ (atreal\ a)\ s) = (\exists\, d>0::real.\ \forall\, x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| \leq d \longrightarrow p\ x)$

thm EVENTUALLY_WITHINREAL:

$\forall (s::real \Rightarrow bool)\ (a::real)\ p::real \Rightarrow bool.\ eventually\ p\ (within\ (atreal\ a)\ s) = (\exists\, d>0::real.\ \forall\, x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| < d \longrightarrow p\ x)$

thm EVENTUALLY_ATREAL:

$\forall (a::real)\ p::real \Rightarrow bool.\ eventually\ p\ (atreal\ a) = (\exists\, d>0::real.\ \forall\, x::real.\ (0::real) < |x - a| \wedge |x - a| < d \longrightarrow p\ x)$

thm EVENTUALLY_AT_POSINFINITY:

$\forall\, p::real \Rightarrow bool.\ eventually\ p\ at\_posinfinity = (\exists\, b::real.\ \forall\, x \geq b.\ p\ x)$

thm EVENTUALLY_AT_NEGINFINITY:

$\forall\, p::real \Rightarrow bool.\ eventually\ p\ at\_neginfinity = (\exists\, b::real.\ \forall\, x \leq b.\ p\ x)$

thm LIM_WITHINREAL_LE:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (a::real)\ s::real \Rightarrow bool.\ -\!-\!>\ f\ l\ (within\ (atreal\ a)\ s) = (\forall\, e>0::real.\ \exists\, d>0::real.\ \forall\, x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| \leq d \longrightarrow distance\ (f\ x,\ l) < e)$

thm LIM_WITHINREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (a::real)\ s::real \Rightarrow bool.\ -\!-\!>\ f\ l\ (within\ (atreal\ a)\ s) = (\forall\, e>0::real.\ \exists\, d>0::real.\ \forall\, x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| < d \longrightarrow distance\ (f\ x,\ l) < e)$

thm LIM_ATREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ a::real.\ -\!-\!>\ f\ l\ (atreal\ a) = (\forall\, e>0::real.\ \exists\, d>0::real.\ \forall\, x::real.\ (0::real) < |x - a| \wedge |x - a| < d \longrightarrow distance\ (f\ x,\ l) < e)$

thm LIM_AT_POSINFINITY:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ -\!-\!>\ f\ l\ at\_posinfinity = (\forall\, e>0::real.\ \exists\, b::real.\ \forall\, x \geq b.\ distance\ (f\ x,\ l) < e)$

thm LIM_AT_NEGINFINITY:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ -\!-\!>\ f\ l\ at\_neginfinity = (\forall\, e>0::real.\ \exists\, b::real.\ \forall\, x \leq b.\ distance\ (f\ x,\ l) < e)$

thm REALLIM_WITHINREAL_LE:

$\forall (f::real \Rightarrow real)\ (l::real)\ (a::real)\ s::real \Rightarrow bool.\ -\!-\!-\!>\ f\ l\ (within\ (atreal\ a)\ s) = (\forall\, e>0::real.\ \exists\, d>0::real.\ \forall\, x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| \leq d \longrightarrow |f\ x - l| < e)$

thm REALLIM_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (l::real)\ (a::real)\ s::real \Rightarrow bool.\ {-}{-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ s) = (\forall\ e{>}0::real.\ \exists\ d{>}0::real.\ \forall\ x::real.\ IN\ x\ s \wedge (0::real) < |x - a| \wedge |x - a| < d \longrightarrow |f\ x - l| < e)$

thm REALLIM_ATREAL:

$\forall (f::real \Rightarrow real)\ (l::real)\ a::real.\ {-}{-}{-}{>}\ f\ l\ (atreal\ a) = (\forall\ e{>}0::real.\ \exists\ d{>}0::real.\ \forall\ x::real.\ (0::real) < |x - a| \wedge |x - a| < d \longrightarrow |f\ x - l| < e)$

thm REALLIM_AT_POSINFINITY:

$\forall (f::real \Rightarrow real)\ l::real.\ {-}{-}{-}{>}\ f\ l\ at\_posinfinity = (\forall\ e{>}0::real.\ \exists\ b::real.\ \forall\ x{\geq}b.\ |f\ x - l| < e)$

thm REALLIM_AT_NEGINFINITY:

$\forall (f::real \Rightarrow real)\ l::real.\ {-}{-}{-}{>}\ f\ l\ at\_neginfinity = (\forall\ e{>}0::real.\ \exists\ b::real.\ \forall\ x{\leq}b.\ |f\ x - l| < e)$

thm LIM_ATREAL_WITHINREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (a::real)\ s::real \Rightarrow bool.\ {-}{-}{>}\ f\ l\ (atreal\ a) \longrightarrow {-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ s)$

thm REALLIM_ATREAL_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (l::real)\ (a::real)\ s::real \Rightarrow bool.\ {-}{-}{-}{>}\ f\ l\ (atreal\ a) \longrightarrow {-}{-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ s)$

thm REALLIM_WITHIN_SUBSET:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (l::real)\ (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ {-}{-}{-}{>}\ f\ l\ (within\ (at\ a)\ s) \wedge SUBSET\ t\ s \longrightarrow {-}{-}{-}{>}\ f\ l\ (within\ (at\ a)\ t)$

thm REALLIM_WITHINREAL_SUBSET:

$\forall (f::real \Rightarrow real)\ (l::real)\ (a::real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ {-}{-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ s) \wedge SUBSET\ t\ s \longrightarrow {-}{-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ t)$

thm LIM_WITHINREAL_SUBSET:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (a::real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ {-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ s) \wedge SUBSET\ t\ s \longrightarrow {-}{-}{>}\ f\ l\ (within\ (atreal\ a)\ t)$

thm REALLIM_ATREAL_ID:

${-}{-}{-}{>}\ (\lambda x::real.\ x)\ (?a::real)\ (atreal\ ?a)$

thm REALLIM_WITHINREAL_ID:

$\forall\ a::real.\ {-}{-}{-}{>}\ (\lambda x::real.\ x)\ a\ (within\ (atreal\ a)\ (?s::real \Rightarrow bool))$

thm LIM_TRANSFORM_WITHINREAL_SET:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (a::real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ eventually\ (\lambda x::real.\ IN\ x\ s = IN\ x\ t)\ (atreal\ a) \longrightarrow -\!-\!>\ f\ (?l::(real,\ ?'a::type)\ cart)\ (within\ (atreal\ a)\ s) = -\!-\!>\ f\ ?l\ (within\ (atreal\ a)\ t)$

thm REALLIM_TRANSFORM_WITHIN_SET:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ eventually\ (\lambda x::(real,\ ?'a::type)\ cart.\ IN\ x\ s = IN\ x\ t)\ (at\ a) \longrightarrow -\!-\!-\!>\ f\ (?l::real)\ (within\ (at\ a)\ s) = -\!-\!-\!>\ f\ ?l\ (within\ (at\ a)\ t)$

thm REALLIM_TRANSFORM_WITHINREAL_SET:

$\forall (f::real \Rightarrow real)\ (a::real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ eventually\ (\lambda x::real.\ IN\ x\ s = IN\ x\ t)\ (atreal\ a) \longrightarrow -\!-\!-\!>\ f\ (?l::real)\ (within\ (atreal\ a)\ s) = -\!-\!-\!>\ f\ ?l\ (within\ (atreal\ a)\ t)$

thm TRIVIAL_LIMIT_WITHINREAL_WITHIN:

$trivial\_limit\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = trivial\_limit\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm TRIVIAL_LIMIT_WITHINREAL_WITHINCOMPLEX:

$trivial\_limit\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = trivial\_limit\ (within\ (at\ (Cx\ ?x))\ (HOL\_Light\_Import.INTER\ HOL\_Light\_Import.real\ (IMAGE\ Cx\ ?s)))$

thm LIM_WITHINREAL_WITHINCOMPLEX:

$-\!-\!>\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (?a::(real,\ ?'a::type)\ cart)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = -\!-\!>\ (?f \circ Re)\ ?a\ (within\ (at\ (Cx\ ?x))\ (HOL\_Light\_Import.INTER\ HOL\_Light\_Import.real\ (IMAGE\ Cx\ ?s)))$

thm LIM_ATREAL_ATCOMPLEX:

$-\!-\!>\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (?a::(real,\ ?'a::type)\ cart)\ (atreal\ (?x::real)) = -\!-\!>\ (?f \circ Re)\ ?a\ (within\ (at\ (Cx\ ?x))\ HOL\_Light\_Import.real)$

thm LIM_WITHINREAL_WITHIN:

$-\!-\!>\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (?a::(real,\ ?'a::type)\ cart)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = -\!-\!>\ (?f \circ HOL\_Light\_Import.drop)\ ?a\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm LIM_ATREAL_AT:

$-\!-\!>\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (?a::(real,\ ?'a::type)\ cart)\ (atreal\ (?x::real)) = -\!-\!>\ (?f \circ HOL\_Light\_Import.drop)\ ?a\ (at\ (lift\ ?x))$

thm REALLIM_WITHINREAL_WITHIN:

$-\!-\!-\!>\ (?f::real \Rightarrow real)\ (?a::real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = -\!-\!-\!>\ (?f \circ HOL\_Light\_Import.drop)\ ?a\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm REALLIM_ATREAL_AT:

$---> (?f::real \Rightarrow real)\ (?a::real)\ (atreal\ (?x::real)) = ---> (?f \circ HOL\_Light\_Import.drop)\ ?a\ (at\ (lift\ ?x))$

thm REALLIM_WITHIN_OPEN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (l::real)\ (a::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ a\ s \wedge HOL\_Light\_Import.open\ s \longrightarrow ---> f\ l\ (within\ (at\ a)\ s) = ---> f\ l\ (at\ a)$

thm LIM_WITHIN_REAL_OPEN:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (l::(real,\ ?'a::type)\ cart)\ (a::real)\ s::real \Rightarrow bool.\ IN\ a\ s \wedge real\_open\ s \longrightarrow --> f\ l\ (within\ (atreal\ a)\ s) = --> f\ l\ (atreal\ a)$

thm REALLIM_WITHIN_REAL_OPEN:

$\forall (f::real \Rightarrow real)\ (l::real)\ (a::real)\ s::real \Rightarrow bool.\ IN\ a\ s \wedge real\_open\ s \longrightarrow ---> f\ l\ (within\ (atreal\ a)\ s) = ---> f\ l\ (atreal\ a)$

thm REAL_ABEL_LIMIT_THEOREM:

$\forall (s::nat \Rightarrow bool)\ a::nat \Rightarrow real.\ real\_summable\ s\ a \longrightarrow (\forall r::real.\ |r| < (1::real) \longrightarrow real\_summable\ s\ (\lambda i::nat.\ a\ i * r^i)) \wedge ---> (\lambda r::real.\ real\_infsum\ s\ (\lambda i::nat.\ a\ i * r^i))\ (real\_infsum\ s\ a)\ (within\ (atreal\ (1::real))\ (GSPEC\ (\lambda GEN\%PVAR\%2410::real.\ \exists z::real.\ SETSPEC\ GEN\%PVAR\%2410\ (z \leq (1::real))\ z)))$

thm DEF_real_continuous:

$real\_continuous = (\lambda(\_1887956::?'a::type \Rightarrow real)\ \_1887957::?'a::type\ net.\ ---> \_1887956\ (\_1887956\ (netlimit\ \_1887957))\ \_1887957)$

thm real_continuous:

$\forall (f::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net = ---> f\ (f\ (netlimit\ net))\ net$

thm REAL_CONTINUOUS_TRIVIAL_LIMIT:

$\forall (f::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ trivial\_limit\ net \longrightarrow real\_continuous\ f\ net$

thm REAL_CONTINUOUS_WITHIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ real\_continuous\ f\ (within\ (at\ x)\ s) = ---> f\ (f\ x)\ (within\ (at\ x)\ s)$

thm REAL_CONTINUOUS_AT:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ x::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (at\ x) = ---> f\ (f\ x)\ (at\ x)$

thm REAL_CONTINUOUS_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ real\_continuous\ f\ (within\ (atreal\ x)\ s) = ---> f\ (f\ x)\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ATREAL:

$\forall (f::real \Rightarrow real)\ x::real.\ real\_continuous\ f\ (atreal\ x) = ---> f\ (f\ x)\ (atreal\ x)$

thm CONTINUOUS_WITHINREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (x::real)\ s::real \Rightarrow bool.\ continuous\ f\ (within\ (atreal\ x)\ s) = --> f\ (f\ x)\ (within\ (atreal\ x)\ s)$

thm CONTINUOUS_ATREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ x::real.\ continuous\ f\ (atreal\ x) = --> f\ (f\ x)\ (atreal\ x)$

thm real_continuous_within:

$real\_continuous\ (?f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (within\ (at\ (?x::(real,\ ?'a::type)\ cart))\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool)) = (\forall e>0::real.\ \exists d>0::real.\ \forall x'::(real,\ ?'a::type)\ cart.\ IN\ x'\ ?s \wedge distance\ (x',\ ?x) < d \longrightarrow |?f\ x' - ?f\ ?x| < e)$

thm real_continuous_at:

$real\_continuous\ (?f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (at\ (?x::(real,\ ?'a::type)\ cart)) = (\forall e>0::real.\ \exists d>0::real.\ \forall x'::(real,\ ?'a::type)\ cart.\ distance\ (x',\ ?x) < d \longrightarrow |?f\ x' - ?f\ ?x| < e)$

thm real_continuous_withinreal:

$real\_continuous\ (?f::real \Rightarrow real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = (\forall e>0::real.\ \exists d>0::real.\ \forall x'::real.\ IN\ x'\ ?s \wedge |x' - ?x| < d \longrightarrow |?f\ x' - ?f\ ?x| < e)$

thm real_continuous_atreal:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?x::real)) = (\forall e>0::real.\ \exists d>0::real.\ \forall x'::real.\ |x' - ?x| < d \longrightarrow |?f\ x' - ?f\ ?x| < e)$

thm REAL_CONTINUOUS_AT_WITHIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (at\ x) \longrightarrow real\_continuous\ f\ (within\ (at\ x)\ s)$

thm REAL_CONTINUOUS_ATREAL_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ x::real.\ real\_continuous\ f\ (atreal\ x) \longrightarrow real\_continuous\ f\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_WITHINREAL_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_continuous\ f\ (within\ (atreal\ (?x::real))\ s) \wedge SUBSET\ t\ s \longrightarrow real\_continuous\ f\ (within\ (atreal\ ?x)\ t)$

thm REAL_CONTINUOUS_WITHIN_SUBSET:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ real\_continuous\ f\ (within\ (at\ (?x::(real,\ ?'a::type)\ cart))\ s) \wedge SUBSET\ t\ s \longrightarrow real\_continuous\ f\ (within\ (at\ ?x)\ t)$

thm CONTINUOUS_WITHINREAL_SUBSET:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ continuous\ f\ (within\ (atreal\ (?x::real))\ s) \wedge SUBSET\ t\ s \longrightarrow continuous\ f\ (within\ (atreal\ ?x)\ t)$

thm continuous_withinreal:

$continuous\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = (\forall\ e>0::real.\ \exists\ d>0::real.\ \forall\ x'::real.\ IN\ x'\ ?s \wedge |x' - ?x| < d \longrightarrow distance\ (?f\ x',\ ?f\ ?x) < e)$

thm continuous_atreal:

$continuous\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (atreal\ (?x::real)) = (\forall\ e>0::real.\ \exists\ d>0::real.\ \forall\ x'::real.\ |x' - ?x| < d \longrightarrow distance\ (?f\ x',\ ?f\ ?x) < e)$

thm CONTINUOUS_ATREAL_WITHINREAL:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (x::real)\ s::real \Rightarrow bool.\ continuous\ f\ (atreal\ x) \longrightarrow continuous\ f\ (within\ (atreal\ x)\ s)$

thm CONTINUOUS_CX_ATREAL:

$\forall\ x::real.\ continuous\ Cx\ (atreal\ x)$

thm CONTINUOUS_CX_WITHINREAL:

$\forall (s::real \Rightarrow bool)\ x::real.\ continuous\ Cx\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_CONST:

$\forall (net::?'a::type\ net)\ c::real.\ real\_continuous\ (\lambda x::?'a::type.\ c)\ net$

thm REAL_CONTINUOUS_LMUL:

$\forall (f::?'a::type \Rightarrow real)\ (c::real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ c * f\ x)\ net$

thm REAL_CONTINUOUS_RMUL:

$\forall (f::?'a::type \Rightarrow real)\ (c::real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ f\ x * c)\ net$

thm REAL_CONTINUOUS_NEG:

$\forall (f::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ - f\ x)\ net$

thm REAL_CONTINUOUS_ADD:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ f\ x + g\ x)\ net$

thm REAL_CONTINUOUS_SUB:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ f\ x - g\ x)\ net$

thm REAL_CONTINUOUS_MUL:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ f\ x * g\ x)\ net$

thm REAL_CONTINUOUS_INV:

$\forall (net::?'a::type\ net)\ f::?'a::type \Rightarrow real.\ real\_continuous\ f\ net \wedge f\ (netlimit\ net) \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ inverse\_class.inverse\ (f\ x))\ net$

thm REAL_CONTINUOUS_DIV:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ g::?'a::type \Rightarrow real.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \wedge g\ (netlimit\ net) \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ f\ x\ /\ g\ x)\ net$

thm REAL_CONTINUOUS_POW:

$\forall (net::?'a::type\ net)\ (f::?'a::type \Rightarrow real)\ n::nat.\ real\_continuous\ f\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ (f\ x)^n)\ net$

thm REAL_CONTINUOUS_ABS:

$\forall (net::?'a::type\ net)\ f::?'a::type \Rightarrow real.\ real\_continuous\ f\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ |f\ x|)\ net$

thm REAL_CONTINUOUS_MAX:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ max\ (f\ x)\ (g\ x))\ net$

thm REAL_CONTINUOUS_MIN:

$\forall (f::?'a::type \Rightarrow real)\ (g::?'a::type \Rightarrow real)\ net::?'a::type\ net.\ real\_continuous\ f\ net \wedge real\_continuous\ g\ net \longrightarrow real\_continuous\ (\lambda x::?'a::type.\ min\ (f\ x)\ (g\ x))\ net$

thm REAL_CONTINUOUS_WITHIN_ID:

$\forall (x::real)\ s::real \Rightarrow bool.\ real\_continuous\ (\lambda x::real.\ x)\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_ID:

$\forall x::real.\ real\_continuous\ (\lambda x::real.\ x)\ (atreal\ x)$

thm REAL_CONTINUOUS_INV_WITHIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (within\ (at\ a)\ s) \wedge f\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ inverse\_class.inverse\ (f\ x))\ (within\ (at\ a)\ s)$

thm REAL_CONTINUOUS_INV_AT:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ a::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (at\ a) \wedge f\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ inverse\_class.inverse\ (f\ x))\ (at\ a)$

thm REAL_CONTINUOUS_INV_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ a::real.\ real\_continuous\ f\ (within\ (atreal\ a)\ s) \wedge f\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::real.\ inverse\_class.inverse\ (f\ x))\ (within\ (atreal\ a)\ s)$

thm REAL_CONTINUOUS_INV_ATREAL:

$\forall (f::real \Rightarrow real)\ a::real.\ real\_continuous\ f\ (atreal\ a) \wedge f\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::real.\ inverse\_class.inverse\ (f\ x))\ (atreal\ a)$

thm REAL_CONTINUOUS_DIV_WITHIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (within\ (at\ a)\ s) \wedge real\_continuous\ (?g::(real,\ ?'a::type)\ cart \Rightarrow real)\ (within\ (at\ a)\ s) \wedge ?g\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ f\ x\ /\ ?g\ x)\ (within\ (at\ a)\ s)$

thm REAL_CONTINUOUS_DIV_AT:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ a::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (at\ a) \wedge real\_continuous\ (?g::(real,\ ?'a::type)\ cart \Rightarrow real)\ (at\ a) \wedge ?g\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ f\ x\ /\ ?g\ x)\ (at\ a)$

thm REAL_CONTINUOUS_DIV_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ a::real.\ real\_continuous\ f\ (within\ (atreal\ a)\ s) \wedge real\_continuous\ (?g::real \Rightarrow real)\ (within\ (atreal\ a)\ s) \wedge ?g\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::real.\ f\ x\ /\ ?g\ x)\ (within\ (atreal\ a)\ s)$

thm REAL_CONTINUOUS_DIV_ATREAL:

$\forall (f::real \Rightarrow real)\ a::real.\ real\_continuous\ f\ (atreal\ a) \wedge real\_continuous\ (?g::real \Rightarrow real)\ (atreal\ a) \wedge ?g\ a \neq (0::real) \longrightarrow real\_continuous\ (\lambda x::real.\ f\ x\ /\ ?g\ x)\ (atreal\ a)$

thm REAL_CONTINUOUS_WITHINREAL_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ real\_continuous\ f\ (within\ (atreal\ x)\ s) \wedge real\_continuous\ g\ (within\ (atreal\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow real\_continuous\ (g \circ f)\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ATREAL_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ x::real.\ real\_continuous\, f\ (atreal\, x) \wedge real\_continuous\ g\ (atreal\ (f\, x)) \longrightarrow real\_continuous\ (g \circ f)\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_COMPOSE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (g::real \Rightarrow real)\ (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ real\_continuous\, f\ (within\ (at\, x)\ s) \wedge real\_continuous\ g\ (within\ (atreal\ (f\, x))\ (IMAGE\, f\, s)) \longrightarrow real\_continuous\ (g \circ f)\ (within\ (at\ x)\ s)$

thm REAL_CONTINUOUS_AT_COMPOSE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow real)\ (g::real \Rightarrow real)\ x::(real,\ ?'a::type)\ cart.\ real\_continuous\ f\ (at\ x) \wedge real\_continuous\ g\ (within\ (atreal\ (f\, x))\ (IMAGE\, f\ HOL\_Light\_Import.UNIV)) \longrightarrow real\_continuous\ (g \circ f)\ (at\ x)$

thm REAL_CONTINUOUS_CONTINUOUS_WITHIN_COMPOSE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ (x::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\ f\ (within\ (at\ x)\ s) \wedge real\_continuous\ g\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow real\_continuous\ (g \circ f)\ (within\ (at\ x)\ s)$

thm REAL_CONTINUOUS_CONTINUOUS_AT_COMPOSE:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ x::(real,\ ?'b::type)\ cart.\ continuous\ f\ (at\ x) \wedge real\_continuous\ g\ (within\ (at\ (f\ x))\ (IMAGE\ f\ HOL\_Light\_Import.UNIV)) \longrightarrow real\_continuous\ (g \circ f)\ (at\ x)$

thm REAL_CONTINUOUS_CONTINUOUS_WITHINREAL_COMPOSE:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ continuous\, f\ (within\ (atreal\ x)\ s) \wedge real\_continuous\ g\ (within\ (at\ (f\, x))\ (IMAGE\, f\, s)) \longrightarrow real\_continuous\ (g \circ f)\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_CONTINUOUS_ATREAL_COMPOSE:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (g::(real,\ ?'a::type)\ cart \Rightarrow real)\ x::real.\ continuous\, f\ (atreal\, x) \wedge real\_continuous\, g\ (within\ (at\ (f\, x))\ (IMAGE\, f\ HOL\_Light\_Import.UNIV)) \longrightarrow real\_continuous\ (g \circ f)\ (atreal\ x)$

thm CONTINUOUS_REAL_CONTINUOUS_WITHINREAL_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow (real,\ ?'a::type)\ cart)\ (x::real)\ s::real \Rightarrow bool.\ real\_continuous\ f\ (within\ (atreal\ x)\ s) \wedge continuous\ g\ (within\ (atreal\ (f\ x))\ (IMAGE\, f\, s)) \longrightarrow continuous\ (g \circ f)\ (within\ (atreal\ x)\ s)$

thm CONTINUOUS_REAL_CONTINUOUS_ATREAL_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow (real,\ ?'a::type)\ cart)\ x::real.\ real\_continuous\ f\ (atreal\, x) \wedge continuous\, g\ (within\ (atreal\ (f\, x))\ (IMAGE\, f\ HOL\_Light\_Import.UNIV)) \longrightarrow continuous\ (g \circ f)\ (atreal\ x)$

thm CONTINUOUS_WITHINREAL_COMPOSE:

$\forall (f::real \Rightarrow (real, ?'b::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::real)\ s::real \Rightarrow bool.\ continuous\ f\ (within\ (atreal\ x)\ s) \wedge continuous\ g\ (within\ (at\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow continuous\ (g \circ f)\ (within\ (atreal\ x)\ s)$

thm CONTINUOUS_ATREAL_COMPOSE:

$\forall (f::real \Rightarrow (real, ?'b::type)\ cart)\ (g::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ x::real.\ continuous\ f\ (atreal\ x) \wedge continuous\ g\ (within\ (at\ (f\ x))\ (IMAGE\ f\ HOL\_Light\_Import.UNIV)) \longrightarrow continuous\ (g \circ f)\ (atreal\ x)$

thm CONTINUOUS_REAL_CONTINUOUS_WITHIN_COMPOSE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow real)\ (g::real \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ real\_continuous\ f\ (within\ (at\ x)\ s) \wedge continuous\ g\ (within\ (atreal\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow continuous\ (g \circ f)\ (within\ (at\ x)\ s)$

thm CONTINUOUS_REAL_CONTINUOUS_AT_COMPOSE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow real)\ (g::real \Rightarrow (real, ?'a::type)\ cart)\ x::(real, ?'b::type)\ cart.\ real\_continuous\ f\ (at\ x) \wedge continuous\ g\ (within\ (atreal\ (f\ x))\ (IMAGE\ f\ HOL\_Light\_Import.UNIV)) \longrightarrow continuous\ (g \circ f)\ (at\ x)$

thm DEF_real_continuous_on:

$real\_continuous\_on = (\lambda(\_1889245::real \Rightarrow real)\ \_1889246::real \Rightarrow bool.\ \forall x::real.\ IN\ x\ \_1889246 \longrightarrow (\forall e>0::real.\ \exists d>0::real.\ \forall x'::real.\ IN\ x'\ \_1889246 \wedge |x' - x| < d \longrightarrow |\_1889245\ x' - \_1889245\ x| < e))$

thm real_continuous_on:

$\forall (s::real \Rightarrow bool)\ f::real \Rightarrow real.\ real\_continuous\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow (\forall e>0::real.\ \exists d>0::real.\ \forall x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow |f\ x' - f\ x| < e))$

thm REAL_CONTINUOUS_ON_EQ_CONTINUOUS_WITHIN:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (within\ (atreal\ x)\ s))$

thm REAL_CONTINUOUS_ON_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow real\_continuous\_on\ f\ t$

thm REAL_CONTINUOUS_ON_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge real\_continuous\_on\ g\ (IMAGE\ f\ s) \longrightarrow real\_continuous\_on\ (g \circ f)\ s$

thm REAL_CONTINUOUS_ON:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s = continuous\_on\ (lift \circ (f \circ HOL\_Light\_Import.drop))\ (IMAGE\ lift\ s)$

thm REAL_CONTINUOUS_ON_CONST:

$\forall (s::real \Rightarrow bool)\ c::real.\ real\_continuous\_on\ (\lambda x::real.\ c)\ s$

thm REAL_CONTINUOUS_ON_ID:

$\forall s::real \Rightarrow bool.\ real\_continuous\_on\ (\lambda x::real.\ x)\ s$

thm REAL_CONTINUOUS_ON_LMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm REAL_CONTINUOUS_ON_RMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm REAL_CONTINUOUS_ON_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ - f\ x)\ s$

thm REAL_CONTINUOUS_ON_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \land real\_continuous\_on\ g\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm REAL_CONTINUOUS_ON_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \land real\_continuous\_on\ g\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ f\ x - g\ x)\ s$

thm REAL_CONTINUOUS_ON_MUL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \land real\_continuous\_on\ g\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ f\ x * g\ x)\ s$

thm REAL_CONTINUOUS_ON_POW:

$\forall (f::real \Rightarrow real)\ (n::nat)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ (\lambda x::real.\ (f\ x)^n)\ s$

thm REAL_CONTINUOUS_ON_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \land real\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ g\ s$

thm REAL_CONTINUOUS_ON_UNION:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_closed\ s \land real\_closed\ t \land real\_continuous\_on\ f\ s \land real\_continuous\_on\ f\ t \longrightarrow real\_continuous\_on\ f\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_CONTINUOUS_ON_CASES:

$\forall (P::real \Rightarrow bool)\ (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_closed\ s \land real\_closed\ t \land real\_continuous\_on\ f\ s \land real\_continuous\_on$

$g\ t \wedge (\forall x::real.\ IN\ x\ s \wedge \neg\ P\ x \vee IN\ x\ t \wedge P\ x \longrightarrow f\ x = g\ x) \longrightarrow$ $real\_continuous\_on\ (\lambda x::real.\ if\ P\ x\ then\ f\ x\ else\ g\ x)\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_CONTINUOUS_ON_SUM:

$\forall (t::real \Rightarrow bool)\ (f::?'a::type \Rightarrow real \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow real\_continuous\_on\ (f\ a)\ t) \longrightarrow real\_continuous\_on\ (\lambda x::real.\ sum\ s\ (\lambda a::?'a::type.\ f\ a\ x))\ t$

thm REALLIM_CONTINUOUS_FUNCTION:

$\forall (f::real \Rightarrow (real,\ ?'b::type)\ cart)\ (net::?'a::type\ net)\ (g::?'a::type \Rightarrow real)\ l::real.\ continuous\ f\ (atreal\ l) \wedge ---> g\ l\ net \longrightarrow --> (\lambda x::?'a::type.\ f\ (g\ x))\ (f\ l)\ net$

thm LIM_REAL_CONTINUOUS_FUNCTION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow real)\ (net::?'a::type\ net)\ (g::?'a::type \Rightarrow (real,\ ?'b::type)\ cart)\ l::(real,\ ?'b::type)\ cart.\ real\_continuous\ f\ (at\ l) \wedge --> g\ l\ net \longrightarrow ---> (\lambda x::?'a::type.\ f\ (g\ x))\ (f\ l)\ net$

thm REALLIM_REAL_CONTINUOUS_FUNCTION:

$\forall (f::real \Rightarrow real)\ (net::?'a::type\ net)\ (g::?'a::type \Rightarrow real)\ l::real.\ real\_continuous\ f\ (atreal\ l) \wedge ---> g\ l\ net \longrightarrow ---> (\lambda x::?'a::type.\ f\ (g\ x))\ (f\ l)\ net$

thm REAL_CONTINUOUS_ON_EQ_REAL_CONTINUOUS_AT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_open\ s \longrightarrow real\_continuous\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (atreal\ x))$

thm REAL_CONTINUOUS_ATTAINS_SUP:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_compact\ s \wedge s \neq EMPTY \wedge real\_continuous\_on\ f\ s \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow f\ y \leq f\ x))$

thm REAL_CONTINUOUS_ATTAINS_INF:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_compact\ s \wedge s \neq EMPTY \wedge real\_continuous\_on\ f\ s \longrightarrow (\exists x::real.\ IN\ x\ s \wedge (\forall y::real.\ IN\ y\ s \longrightarrow f\ x \leq f\ y))$

thm DEF_real_uniformly_continuous_on:

$real\_uniformly\_continuous\_on = (\lambda(\_1889471::real \Rightarrow real)\ \_1889472::real \Rightarrow bool.\ \forall e>0::real.\ \exists d>0::real.\ \forall (x::real)\ x'::real.\ IN\ x\ \_1889472 \wedge IN\ x'\ \_1889472 \wedge |x' - x| < d \longrightarrow |\_1889471\ x' - \_1889471\ x| < e)$

thm real_uniformly_continuous_on:

$\forall (s::real \Rightarrow bool)\ f::real \Rightarrow real.\ real\_uniformly\_continuous\_on\ f\ s = (\forall e>0::real.\ \exists d>0::real.\ \forall (x::real)\ x'::real.\ IN\ x\ s \wedge IN\ x'\ s \wedge |x' - x| < d \longrightarrow |f\ x' - f\ x| < e)$

thm REAL_UNIFORMLY_CONTINUOUS_ON:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s = uniformly\_continuous\_on\ (lift \circ (f \circ HOL\_Light\_Import.drop))\ (IMAGE\ lift\ s)$

thm REAL_UNIFORMLY_CONTINUOUS_IMP_REAL_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \longrightarrow real\_continuous\_on\ f\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_SEQUENTIALLY:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s = (\forall (x::nat \Rightarrow real)\ y::nat \Rightarrow real.\ (\forall n::nat.\ IN\ (x\ n)\ s) \wedge (\forall n::nat.\ IN\ (y\ n)\ s) \wedge ---> (\lambda n::nat.\ x\ n - y\ n)\ (0::real)\ sequentially \longrightarrow ---> (\lambda n::nat.\ f\ (x\ n) - f\ (y\ n))\ (0::real)\ sequentially)$

thm REAL_UNIFORMLY_CONTINUOUS_ON_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow real\_uniformly\_continuous\_on\ f\ t$

thm REAL_UNIFORMLY_CONTINUOUS_ON_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \wedge real\_uniformly\_continuous\_on\ g\ (IMAGE\ f\ s) \longrightarrow real\_uniformly\_continuous\_on\ (g \circ f)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_CONST:

$\forall (s::real \Rightarrow bool)\ c::real.\ real\_uniformly\_continuous\_on\ (\lambda x::real.\ c)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_LMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_RMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_ID:

$\forall s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ (\lambda x::real.\ x)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ - f\ x)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\ s \wedge real\_uniformly\_continuous\_on\ g\ s \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_uniformly\_continuous\_on\ f\,s \wedge real\_uniformly\_continuous\_on\ g\,s \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ f\,x - g\,x)\ s$

thm REAL_UNIFORMLY_CONTINUOUS_ON_SUM:

$\forall (t::real \Rightarrow bool)\ (f::?'a::type \Rightarrow real \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow real\_uniformly\_continuous\_on\ (f\ a)\ t) \longrightarrow real\_uniformly\_continuous\_on\ (\lambda x::real.\ sum\ s\ (\lambda a::?'a::type.\ f\ a\ x))\ t$

thm REAL_COMPACT_UNIFORMLY_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge real\_compact\ s \longrightarrow real\_uniformly\_continuous\_on\ f\ s$

thm REAL_COMPACT_CONTINUOUS_IMAGE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge real\_compact\ s \longrightarrow real\_compact\ (IMAGE\ f\ s)$

thm REAL_DINI:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_compact\ s \wedge (\forall n::nat.\ real\_continuous\_on\ (f\ n)\ s) \wedge real\_continuous\_on\ g\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow --->\ (\lambda n::nat.\ f\ n\ x)\ (g\ x)\ sequentially) \wedge (\forall (n::nat)\ x::real.\ IN\ x\ s \longrightarrow f\ n\ x \leq f\ (n + (1::nat))\ x) \longrightarrow (\forall e>0::real.\ eventually\ (\lambda n::nat.\ \forall x::real.\ IN\ x\ s \longrightarrow |f\ n\ x - g\ x| < e)\ sequentially)$

thm CONTINUOUS_COMPONENTWISE:

$\forall (net::?'b::type\ net)\ f::?'b::type \Rightarrow (real,\ ?'a::type)\ cart.\ continuous\ f\ net = (\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow real\_continuous\ (\lambda x::?'b::type.\ \$\ (f\ x)\ i)\ net)$

thm REAL_CONTINUOUS_COMPLEX_COMPONENTS_AT:

$\forall z::(real,\ 2)\ cart.\ real\_continuous\ Re\ (at\ z) \wedge real\_continuous\ Im\ (at\ z)$

thm REAL_CONTINUOUS_COMPLEX_COMPONENTS_WITHIN:

$\forall (s::(real,\ 2)\ cart \Rightarrow bool)\ z::(real,\ 2)\ cart.\ real\_continuous\ Re\ (within\ (at\ z)\ s) \wedge real\_continuous\ Im\ (within\ (at\ z)\ s)$

thm REAL_CONTINUOUS_NORM_AT:

$\forall z::(real,\ ?'a::type)\ cart.\ real\_continuous\ vector\_norm\ (at\ z)$

thm REAL_CONTINUOUS_NORM_WITHIN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ z::(real,\ ?'a::type)\ cart.\ real\_continuous\ vector\_norm\ (within\ (at\ z)\ s)$

thm REAL_CONTINUOUS_DIST_AT:

$\forall (a::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ distance\ (a,\ x))\ (at\ z)$

thm REAL_CONTINUOUS_DIST_WITHIN:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ z::(real,\ ?'a::type)\ cart.\ real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ distance\ (a,\ x))\ (within\ (at\ z)\ s)$

thm DEF_has_real_derivative:

$has\_real\_derivative = (\lambda(\_1889561::real \Rightarrow real)\ (\_1889562::real)\ \_1889563::real\ net.\ --->\ (\lambda x::real.\ inverse\_class.inverse\ (x - netlimit\ \_1889563) * (\_1889561\ x - (\_1889561\ (netlimit\ \_1889563) + \_1889562 * (x - netlimit\ \_1889563))))\ (0::real)\ \_1889563)$

thm has_real_derivative:

$\forall (f::real \Rightarrow real)\ (f'::real)\ net::real\ net.\ has\_real\_derivative\ f\ f'\ net = --->\ (\lambda x::real.\ inverse\_class.inverse\ (x - netlimit\ net) * (f\ x - (f\ (netlimit\ net) + f' * (x - netlimit\ net))))\ (0::real)\ net$

thm DEF_real_differentiable:

$real\_differentiable = (\lambda(\_1889582::real \Rightarrow real)\ \_1889583::real\ net.\ \exists f'::real.\ has\_real\_derivative\ \_1889582\ f'\ \_1889583)$

thm real_differentiable:

$\forall (f::real \Rightarrow real)\ net::real\ net.\ real\_differentiable\ f\ net = (\exists f'::real.\ has\_real\_derivative\ f\ f'\ net)$

thm DEF_real_derivative:

$real\_derivative = (\lambda(\_1889594::real \Rightarrow real)\ \_1889595::real.\ SOME\ f'::real.\ has\_real\_derivative\ \_1889594\ f'\ (atreal\ \_1889595))$

thm real_derivative:

$\forall (f::real \Rightarrow real)\ x::real.\ real\_derivative\ f\ x = (SOME\ f'::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x))$

thm DEF_higher_real_derivative:

$higher\_real\_derivative = (SOME\ higher\_real\_derivative::nat \Rightarrow nat \Rightarrow (real \Rightarrow real) \Rightarrow real \Rightarrow real.\ \forall \_1889990::nat.\ (\forall f::real \Rightarrow real.\ higher\_real\_derivative\ \_1889990\ (0::nat)\ f = f) \wedge (\forall (f::real \Rightarrow real)\ n::nat.\ higher\_real\_derivative\ \_1889990\ (Suc\ n)\ f = real\_derivative\ (higher\_real\_derivative\ \_1889990\ n\ f)))\ (57::nat)$

thm higher_real_derivative:

$higher\_real\_derivative\ (0::nat)\ (?f::real \Rightarrow real) = ?f \wedge (\forall n::nat.\ higher\_real\_derivative\ (Suc\ n)\ ?f = real\_derivative\ (higher\_real\_derivative\ n\ ?f))$

thm DEF_real_differentiable_on:

$real\_differentiable\_on = (\lambda(\_1889991::real \Rightarrow real)\ \_1889992::real \Rightarrow bool.\ \forall x::real.\ IN\ x\ \_1889992 \longrightarrow (\exists f'::real.\ has\_real\_derivative\ \_1889991\ f'\ (within\ (atreal\ x)\ \_1889992)))$

thm real_differentiable_on:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow (\exists f'::real.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s)))$

thm HAS_REAL_DERIVATIVE_WITHINREAL:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (within\ (atreal\ (?a::real))\ (?s::real \Rightarrow bool)) = ---> (\lambda x::real.\ (?f\ x - ?f\ ?a)\ /\ (x - ?a))\ ?f'\ (within\ (atreal\ ?a)\ ?s)$

thm HAS_REAL_DERIVATIVE_ATREAL:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (atreal\ (?a::real)) = ---> (\lambda x::real.\ (?f\ x - ?f\ ?a)\ /\ (x - ?a))\ ?f'\ (atreal\ ?a)$

thm HAS_REAL_FRECHET_DERIVATIVE_WITHIN:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = has\_derivative\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (\%\ ?f')\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm HAS_REAL_FRECHET_DERIVATIVE_AT:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (atreal\ (?x::real)) = has\_derivative\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (\%\ ?f')\ (at\ (lift\ ?x))$

thm HAS_REAL_VECTOR_DERIVATIVE_WITHIN:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = has\_vector\_derivative\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (lift\ ?f')\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm HAS_REAL_VECTOR_DERIVATIVE_AT:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (atreal\ (?x::real)) = has\_vector\_derivative\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (lift\ ?f')\ (at\ (lift\ ?x))$

thm HAS_REAL_COMPLEX_DERIVATIVE_WITHIN:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (within\ (atreal\ (?a::real))\ (?s::real \Rightarrow bool)) = has\_complex\_derivative\ (Cx \circ (?f \circ Re))\ (Cx\ ?f')\ (within\ (at\ (Cx\ ?a))\ (GSPEC\ (\lambda GEN\%PVAR\%2411::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2411\ (HOL\_Light\_Import.real\ z \wedge IN\ (Re\ z)\ ?s)\ z)))$

thm HAS_REAL_COMPLEX_DERIVATIVE_AT:

$has\_real\_derivative\ (?f::real \Rightarrow real)\ (?f'::real)\ (atreal\ (?a::real)) = has\_complex\_derivative\ (Cx \circ (?f \circ Re))\ (Cx\ ?f')\ (within\ (at\ (Cx\ ?a))\ HOL\_Light\_Import.real)$

thm REAL_DIFFERENTIABLE_ON_DIFFERENTIABLE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow real\_differentiable\ f\ (within\ (atreal\ x)\ s))$

thm REAL_DIFFERENTIABLE_ON_REAL_OPEN:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_open\ s \longrightarrow real\_differentiable\_on\ f\ s = (\forall x::real.\ IN\ x\ s \longrightarrow (\exists f'::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x)))$

thm REAL_DIFFERENTIABLE_ON_IMP_DIFFERENTIABLE_WITHIN:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\_on\ f\ s \wedge IN\ x\ s \longrightarrow real\_differentiable\ f\ (within\ (atreal\ x)\ s)$

thm REAL_DIFFERENTIABLE_ON_IMP_DIFFERENTIABLE_ATREAL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\_on\ f\ s \wedge real\_open\ s \wedge IN\ x\ s \longrightarrow real\_differentiable\ f\ (atreal\ x)$

thm HAS_COMPLEX_REAL_DERIVATIVE_WITHIN_GEN:

$\forall (f::real \Rightarrow real)\ (g::real)\ (h::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ (s::real \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ (?x::real)\ s \wedge has\_complex\_derivative\ h\ (Cx\ g)\ (within\ (at\ (Cx\ ?x))\ (GSPEC\ (\lambda GEN\%PVAR\%2412::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2412\ (HOL\_Light\_Import.real\ z \wedge IN\ (Re\ z)\ s)\ z))) \wedge (\forall y::real.\ IN\ y\ s \wedge |y - ?x| < d \longrightarrow h\ (Cx\ y) = Cx\ (f\ y)) \longrightarrow has\_real\_derivative\ f\ g\ (within\ (atreal\ ?x)\ s)$

thm HAS_COMPLEX_REAL_DERIVATIVE_AT_GEN:

$\forall (f::real \Rightarrow real)\ (g::real)\ (h::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ d::real.\ (0::real) < d \wedge has\_complex\_derivative\ h\ (Cx\ g)\ (within\ (at\ (Cx\ (?x::real)))\ HOL\_Light\_Import.real) \wedge (\forall y::real.\ |y - ?x| < d \longrightarrow h\ (Cx\ y) = Cx\ (f\ y)) \longrightarrow has\_real\_derivative\ f\ g\ (atreal\ ?x)$

thm HAS_COMPLEX_REAL_DERIVATIVE_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real)\ (h::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ s::real \Rightarrow bool.\ IN\ (?x::real)\ s \wedge has\_complex\_derivative\ h\ (Cx\ g)\ (within\ (at\ (Cx\ ?x))\ (GSPEC\ (\lambda GEN\%PVAR\%2414::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2414\ (HOL\_Light\_Import.real\ z \wedge IN\ (Re\ z)\ s)\ z))) \wedge (\forall y::real.\ IN\ y\ s \longrightarrow h\ (Cx\ y) = Cx\ (f\ y)) \longrightarrow has\_real\_derivative\ f\ g\ (within\ (atreal\ ?x)\ s)$

thm HAS_COMPLEX_REAL_DERIVATIVE_AT:

$\forall (f::real \Rightarrow real)\ (g::real)\ h::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ has\_complex\_derivative\ h\ (Cx\ g)\ (within\ (at\ (Cx\ (?x::real)))\ HOL\_Light\_Import.real) \wedge (\forall y::real.\ h\ (Cx\ y) = Cx\ (f\ y)) \longrightarrow has\_real\_derivative\ f\ g\ (atreal\ ?x)$

thm DEF_is_realinterval:

$is\_realinterval = (\lambda\_1890054::real \Rightarrow bool.\ \forall (a::real)\ (b::real)\ c::real.\ IN\ a\ \_1890054 \wedge IN\ b\ \_1890054 \wedge a \leq c \wedge c \leq b \longrightarrow IN\ c\ \_1890054)$

thm is_realinterval:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = (\forall (a::real)\ (b::real)\ c::real.\ IN\ a\ s \wedge IN\ b\ s \wedge a \leq c \wedge c \leq b \longrightarrow IN\ c\ s)$

thm IS_REALINTERVAL_IS_INTERVAL:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = is\_interval\ (IMAGE\ lift\ s)$

thm IS_REALINTERVAL_CONVEX:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = convex\ (IMAGE\ lift\ s)$

thm IS_REALINTERVAL_CONNECTED:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = connected\ (IMAGE\ lift\ s)$

thm TRIVIAL_LIMIT_WITHIN_REALINTERVAL:

$\forall (s::real \Rightarrow bool)\ x::real.\ is\_realinterval\ s \wedge IN\ x\ s \longrightarrow trivial\_limit\ (within\ (atreal\ x)\ s) = (s = INSERT\ x\ EMPTY)$

thm IS_REALINTERVAL_EMPTY:

$is\_realinterval\ EMPTY$

thm IS_REALINTERVAL_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ is\_realinterval\ s \wedge is\_realinterval\ t \wedge HOL\_Light\_Import.INTER\ s\ t \neq EMPTY \longrightarrow is\_realinterval\ (HOL\_Light\_Import.UNION\ s\ t)$

thm IS_REALINTERVAL_UNIV:

$is\_realinterval\ HOL\_Light\_Import.UNIV$

thm IS_REAL_INTERVAL_CASES:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = (s = EMPTY \vee s = HOL\_Light\_Import.UNIV \vee (\exists a::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2416::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2416\ (a < x)\ x)) \vee (\exists a::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2417::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2417\ (a \leq x)\ x)) \vee (\exists b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2418::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2418\ (x \leq b)\ x)) \vee (\exists b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2419::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2419\ (x < b)\ x)) \vee (\exists (a::real)\ b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2420::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2420\ (a < x \wedge x < b)\ x)) \vee (\exists (a::real)\ b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2421::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2421\ (a < x \wedge x \leq b)\ x)) \vee (\exists (a::real)\ b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2422::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2422\ (a \leq x \wedge x < b)\ x)) \vee (\exists (a::real)\ b::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%2423::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2423\ (a \leq x \wedge x \leq b)\ x)))$

thm IMAGE_LIFT_DROP_conjunct1:

$\forall s::real \Rightarrow bool.\ IMAGE\ (HOL\_Light\_Import.drop \circ lift)\ s = s$

thm IMAGE_LIFT_DROP_conjunct0:

$\forall s::(real,\ unit)\ cart \Rightarrow bool.\ IMAGE\ (lift \circ HOL\_Light\_Import.drop)\ s = s$

thm IS_REALINTERVAL_CONVEX_COMPLEX:

$\forall s::real \Rightarrow bool.\ is\_realinterval\ s = convex\ (GSPEC\ (\lambda GEN\%PVAR\%2424::(real,\ 2)\ cart.\ \exists z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2424\ (HOL\_Light\_Import.real\ z \wedge IN\ (Re\ z)\ s)\ z))$

thm DEF_open_real_interval:

$open\_real\_interval = (\lambda\_1890284::real \times real.\ GSPEC\ (\lambda GEN\%PVAR\%2425::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2425\ (fst\ \_1890284 < x \wedge x < snd\ \_1890284)\ x))$

thm open_real_interval:

$\forall (a::real)\ b::real.\ open\_real\_interval\ (a,\ b) = GSPEC\ (\lambda GEN\%PVAR\%2425::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2425\ (a < x \wedge x < b)\ x)$

thm DEF_closed_real_interval:

$closed\_real\_interval = (SOME\ closed\_real\_interval::nat \Rightarrow (real \times real)\ list \Rightarrow real \Rightarrow bool.\ \forall (\_1890403::nat)\ (a::real)\ b::real.\ closed\_real\_interval\ \_1890403\ [(a,\ b)] = GSPEC\ (\lambda GEN\%PVAR\%2426::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2426\ (a \leq x \wedge x \leq b)\ x))\ (58::nat)$

thm closed_real_interval:

$closed\_real\_interval\ [(?a::real,\ ?b::real)] = GSPEC\ (\lambda GEN\%PVAR\%2426::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2426\ (?a \leq x \wedge x \leq ?b)\ x)$

thm real_interval:

$open\_real\_interval\ (?a::real,\ ?b::real) = GSPEC\ (\lambda GEN\%PVAR\%2427::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2427\ (?a < x \wedge x < ?b)\ x) \wedge closed\_real\_interval\ [(?a,\ ?b)] = GSPEC\ (\lambda GEN\%PVAR\%2428::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2428\ (?a \leq x \wedge x \leq ?b)\ x)$

thm real_interval_conjunct1:

$closed\_real\_interval\ [(?a::real,\ ?b::real)] = GSPEC\ (\lambda GEN\%PVAR\%2428::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2428\ (?a \leq x \wedge x \leq ?b)\ x)$

thm real_interval_conjunct0:

$open\_real\_interval\ (?a::real,\ ?b::real) = GSPEC\ (\lambda GEN\%PVAR\%2427::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2427\ (?a < x \wedge x < ?b)\ x)$

thm IN_REAL_INTERVAL:

$\forall (a::real)\ (b::real)\ x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) = (a \leq x \wedge x \leq b) \wedge IN\ x\ (open\_real\_interval\ (a,\ b)) = (a < x \wedge x < b)$

thm REAL_INTERVAL_INTERVAL:

$closed\_real\_interval\ [(?a::real,\ ?b::real)] = IMAGE\ HOL\_Light\_Import.drop\ (closed\_interval\ [(lift\ ?a,\ lift\ ?b)]) \wedge open\_real\_interval\ (?a,\ ?b) = IMAGE\ HOL\_Light\_Import.drop\ (open\_interval\ (lift\ ?a,\ lift\ ?b))$

thm INTERVAL_REAL_INTERVAL:

$closed\_interval\ [(?a::(real,\ unit)\ cart,\ ?b::(real,\ unit)\ cart)] = IMAGE\ lift\ (closed\_real\_interval\ [(HOL\_Light\_Import.drop\ ?a,\ HOL\_Light\_Import.drop\ ?b)]) \wedge open\_interval$

(?a, ?b) = IMAGE lift (open_real_interval (HOL_Light_Import.drop ?a, HOL_Light_Import.drop ?b))

thm REAL_INTERVAL_INTERVAL_conjunct1:

open_real_interval (?a::real, ?b::real) = IMAGE HOL_Light_Import.drop (open_interval (lift ?a, lift ?b))

thm REAL_INTERVAL_INTERVAL_conjunct0:

closed_real_interval [(?a::real, ?b::real)] = IMAGE HOL_Light_Import.drop (closed_interval [(lift ?a, lift ?b)])

thm EMPTY_AS_REAL_INTERVAL:

EMPTY = closed_real_interval [(1::real, 0::real)]

thm IMAGE_LIFT_REAL_INTERVAL:

IMAGE lift (closed_real_interval [(?a::real, ?b::real)]) = closed_interval [(lift ?a, lift ?b)] ∧ IMAGE lift (open_real_interval (?a, ?b)) = open_interval (lift ?a, lift ?b)

thm INTERVAL_REAL_INTERVAL_conjunct1:

open_interval (?a::(real, unit) cart, ?b::(real, unit) cart) = IMAGE lift (open_real_interval (HOL_Light_Import.drop ?a, HOL_Light_Import.drop ?b))

thm INTERVAL_REAL_INTERVAL_conjunct0:

closed_interval [(?a::(real, unit) cart, ?b::(real, unit) cart)] = IMAGE lift (closed_real_interval [(HOL_Light_Import.drop ?a, HOL_Light_Import.drop ?b)])

thm IMAGE_DROP_INTERVAL:

IMAGE HOL_Light_Import.drop (closed_interval [(?a::(real, unit) cart, ?b::(real, unit) cart)]) = closed_real_interval [(HOL_Light_Import.drop ?a, HOL_Light_Import.drop ?b)] ∧ IMAGE HOL_Light_Import.drop (open_interval (?a, ?b)) = open_real_interval (HOL_Light_Import.drop ?a, HOL_Light_Import.drop ?b)

thm SUBSET_REAL_INTERVAL:

∀ (*a::real*) (*b::real*) (*c::real*) *d::real*. *SUBSET (closed_real_interval [(a, b)]) (closed_real_interval [(c, d)])* = ($b < a \lor c \le a \land a \le b \land b \le d$) ∧ *SUBSET (closed_real_interval [(a, b)]) (open_real_interval (c, d))* = ($b < a \lor c < a \land a \le b \land b < d$) ∧ *SUBSET (open_real_interval (a, b)) (closed_real_interval [(c, d)])* = ($b \le a \lor c \le a \land a < b \land b \le d$) ∧ *SUBSET (open_real_interval (a, b)) (open_real_interval (c, d))* = ($b \le a \lor c \le a \land a < b \land b \le d$)

thm REAL_INTERVAL_OPEN_SUBSET_CLOSED:

∀ (*a::real*) *b::real*. *SUBSET (open_real_interval (a, b)) (closed_real_interval [(a, b)])*

thm REAL_INTERVAL_EQ_EMPTY:

$(\forall (a::real)\ b::real.\ (closed\_real\_interval\ [(a,\ b)] = EMPTY) = (b < a)) \wedge (\forall (a::real)\ b::real.\ (open\_real\_interval\ (a,\ b) = EMPTY) = (b \leq a))$

thm REAL_INTERVAL_EQ_EMPTY_conjunct1:

$\forall (a::real)\ b::real.\ (open\_real\_interval\ (a,\ b) = EMPTY) = (b \leq a)$

thm REAL_INTERVAL_EQ_EMPTY_conjunct0:

$\forall (a::real)\ b::real.\ (closed\_real\_interval\ [(a,\ b)] = EMPTY) = (b < a)$

thm REAL_INTERVAL_NE_EMPTY:

$(\forall (a::real)\ b::real.\ (closed\_real\_interval\ [(a,\ b)] \neq EMPTY) = (a \leq b)) \wedge (\forall (a::real)\ b::real.\ (open\_real\_interval\ (a,\ b) \neq EMPTY) = (a < b))$

thm REAL_OPEN_CLOSED_INTERVAL:

$\forall (a::real)\ b::real.\ open\_real\_interval\ (a,\ b) = DIFF\ (closed\_real\_interval\ [(a,\ b)])\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm REAL_CLOSED_OPEN_INTERVAL:

$\forall (a::real)\ b::real.\ a \leq b \longrightarrow closed\_real\_interval\ [(a,\ b)] = HOL\_Light\_Import.UNION\ (open\_real\_interval\ (a,\ b))\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm IMAGE_LIFT_REAL_INTERVAL_conjunct1:

$IMAGE\ lift\ (open\_real\_interval\ (?a::real,\ ?b::real)) = open\_interval\ (lift\ ?a,\ lift\ ?b)$

thm IMAGE_LIFT_REAL_INTERVAL_conjunct0:

$IMAGE\ lift\ (closed\_real\_interval\ [(?a::real,\ ?b::real)]) = closed\_interval\ [(lift\ ?a,\ lift\ ?b)]$

thm REAL_CLOSED_REAL_INTERVAL:

$\forall (a::real)\ b::real.\ real\_closed\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_OPEN_REAL_INTERVAL:

$\forall (a::real)\ b::real.\ real\_open\ (open\_real\_interval\ (a,\ b))$

thm REAL_INTERVAL_SING:

$\forall\ a::real.\ closed\_real\_interval\ [(a,\ a)] = INSERT\ a\ EMPTY \wedge open\_real\_interval\ (a,\ a) = EMPTY$

thm REAL_COMPACT_INTERVAL:

$\forall (a::real)\ b::real.\ real\_compact\ (closed\_real\_interval\ [(a,\ b)])$

thm IS_REALINTERVAL_INTERVAL:

$\forall (a::real)\ b::real.\ is\_realinterval\ (open\_real\_interval\ (a,\ b)) \wedge is\_realinterval\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_BOUNDED_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ real\_bounded\ (closed\_real\_interval\ [(a,\ b)])) \land (\forall (a::real)\ b::real.\ real\_bounded\ (open\_real\_interval\ (a,\ b)))$

thm ENDS_IN_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ IN\ a\ (closed\_real\_interval\ [(a,\ b)]) = (closed\_real\_interval\ [(a,\ b)] \neq EMPTY)) \land (\forall (a::real)\ b::real.\ IN\ b\ (closed\_real\_interval\ [(a,\ b)]) = (closed\_real\_interval\ [(a,\ b)] \neq EMPTY)) \land (\forall (a::real)\ b::real.\ \neg\ IN\ a\ (open\_real\_interval\ (a,\ b))) \land (\forall (a::real)\ b::real.\ \neg\ IN\ b\ (open\_real\_interval\ (a,\ b)))$

thm IMAGE_AFFINITY_REAL_INTERVAL:

$\forall (a::real)\ (b::real)\ (m::real)\ c::real.\ IMAGE\ (\lambda x::real.\ m * x + c)\ (closed\_real\_interval\ [(a,\ b)]) = (if\ closed\_real\_interval\ [(a,\ b)] = EMPTY\ then\ EMPTY\ else\ if\ (0::real) \leq m\ then\ closed\_real\_interval\ [(m * a + c,\ m * b + c)]\ else\ closed\_real\_interval\ [(m * b + c,\ m * a + c)])$

thm IMAGE_STRETCH_REAL_INTERVAL:

$\forall (a::real)\ (b::real)\ m::real.\ IMAGE\ (op * m)\ (closed\_real\_interval\ [(a,\ b)]) = (if\ closed\_real\_interval\ [(a,\ b)] = EMPTY\ then\ EMPTY\ else\ if\ (0::real) \leq m\ then\ closed\_real\_interval\ [(m * a,\ m * b)]\ else\ closed\_real\_interval\ [(m * b,\ m * a)])$

thm REAL_INTERVAL_TRANSLATION_conjunct1:

$\forall (c::real)\ (a::real)\ b::real.\ open\_real\_interval\ (c + a,\ c + b) = IMAGE\ (op + c)\ (open\_real\_interval\ (a,\ b))$

thm REAL_INTERVAL_TRANSLATION_conjunct0:

$\forall (c::real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(c + a,\ c + b)] = IMAGE\ (op + c)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTERVAL_TRANSLATION:

$(\forall (c::real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(c + a,\ c + b)] = IMAGE\ (op + c)\ (closed\_real\_interval\ [(a,\ b)])) \land (\forall (c::real)\ (a::real)\ b::real.\ open\_real\_interval\ (c + a,\ c + b) = IMAGE\ (op + c)\ (open\_real\_interval\ (a,\ b)))$

thm IN_REAL_INTERVAL_REFLECT:

$(\forall (a::real)\ (b::real)\ x::real.\ IN\ (-\ x)\ (closed\_real\_interval\ [(-\ b,\ -\ a)]) = IN\ x\ (closed\_real\_interval\ [(a,\ b)])) \land (\forall (a::real)\ (b::real)\ x::real.\ IN\ (-\ x)\ (open\_real\_interval\ (-\ b,\ -\ a)) = IN\ x\ (open\_real\_interval\ (a,\ b)))$

thm REFLECT_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ IMAGE\ uminus\ (closed\_real\_interval\ [(a,\ b)]) = closed\_real\_interval\ [(-\ b,\ -\ a)]) \land (\forall (a::real)\ b::real.\ IMAGE\ uminus\ (open\_real\_interval\ (a,\ b)) = open\_real\_interval\ (-\ b,\ -\ a))$

thm REAL_CONTINUOUS_CONTINUOUS:

$real\_continuous\ (?f::?'a::type \Rightarrow real)\ (?net::?'a::type\ net) = continuous\ (Cx \circ ?f)\ ?net$

thm REAL_CONTINUOUS_CONTINUOUS1:

$real\_continuous\ (?f::?'a::type \Rightarrow real)\ (?net::?'a::type\ net) = continuous\ (lift \circ ?f)\ ?net$

thm REAL_CONTINUOUS_CONTINUOUS_ATREAL:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?x::real)) = continuous\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (at\ (lift\ ?x))$

thm REAL_CONTINUOUS_CONTINUOUS_WITHINREAL:

$real\_continuous\ (?f::real \Rightarrow real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = continuous\ (lift \circ (?f \circ HOL\_Light\_Import.drop))\ (within\ (at\ (lift\ ?x))\ (IMAGE\ lift\ ?s))$

thm REAL_COMPLEX_CONTINUOUS_WITHINREAL:

$real\_continuous\ (?f::real \Rightarrow real)\ (within\ (atreal\ (?x::real))\ (?s::real \Rightarrow bool)) = continuous\ (Cx \circ (?f \circ Re))\ (within\ (at\ (Cx\ ?x))\ (HOL\_Light\_Import.INTER\ HOL\_Light\_Import.real\ (IMAGE\ Cx\ ?s)))$

thm REAL_COMPLEX_CONTINUOUS_ATREAL:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?x::real)) = continuous\ (Cx \circ (?f \circ Re))\ (within\ (at\ (Cx\ ?x))\ HOL\_Light\_Import.real)$

thm HAS_REAL_DERIVATIVE_IMP_CONTINUOUS_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow real\_continuous\ f\ (within\ (atreal\ x)\ s)$

thm REAL_DIFFERENTIABLE_IMP_CONTINUOUS_WITHINREAL:

$\forall (f::real \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ x)\ s) \longrightarrow real\_continuous\ f\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_IMP_CONTINUOUS_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow real\_continuous\ f\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_IMP_CONTINUOUS_ATREAL:

$\forall (f::real \Rightarrow real)\ x::real.\ real\_differentiable\ f\ (atreal\ x) \longrightarrow real\_continuous\ f\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_ON_IMP_REAL_CONTINUOUS_ON:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \longrightarrow real\_continuous\_on\ f\ s$

thm REAL_CONTINUOUS_AT_COMPONENT:

$\forall (i::nat)\ a::(real,\ ?'a::type)\ cart.\ (1::nat) \le i \wedge i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ \$\ x\ i)\ (at\ a)$

thm REAL_CONTINUOUS_AT_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (z::(real,\ ?'a::type)\ cart)\ f::(real,\ ?'a::type)\ cart \Rightarrow real.\ real\_continuous\ f\ (at\ (vector\_add\ a\ z)) = real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ f\ (vector\_add\ a\ x))\ (at\ z)$

thm REAL_CONTINUOUS_AT_LINEAR_IMAGE:

$\forall (h::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (z::(real,\ ?'a::type)\ cart)\ f::(real,\ ?'a::type)\ cart \Rightarrow real.\ linear\ h \wedge (\forall x::(real,\ ?'a::type)\ cart.\ vector\_norm\ (h\ x) = vector\_norm\ x) \longrightarrow real\_continuous\ f\ (at\ (h\ z)) = real\_continuous\ (\lambda x::(real,\ ?'a::type)\ cart.\ f\ (h\ x))\ (at\ z)$

thm REAL_CONTINUOUS_AT_ARG:

$\forall z::(real,\ 2)\ cart.\ \neg\ (HOL\_Light\_Import.real\ z \wedge (0::real) \le Re\ z) \longrightarrow real\_continuous\ Arg\ (at\ z)$

thm HAS_REAL_DERIVATIVE_WITHIN_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ x::real.\ has\_real\_derivative\ f\ (?f'::real)\ (within\ (atreal\ x)\ s) \wedge SUBSET\ t\ s \longrightarrow has\_real\_derivative\ f\ ?f'\ (within\ (atreal\ x)\ t)$

thm REAL_DIFFERENTIABLE_ON_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow real\_differentiable\_on\ f\ t$

thm REAL_DIFFERENTIABLE_WITHIN_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ (?x::real))\ s) \wedge SUBSET\ t\ s \longrightarrow real\_differentiable\ f\ (within\ (atreal\ ?x)\ t)$

thm HAS_REAL_DERIVATIVE_ATREAL_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_WITHIN_REAL_OPEN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (a::real)\ s::real \Rightarrow bool.\ IN\ a\ s \wedge real\_open\ s \longrightarrow has\_real\_derivative\ f\ f'\ (within\ (atreal\ a)\ s) = has\_real\_derivative\ f\ f'\ (atreal\ a)$

thm REAL_DIFFERENTIABLE_ATREAL_WITHIN:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ z::real.\ real\_differentiable\ f\ (atreal\ z) \longrightarrow real\_differentiable\ f\ (within\ (atreal\ z)\ s)$

thm HAS_REAL_DERIVATIVE_TRANSFORM_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (x::real)\ (s::real \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ x\ s \wedge (\forall x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow f\ x' = g\ x') \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ g\ f'\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_TRANSFORM_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (x::real)\ d::real.\ (0::real) < d \wedge (\forall x'::real.\ |x' - x| < d \longrightarrow f\ x' = g\ x') \wedge has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ g\ f'\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_ZERO_CONSTANT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (0::real)\ (within\ (atreal\ x)\ s)) \longrightarrow (\exists c::real.\ \forall x::real.\ IN\ x\ s \longrightarrow f\ x = c)$

thm HAS_REAL_DERIVATIVE_ZERO_UNIQUE:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (c::real)\ a::real.\ is\_realinterval\ s \wedge IN\ a\ s \wedge f\ a = c \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (0::real)\ (within\ (atreal\ x)\ s)) \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = c)$

thm REAL_DIFF_CHAIN_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (f'::real)\ (g'::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \wedge has\_real\_derivative\ g\ g'\ (within\ (atreal\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow has\_real\_derivative\ (g \circ f)\ (g' * f')\ (within\ (atreal\ x)\ s)$

thm REAL_DIFF_CHAIN_ATREAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (f'::real)\ (g'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \wedge has\_real\_derivative\ g\ g'\ (atreal\ (f\ x)) \longrightarrow has\_real\_derivative\ (g \circ f)\ (g' * f')\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_CHAIN:

$\forall (P::real \Rightarrow bool)\ (f::real \Rightarrow real)\ g::real \Rightarrow real.\ (\forall x::real.\ P\ x \longrightarrow has\_real\_derivative\ g\ ((?g'::real \Rightarrow real)\ x)\ (atreal\ x)) \longrightarrow (\forall (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ (?f'::real)\ (within\ (atreal\ x)\ s) \wedge P\ (f\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ g\ (f\ x))\ (?f' * ?g'\ (f\ x))\ (within\ (atreal\ x)\ s)) \wedge (\forall x::real.\ has\_real\_derivative\ f\ ?f'\ (atreal\ x) \wedge P\ (f\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ g\ (f\ x))\ (?f' * ?g'\ (f\ x))\ (atreal\ x))$

thm HAS_REAL_DERIVATIVE_CHAIN_UNIV:

$\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ (\forall x::real.\ has\_real\_derivative\ g\ ((?g'::real \Rightarrow real)\ x)\ (atreal\ x)) \longrightarrow (\forall (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ (?f'::real)\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ g\ (f\ x))\ (?f' * ?g'\ (f\ x))\ (within\ (atreal\ x)\ s)) \wedge (\forall x::real.\ has\_real\_derivative\ f\ ?f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ g\ (f\ x))\ (?f' * ?g'\ (f\ x))\ (atreal\ x))$

thm REAL_DERIVATIVE_UNIQUE_ATREAL:

$\forall (f::real \Rightarrow real)\ (z::real)\ (f'::real)\ f''::real.\ has\_real\_derivative\ f\ f'\ (atreal\ z) \wedge has\_real\_derivative\ f\ f''\ (atreal\ z) \longrightarrow f' = f''$

thm HAS_REAL_DERIVATIVE_DERIVATIVE:

$\forall (f::real \Rightarrow real)\ (f'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow real\_derivative\ f\ x = f'$

thm HAS_REAL_DERIVATIVE_DIFFERENTIABLE:

$\forall (f::real \Rightarrow real)\ x::real.\ has\_real\_derivative\ f\ (real\_derivative\ f\ x)\ (atreal\ x) = real\_differentiable\ f\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_LMUL_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ c * f\ x)\ (c * f')\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_LMUL_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ c * f\ x)\ (c * f')\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_RMUL_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x * c)\ (f' * c)\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_RMUL_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x * c)\ (f' * c)\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_CDIV_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x\ /\ c)\ (f'\ /\ c)\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_CDIV_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (c::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x\ /\ c)\ (f'\ /\ c)\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_ID:

$\forall net::real\ net.\ has\_real\_derivative\ (\lambda x::real.\ x)\ (1::real)\ net$

thm HAS_REAL_DERIVATIVE_CONST:

$\forall (c::real)\ net::real\ net.\ has\_real\_derivative\ (\lambda x::real.\ c)\ (0::real)\ net$

thm HAS_REAL_DERIVATIVE_NEG:

$\forall (f::real \Rightarrow real)\ (f'::real)\ net::real\ net.\ has\_real\_derivative\ f\ f'\ net \longrightarrow has\_real\_derivative\ (\lambda x::real.\ -\ f\ x)\ (-\ f')\ net$

thm HAS_REAL_DERIVATIVE_ADD:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ net::real\ net.\ has\_real\_derivative\ f\ f'\ net \wedge has\_real\_derivative\ g\ g'\ net \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x + g\ x)\ (f' + g')\ net$

thm HAS_REAL_DERIVATIVE_SUB:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ net::real\ net.\ has\_real\_derivative\ f\ f'\ net \wedge has\_real\_derivative\ g\ g'\ net \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x - g\ x)\ (f' - g')\ net$

thm HAS_REAL_DERIVATIVE_MUL_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \wedge has\_real\_derivative\ g\ g'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x * g\ x)\ (f\ x * g' + f' * g\ x)\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_MUL_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \wedge has\_real\_derivative\ g\ g'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x * g\ x)\ (f\ x * g' + f' * g\ x)\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_POW_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ (s::real \Rightarrow bool)\ n::nat.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ (f\ x)^n)\ (real\_of\_nat\ n * ((f\ x)^{n - (1::nat)} * f'))\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_POW_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ n::nat.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ (f\ x)^n)\ (real\_of\_nat\ n * ((f\ x)^{n - (1::nat)} * f'))\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_INV_BASIC:

$\forall x::real.\ x \neq (0::real) \longrightarrow has\_real\_derivative\ inverse\_class.inverse\ (-\ inverse\_class.inverse\ (x^2))\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_INV_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \wedge f\ x \neq (0::real) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ inverse\_class.inverse\ (f\ x))\ (-\ f' / (f\ x)^2)\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_INV_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \wedge f\ x \neq (0::real) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ inverse\_class.inverse\ (f\ x))\ (-\ f' / (f\ x)^2)\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_DIV_WITHIN:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ (x::real)\ s::real \Rightarrow bool.$ $has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \wedge has\_real\_derivative\ g\ g'\ (within\ (atreal\ x)\ s) \wedge g\ x \neq (0::real) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x\ /\ g\ x)\ ((f' * g\ x - f\ x * g')\ /\ (g\ x)^2)\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_DIV_ATREAL:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (g::real \Rightarrow real)\ (g'::real)\ x::real.\ has\_real\_derivative\ f\ f'\ (atreal\ x) \wedge has\_real\_derivative\ g\ g'\ (atreal\ x) \wedge g\ x \neq (0::real) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ f\ x\ /\ g\ x)\ ((f' * g\ x - f\ x * g')\ /\ (g\ x)^2)\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (net::real\ net)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall a::?'a::type.\ IN\ a\ s \longrightarrow has\_real\_derivative\ (f\ a)\ ((?f'::?'a::type \Rightarrow real)\ a)\ net) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ sum\ s\ (\lambda a::?'a::type.\ f\ a\ x))\ (sum\ s\ ?f')\ net$

thm REAL_DIFFERENTIABLE_CONST:

$\forall (c::real)\ net::real\ net.\ real\_differentiable\ (\lambda z::real.\ c)\ net$

thm REAL_DIFFERENTIABLE_ID:

$\forall net::real\ net.\ real\_differentiable\ (\lambda z::real.\ z)\ net$

thm REAL_DIFFERENTIABLE_NEG:

$\forall (f::real \Rightarrow real)\ net::real\ net.\ real\_differentiable\ f\ net \longrightarrow real\_differentiable\ (\lambda z::real.\ - f\ z)\ net$

thm REAL_DIFFERENTIABLE_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ net::real\ net.\ real\_differentiable\ f\ net \wedge real\_differentiable\ g\ net \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z + g\ z)\ net$

thm REAL_DIFFERENTIABLE_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ net::real\ net.\ real\_differentiable\ f\ net \wedge real\_differentiable\ g\ net \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z - g\ z)\ net$

thm REAL_DIFFERENTIABLE_INV_WITHIN:

$\forall (f::real \Rightarrow real)\ (z::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ z)\ s) \wedge f\ z \neq (0::real) \longrightarrow real\_differentiable\ (\lambda z::real.\ inverse\_class.inverse\ (f\ z))\ (within\ (atreal\ z)\ s)$

thm REAL_DIFFERENTIABLE_MUL_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (z::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ z)\ s) \wedge real\_differentiable\ g\ (within\ (atreal\ z)\ s) \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z * g\ z)\ (within\ (atreal\ z)\ s)$

thm REAL_DIFFERENTIABLE_DIV_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (z::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ z)\ s) \wedge real\_differentiable\ g\ (within\ (atreal\ z)\ s) \wedge g\ z \neq (0::real) \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z\ /\ g\ z)\ (within\ (atreal\ z)\ s)$

thm REAL_DIFFERENTIABLE_POW_WITHIN:

$\forall (f::real \Rightarrow real)\ (n::nat)\ (z::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ z)\ s) \longrightarrow real\_differentiable\ (\lambda z::real.\ (f\ z)^n)\ (within\ (atreal\ z)\ s)$

thm REAL_DIFFERENTIABLE_TRANSFORM_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (x::real)\ (s::real \Rightarrow bool)\ d::real.\ (0::real) < d \wedge IN\ x\ s \wedge (\forall x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow f\ x' = g\ x') \wedge real\_differentiable\ f\ (within\ (atreal\ x)\ s) \longrightarrow real\_differentiable\ g\ (within\ (atreal\ x)\ s)$

thm REAL_DIFFERENTIABLE_TRANSFORM:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge real\_differentiable\_on\ f\ s \longrightarrow real\_differentiable\_on\ g\ s$

thm REAL_DIFFERENTIABLE_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow real\_differentiable\_on\ f\ s = real\_differentiable\_on\ g\ s$

thm REAL_DIFFERENTIABLE_INV_ATREAL:

$\forall (f::real \Rightarrow real)\ z::real.\ real\_differentiable\ f\ (atreal\ z) \wedge f\ z \neq (0::real) \longrightarrow real\_differentiable\ (\lambda z::real.\ inverse\_class.inverse\ (f\ z))\ (atreal\ z)$

thm REAL_DIFFERENTIABLE_MUL_ATREAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ z::real.\ real\_differentiable\ f\ (atreal\ z) \wedge real\_differentiable\ g\ (atreal\ z) \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z * g\ z)\ (atreal\ z)$

thm REAL_DIFFERENTIABLE_DIV_ATREAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ z::real.\ real\_differentiable\ f\ (atreal\ z) \wedge real\_differentiable\ g\ (atreal\ z) \wedge g\ z \neq (0::real) \longrightarrow real\_differentiable\ (\lambda z::real.\ f\ z\ /\ g\ z)\ (atreal\ z)$

thm REAL_DIFFERENTIABLE_POW_ATREAL:

$\forall (f::real \Rightarrow real)\ (n::nat)\ z::real.\ real\_differentiable\ f\ (atreal\ z) \longrightarrow real\_differentiable\ (\lambda z::real.\ (f\ z)^n)\ (atreal\ z)$

thm REAL_DIFFERENTIABLE_TRANSFORM_ATREAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (x::real)\ d::real.\ (0::real) < d \wedge (\forall x'::real.\ |x' - x| < d \longrightarrow f\ x' = g\ x') \wedge real\_differentiable\ f\ (atreal\ x) \longrightarrow real\_differentiable\ g\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_COMPOSE_WITHIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ real\_differentiable\ f\ (within\ (atreal\ x)\ s) \wedge real\_differentiable\ g\ (within\ (atreal\ (f\ x))\ (IMAGE\ f\ s)) \longrightarrow real\_differentiable\ (g \circ f)\ (within\ (atreal\ x)\ s)$

thm REAL_DIFFERENTIABLE_COMPOSE_ATREAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ x::real.\ real\_differentiable\ f\ (atreal\ x) \wedge real\_differentiable\ g\ (atreal\ (f\ x)) \longrightarrow real\_differentiable\ (g \circ f)\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_ON_CONST:

$\forall (c::real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ (\lambda z::real.\ c)\ s$

thm REAL_DIFFERENTIABLE_ON_ID:

$\forall s::real \Rightarrow bool.\ real\_differentiable\_on\ (\lambda z::real.\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_COMPOSE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge real\_differentiable\_on\ g\ (IMAGE\ f\ s) \longrightarrow real\_differentiable\_on\ (g \circ f)\ s$

thm REAL_DIFFERENTIABLE_ON_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ -\ f\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge real\_differentiable\_on\ g\ s \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ f\ z + g\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge real\_differentiable\_on\ g\ s \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ f\ z - g\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_MUL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge real\_differentiable\_on\ g\ s \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ f\ z * g\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_INV:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge (\forall z::real.\ IN\ z\ s \longrightarrow f\ z \neq (0::real)) \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ inverse\_class.inverse\ (f\ z))\ s$

thm REAL_DIFFERENTIABLE_ON_DIV:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_differentiable\_on\ f\ s \wedge real\_differentiable\_on\ g\ s \wedge (\forall z::real.\ IN\ z\ s \longrightarrow g\ z \neq (0::real)) \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ f\ z\ /\ g\ z)\ s$

thm REAL_DIFFERENTIABLE_ON_POW:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ n::nat.\ real\_differentiable\_on\ f\ s \longrightarrow real\_differentiable\_on\ (\lambda z::real.\ (f\ z)^n)\ s$

thm REAL_DIFFERENTIABLE_ON_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (s::real \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge (\forall\ a::?'a::type.\ IN\ a\ k \longrightarrow real\_differentiable\_on\ (f\ a)\ s) \longrightarrow real\_differentiable\_on\ (\lambda x::real.\ sum\ k\ (\lambda a::?'a::type.\ f\ a\ x))\ s$

thm HAS_REAL_DERIVATIVE_EXP:

$\forall\ x::real.\ has\_real\_derivative\ exp\ (exp\ x)\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_EXP:

$\forall\ x::real.\ real\_differentiable\ exp\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_EXP:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\ exp\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_EXP:

$\forall\ x::real.\ real\_continuous\ exp\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_EXP:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_continuous\ exp\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_EXP:

$\forall\ s::real \Rightarrow bool.\ real\_continuous\_on\ exp\ s$

thm HAS_REAL_DERIVATIVE_SIN:

$\forall\ x::real.\ has\_real\_derivative\ sin\ (cos\ x)\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_SIN:

$\forall\ x::real.\ real\_differentiable\ sin\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_SIN:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\ sin\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_SIN:

$\forall\ x::real.\ real\_continuous\ sin\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_SIN:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_continuous\ sin\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_SIN:

$\forall\ s::real \Rightarrow bool.\ real\_continuous\_on\ sin\ s$

thm HAS_REAL_DERIVATIVE_COS:

$\forall\ x::real.\ has\_real\_derivative\ cos\ (-\ sin\ x)\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_COS:

$\forall\ x::real.\ real\_differentiable\ cos\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_COS:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\ cos\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_COS:

$\forall x::real.\ real\_continuous\ cos\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_COS:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_continuous\ cos\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_COS:

$\forall s::real \Rightarrow bool.\ real\_continuous\_on\ cos\ s$

thm HAS_REAL_DERIVATIVE_TAN:

$\forall x::real.\ cos\ x \neq (0::real) \longrightarrow has\_real\_derivative\ tan\ (inverse\_class.inverse\ ((cos\ x)^2))\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_TAN:

$\forall x::real.\ cos\ x \neq (0::real) \longrightarrow real\_differentiable\ tan\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_TAN:

$\forall (s::real \Rightarrow bool)\ x::real.\ cos\ x \neq (0::real) \longrightarrow real\_differentiable\ tan\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_TAN:

$\forall x::real.\ cos\ x \neq (0::real) \longrightarrow real\_continuous\ tan\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_TAN:

$\forall (s::real \Rightarrow bool)\ x::real.\ cos\ x \neq (0::real) \longrightarrow real\_continuous\ tan\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_TAN:

$\forall s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow cos\ x \neq (0::real)) \longrightarrow real\_continuous\_on\ tan\ s$

thm HAS_REAL_DERIVATIVE_LOG:

$\forall x>0::real.\ has\_real\_derivative\ log\ (inverse\_class.inverse\ x)\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_LOG:

$\forall x>0::real.\ real\_differentiable\ log\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_LOG:

$\forall (s::real \Rightarrow bool)\ x::real.\ (0::real) < x \longrightarrow real\_differentiable\ log\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_LOG:

$\forall x>0::real.\ real\_continuous\ log\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_LOG:

$\forall (s::real \Rightarrow bool)\ x::real.\ (0::real) < x \longrightarrow real\_continuous\ log\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_LOG:

$\forall s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow (0::real) < x) \longrightarrow real\_continuous\_on\ log\ s$

thm HAS_REAL_DERIVATIVE_SQRT:

$\forall x>0::real.\ has\_real\_derivative\ sqrt\ (inverse\_class.inverse\ (real\_of\_nat\ (2::nat) * sqrt\ x))\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_SQRT:

$\forall x>0::real.\ real\_differentiable\ sqrt\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_SQRT:

$\forall (s::real \Rightarrow bool)\ x::real.\ (0::real) < x \longrightarrow real\_differentiable\ sqrt\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_SQRT:

$\forall x>0::real.\ real\_continuous\ sqrt\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_SQRT:

$\forall (s::real \Rightarrow bool)\ x::real.\ (0::real) < x \longrightarrow real\_continuous\ sqrt\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_ATN:

$\forall x::real.\ has\_real\_derivative\ atn\ (inverse\_class.inverse\ ((1::real) + x^2))\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_ATN:

$\forall x::real.\ real\_differentiable\ atn\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_ATN:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_differentiable\ atn\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_ATN:

$\forall x::real.\ real\_continuous\ atn\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_ATN:

$\forall (s::real \Rightarrow bool)\ x::real.\ real\_continuous\ atn\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_ON_ATN:

$\forall s::real \Rightarrow bool.\ real\_continuous\_on\ atn\ s$

thm HAS_REAL_DERIVATIVE_ASN_COS:

$\forall x::real.\ |x| < (1::real) \longrightarrow has\_real\_derivative\ asn\ (inverse\_class.inverse\ (cos\ (asn\ x)))\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_ASN:

$\forall x::real.\ |x| < (1::real) \longrightarrow has\_real\_derivative\ asn\ (inverse\_class.inverse\ (sqrt\ ((1::real) - x^2)))\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_ASN:

$\forall x::real.\ |x| < (1::real) \longrightarrow real\_differentiable\ asn\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_ASN:

$\forall (s::real \Rightarrow bool)\ x::real.\ |x| < (1::real) \longrightarrow real\_differentiable\ asn\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_ASN:

$\forall x::real.\ |x| < (1::real) \longrightarrow real\_continuous\ asn\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_ASN:

$\forall (s::real \Rightarrow bool)\ x::real.\ |x| < (1::real) \longrightarrow real\_continuous\ asn\ (within\ (atreal\ x)\ s)$

thm HAS_REAL_DERIVATIVE_ACS_SIN:

$\forall x::real.\ |x| < (1::real) \longrightarrow has\_real\_derivative\ acs\ (-\ inverse\_class.inverse\ (sin\ (acs\ x)))\ (atreal\ x)$

thm HAS_REAL_DERIVATIVE_ACS:

$\forall x::real.\ |x| < (1::real) \longrightarrow has\_real\_derivative\ acs\ (-\ inverse\_class.inverse\ (sqrt\ ((1::real) - x^2)))\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_AT_ACS:

$\forall x::real.\ |x| < (1::real) \longrightarrow real\_differentiable\ acs\ (atreal\ x)$

thm REAL_DIFFERENTIABLE_WITHIN_ACS:

$\forall (s::real \Rightarrow bool)\ x::real.\ |x| < (1::real) \longrightarrow real\_differentiable\ acs\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_AT_ACS:

$\forall x::real.\ |x| < (1::real) \longrightarrow real\_continuous\ acs\ (atreal\ x)$

thm REAL_CONTINUOUS_WITHIN_ACS:

$\forall (s::real \Rightarrow bool)\ x::real.\ |x| < (1::real) \longrightarrow real\_continuous\ acs\ (within\ (atreal\ x)\ s)$

thm REAL_CONTINUOUS_WITHIN_SQRT_STRONG:

$\forall x::real.\ real\_continuous\ sqrt\ (within\ (atreal\ x)\ (GSPEC\ (\lambda GEN\%PVAR\%2435::real.\ \exists t::real.\ SETSPEC\ GEN\%PVAR\%2435\ ((0::real) \leq t)\ t)))$

thm REAL_CONTINUOUS_ON_SQRT:

$\forall$ *s::real* $\Rightarrow$ *bool*. ($\forall$ *x::real*. *IN x s* $\longrightarrow$ (*0::real*) $\leq$ *x*) $\longrightarrow$ *real_continuous_on sqrt s*

thm REAL_CONTINUOUS_WITHIN_ASN_STRONG:

$\forall$ *x::real*. *real_continuous asn* (*within* (*atreal x*) (*GSPEC* (λ*GEN%PVAR%2441::real*. $\exists$ *t::real*. *SETSPEC GEN%PVAR%2441* (|*t*| $\leq$ (*1::real*)) *t*)))

thm REAL_CONTINUOUS_ON_ASN:

$\forall$ *s::real* $\Rightarrow$ *bool*. ($\forall$ *x::real*. *IN x s* $\longrightarrow$ |*x*| $\leq$ (*1::real*)) $\longrightarrow$ *real_continuous_on asn s*

thm REAL_CONTINUOUS_WITHIN_ACS_STRONG:

$\forall$ *x::real*. *real_continuous acs* (*within* (*atreal x*) (*GSPEC* (λ*GEN%PVAR%2447::real*. $\exists$ *t::real*. *SETSPEC GEN%PVAR%2447* (|*t*| $\leq$ (*1::real*)) *t*)))

thm REAL_CONTINUOUS_ON_ACS:

$\forall$ *s::real* $\Rightarrow$ *bool*. ($\forall$ *x::real*. *IN x s* $\longrightarrow$ |*x*| $\leq$ (*1::real*)) $\longrightarrow$ *real_continuous_on acs s*

thm HAS_REAL_DERIVATIVE_RPOW:

$\forall$ (*x::real*) *y::real*. (*0::real*) < *x* $\longrightarrow$ *has_real_derivative* (λ*x::real*. *rpow x y*) (*y* $*$ *rpow x* (*y* $-$ (*1::real*))) (*atreal x*)

thm REAL_DIFFERENTIABLE_AT_RPOW:

$\forall$ (*x::real*) *y::real*. *x* $\neq$ (*0::real*) $\longrightarrow$ *real_differentiable* (λ*x::real*. *rpow x y*) (*atreal x*)

thm REAL_CONTINUOUS_AT_RPOW:

$\forall$ (*x::real*) *y::real*. (*x* = (*0::real*) $\longrightarrow$ (*0::real*) $\leq$ *y*) $\longrightarrow$ *real_continuous* (λ*x::real*. *rpow x y*) (*atreal x*)

thm REAL_CONTINUOUS_WITHIN_RPOW:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*x::real*) *y::real*. (*x* = (*0::real*) $\longrightarrow$ (*0::real*) $\leq$ *y*) $\longrightarrow$ *real_continuous* (λ*x::real*. *rpow x y*) (*within* (*atreal x*) *s*)

thm REAL_CONTINUOUS_ON_RPOW:

$\forall$ (*s::real* $\Rightarrow$ *bool*) *y::real*. (*IN* (*0::real*) *s* $\longrightarrow$ (*0::real*) $\leq$ *y*) $\longrightarrow$ *real_continuous_on* (λ*x::real*. *rpow x y*) *s*

thm REALLIM_RPOW:

$\forall$ (*net::?'a::type net*) (*f::?'a::type* $\Rightarrow$ *real*) (*l::real*) *n::real*. ---> *f l net* $\wedge$ (*l* = (*0::real*) $\longrightarrow$ (*0::real*) $\leq$ *n*) $\longrightarrow$ ---> (λ*x::?'a::type*. *rpow* (*f x*) *n*) (*rpow l n*) *net*

thm REALLIM_NULL_POW_EQ:

$\forall$ (*net*::?'*a*::*type net*) (*f*::?'*a*::*type* $\Rightarrow$ *real*) *n*::*nat*. *n* $\neq$ (*0*::*nat*) $\longrightarrow$ −−−> (λ*x*::?'*a*::*type*. (*f x*)^n) (*0*::*real*) *net* = −−−> *f* (*0*::*real*) *net*

thm LIM_NULL_COMPLEX_POW_EQ:

$\forall$ (*net*::?'*a*::*type net*) (*f*::?'*a*::*type* $\Rightarrow$ (*real*, *2*) *cart*) *n*::*nat*. *n* $\neq$ (*0*::*nat*) $\longrightarrow$ −−> (λ*x*::?'*a*::*type*. *complex_pow* (*f x*) *n*) (*Cx* (*0*::*real*)) *net* = −−> *f* (*Cx* (*0*::*real*)) *net*

thm REAL_IVT_INCREASING:

$\forall$ (*f*::*real* $\Rightarrow$ *real*) (*a*::*real*) (*b*::*real*) *y*::*real*. *a* $\leq$ *b* $\wedge$ *real_continuous_on f* (*closed_real_interval* [(*a*, *b*)]) $\wedge$ *f a* $\leq$ *y* $\wedge$ *y* $\leq$ *f b* $\longrightarrow$ ($\exists$ *x*::*real*. *IN x* (*closed_real_interval* [(*a*, *b*)]) $\wedge$ *f x* = *y*)

thm REAL_IVT_DECREASING:

$\forall$ (*f*::*real* $\Rightarrow$ *real*) (*a*::*real*) (*b*::*real*) *y*::*real*. *a* $\leq$ *b* $\wedge$ *real_continuous_on f* (*closed_real_interval* [(*a*, *b*)]) $\wedge$ *f b* $\leq$ *y* $\wedge$ *y* $\leq$ *f a* $\longrightarrow$ ($\exists$ *x*::*real*. *IN x* (*closed_real_interval* [(*a*, *b*)]) $\wedge$ *f x* = *y*)

thm IS_REALINTERVAL_CONTINUOUS_IMAGE:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_continuous_on* (?*f*::*real* $\Rightarrow$ *real*) *s* $\wedge$ *is_realinterval s* $\longrightarrow$ *is_realinterval* (*IMAGE* ?*f s*)

thm REAL_INTERVAL_NE_EMPTY_conjunct1:

$\forall$ (*a*::*real*) *b*::*real*. (*open_real_interval* (*a*, *b*) $\neq$ *EMPTY*) = (*a* < *b*)

thm REAL_INTERVAL_NE_EMPTY_conjunct0:

$\forall$ (*a*::*real*) *b*::*real*. (*closed_real_interval* [(*a*, *b*)] $\neq$ *EMPTY*) = (*a* $\leq$ *b*)

thm ENDS_IN_REAL_INTERVAL_conjunct3:

$\forall$ (*a*::*real*) *b*::*real*. $\neg$ *IN b* (*open_real_interval* (*a*, *b*))

thm ENDS_IN_REAL_INTERVAL_conjunct2:

$\forall$ (*a*::*real*) *b*::*real*. $\neg$ *IN a* (*open_real_interval* (*a*, *b*))

thm ENDS_IN_REAL_INTERVAL_conjunct1:

$\forall$ (*a*::*real*) *b*::*real*. *IN b* (*closed_real_interval* [(*a*, *b*)]) = (*closed_real_interval* [(*a*, *b*)] $\neq$ *EMPTY*)

thm ENDS_IN_REAL_INTERVAL_conjunct0:

$\forall$ (*a*::*real*) *b*::*real*. *IN a* (*closed_real_interval* [(*a*, *b*)]) = (*closed_real_interval* [(*a*, *b*)] $\neq$ *EMPTY*)

thm REAL_DERIVATIVE_POS_LEFT_MINIMUM:

$\forall$ (*f*::*real* $\Rightarrow$ *real*) (*f'*::*real*) (*a*::*real*) (*b*::*real*) *e*::*real*. *a* < *b* $\wedge$ (*0*::*real*) < *e* $\wedge$ *has_real_derivative f f'* (*within* (*atreal a*) (*closed_real_interval* [(*a*, *b*)])) $\wedge$

$(\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge |x - a| < e \longrightarrow f\ a \leq f\ x) \longrightarrow (0::real) \leq f'$

thm REAL_DERIVATIVE_NEG_LEFT_MAXIMUM:

$\forall\, (f::real \Rightarrow real)\ (f'::real)\ (a::real)\ (b::real)\ e::real.\ a < b \wedge (0::real) < e \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ a)\ (closed\_real\_interval\ [(a,\ b)])) \wedge (\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge |x - a| < e \longrightarrow f\ x \leq f\ a) \longrightarrow f' \leq (0::real)$

thm REAL_DERIVATIVE_POS_RIGHT_MAXIMUM:

$\forall\, (f::real \Rightarrow real)\ (f'::real)\ (a::real)\ (b::real)\ e::real.\ a < b \wedge (0::real) < e \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ b)\ (closed\_real\_interval\ [(a,\ b)])) \wedge (\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge |x - b| < e \longrightarrow f\ x \leq f\ b) \longrightarrow (0::real) \leq f'$

thm REAL_DERIVATIVE_NEG_RIGHT_MINIMUM:

$\forall\, (f::real \Rightarrow real)\ (f'::real)\ (a::real)\ (b::real)\ e::real.\ a < b \wedge (0::real) < e \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ b)\ (closed\_real\_interval\ [(a,\ b)])) \wedge (\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge |x - b| < e \longrightarrow f\ b \leq f\ x) \longrightarrow f' \leq (0::real)$

thm REAL_DERIVATIVE_ZERO_MAXMIN:

$\forall\, (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ IN\ x\ s \wedge real\_open\ s \wedge has\_real\_derivative\ f\ f'\ (atreal\ x) \wedge ((\forall\, y::real.\ IN\ y\ s \longrightarrow f\ y \leq f\ x) \vee (\forall\, y::real.\ IN\ y\ s \longrightarrow f\ x \leq f\ y)) \longrightarrow f' = (0::real)$

thm REAL_ROLLE:

$\forall\, (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a < b \wedge f\ a = f\ b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall\, x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x)) \longrightarrow (\exists\, x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \wedge f'\ x = (0::real))$

thm REAL_MVT:

$\forall\, (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a < b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall\, x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x)) \longrightarrow (\exists\, x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \wedge f\ b - f\ a = f'\ x * (b - a))$

thm REAL_MVT_SIMPLE:

$\forall\, (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a < b \wedge (\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\exists\, x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \wedge f\ b - f\ a = f'\ x * (b - a))$

thm REAL_MVT_VERY_SIMPLE:

$\forall\, (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a \leq b \wedge (\forall\, x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal$

$x)$ $(closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\exists\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge f\ b - f\ a = f'\ x * (b - a))$

thm REAL_ROLLE_SIMPLE:

$\forall (f{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (a{::}real)\ b{::}real.\ a < b \wedge f\ a = f\ b \wedge (\forall\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\exists\, x{::}real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \wedge f'\ x = (0{::}real))$

thm REAL_MVT_CAUCHY:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (g'{::}real \Rightarrow real)\ (a{::}real)\ b{::}real.\ a < b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_continuous\_on\ g\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall\, x{::}real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x) \wedge has\_real\_derivative\ g\ (g'\ x)\ (atreal\ x)) \longrightarrow (\exists\, x{::}real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \wedge (f\ b - f\ a) * g'\ x = (g\ b - g\ a) * f'\ x)$

thm LHOSPITAL:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (g'{::}real \Rightarrow real)\ (c{::}real)\ (l{::}real)\ d{::}real.\ (0{::}real) < d \wedge (\forall\, x{::}real.\ (0{::}real) < |x - c| \wedge |x - c| < d \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x) \wedge has\_real\_derivative\ g\ (g'\ x)\ (atreal\ x) \wedge g'\ x \neq (0{::}real)) \wedge ---> f\ (0{::}real)\ (atreal\ c) \wedge ---> g\ (0{::}real)\ (atreal\ c) \wedge ---> (\lambda x{::}real.\ f'\ x\ /\ g'\ x)\ l\ (atreal\ c) \longrightarrow ---> (\lambda x{::}real.\ f\ x\ /\ g\ x)\ l\ (atreal\ c)$

thm REAL_DERIVATIVE_IVT_INCREASING:

$\forall (f{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (a{::}real)\ b{::}real.\ a \leq b \wedge (\forall\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\forall\, t{::}real.\ f'\ a \leq t \wedge t \leq f'\ b \longrightarrow (\exists\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge f'\ x = t))$

thm REAL_DERIVATIVE_IVT_DECREASING:

$\forall (f{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (a{::}real)\ (b{::}real)\ t{::}?'a{::}type.\ a \leq b \wedge (\forall\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\forall\, t{::}real.\ f'\ b \leq t \wedge t \leq f'\ a \longrightarrow (\exists\, x{::}real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge f'\ x = t))$

thm REAL_CONTINUOUS_ON_INVERSE:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ s{::}real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge real\_compact\ s \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \longrightarrow real\_continuous\_on\ g\ (IMAGE\ f\ s)$

thm HAS_REAL_DERIVATIVE_INVERSE_BASIC:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ (f'{::}real)\ (t{::}real \Rightarrow bool)\ y{::}real.\ has\_real\_derivative\ f\ f'\ (atreal\ (g\ y)) \wedge f' \neq (0{::}real) \wedge real\_continuous\ g\ (atreal\ y) \wedge real\_open$

t ∧ IN y t ∧ (∀z::real. IN z t ⟶ f (g z) = z) ⟶ has_real_derivative g (inverse_class.inverse f′) (atreal y)

thm HAS_REAL_DERIVATIVE_INVERSE_STRONG:

∀(f::real ⇒ real) (g::real ⇒ real) (f′::real) (s::real ⇒ bool) x::real. real_open s ∧ IN x s ∧ real_continuous_on f s ∧ (∀x::real. IN x s ⟶ g (f x) = x) ∧ has_real_derivative f f′ (atreal x) ∧ f′ ≠ (0::real) ⟶ has_real_derivative g (inverse_class.inverse f′) (atreal (f x))

thm HAS_REAL_DERIVATIVE_INVERSE_STRONG_X:

∀(f::real ⇒ real) (g::real ⇒ real) (f′::real) (s::real ⇒ bool) y::real. real_open s ∧ IN (g y) s ∧ real_continuous_on f s ∧ (∀x::real. IN x s ⟶ g (f x) = x) ∧ has_real_derivative f f′ (atreal (g y)) ∧ f′ ≠ (0::real) ∧ f (g y) = y ⟶ has_real_derivative g (inverse_class.inverse f′) (atreal y)

thm HAS_REAL_DERIVATIVE_SEQUENCE:

∀(s::real ⇒ bool) (f::nat ⇒ real ⇒ real) (f′::nat ⇒ real ⇒ real) g′::real ⇒ real. is_realinterval s ∧ (∀(n::nat) x::real. IN x s ⟶ has_real_derivative (f n) (f′ n x) (within (atreal x) s)) ∧ (∀e>0::real. ∃N::nat. ∀(n::nat) x::real. N ≤ n ∧ IN x s ⟶ |f′ n x − g′ x| ≤ e) ∧ (∃(x::real) l::real. IN x s ∧ −−−> (λn::nat. f n x) l sequentially) ⟶ (∃g::real ⇒ real. ∀x::real. IN x s ⟶ −−−> (λn::nat. f n x) (g x) sequentially ∧ has_real_derivative g (g′ x) (within (atreal x) s))

thm HAS_REAL_DERIVATIVE_SERIES:

∀(s::real ⇒ bool) (f::nat ⇒ real ⇒ real) (f′::nat ⇒ real ⇒ real) (g′::real ⇒ real) k::nat ⇒ bool. is_realinterval s ∧ (∀(n::nat) x::real. IN x s ⟶ has_real_derivative (f n) (f′ n x) (within (atreal x) s)) ∧ (∀e>0::real. ∃N::nat. ∀(n::nat) x::real. N ≤ n ∧ IN x s ⟶ |sum (HOL_Light_Import.INTER k (dotdot (0::nat) n)) (λi::nat. f′ i x) − g′ x| ≤ e) ∧ (∃(x::real) l::real. IN x s ∧ real_sums (λn::nat. f n x) l k) ⟶ (∃g::real ⇒ real. ∀x::real. IN x s ⟶ real_sums (λn::nat. f n x) (g x) k ∧ has_real_derivative g (g′ x) (within (atreal x) s))

thm REAL_DIFFERENTIABLE_BOUND:

∀(f::real ⇒ real) (f′::real ⇒ real) (s::real ⇒ bool) B::real. is_realinterval s ∧ (∀x::real. IN x s ⟶ has_real_derivative f (f′ x) (within (atreal x) s) ∧ |f′ x| ≤ B) ⟶ (∀(x::real) y::real. IN x s ∧ IN y s ⟶ |f x − f y| ≤ B ∗ |x − y|)

thm REAL_TAYLOR_MVT_POS:

∀(f::nat ⇒ real ⇒ real) (a::real) (x::real) n::nat. a < x ∧ (∀(i::nat) t::real. IN t (closed_real_interval [(a, x)]) ∧ i ≤ n ⟶ has_real_derivative (f i) (f (i + (1::nat)) t) (within (atreal t) (closed_real_interval [(a, x)]))) ⟶ (∃t::real. IN t (open_real_interval (a, x)) ∧ f (0::nat) x = sum (dotdot (0::nat) n) (λi::nat. f i a ∗ ($(x - a)^{i}$ / real_of_nat (fact i))) + f (n + (1::nat)) t ∗ ($(x - a)^{n + (1::nat)}$ / real_of_nat (fact (n + (1::nat)))))

thm REAL_TAYLOR_MVT_NEG:

∀ (f::nat ⇒ real ⇒ real) (a::real) (x::real) n::nat. x < a ∧ (∀ (i::nat) t::real. IN t (closed_real_interval [(x, a)]) ∧ i ≤ n ⟶ has_real_derivative (f i) (f (i + (1::nat)) t) (within (atreal t) (closed_real_interval [(x, a)]))) ⟶ (∃ t::real. IN t (open_real_interval (x, a)) ∧ f (0::nat) x = sum (dotdot (0::nat) n) (λi::nat. f i a ∗ ($(x - a)^{i}$ / real_of_nat (fact i))) + f (n + (1::nat)) t ∗ ($(x - a)^{n + (1::nat)}$ / real_of_nat (fact (n + (1::nat)))))

thm REAL_TAYLOR:

∀ (f::nat ⇒ real ⇒ real) (n::nat) (s::real ⇒ bool) B::real. is_realinterval s ∧ (∀ (i::nat) x::real. IN x s ∧ i ≤ n ⟶ has_real_derivative (f i) (f (i + (1::nat)) x) (within (atreal x) s)) ∧ (∀ x::real. IN x s ⟶ |f (n + (1::nat)) x| ≤ B) ⟶ (∀ (w::real) z::real. IN w s ∧ IN z s ⟶ |f (0::nat) z − sum (dotdot (0::nat) n) (λi::nat. f i w ∗ ($(z - w)^{i}$ / real_of_nat (fact i)))| ≤ B ∗ ($|z - w|^{n + (1::nat)}$ / real_of_nat (fact (n + (1::nat)))))

thm REAL_SUM_INTEGRAL_UBOUND_INCREASING:

∀ (f::real ⇒ real) (g::real ⇒ real) (m::nat) n::nat. m ≤ n ∧ (∀ x::real. IN x (closed_real_interval [(real_of_nat m, real_of_nat n + (1::real))]) ⟶ has_real_derivative g (f x) (within (atreal x) (closed_real_interval [(real_of_nat m, real_of_nat n + (1::real))]))) ∧ (∀ (x::real) y::real. real_of_nat m ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n + (1::real) ⟶ f x ≤ f y) ⟶ sum (dotdot m n) (λk::nat. f (real_of_nat k)) ≤ g (real_of_nat n + (1::real)) − g (real_of_nat m)

thm REAL_SUM_INTEGRAL_UBOUND_DECREASING:

∀ (f::real ⇒ real) (g::real ⇒ real) (m::nat) n::nat. m ≤ n ∧ (∀ x::real. IN x (closed_real_interval [(real_of_nat m − (1::real), real_of_nat n)]) ⟶ has_real_derivative g (f x) (within (atreal x) (closed_real_interval [(real_of_nat m − (1::real), real_of_nat n)]))) ∧ (∀ (x::real) y::real. real_of_nat m − (1::real) ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n ⟶ f y ≤ f x) ⟶ sum (dotdot m n) (λk::nat. f (real_of_nat k)) ≤ g (real_of_nat n) − g (real_of_nat m − (1::real))

thm REAL_SUM_INTEGRAL_LBOUND_INCREASING:

∀ (f::real ⇒ real) (g::real ⇒ real) (m::nat) n::nat. m ≤ n ∧ (∀ x::real. IN x (closed_real_interval [(real_of_nat m − (1::real), real_of_nat n)]) ⟶ has_real_derivative g (f x) (within (atreal x) (closed_real_interval [(real_of_nat m − (1::real), real_of_nat n)]))) ∧ (∀ (x::real) y::real. real_of_nat m − (1::real) ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n ⟶ f x ≤ f y) ⟶ g (real_of_nat n) − g (real_of_nat m − (1::real)) ≤ sum (dotdot m n) (λk::nat. f (real_of_nat k))

thm REAL_SUM_INTEGRAL_LBOUND_DECREASING:

∀ (f::real ⇒ real) (g::real ⇒ real) (m::nat) n::nat. m ≤ n ∧ (∀ x::real. IN x (closed_real_interval [(real_of_nat m, real_of_nat n + (1::real))]) ⟶ has_real_derivative g (f x) (within (atreal x) (closed_real_interval [(real_of_nat m, real_of_nat n + (1::real))]))) ∧ (∀ (x::real) y::real. real_of_nat m ≤ x ∧ x ≤ y ∧ y ≤ real_of_nat n + (1::real) ⟶ f y ≤ f x) ⟶ g (real_of_nat n + (1::real)) − g (real_of_nat m) ≤ sum (dotdot m n) (λk::nat. f (real_of_nat k))

thm REAL_SUM_INTEGRAL_BOUNDS_INCREASING:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (m::nat)\ n::nat.\ m \leq n \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(real\_of\_nat\ m - (1::real),\ real\_of\_nat\ n + (1::real))])\ \longrightarrow has\_real\_derivative\ g\ (f\,x)\ (within\ (atreal\,x)\ (closed\_real\_interval\ [(real\_of\_nat\ m - (1::real),\ real\_of\_nat\ n + (1::real))]))) \wedge (\forall (x::real)\ y::real.\ real\_of\_nat\ m - (1::real) \leq x \wedge x \leq y \wedge y \leq real\_of\_nat\ n + (1::real) \longrightarrow f\ x \leq f\ y) \longrightarrow g\ (real\_of\_nat\ n) - g\ (real\_of\_nat\ m - (1::real)) \leq sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ (real\_of\_nat\ k)) \wedge sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ (real\_of\_nat\ k)) \leq g\ (real\_of\_nat\ n + (1::real)) - g\ (real\_of\_nat\ m)$

thm REAL_SUM_INTEGRAL_BOUNDS_DECREASING:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (m::nat)\ n::nat.\ m \leq n \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(real\_of\_nat\ m - (1::real),\ real\_of\_nat\ n + (1::real))])\ \longrightarrow has\_real\_derivative\ g\ (f\,x)\ (within\ (atreal\,x)\ (closed\_real\_interval\ [(real\_of\_nat\ m - (1::real),\ real\_of\_nat\ n + (1::real))]))) \wedge (\forall (x::real)\ y::real.\ real\_of\_nat\ m - (1::real) \leq x \wedge x \leq y \wedge y \leq real\_of\_nat\ n + (1::real) \longrightarrow f\ y \leq f\ x) \longrightarrow g\ (real\_of\_nat\ n + (1::real)) - g\ (real\_of\_nat\ m) \leq sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ (real\_of\_nat\ k)) \wedge sum\ (dotdot\ m\ n)\ (\lambda k::nat.\ f\ (real\_of\_nat\ k)) \leq g\ (real\_of\_nat\ n) - g\ (real\_of\_nat\ m - (1::real))$

thm LIM_POSINFINITY_SEQUENTIALLY:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ -\!-\!>\ f\ l\ at\_posinfinity \longrightarrow -\!-\!>\ (\lambda n::nat.\ f\ (real\_of\_nat\ n))\ l\ sequentially$

thm REALLIM_POSINFINITY_SEQUENTIALLY:

$\forall (f::real \Rightarrow real)\ l::real.\ -\!-\!-\!>\ f\ l\ at\_posinfinity \longrightarrow -\!-\!-\!>\ (\lambda n::nat.\ f\ (real\_of\_nat\ n))\ l\ sequentially$

thm LIM_ZERO_POSINFINITY:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ -\!-\!>\ (\lambda x::real.\ f\ ((1::real)\ /\ x))\ l\ (atreal\ (0::real)) \longrightarrow -\!-\!>\ f\ l\ at\_posinfinity$

thm LIM_ZERO_NEGINFINITY:

$\forall (f::real \Rightarrow (real,\ ?'a::type)\ cart)\ l::(real,\ ?'a::type)\ cart.\ -\!-\!>\ (\lambda x::real.\ f\ ((1::real)\ /\ x))\ l\ (atreal\ (0::real)) \longrightarrow -\!-\!>\ f\ l\ at\_neginfinity$

thm REALLIM_ZERO_POSINFINITY:

$\forall (f::real \Rightarrow real)\ l::real.\ -\!-\!-\!>\ (\lambda x::real.\ f\ ((1::real)\ /\ x))\ l\ (atreal\ (0::real)) \longrightarrow -\!-\!-\!>\ f\ l\ at\_posinfinity$

thm REALLIM_ZERO_NEGINFINITY:

$\forall (f::real \Rightarrow real)\ l::real.\ -\!-\!-\!>\ (\lambda x::real.\ f\ ((1::real)\ /\ x))\ l\ (atreal\ (0::real)) \longrightarrow -\!-\!-\!>\ f\ l\ at\_neginfinity$

thm DEF_closed_real_segment:

closed_real_segment = (*SOME closed_real_segment*::*nat* ⇒ (*real* × *real*) *list* ⇒ *real* ⇒ *bool*. ∀ (*_1894851*::*nat*) (*a*::*real*) *b*::*real*. *closed_real_segment _1894851* [(*a*, *b*)] = *GSPEC* (λ*GEN%PVAR%2449*::*real*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%2449* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (((*1*::*real*) − *u*) ∗ *a* + *u* ∗ *b*))) (*59*::*nat*)

thm closed_real_segment:

closed_real_segment [(*?a*::*real*, *?b*::*real*)] = *GSPEC* (λ*GEN%PVAR%2449*::*real*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%2449* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (((*1*::*real*) − *u*) ∗ *?a* + *u* ∗ *?b*))

thm DEF_open_real_segment:

open_real_segment = (λ*_1894852*::*real* × *real*. *DIFF* (*closed_real_segment* [(*fst _1894852*, *snd _1894852*)]) (*INSERT* (*fst _1894852*) (*INSERT* (*snd _1894852*) *EMPTY*)))

thm real_segment_conjunct1:

open_real_segment (*?a*::*real*, *?b*::*real*) = *DIFF* (*closed_real_segment* [(*?a*, *?b*)]) (*INSERT ?a* (*INSERT ?b EMPTY*))

thm open_real_segment:

∀ (*a*::*real*) *b*::*real*. *open_real_segment* (*a*, *b*) = *DIFF* (*closed_real_segment* [(*a*, *b*)]) (*INSERT a* (*INSERT b EMPTY*))

thm real_segment:

closed_real_segment [(*?a*::*real*, *?b*::*real*)] = *GSPEC* (λ*GEN%PVAR%2450*::*real*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%2450* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (((*1*::*real*) − *u*) ∗ *?a* + *u* ∗ *?b*)) ∧ *open_real_segment* (*?a*, *?b*) = *DIFF* (*closed_real_segment* [(*?a*, *?b*)]) (*INSERT ?a* (*INSERT ?b EMPTY*))

thm real_segment_conjunct0:

closed_real_segment [(*?a*::*real*, *?b*::*real*)] = *GSPEC* (λ*GEN%PVAR%2450*::*real*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%2450* ((*0*::*real*) ≤ *u* ∧ *u* ≤ (*1*::*real*)) (((*1*::*real*) − *u*) ∗ *?a* + *u* ∗ *?b*))

thm REAL_SEGMENT_SEGMENT:

(∀ (*a*::*real*) *b*::*real*. *closed_real_segment* [(*a*, *b*)] = *IMAGE HOL_Light_Import.drop* (*closed_segment* [(*lift a*, *lift b*)])) ∧ (∀ (*a*::*real*) *b*::*real*. *open_real_segment* (*a*, *b*) = *IMAGE HOL_Light_Import.drop* (*open_segment* (*lift a*, *lift b*)))

thm REAL_SEGMENT_SEGMENT_conjunct1:

∀ (*a*::*real*) *b*::*real*. *open_real_segment* (*a*, *b*) = *IMAGE HOL_Light_Import.drop* (*open_segment* (*lift a*, *lift b*))

thm REAL_SEGMENT_SEGMENT_conjunct0:

∀ (*a*::*real*) *b*::*real*. *closed_real_segment* [(*a*, *b*)] = *IMAGE HOL_Light_Import.drop* (*closed_segment* [(*lift a*, *lift b*)])

thm SEGMENT_REAL_SEGMENT:

$(\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ closed\_segment\ [(a,\ b)] = IMAGE\ lift\ (closed\_real\_segment\ [(HOL\_Light\_Import.drop\ a,\ HOL\_Light\_Import.drop\ b)])) \wedge (\forall (a::(real,\ unit)\ cart)\ b::(real,\ unit)\ cart.\ open\_segment\ (a,\ b) = IMAGE\ lift\ (open\_real\_segment\ (HOL\_Light\_Import.drop\ a,\ HOL\_Light\_Import.drop\ b)))$

thm REAL_SEGMENT_INTERVAL:

$(\forall (a::real)\ b::real.\ closed\_real\_segment\ [(a,\ b)] = (if\ a \leq b\ then\ closed\_real\_interval\ [(a,\ b)]\ else\ closed\_real\_interval\ [(b,\ a)])) \wedge (\forall (a::real)\ b::real.\ open\_real\_segment\ (a,\ b) = (if\ a \leq b\ then\ open\_real\_interval\ (a,\ b)\ else\ open\_real\_interval\ (b,\ a)))$

thm DEF_real_convex_on:

$real\_convex\_on = (\lambda(\_1894879::real \Rightarrow real)\ \_1894880::real \Rightarrow bool.\ \forall (x::real)\ (y::real)\ (u::real)\ v::real.\ IN\ x\ \_1894880 \wedge IN\ y\ \_1894880 \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow \_1894879\ (u * x + v * y) \leq u * \_1894879\ x + v * \_1894879\ y)$

thm real_convex_on:

$\forall (s::real \Rightarrow bool)\ f::real \Rightarrow real.\ real\_convex\_on\ f\ s = (\forall (x::real)\ (y::real)\ (u::real)\ v::real.\ IN\ x\ s \wedge IN\ y\ s \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow f\ (u * x + v * y) \leq u * f\ x + v * f\ y)$

thm REAL_CONVEX_ON:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_convex\_on\ f\ s = convex\_on\ (f \circ HOL\_Light\_Import.drop)\ (IMAGE\ lift\ s)$

thm REAL_CONVEX_ON_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_convex\_on\ f\ t \wedge SUBSET\ s\ t \longrightarrow real\_convex\_on\ f\ s$

thm REAL_CONVEX_ADD:

$\forall (s::real \Rightarrow bool)\ (f::real \Rightarrow real)\ g::real \Rightarrow real.\ real\_convex\_on\ f\ s \wedge real\_convex\_on\ g\ s \longrightarrow real\_convex\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm REAL_CONVEX_LMUL:

$\forall (s::real \Rightarrow bool)\ (c::real)\ f::real \Rightarrow real.\ (0::real) \leq c \wedge real\_convex\_on\ f\ s \longrightarrow real\_convex\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm REAL_CONVEX_RMUL:

$\forall (s::real \Rightarrow bool)\ (c::real)\ f::real \Rightarrow real.\ (0::real) \leq c \wedge real\_convex\_on\ f\ s \longrightarrow real\_convex\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm REAL_CONVEX_LOWER:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (x::real)\ y::real.\ real\_convex\_on\ f\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge (0::real) \leq (?u::real) \wedge (0::real) \leq (?v::real) \wedge ?u + ?v = (1::real) \longrightarrow f\ (?u * x + ?v * y) \leq max\ (f\ x)\ (f\ y)$

thm REAL_CONVEX_LOCAL_GLOBAL_MINIMUM:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ x::real.\ real\_convex\_on\ f\ s \wedge IN\ x\ t \wedge real\_open\ t \wedge SUBSET\ t\ s \wedge (\forall y::real.\ IN\ y\ t \longrightarrow f\ x \leq f\ y) \longrightarrow (\forall y::real.\ IN\ y\ s \longrightarrow f\ x \leq f\ y)$

thm REAL_CONVEX_DISTANCE:

$\forall (s::real \Rightarrow bool)\ a::real.\ real\_convex\_on\ (\lambda x::real.\ |a - x|)\ s$

thm REAL_CONVEX_ON_JENSEN:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \longrightarrow real\_convex\_on\ f\ s = (\forall (k::nat)\ (u::nat \Rightarrow real)\ x::nat \Rightarrow real.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq k \longrightarrow (0::real) \leq u\ i \wedge IN\ (x\ i)\ s) \wedge sum\ (dotdot\ (1::nat)\ k)\ u = (1::real) \longrightarrow f\ (sum\ (dotdot\ (1::nat)\ k)\ (\lambda i::nat.\ u\ i * x\ i)) \leq sum\ (dotdot\ (1::nat)\ k)\ (\lambda i::nat.\ u\ i * f\ (x\ i)))$

thm REAL_CONVEX_ON_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_open\ s \wedge real\_convex\_on\ f\ s \longrightarrow real\_continuous\_on\ f\ s$

thm REAL_CONVEX_ON_LEFT_SECANT_MUL:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_convex\_on\ f\ s = (\forall (a::real)\ (b::real)\ x::real.\ IN\ a\ s \wedge IN\ b\ s \wedge IN\ x\ (closed\_real\_segment\ [(a,\ b)]) \longrightarrow (f\ x - f\ a) * |b - a| \leq (f\ b - f\ a) * |x - a|)$

thm REAL_CONVEX_ON_RIGHT_SEQUENT_MUL:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_convex\_on\ f\ s = (\forall (a::real)\ (b::real)\ x::real.\ IN\ a\ s \wedge IN\ b\ s \wedge IN\ x\ (closed\_real\_segment\ [(a,\ b)]) \longrightarrow (f\ b - f\ a) * |b - x| \leq (f\ b - f\ x) * |b - a|)$

thm REAL_CONVEX_ON_LEFT_SECANT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_convex\_on\ f\ s = (\forall (a::real)\ (b::real)\ x::real.\ IN\ a\ s \wedge IN\ b\ s \wedge IN\ x\ (open\_real\_segment\ (a,\ b)) \longrightarrow (f\ x - f\ a)\ /\ |x - a| \leq (f\ b - f\ a)\ /\ |b - a|)$

thm REAL_CONVEX_ON_RIGHT_SEQUENT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_convex\_on\ f\ s = (\forall (a::real)\ (b::real)\ x::real.\ IN\ a\ s \wedge IN\ b\ s \wedge IN\ x\ (open\_real\_segment\ (a,\ b)) \longrightarrow (f\ b - f\ a)\ /\ |b - a| \leq (f\ b - f\ x)\ /\ |b - x|)$

thm REAL_CONVEX_ON_DERIVATIVE_SECANT_IMP:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (s::real \Rightarrow bool)\ (x::real)\ y::real.\ real\_convex\_on\ f\ s \wedge SUBSET\ (closed\_real\_segment\ [(x,\ y)])\ s \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s) \longrightarrow f' * (y - x) \leq f\ y - f\ x$

thm REAL_CONVEX_ON_SECANT_DERIVATIVE_IMP:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (s::real \Rightarrow bool)\ (x::real)\ y::real.\ real\_convex\_on\ f\ s \wedge SUBSET\ (closed\_real\_segment\ [(x,\ y)])\ s \wedge has\_real\_derivative\ f\ f'\ (within\ (atreal\ y)\ s) \longrightarrow f\ y - f\ x \leq f' * (y - x)$

thm REAL_CONVEX_ON_DERIVATIVES_IMP:

$\forall (f::real \Rightarrow real)\ (f'x::real)\ (f'y::real)\ (s::real \Rightarrow bool)\ (x::real)\ y::real.\ real\_convex\_on\ f\ s \wedge SUBSET\ (closed\_real\_segment\ [(x,\ y)])\ s \wedge has\_real\_derivative\ f\ f'x\ (within\ (atreal\ x)\ s) \wedge has\_real\_derivative\ f\ f'y\ (within\ (atreal\ y)\ s) \longrightarrow f'x * (y - x) \leq f'y * (y - x)$

thm REAL_SEGMENT_INTERVAL_conjunct1:

$\forall (a::real)\ b::real.\ open\_real\_segment\ (a,\ b) = (if\ a \leq b\ then\ open\_real\_interval\ (a,\ b)\ else\ open\_real\_interval\ (b,\ a))$

thm REAL_SEGMENT_INTERVAL_conjunct0:

$\forall (a::real)\ b::real.\ closed\_real\_segment\ [(a,\ b)] = (if\ a \leq b\ then\ closed\_real\_interval\ [(a,\ b)]\ else\ closed\_real\_interval\ [(b,\ a)])$

thm REAL_CONVEX_ON_DERIVATIVE_INCREASING_IMP:

$\forall (f::real \Rightarrow real)\ (f'x::real)\ (f'y::real)\ (s::real \Rightarrow bool)\ (x::real)\ y::real.\ real\_convex\_on\ f\ s \wedge SUBSET\ (closed\_real\_interval\ [(x,\ y)])\ s \wedge has\_real\_derivative\ f\ f'x\ (within\ (atreal\ x)\ s) \wedge has\_real\_derivative\ f\ f'y\ (within\ (atreal\ y)\ s) \wedge x < y \longrightarrow f'x \leq f'y$

thm REAL_CONVEX_ON_DERIVATIVE_SECANT:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow real\_convex\_on\ f\ s = (\forall (x::real)\ y::real.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f'\ x * (y - x) \leq f\ y - f\ x)$

thm REAL_CONVEX_ON_SECANT_DERIVATIVE:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow real\_convex\_on\ f\ s = (\forall (x::real)\ y::real.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f\ y - f\ x \leq f'\ y * (y - x))$

thm REAL_CONVEX_ON_DERIVATIVES:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow real\_convex\_on\ f\ s = (\forall (x::real)\ y::real.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f'\ x * (y - x) \leq f'\ y * (y - x))$

thm REAL_CONVEX_ON_DERIVATIVE_INCREASING:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow real\_convex\_on\ f\ s = (\forall (x::real)\ y::real.\ IN\ x\ s \wedge IN\ y\ s \wedge x \leq y \longrightarrow f'\ x \leq f'\ y)$

thm HAS_REAL_DERIVATIVE_INCREASING_IMP:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ is\_realinterval\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \wedge$

$(\forall\, x{::}real.\ IN\ x\ s \longrightarrow (0{::}real) \le f'\ x) \wedge IN\ a\ s \wedge IN\ b\ s \wedge a \le b \longrightarrow f\ a \le f\ b$

thm HAS_REAL_DERIVATIVE_INCREASING:

$\forall\, (f{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ s{::}real \Rightarrow bool.\ is\_realinterval\ s \wedge \neg\, (\exists\, a{::}real.\ s = INSERT\ a\ EMPTY) \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow (\forall\, x{::}real.\ IN\ x\ s \longrightarrow (0{::}real) \le f'\ x) = (\forall\, (x{::}real)\ y{::}real.\ IN\ x\ s \wedge IN\ y\ s \wedge x \le y \longrightarrow f\, x \le f\, y)$

thm REAL_CONVEX_ON_SECOND_DERIVATIVE:

$\forall\, (f{::}real \Rightarrow real)\ (f'{::}real \Rightarrow real)\ (f''{::}real \Rightarrow real)\ s{::}real \Rightarrow bool.\ is\_realinterval\ s \wedge \neg\, (\exists\, a{::}real.\ s = INSERT\ a\ EMPTY) \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \wedge (\forall\, x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f'\ (f''\ x)\ (within\ (atreal\ x)\ s)) \longrightarrow real\_convex\_on\ f\ s = (\forall\, x{::}real.\ IN\ x\ s \longrightarrow (0{::}real) \le f''\ x)$

thm REAL_CONVEX_ON_ASYM:

$\forall\, (s{::}real \Rightarrow bool)\ f{::}real \Rightarrow real.\ real\_convex\_on\ f\ s = (\forall\, (x{::}real)\ (y{::}real)\ (u{::}real)\ v{::}real.\ IN\ x\ s \wedge IN\ y\ s \wedge x < y \wedge (0{::}real) \le u \wedge (0{::}real) \le v \wedge u + v = (1{::}real) \longrightarrow f\ (u * x + v * y) \le u * f\, x + v * f\, y)$

thm REAL_CONVEX_ON_EXP:

$\forall\, s{::}real \Rightarrow bool.\ real\_convex\_on\ exp\ s$

thm REAL_CONVEX_ON_RPOW:

$\forall\, (s{::}real \Rightarrow bool)\ t{::}real.\ SUBSET\ s\ (GSPEC\ (\lambda GEN\%PVAR\%2454{::}real.\ \exists\, x{::}real.\ SETSPEC\ GEN\%PVAR\%2454\ ((0{::}real) \le x)\ x)) \wedge (1{::}real) \le t \longrightarrow real\_convex\_on\ (\lambda x{::}real.\ rpow\ x\ t)\ s$

thm DEF_has_real_integral:

$has\_real\_integral = (\lambda(\_1895270{::}real \Rightarrow real)\ (\_1895271{::}real)\ \_1895272{::}real \Rightarrow bool.\ has\_integral\ (lift \circ (\_1895270 \circ HOL\_Light\_Import.drop))\ (lift\ \_1895271)\ (IMAGE\ lift\ \_1895272))$

thm has_real_integral:

$\forall\, (f{::}real \Rightarrow real)\ (y{::}real)\ s{::}real \Rightarrow bool.\ has\_real\_integral\, f\, y\, s = has\_integral\ (lift \circ (f \circ HOL\_Light\_Import.drop))\ (lift\ y)\ (IMAGE\ lift\ s)$

thm DEF_real_integrable_on:

$real\_integrable\_on = (\lambda(\_1895291{::}real \Rightarrow real)\ \_1895292{::}real \Rightarrow bool.\ \exists\, y{::}real.\ has\_real\_integral\ \_1895291\ y\ \_1895292)$

thm real_integrable_on:

$\forall\, (f{::}real \Rightarrow real)\ i{::}real \Rightarrow bool.\ real\_integrable\_on\, f\, i = (\exists\, y{::}real.\ has\_real\_integral\ f\ y\ i)$

thm DEF_real_integral:

$real\_integral = (\lambda(\_1895303::real \Rightarrow bool)\ \_1895304::real \Rightarrow real.\ SOME\ y::real.\ has\_real\_integral\ \_1895304\ y\ \_1895303)$

thm real_integral:

$\forall (f::real \Rightarrow real)\ i::real \Rightarrow bool.\ real\_integral\ i\ f = (SOME\ y::real.\ has\_real\_integral\ f\ y\ i)$

thm DEF_real_negligible:

$real\_negligible = (\lambda\_1895315::real \Rightarrow bool.\ negligible\ (IMAGE\ lift\ \_1895315))$

thm real_negligible:

$\forall s::real \Rightarrow bool.\ real\_negligible\ s = negligible\ (IMAGE\ lift\ s)$

thm DEF_absolutely_real_integrable_on:

$absolutely\_real\_integrable\_on = (\lambda(\_1895320::real \Rightarrow real)\ \_1895321::real \Rightarrow bool.\ real\_integrable\_on\ \_1895320\ \_1895321 \wedge real\_integrable\_on\ (\lambda x::real.\ |\_1895320\ x|)\ \_1895321)$

thm absolutely_real_integrable_on:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s = (real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ (\lambda x::real.\ |f\ x|)\ s)$

thm DEF_has_real_measure:

$has\_real\_measure = (\lambda(\_1895332::real \Rightarrow bool)\ \_1895333::real.\ has\_real\_integral\ (\lambda x::real.\ 1::real)\ \_1895333\ \_1895332)$

thm has_real_measure:

$\forall (m::real)\ s::real \Rightarrow bool.\ has\_real\_measure\ s\ m = has\_real\_integral\ (\lambda x::real.\ 1::real)\ m\ s$

thm DEF_real_measurable:

$real\_measurable = (\lambda\_1895344::real \Rightarrow bool.\ \exists\, m::real.\ has\_real\_measure\ \_1895344\ m)$

thm real_measurable:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s = (\exists\, m::real.\ has\_real\_measure\ s\ m)$

thm DEF_real_measure:

$real\_measure = (\lambda\_1895349::real \Rightarrow bool.\ SOME\ m::real.\ has\_real\_measure\ \_1895349\ m)$

thm real_measure:

$\forall s::real \Rightarrow bool.\ real\_measure\ s = (SOME\ m::real.\ has\_real\_measure\ s\ m)$

thm HAS_REAL_INTEGRAL:

has_real_integral (*?f*::*real* ⇒ *real*) (*?y*::*real*) (*closed_real_interval* [(*?a*::*real*, *?b*::*real*)]) = *has_integral* (*lift* ∘ (*?f* ∘ *HOL_Light_Import.drop*)) (*lift ?y*) (*closed_interval* [(*lift ?a*, *lift ?b*)])

thm REAL_INTEGRABLE_INTEGRAL:

∀ (*f*::*real* ⇒ *real*) *i*::*real* ⇒ *bool*. *real_integrable_on f i* ⟶ *has_real_integral f* (*real_integral i f*) *i*

thm HAS_REAL_INTEGRAL_INTEGRABLE:

∀ (*f*::*real* ⇒ *real*) (*i*::*real*) *s*::*real* ⇒ *bool*. *has_real_integral f i s* ⟶ *real_integrable_on f s*

thm HAS_REAL_INTEGRAL_INTEGRAL:

∀ (*f*::*real* ⇒ *real*) *s*::*real* ⇒ *bool*. *real_integrable_on f s* = *has_real_integral f* (*real_integral s f*) *s*

thm HAS_REAL_INTEGRAL_UNIQUE:

∀ (*f*::*real* ⇒ *real*) (*i*::*real* ⇒ *bool*) (*k1*::*real*) *k2*::*real*. *has_real_integral f k1 i* ∧ *has_real_integral f k2 i* ⟶ *k1* = *k2*

thm REAL_INTEGRAL_UNIQUE:

∀ (*f*::*real* ⇒ *real*) (*y*::*real*) *k*::*real* ⇒ *bool*. *has_real_integral f y k* ⟶ *real_integral k f* = *y*

thm HAS_REAL_INTEGRAL_INTEGRABLE_INTEGRAL:

∀ (*f*::*real* ⇒ *real*) (*i*::*real*) *s*::*real* ⇒ *bool*. *has_real_integral f i s* = (*real_integrable_on f s* ∧ *real_integral s f* = *i*)

thm REAL_INTEGRAL_EQ_HAS_INTEGRAL:

∀ (*s*::*real* ⇒ *bool*) (*f*::*real* ⇒ *real*) *y*::*real*. *real_integrable_on f s* ⟶ (*real_integral s f* = *y*) = *has_real_integral f y s*

thm REAL_INTEGRABLE_ON:

real_integrable_on (*?f*::*real* ⇒ *real*) (*?s*::*real* ⇒ *bool*) = *integrable_on* (*lift* ∘ (*?f* ∘ *HOL_Light_Import.drop*)) (*IMAGE lift ?s*)

thm ABSOLUTELY_REAL_INTEGRABLE_ON:

absolutely_real_integrable_on (*?f*::*real* ⇒ *real*) (*?s*::*real* ⇒ *bool*) = *absolutely_integrable_on* (*lift* ∘ (*?f* ∘ *HOL_Light_Import.drop*)) (*IMAGE lift ?s*)

thm REAL_INTEGRAL:

real_integrable_on (*?f*::*real* ⇒ *real*) (*?s*::*real* ⇒ *bool*) ⟶ *real_integral ?s ?f* = *HOL_Light_Import.drop* (*integral* (*IMAGE lift ?s*) (*lift* ∘ (*?f* ∘ *HOL_Light_Import.drop*)))

thm HAS_REAL_INTEGRAL_IS_0:

∀ (*f*::*real* ⇒ *real*) *s*::*real* ⇒ *bool*. (∀ *x*::*real*. *IN x s* ⟶ *f x* = (*0*::*real*)) ⟶ *has_real_integral f* (*0*::*real*) *s*

thm HAS_REAL_INTEGRAL_0:

$\forall s::real \Rightarrow bool.\ has\_real\_integral\ (\lambda x::real.\ 0::real)\ (0::real)\ s$

thm HAS_REAL_INTEGRAL_0_EQ:

$\forall (i::real)\ s::real \Rightarrow bool.\ has\_real\_integral\ (\lambda x::real.\ 0::real)\ i\ s = (i = (0::real))$

thm HAS_REAL_INTEGRAL_LINEAR:

$\forall (f::real \Rightarrow real)\ (y::real)\ (s::real \Rightarrow bool)\ h::real \Rightarrow real.\ has\_real\_integral\ f\ y\ s \wedge linear\ (lift \circ (h \circ HOL\_Light\_Import.drop)) \longrightarrow has\_real\_integral\ (h \circ f)\ (h\ y)\ s$

thm HAS_REAL_INTEGRAL_LMUL:

$\forall (f::real \Rightarrow real)\ (k::real)\ (s::real \Rightarrow bool)\ c::real.\ has\_real\_integral\ f\ k\ s \longrightarrow has\_real\_integral\ (\lambda x::real.\ c * f\ x)\ (c * k)\ s$

thm HAS_REAL_INTEGRAL_RMUL:

$\forall (f::real \Rightarrow real)\ (k::real)\ (s::real \Rightarrow bool)\ c::real.\ has\_real\_integral\ f\ k\ s \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ x * c)\ (k * c)\ s$

thm HAS_REAL_INTEGRAL_NEG:

$\forall (f::real \Rightarrow real)\ (k::real)\ s::real \Rightarrow bool.\ has\_real\_integral\ f\ k\ s \longrightarrow has\_real\_integral\ (\lambda x::real.\ - f\ x)\ (- k)\ s$

thm HAS_REAL_INTEGRAL_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (k::real)\ (l::real)\ s::real \Rightarrow bool.\ has\_real\_integral\ f\ k\ s \wedge has\_real\_integral\ g\ l\ s \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ x + g\ x)\ (k + l)\ s$

thm HAS_REAL_INTEGRAL_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (k::real)\ (l::real)\ s::real \Rightarrow bool.\ has\_real\_integral\ f\ k\ s \wedge has\_real\_integral\ g\ l\ s \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ x - g\ x)\ (k - l)\ s$

thm REAL_INTEGRAL_0:

$\forall s::real \Rightarrow bool.\ real\_integral\ s\ (\lambda x::real.\ 0::real) = (0::real)$

thm REAL_INTEGRAL_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ f\ x + g\ x) = real\_integral\ s\ f + real\_integral\ s\ g$

thm REAL_INTEGRAL_LMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ c * f\ x) = c * real\_integral\ s\ f$

thm REAL_INTEGRAL_RMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ f\ x * c) = real\_integral\ s\ f * c$

thm REAL_INTEGRAL_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ - f\ x) = - real\_integral\ s\ f$

thm REAL_INTEGRAL_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ f\ x - g\ x) = real\_integral\ s\ f - real\_integral\ s\ g$

thm REAL_INTEGRABLE_0:

$\forall s::real \Rightarrow bool.\ real\_integrable\_on\ (\lambda x::real.\ 0::real)\ s$

thm REAL_INTEGRABLE_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow real\_integrable\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm REAL_INTEGRABLE_LMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm REAL_INTEGRABLE_RMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm REAL_INTEGRABLE_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ (\lambda x::real.\ - f\ x)\ s$

thm REAL_INTEGRABLE_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow real\_integrable\_on\ (\lambda x::real.\ f\ x - g\ x)\ s$

thm REAL_INTEGRABLE_LINEAR:

$\forall (f::real \Rightarrow real)\ (h::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge linear\ (lift \circ (h \circ HOL\_Light\_Import.drop)) \longrightarrow real\_integrable\_on\ (h \circ f)\ s$

thm REAL_INTEGRAL_LINEAR:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ h::real \Rightarrow real.\ real\_integrable\_on\ f\ s \wedge linear\ (lift \circ (h \circ HOL\_Light\_Import.drop)) \longrightarrow real\_integral\ s\ (h \circ f) = h\ (real\_integral\ s\ f)$

thm HAS_REAL_INTEGRAL_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'a::type.\ IN\ a\ t \longrightarrow has\_real\_integral\ (f\ a)\ ((?i::?'a::type \Rightarrow real)\ a)\ s) \longrightarrow has\_real\_integral\ (\lambda x::real.\ sum\ t\ (\lambda a::?'a::type.\ f\ a\ x))\ (sum\ t\ ?i)\ s$

thm REAL_INTEGRAL_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'a::type.\ IN\ a\ t \longrightarrow real\_integrable\_on\ (f\ a)\ s) \longrightarrow real\_integral\ s\ (\lambda x::real.\ sum\ t\ (\lambda a::?'a::type.\ f\ a\ x)) = sum\ t\ (\lambda a::?'a::type.\ real\_integral\ s\ (f\ a))$

thm REAL_INTEGRABLE_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'a::type.\ IN\ a\ t \longrightarrow real\_integrable\_on\ (f\ a)\ s) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ sum\ t\ (\lambda a::?'a::type.\ f\ a\ x))\ s$

thm HAS_REAL_INTEGRAL_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (k::real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge has\_real\_integral\ f\ k\ s \longrightarrow has\_real\_integral\ g\ k\ s$

thm REAL_INTEGRABLE_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ g\ s$

thm HAS_REAL_INTEGRAL_EQ_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (k::real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow has\_real\_integral\ f\ k\ s = has\_real\_integral\ g\ k\ s$

thm HAS_REAL_INTEGRAL_NULL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ b \leq a \longrightarrow has\_real\_integral\ f\ (0::real)\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_REAL_INTEGRAL_NULL_EQ:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ i::real.\ b \leq a \longrightarrow has\_real\_integral\ f\ i\ (closed\_real\_interval\ [(a,\ b)]) = (i = (0::real))$

thm REAL_INTEGRAL_NULL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ b \leq a \longrightarrow real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f = (0::real)$

thm REAL_INTEGRABLE_ON_NULL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ b \leq a \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_REAL_INTEGRAL_EMPTY:

$\forall f::real \Rightarrow real.\ has\_real\_integral\ f\ (0::real)\ EMPTY$

thm HAS_REAL_INTEGRAL_EMPTY_EQ:

$\forall (f::real \Rightarrow real)\ i::real.\ has\_real\_integral\ f\ i\ EMPTY\ = (i = (0::real))$

thm REAL_INTEGRABLE_ON_EMPTY:

$\forall f::real \Rightarrow real.\ real\_integrable\_on\ f\ EMPTY$

thm REAL_INTEGRAL_EMPTY:

$\forall f::real \Rightarrow real.\ real\_integral\ EMPTY\ f = (0::real)$

thm HAS_REAL_INTEGRAL_REFL:

$\forall (f::real \Rightarrow real)\ a::real.\ has\_real\_integral\ f\ (0::real)\ (closed\_real\_interval\ [(a, a)])$

thm REAL_INTEGRABLE_ON_REFL:

$\forall (f::real \Rightarrow real)\ a::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a, a)])$

thm REAL_INTEGRAL_REFL:

$\forall (f::real \Rightarrow real)\ a::real.\ real\_integral\ (closed\_real\_interval\ [(a, a)])\ f = (0::real)$

thm HAS_REAL_INTEGRAL_CONST:

$\forall (a::real)\ (b::real)\ c::real.\ a \leq b \longrightarrow has\_real\_integral\ (\lambda x::real.\ c)\ (c * (b - a))\ (closed\_real\_interval\ [(a, b)])$

thm REAL_INTEGRABLE_CONST:

$\forall (a::real)\ (b::real)\ c::real.\ real\_integrable\_on\ (\lambda x::real.\ c)\ (closed\_real\_interval\ [(a, b)])$

thm REAL_INTEGRAL_CONST:

$\forall (a::real)\ (b::real)\ c::real.\ a \leq b \longrightarrow real\_integral\ (closed\_real\_interval\ [(a, b)])\ (\lambda x::real.\ c) = c * (b - a)$

thm HAS_REAL_INTEGRAL_BOUND:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ (i::real)\ B::real.\ (0::real) \leq B \wedge a \leq b \wedge has\_real\_integral\ f\ i\ (closed\_real\_interval\ [(a, b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a, b)]) \longrightarrow |f\ x| \leq B) \longrightarrow |i| \leq B * (b - a)$

thm HAS_REAL_INTEGRAL_LE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (i::real)\ j::real.\ has\_real\_integral\ f\ i\ s \wedge has\_real\_integral\ g\ j\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \longrightarrow i \leq j$

thm REAL_INTEGRAL_LE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \longrightarrow real\_integral\ s\ f \leq real\_integral\ s\ g$

thm HAS_REAL_INTEGRAL_POS:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ i::real.\ has\_real\_integral\ f\ i\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow (0::real) \leq i$

thm REAL_INTEGRAL_POS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow (0::real) \leq real\_integral\ s\ f$

thm HAS_REAL_INTEGRAL_ISNEG:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ i::real.\ has\_real\_integral\ f\ i\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow f\ x \leq (0::real)) \longrightarrow i \leq (0::real)$

thm HAS_REAL_INTEGRAL_LBOUND:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ i::real.\ a \leq b \wedge has\_real\_integral\ f\ i\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (?B::real) \leq f\ x) \longrightarrow ?B * (b - a) \leq i$

thm HAS_REAL_INTEGRAL_UBOUND:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ i::real.\ a \leq b \wedge has\_real\_integral\ f\ i\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow f\ x \leq (?B::real)) \longrightarrow i \leq ?B * (b - a)$

thm REAL_INTEGRAL_LBOUND:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ a \leq b \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (?B::real) \leq f\ x) \longrightarrow ?B * (b - a) \leq real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f$

thm REAL_INTEGRAL_UBOUND:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ a \leq b \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow f\ x \leq (?B::real)) \longrightarrow real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f \leq ?B * (b - a)$

thm REAL_INTEGRABLE_UNIFORM_LIMIT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall e>0::real.\ \exists g::real \Rightarrow real.\ (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow |f\ x - g\ x| \leq e) \wedge real\_integrable\_on\ g\ (closed\_real\_interval\ [(a,\ b)])) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_REAL_INTEGRAL_NEGLIGIBLE:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow f\ x = (0::real)) \longrightarrow has\_real\_integral\ f\ (0::real)\ t$

thm HAS_REAL_INTEGRAL_SPIKE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ y::real.\ real\_negligible\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \wedge has\_real\_integral\ f\ y\ t \longrightarrow has\_real\_integral\ g\ y\ t$

thm HAS_REAL_INTEGRAL_SPIKE_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ y::real.\ real\_negligible\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow has\_real\_integral\ f\ y\ t = has\_real\_integral\ g\ y\ t$

thm REAL_INTEGRABLE_SPIKE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow real\_integrable\_on\ f\ t \longrightarrow real\_integrable\_on\ g\ t$

thm REAL_INTEGRAL_SPIKE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \longrightarrow real\_integral\ t\ f = real\_integral\ t\ g$

thm REAL_NEGLIGIBLE_SUBSET:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \wedge SUBSET\ t\ s \longrightarrow real\_negligible\ t$

thm REAL_NEGLIGIBLE_DIFF:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \longrightarrow real\_negligible\ (DIFF\ s\ t)$

thm REAL_NEGLIGIBLE_INTER:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \vee real\_negligible\ t \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_NEGLIGIBLE_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ s \wedge real\_negligible\ t \longrightarrow real\_negligible\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_NEGLIGIBLE_UNION_EQ:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ (HOL\_Light\_Import.UNION\ s\ t) = (real\_negligible\ s \wedge real\_negligible\ t)$

thm REAL_NEGLIGIBLE_SING:

$\forall a::real.\ real\_negligible\ (INSERT\ a\ EMPTY)$

thm REAL_NEGLIGIBLE_INSERT:

$\forall (a::real)\ s::real \Rightarrow bool.\ real\_negligible\ (INSERT\ a\ s) = real\_negligible\ s$

thm REAL_NEGLIGIBLE_EMPTY:

$real\_negligible\ EMPTY$

thm REAL_NEGLIGIBLE_FINITE:

$\forall s::real \Rightarrow bool.\ FINITE\ s \longrightarrow real\_negligible\ s$

thm REAL_NEGLIGIBLE_UNIONS:

$\forall s::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ s \wedge (\forall t::real \Rightarrow bool.\ IN\ t\ s \longrightarrow real\_negligible\ t) \longrightarrow real\_negligible\ (UNIONS\ s)$

thm HAS_REAL_INTEGRAL_SPIKE_FINITE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ y::real.\ FINITE\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ t\ s) \longrightarrow g\ x = f\ x) \wedge has\_real\_integral\ f\ y\ t \longrightarrow has\_real\_integral\ g\ y\ t$

thm HAS_REAL_INTEGRAL_SPIKE_FINITE_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ y::real.\ FINITE\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ (?t::real \Rightarrow bool)\ s) \longrightarrow g\ x = f\ x) \longrightarrow has\_real\_integral\ f\ y\ ?t = has\_real\_integral\ g\ y\ ?t$

thm REAL_INTEGRABLE_SPIKE_FINITE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ FINITE\ s \wedge (\forall x::real.\ IN\ x\ (DIFF\ (?t::real \Rightarrow bool)\ s) \longrightarrow g\ x = f\ x) \longrightarrow real\_integrable\_on\ f\ ?t \longrightarrow real\_integrable\_on\ g\ ?t$

thm REAL_NEGLIGIBLE_FRONTIER_INTERVAL:

$\forall (a::real)\ b::real.\ real\_negligible\ (DIFF\ (closed\_real\_interval\ [(a,\ b)])\ (open\_real\_interval\ (a,\ b)))$

thm HAS_REAL_INTEGRAL_SPIKE_INTERIOR:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ (b::real)\ y::real.\ (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow g\ x = f\ x) \wedge has\_real\_integral\ f\ y\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_integral\ g\ y\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_REAL_INTEGRAL_SPIKE_INTERIOR_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ (b::real)\ y::real.\ (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow g\ x = f\ x) \longrightarrow has\_real\_integral\ f\ y\ (closed\_real\_interval\ [(a,\ b)]) = has\_real\_integral\ g\ y\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_SPIKE_INTERIOR:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow g\ x = f\ x) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_integrable\_on\ g\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRAL_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \longrightarrow real\_integral\ s\ f = real\_integral\ s\ g$

thm REAL_INTEGRAL_EQ_0:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = (0::real)) \longrightarrow real\_integral\ s\ f = (0::real)$

thm REAL_INTEGRABLE_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_FUNDAMENTAL_THEOREM_OF_CALCULUS:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a \le b \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow has\_real\_integral\ f'\ (f\ b - f\ a)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_SUBINTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ (c::real)\ d::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge SUBSET\ (closed\_real\_interval\ [(c,\ d)])\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(c,\ d)])$

thm HAS_REAL_INTEGRAL_COMBINE:

$\forall (f::real \Rightarrow real)\ (i::real)\ (j::real)\ (a::real)\ (b::real)\ c::real.\ a \le c \wedge c \le b \wedge has\_real\_integral\ f\ i\ (closed\_real\_interval\ [(a,\ c)]) \wedge has\_real\_integral\ f\ j\ (closed\_real\_interval\ [(c,\ b)]) \longrightarrow has\_real\_integral\ f\ (i + j)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRAL_COMBINE:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ a \le c \wedge c \le b \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_integral\ (closed\_real\_interval\ [(a,\ c)])\ f + real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f = real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f$

thm REAL_INTEGRABLE_COMBINE:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ a \le c \wedge c \le b \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ c)]) \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(c,\ b)]) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_ON_LITTLE_SUBINTERVALS:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists d{>}0::real.\ \forall (u::real)\ v::real.\ IN\ x\ (closed\_real\_interval\ [(u,\ v)]) \wedge (\forall y::real.\ IN\ y\ (closed\_real\_interval\ [(u,\ v)]) \longrightarrow |y - x| < d \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)])) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(u,\ v)]))) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRAL_HAS_REAL_DERIVATIVE:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ (\lambda u::real.\ real\_integral\ (closed\_real\_interval\ [(a,\ u)])\ f)\ (f\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)])))$

thm REAL_ANTIDERIVATIVE_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists g::real \Rightarrow real.\ \forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_derivative\ g\ (f\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)])))$

thm REAL_ANTIDERIVATIVE_INTEGRAL_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists\, g::real \Rightarrow real.\ \forall (u::real)\ v::real.\ IN\ u\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ v\ (closed\_real\_interval\ [(a,\ b)]) \wedge u \leq v \longrightarrow has\_real\_integral\, f\ (g\ v\ -\ g\ u)\ (closed\_real\_interval\ [(u,\ v)]))$

thm HAS_REAL_INTEGRAL_AFFINITY:

$\forall (f::real \Rightarrow real)\ (i::real)\ (a::real)\ (b::real)\ (m::real)\ c::real.\ has\_real\_integral\, f\ i\ (closed\_real\_interval\ [(a,\ b)]) \wedge m \neq (0::real) \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ (m * x + c))\ (inverse\_class.inverse\ |m| * i)\ (IMAGE\ (\lambda x::real.\ inverse\_class.inverse\ m * (x\ -\ c))\ (closed\_real\_interval\ [(a,\ b)]))$

thm REAL_INTEGRABLE_AFFINITY:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ (m::real)\ c::real.\ real\_integrable\_on\, f\ (closed\_real\_interval\ [(a,\ b)]) \wedge m \neq (0::real) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ f\ (m * x + c))\ (IMAGE\ (\lambda x::real.\ inverse\_class.inverse\ m * (x\ -\ c))\ (closed\_real\_interval\ [(a,\ b)]))$

thm HAS_REAL_INTEGRAL_STRETCH:

$\forall (f::real \Rightarrow real)\ (i::real)\ (a::real)\ (b::real)\ m::real.\ has\_real\_integral\, f\, i\ (closed\_real\_interval\ [(a,\ b)]) \wedge m \neq (0::real) \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ (m * x))\ (inverse\_class.inverse\ |m| * i)\ (IMAGE\ (op\ *\ (inverse\_class.inverse\ m))\ (closed\_real\_interval\ [(a,\ b)]))$

thm REAL_INTEGRABLE_STRETCH:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ m::real.\ real\_integrable\_on\, f\ (closed\_real\_interval\ [(a,\ b)]) \wedge m \neq (0::real) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ f\ (m * x))\ (IMAGE\ (op\ *\ (inverse\_class.inverse\ m))\ (closed\_real\_interval\ [(a,\ b)]))$

thm HAS_REAL_INTEGRAL_REFLECT_LEMMA:

$\forall (f::real \Rightarrow real)\ (i::real)\ (a::real)\ b::real.\ has\_real\_integral\, f\, i\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow has\_real\_integral\ (\lambda x::real.\ f\ (-\ x))\ i\ (closed\_real\_interval\ [(-\ b,\ -\ a)])$

thm HAS_REAL_INTEGRAL_REFLECT:

$\forall (f::real \Rightarrow real)\ (i::real)\ (a::real)\ b::real.\ has\_real\_integral\ (\lambda x::real.\ f\ (-\ x))\ i\ (closed\_real\_interval\ [(-\ b,\ -\ a)]) = has\_real\_integral\, f\, i\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_REFLECT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ (\lambda x::real.\ f\ (-\ x))\ (closed\_real\_interval\ [(-\ b,\ -\ a)]) = real\_integrable\_on\, f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRAL_REFLECT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integral\ (closed\_real\_interval\ [(-\ b,\ -\ a)])\ (\lambda x::real.\ f\ (-\ x)) = real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f$

thm REAL_FUNDAMENTAL_THEOREM_OF_CALCULUS_INTERIOR:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (a::real)\ b::real.\ a \leq b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x)) \longrightarrow has\_real\_integral\ f'\ (f\ b - f\ a)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_FUNDAMENTAL_THEOREM_OF_CALCULUS_INTERIOR_STRONG:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ COUNTABLE\ s \wedge a \leq b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (DIFF\ (open\_real\_interval\ (a,\ b))\ s) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x)) \longrightarrow has\_real\_integral\ f'\ (f\ b - f\ a)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_FUNDAMENTAL_THEOREM_OF_CALCULUS_STRONG:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ COUNTABLE\ s \wedge a \leq b \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (DIFF\ (closed\_real\_interval\ [(a,\ b)])\ s) \longrightarrow has\_real\_derivative\ f\ (f'\ x)\ (atreal\ x)) \longrightarrow has\_real\_integral\ f'\ (f\ b - f\ a)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INDEFINITE_INTEGRAL_CONTINUOUS_RIGHT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_continuous\_on\ (\lambda x::real.\ real\_integral\ (closed\_real\_interval\ [(a,\ x)])\ f)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INDEFINITE_INTEGRAL_CONTINUOUS_LEFT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_continuous\_on\ (\lambda x::real.\ real\_integral\ (closed\_real\_interval\ [(x,\ b)])\ f)\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_REAL_DERIVATIVE_ZERO_UNIQUE_STRONG_INTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ (k::real \Rightarrow bool)\ y::real.\ COUNTABLE\ k \wedge real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge f\ a = y \wedge (\forall x::real.\ IN\ x\ (DIFF\ (closed\_real\_interval\ [(a,\ b)])\ k) \longrightarrow has\_real\_derivative\ f\ (0::real)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))) \longrightarrow (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow f\ x = y)$

thm HAS_REAL_DERIVATIVE_ZERO_UNIQUE_STRONG_CONVEX:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (k::real \Rightarrow bool)\ (c::real)\ y::real.\ is\_realinterval\ s \wedge COUNTABLE\ k \wedge real\_continuous\_on\ f\ s \wedge IN\ c\ s \wedge f\ c = y \wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow has\_real\_derivative\ f\ (0::real)\ (within\ (atreal\ x)\ s)) \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = y)$

thm HAS_REAL_DERIVATIVE_INDEFINITE_INTEGRAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists k::real \Rightarrow bool.\ real\_negligible\ k \wedge (\forall x::real.\ IN\ x\ (DIFF\ (closed\_real\_interval$

$[(a,\ b)])\ k) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ real\_integral\ (closed\_real\_interval\ [(a,\ x)])\ f)\ (f\ x)\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))))$

thm HAS_REAL_INTEGRAL_RESTRICT:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow has\_real\_integral\ (\lambda x::real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::real))\ (?i::real)\ t = has\_real\_integral\ f\ ?i\ s$

thm HAS_REAL_INTEGRAL_RESTRICT_UNIV:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ i::real.\ has\_real\_integral\ (\lambda x::real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::real))\ i\ HOL\_Light\_Import.UNIV = has\_real\_integral\ f\ i\ s$

thm HAS_REAL_INTEGRAL_SPIKE_SET_EQ:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ y::real.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow has\_real\_integral\ f\ y\ s = has\_real\_integral\ f\ y\ t$

thm HAS_REAL_INTEGRAL_SPIKE_SET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ y::real.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \wedge has\_real\_integral\ f\ y\ s \longrightarrow has\_real\_integral\ f\ y\ t$

thm REAL_INTEGRABLE_SPIKE_SET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ f\ t$

thm REAL_INTEGRABLE_SPIKE_SET_EQ:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_integrable\_on\ f\ s = real\_integrable\_on\ f\ t$

thm REAL_INTEGRAL_SPIKE_SET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_integral\ s\ f = real\_integral\ t\ f$

thm HAS_REAL_INTEGRAL_OPEN_INTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ y::real.\ has\_real\_integral\ f\ y\ (open\_real\_interval\ (a,\ b)) = has\_real\_integral\ f\ y\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_ON_OPEN_INTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (open\_real\_interval\ (a,\ b)) = real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRAL_OPEN_INTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integral\ (open\_real\_interval\ (a,\ b))\ f = real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ f$

thm HAS_REAL_INTEGRAL_ON_SUPERSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ (\forall x::real.\ \neg\ IN\ x\ s \longrightarrow f\ x = (0::real)) \wedge SUBSET\ s\ t \wedge has\_real\_integral\ f\ (?i::real)\ s \longrightarrow has\_real\_integral\ f\ ?i\ t$

thm REAL_INTEGRABLE_ON_SUPERSET:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ (\forall x{::}real.\ \neg\ IN\ x\ s \longrightarrow f\ x = (0{::}real)) \wedge SUBSET\ s\ t \wedge real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ f\ t$

thm REAL_INTEGRABLE_RESTRICT_UNIV:

$\forall (f{::}real \Rightarrow real)\ s{::}real \Rightarrow bool.\ real\_integrable\_on\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real))\ HOL\_Light\_Import.UNIV = real\_integrable\_on\ f\ s$

thm REAL_INTEGRAL_RESTRICT_UNIV:

$\forall (f{::}real \Rightarrow real)\ s{::}real \Rightarrow bool.\ real\_integral\ HOL\_Light\_Import.UNIV\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real)) = real\_integral\ s\ f$

thm REAL_INTEGRAL_RESTRICT:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ SUBSET\ s\ t \longrightarrow real\_integral\ t\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real)) = real\_integral\ s\ f$

thm HAS_REAL_INTEGRAL_RESTRICT_INTER:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ has\_real\_integral\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real))\ (?i{::}real)\ t = has\_real\_integral\ f\ ?i\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_INTEGRAL_RESTRICT_INTER:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_integral\ t\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real)) = real\_integral\ (HOL\_Light\_Import.INTER\ s\ t)\ f$

thm REAL_INTEGRABLE_RESTRICT_INTER:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_integrable\_on\ (\lambda x{::}real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0{::}real))\ t = real\_integrable\_on\ f\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_NEGLIGIBLE_ON_INTERVALS:

$\forall s{::}real \Rightarrow bool.\ real\_negligible\ s = (\forall (a{::}real)\ b{::}real.\ real\_negligible\ (HOL\_Light\_Import.INTER\ s\ (closed\_real\_interval\ [(a,\ b)])))$

thm HAS_REAL_INTEGRAL_SUBSET_LE:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ (t{::}real \Rightarrow bool)\ (i{::}real)\ j{::}real.\ SUBSET\ s\ t \wedge has\_real\_integral\ f\ i\ s \wedge has\_real\_integral\ f\ j\ t \wedge (\forall x{::}real.\ IN\ x\ t \longrightarrow (0{::}real) \leq f\ x) \longrightarrow i \leq j$

thm REAL_INTEGRAL_SUBSET_LE:

$\forall (f{::}real \Rightarrow real)\ (s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ SUBSET\ s\ t \wedge real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ f\ t \wedge (\forall x{::}real.\ IN\ x\ t \longrightarrow (0{::}real) \leq f\ x) \longrightarrow real\_integral\ s\ f \leq real\_integral\ t\ f$

thm REAL_INTEGRABLE_ON_SUBINTERVAL:

$\forall (f::real \Rightarrow real)$ $(s::real \Rightarrow bool)$ $(a::real)$ $b::real.$ *real_integrable_on f s* $\wedge$ *SUBSET* (*closed_real_interval* $[(a,\ b)]$) $s \longrightarrow$ *real_integrable_on f* (*closed_real_interval* $[(a,\ b)]$)

thm REAL_INTEGRABLE_STRADDLE:

$\forall (f::real \Rightarrow real)$ $s::real \Rightarrow bool.$ $(\forall e{>}0::real.$ $\exists (g::real \Rightarrow real)$ $(h::real \Rightarrow real)$ $(i::real)$ $j::real.$ *has_real_integral g i s* $\wedge$ *has_real_integral h j s* $\wedge$ $|i - j| < e \wedge (\forall x::real.$ *IN x s* $\longrightarrow g\ x \leq f\ x \wedge f\ x \leq h\ x)) \longrightarrow$ *real_integrable_on f s*

thm HAS_REAL_INTEGRAL_STRADDLE_NULL:

$\forall (f::real \Rightarrow real)$ $(g::real \Rightarrow real)$ $s::real \Rightarrow bool.$ $(\forall x::real.$ *IN x s* $\longrightarrow (0::real) \leq f\ x \wedge f\ x \leq g\ x) \wedge$ *has_real_integral g* $(0::real)$ $s \longrightarrow$ *has_real_integral f* $(0::real)$ s

thm HAS_REAL_INTEGRAL_UNION:

$\forall (f::real \Rightarrow real)$ $(i::real)$ $(j::real)$ $(s::real \Rightarrow bool)$ $t::real \Rightarrow bool.$ *has_real_integral f i s* $\wedge$ *has_real_integral f j t* $\wedge$ *real_negligible* (*HOL_Light_Import.INTER s t*) $\longrightarrow$ *has_real_integral f* $(i + j)$ (*HOL_Light_Import.UNION s t*)

thm HAS_REAL_INTEGRAL_UNIONS:

$\forall (f::real \Rightarrow real)$ $(i::(real \Rightarrow bool) \Rightarrow real)$ $t::(real \Rightarrow bool) \Rightarrow bool.$ *FINITE t* $\wedge (\forall s::real \Rightarrow bool.$ *IN s t* $\longrightarrow$ *has_real_integral f* $(i\ s)\ s) \wedge (\forall (s::real \Rightarrow bool)$ $s'::real \Rightarrow bool.$ *IN s t* $\wedge$ *IN s' t* $\wedge s \neq s' \longrightarrow$ *real_negligible* (*HOL_Light_Import.INTER s s'*)) $\longrightarrow$ *has_real_integral f* (*sum t i*) (*UNIONS t*)

thm REAL_MONOTONE_CONVERGENCE_INCREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)$ $(g::real \Rightarrow real)$ $s::real \Rightarrow bool.$ $(\forall k::nat.$ *real_integrable_on* $(f\ k)\ s) \wedge (\forall (k::nat)\ x::real.$ *IN x s* $\longrightarrow f\ k\ x \leq f\ (Suc\ k)\ x) \wedge (\forall x::real.$ *IN x s* $\longrightarrow$ ---> $(\lambda k::nat.\ f\ k\ x)$ $(g\ x)$ *sequentially*) $\wedge$ *real_bounded* (*GSPEC* (λ*GEN%PVAR%2455*::*real*. $\exists k::nat.$ *SETSPEC GEN%PVAR%2455* (*IN k HOL_Light_Import.UNIV*) (*real_integral s* $(f\ k)$)))) $\longrightarrow$ *real_integrable_on g s* $\wedge$ ---> $(\lambda k::nat.$ *real_integral s* $(f\ k))$ (*real_integral s g*) *sequentially*

thm REAL_MONOTONE_CONVERGENCE_DECREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)$ $(g::real \Rightarrow real)$ $s::real \Rightarrow bool.$ $(\forall k::nat.$ *real_integrable_on* $(f\ k)\ s) \wedge (\forall (k::nat)\ x::real.$ *IN x s* $\longrightarrow f\ (Suc\ k)\ x \leq f\ k\ x) \wedge (\forall x::real.$ *IN x s* $\longrightarrow$ ---> $(\lambda k::nat.\ f\ k\ x)$ $(g\ x)$ *sequentially*) $\wedge$ *real_bounded* (*GSPEC* (λ*GEN%PVAR%2456*::*real*. $\exists k::nat.$ *SETSPEC GEN%PVAR%2456* (*IN k HOL_Light_Import.UNIV*) (*real_integral s* $(f\ k)$)))) $\longrightarrow$ *real_integrable_on g s* $\wedge$ ---> $(\lambda k::nat.$ *real_integral s* $(f\ k))$ (*real_integral s g*) *sequentially*

thm REAL_BEPPO_LEVI_INCREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)$ $s::real \Rightarrow bool.$ $(\forall k::nat.$ *real_integrable_on* $(f\ k)\ s)$ $\wedge (\forall (k::nat)\ x::real.$ *IN x s* $\longrightarrow f\ k\ x \leq f\ (Suc\ k)\ x) \wedge$ *real_bounded* (*GSPEC* (λ*GEN%PVAR%2457*::*real*. $\exists k::nat.$ *SETSPEC GEN%PVAR%2457* (*IN k HOL_Light_Import.UNIV*)

$(real\_integral\ s\ (f\ k)))) \longrightarrow (\exists (g::real \Rightarrow real)\ k::real \Rightarrow bool.\ real\_negligible\ k$
$\wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow ---> (\lambda k::nat.\ f\ k\ x)\ (g\ x)\ sequentially))$

thm REAL_BEPPO_LEVI_DECREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall k::nat.\ real\_integrable\_on\ (f\ k)\ s)$
$\wedge (\forall (k::nat)\ x::real.\ IN\ x\ s \longrightarrow f\ (Suc\ k)\ x \leq f\ k\ x) \wedge real\_bounded\ (GSPEC$
$(\lambda GEN\%PVAR\%2458::real.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%2458\ (IN\ k\ HOL\_Light\_Import.UNIV)$
$(real\_integral\ s\ (f\ k)))) \longrightarrow (\exists (g::real \Rightarrow real)\ k::real \Rightarrow bool.\ real\_negligible\ k$
$\wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow ---> (\lambda k::nat.\ f\ k\ x)\ (g\ x)\ sequentially))$

thm REAL_BEPPO_LEVI_MONOTONE_CONVERGENCE_INCREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall k::nat.\ real\_integrable\_on\ (f\ k)\ s)$
$\wedge (\forall (k::nat)\ x::real.\ IN\ x\ s \longrightarrow f\ k\ x \leq f\ (Suc\ k)\ x) \wedge real\_bounded\ (GSPEC$
$(\lambda GEN\%PVAR\%2459::real.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%2459\ (IN\ k\ HOL\_Light\_Import.UNIV)$
$(real\_integral\ s\ (f\ k)))) \longrightarrow (\exists (g::real \Rightarrow real)\ k::real \Rightarrow bool.\ real\_negligible\ k$
$\wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow ---> (\lambda k::nat.\ f\ k\ x)\ (g\ x)\ sequentially) \wedge$
$real\_integrable\_on\ g\ s \wedge ---> (\lambda k::nat.\ real\_integral\ s\ (f\ k))\ (real\_integral\ s$
$g)\ sequentially)$

thm REAL_BEPPO_LEVI_MONOTONE_CONVERGENCE_DECREASING:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall k::nat.\ real\_integrable\_on\ (f\ k)\ s)$
$\wedge (\forall (k::nat)\ x::real.\ IN\ x\ s \longrightarrow f\ (Suc\ k)\ x \leq f\ k\ x) \wedge real\_bounded\ (GSPEC$
$(\lambda GEN\%PVAR\%2460::real.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%2460\ (IN\ k\ HOL\_Light\_Import.UNIV)$
$(real\_integral\ s\ (f\ k)))) \longrightarrow (\exists (g::real \Rightarrow real)\ k::real \Rightarrow bool.\ real\_negligible\ k$
$\wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow ---> (\lambda k::nat.\ f\ k\ x)\ (g\ x)\ sequentially) \wedge$
$real\_integrable\_on\ g\ s \wedge ---> (\lambda k::nat.\ real\_integral\ s\ (f\ k))\ (real\_integral\ s$
$g)\ sequentially)$

thm REAL_INTEGRAL_ABS_BOUND_INTEGRAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \wedge$
$real\_integrable\_on\ g\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x| \leq g\ x) \longrightarrow |real\_integral\ s$
$f| \leq real\_integral\ s\ g$

thm ABSOLUTELY_REAL_INTEGRABLE_LE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow |real\_integral$
$s\ f| \leq real\_integral\ s\ (\lambda x::real.\ |f\ x|)$

thm ABSOLUTELY_REAL_INTEGRABLE_0:

$\forall s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ (\lambda x::real.\ 0::real)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_CONST:

$\forall (a::real)\ (b::real)\ c::real.\ absolutely\_real\_integrable\_on\ (\lambda x::real.\ c)\ (closed\_real\_interval$
$[(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_LMUL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ c::real.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow$
$absolutely\_real\_integrable\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_RMUL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ c::real.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ -\ f\ x)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ABS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ |f\ x|)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ON_SUBINTERVAL:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ absolutely\_real\_integrable\_on\ f\ s \wedge SUBSET\ (closed\_real\_interval\ [(a,\ b)])\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_RESTRICT_UNIV:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ (\lambda x::real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::real))\ HOL\_Light\_Import.UNIV = absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x - g\ x)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_LINEAR:

$\forall (f::real \Rightarrow real)\ (h::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \wedge linear\ (lift \circ (h \circ HOL\_Light\_Import.drop)) \longrightarrow absolutely\_real\_integrable\_on\ (h \circ f)\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_SUM:

$\forall (f::?'a::type \Rightarrow real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ t \wedge (\forall a::?'a::type.\ IN\ a\ t \longrightarrow absolutely\_real\_integrable\_on\ (f\ a)\ s) \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ sum\ t\ (\lambda a::?'a::type.\ f\ a\ x))\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_MAX:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ max\ (f\ x)\ (g\ x))\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_MIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ min\ (f\ x)\ (g\ x))\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_IMP_INTEGRABLE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s \longrightarrow real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow absolutely\_real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm NONNEGATIVE_ABSOLUTELY_REAL_INTEGRABLE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \wedge real\_integrable\_on\ f\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_INTEGRABLE_BOUND:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x| \leq g\ x) \wedge real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ABSOLUTELY_REAL_INTEGRABLE_BOUND:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x| \leq |g\ x|) \wedge real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ABSOLUTELY_REAL_INTEGRABLE_UBOUND:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \wedge real\_integrable\_on\ f\ s \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_ABSOLUTELY_REAL_INTEGRABLE_LBOUND:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x \leq g\ x) \wedge absolutely\_real\_integrable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_INF:

$\forall (fs::real \Rightarrow ?'a::type \Rightarrow real)\ (s::real \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge k \neq EMPTY \wedge (\forall i::?'a::type.\ IN\ i\ k \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ fs\ x\ i)\ s) \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ HOL\_Light\_Import.inf\ (IMAGE\ (fs\ x)\ k))\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_SUP:

$\forall (fs::real \Rightarrow ?'a::type \Rightarrow real)\ (s::real \Rightarrow bool)\ k::?'a::type \Rightarrow bool.\ FINITE\ k \wedge k \neq EMPTY \wedge (\forall i::?'a::type.\ IN\ i\ k \longrightarrow absolutely\_real\_integrable\_on$

(λx::*real*. *fs x i*) *s*) $\longrightarrow$ *absolutely_real_integrable_on* (λx::*real*. *HOL_Light_Import.sup* (*IMAGE* (*fs x*) *k*)) *s*

thm REAL_DOMINATED_CONVERGENCE:

$\forall$ (*f*::*nat* $\Rightarrow$ *real* $\Rightarrow$ *real*) (*g*::*real* $\Rightarrow$ *real*) (*h*::*real* $\Rightarrow$ *real*) *s*::*real* $\Rightarrow$ *bool*. ($\forall$ *k*::*nat*. *real_integrable_on* (*f k*) *s*) $\wedge$ *real_integrable_on h s* $\wedge$ ($\forall$ (*k*::*nat*) *x*::*real*. *IN x s* $\longrightarrow$ $|f\,k\,x| \leq h\,x$) $\wedge$ ($\forall$ *x*::*real*. *IN x s* $\longrightarrow$ −−−> (λk::*nat*. *f k x*) (*g x*) *sequentially*) $\longrightarrow$ *real_integrable_on g s* $\wedge$ −−−> (λk::*nat*. *real_integral s* (*f k*)) (*real_integral s g*) *sequentially*

thm HAS_REAL_MEASURE_MEASURE:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_measurable s* = *has_real_measure s* (*real_measure s*)

thm HAS_REAL_MEASURE_UNIQUE:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) (*m1*::*real*) *m2*::*real*. *has_real_measure s m1* $\wedge$ *has_real_measure s m2* $\longrightarrow$ *m1* = *m2*

thm REAL_MEASURE_UNIQUE:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *m*::*real*. *has_real_measure s m* $\longrightarrow$ *real_measure s* = *m*

thm HAS_REAL_MEASURE_REAL_MEASURABLE_REAL_MEASURE:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *m*::*real*. *has_real_measure s m* = (*real_measurable s* $\wedge$ *real_measure s* = *m*)

thm HAS_REAL_MEASURE_IMP_REAL_MEASURABLE:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *m*::*real*. *has_real_measure s m* $\longrightarrow$ *real_measurable s*

thm HAS_REAL_MEASURE:

$\forall$ (*s*::*real* $\Rightarrow$ *bool*) *m*::*real*. *has_real_measure s m* = *has_real_integral* (λx::*real*. *if IN x s then 1*::*real else* (*0*::*real*)) *m HOL_Light_Import.UNIV*

thm REAL_MEASURABLE:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_measurable s* = *real_integrable_on* (λx::*real*. *1*::*real*) *s*

thm REAL_MEASURABLE_REAL_INTEGRABLE:

real_measurable (*?s*::*real* $\Rightarrow$ *bool*) = *real_integrable_on* (λx::*real*. *if IN x ?s then 1*::*real else* (*0*::*real*)) *HOL_Light_Import.UNIV*

thm REAL_MEASURE_REAL_INTEGRAL:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_measurable s* $\longrightarrow$ *real_measure s* = *real_integral s* (λx::*real*. *1*::*real*)

thm REAL_MEASURE_REAL_INTEGRAL_UNIV:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_measurable s* $\longrightarrow$ *real_measure s* = *real_integral HOL_Light_Import.UNIV* (λx::*real*. *if IN x s then 1*::*real else* (*0*::*real*))

thm REAL_INTEGRAL_REAL_MEASURE:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s \longrightarrow real\_integral\ s\ (\lambda x::real.\ 1::real) = real\_measure\ s$

thm REAL_INTEGRAL_REAL_MEASURE_UNIV:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s \longrightarrow real\_integral\ HOL\_Light\_Import.UNIV\ (\lambda x::real.\ if\ IN\ x\ s\ then\ 1::real\ else\ (0::real)) = real\_measure\ s$

thm HAS_REAL_MEASURE_HAS_MEASURE:

$\forall (s::real \Rightarrow bool)\ m::real.\ has\_real\_measure\ s\ m = has\_measure\ (IMAGE\ lift\ s)\ m$

thm REAL_MEASURABLE_MEASURABLE:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s = measurable\ (IMAGE\ lift\ s)$

thm REAL_MEASURE_MEASURE:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s \longrightarrow real\_measure\ s = HOL\_Light\_Import.measure\ (IMAGE\ lift\ s)$

thm HAS_REAL_MEASURE_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ has\_real\_measure\ (closed\_real\_interval\ [(a,\ b)])\ (max\ (b - a)\ (0::real))) \wedge (\forall (a::real)\ b::real.\ has\_real\_measure\ (open\_real\_interval\ (a,\ b))\ (max\ (b - a)\ (0::real)))$

thm REAL_MEASURABLE_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ real\_measurable\ (closed\_real\_interval\ [(a,\ b)])) \wedge (\forall (a::real)\ b::real.\ real\_measurable\ (open\_real\_interval\ (a,\ b)))$

thm HAS_REAL_MEASURE_REAL_INTERVAL_conjunct1:

$\forall (a::real)\ b::real.\ has\_real\_measure\ (open\_real\_interval\ (a,\ b))\ (max\ (b - a)\ (0::real))$

thm HAS_REAL_MEASURE_REAL_INTERVAL_conjunct0:

$\forall (a::real)\ b::real.\ has\_real\_measure\ (closed\_real\_interval\ [(a,\ b)])\ (max\ (b - a)\ (0::real))$

thm REAL_MEASURE_REAL_INTERVAL_conjunct1:

$\forall (a::real)\ b::real.\ real\_measure\ (open\_real\_interval\ (a,\ b)) = max\ (b - a)\ (0::real)$

thm REAL_MEASURE_REAL_INTERVAL_conjunct0:

$\forall (a::real)\ b::real.\ real\_measure\ (closed\_real\_interval\ [(a,\ b)]) = max\ (b - a)\ (0::real)$

thm REAL_MEASURE_REAL_INTERVAL:

$(\forall (a::real)\ b::real.\ real\_measure\ (closed\_real\_interval\ [(a,\ b)]) = max\ (b - a)\ (0::real)) \wedge (\forall (a::real)\ b::real.\ real\_measure\ (open\_real\_interval\ (a,\ b)) = max\ (b - a)\ (0::real))$

thm REAL_MEASURABLE_INTER:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_measurable\ t \longrightarrow real\_measurable\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_MEASURABLE_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_measurable\ t \longrightarrow real\_measurable\ (HOL\_Light\_Import.UNION\ s\ t)$

thm HAS_REAL_MEASURE_DISJOINT_UNION:

$\forall (s1::real \Rightarrow bool)\ (s2::real \Rightarrow bool)\ (m1::real)\ m2::real.\ has\_real\_measure\ s1\ m1 \wedge has\_real\_measure\ s2\ m2 \wedge DISJOINT\ s1\ s2 \longrightarrow has\_real\_measure\ (HOL\_Light\_Import.UNION\ s1\ s2)\ (m1 + m2)$

thm REAL_MEASURE_DISJOINT_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_measurable\ t \wedge DISJOINT\ s\ t \longrightarrow real\_measure\ (HOL\_Light\_Import.UNION\ s\ t) = real\_measure\ s + real\_measure\ t$

thm HAS_REAL_MEASURE_POS_LE:

$\forall (m::real)\ s::real \Rightarrow bool.\ has\_real\_measure\ s\ m \longrightarrow (0::real) \leq m$

thm REAL_MEASURE_POS_LE:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s \longrightarrow (0::real) \leq real\_measure\ s$

thm HAS_REAL_MEASURE_SUBSET:

$\forall (s1::real \Rightarrow bool)\ (s2::real \Rightarrow bool)\ (m1::real)\ m2::real.\ has\_real\_measure\ s1\ m1 \wedge has\_real\_measure\ s2\ m2 \wedge SUBSET\ s1\ s2 \longrightarrow m1 \leq m2$

thm REAL_MEASURE_SUBSET:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_measurable\ t \wedge SUBSET\ s\ t \longrightarrow real\_measure\ s \leq real\_measure\ t$

thm HAS_REAL_MEASURE_0:

$\forall s::real \Rightarrow bool.\ has\_real\_measure\ s\ (0::real) = real\_negligible\ s$

thm REAL_MEASURE_EQ_0:

$\forall s::real \Rightarrow bool.\ real\_negligible\ s \longrightarrow real\_measure\ s = (0::real)$

thm HAS_REAL_MEASURE_EMPTY:

$has\_real\_measure\ EMPTY\ (0::real)$

thm REAL_MEASURE_EMPTY:

$real\_measure\ EMPTY = (0::real)$

thm REAL_MEASURABLE_EMPTY:

$real\_measurable\ EMPTY$

thm REAL_MEASURABLE_REAL_MEASURE_EQ_0:

$\forall$ *s::real* $\Rightarrow$ *bool. real_measurable s* $\longrightarrow$ (*real_measure s* = (*0::real*)) = *real_negligible s*

thm REAL_MEASURABLE_REAL_MEASURE_POS_LT:

$\forall$ *s::real* $\Rightarrow$ *bool. real_measurable s* $\longrightarrow$ ((*0::real*) < *real_measure s*) = ($\neg$ *real_negligible s*)

thm REAL_NEGLIGIBLE_REAL_INTERVAL:

($\forall$ (*a::real*) *b::real. real_negligible* (*closed_real_interval* [(*a*, *b*)]) = (*open_real_interval* (*a*, *b*) = *EMPTY*)) $\wedge$ ($\forall$ (*a::real*) *b::real. real_negligible* (*open_real_interval* (*a*, *b*)) = (*open_real_interval* (*a*, *b*) = *EMPTY*))

thm REAL_MEASURABLE_UNIONS:

$\forall$ *f::*(*real* $\Rightarrow$ *bool*) $\Rightarrow$ *bool. FINITE f* $\wedge$ ($\forall$ *s::real* $\Rightarrow$ *bool. IN s f* $\longrightarrow$ *real_measurable s*) $\longrightarrow$ *real_measurable* (*UNIONS f*)

thm HAS_REAL_MEASURE_DIFF_SUBSET:

$\forall$ (*s1::real* $\Rightarrow$ *bool*) (*s2::real* $\Rightarrow$ *bool*) (*m1::real*) *m2::real. has_real_measure s1 m1* $\wedge$ *has_real_measure s2 m2* $\wedge$ *SUBSET s2 s1* $\longrightarrow$ *has_real_measure* (*DIFF s1 s2*) (*m1* − *m2*)

thm REAL_MEASURABLE_DIFF:

$\forall$ (*s::real* $\Rightarrow$ *bool*) *t::real* $\Rightarrow$ *bool. real_measurable s* $\wedge$ *real_measurable t* $\longrightarrow$ *real_measurable* (*DIFF s t*)

thm REAL_MEASURE_DIFF_SUBSET:

$\forall$ (*s::real* $\Rightarrow$ *bool*) *t::real* $\Rightarrow$ *bool. real_measurable s* $\wedge$ *real_measurable t* $\wedge$ *SUBSET t s* $\longrightarrow$ *real_measure* (*DIFF s t*) = *real_measure s* − *real_measure t*

thm HAS_REAL_MEASURE_UNION_REAL_NEGLIGIBLE:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*t::real* $\Rightarrow$ *bool*) *m::real. has_real_measure s m* $\wedge$ *real_negligible t* $\longrightarrow$ *has_real_measure* (*HOL_Light_Import.UNION s t*) *m*

thm HAS_REAL_MEASURE_DIFF_REAL_NEGLIGIBLE:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*t::real* $\Rightarrow$ *bool*) *m::real. has_real_measure s m* $\wedge$ *real_negligible t* $\longrightarrow$ *has_real_measure* (*DIFF s t*) *m*

thm HAS_REAL_MEASURE_UNION_REAL_NEGLIGIBLE_EQ:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*t::real* $\Rightarrow$ *bool*) *m::real. real_negligible t* $\longrightarrow$ *has_real_measure* (*HOL_Light_Import.UNION s t*) *m* = *has_real_measure s m*

thm HAS_REAL_MEASURE_DIFF_REAL_NEGLIGIBLE_EQ:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*t::real* $\Rightarrow$ *bool*) *m::real. real_negligible t* $\longrightarrow$ *has_real_measure* (*DIFF s t*) *m* = *has_real_measure s m*

thm HAS_REAL_MEASURE_ALMOST:

$\forall (s::real \Rightarrow bool)\ (s'::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ m::real.\ has\_real\_measure\ s\ m \wedge real\_negligible\ t \wedge HOL\_Light\_Import.UNION\ s\ t = HOL\_Light\_Import.UNION\ s'\ t \longrightarrow has\_real\_measure\ s'\ m$

thm HAS_REAL_MEASURE_ALMOST_EQ:

$\forall (s::real \Rightarrow bool)\ (s'::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ t \wedge HOL\_Light\_Import.UNION\ s\ t = HOL\_Light\_Import.UNION\ s'\ t \longrightarrow has\_real\_measure\ s\ (?m::real) = has\_real\_measure\ s'\ ?m$

thm REAL_MEASURABLE_ALMOST:

$\forall (s::real \Rightarrow bool)\ (s'::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_negligible\ t \wedge HOL\_Light\_Import.UNION\ s\ t = HOL\_Light\_Import.UNION\ s'\ t \longrightarrow real\_measurable\ s'$

thm HAS_REAL_MEASURE_REAL_NEGLIGIBLE_UNION:

$\forall (s1::real \Rightarrow bool)\ (s2::real \Rightarrow bool)\ (m1::real)\ m2::real.\ has\_real\_measure\ s1\ m1 \wedge has\_real\_measure\ s2\ m2 \wedge real\_negligible\ (HOL\_Light\_Import.INTER\ s1\ s2) \longrightarrow has\_real\_measure\ (HOL\_Light\_Import.UNION\ s1\ s2)\ (m1 + m2)$

thm REAL_MEASURE_REAL_NEGLIGIBLE_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_measurable\ t \wedge real\_negligible\ (HOL\_Light\_Import.INTER\ s\ t) \longrightarrow real\_measure\ (HOL\_Light\_Import.UNION\ s\ t) = real\_measure\ s + real\_measure\ t$

thm HAS_REAL_MEASURE_REAL_NEGLIGIBLE_SYMDIFF:

$\forall (s::real \Rightarrow bool)\ (t::real \Rightarrow bool)\ m::real.\ has\_real\_measure\ s\ m \wedge real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow has\_real\_measure\ t\ m$

thm REAL_MEASURABLE_REAL_NEGLIGIBLE_SYMDIFF:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_measurable\ s \wedge real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_measurable\ t$

thm REAL_MEASURE_REAL_NEGLIGIBLE_SYMDIFF:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ (real\_measurable\ s \vee real\_measurable\ t) \wedge real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_measure\ s = real\_measure\ t$

thm HAS_REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS:

$\forall (m::(real \Rightarrow bool) \Rightarrow real)\ f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow has\_real\_measure\ s\ (m\ s)) \wedge (\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ IN\ s\ f \wedge IN\ t\ f \wedge s \neq t \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ s\ t)) \longrightarrow has\_real\_measure\ (UNIONS\ f)\ (sum\ f\ m)$

thm REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS:

$\forall (m::(real \Rightarrow bool) \Rightarrow real)\ f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow has\_real\_measure\ s\ (m\ s)) \wedge (\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ IN\ s\ f \wedge IN\ t\ f \wedge s \neq t \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ s\ t)) \longrightarrow real\_measure\ (UNIONS\ f) = sum\ f\ m$

thm HAS_REAL_MEASURE_DISJOINT_UNIONS:

$\forall (m::(real \Rightarrow bool) \Rightarrow real)\ f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow has\_real\_measure\ s\ (m\ s)) \wedge (\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ IN\ s\ f \wedge IN\ t\ f \wedge s \neq t \longrightarrow DISJOINT\ s\ t) \longrightarrow has\_real\_measure\ (UNIONS\ f)\ (sum\ f\ m)$

thm REAL_MEASURE_DISJOINT_UNIONS:

$\forall (m::(real \Rightarrow bool) \Rightarrow real)\ f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow has\_real\_measure\ s\ (m\ s)) \wedge (\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ IN\ s\ f \wedge IN\ t\ f \wedge s \neq t \longrightarrow DISJOINT\ s\ t) \longrightarrow real\_measure\ (UNIONS\ f) = sum\ f\ m$

thm HAS_REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS_IMAGE:

$\forall (f::?'a::type \Rightarrow real \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow real\_measurable\ (f\ x)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ (f\ x)\ (f\ y))) \longrightarrow has\_real\_measure\ (UNIONS\ (IMAGE\ f\ s))\ (sum\ s\ (\lambda x::?'a::type.\ real\_measure\ (f\ x)))$

thm REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS_IMAGE:

$\forall (f::?'a::type \Rightarrow real \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow real\_measurable\ (f\ x)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ (f\ x)\ (f\ y))) \longrightarrow real\_measure\ (UNIONS\ (IMAGE\ f\ s)) = sum\ s\ (\lambda x::?'a::type.\ real\_measure\ (f\ x))$

thm HAS_REAL_MEASURE_DISJOINT_UNIONS_IMAGE:

$\forall (f::?'a::type \Rightarrow real \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow real\_measurable\ (f\ x)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (f\ x)\ (f\ y)) \longrightarrow has\_real\_measure\ (UNIONS\ (IMAGE\ f\ s))\ (sum\ s\ (\lambda x::?'a::type.\ real\_measure\ (f\ x)))$

thm REAL_MEASURE_DISJOINT_UNIONS_IMAGE:

$\forall (f::?'a::type \Rightarrow real \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow real\_measurable\ (f\ x)) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow DISJOINT\ (f\ x)\ (f\ y)) \longrightarrow real\_measure\ (UNIONS\ (IMAGE\ f\ s)) = sum\ s\ (\lambda x::?'a::type.\ real\_measure\ (f\ x))$

thm HAS_REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS_IMAGE_STRONG:

$\forall (f::?'a::type \Rightarrow real \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2462::?'a::type.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%2462\ (IN\ x\ s \wedge f\ x \neq EMPTY)\ x)) \wedge$

$(\forall x{::}?'a{::}type.\ IN\,x\,s \longrightarrow real\_measurable\ (f\,x)) \land (\forall\,(x{::}?'a{::}type)\ y{::}?'a{::}type.$
$IN\ x\ s \land IN\ y\ s \land x \neq y \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ (f\,x)$
$(f\ y))) \longrightarrow has\_real\_measure\ (UNIONS\ (IMAGE\,f\,s))\ (sum\ s\ (\lambda x{::}?'a{::}type.$
$real\_measure\ (f\,x)))$

thm REAL_MEASURE_REAL_NEGLIGIBLE_UNIONS_IMAGE_STRONG:

$\forall\,(f{::}?'a{::}type \Rightarrow real \Rightarrow bool)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2463{::}?'a{::}type.$
$\exists\,x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%2463\ (IN\,x\,s \land f\,x \neq EMPTY)\ x)) \land$
$(\forall\,x{::}?'a{::}type.\ IN\,x\,s \longrightarrow real\_measurable\ (f\,x)) \land (\forall\,(x{::}?'a{::}type)\ y{::}?'a{::}type.$
$IN\ x\ s \land IN\ y\ s \land x \neq y \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ (f$
$x)\ (f\ y))) \longrightarrow real\_measure\ (UNIONS\ (IMAGE\,f\,s)) = sum\ s\ (\lambda x{::}?'a{::}type.$
$real\_measure\ (f\,x))$

thm HAS_REAL_MEASURE_DISJOINT_UNIONS_IMAGE_STRONG:

$\forall\,(f{::}?'a{::}type \Rightarrow real \Rightarrow bool)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2464{::}?'a{::}type.$
$\exists\,x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%2464\ (IN\,x\,s \land f\,x \neq EMPTY)\ x)) \land$
$(\forall\,x{::}?'a{::}type.\ IN\,x\,s \longrightarrow real\_measurable\ (f\,x)) \land (\forall\,(x{::}?'a{::}type)\ y{::}?'a{::}type.$
$IN\ x\ s \land IN\ y\ s \land x \neq y \longrightarrow DISJOINT\ (f\ x)\ (f\ y)) \longrightarrow has\_real\_measure$
$(UNIONS\ (IMAGE\,f\,s))\ (sum\ s\ (\lambda x{::}?'a{::}type.\ real\_measure\ (f\,x)))$

thm REAL_MEASURE_DISJOINT_UNIONS_IMAGE_STRONG:

$\forall\,(f{::}?'a{::}type \Rightarrow real \Rightarrow bool)\ s{::}?'a{::}type \Rightarrow bool.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2465{::}?'a{::}type.$
$\exists\,x{::}?'a{::}type.\ SETSPEC\ GEN\%PVAR\%2465\ (IN\,x\,s \land f\,x \neq EMPTY)\ x)) \land$
$(\forall\,x{::}?'a{::}type.\ IN\,x\,s \longrightarrow real\_measurable\ (f\,x)) \land (\forall\,(x{::}?'a{::}type)\ y{::}?'a{::}type.$
$IN\,x\,s \land IN\,y\,s \land x \neq y \longrightarrow DISJOINT\ (f\,x)\ (f\,y)) \longrightarrow real\_measure\ (UNIONS$
$(IMAGE\,f\,s)) = sum\ s\ (\lambda x{::}?'a{::}type.\ real\_measure\ (f\,x))$

thm REAL_MEASURE_UNION:

$\forall\,(s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_measurable\ s \land real\_measurable\ t \longrightarrow$
$real\_measure\ (HOL\_Light\_Import.UNION\,s\,t) = real\_measure\,s + (real\_measure$
$t - real\_measure\ (HOL\_Light\_Import.INTER\ s\ t))$

thm REAL_MEASURE_UNION_LE:

$\forall\,(s{::}real \Rightarrow bool)\ t{::}real \Rightarrow bool.\ real\_measurable\ s \land real\_measurable\ t \longrightarrow$
$real\_measure\ (HOL\_Light\_Import.UNION\,s\,t) \leq real\_measure\,s + real\_measure$
t

thm REAL_MEASURE_UNIONS_LE:

$\forall\,f{::}(real \Rightarrow bool) \Rightarrow bool.\ FINITE\,f \land (\forall\,s{::}real \Rightarrow bool.\ IN\,s\,f \longrightarrow real\_measurable$
$s) \longrightarrow real\_measure\ (UNIONS\,f) \leq sum\,f\,real\_measure$

thm REAL_MEASURE_UNIONS_LE_IMAGE:

$\forall\,(f{::}?'a{::}type \Rightarrow bool)\ s{::}?'a{::}type \Rightarrow real \Rightarrow bool.\ FINITE\,f \land (\forall\,a{::}?'a{::}type.$
$IN\ a\ f \longrightarrow real\_measurable\ (s\ a)) \longrightarrow real\_measure\ (UNIONS\ (IMAGE\ s\ f))$
$\leq sum\ f\ (\lambda a{::}?'a{::}type.\ real\_measure\ (s\ a))$

thm REAL_MEASURABLE_INNER_OUTER:

$\forall s{::}real \Rightarrow bool.\ real\_measurable\ s = (\forall e{>}0{::}real.\ \exists (t{::}real \Rightarrow bool)\ u{::}real \Rightarrow bool.\ SUBSET\ t\ s \wedge SUBSET\ s\ u \wedge real\_measurable\ t \wedge real\_measurable\ u \wedge |real\_measure\ t - real\_measure\ u| < e)$

thm HAS_REAL_MEASURE_INNER_OUTER:

$\forall (s{::}real \Rightarrow bool)\ m{::}real.\ has\_real\_measure\ s\ m = ((\forall e{>}0{::}real.\ \exists t{::}real \Rightarrow bool.\ SUBSET\ t\ s \wedge real\_measurable\ t \wedge m - e < real\_measure\ t) \wedge (\forall e{>}0{::}real.\ \exists u{::}real \Rightarrow bool.\ SUBSET\ s\ u \wedge real\_measurable\ u \wedge real\_measure\ u < m + e))$

thm HAS_REAL_MEASURE_INNER_OUTER_LE:

$\forall (s{::}real \Rightarrow bool)\ m{::}real.\ has\_real\_measure\ s\ m = ((\forall e{>}0{::}real.\ \exists t{::}real \Rightarrow bool.\ SUBSET\ t\ s \wedge real\_measurable\ t \wedge m - e \leq real\_measure\ t) \wedge (\forall e{>}0{::}real.\ \exists u{::}real \Rightarrow bool.\ SUBSET\ s\ u \wedge real\_measurable\ u \wedge real\_measure\ u \leq m + e))$

thm HAS_REAL_MEASURE_AFFINITY:

$\forall (s{::}real \Rightarrow bool)\ (m{::}real)\ (c{::}real)\ y{::}real.\ has\_real\_measure\ s\ y \longrightarrow has\_real\_measure\ (IMAGE\ (\lambda x{::}real.\ m * x + c)\ s)\ (|m| * y)$

thm HAS_REAL_MEASURE_SCALING:

$\forall (s{::}real \Rightarrow bool)\ (m{::}real)\ y{::}real.\ has\_real\_measure\ s\ y \longrightarrow has\_real\_measure\ (IMAGE\ (op * m)\ s)\ (|m| * y)$

thm HAS_REAL_MEASURE_TRANSLATION:

$\forall (s{::}real \Rightarrow bool)\ (m{::}real)\ a{::}real.\ has\_real\_measure\ s\ m \longrightarrow has\_real\_measure\ (IMAGE\ (op + a)\ s)\ m$

thm REAL_NEGLIGIBLE_TRANSLATION:

$\forall (s{::}real \Rightarrow bool)\ a{::}real.\ real\_negligible\ s \longrightarrow real\_negligible\ (IMAGE\ (op + a)\ s)$

thm HAS_REAL_MEASURE_TRANSLATION_EQ:

$\forall (s{::}real \Rightarrow bool)\ m{::}real.\ has\_real\_measure\ (IMAGE\ (op + (?a{::}real))\ s)\ m = has\_real\_measure\ s\ m$

thm REAL_NEGLIGIBLE_TRANSLATION_REV:

$\forall (s{::}real \Rightarrow bool)\ a{::}real.\ real\_negligible\ (IMAGE\ (op + a)\ s) \longrightarrow real\_negligible\ s$

thm REAL_NEGLIGIBLE_TRANSLATION_EQ:

$\forall (s{::}real \Rightarrow bool)\ a{::}real.\ real\_negligible\ (IMAGE\ (op + a)\ s) = real\_negligible\ s$

thm REAL_MEASURABLE_TRANSLATION:

$\forall s{::}real \Rightarrow bool.\ real\_measurable\ (IMAGE\ (op + (?a{::}real))\ s) = real\_measurable\ s$

thm REAL_MEASURE_TRANSLATION:

$\forall$ *s::real* $\Rightarrow$ *bool. real_measurable s* $\longrightarrow$ *real_measure* (*IMAGE* (*op* + (*?a::real*)) *s*) = *real_measure s*

thm HAS_REAL_MEASURE_SCALING_EQ:

$\forall$ (*s::real* $\Rightarrow$ *bool*) (*m::real*) *c::real. c* $\neq$ (*0::real*) $\longrightarrow$ *has_real_measure* (*IMAGE* (*op* $*$ *c*) *s*) ($|c|$ $*$ *m*) = *has_real_measure s m*

thm REAL_MEASURABLE_SCALING:

$\forall$ (*s::real* $\Rightarrow$ *bool*) *c::real. real_measurable s* $\longrightarrow$ *real_measurable* (*IMAGE* (*op* $*$ *c*) *s*)

thm REAL_MEASURABLE_SCALING_EQ:

$\forall$ (*s::real* $\Rightarrow$ *bool*) *c::real. c* $\neq$ (*0::real*) $\longrightarrow$ *real_measurable* (*IMAGE* (*op* $*$ *c*) *s*) = *real_measurable s*

thm REAL_MEASURE_SCALING:

$\forall$ *s::real* $\Rightarrow$ *bool. real_measurable s* $\longrightarrow$ *real_measure* (*IMAGE* (*op* $*$ (*?c::real*)) *s*) = $|?c|$ $*$ *real_measure s*

thm HAS_REAL_MEASURE_NESTED_UNIONS:

$\forall$ (*s::nat* $\Rightarrow$ *real* $\Rightarrow$ *bool*) *B::real.* ($\forall$ *n::nat. real_measurable* (*s n*)) $\wedge$ ($\forall$ *n::nat. real_measure* (*s n*) $\leq$ *B*) $\wedge$ ($\forall$ *n::nat. SUBSET* (*s n*) (*s* (*Suc n*))) $\longrightarrow$ *real_measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2468::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%2468* (*IN n HOL_Light_Import.UNIV*) (*s n*)))) $\wedge$ ---> (λ*n::nat. real_measure* (*s n*)) (*real_measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2469::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%2469* (*IN n HOL_Light_Import.UNIV*) (*s n*))))) *sequentially*

thm REAL_MEASURABLE_NESTED_UNIONS:

$\forall$ (*s::nat* $\Rightarrow$ *real* $\Rightarrow$ *bool*) *B::real.* ($\forall$ *n::nat. real_measurable* (*s n*)) $\wedge$ ($\forall$ *n::nat. real_measure* (*s n*) $\leq$ *B*) $\wedge$ ($\forall$ *n::nat. SUBSET* (*s n*) (*s* (*Suc n*))) $\longrightarrow$ *real_measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2470::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%2470* (*IN n HOL_Light_Import.UNIV*) (*s n*))))

thm HAS_REAL_MEASURE_COUNTABLE_REAL_NEGLIGIBLE_UNIONS:

$\forall$ (*s::nat* $\Rightarrow$ *real* $\Rightarrow$ *bool*) *B::real.* ($\forall$ *n::nat. real_measurable* (*s n*)) $\wedge$ ($\forall$ (*m::nat*) *n::nat. m* $\neq$ *n* $\longrightarrow$ *real_negligible* (*HOL_Light_Import.INTER* (*s m*) (*s n*))) $\wedge$ ($\forall$ *n::nat. sum* (*dotdot* (*0::nat*) *n*) (λ*k::nat. real_measure* (*s k*)) $\leq$ *B*) $\longrightarrow$ *real_measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2472::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%2472* (*IN n HOL_Light_Import.UNIV*) (*s n*)))) $\wedge$ *real_sums* (λ*n::nat. real_measure* (*s n*)) (*real_measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2473::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%2473* (*IN n HOL_Light_Import.UNIV*) (*s n*))))) (*from* (*0::nat*))

thm REAL_NEGLIGIBLE_COUNTABLE_UNIONS:

$\forall\, s::nat \Rightarrow real \Rightarrow bool.\ (\forall\, n::nat.\ real\_negligible\ (s\ n)) \longrightarrow real\_negligible\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2474::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2474\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_MEASURABLE_COUNTABLE_UNIONS_STRONG:

$\forall\, (s::nat \Rightarrow real \Rightarrow bool)\ B::real.\ (\forall\, n::nat.\ real\_measurable\ (s\ n)) \wedge (\forall\, n::nat.\ real\_measure\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2476::real \Rightarrow bool.\ \exists\, k::nat.\ SETSPEC\ GEN\%PVAR\%2476\ (k \leq n)\ (s\ k)))) \leq B) \longrightarrow real\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2477::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2477\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm HAS_REAL_MEASURE_COUNTABLE_REAL_NEGLIGIBLE_UNIONS_BOUNDED:

$\forall\, s::nat \Rightarrow real \Rightarrow bool.\ (\forall\, n::nat.\ real\_measurable\ (s\ n)) \wedge (\forall\, (m::nat)\ n::nat.\ m \neq n \longrightarrow real\_negligible\ (HOL\_Light\_Import.INTER\ (s\ m)\ (s\ n))) \wedge real\_bounded\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2480::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2480\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n)))) \longrightarrow real\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2481::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2481\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n)))) \wedge real\_sums\ (\lambda n::nat.\ real\_measure\ (s\ n))\ (real\_measure\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2482::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2482\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n)))))\ (from\ (0::nat))$

thm REAL_MEASURABLE_COUNTABLE_UNIONS:

$\forall\, (s::nat \Rightarrow real \Rightarrow bool)\ B::real.\ (\forall\, n::nat.\ real\_measurable\ (s\ n)) \wedge (\forall\, n::nat.\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ real\_measure\ (s\ k)) \leq B) \longrightarrow real\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2483::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2483\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_MEASURABLE_COUNTABLE_UNIONS_BOUNDED:

$\forall\, s::nat \Rightarrow real \Rightarrow bool.\ (\forall\, n::nat.\ real\_measurable\ (s\ n)) \wedge real\_bounded\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2486::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2486\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n)))) \longrightarrow real\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2487::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2487\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_MEASURABLE_COUNTABLE_INTERS:

$\forall\, s::nat \Rightarrow real \Rightarrow bool.\ (\forall\, n::nat.\ real\_measurable\ (s\ n)) \longrightarrow real\_measurable\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%2490::real \Rightarrow bool.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%2490\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_NEGLIGIBLE_COUNTABLE:

$\forall\, s::real \Rightarrow bool.\ COUNTABLE\ s \longrightarrow real\_negligible\ s$

thm REAL_MEASURABLE_COMPACT:

$\forall\, s::real \Rightarrow bool.\ real\_compact\ s \longrightarrow real\_measurable\ s$

thm REAL_MEASURABLE_OPEN:

$\forall s::real \Rightarrow bool.\ real\_bounded\ s \wedge real\_open\ s \longrightarrow real\_measurable\ s$

thm HAS_REAL_INTEGRAL_NEGLIGIBLE_EQ:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow (0::real) \leq f\ x) \longrightarrow has\_real\_integral\ f\ (0::real)\ s = real\_negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2495::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2495\ (IN\ x\ s \wedge f\ x \neq (0::real))\ x))$

thm DEF_dropout:

$dropout = (\lambda(\_1900031::nat)\ \_1900032::(real,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ if\ i < \_1900031\ then\ \$\ \_1900032\ i\ else\ \$\ \_1900032\ (i + (1::nat))))$

thm dropout:

$\forall (k::nat)\ x::(real,\ ?'b::type)\ cart.\ dropout\ k\ x = lambda\ (\lambda i::nat.\ if\ i < k\ then\ \$\ x\ i\ else\ \$\ x\ (i + (1::nat)))$

thm DEF_pushin:

$pushin = (\lambda(\_1900043::nat)\ (\_1900044::?'c::type)\ \_1900045::(?'c::type,\ ?'b::type)\ cart.\ lambda\ (\lambda i::nat.\ if\ i < \_1900043\ then\ \$\ \_1900045\ i\ else\ if\ i = \_1900043\ then\ \_1900044\ else\ \$\ \_1900045\ (i - (1::nat))))$

thm pushin:

$\forall (k::nat)\ (t::?'c::type)\ x::(?'c::type,\ ?'b::type)\ cart.\ pushin\ k\ t\ x = lambda\ (\lambda i::nat.\ if\ i < k\ then\ \$\ x\ i\ else\ if\ i = k\ then\ t\ else\ \$\ x\ (i - (1::nat)))$

thm DROPOUT_PUSHIN:

$\forall (k::nat)\ (t::real)\ x::(real,\ ?'b::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \longrightarrow dropout\ k\ (pushin\ k\ t\ x) = x$

thm PUSHIN_DROPOUT:

$\forall (k::nat)\ x::(real,\ ?'b::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow pushin\ k\ (\$\ x\ k)\ (dropout\ k\ x) = x$

thm DROPOUT_GALOIS:

$\forall (k::nat)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (y = dropout\ k\ x) = (\exists t::real.\ x = pushin\ k\ t\ y)$

thm IN_IMAGE_DROPOUT:

$\forall (x::(real,\ ?'b::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?k::nat) \wedge ?k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ x\ (IMAGE\ (dropout\ ?k)\ s) = (\exists t::real.\ IN\ (pushin\ ?k\ t\ x)\ s)$

thm CLOSED_INTERVAL_DROPOUT:

$\forall$ (*k*::*nat*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ \$ *a k* $\leq$ \$ *b k* $\longrightarrow$ *closed_interval* [(*dropout k a*, *dropout k b*)] = *IMAGE* (*dropout k*) (*closed_interval* [(*a*, *b*)])

thm IMAGE_DROPOUT_CLOSED_INTERVAL:

$\forall$ (*k*::*nat*) (*a*::(*real*, *?'b*::*type*) *cart*) *b*::(*real*, *?'b*::*type*) *cart*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *IMAGE* (*dropout k*) (*closed_interval* [(*a*, *b*)]) = (*if* \$ *a k* $\leq$ \$ *b k then closed_interval* [(*dropout k a*, *dropout k b*)] *else EMPTY*)

thm LINEAR_DROPOUT:

$\forall$ *k*::*nat*. *dimindex HOL_Light_Import.UNIV* < *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *linear* (*dropout k*)

thm DROPOUT_EQ:

$\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) (*y*::(*real*, *?'b*::*type*) *cart*) *k*::*nat*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ \$ *x k* = \$ *y k* $\wedge$ *dropout k x* = *dropout k y* $\longrightarrow$ *x* = *y*

thm DROPOUT_0:

dropout (*?k*::*nat*) (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*)

thm DOT_DROPOUT:

$\forall$ (*k*::*nat*) (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *dot* (*dropout k x*) (*dropout k y*) = *dot x y* − \$ *x k* $*$ \$ *y k*

thm DOT_PUSHIN:

$\forall$ (*k*::*nat*) (*a*::*real*) (*b*::*real*) (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *dot* (*pushin k a x*) (*pushin k b y*) = *dot x y* + *a* $*$ *b*

thm DROPOUT_ADD:

$\forall$ (*k*::*nat*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *dropout k* (*vector_add x y*) = *vector_add* (*dropout k x*) (*dropout k y*)

thm DROPOUT_SUB:

$\forall$ (*k*::*nat*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *dropout k* (*vector_sub x y*) = *vector_sub* (*dropout k x*) (*dropout k y*)

thm DROPOUT_MUL:

∀ (k::nat) (c::real) x::(real, ?'b::type) cart. dropout k (% c x) = % c (dropout k x)

thm DEF_slice:

slice = (λ(_1900507::nat) (_1900508::real) _1900509::(real, ?'b::type) cart ⇒ bool. IMAGE (dropout _1900507) (HOL_Light_Import.INTER _1900509 (GSPEC (λGEN%PVAR%2496::(real, ?'b::type) cart. ∃ x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%2496 ($ x _1900507 = _1900508) x))))

thm slice:

∀ (s::(real, ?'b::type) cart ⇒ bool) (k::nat) t::real. slice k t s = IMAGE (dropout k) (HOL_Light_Import.INTER s (GSPEC (λGEN%PVAR%2496::(real, ?'b::type) cart. ∃ x::(real, ?'b::type) cart. SETSPEC GEN%PVAR%2496 ($ x k = t) x)))

thm IN_SLICE:

∀ (s::(real, ?'b::type) cart ⇒ bool) y::(real, ?'a::type) cart. dimindex HOL_Light_Import.UNIV + (1::nat) = dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?k::nat) ∧ ?k ≤ dimindex HOL_Light_Import.UNIV ⟶ IN y (slice ?k (?t::real) s) = IN (pushin ?k ?t y) s

thm INTERVAL_INTER_HYPERPLANE:

∀ (k::nat) (t::real) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ HOL_Light_Import.INTER (closed_interval [(a, b)]) (GSPEC (λGEN%PVAR%2497::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2497 ($ x k = t) x)) = (if $ a k ≤ t ∧ t ≤ $ b k then closed_interval [(lambda (λi::nat. if i = k then t else $ a i), lambda (λi::nat. if i = k then t else $ b i))] else EMPTY)

thm SLICE_INTERVAL:

∀ (k::nat) (a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) t::real. dimindex HOL_Light_Import.UNIV + (1::nat) = dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ slice k t (closed_interval [(a, b)]) = (if $ a k ≤ t ∧ t ≤ $ b k then closed_interval [(dropout k a, dropout k b)] else EMPTY)

thm SLICE_DIFF:

∀ (k::nat) (a::real) (s::(real, ?'b::type) cart ⇒ bool) t::(real, ?'b::type) cart ⇒ bool. dimindex HOL_Light_Import.UNIV + (1::nat) = dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ slice k a (DIFF s t) = DIFF (slice k a s) (slice k a t)

thm SLICE_UNIV:

∀ (k::nat) a::real. dimindex HOL_Light_Import.UNIV + (1::nat) = dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ k ∧ k ≤ dimindex HOL_Light_Import.UNIV ⟶ slice k a HOL_Light_Import.UNIV = HOL_Light_Import.UNIV

thm SLICE_EMPTY:

$\forall (k::nat)\ a::real.\ slice\ k\ a\ EMPTY = EMPTY$

thm SLICE_SUBSET:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'b::type)\ cart \Rightarrow bool)\ (k::nat)\ a::real.\ SUBSET\ s\ t \longrightarrow SUBSET\ (slice\ k\ a\ s)\ (slice\ k\ a\ t)$

thm SLICE_UNIONS:

$\forall (s::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (k::nat)\ a::real.\ slice\ k\ a\ (UNIONS\ s) = UNIONS\ (IMAGE\ (slice\ k\ a)\ s)$

thm SLICE_UNION:

$\forall (k::nat)\ (a::real)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow slice\ k\ a\ (HOL\_Light\_Import.UNION\ s\ t) = HOL\_Light\_Import.UNION\ (slice\ k\ a\ s)\ (slice\ k\ a\ t)$

thm SLICE_INTER:

$\forall (k::nat)\ (a::real)\ (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow slice\ k\ a\ (HOL\_Light\_Import.INTER\ s\ t) = HOL\_Light\_Import.INTER\ (slice\ k\ a\ s)\ (slice\ k\ a\ t)$

thm CONVEX_SLICE:

$\forall (k::nat)\ (t::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV < dimindex\ HOL\_Light\_Import.UNIV \wedge convex\ s \longrightarrow convex\ (slice\ k\ t\ s)$

thm COMPACT_SLICE:

$\forall (k::nat)\ (t::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV < dimindex\ HOL\_Light\_Import.UNIV \wedge compact\ s \longrightarrow compact\ (slice\ k\ t\ s)$

thm CLOSED_SLICE:

$\forall (k::nat)\ (t::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge HOL\_Light\_Import.closed\ s \longrightarrow HOL\_Light\_Import.closed\ (slice\ k\ t\ s)$

thm OPEN_SLICE:

$\forall (k::nat)\ (t::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge HOL\_Light\_Import.open\ s \longrightarrow HOL\_Light\_Import.open\ (slice\ k\ t\ s)$

thm BOUNDED_SLICE:

$\forall (k::nat)\ (t::real)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge bounded\ s \longrightarrow bounded\ (slice\ k\ t\ s)$

thm SLICE_CBALL:

$\forall (k::nat)\ (t::real)\ (x::(real,\ ?'b::type)\ cart)\ r::real.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow slice\ k\ t\ (cball\ (x,\ r)) = (if\ |t - \$\ x\ k| \leq r\ then\ cball\ (dropout\ k\ x,\ sqrt\ (r^2 - (t - \$\ x\ k)^2))\ else\ EMPTY)$

thm SLICE_BALL:

$\forall (k::nat)\ (t::real)\ (x::(real,\ ?'b::type)\ cart)\ r::real.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow slice\ k\ t\ (ball\ (x,\ r)) = (if\ |t - \$\ x\ k| < r\ then\ ball\ (dropout\ k\ x,\ sqrt\ (r^2 - (t - \$\ x\ k)^2))\ else\ EMPTY)$

thm FUBINI_CLOSED_INTERVAL:

$\forall (k::nat)\ (a::(real,\ ?'b::type)\ cart)\ b::(real,\ ?'b::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV + (1::nat) = dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq k \wedge k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge \$\ a\ k \leq \$\ b\ k \longrightarrow has\_real\_integral\ (\lambda t::real.\ HOL\_Light\_Import.measure\ (slice\ k\ t\ (closed\_interval\ [(a,\ b)])))\ (HOL\_Light\_Import.measure\ (closed\_interval\ [(a,\ b)]))\ HOL\_Light\_Import.UNIV$

thm MEASURABLE_OUTER_INTERVALS_BOUNDED_EXPLICIT_SPECIAL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ e::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?k::nat) \wedge ?k \leq dimindex\ HOL\_Light\_Import.UNIV \wedge measurable\ s \wedge SUBSET\ s\ (closed\_interval\ [(a,\ b)]) \wedge (0::real) < e \longrightarrow (\exists f::nat \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool.\ (\forall i::nat.\ SUBSET\ (f\ i)\ (closed\_interval\ [(a,\ b)]) \wedge (\exists (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ \$\ c\ ?k \leq \$\ d\ ?k \wedge f\ i = closed\_interval\ [(c,\ d)])) \wedge (\forall (i::nat)\ j::nat.\ i \neq j \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ (f\ i)\ (f\ j))) \wedge SUBSET\ s\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2501::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2501\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (f\ n)))) \wedge measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2502::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2502\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (f\ n)))) \wedge HOL\_Light\_Import.measure\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2503::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2503\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (f\ n)))) \leq HOL\_Light\_Import.measure\ s + e)$

thm REAL_MONOTONE_CONVERGENCE_INCREASING_AE:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall k::nat.\ real\_integrable\_on\ (f\ k)\ s) \wedge (\forall (k::nat)\ x::real.\ IN\ x\ s \longrightarrow f\ k\ x \leq f\ (Suc\ k)\ x) \wedge (\exists t::real \Rightarrow bool.\ real\_negligible\ t \wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ t) \longrightarrow --->\ (\lambda k::nat.\ f\ k\ x)\ (g\ x)\ sequentially)) \wedge real\_bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2504::real.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%2504\ (IN\ k\ HOL\_Light\_Import.UNIV)\ (real\_integral\ s\ (f\ k)))) \longrightarrow real\_integrable\_on\ g\ s \wedge --->\ (\lambda k::nat.\ real\_integral\ s\ (f\ k))\ (real\_integral\ s\ g)\ sequentially$

thm FUBINI_SIMPLE_LEMMA:

∀ (*k*::*nat*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *e*::*real*. (*0*::*real*) < *e* ∧ *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *bounded s* ∧ *measurable s* ∧ (∀ *t*::*real*. *measurable* (*slice k t s*)) ∧ *real_integrable_on* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) *HOL_Light_Import.UNIV* ⟶ *real_integral HOL_Light_Import.UNIV* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) ≤ *HOL_Light_Import.measure s* + *e*

thm FUBINI_SIMPLE:

∀ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *bounded s* ∧ *measurable s* ∧ (∀ *t*::*real*. *measurable* (*slice k t s*)) ∧ *real_integrable_on* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) *HOL_Light_Import.UNIV* ⟶ *HOL_Light_Import.measure s* = *real_integral HOL_Light_Import.UNIV* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*))

thm FUBINI_SIMPLE_ALT:

∀ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *bounded s* ∧ *measurable s* ∧ (∀ *t*::*real*. *measurable* (*slice k t s*)) ∧ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* ⟶ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_COMPACT_STRONG:

∀ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *compact s* ∧ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* ⟶ *measurable s* ∧ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_COMPACT:

∀ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *compact s* ∧ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* ⟶ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_CONVEX_STRONG:

∀ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *k* ∧ *k* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *bounded s* ∧ *convex s* ∧ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* ⟶ *measurable s* ∧ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_CONVEX:

$\forall$ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\wedge$ *convex s* $\wedge$ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* $\longrightarrow$ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_OPEN_STRONG:

$\forall$ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\wedge$ *HOL_Light_Import.open s* $\wedge$ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* $\longrightarrow$ *measurable s* $\wedge$ *HOL_Light_Import.measure s* = *?B*

thm FUBINI_SIMPLE_OPEN:

$\forall$ (*k*::*nat*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) = *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *bounded s* $\wedge$ *HOL_Light_Import.open s* $\wedge$ *has_real_integral* (λ*t*::*real*. *HOL_Light_Import.measure* (*slice k t s*)) (*?B*::*real*) *HOL_Light_Import.UNIV* $\longrightarrow$ *HOL_Light_Import.measure s* = *?B*

thm REAL_OPEN_SET_RATIONAL:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_open s* $\wedge$ *s* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *x*::*real*. *rational x* $\wedge$ *IN x s*)

thm REAL_OPEN_RATIONAL:

$\forall$ *P*::*real* $\Rightarrow$ *bool*. *real_open* (*GSPEC* (λ*GEN%PVAR%2509*::*real*. $\exists$ *x*::*real*. *SETSPEC GEN%PVAR%2509* (*P x*) *x*)) $\wedge$ ($\exists$ *x*::*real*. *P x*) $\longrightarrow$ ($\exists$ *x*::*real*. *rational x* $\wedge$ *P x*)

thm REAL_OPEN_SET_EXISTS_RATIONAL:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_open s* $\longrightarrow$ ($\exists$ *x*::*real*. *rational x* $\wedge$ *IN x s*) = ($\exists$ *x*::*real*. *IN x s*)

thm REAL_OPEN_EXISTS_RATIONAL:

$\forall$ *P*::*real* $\Rightarrow$ *bool*. *real_open* (*GSPEC* (λ*GEN%PVAR%2510*::*real*. $\exists$ *x*::*real*. *SETSPEC GEN%PVAR%2510* (*P x*) *x*)) $\longrightarrow$ ($\exists$ *x*::*real*. *rational x* $\wedge$ *P x*) = ($\exists$ *x*::*real*. *P x*)

thm CONTINUOUS_ON_CONST_DYADIC_RATIONALS:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous_on f HOL_Light_Import.UNIV* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. ($\forall$ *i*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *integer* ($ *x i*)) $\longrightarrow$ *f x* = *a*) $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *f x* = *a* $\longrightarrow$ *f* (% (*inverse_class.inverse* (*real_of_nat* (*2*::*nat*))) *x*) = *a*) $\longrightarrow$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *f x* = *a*)

thm REAL_CONTINUOUS_ON_CONST_DYADIC_RATIONALS:

∀(f::real ⇒ real) a::real. real_continuous_on f HOL_Light_Import.UNIV ∧ (∀x::real. integer x ⟶ f x = a) ∧ (∀x::real. f x = a ⟶ f (x / real_of_nat (2::nat)) = a) ⟶ (∀x::real. f x = a)

thm CONTINUOUS_ADDITIVE_IMP_LINEAR:

∀f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. continuous_on f HOL_Light_Import.UNIV ∧ (∀(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f (vector_add x y) = vector_add (f x) (f y)) ⟶ linear f

thm OSTROWSKI_THEOREM:

∀(f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (B::real) s::(real, ?'b::type) cart ⇒ bool. (∀(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f (vector_add x y) = vector_add (f x) (f y)) ∧ (∀x::(real, ?'b::type) cart. IN x s ⟶ vector_norm (f x) ≤ B) ∧ measurable s ∧ (0::real) < HOL_Light_Import.measure s ⟶ linear f

thm MEASURABLE_ADDITIVE_IMP_LINEAR:

∀f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart. measurable_on f HOL_Light_Import.UNIV ∧ (∀(x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. f (vector_add x y) = vector_add (f x) (f y)) ⟶ linear f

thm REAL_CONTINUOUS_ADDITIVE_IMP_LINEAR:

∀f::real ⇒ real. real_continuous_on f HOL_Light_Import.UNIV ∧ (∀(x::real) y::real. f (x + y) = f x + f y) ⟶ (∀(a::real) x::real. f (a * x) = a * f x)

thm REAL_CONTINUOUS_FLOOR:

∀x::real. ¬ integer x ⟶ real_continuous HOL_Light_Import.floor (atreal x)

thm REAL_CONTINUOUS_FRAC:

∀x::real. ¬ integer x ⟶ real_continuous frac (atreal x)

thm REAL_CONTINUOUS_ON_COMPOSE_FRAC:

∀f::real ⇒ real. real_continuous_on f (closed_real_interval [(0::real, 1::real)]) ∧ f (1::real) = f (0::real) ⟶ real_continuous_on (f ∘ frac) HOL_Light_Import.UNIV

thm REAL_TIETZE_PERIODIC_INTERVAL:

∀(f::real ⇒ real) (a::real) b::real. real_continuous_on f (closed_real_interval [(a, b)]) ∧ f a = f b ⟶ (∃g::real ⇒ real. real_continuous_on g HOL_Light_Import.UNIV ∧ (∀x::real. IN x (closed_real_interval [(a, b)]) ⟶ g x = f x) ∧ (∀x::real. g (x + (b − a)) = g x))

thm REAL_CONTINUOUS_ADDITIVE_EXTEND:

∀f::real ⇒ real. real_continuous_on f (closed_real_interval [(0::real, 1::real)]) ∧ (∀(x::real) y::real. (0::real) ≤ x ∧ (0::real) ≤ y ∧ x + y ≤ (1::real) ⟶ f (x + y) = f x + f y) ⟶ (∃g::real ⇒ real. real_continuous_on g HOL_Light_Import.UNIV

$\wedge$ ($\forall$ (*x*::*real*) *y*::*real*. *g* (*x* + *y*) = *g x* + *g y*) $\wedge$ ($\forall$ *x*::*real*. *IN x* (*closed_real_interval* [(*0*::*real*, *1*::*real*)]) $\longrightarrow$ *g x* = *f x*))

thm REAL_CONTINUOUS_ADDITIVE_IMP_LINEAR_INTERVAL:

$\forall$ (*f*::*real* $\Rightarrow$ *real*) *b*::*real*. −−−> *f* (*0*::*real*) (*within* (*atreal* (*0*::*real*)) (*GSPEC* (λ*GEN%PVAR%2516*::*real*. $\exists$ *x*::*real*. *SETSPEC GEN%PVAR%2516* ((*0*::*real*) $\leq$ *x*) *x*))) $\wedge$ ($\forall$ (*x*::*real*) *y*::*real*. (*0*::*real*) $\leq$ *x* $\wedge$ (*0*::*real*) $\leq$ *y* $\wedge$ *x* + *y* $\leq$ *b* $\longrightarrow$ *f* (*x* + *y*) = *f x* + *f y*) $\longrightarrow$ ($\forall$ (*a*::*real*) *x*::*real*. (*0*::*real*) $\leq$ *x* $\wedge$ *x* $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *a* ∗ *x* $\wedge$ *a* ∗ *x* $\leq$ *b* $\longrightarrow$ *f* (*a* ∗ *x*) = *a* ∗ *f x*)

thm DISCRETE_IMP_COUNTABLE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ ($\exists$ *e*>*0*::*real*. $\forall$ *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* $\wedge$ *y* $\neq$ *x* $\longrightarrow$ *e* $\leq$ *vector_norm* (*vector_sub y x*))) $\longrightarrow$ *COUNTABLE s*

thm UNCOUNTABLE_CONTAINS_LIMIT_POINT:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\neg$ *COUNTABLE s* $\longrightarrow$ ($\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *limit_point_of x s*)

thm STEINHAUS_TRIVIAL:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *e*::*real*. $\neg$ *negligible s* $\wedge$ (*0*::*real*) < *e* $\longrightarrow$ ($\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\wedge$ *IN y s* $\wedge$ *x* $\neq$ *y* $\wedge$ *vector_norm* (*vector_sub x y*) < *e*)

thm REAL_STEINHAUS:

$\forall$ *s*::*real* $\Rightarrow$ *bool*. *real_measurable s* $\wedge$ (*0*::*real*) < *real_measure s* $\longrightarrow$ ($\exists$ *d*>*0*::*real*. *SUBSET* (*open_real_interval* (− *d*, *d*)) (*GSPEC* (λ*GEN%PVAR%2522*::*real*. $\exists$ (*x*::*real*) *y*::*real*. *SETSPEC GEN%PVAR%2522* (*IN x s* $\wedge$ *IN y s*) (*x* − *y*))))

thm DEF_bernstein:

bernstein = (λ(*_1910044*::*nat*) (*_1910045*::*nat*) *_1910046*::*real*. *real_of_nat* (*binom* (*_1910044*, *_1910045*)) ∗ ($\_1910046^{\_1910045}$ ∗ ((*1*::*real*) − $\_1910046)^{\_1910044\ -\ \_1910045}$))

thm bernstein:

$\forall$ (*x*::*real*) (*n*::*nat*) *k*::*nat*. *bernstein n k x* = *real_of_nat* (*binom* (*n*, *k*)) ∗ (x^k ∗ ((*1*::*real*) − $x)^{n\ -\ k}$)

thm BERNSTEIN_POS:

$\forall$ (*n*::*nat*) (*k*::*nat*) *x*::*real*. (*0*::*real*) $\leq$ *x* $\wedge$ *x* $\leq$ (*1*::*real*) $\longrightarrow$ (*0*::*real*) $\leq$ *bernstein n k x*

thm SUM_BERNSTEIN:

$\forall$ *n*::*nat*. *sum* (*dotdot* (*0*::*nat*) *n*) (λ*k*::*nat*. *bernstein n k* (*?x*::*real*)) = (*1*::*real*)

thm BERNSTEIN_LEMMA:

$\forall$ (*n*::*nat*) *x*::*real*. *sum* (*dotdot* (*0*::*nat*) *n*) (λ*k*::*nat*. (*real_of_nat k* − *real_of_nat n* ∗ $x)^2$ ∗ *bernstein n k x*) = *real_of_nat n* ∗ (*x* ∗ ((*1*::*real*) − *x*))

thm BERNSTEIN_WEIERSTRASS:

$\forall (f::real \Rightarrow real)\ e::real.\ real\_continuous\_on\ f\ (closed\_real\_interval\ [(0::real, 1::real)])\ \wedge\ (0::real) < e \longrightarrow (\exists N::nat.\ \forall (n::nat)\ x::real.\ N \leq n \wedge IN\ x\ (closed\_real\_interval\ [(0::real, 1::real)]) \longrightarrow |f\ x - sum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ f\ (real\_of\_nat\ k\ /\ real\_of\_nat\ n) * bernstein\ n\ k\ x)| < e)$

thm STONE_WEIERSTRASS_ALT:

$\forall (P::((real, ?'a::type)\ cart \Rightarrow real) \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge (\forall c::real.\ P\ (\lambda x::(real, ?'a::type)\ cart.\ c)) \wedge (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::(real, ?'a::type)\ cart.\ f\ x + g\ x)) \wedge (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::(real, ?'a::type)\ cart.\ f\ x * g\ x)) \wedge (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow (\exists f::(real, ?'a::type)\ cart \Rightarrow real.\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (within\ (at\ x)\ s)) \wedge P\ f \wedge f\ x \neq f\ y)) \longrightarrow (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ e::real.\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (within\ (at\ x)\ s)) \wedge (0::real) < e \longrightarrow (\exists g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ g \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow |f\ x - g\ x| < e)))$

thm STONE_WEIERSTRASS:

$\forall (P::((real, ?'a::type)\ cart \Rightarrow real) \Rightarrow bool)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ compact\ s \wedge (\forall f::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \longrightarrow (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (within\ (at\ x)\ s))) \wedge (\forall c::real.\ P\ (\lambda x::(real, ?'a::type)\ cart.\ c)) \wedge (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::(real, ?'a::type)\ cart.\ f\ x + g\ x)) \wedge (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::(real, ?'a::type)\ cart.\ f\ x * g\ x)) \wedge (\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow (\exists f::(real, ?'a::type)\ cart \Rightarrow real.\ P\ f \wedge f\ x \neq f\ y)) \longrightarrow (\forall (f::(real, ?'a::type)\ cart \Rightarrow real)\ e::real.\ (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow real\_continuous\ f\ (within\ (at\ x)\ s)) \wedge (0::real) < e \longrightarrow (\exists g::(real, ?'a::type)\ cart \Rightarrow real.\ P\ g \wedge (\forall x::(real, ?'a::type)\ cart.\ IN\ x\ s \longrightarrow |f\ x - g\ x| < e)))$

thm REAL_STONE_WEIERSTRASS_ALT:

$\forall (P::(real \Rightarrow real) \Rightarrow bool)\ s::real \Rightarrow bool.\ real\_compact\ s \wedge (\forall c::real.\ P\ (\lambda x::real.\ c)) \wedge (\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::real.\ f\ x + g\ x)) \wedge (\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ P\ f \wedge P\ g \longrightarrow P\ (\lambda x::real.\ f\ x * g\ x)) \wedge (\forall (x::real)\ y::real.\ IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow (\exists f::real \Rightarrow real.\ real\_continuous\_on\ f\ s \wedge P\ f \wedge f\ x \neq f\ y)) \longrightarrow (\forall (f::real \Rightarrow real)\ e::real.\ real\_continuous\_on\ f\ s \wedge (0::real) < e \longrightarrow (\exists g::real \Rightarrow real.\ P\ g \wedge (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x - g\ x| < e)))$

thm REAL_STONE_WEIERSTRASS:

$\forall (P::(real \Rightarrow real) \Rightarrow bool)\ s::real \Rightarrow bool.\ real\_compact\ s \wedge (\forall f::real \Rightarrow real.\ P\ f \longrightarrow real\_continuous\_on\ f\ s) \wedge (\forall c::real.\ P\ (\lambda x::real.\ c)) \wedge (\forall (f::real \Rightarrow$

real) *g*::*real* ⇒ *real*. *P f* ∧ *P g* ⟶ *P* (λ*x*::*real*. *f x* + *g x*)) ∧ (∀ (*f*::*real* ⇒ *real*) *g*::*real* ⇒ *real*. *P f* ∧ *P g* ⟶ *P* (λ*x*::*real*. *f x* ∗ *g x*)) ∧ (∀ (*x*::*real*) *y*::*real*. *IN x s* ∧ *IN y s* ∧ *x* ≠ *y* ⟶ (∃ *f*::*real* ⇒ *real*. *P f* ∧ *f x* ≠ *f y*)) ⟶ (∀ (*f*::*real* ⇒ *real*) *e*::*real*. *real_continuous_on f s* ∧ (*0*::*real*) < *e* ⟶ (∃ *g*::*real* ⇒ *real*. *P g* ∧ (∀ *x*::*real*. *IN x s* ⟶ |*f x* − *g x*| < *e*)))

thm COMPLEX_STONE_WEIERSTRASS_ALT:

∀ (*P*::((*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ (∀ *c*::(*real*, *2*) *cart*. *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *c*)) ∧ (∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *cnj* (*f x*))) ∧ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *P g* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vector_add* (*f x*) (*g x*))) ∧ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *P g* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_mul* (*f x*) (*g x*))) ∧ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *IN y s* ∧ *x* ≠ *y* ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *continuous_on f s* ∧ *f x* ≠ *f y*)) ⟶ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *e*::*real*. *continuous_on f s* ∧ (*0*::*real*) < *e* ⟶ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P g* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *vector_norm* (*vector_sub* (*f x*) (*g x*)) < *e*)))

thm COMPLEX_STONE_WEIERSTRASS:

∀ (*P*::((*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) ⇒ *bool*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *compact s* ∧ (∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ⟶ *continuous_on f s*) ∧ (∀ *c*::(*real*, *2*) *cart*. *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *c*)) ∧ (∀ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *cnj* (*f x*))) ∧ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *P g* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *vector_add* (*f x*) (*g x*))) ∧ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *P g* ⟶ *P* (λ*x*::(*real*, *?'a*::*type*) *cart*. *complex_mul* (*f x*) (*g x*))) ∧ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN x s* ∧ *IN y s* ∧ *x* ≠ *y* ⟶ (∃ *f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P f* ∧ *f x* ≠ *f y*)) ⟶ (∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*) *e*::*real*. *continuous_on f s* ∧ (*0*::*real*) < *e* ⟶ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *2*) *cart*. *P g* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *vector_norm* (*vector_sub* (*f x*) (*g x*)) < *e*)))

thm DEF_real_polynomial_function:

real_polynomial_function = (λ*a*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. ∀ *real_polynomial_function'*::((*real*, *?'a*::*type*) *cart* ⇒ *real*) ⇒ *bool*. (∀ *a*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. (∃ *i*::*nat*. *a* = (λ*x*::(*real*, *?'a*::*type*) *cart*. $ *x i*) ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) ∨ (∃ *c*::*real*. *a* = (λ*x*::(*real*, *?'a*::*type*) *cart*. *c*)) ∨ (∃ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *a* = (λ*x*::(*real*, *?'a*::*type*) *cart*. *f x* + *g x*) ∧ *real_polynomial_function' f* ∧ *real_polynomial_function' g*) ∨ (∃ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *g*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *a* = (λ*x*::(*real*, *?'a*::*type*) *cart*. *f x* ∗ *g x*) ∧ *real_polynomial_function' f* ∧ *real_polynomial_function' g*) ⟶ *real_polynomial_function' a*) ⟶ *real_polynomial_function' a*)

thm real_polynomial_function_RULES:

$(\forall\, i{::}nat.\ (1{::}nat) \le i \wedge i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ \$\ x\ i)) \wedge (\forall\, c{::}real.\ real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ c)) \wedge (\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g \longrightarrow real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x + g\ x)) \wedge (\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g \longrightarrow real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x * g\ x))$

thm real_polynomial_function_CASES:

$\forall\, a{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ a = ((\exists\, i{::}nat.\ a = (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ \$\ x\ i) \wedge (1{::}nat) \le i \wedge i \le dimindex\ HOL\_Light\_Import.UNIV) \vee (\exists\, c{::}real.\ a = (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ c)) \vee (\exists\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ a = (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x + g\ x) \wedge real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g) \vee (\exists\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ a = (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x * g\ x) \wedge real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g))$

thm real_polynomial_function_INDUCT:

$\forall\, real\_polynomial\_function'{::}((real,\ ?'a{::}type)\ cart \Rightarrow real) \Rightarrow bool.\ (\forall\, i{::}nat.\ (1{::}nat) \le i \wedge i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow real\_polynomial\_function'\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ \$\ x\ i)) \wedge (\forall\, c{::}real.\ real\_polynomial\_function'\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ c)) \wedge (\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function'\ f \wedge real\_polynomial\_function'\ g \longrightarrow real\_polynomial\_function'\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x + g\ x)) \wedge (\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function'\ f \wedge real\_polynomial\_function'\ g \longrightarrow real\_polynomial\_function'\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x * g\ x)) \longrightarrow (\forall\, a{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ a \longrightarrow real\_polynomial\_function'\ a)$

thm REAL_CONTINUOUS_REAL_POLYMONIAL_FUNCTION:

$\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ x{::}(real,\ ?'a{::}type)\ cart.\ real\_polynomial\_function\ f \longrightarrow real\_continuous\ f\ (at\ x)$

thm real_polynomial_function_RULES_conjunct3:

$\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g \longrightarrow real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x * g\ x)$

thm real_polynomial_function_RULES_conjunct2:

$\forall\, (f{::}(real,\ ?'a{::}type)\ cart \Rightarrow real)\ g{::}(real,\ ?'a{::}type)\ cart \Rightarrow real.\ real\_polynomial\_function\ f \wedge real\_polynomial\_function\ g \longrightarrow real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ f\ x + g\ x)$

thm real_polynomial_function_RULES_conjunct1:

$\forall\, c{::}real.\ real\_polynomial\_function\ (\lambda x{::}(real,\ ?'a{::}type)\ cart.\ c)$

thm real_polynomial_function_RULES_conjunct0:

$\forall$ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *real_polynomial_function* (λ*x*::(*real*, *?'a*::*type*) *cart*. $ *x i*)

thm STONE_WEIERSTRASS_REAL_POLYNOMIAL_FUNCTION:

$\forall$ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *e*::*real*. *compact s* ∧ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *real_continuous f* (*within* (*at x*) *s*)) ∧ (*0*::*real*) < *e* ⟶ (∃ *g*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *real_polynomial_function g* ∧ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ |*f x* − *g x*| < *e*))

thm DEF_vector_polynomial_function:

vector_polynomial_function = (λ*_1922923*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. $\forall$ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *real_polynomial_function* (λ*x*::(*real*, *?'b*::*type*) *cart*. $ (*_1922923 x*) *i*))

thm vector_polynomial_function:

$\forall$ *f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function f* = ($\forall$ *i*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *real_polynomial_function* (λ*x*::(*real*, *?'b*::*type*) *cart*. $ (*f x*) *i*))

thm CONTINUOUS_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'b*::*type*) *cart*. *vector_polynomial_function f* ⟶ *continuous f* (*at x*)

thm CONTINUOUS_ON_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *vector_polynomial_function f* ⟶ *continuous_on f s*

thm HAS_VECTOR_DERIVATIVE_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function p* ⟶ (∃ *p'*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function p'* ∧ ($\forall$ *x*::(*real*, *unit*) *cart*. *has_vector_derivative p* (*p' x*) (*at x*)))

thm STONE_WEIERSTRASS_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *e*::*real*. *compact s* ∧ *continuous_on f s* ∧ (*0*::*real*) < *e* ⟶ (∃ *g*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function g* ∧ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *IN x s* ⟶ *vector_norm* (*vector_sub* (*f x*) (*g x*)) < *e*))

thm PATH_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ *g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function g* ⟶ *path g*

thm PATH_APPROX_VECTOR_POLYNOMIAL_FUNCTION:

$\forall$ (*g*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *e*::*real*. *path g* ∧ (*0*::*real*) < *e* ⟶ (∃ *p*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *vector_polynomial_function p* ∧

pathstart p = pathstart g ∧ pathfinish p = pathfinish g ∧ (∀ t::(real, unit) cart. IN t (closed_interval [(vec (0::nat), vec (1::nat))]) ⟶ vector_norm (vector_sub (p t) (g t)) < e))

thm CONNECTED_OPEN_VECTOR_POLYNOMIAL_CONNECTED:

∀ s::(real, ?'a::type) cart ⇒ bool. HOL_Light_Import.open s ∧ connected s ⟶ (∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y s ⟶ (∃ g::(real, unit) cart ⇒ (real, ?'a::type) cart. vector_polynomial_function g ∧ SUBSET (path_image g) s ∧ pathstart g = x ∧ pathfinish g = y))

thm LIPSCHITZ_REAL_POLYNOMIAL_FUNCTION:

∀ (f::(real, ?'a::type) cart ⇒ real) s::(real, ?'a::type) cart ⇒ bool. real_polynomial_function f ∧ bounded s ⟶ (∃ B>0::real. ∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x s ∧ IN y s ⟶ |f x − f y| ≤ B ∗ vector_norm (vector_sub x y))

thm LIPSCHITZ_VECTOR_POLYNOMIAL_FUNCTION:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. vector_polynomial_function f ∧ bounded s ⟶ (∃ B>0::real. ∀ (x::(real, ?'b::type) cart) y::(real, ?'b::type) cart. IN x s ∧ IN y s ⟶ vector_norm (vector_sub (f x) (f y)) ≤ B ∗ vector_norm (vector_sub x y))

thm MEASURABLE_ON_COMPLEX_MUL:

∀ (f::(real, ?'a::type) cart ⇒ (real, 2) cart) (g::(real, ?'a::type) cart ⇒ (real, 2) cart) s::(real, ?'a::type) cart ⇒ bool. measurable_on f s ∧ measurable_on g s ⟶ measurable_on (λx::(real, ?'a::type) cart. complex_mul (f x) (g x)) s

thm MEASURABLE_ON_COMPLEX_INV:

∀ f::(real, ?'a::type) cart ⇒ (real, 2) cart. measurable_on f HOL_Light_Import.UNIV ∧ negligible (GSPEC (λGEN%PVAR%2538::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2538 (f x = Cx (0::real)) x)) ⟶ measurable_on (λx::(real, ?'a::type) cart. complex_inv (f x)) HOL_Light_Import.UNIV

thm MEASURABLE_ON_COMPLEX_DIV:

∀ (f::(real, ?'a::type) cart ⇒ (real, 2) cart) (g::(real, ?'a::type) cart ⇒ (real, 2) cart) s::(real, ?'a::type) cart ⇒ bool. measurable_on f s ∧ measurable_on g HOL_Light_Import.UNIV ∧ negligible (GSPEC (λGEN%PVAR%2540::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2540 (g x = Cx (0::real)) x)) ⟶ measurable_on (λx::(real, ?'a::type) cart. complex_div (f x) (g x)) s

thm DEF_real_measurable_on:

real_measurable_on = (λ(_1923568::real ⇒ real) _1923569::real ⇒ bool. measurable_on (lift ∘ (_1923568 ∘ HOL_Light_Import.drop)) (IMAGE lift _1923569))

thm real_measurable_on:

∀ (f::real ⇒ real) s::real ⇒ bool. real_measurable_on f s = measurable_on (lift ∘ (f ∘ HOL_Light_Import.drop)) (IMAGE lift s)

thm DEF_real_lebesgue_measurable:

$real\_lebesgue\_measurable = (\lambda\_1923580::real \Rightarrow bool.\ real\_measurable\_on\ (\lambda x::real.\ if\ IN\ x\ \_1923580\ then\ 1::real\ else\ (0::real))\ HOL\_Light\_Import.UNIV)$

thm real_lebesgue_measurable:

$\forall s::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s = real\_measurable\_on\ (\lambda x::real.\ if\ IN\ x\ s\ then\ 1::real\ else\ (0::real))\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_UNIV:

$real\_measurable\_on\ (\lambda x::real.\ if\ IN\ x\ (?s::real \Rightarrow bool)\ then\ (?f::real \Rightarrow real)\ x\ else\ (0::real))\ HOL\_Light\_Import.UNIV = real\_measurable\_on\ ?f\ ?s$

thm REAL_LEBESGUE_MEASURABLE:

$\forall s::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s = lebesgue\_measurable\ (IMAGE\ lift\ s)$

thm REAL_MEASURABLE_BOUNDED_BY_INTEGRABLE_IMP_INTEGRABLE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x| \leq g\ x) \longrightarrow real\_integrable\_on\ f\ s$

thm REAL_MEASURABLE_BOUNDED_BY_INTEGRABLE_IMP_ABSOLUTELY_INTEGRABLE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_integrable\_on\ g\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow |f\ x| \leq g\ x) \longrightarrow absolutely\_real\_integrable\_on\ f\ s$

thm INTEGRABLE_SUBINTERVALS_IMP_REAL_MEASURABLE:

$\forall f::real \Rightarrow real.\ (\forall (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])) \longrightarrow real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV$

thm INTEGRABLE_IMP_REAL_MEASURABLE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_integrable\_on\ f\ s \longrightarrow real\_measurable\_on\ f\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_REAL_MEASURABLE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ absolutely\_real\_integrable\_on\ f\ s = (real\_measurable\_on\ f\ s \wedge real\_integrable\_on\ (\lambda x::real.\ |f\ x|)\ s)$

thm REAL_MEASURABLE_ON_COMPOSE_CONTINUOUS:

$\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge real\_continuous\_on\ g\ HOL\_Light\_Import.UNIV \longrightarrow real\_measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_COMPOSE_CONTINUOUS_0:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_continuous\_on\ g\ HOL\_Light\_Import.UNIV \wedge g\ (0::real) = (0::real) \longrightarrow real\_measurable\_on\ (g \circ f)\ s$

thm REAL_MEASURABLE_ON_COMPOSE_CONTINUOUS_OPEN_INTERVAL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall x::real.\ IN\ (f\ x)\ (open\_real\_interval\ (a,\ b))) \wedge real\_continuous\_on\ g\ (open\_real\_interval\ (a,\ b)) \longrightarrow real\_measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_COMPOSE_CONTINUOUS_CLOSED_SET:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_closed\ s \wedge real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall x::real.\ IN\ (f\ x)\ s) \wedge real\_continuous\_on\ g\ s \longrightarrow real\_measurable\_on\ (g \circ f)\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_COMPOSE_CONTINUOUS_CLOSED_SET_0:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_closed\ s \wedge real\_measurable\_on\ f\ t \wedge (\forall x::real.\ IN\ (f\ x)\ s) \wedge real\_continuous\_on\ g\ s \wedge IN\ (0::real)\ s \wedge g\ (0::real) = (0::real) \longrightarrow real\_measurable\_on\ (g \circ f)\ t$

thm CONTINUOUS_IMP_REAL_MEASURABLE_ON:

$\forall f::real \Rightarrow real.\ real\_continuous\_on\ f\ HOL\_Light\_Import.UNIV \longrightarrow real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_CONST:

$\forall k::real.\ real\_measurable\_on\ (\lambda x::real.\ k)\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_0:

$\forall s::real \Rightarrow bool.\ real\_measurable\_on\ (\lambda x::real.\ 0::real)\ s$

thm REAL_MEASURABLE_ON_LMUL:

$\forall (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm REAL_MEASURABLE_ON_RMUL:

$\forall (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm REAL_MEASURABLE_ON_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ -\ f\ x)\ s$

thm REAL_MEASURABLE_ON_NEG_EQ:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ (\lambda x::real.\ -\ f\ x)\ s = real\_measurable\_on\ f\ s$

thm REAL_MEASURABLE_ON_ABS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ |f\ x|)\ s$

thm REAL_MEASURABLE_ON_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_measurable\_on\ g\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm REAL_MEASURABLE_ON_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_measurable\_on\ g\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ f\ x - g\ x)\ s$

thm REAL_MEASURABLE_ON_MAX:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_measurable\_on\ g\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ max\ (f\ x)\ (g\ x))\ s$

thm REAL_MEASURABLE_ON_MIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_measurable\_on\ g\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ min\ (f\ x)\ (g\ x))\ s$

thm REAL_MEASURABLE_ON_SPIKE_SET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_negligible\ (HOL\_Light\_Import.UNION\ (DIFF\ s\ t)\ (DIFF\ t\ s)) \longrightarrow real\_measurable\_on\ f\ s \longrightarrow real\_measurable\_on\ f\ t$

thm REAL_MEASURABLE_ON_RESTRICT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge real\_lebesgue\_measurable\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ if\ IN\ x\ s\ then\ f\ x\ else\ (0::real))\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_LIMIT:

$\forall (f::nat \Rightarrow real \Rightarrow real)\ (g::real \Rightarrow real)\ (s::real \Rightarrow bool)\ k::real \Rightarrow bool.\ (\forall n::nat.\ real\_measurable\_on\ (f\ n)\ s) \wedge real\_negligible\ k \wedge (\forall x::real.\ IN\ x\ (DIFF\ s\ k) \longrightarrow --->\ (\lambda n::nat.\ f\ n\ x)\ (g\ x)\ sequentially) \longrightarrow real\_measurable\_on\ g\ s$

thm ABSOLUTELY_REAL_INTEGRABLE_BOUNDED_MEASURABLE_PRODUCT:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s \wedge real\_bounded\ (IMAGE\ f\ s) \wedge absolutely\_real\_integrable\_on\ g\ s \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x * g\ x)\ s$

thm REAL_COMPLEX_MEASURABLE_ON:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_measurable\_on\ f\ s = measurable\_on\ (Cx \circ (f \circ HOL\_Light\_Import.drop))\ (IMAGE\ lift\ s)$

thm REAL_MEASURABLE_ON_INV:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \wedge real\_negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2541::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2541\ (f\ x = (0::real))\ x)) \longrightarrow real\_measurable\_on\ (\lambda x::real.\ inverse\_class.inverse\ (f\ x))\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_MUL:

$\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ real\_measurable\_on\ f\ (?s::real \Rightarrow bool) \wedge real\_measurable\_on\ g\ ?s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ f\ x * g\ x)\ ?s$

thm REAL_MEASURABLE_ON_DIV:

$\forall (f::real \Rightarrow real)\ g::real \Rightarrow real.\ real\_measurable\_on\ f\ (?s::real \Rightarrow bool) \wedge real\_measurable\_on\ g\ HOL\_Light\_Import.UNIV \wedge real\_negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2542::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2542\ (g\ x = (0::real))\ x)) \longrightarrow real\_measurable\_on\ (\lambda x::real.\ f\ x\ /\ g\ x)\ ?s$

thm REAL_MEASURABLE_IMP_REAL_LEBESGUE_MEASURABLE:

$\forall s::real \Rightarrow bool.\ real\_measurable\ s \longrightarrow real\_lebesgue\_measurable\ s$

thm REAL_LEBESGUE_MEASURABLE_EMPTY:

$real\_lebesgue\_measurable\ EMPTY$

thm REAL_LEBESGUE_MEASURABLE_UNIV:

$real\_lebesgue\_measurable\ HOL\_Light\_Import.UNIV$

thm REAL_LEBESGUE_MEASURABLE_COMPACT:

$\forall s::real \Rightarrow bool.\ real\_compact\ s \longrightarrow real\_lebesgue\_measurable\ s$

thm REAL_MEASURABLE_REAL_INTERVAL_conjunct1:

$\forall (a::real)\ b::real.\ real\_measurable\ (open\_real\_interval\ (a,\ b))$

thm REAL_MEASURABLE_REAL_INTERVAL_conjunct0:

$\forall (a::real)\ b::real.\ real\_measurable\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_LEBESGUE_MEASURABLE_INTERVAL:

$(\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (closed\_real\_interval\ [(a,\ b)])) \wedge (\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (open\_real\_interval\ (a,\ b)))$

thm REAL_LEBESGUE_MEASURABLE_INTER:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s \wedge real\_lebesgue\_measurable\ t \longrightarrow real\_lebesgue\_measurable\ (HOL\_Light\_Import.INTER\ s\ t)$

thm REAL_LEBESGUE_MEASURABLE_UNION:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s \wedge real\_lebesgue\_measurable\ t \longrightarrow real\_lebesgue\_measurable\ (HOL\_Light\_Import.UNION\ s\ t)$

thm REAL_LEBESGUE_MEASURABLE_COMPL:

$\forall s::real \Rightarrow bool.\ real\_lebesgue\_measurable\ (DIFF\ HOL\_Light\_Import.UNIV\ s) = real\_lebesgue\_measurable\ s$

thm REAL_LEBESGUE_MEASURABLE_DIFF:

$\forall (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s \wedge real\_lebesgue\_measurable\ t \longrightarrow real\_lebesgue\_measurable\ (DIFF\ s\ t)$

thm REAL_LEBESGUE_MEASURABLE_ON_SUBINTERVALS:

$\forall s::real \Rightarrow bool.\ real\_lebesgue\_measurable\ s = (\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (HOL\_Light\_Import.INTER\ s\ (closed\_real\_interval\ [(a,\ b)])))$

thm REAL_LEBESGUE_MEASURABLE_CLOSED:

$\forall s::real \Rightarrow bool.\ real\_closed\ s \longrightarrow real\_lebesgue\_measurable\ s$

thm REAL_LEBESGUE_MEASURABLE_OPEN:

$\forall s::real \Rightarrow bool.\ real\_open\ s \longrightarrow real\_lebesgue\_measurable\ s$

thm REAL_LEBESGUE_MEASURABLE_UNIONS:

$\forall f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow real\_lebesgue\_measurable\ s) \longrightarrow real\_lebesgue\_measurable\ (UNIONS\ f)$

thm REAL_LEBESGUE_MEASURABLE_COUNTABLE_UNIONS_EXPLICIT:

$\forall s::nat \Rightarrow real \Rightarrow bool.\ (\forall n::nat.\ real\_lebesgue\_measurable\ (s\ n)) \longrightarrow real\_lebesgue\_measurable\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%2543::real \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2543\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_LEBESGUE_MEASURABLE_COUNTABLE_UNIONS:

$\forall f::(real \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow real\_lebesgue\_measurable\ s) \longrightarrow real\_lebesgue\_measurable\ (UNIONS\ f)$

thm REAL_LEBESGUE_MEASURABLE_COUNTABLE_INTERS:

$\forall f::(real \Rightarrow bool) \Rightarrow bool.\ COUNTABLE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow real\_lebesgue\_measurable\ s) \longrightarrow real\_lebesgue\_measurable\ (INTERS\ f)$

thm REAL_LEBESGUE_MEASURABLE_COUNTABLE_INTERS_EXPLICIT:

$\forall s::nat \Rightarrow real \Rightarrow bool.\ (\forall n::nat.\ real\_lebesgue\_measurable\ (s\ n)) \longrightarrow real\_lebesgue\_measurable\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%2544::real \Rightarrow bool.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%2544\ (IN\ n\ HOL\_Light\_Import.UNIV)\ (s\ n))))$

thm REAL_LEBESGUE_MEASURABLE_INTERS:

$\forall f::(real \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::real \Rightarrow bool.\ IN\ s\ f \longrightarrow real\_lebesgue\_measurable\ s) \longrightarrow real\_lebesgue\_measurable\ (INTERS\ f)$

thm REAL_LEBESGUE_MEASURABLE_IFF_MEASURABLE:

$\forall s::real \Rightarrow bool.\ real\_bounded\ s \longrightarrow real\_lebesgue\_measurable\ s = real\_measurable\ s$

thm REAL_MEASURABLE_ON_LEBESGUE_MEASURABLE_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ SUBSET\ s\ t \wedge real\_measurable\_on\ f\ t \wedge real\_lebesgue\_measurable\ s \longrightarrow real\_measurable\_on\ f\ s$

thm REAL_MEASURABLE_ON_MEASURABLE_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ SUBSET\ s\ t \wedge real\_measurable\_on\ f\ t \wedge real\_measurable\ s \longrightarrow real\_measurable\_on\ f\ s$

thm REAL_CONTINUOUS_IMP_REAL_MEASURABLE_ON_CLOSED_SUBSET:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_continuous\_on\ f\ s \wedge real\_closed\ s \longrightarrow real\_measurable\_on\ f\ s$

thm REAL_MEASURABLE_ON_CASES:

$\forall (P::real \Rightarrow bool)\ (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2547::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2547\ (P\ x)\ x)) \wedge real\_measurable\_on\ f\ s \wedge real\_measurable\_on\ g\ s \longrightarrow real\_measurable\_on\ (\lambda x::real.\ if\ P\ x\ then\ f\ x\ else\ g\ x)\ s$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_LT:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall a::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2548::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2548\ (f\ x < a)\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_LE:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall a::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2549::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2549\ (f\ x \leq a)\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_GT:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall a::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2550::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2550\ (a < f\ x)\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN_HALFSPACE_GE:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall a::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2551::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2551\ (a \leq f\ x)\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN_INTERVAL:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2552::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2552\ (IN\ (f\ x)\ (open\_real\_interval\ (a,\ b)))\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_CLOSED_INTERVAL:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2553::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2553\ (IN\ (f\ x)\ (closed\_real\_interval\ [(a,\ b)]))\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_OPEN:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall t::real \Rightarrow bool.\ real\_open\ t \longrightarrow real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2554::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2554\ (IN\ (f\ x)\ t)\ x)))$

thm REAL_MEASURABLE_ON_PREIMAGE_CLOSED:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\forall t::real \Rightarrow bool.\ real\_closed\ t \longrightarrow real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2555::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2555\ (IN\ (f\ x)\ t)\ x)))$

thm REAL_MEASURABLE_ON_SIMPLE_FUNCTION_LIMIT:

$\forall f::real \Rightarrow real.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV = (\exists g::nat \Rightarrow real \Rightarrow real.\ (\forall n::nat.\ real\_measurable\_on\ (g\ n)\ HOL\_Light\_Import.UNIV) \land (\forall n::nat.\ FINITE\ (IMAGE\ (g\ n)\ HOL\_Light\_Import.UNIV)) \land (\forall x::real.\ ---> (\lambda n::nat.\ g\ n\ x)\ (f\ x)\ sequentially))$

thm REAL_LEBESGUE_MEASURABLE_PREIMAGE_OPEN:

$\forall (f::real \Rightarrow real)\ t::real \Rightarrow bool.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \land real\_open\ t \longrightarrow real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2556::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2556\ (IN\ (f\ x)\ t)\ x))$

thm REAL_LEBESGUE_MEASURABLE_PREIMAGE_CLOSED:

$\forall (f::real \Rightarrow real)\ t::real \Rightarrow bool.\ real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV \land real\_closed\ t \longrightarrow real\_lebesgue\_measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2557::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2557\ (IN\ (f\ x)\ t)\ x))$

thm REAL_CONTINUOUS_MEASURE_IN_HALFSPACE_LE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (a::real)\ i::nat.\ measurable\ s \land (1::nat) \leq i \land i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow real\_continuous\ (\lambda a::real.\ HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ s\ (GSPEC\ (\lambda GEN\%PVAR\%2562::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2562\ (\$\ x\ i \leq a)\ x))))\ (atreal\ a)$

thm REAL_SECOND_MEAN_VALUE_THEOREM_FULL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \land real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \land (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \land IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \land x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \land has\_real\_integral\ (\lambda x::real.\ g\ x * f\ x)\ (g\ a * real\_integral\ (closed\_real\_interval\ [(a,\ c)])\ f + g\ b * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)\ (closed\_real\_interval\ [(a,\ b)]))$

thm REAL_SECOND_MEAN_VALUE_THEOREM:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \land real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \land (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \land IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \land x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \land real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ (\lambda x::real.\ g\ x * f\ x) = g\ a * real\_integral\ (closed\_real\_interval\ [(a,\ c)])\ f + g\ b * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)$

thm REAL_SECOND_MEAN_VALUE_THEOREM_GEN_FULL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ (b::real)\ (u::real)\ v::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow u \leq g\ x \wedge g\ x \leq v) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \wedge has\_real\_integral\ (\lambda x::real.\ g\ x * f\ x)\ (u * real\_integral\ (closed\_real\_interval\ [(a,\ c)])\ f + v * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)\ (closed\_real\_interval\ [(a,\ b)]))$

thm REAL_SECOND_MEAN_VALUE_THEOREM_GEN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ (b::real)\ (u::real)\ v::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (open\_real\_interval\ (a,\ b)) \longrightarrow u \leq g\ x \wedge g\ x \leq v) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ (\lambda x::real.\ g\ x * f\ x) = u * real\_integral\ (closed\_real\_interval\ [(a,\ c)])\ f + v * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)$

thm REAL_SECOND_MEAN_VALUE_THEOREM_BONNET_FULL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (0::real) \leq g\ x) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \wedge has\_real\_integral\ (\lambda x::real.\ g\ x * f\ x)\ (g\ b * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)\ (closed\_real\_interval\ [(a,\ b)]))$

thm REAL_SECOND_MEAN_VALUE_THEOREM_BONNET:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \wedge real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (0::real) \leq g\ x) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow (\exists c::real.\ IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_integral\ (closed\_real\_interval\ [(a,\ b)])\ (\lambda x::real.\ g\ x * f\ x) = g\ b * real\_integral\ (closed\_real\_interval\ [(c,\ b)])\ f)$

thm REAL_INTEGRABLE_INCREASING_PRODUCT:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ g\ x * f\ x)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_INCREASING_PRODUCT_UNIV:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ B::real.\ real\_integrable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall (x::real)\ y::real.\ x \leq y \longrightarrow g\ x \leq g\ y) \wedge (\forall x::real.\ |g\ x| \leq B) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ g\ x * f\ x)\ HOL\_Light\_Import.UNIV$

thm REAL_INTEGRABLE_INCREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_DECREASING_PRODUCT:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ y \leq g\ x) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ g\ x * f\ x)\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_INTEGRABLE_DECREASING_PRODUCT_UNIV:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ B::real.\ real\_integrable\_on\ f\ HOL\_Light\_Import.UNIV \wedge (\forall (x::real)\ y::real.\ x \leq y \longrightarrow g\ y \leq g\ x) \wedge (\forall x::real.\ |g\ x| \leq B) \longrightarrow real\_integrable\_on\ (\lambda x::real.\ g\ x * f\ x)\ HOL\_Light\_Import.UNIV$

thm REAL_INTEGRABLE_DECREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ y \leq f\ x) \longrightarrow real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_MEASURABLE_ON_INCREASING_UNIV:

$\forall f::real \Rightarrow real.\ (\forall (x::real)\ y::real.\ x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV$

thm REAL_LEBESGUE_MEASURABLE_INTERVAL_conjunct1:

$\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (open\_real\_interval\ (a,\ b))$

thm REAL_LEBESGUE_MEASURABLE_INTERVAL_conjunct0:

$\forall (a::real)\ b::real.\ real\_lebesgue\_measurable\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_MEASURABLE_ON_INCREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow real\_measurable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm REAL_MEASURABLE_ON_DECREASING_UNIV:

$\forall f::real \Rightarrow real.\ (\forall (x::real)\ y::real.\ x \leq y \longrightarrow f\ y \leq f\ x) \longrightarrow real\_measurable\_on\ f\ HOL\_Light\_Import.UNIV$

thm REAL_MEASURABLE_ON_DECREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ x \leq y \longrightarrow f\ y \leq f\ x) \longrightarrow real\_measurable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_INCREASING_PRODUCT:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ x \leq y \longrightarrow f\ x \leq f\ y)\ \wedge\ absolutely\_real\_integrable\_on\ g\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x * g\ x)\ (closed\_real\_interval\ [(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_INCREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow absolutely\_real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_DECREASING_PRODUCT:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ x \leq y \longrightarrow f\ y \leq f\ x)\ \wedge\ absolutely\_real\_integrable\_on\ g\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow absolutely\_real\_integrable\_on\ (\lambda x::real.\ f\ x * g\ x)\ (closed\_real\_interval\ [(a,\ b)])$

thm ABSOLUTELY_REAL_INTEGRABLE_DECREASING:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ IN\ y\ (closed\_real\_interval\ [(a,\ b)])\ \wedge\ x \leq y \longrightarrow f\ y \leq f\ x) \longrightarrow absolutely\_real\_integrable\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm DEF_has_bounded_real_variation_on:

$has\_bounded\_real\_variation\_on = (\lambda(\_1927515::real \Rightarrow real)\ \_1927516::real \Rightarrow bool.\ has\_bounded\_variation\_on\ (lift \circ (\_1927515 \circ HOL\_Light\_Import.drop))\ (IMAGE\ lift\ \_1927516))$

thm has_bounded_real_variation_on:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s = has\_bounded\_variation\_on\ (lift \circ (f \circ HOL\_Light\_Import.drop))\ (IMAGE\ lift\ s)$

thm DEF_real_variation:

$real\_variation = (\lambda(\_1927527::real \Rightarrow bool)\ \_1927528::real \Rightarrow real.\ vector\_variation\ (IMAGE\ lift\ \_1927527)\ (lift \circ (\_1927528 \circ HOL\_Light\_Import.drop)))$

thm real_variation:

$\forall (s::real \Rightarrow bool)\ f::real \Rightarrow real.\ real\_variation\ s\ f = vector\_variation\ (IMAGE\ lift\ s)\ (lift \circ (f \circ HOL\_Light\_Import.drop))$

thm HAS_BOUNDED_REAL_VARIATION_ON_EQ:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ g\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_SUBSET:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow has\_bounded\_real\_variation\_on\ f\ t$

thm HAS_BOUNDED_REAL_VARIATION_ON_LMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ c * f\ x)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_RMUL:

$\forall (f::real \Rightarrow real)\ (c::real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ x * c)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ -\ f\ x)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_ADD:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge has\_bounded\_real\_variation\_on\ g\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ x + g\ x)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_SUB:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge has\_bounded\_real\_variation\_on\ g\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ x - g\ x)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_NULL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ b \leq a \longrightarrow has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm HAS_BOUNDED_REAL_VARIATION_ON_EMPTY:

$\forall f::real \Rightarrow real.\ has\_bounded\_real\_variation\_on\ f\ EMPTY$

thm HAS_BOUNDED_REAL_VARIATION_ON_ABS:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ |f\ x|)\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_MAX:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge has\_bounded\_real\_variation\_on\ g\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ max\ (f\ x)\ (g\ x))\ s$

thm HAS_BOUNDED_REAL_VARIATION_ON_MIN:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge has\_bounded\_real\_variation\_on\ g\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ min\ (f\ x)\ (g\ x))\ s$

thm IMAGE_DROP_INTERVAL_conjunct1:

$IMAGE\ HOL\_Light\_Import.drop\ (open\_interval\ (?a::(real, unit)\ cart, ?b::(real, unit)\ cart)) = open\_real\_interval\ (HOL\_Light\_Import.drop\ ?a, HOL\_Light\_Import.drop\ ?b)$

thm IMAGE_DROP_INTERVAL_conjunct0:

$IMAGE\ HOL\_Light\_Import.drop\ (closed\_interval\ [(?a::(real, unit)\ cart, ?b::(real, unit)\ cart)]) = closed\_real\_interval\ [(HOL\_Light\_Import.drop\ ?a, HOL\_Light\_Import.drop\ ?b)]$

thm HAS_BOUNDED_REAL_VARIATION_ON_IMP_BOUNDED_ON_INTERVAL:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a, b)]) \longrightarrow real\_bounded\ (IMAGE\ f\ (closed\_real\_interval\ [(a, b)]))$

thm HAS_BOUNDED_REAL_VARIATION_ON_MUL:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a, b)]) \wedge has\_bounded\_real\_variation\_on\ g\ (closed\_real\_interval\ [(a, b)]) \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ x * g\ x)\ (closed\_real\_interval\ [(a, b)])$

thm REAL_VARIATION_POS_LE:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow (0::real) \leq real\_variation\ s\ f$

thm REAL_VARIATION_GE_ABS_FUNCTION:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ s \wedge SUBSET\ (closed\_real\_interval\ [(a, b)])\ s \wedge closed\_real\_interval\ [(a, b)] \neq EMPTY \longrightarrow |f\ b - f\ a| \leq real\_variation\ s\ f$

thm REAL_VARIATION_GE_FUNCTION:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ s \wedge SUBSET\ (closed\_real\_interval\ [(a, b)])\ s \wedge closed\_real\_interval\ [(a, b)] \neq EMPTY \longrightarrow f\ b - f\ a \leq real\_variation\ s\ f$

thm REAL_VARIATION_MONOTONE:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ t::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge SUBSET\ t\ s \longrightarrow real\_variation\ t\ f \leq real\_variation\ s\ f$

thm REAL_VARIATION_NEG:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ s\ (\lambda x::real.\ - f\ x) = real\_variation\ s\ f$

thm REAL_VARIATION_TRIANGLE:

$\forall (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ f\ s \wedge has\_bounded\_real\_variation\_on\ g\ s \longrightarrow real\_variation\ s\ (\lambda x::real.\ f\ x + g\ x) \leq real\_variation\ s\ f + real\_variation\ s\ g$

thm HAS_BOUNDED_REAL_VARIATION_ON_COMBINE:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ a \leq c \wedge c \leq b \longrightarrow has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) = (has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ c)]) \wedge has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(c,\ b)]))$

thm REAL_VARIATION_COMBINE:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ a \leq c \wedge c \leq b \wedge has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow real\_variation\ (closed\_real\_interval\ [(a,\ c)])\ f + real\_variation\ (closed\_real\_interval\ [(c,\ b)])\ f = real\_variation\ (closed\_real\_interval\ [(a,\ b)])\ f$

thm REAL_VARIATION_MINUS_FUNCTION_MONOTONE:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ (c::real)\ d::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \wedge SUBSET\ (closed\_real\_interval\ [(c,\ d)])\ (closed\_real\_interval\ [(a,\ b)]) \wedge closed\_real\_interval\ [(c,\ d)] \neq EMPTY \longrightarrow real\_variation\ (closed\_real\_interval\ [(c,\ d)])\ f - (f\ d - f\ c) \leq real\_variation\ (closed\_real\_interval\ [(a,\ b)])\ f - (f\ b - f\ a)$

thm INCREASING_BOUNDED_REAL_VARIATION:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)])$

thm INCREASING_REAL_VARIATION:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ closed\_real\_interval\ [(a,\ b)] \neq EMPTY \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \longrightarrow real\_variation\ (closed\_real\_interval\ [(a,\ b)])\ f = f\ b - f\ a$

thm HAS_BOUNDED_REAL_VARIATION_AFFINITY2_EQ:

$\forall (m::real)\ (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (m * x + c))\ (IMAGE\ (\lambda x::real.\ inverse\_class.inverse\ m * x + - (inverse\_class.inverse\ m * c))\ s) = (m = (0::real) \vee has\_bounded\_real\_variation\_on\ f\ s)$

thm REAL_VARIATION_AFFINITY2:

$\forall (m::real)\ (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ (IMAGE\ (\lambda x::real.\ inverse\_class.inverse\ m * x + - (inverse\_class.inverse\ m * c))\ s)\ (\lambda x::real.\ f\ (m * x + c)) = (if\ m = (0::real)\ then\ 0::real\ else\ real\_variation\ s\ f)$

thm HAS_BOUNDED_REAL_VARIATION_AFFINITY_EQ:

$\forall (m::real)\ (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (m * x + c))\ s = (m = (0::real) \vee has\_bounded\_real\_variation\_on\ f\ (IMAGE\ (\lambda x::real.\ m * x + c)\ s))$

thm REAL_VARIATION_AFFINITY:

$\forall (m::real)\ (c::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ s\ (\lambda x::real.\ f\ (m * x + c)) = (if\ m = (0::real)\ then\ 0::real\ else\ real\_variation\ (IMAGE\ (\lambda x::real.\ m * x + c)\ s)\ f)$

thm HAS_BOUNDED_REAL_VARIATION_TRANSLATION2_EQ:

$\forall (a::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (a + x))\ (IMAGE\ (op + (- a))\ s) = has\_bounded\_real\_variation\_on\ f\ s$

thm REAL_VARIATION_TRANSLATION2:

$\forall (a::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ (IMAGE\ (op + (- a))\ s)\ (\lambda x::real.\ f\ (a + x)) = real\_variation\ s\ f$

thm HAS_BOUNDED_REAL_VARIATION_TRANSLATION_EQ:

$\forall (a::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (a + x))\ s = has\_bounded\_real\_variation\_on\ f\ (IMAGE\ (op + a)\ s)$

thm REAL_VARIATION_TRANSLATION:

$\forall (a::real)\ (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ s\ (\lambda x::real.\ f\ (a + x)) = real\_variation\ (IMAGE\ (op + a)\ s)\ f$

thm HAS_BOUNDED_REAL_VARIATION_TRANSLATION_EQ_INTERVAL:

$\forall (a::real)\ (f::real \Rightarrow real)\ (u::real)\ v::real.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (a + x))\ (closed\_real\_interval\ [(u,\ v)]) = has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a + u,\ a + v)])$

thm REAL_VARIATION_TRANSLATION_INTERVAL:

$\forall (a::real)\ (f::real \Rightarrow real)\ (u::real)\ v::real.\ real\_variation\ (closed\_real\_interval\ [(u,\ v)])\ (\lambda x::real.\ f\ (a + x)) = real\_variation\ (closed\_real\_interval\ [(a + u,\ a + v)])\ f$

thm HAS_BOUNDED_REAL_VARIATION_TRANSLATION:

$\forall (f::real \Rightarrow real)\ (s::real \Rightarrow bool)\ a::real.\ has\_bounded\_real\_variation\_on\ f\ s \longrightarrow has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (a + x))\ (IMAGE\ (op + (- a))\ s)$

thm HAS_BOUNDED_REAL_VARIATION_REFLECT2_EQ:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (- x))\ (IMAGE\ uminus\ s) = has\_bounded\_real\_variation\_on\ f\ s$

thm REAL_VARIATION_REFLECT2:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ (IMAGE\ uminus\ s)\ (\lambda x::real.\ f\ (- x)) = real\_variation\ s\ f$

thm HAS_BOUNDED_REAL_VARIATION_REFLECT_EQ:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (-\ x))\ s = has\_bounded\_real\_variation\_on\ f\ (IMAGE\ uminus\ s)$

thm REAL_VARIATION_REFLECT:

$\forall (f::real \Rightarrow real)\ s::real \Rightarrow bool.\ real\_variation\ s\ (\lambda x::real.\ f\ (-\ x)) = real\_variation\ (IMAGE\ uminus\ s)\ f$

thm HAS_BOUNDED_REAL_VARIATION_REFLECT_EQ_INTERVAL:

$\forall (f::real \Rightarrow real)\ (u::real)\ v::real.\ has\_bounded\_real\_variation\_on\ (\lambda x::real.\ f\ (-\ x))\ (closed\_real\_interval\ [(u,\ v)]) = has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(-\ v,\ -\ u)])$

thm REAL_VARIATION_REFLECT_INTERVAL:

$\forall (f::real \Rightarrow real)\ (u::real)\ v::real.\ real\_variation\ (closed\_real\_interval\ [(u,\ v)])\ (\lambda x::real.\ f\ (-\ x)) = real\_variation\ (closed\_real\_interval\ [(-\ v,\ -\ u)])\ f$

thm HAS_BOUNDED_REAL_VARIATION_DARBOUX:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) = (\exists (g::real \Rightarrow real)\ h::real \Rightarrow real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow g\ x \leq g\ y) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow h\ x \leq h\ y) \wedge (\forall x::real.\ f\ x = g\ x - h\ x))$

thm HAS_BOUNDED_REAL_VARIATION_DARBOUX_STRICT:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) = (\exists (g::real \Rightarrow real)\ h::real \Rightarrow real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x < y \longrightarrow g\ x < g\ y) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x < y \longrightarrow h\ x < h\ y) \wedge (\forall x::real.\ f\ x = g\ x - h\ x))$

thm INCREASING_LEFT_LIMIT:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \wedge IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists l::real.\ --->\ f\ l\ (within\ (atreal\ c)\ (closed\_real\_interval\ [(a,\ c)])))$

thm DECREASING_LEFT_LIMIT:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ y \leq f\ x) \wedge IN\ c\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow (\exists l::real.\ --->\ f\ l\ (within\ (atreal\ c)\ (closed\_real\_interval\ [(a,\ c)])))$

thm INCREASING_RIGHT_LIMIT:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::real)\ c::real.\ (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x \leq y \longrightarrow f\ x \leq f\ y) \wedge$

IN c (*closed_real_interval* [(a, b)]) $\longrightarrow$ ($\exists$ l::*real*. $---$> f l (*within* (*atreal* c) (*closed_real_interval* [(c, b)])))

thm DECREASING_RIGHT_LIMIT:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. ($\forall$ (x::*real*) y::*real*. *IN* x (*closed_real_interval* [(a, b)]) $\wedge$ *IN* y (*closed_real_interval* [(a, b)]) $\wedge$ $x \leq y$ $\longrightarrow$ $f\ y \leq f\ x$) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ ($\exists$ l::*real*. $---$> f l (*within* (*atreal* c) (*closed_real_interval* [(c, b)])))

thm HAS_BOUNDED_REAL_VARIATION_LEFT_LIMIT:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ ($\exists$ l::*real*. $---$> f l (*within* (*atreal* c) (*closed_real_interval* [(a, c)])))

thm HAS_BOUNDED_REAL_VARIATION_RIGHT_LIMIT:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ ($\exists$ l::*real*. $---$> f l (*within* (*atreal* c) (*closed_real_interval* [(c, b)])))

thm REAL_VARIATION_CONTINUOUS_LEFT:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ *real_continuous* (λx::*real*. *real_variation* (*closed_real_interval* [(a, x)]) f) (*within* (*atreal* c) (*closed_real_interval* [(a, c)])) = *real_continuous* f (*within* (*atreal* c) (*closed_real_interval* [(a, c)]))

thm REAL_VARIATION_CONTINUOUS_RIGHT:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ *real_continuous* (λx::*real*. *real_variation* (*closed_real_interval* [(a, x)]) f) (*within* (*atreal* c) (*closed_real_interval* [(c, b)])) = *real_continuous* f (*within* (*atreal* c) (*closed_real_interval* [(c, b)]))

thm REAL_VARIATION_CONTINUOUS:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) (b::*real*) c::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\wedge$ *IN* c (*closed_real_interval* [(a, b)]) $\longrightarrow$ *real_continuous* (λx::*real*. *real_variation* (*closed_real_interval* [(a, x)]) f) (*within* (*atreal* c) (*closed_real_interval* [(a, b)])) = *real_continuous* f (*within* (*atreal* c) (*closed_real_interval* [(a, b)]))

thm HAS_BOUNDED_REAL_VARIATION_DARBOUX_STRONG:

$\forall$ (f::*real* $\Rightarrow$ *real*) (a::*real*) b::*real*. *has_bounded_real_variation_on* f (*closed_real_interval* [(a, b)]) $\longrightarrow$ ($\exists$ (g::*real* $\Rightarrow$ *real*) h::*real* $\Rightarrow$ *real*. ($\forall$ x::*real*. $f\ x = g\ x - h\ x$) $\wedge$ ($\forall$ (x::*real*) y::*real*. *IN* x (*closed_real_interval* [(a, b)]) $\wedge$ *IN* y (*closed_real_interval* [(a, b)]) $\wedge$ $x \leq y$ $\longrightarrow$ $g\ x \leq g\ y$) $\wedge$ ($\forall$ (x::*real*) y::*real*. *IN* x (*closed_real_interval* [(a, b)]) $\wedge$ *IN* y (*closed_real_interval* [(a, b)]) $\wedge$ $x \leq y$ $\longrightarrow$ $h\ x \leq h\ y$) $\wedge$

$(\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x < y \longrightarrow g\ x < g\ y) \wedge (\forall (x::real)\ y::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge IN\ y\ (closed\_real\_interval\ [(a,\ b)]) \wedge x < y \longrightarrow h\ x < h\ y) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_continuous\ f\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ x)])) \longrightarrow real\_continuous\ g\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ x)])) \wedge real\_continuous\ h\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ x)]))) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_continuous\ f\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(x,\ b)])) \longrightarrow real\_continuous\ g\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(x,\ b)])) \wedge real\_continuous\ h\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(x,\ b)]))) \wedge (\forall x::real.\ IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge real\_continuous\ f\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)])) \longrightarrow real\_continuous\ g\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)])) \wedge real\_continuous\ h\ (within\ (atreal\ x)\ (closed\_real\_interval\ [(a,\ b)]))))$

thm HAS_BOUNDED_REAL_VARIATION_COUNTABLE_DISCONTINUITIES:

$\forall (f::real \Rightarrow real)\ (a::real)\ b::real.\ has\_bounded\_real\_variation\_on\ f\ (closed\_real\_interval\ [(a,\ b)]) \longrightarrow COUNTABLE\ (GSPEC\ (\lambda GEN\%PVAR\%2567::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%2567\ (IN\ x\ (closed\_real\_interval\ [(a,\ b)]) \wedge \neg\ real\_continuous\ f\ (atreal\ x))\ x))$

thm LEBESGUE_DENSITY_THEOREM:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ lebesgue\_measurable\ s \longrightarrow (\exists k::(real,\ ?'a::type)\ cart \Rightarrow bool.\ negligible\ k \wedge (\forall x::(real,\ ?'a::type)\ cart.\ \neg\ IN\ x\ k \longrightarrow --->\ (\lambda e::real.\ HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ s\ (cball\ (x,\ e)))\ /\ HOL\_Light\_Import.measure\ (cball\ (x,\ e)))\ (if\ IN\ x\ s\ then\ 1::real\ else\ (0::real))\ (within\ (atreal\ (0::real))\ (GSPEC\ (\lambda GEN\%PVAR\%2568::real.\ \exists e::real.\ SETSPEC\ GEN\%PVAR\%2568\ ((0::real) < e)\ e)))))$

thm INVARIANCE_OF_DOMAIN:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.open\ s \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ f\ s)$

thm INVARIANCE_OF_DIMENSION:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge HOL\_Light\_Import.open\ s \wedge s \neq EMPTY \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow dimindex\ HOL\_Light\_Import.UNIV \leq dimindex\ HOL\_Light\_Import.UNIV$

thm HOMEOMORPHIC_UNIV_UNIV:

$homeomorphic\ HOL\_Light\_Import.UNIV\ HOL\_Light\_Import.UNIV = (dimindex\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV)$

thm INVARIANCE_OF_DOMAIN_GEN:

$\forall f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV \wedge continuous\_on\ f\ (?s::(real,\ ?'b::type)$

$cart \Rightarrow bool) \wedge HOL\_Light\_Import.open\ ?s \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ ?s \wedge IN\ y\ ?s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow HOL\_Light\_Import.open\ (IMAGE\ f\ ?s)$

thm CONTINUOUS_IMAGE_SUBSET_INTERIOR:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ continuous\_on\ f\ s \wedge dimindex\ HOL\_Light\_Import.UNIV \le dimindex\ HOL\_Light\_Import.UNIV \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ IN\ x\ s \wedge IN\ y\ s \wedge f\ x = f\ y \longrightarrow x = y) \longrightarrow SUBSET\ (IMAGE\ f\ (interior\ s))\ (interior\ (IMAGE\ f\ s))$

thm HOMEOMORPHIC_INTERIORS_SAME_DIMENSION:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ dimindex\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV \wedge homeomorphic\ s\ t \longrightarrow homeomorphic\ (interior\ s)\ (interior\ t)$

thm HOMEOMORPHIC_INTERIORS:

$\forall (s::(real,\ ?'b::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ homeomorphic\ s\ t \wedge (interior\ s = EMPTY) = (interior\ t = EMPTY) \longrightarrow homeomorphic\ (interior\ s)\ (interior\ t)$

thm PYTHAGORAS:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ orthogonal\ (vector\_sub\ A\ B)\ (vector\_sub\ C\ B) \longrightarrow (vector\_norm\ (vector\_sub\ C\ A))^2 = (vector\_norm\ (vector\_sub\ B\ A))^2 + (vector\_norm\ (vector\_sub\ C\ B))^2$

thm DEF_vector_angle:

$vector\_angle = (\lambda(\_1934208::(real,\ ?'a::type)\ cart)\ \_1934209::(real,\ ?'a::type)\ cart.\ if\ \_1934208 = vec\ (0::nat) \vee \_1934209 = vec\ (0::nat)\ then\ pi\ /\ real\_of\_nat\ (2::nat)\ else\ acs\ (dot\ \_1934208\ \_1934209\ /\ (vector\_norm\ \_1934208 * vector\_norm\ \_1934209)))$

thm vector_angle:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_angle\ x\ y = (if\ x = vec\ (0::nat) \vee y = vec\ (0::nat)\ then\ pi\ /\ real\_of\_nat\ (2::nat)\ else\ acs\ (dot\ x\ y\ /\ (vector\_norm\ x * vector\_norm\ y)))$

thm VECTOR_ANGLE_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow vector\_angle\ (f\ x)\ (f\ y) = vector\_angle\ x\ y$

thm VECTOR_ANGLE_REFL:

$\forall x::(real,\ ?'a::type)\ cart.\ vector\_angle\ x\ x = (if\ x = vec\ (0::nat)\ then\ pi\ /\ real\_of\_nat\ (2::nat)\ else\ (0::real))$

thm VECTOR_ANGLE_SYM:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle x y = vector_angle y x

thm VECTOR_ANGLE_LMUL:

∀ (a::real) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle (% a x) y = (if a = (0::real) then pi / real_of_nat (2::nat) else if (0::real) ≤ a then vector_angle x y else pi − vector_angle x y)

thm VECTOR_ANGLE_RMUL:

∀ (a::real) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle x (% a y) = (if a = (0::real) then pi / real_of_nat (2::nat) else if (0::real) ≤ a then vector_angle x y else pi − vector_angle x y)

thm VECTOR_ANGLE_LNEG:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle (vector_neg x) y = pi − vector_angle x y

thm VECTOR_ANGLE_RNEG:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle x (vector_neg y) = pi − vector_angle x y

thm VECTOR_ANGLE_NEG2:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_angle (vector_neg x) (vector_neg y) = vector_angle x y

thm VECTOR_ANGLE:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. dot x y = vector_norm x ∗ (vector_norm y ∗ cos (vector_angle x y))

thm VECTOR_ANGLE_RANGE:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. (0::real) ≤ vector_angle x y ∧ vector_angle x y ≤ pi

thm ORTHOGONAL_VECTOR_ANGLE:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. orthogonal x y = (vector_angle x y = pi / real_of_nat (2::nat))

thm VECTOR_ANGLE_EQ_0:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. (vector_angle x y = (0::real)) = (x ≠ vec (0::nat) ∧ y ≠ vec (0::nat) ∧ % (vector_norm x) y = % (vector_norm y) x)

thm VECTOR_ANGLE_EQ_PI:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. (vector_angle x y = pi) = (x ≠ vec (0::nat) ∧ y ≠ vec (0::nat) ∧ vector_add (% (vector_norm x) y) (% (vector_norm y) x) = vec (0::nat))

thm VECTOR_ANGLE_EQ_0_DIST:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (vector\_angle\ x\ y = (0::real)) = (x \neq vec\ (0::nat) \wedge y \neq vec\ (0::nat) \wedge vector\_norm\ (vector\_add\ x\ y) = vector\_norm\ x + vector\_norm\ y)$

thm VECTOR_ANGLE_EQ_PI_DIST:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (vector\_angle\ x\ y = pi) = (x \neq vec\ (0::nat) \wedge y \neq vec\ (0::nat) \wedge vector\_norm\ (vector\_sub\ x\ y) = vector\_norm\ x + vector\_norm\ y)$

thm SIN_VECTOR_ANGLE_POS:

$\forall (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ (0::real) \leq sin\ (vector\_angle\ v\ w)$

thm SIN_VECTOR_ANGLE_EQ_0:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (sin\ (vector\_angle\ x\ y) = (0::real)) = (vector\_angle\ x\ y = (0::real) \vee vector\_angle\ x\ y = pi)$

thm ASN_SIN_VECTOR_ANGLE:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ asn\ (sin\ (vector\_angle\ x\ y)) = (if\ vector\_angle\ x\ y \leq pi\ /\ real\_of\_nat\ (2::nat)\ then\ vector\_angle\ x\ y\ else\ pi - vector\_angle\ x\ y)$

thm SIN_VECTOR_ANGLE_EQ:

$\forall (x::(real,\ ?'b::type)\ cart)\ (y::(real,\ ?'b::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ (sin\ (vector\_angle\ x\ y) = sin\ (vector\_angle\ w\ z)) = (vector\_angle\ x\ y = vector\_angle\ w\ z \vee vector\_angle\ x\ y = pi - vector\_angle\ w\ z)$

thm CONTINUOUS_AT_CX_VECTOR_ANGLE:

$\forall (c::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ x \neq vec\ (0::nat) \longrightarrow continuous\ (Cx \circ vector\_angle\ c)\ (at\ x)$

thm CONTINUOUS_WITHIN_CX_VECTOR_ANGLE:

$\forall (c::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ x \neq vec\ (0::nat) \longrightarrow continuous\ (Cx \circ vector\_angle\ c)\ (within\ (at\ x)\ s)$

thm REAL_CONTINUOUS_AT_VECTOR_ANGLE:

$\forall (c::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart.\ x \neq vec\ (0::nat) \longrightarrow real\_continuous\ (vector\_angle\ c)\ (at\ x)$

thm REAL_CONTINUOUS_WITHIN_VECTOR_ANGLE:

$\forall (c::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ x \neq vec\ (0::nat) \longrightarrow real\_continuous\ (vector\_angle\ c)\ (within\ (at\ x)\ s)$

thm CONTINUOUS_ON_CX_VECTOR_ANGLE:

$\forall\, s$::(*real*, ?'*a*::*type*) *cart* ⇒ *bool*. ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ *continuous_on* (*Cx* ∘ *vector_angle* (?*c*::(*real*, ?'*a*::*type*) *cart*)) *s*

thm VECTOR_ANGLE_EQ:

∀ (*u*::(*real*, ?'*b*::*type*) *cart*) (*v*::(*real*, ?'*b*::*type*) *cart*) (*x*::(*real*, ?'*a*::*type*) *cart*) *y*::(*real*, ?'*a*::*type*) *cart*. *u* ≠ *vec* (*0*::*nat*) ∧ *v* ≠ *vec* (*0*::*nat*) ∧ *x* ≠ *vec* (*0*::*nat*) ∧ *y* ≠ *vec* (*0*::*nat*) ⟶ (*vector_angle u v* = *vector_angle x y*) = (*dot x y* ∗ (*vector_norm u* ∗ *vector_norm v*) = *dot u v* ∗ (*vector_norm x* ∗ *vector_norm y*))

thm COS_VECTOR_ANGLE_EQ:

∀ (*u*::(*real*, ?'*b*::*type*) *cart*) (*v*::(*real*, ?'*b*::*type*) *cart*) (*x*::(*real*, ?'*a*::*type*) *cart*) *y*::(*real*, ?'*a*::*type*) *cart*. (*cos* (*vector_angle u v*) = *cos* (*vector_angle x y*)) = (*vector_angle u v* = *vector_angle x y*)

thm COLLINEAR_VECTOR_ANGLE:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) *y*::(*real*, ?'*a*::*type*) *cart*. *x* ≠ *vec* (*0*::*nat*) ∧ *y* ≠ *vec* (*0*::*nat*) ⟶ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT x* (*INSERT y EMPTY*))) = (*vector_angle x y* = (*0*::*real*) ∨ *vector_angle x y* = *pi*)

thm COLLINEAR_SIN_VECTOR_ANGLE:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) *y*::(*real*, ?'*a*::*type*) *cart*. *x* ≠ *vec* (*0*::*nat*) ∧ *y* ≠ *vec* (*0*::*nat*) ⟶ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT x* (*INSERT y EMPTY*))) = (*sin* (*vector_angle x y*) = (*0*::*real*))

thm COLLINEAR_SIN_VECTOR_ANGLE_IMP:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) *y*::(*real*, ?'*a*::*type*) *cart*. *sin* (*vector_angle x y*) = (*0*::*real*) ⟶ *x* ≠ *vec* (*0*::*nat*) ∧ *y* ≠ *vec* (*0*::*nat*) ∧ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT x* (*INSERT y EMPTY*)))

thm VECTOR_ANGLE_EQ_0_RIGHT:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) (*y*::(*real*, ?'*a*::*type*) *cart*) *z*::(*real*, ?'*a*::*type*) *cart*. *vector_angle x y* = (*0*::*real*) ⟶ *vector_angle x z* = *vector_angle y z*

thm VECTOR_ANGLE_EQ_0_LEFT:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) (*y*::(*real*, ?'*a*::*type*) *cart*) *z*::(*real*, ?'*a*::*type*) *cart*. *vector_angle x y* = (*0*::*real*) ⟶ *vector_angle z x* = *vector_angle z y*

thm VECTOR_ANGLE_EQ_PI_RIGHT:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) (*y*::(*real*, ?'*a*::*type*) *cart*) *z*::(*real*, ?'*a*::*type*) *cart*. *vector_angle x y* = *pi* ⟶ *vector_angle x z* = *pi* − *vector_angle y z*

thm VECTOR_ANGLE_EQ_PI_LEFT:

∀ (*x*::(*real*, ?'*a*::*type*) *cart*) (*y*::(*real*, ?'*a*::*type*) *cart*) *z*::(*real*, ?'*a*::*type*) *cart*. *vector_angle x y* = *pi* ⟶ *vector_angle z x* = *pi* − *vector_angle z y*

thm COS_VECTOR_ANGLE:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *cos* (*vector_angle x y*) = (*if* *x* = *vec* (*0*::*nat*) $\vee$ *y* = *vec* (*0*::*nat*) *then 0*::*real else dot x y* / (*vector_norm x* $*$ *vector_norm y*))

thm SIN_VECTOR_ANGLE:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *sin* (*vector_angle x y*) = (*if* *x* = *vec* (*0*::*nat*) $\vee$ *y* = *vec* (*0*::*nat*) *then 1*::*real else sqrt* ((*1*::*real*) − (*dot x y* / (*vector_norm x* $*$ *vector_norm y*))^2))

thm Trigonometry2.SIN_POW2_EQ_1_SUB_COS_POW2:

(*sin* (*?x*::*real*))^2 = (*1*::*real*) − (*cos ?x*)^2

thm SIN_SQUARED_VECTOR_ANGLE:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. (*sin* (*vector_angle x y*))^2 = (*if x* = *vec* (*0*::*nat*) $\vee$ *y* = *vec* (*0*::*nat*) *then 1*::*real else* (*1*::*real*) − (*dot x y* / (*vector_norm x* $*$ *vector_norm y*))^2)

thm VECTOR_ANGLE_COMPLEX_LMUL:

$\forall$ *a*::(*real*, *2*) *cart*. *a* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *vector_angle* (*complex_mul a* (*?x*::(*real*, *2*) *cart*)) (*complex_mul a* (*?y*::(*real*, *2*) *cart*)) = *vector_angle ?x ?y*

thm VECTOR_ANGLE_1:

$\forall$ *x*::(*real*, *2*) *cart*. *vector_angle x* (*Cx* (*1*::*real*)) = *acs* (*Re x* / *vector_norm x*)

thm ARG_EQ_VECTOR_ANGLE_1:

$\forall$ *z*::(*real*, *2*) *cart*. *z* $\neq$ *Cx* (*0*::*real*) $\wedge$ (*0*::*real*) $\leq$ *Im z* $\longrightarrow$ *Arg z* = *vector_angle z* (*Cx* (*1*::*real*))

thm VECTOR_ANGLE_ARG:

$\forall$ (*w*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *w* $\neq$ *Cx* (*0*::*real*) $\wedge$ *z* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *vector_angle w z* = (*if* (*0*::*real*) $\leq$ *Im* (*complex_div z w*) *then Arg* (*complex_div z w*) *else real_of_nat* (*2*::*nat*) $*$ *pi* − *Arg* (*complex_div z w*))

thm DEF_angle:

angle = (λ*_1935504*::(*real*, *?'a*::*type*) *cart* $\times$ (*real*, *?'a*::*type*) *cart* $\times$ (*real*, *?'a*::*type*) *cart*. *vector_angle* (*vector_sub* (*fst _1935504*) (*fst* (*snd _1935504*))) (*vector_sub* (*snd* (*snd _1935504*)) (*fst* (*snd _1935504*))))

thm angle:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *angle* (*a*, *b*, *c*) = *vector_angle* (*vector_sub a b*) (*vector_sub c b*)

thm ANGLE_LINEAR_IMAGE_EQ:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'b*::*type*) *cart*) (*b*::(*real*, *?'b*::*type*) *cart*) *c*::(*real*, *?'b*::*type*) *cart*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart. vector_norm (f x) = vector_norm x)* ⟶ *angle (f a, f b, f c) = angle (a, b, c)*

thm ANGLE_TRANSLATION_EQ:

∀ *(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (c::(real, ?'a::type) cart) d::(real, ?'a::type) cart. angle (vector_add a b, vector_add a c, vector_add a d) = angle (b, c, d)*

thm VECTOR_ANGLE_ANGLE:

vector_angle (?x::(real, ?'a::type) cart) (?y::(real, ?'a::type) cart) = angle (?x, vec (0::nat), ?y)

thm ANGLE_EQ_PI_DIST:

∀ *(A::(real, ?'a::type) cart) (B::(real, ?'a::type) cart) C::(real, ?'a::type) cart. (angle (A, B, C) = pi) = (A ≠ B ∧ C ≠ B ∧ distance (A, C) = distance (A, B) + distance (B, C))*

thm SIN_ANGLE_POS:

∀ *(A::(real, ?'a::type) cart) (B::(real, ?'a::type) cart) C::(real, ?'a::type) cart. (0::real) ≤ sin (angle (A, B, C))*

thm ANGLE:

∀ *(A::(real, ?'a::type) cart) (B::(real, ?'a::type) cart) C::(real, ?'a::type) cart. dot (vector_sub A C) (vector_sub B C) = distance (A, C) ∗ (distance (B, C) ∗ cos (angle (A, C, B)))*

thm ANGLE_REFL:

∀ *(A::(real, ?'a::type) cart) B::(real, ?'a::type) cart. angle (A, A, B) = pi / real_of_nat (2::nat) ∧ angle (B, A, A) = pi / real_of_nat (2::nat)*

thm ANGLE_REFL_MID:

∀ *(A::(real, ?'a::type) cart) B::(real, ?'a::type) cart. A ≠ B* ⟶ *angle (A, B, A) = (0::real)*

thm ANGLE_SYM:

∀ *(A::(real, ?'a::type) cart) (B::(real, ?'a::type) cart) C::(real, ?'a::type) cart. angle (A, B, C) = angle (C, B, A)*

thm ANGLE_RANGE:

∀ *(A::(real, ?'a::type) cart) (B::(real, ?'a::type) cart) C::(real, ?'a::type) cart. (0::real) ≤ angle (A, B, C) ∧ angle (A, B, C) ≤ pi*

thm COS_ANGLE_EQ:

∀ *(a::(real, ?'b::type) cart) (b::(real, ?'b::type) cart) (c::(real, ?'b::type) cart) (a'::(real, ?'a::type) cart) (b'::(real, ?'a::type) cart) c'::(real, ?'a::type) cart.*

$(cos\ (angle\ (a,\ b,\ c)) = cos\ (angle\ (a',\ b',\ c'))) = (angle\ (a,\ b,\ c) = angle\ (a',\ b',\ c'))$

thm ANGLE_EQ:

$\forall (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ (c::(real,\ ?'b::type)\ cart)\ (a'::(real,\ ?'a::type)\ cart)\ (b'::(real,\ ?'a::type)\ cart)\ c'::(real,\ ?'a::type)\ cart.\ a \neq b \wedge c \neq b \wedge a' \neq b' \wedge c' \neq b' \longrightarrow (angle\ (a,\ b,\ c) = angle\ (a',\ b',\ c')) = (dot\ (vector\_sub\ a'\ b')\ (vector\_sub\ c'\ b') * (vector\_norm\ (vector\_sub\ a\ b) * vector\_norm\ (vector\_sub\ c\ b)) = dot\ (vector\_sub\ a\ b)\ (vector\_sub\ c\ b) * (vector\_norm\ (vector\_sub\ a'\ b') * vector\_norm\ (vector\_sub\ c'\ b')))$

thm SIN_ANGLE_EQ_0:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ (sin\ (angle\ (A,\ B,\ C)) = (0::real)) = (angle\ (A,\ B,\ C) = (0::real) \vee angle\ (A,\ B,\ C) = pi)$

thm SIN_ANGLE_EQ:

$\forall (A::(real,\ ?'b::type)\ cart)\ (B::(real,\ ?'b::type)\ cart)\ (C::(real,\ ?'b::type)\ cart)\ (A'::(real,\ ?'a::type)\ cart)\ (B'::(real,\ ?'a::type)\ cart)\ C'::(real,\ ?'a::type)\ cart.\ (sin\ (angle\ (A,\ B,\ C)) = sin\ (angle\ (A',\ B',\ C'))) = (angle\ (A,\ B,\ C) = angle\ (A',\ B',\ C') \vee angle\ (A,\ B,\ C) = pi - angle\ (A',\ B',\ C'))$

thm COLLINEAR_ANGLE:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ A \neq B \wedge B \neq C \longrightarrow collinear\ (INSERT\ A\ (INSERT\ B\ (INSERT\ C\ EMPTY))) = (angle\ (A,\ B,\ C) = (0::real) \vee angle\ (A,\ B,\ C) = pi)$

thm COLLINEAR_SIN_ANGLE:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ A \neq B \wedge B \neq C \longrightarrow collinear\ (INSERT\ A\ (INSERT\ B\ (INSERT\ C\ EMPTY))) = (sin\ (angle\ (A,\ B,\ C)) = (0::real))$

thm COLLINEAR_SIN_ANGLE_IMP:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ sin\ (angle\ (A,\ B,\ C)) = (0::real) \longrightarrow A \neq B \wedge B \neq C \wedge collinear\ (INSERT\ A\ (INSERT\ B\ (INSERT\ C\ EMPTY)))$

thm ANGLE_EQ_0_RIGHT:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ angle\ (A,\ B,\ C) = (0::real) \longrightarrow angle\ (A,\ B,\ ?D::(real,\ ?'a::type)\ cart) = angle\ (C,\ B,\ ?D)$

thm ANGLE_EQ_0_LEFT:

$\forall (A::(real,\ ?'a::type)\ cart)\ (B::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart.\ angle\ (A,\ B,\ C) = (0::real) \longrightarrow angle\ (?D::(real,\ ?'a::type)\ cart,\ B,\ A) = angle\ (?D,\ B,\ C)$

thm ANGLE_EQ_PI_RIGHT:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $angle\ (A, B, C) = pi \longrightarrow angle\ (A, B, ?D::(real, ?'a::type)\ cart) = pi - angle\ (C, B, ?D)$

thm ANGLE_EQ_PI_LEFT:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $angle\ (A, B, C) = pi \longrightarrow angle\ (A, B, ?D::(real, ?'a::type)\ cart) = pi - angle\ (C, B, ?D)$

thm COS_ANGLE:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.$ $cos\ (angle\ (a, b, c)) = (if\ a = b \lor c = b\ then\ 0::real\ else\ dot\ (vector\_sub\ a\ b)\ (vector\_sub\ c\ b)\ /\ (vector\_norm\ (vector\_sub\ a\ b) * vector\_norm\ (vector\_sub\ c\ b)))$

thm SIN_ANGLE:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.$ $sin\ (angle\ (a, b, c)) = (if\ a = b \lor c = b\ then\ 1::real\ else\ sqrt\ ((1::real) - (dot\ (vector\_sub\ a\ b)\ (vector\_sub\ c\ b)\ /\ (vector\_norm\ (vector\_sub\ a\ b) * vector\_norm\ (vector\_sub\ c\ b)))^2))$

thm SIN_SQUARED_ANGLE:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.$ $(sin\ (angle\ (a, b, c)))^2 = (if\ a = b \lor c = b\ then\ 1::real\ else\ (1::real) - (dot\ (vector\_sub\ a\ b)\ (vector\_sub\ c\ b)\ /\ (vector\_norm\ (vector\_sub\ a\ b) * vector\_norm\ (vector\_sub\ c\ b)))^2)$

thm LAW_OF_COSINES:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $(distance\ (B, C))^2 = (distance\ (A, B))^2 + (distance\ (A, C))^2 - real\_of\_nat\ (2::nat) * (distance\ (A, B) * (distance\ (A, C) * cos\ (angle\ (B, A, C))))$

thm LAW_OF_SINES:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $sin\ (angle\ (A, B, C)) * distance\ (B, C) = sin\ (angle\ (B, A, C)) * distance\ (A, C)$

thm TRIANGLE_ANGLE_SUM_LEMMA:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $A \neq B \land A \neq C \land B \neq C \longrightarrow cos\ (angle\ (B, A, C) + (angle\ (A, B, C) + angle\ (B, C, A))) = - (1::real)$

thm COS_MINUS1_LEMMA:

$\forall x::real.\ cos\ x = - (1::real) \land (0::real) \leq x \land x < real\_of\_nat\ (3::nat) * pi \longrightarrow x = pi$

thm TRIANGLE_ANGLE_SUM:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $\neg (A = B \wedge B = C \wedge A = C) \longrightarrow angle\ (B, A, C) + (angle\ (A, B, C) + angle\ (B, C, A)) = pi$

thm ANGLE_EQ_PI_OTHERS:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $angle\ (A, B, C) = pi \longrightarrow angle\ (B, C, A) = (0::real) \wedge angle\ (A, C, B) = (0::real) \wedge angle\ (B, A, C) = (0::real) \wedge angle\ (C, A, B) = (0::real)$

thm ANGLE_EQ_0_DIST:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $(angle\ (A, B, C) = (0::real)) = (A \neq B \wedge C \neq B \wedge (distance\ (A, B) = distance\ (A, C) + distance\ (C, B) \vee distance\ (B, C) = distance\ (A, C) + distance\ (A, B)))$

thm ANGLE_EQ_0_DIST_ABS:

$\forall (A::(real, ?'a::type)\ cart)\ (B::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart.$ $(angle\ (A, B, C) = (0::real)) = (A \neq B \wedge C \neq B \wedge distance\ (A, C) = |distance\ (A, B) - distance\ (C, B)|)$

thm CONGRUENT_TRIANGLES_SSS:

$\forall (A::(real, ?'b::type)\ cart)\ (B::(real, ?'b::type)\ cart)\ (C::(real, ?'b::type)\ cart)\ (A'::(real, ?'a::type)\ cart)\ (B'::(real, ?'a::type)\ cart)\ C'::(real, ?'a::type)\ cart.$ $distance\ (A, B) = distance\ (A', B') \wedge distance\ (B, C) = distance\ (B', C') \wedge distance\ (C, A) = distance\ (C', A') \longrightarrow angle\ (A, B, C) = angle\ (A', B', C')$

thm CONGRUENT_TRIANGLES_SAS:

$\forall (A::(real, ?'b::type)\ cart)\ (B::(real, ?'b::type)\ cart)\ (C::(real, ?'b::type)\ cart)\ (A'::(real, ?'a::type)\ cart)\ (B'::(real, ?'a::type)\ cart)\ C'::(real, ?'a::type)\ cart.$ $distance\ (A, B) = distance\ (A', B') \wedge angle\ (A, B, C) = angle\ (A', B', C') \wedge distance\ (B, C) = distance\ (B', C') \longrightarrow distance\ (A, C) = distance\ (A', C')$

thm CONGRUENT_TRIANGLES_AAS:

$\forall (A::(real, ?'b::type)\ cart)\ (B::(real, ?'b::type)\ cart)\ (C::(real, ?'b::type)\ cart)\ (A'::(real, ?'a::type)\ cart)\ (B'::(real, ?'a::type)\ cart)\ C'::(real, ?'a::type)\ cart.$ $angle\ (A, B, C) = angle\ (A', B', C') \wedge angle\ (B, C, A) = angle\ (B', C', A') \wedge distance\ (A, B) = distance\ (A', B') \wedge \neg\ collinear\ (INSERT\ A\ (INSERT\ B\ (INSERT\ C\ EMPTY))) \longrightarrow distance\ (A, C) = distance\ (A', C') \wedge distance\ (B, C) = distance\ (B', C')$

thm CONGRUENT_TRIANGLES_ASA:

$\forall (A::(real, ?'b::type)\ cart)\ (B::(real, ?'b::type)\ cart)\ (C::(real, ?'b::type)\ cart)\ (A'::(real, ?'a::type)\ cart)\ (B'::(real, ?'a::type)\ cart)\ C'::(real, ?'a::type)\ cart.$ $angle\ (A, B, C) = angle\ (A', B', C') \wedge distance\ (A, B) = distance\ (A', B') \wedge$

angle (B, A, C) = *angle* (B', A', C') ∧ ¬ *collinear* (*INSERT A* (*INSERT B* (*INSERT C EMPTY*))) ⟶ *distance* (A, C) = *distance* (A', C')

thm CONGRUENT_TRIANGLES_SSS_FULL:

∀ (*A*::(*real*, *?'b*::*type*) *cart*) (*B*::(*real*, *?'b*::*type*) *cart*) (*C*::(*real*, *?'b*::*type*) *cart*) (*A'*::(*real*, *?'a*::*type*) *cart*) (*B'*::(*real*, *?'a*::*type*) *cart*) *C'*::(*real*, *?'a*::*type*) *cart*. *distance* (A, B) = *distance* (A', B') ∧ *distance* (B, C) = *distance* (B', C') ∧ *distance* (C, A) = *distance* (C', A') ⟶ *distance* (A, B) = *distance* (A', B') ∧ *distance* (B, C) = *distance* (B', C') ∧ *distance* (C, A) = *distance* (C', A') ∧ *angle* (A, B, C) = *angle* (A', B', C') ∧ *angle* (B, C, A) = *angle* (B', C', A') ∧ *angle* (C, A, B) = *angle* (C', A', B')

thm CONGRUENT_TRIANGLES_SAS_FULL:

∀ (*A*::(*real*, *?'b*::*type*) *cart*) (*B*::(*real*, *?'b*::*type*) *cart*) (*C*::(*real*, *?'b*::*type*) *cart*) (*A'*::(*real*, *?'a*::*type*) *cart*) (*B'*::(*real*, *?'a*::*type*) *cart*) *C'*::(*real*, *?'a*::*type*) *cart*. *distance* (A, B) = *distance* (A', B') ∧ *angle* (A, B, C) = *angle* (A', B', C') ∧ *distance* (B, C) = *distance* (B', C') ⟶ *distance* (A, B) = *distance* (A', B') ∧ *distance* (B, C) = *distance* (B', C') ∧ *distance* (C, A) = *distance* (C', A') ∧ *angle* (A, B, C) = *angle* (A', B', C') ∧ *angle* (B, C, A) = *angle* (B', C', A') ∧ *angle* (C, A, B) = *angle* (C', A', B')

thm CONGRUENT_TRIANGLES_AAS_FULL:

∀ (*A*::(*real*, *?'b*::*type*) *cart*) (*B*::(*real*, *?'b*::*type*) *cart*) (*C*::(*real*, *?'b*::*type*) *cart*) (*A'*::(*real*, *?'a*::*type*) *cart*) (*B'*::(*real*, *?'a*::*type*) *cart*) *C'*::(*real*, *?'a*::*type*) *cart*. *angle* (A, B, C) = *angle* (A', B', C') ∧ *angle* (B, C, A) = *angle* (B', C', A') ∧ *distance* (A, B) = *distance* (A', B') ∧ ¬ *collinear* (*INSERT A* (*INSERT B* (*INSERT C EMPTY*))) ⟶ *distance* (A, B) = *distance* (A', B') ∧ *distance* (B, C) = *distance* (B', C') ∧ *distance* (C, A) = *distance* (C', A') ∧ *angle* (A, B, C) = *angle* (A', B', C') ∧ *angle* (B, C, A) = *angle* (B', C', A') ∧ *angle* (C, A, B) = *angle* (C', A', B')

thm CONGRUENT_TRIANGLES_ASA_FULL:

∀ (*A*::(*real*, *?'b*::*type*) *cart*) (*B*::(*real*, *?'b*::*type*) *cart*) (*C*::(*real*, *?'b*::*type*) *cart*) (*A'*::(*real*, *?'a*::*type*) *cart*) (*B'*::(*real*, *?'a*::*type*) *cart*) *C'*::(*real*, *?'a*::*type*) *cart*. *angle* (A, B, C) = *angle* (A', B', C') ∧ *distance* (A, B) = *distance* (A', B') ∧ *angle* (B, A, C) = *angle* (B', A', C') ∧ ¬ *collinear* (*INSERT A* (*INSERT B* (*INSERT C EMPTY*))) ⟶ *distance* (A, B) = *distance* (A', B') ∧ *distance* (B, C) = *distance* (B', C') ∧ *distance* (C, A) = *distance* (C', A') ∧ *angle* (A, B, C) = *angle* (A', B', C') ∧ *angle* (B, C, A) = *angle* (B', C', A') ∧ *angle* (C, A, B) = *angle* (C', A', B')

thm ANGLE_BETWEEN:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. (*angle* (a, x, b) = *pi*) = $(x \neq a \wedge x \neq b \wedge$ *between* $x\ (a, b))$

thm BETWEEN_ANGLE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *between x* (*a*, *b*) = (*x* = *a* $\vee$ *x* = *b* $\vee$ *angle* (*a*, *x*, *b*) = *pi*)

thm ANGLES_ALONG_LINE:

$\forall$ (*A*::(*real*, *?'a*::*type*) *cart*) (*B*::(*real*, *?'a*::*type*) *cart*) (*C*::(*real*, *?'a*::*type*) *cart*) *D*::(*real*, *?'a*::*type*) *cart*. *C* $\neq$ *A* $\wedge$ *C* $\neq$ *B* $\wedge$ *between C* (*A*, *B*) $\longrightarrow$ *angle* (*A*, *C*, *D*) + *angle* (*B*, *C*, *D*) = *pi*

thm ANGLES_ADD_BETWEEN:

$\forall$ (*A*::(*real*, *?'a*::*type*) *cart*) (*B*::(*real*, *?'a*::*type*) *cart*) (*C*::(*real*, *?'a*::*type*) *cart*) *D*::(*real*, *?'a*::*type*) *cart*. *between C* (*A*, *B*) $\wedge$ *D* $\neq$ *A* $\wedge$ *D* $\neq$ *B* $\longrightarrow$ *angle* (*A*, *D*, *C*) + *angle* (*C*, *D*, *B*) = *angle* (*A*, *D*, *B*)

thm DEF_cross:

cross = (λ(*_1939603*::(*real*, *3*) *cart*) *_1939604*::(*real*, *3*) *cart*. *vector* [$ *_1939603* (*2*::*nat*) $*$ $ *_1939604* (*3*::*nat*) $-$ $ *_1939603* (*3*::*nat*) $*$ $ *_1939604* (*2*::*nat*), $ *_1939603* (*3*::*nat*) $*$ $ *_1939604* (*1*::*nat*) $-$ $ *_1939603* (*1*::*nat*) $*$ $ *_1939604* (*3*::*nat*), $ *_1939603* (*1*::*nat*) $*$ $ *_1939604* (*2*::*nat*) $-$ $ *_1939603* (*2*::*nat*) $*$ $ *_1939604* (*1*::*nat*)])

thm cross:

$\forall$ (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *cross a b* = *vector* [$ *a* (*2*::*nat*) $*$ $ *b* (*3*::*nat*) $-$ $ *a* (*3*::*nat*) $*$ $ *b* (*2*::*nat*), $ *a* (*3*::*nat*) $*$ $ *b* (*1*::*nat*) $-$ $ *a* (*1*::*nat*) $*$ $ *b* (*3*::*nat*), $ *a* (*1*::*nat*) $*$ $ *b* (*2*::*nat*) $-$ $ *a* (*2*::*nat*) $*$ $ *b* (*1*::*nat*)]

thm ORTHOGONAL_CROSS:

$\forall$ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *orthogonal* (*cross x y*) *x* $\wedge$ *orthogonal* (*cross x y*) *y* $\wedge$ *orthogonal x* (*cross x y*) $\wedge$ *orthogonal y* (*cross x y*)

thm CROSS_LZERO:

$\forall$ *x*::(*real*, *3*) *cart*. *cross* (*vec* (*0*::*nat*)) *x* = *vec* (*0*::*nat*)

thm CROSS_RZERO:

$\forall$ *x*::(*real*, *3*) *cart*. *cross x* (*vec* (*0*::*nat*)) = *vec* (*0*::*nat*)

thm CROSS_SKEW:

$\forall$ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *cross x y* = *vector_neg* (*cross y x*)

thm CROSS_REFL:

$\forall$ *x*::(*real*, *3*) *cart*. *cross x x* = *vec* (*0*::*nat*)

thm CROSS_LADD:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *cross* (*vector_add x y*) *z* = *vector_add* (*cross x z*) (*cross y z*)

thm CROSS_RADD:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ cross\ x\ (vector\_add\ y\ z) = vector\_add\ (cross\ x\ y)\ (cross\ x\ z)$

thm CROSS_LMUL:

$\forall (c::real)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ cross\ (\%\ c\ x)\ y = \%\ c\ (cross\ x\ y)$

thm CROSS_RMUL:

$\forall (c::real)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ cross\ x\ (\%\ c\ y) = \%\ c\ (cross\ x\ y)$

thm CROSS_LNEG:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ cross\ (vector\_neg\ x)\ y = vector\_neg\ (cross\ x\ y)$

thm CROSS_RNEG:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ cross\ x\ (vector\_neg\ y) = vector\_neg\ (cross\ x\ y)$

thm CROSS_JACOBI:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ vector\_add\ (cross\ x\ (cross\ y\ z))\ (vector\_add\ (cross\ y\ (cross\ z\ x))\ (cross\ z\ (cross\ x\ y))) = vec\ (0::nat)$

thm CROSS_LAGRANGE:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ cross\ x\ (cross\ y\ z) = vector\_sub\ (\%\ (dot\ x\ z)\ y)\ (\%\ (dot\ x\ y)\ z)$

thm CROSS_TRIPLE:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ dot\ (cross\ x\ y)\ z = dot\ (cross\ y\ z)\ x$

thm DOT_CROSS_SELF:

$(\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ dot\ x\ (cross\ x\ y) = (0::real)) \land (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ dot\ x\ (cross\ y\ x) = (0::real)) \land (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ dot\ (cross\ x\ y)\ y = (0::real)) \land (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ dot\ (cross\ y\ x)\ y = (0::real))$

thm CROSS_COMPONENTS:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ \$\ (cross\ x\ y)\ (1::nat) = \$\ x\ (2::nat) * \$\ y\ (3::nat) - \$\ y\ (2::nat) * \$\ x\ (3::nat) \land \$\ (cross\ x\ y)\ (2::nat) = \$\ x\ (3::nat) * \$\ y\ (1::nat) - \$\ y\ (3::nat) * \$\ x\ (1::nat) \land \$\ (cross\ x\ y)\ (3::nat) = \$\ x\ (1::nat) * \$\ y\ (2::nat) - \$\ y\ (1::nat) * \$\ x\ (2::nat)$

thm Trigonometry1.BASIS_3_conjunct1:

$\$\ (basis\ (1::nat))\ (2::nat) = (0::real)$

thm Trigonometry1.BASIS_3_conjunct5:

$ (*basis* (*2*::*nat*)) (*3*::*nat*) = (*0*::*real*)

thm Trigonometry1.BASIS_3_conjunct2:

$ (*basis* (*1*::*nat*)) (*3*::*nat*) = (*0*::*real*)

thm Trigonometry1.BASIS_3_conjunct4:

$ (*basis* (*2*::*nat*)) (*2*::*nat*) = (*1*::*real*)

thm Trigonometry1.BASIS_3_conjunct3:

$ (*basis* (*2*::*nat*)) (*1*::*nat*) = (*0*::*real*)

thm Trigonometry1.BASIS_3_conjunct0:

$ (*basis* (*1*::*nat*)) (*1*::*nat*) = (*1*::*real*)

thm Trigonometry1.BASIS_3_conjunct8:

$ (*basis* (*3*::*nat*)) (*3*::*nat*) = (*1*::*real*)

thm Trigonometry1.BASIS_3_conjunct7:

$ (*basis* (*3*::*nat*)) (*2*::*nat*) = (*0*::*real*)

thm Trigonometry1.BASIS_3_conjunct6:

$ (*basis* (*3*::*nat*)) (*1*::*nat*) = (*0*::*real*)

thm CROSS_BASIS:

cross (*basis* (*1*::*nat*)) (*basis* (*2*::*nat*)) = *basis* (*3*::*nat*) ∧ *cross* (*basis* (*2*::*nat*)) (*basis* (*1*::*nat*)) = *vector_neg* (*basis* (*3*::*nat*)) ∧ *cross* (*basis* (*2*::*nat*)) (*basis* (*3*::*nat*)) = *basis* (*1*::*nat*) ∧ *cross* (*basis* (*3*::*nat*)) (*basis* (*2*::*nat*)) = *vector_neg* (*basis* (*1*::*nat*)) ∧ *cross* (*basis* (*3*::*nat*)) (*basis* (*1*::*nat*)) = *basis* (*2*::*nat*) ∧ *cross* (*basis* (*1*::*nat*)) (*basis* (*3*::*nat*)) = *vector_neg* (*basis* (*2*::*nat*))

thm CROSS_BASIS_NONZERO:

∀ *u*::(*real*, *3*) *cart*. *u* ≠ *vec* (*0*::*nat*) ⟶ *cross u* (*basis* (*1*::*nat*)) ≠ *vec* (*0*::*nat*) ∨ *cross u* (*basis* (*2*::*nat*)) ≠ *vec* (*0*::*nat*) ∨ *cross u* (*basis* (*3*::*nat*)) ≠ *vec* (*0*::*nat*)

thm CROSS_DOT_CANCEL:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *dot x y* = *dot x z* ∧ *cross x y* = *cross x z* ∧ *x* ≠ *vec* (*0*::*nat*) ⟶ *y* = *z*

thm NORM_CROSS_DOT:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. (*vector_norm* (*cross x y*))^2 + (*dot x y*)^2 = (*vector_norm x* ∗ *vector_norm y*)^2

thm DOT_CROSS_DET:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ dot\ x\ (cross\ y\ z) = det\ (vector\ [x,\ y,\ z])$

thm CROSS_CROSS_DET:

$\forall (w::(real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ cross\ (cross\ w\ x)\ (cross\ y\ z) = vector\_sub\ (\%\ (det\ (vector\ [w,\ x,\ z]))\ y)\ (\%\ (det\ (vector\ [w,\ x,\ y]))\ z)$

thm DOT_CROSS:

$\forall (w::(real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ dot\ (cross\ w\ x)\ (cross\ y\ z) = dot\ w\ y * dot\ x\ z - dot\ w\ z * dot\ x\ y$

thm CROSS_EQ_0:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (cross\ x\ y = vec\ (0::nat)) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY)))$

thm CROSS_0:

$(\forall x::(real,\ 3)\ cart.\ cross\ (vec\ (0::nat))\ x = vec\ (0::nat)) \wedge (\forall x::(real,\ 3)\ cart.\ cross\ x\ (vec\ (0::nat)) = vec\ (0::nat))$

thm CROSS_EQ_SELF:

$(\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (cross\ x\ y = x) = (x = vec\ (0::nat))) \wedge (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (cross\ x\ y = y) = (y = vec\ (0::nat)))$

thm NORM_AND_CROSS_EQ_0:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (dot\ x\ y = (0::real) \wedge cross\ x\ y = vec\ (0::nat)) = (x = vec\ (0::nat) \vee y = vec\ (0::nat))$

thm CROSS_MATRIX_MUL:

$\forall (A::((real,\ 3)\ cart,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ matrix\_vector\_mul\ (HOL\_Light\_Import.transp\ A)\ (cross\ (matrix\_vector\_mul\ A\ x)\ (matrix\_vector\_mul\ A\ y)) = \%\ (det\ A)\ (cross\ x\ y)$

thm CROSS_ORTHOGONAL_MATRIX:

$\forall (A::((real,\ 3)\ cart,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ orthogonal\_matrix\ A \longrightarrow cross\ (matrix\_vector\_mul\ A\ x)\ (matrix\_vector\_mul\ A\ y) = \%\ (det\ A)\ (matrix\_vector\_mul\ A\ (cross\ x\ y))$

thm CROSS_ROTATION_MATRIX:

$\forall (A::((real,\ 3)\ cart,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ rotation\_matrix\ A \longrightarrow cross\ (matrix\_vector\_mul\ A\ x)\ (matrix\_vector\_mul\ A\ y) = matrix\_vector\_mul\ A\ (cross\ x\ y)$

thm CROSS_ROTOINVERSION_MATRIX:

$\forall (A::((real,\ 3)\ cart,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ rotoinversion\_matrix\ A \longrightarrow cross\ (matrix\_vector\_mul\ A\ x)\ (matrix\_vector\_mul\ A\ y) = vector\_neg\ (matrix\_vector\_mul\ A\ (cross\ x\ y))$

thm CROSS_ORTHOGONAL_TRANSFORMATION:

$\forall (f::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ orthogonal\_transformation\ f \longrightarrow cross\ (f\ x)\ (f\ y) = \%\ (det\ (matrix\ f))\ (f\ (cross\ x\ y))$

thm CROSS_LINEAR_IMAGE:

$\forall (f::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ linear\ f \wedge (\forall x::(real,\ 3)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \wedge det\ (matrix\ f) = (1::real) \longrightarrow cross\ (f\ x)\ (f\ y) = f\ (cross\ x\ y)$

thm CROSS_LINEAR_IMAGE_WEAK:

$\forall (f::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ linear\ f \wedge (\forall x::(real,\ 3)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \wedge ((2::nat) \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ (matrix\ f) = (1::real)) \longrightarrow cross\ (f\ x)\ (f\ y) = f\ (cross\ x\ y)$

thm NORM_CROSS:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ vector\_norm\ (cross\ x\ y) = vector\_norm\ x * (vector\_norm\ y * sin\ (vector\_angle\ x\ y))$

thm COPLANAR_INSERT_0_NEG:

$coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_neg\ (?x::(real,\ ?'a::type)\ cart))\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool))) = coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?x\ ?s))$

thm COPLANAR_IMP_NEGLIGIBLE:

$\forall s::(real,\ 3)\ cart \Rightarrow bool.\ coplanar\ s \longrightarrow negligible\ s$

thm NOT_COPLANAR_0_4_IMP_INDEPENDENT:

$\forall (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow independent\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))$

thm NOT_COPLANAR_NOT_COLLINEAR:

$\forall (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ (v3::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ w\ EMPTY)))) \longrightarrow \neg\ collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))$

thm SUBSET_AFFINE_HULL_SPECIAL_SCALE:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ a \ne (0::real) \longrightarrow SUBSET\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (\%\ a\ x)\ s))\ (hull\ affine\ t) = SUBSET\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ s))\ (hull\ affine\ t)$

thm COLLINEAR_SPECIAL_SCALE:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \longrightarrow collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (\%\ a\ x)\ (INSERT\ y\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY)))$

thm COLLINEAR_SCALE_ALL:

$\forall (a::real)\ (b::real)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \wedge b \neq (0::real) \longrightarrow collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (\%\ a\ v)\ (INSERT\ (\%\ b\ w)\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm COPLANAR_SPECIAL_SCALE:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \longrightarrow coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (\%\ a\ x)\ (INSERT\ y\ (INSERT\ z\ EMPTY)))) = coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))))$

thm COPLANAR_SCALE_ALL:

$\forall (a::real)\ (b::real)\ (c::real)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \wedge b \neq (0::real) \wedge c \neq (0::real) \longrightarrow coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (\%\ a\ x)\ (INSERT\ (\%\ b\ y)\ (INSERT\ (\%\ c\ z)\ EMPTY)))) = coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))))$

thm DROPOUT_BASIS_3:

$dropout\ (3::nat)\ (basis\ (1::nat)) = basis\ (1::nat) \wedge dropout\ (3::nat)\ (basis\ (2::nat)) = basis\ (2::nat) \wedge dropout\ (3::nat)\ (basis\ (3::nat)) = vec\ (0::nat)$

thm COLLINEAR_BASIS_3:

$collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (basis\ (3::nat))\ (INSERT\ (?x::(real,\ 3)\ cart)\ EMPTY))) = (dropout\ (3::nat)\ ?x = vec\ (0::nat))$

thm OPEN_DROPOUT_3:

$\forall P::(real,\ 2)\ cart \Rightarrow bool.\ HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%2576::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2576\ (P\ x)\ x)) \longrightarrow HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%2577::(real,\ 3)\ cart.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2577\ (P\ (dropout\ (3::nat)\ x))\ x))$

thm SLICE_DROPOUT_3:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ t::real.\ slice\ (3::nat)\ t\ (GSPEC\ (\lambda GEN\%PVAR\%2578::(real,\ 3)\ cart.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2578\ (P\ (dropout\ (3::nat)\ x))\ x)) = GSPEC\ (\lambda GEN\%PVAR\%2579::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2579\ (P\ x)\ x)$

thm NOT_COPLANAR_IMP_NOT_COLLINEAR_DROPOUT_3:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (basis\ (3::nat))\ (INSERT\ x\ (INSERT\ y\ EMPTY)))) \longrightarrow \neg\ collinear\ (INSERT$

(*vec* (*0*::*nat*)) (*INSERT* (*dropout* (*3*::*nat*) *x*) (*INSERT* (*dropout* (*3*::*nat*) *y*) *EMPTY*)))

thm SLICE_312:

∀ *s*::(*real*, *3*) *cart* ⇒ *bool*. *slice* (*1*::*nat*) (*?t*::*real*) *s* = *GSPEC* (λ*GEN%PVAR%2580*::(*real*, *2*) *cart*. ∃ *y*::(*real*, *2*) *cart*. *SETSPEC* *GEN%PVAR%2580* (*IN* (*vector* [*?t*, $ *y* (*1*::*nat*), $ *y* (*2*::*nat*)]) *s*) *y*)

thm SLICE_123:

∀ *s*::(*real*, *3*) *cart* ⇒ *bool*. *slice* (*3*::*nat*) (*?t*::*real*) *s* = *GSPEC* (λ*GEN%PVAR%2581*::(*real*, *2*) *cart*. ∃ *y*::(*real*, *2*) *cart*. *SETSPEC* *GEN%PVAR%2581* (*IN* (*vector* [$ *y* (*1*::*nat*), $ *y* (*2*::*nat*), *?t*]) *s*) *y*)

thm DEF_pad2d3d:

pad2d3d = (λ*_1940432*::(*real*, *2*) *cart*. *lambda* (λ*i*::*nat*. *if* *i* < (*3*::*nat*) *then* $ *_1940432* *i* *else* (*0*::*real*)))

thm pad2d3d:

∀ *x*::(*real*, *2*) *cart*. *pad2d3d* *x* = *lambda* (λ*i*::*nat*. *if* *i* < (*3*::*nat*) *then* $ *x* *i* *else* (*0*::*real*))

thm FORALL_PAD2D3D_THM:

∀ *P*::(*real*, *3*) *cart* ⇒ *bool*. (∀ *y*::(*real*, *3*) *cart*. $ *y* (*3*::*nat*) = (*0*::*real*) ⟶ *P* *y*) = (∀ *x*::(*real*, *2*) *cart*. *P* (*pad2d3d* *x*))

thm QUANTIFY_PAD2D3D_THM:

(∀ *P*::(*real*, *3*) *cart* ⇒ *bool*. (∀ *y*::(*real*, *3*) *cart*. $ *y* (*3*::*nat*) = (*0*::*real*) ⟶ *P* *y*) = (∀ *x*::(*real*, *2*) *cart*. *P* (*pad2d3d* *x*))) ∧ (∀ *P*::(*real*, *3*) *cart* ⇒ *bool*. (∃ *y*::(*real*, *3*) *cart*. $ *y* (*3*::*nat*) = (*0*::*real*) ∧ *P* *y*) = (∃ *x*::(*real*, *2*) *cart*. *P* (*pad2d3d* *x*)))

thm LINEAR_PAD2D3D:

linear *pad2d3d*

thm INJECTIVE_PAD2D3D:

∀ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *pad2d3d* *x* = *pad2d3d* *y* ⟶ *x* = *y*

thm NORM_PAD2D3D:

∀ *x*::(*real*, *2*) *cart*. *vector_norm* (*pad2d3d* *x*) = *vector_norm* *x*

thm Counting_spheres.pad2d3d_dot_v:

∀ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *dot* (*pad2d3d* *x*) (*pad2d3d* *y*) = *dot* *x* *y*

thm Counting_spheres.pad2d3d_SUB:

∀ (*x*::(*real*, *2*) *cart*) *y*::(*real*, *2*) *cart*. *vector_sub* (*pad2d3d* *x*) (*pad2d3d* *y*) = *pad2d3d* (*vector_sub* *x* *y*)

thm DEF_plane:

plane = (λ*_1940614*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT u* (*INSERT v* (*INSERT w EMPTY*))) ∧ *_1940614* = *hull affine* (*INSERT u* (*INSERT v* (*INSERT w EMPTY*))))

thm plane:

∀ *x*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *plane x* = (∃ (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT u* (*INSERT v* (*INSERT w EMPTY*))) ∧ *x* = *hull affine* (*INSERT u* (*INSERT v* (*INSERT w EMPTY*))))

thm PLANE_TRANSLATION_EQ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *plane* (*IMAGE* (*vector_add a*) *s*) = *plane s*

thm PLANE_TRANSLATION:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *plane s* ⟶ *plane* (*IMAGE* (*vector_add a*) *s*)

thm PLANE_LINEAR_IMAGE_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *plane* (*IMAGE f p*) = *plane p*

thm PLANE_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *linear f* ∧ *plane p* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *plane* (*IMAGE f p*)

thm AFFINE_PLANE:

∀ *p*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *plane p* ⟶ *affine p*

thm ROTATION_PLANE_HORIZONTAL:

∀ *s*::(*real*, *3*) *cart* ⇒ *bool*. *plane s* ⟶ (∃ (*a*::(*real*, *3*) *cart*) *f*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. *orthogonal_transformation f* ∧ *det* (*matrix f*) = (*1*::*real*) ∧ *IMAGE f* (*IMAGE* (*vector_add a*) *s*) = *GSPEC* (λ*GEN%PVAR%2587*::(*real*, *3*) *cart*. ∃ *z*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2587* ($ *z* (*3*::*nat*) = (*0*::*real*)) *z*))

thm ROTATION_HORIZONTAL_PLANE:

∀ *p*::(*real*, *3*) *cart* ⇒ *bool*. *plane p* ⟶ (∃ (*a*::(*real*, *3*) *cart*) *f*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. *orthogonal_transformation f* ∧ *det* (*matrix f*) = (*1*::*real*) ∧ *IMAGE* (*vector_add a*) (*IMAGE f* (*GSPEC* (λ*GEN%PVAR%2588*::(*real*, *3*) *cart*. ∃ *z*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2588* ($ *z* (*3*::*nat*) = (*0*::*real*)) *z*))) = *p*)

thm COPLANAR:

$(2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ coplanar\ s = (\exists x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ plane\ x \wedge SUBSET\ s\ x))$

thm COPLANAR_DET_EQ_0:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) = (det\ (vector\ [vector\_sub\ v1\ v0,\ vector\_sub\ v2\ v0,\ vector\_sub\ v3\ v0]) = (0::real))$

thm DEF_lin_combo:

$lin\_combo = (\lambda(\_1942651::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1942652::(real,\ ?'a::type)\ cart \Rightarrow real.\ vsum\ \_1942651\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (\_1942652\ v)\ v))$

thm lin_combo:

$\forall (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::(real,\ ?'a::type)\ cart \Rightarrow real.\ lin\_combo\ V\ f = vsum\ V\ (\lambda v::(real,\ ?'a::type)\ cart.\ \%\ (f\ v)\ v)$

thm DEF_affsign:

$affsign = (\lambda(\_1942663::real \Rightarrow bool)\ (\_1942664::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (\_1942665::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1942666::(real,\ ?'a::type)\ cart.\ \exists f::(real,\ ?'a::type)\ cart \Rightarrow real.\ \_1942666 = lin\_combo\ (HOL\_Light\_Import.UNION\ \_1942664\ \_1942665)\ f \wedge (\forall w::(real,\ ?'a::type)\ cart.\ \_1942665\ w \longrightarrow \_1942663\ (f\ w)) \wedge sum\ (HOL\_Light\_Import.UNION\ \_1942664\ \_1942665)\ f = (1::real))$

thm affsign:

$\forall (v::(real,\ ?'a::type)\ cart)\ (sgn::real \Rightarrow bool)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affsign\ sgn\ s\ t\ v = (\exists f::(real,\ ?'a::type)\ cart \Rightarrow real.\ v = lin\_combo\ (HOL\_Light\_Import.UNION\ s\ t)\ f \wedge (\forall w::(real,\ ?'a::type)\ cart.\ t\ w \longrightarrow sgn\ (f\ w)) \wedge sum\ (HOL\_Light\_Import.UNION\ s\ t)\ f = (1::real))$

thm sgn_gt:

$sgn\_gt = op < (0::real)$

thm sgn_ge:

$sgn\_ge = op \leq (0::real)$

thm sgn_lt:

$sgn\_lt = (\lambda t::real.\ t < (0::real))$

thm sgn_le:

$sgn\_le = (\lambda t::real.\ t \leq (0::real))$

thm aff_gt_def:

$aff\_gt = affsign\ sgn\_gt$

thm aff_ge_def:

aff_ge = affsign sgn_ge

thm aff_lt_def:

aff_lt = affsign sgn_lt

thm aff_le_def:

aff_le = affsign sgn_le

thm AFFSIGN:

affsign (?sgn::real ⇒ bool) (?s::(real, ?'a::type) cart ⇒ bool) (?t::(real, ?'a::type) cart ⇒ bool) = GSPEC (λGEN%PVAR%2593::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2593 (∃ f::(real, ?'a::type) cart ⇒ real. y = vsum (HOL_Light_Import.UNION ?s ?t) (λv::(real, ?'a::type) cart. % (f v) v) ∧ (∀ w::(real, ?'a::type) cart. IN w ?t ⟶ ?sgn (f w)) ∧ sum (HOL_Light_Import.UNION ?s ?t) f = (1::real)) y)

thm AFFSIGN_ALT:

affsign (?sgn::real ⇒ bool) (?s::(real, ?'a::type) cart ⇒ bool) (?t::(real, ?'a::type) cart ⇒ bool) = GSPEC (λGEN%PVAR%2594::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2594 (∃ f::(real, ?'a::type) cart ⇒ real. (∀ w::(real, ?'a::type) cart. IN w (HOL_Light_Import.UNION ?s ?t) ⟶ IN w ?t ⟶ ?sgn (f w)) ∧ sum (HOL_Light_Import.UNION ?s ?t) f = (1::real) ∧ vsum (HOL_Light_Import.UNION ?s ?t) (λv::(real, ?'a::type) cart. % (f v) v) = y) y)

thm IN_AFFSIGN:

IN (?y::(real, ?'a::type) cart) (affsign (?sgn::real ⇒ bool) (?s::(real, ?'a::type) cart ⇒ bool) (?t::(real, ?'a::type) cart ⇒ bool)) = (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x ?t ⟶ ?sgn (u x)) ∧ sum (HOL_Light_Import.UNION ?s ?t) u = (1::real) ∧ vsum (HOL_Light_Import.UNION ?s ?t) (λx::(real, ?'a::type) cart. % (u x) x) = ?y)

thm AFFSIGN_DISJOINT_DIFF:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. affsign (?sgn::real ⇒ bool) s t = affsign ?sgn (DIFF s t) t

thm AFF_GE_DISJOINT_DIFF:

∀ (s::(real, ?'a::type) cart ⇒ bool) t::(real, ?'a::type) cart ⇒ bool. aff_ge s t = aff_ge (DIFF s t) t

thm AFFSIGN_INJECTIVE_LINEAR_IMAGE:

∀ (f::(real, ?'c::type) cart ⇒ (real, ?'b::type) cart) (sgn::real ⇒ bool) (s::(real, ?'c::type) cart ⇒ bool) (t::(real, ?'c::type) cart ⇒ bool) v::?'a::type. linear f ∧ (∀ (x::(real, ?'c::type) cart) y::(real, ?'c::type) cart. f x = f y ⟶ x = y) ⟶ affsign sgn (IMAGE f s) (IMAGE f t) = IMAGE f (affsign sgn s t)

thm AFF_GE_INJECTIVE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *aff_ge* (*IMAGE f s*) (*IMAGE f t*) = *IMAGE f* (*aff_ge s t*)

thm AFF_GT_INJECTIVE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *aff_gt* (*IMAGE f s*) (*IMAGE f t*) = *IMAGE f* (*aff_gt s t*)

thm AFF_LE_INJECTIVE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *aff_le* (*IMAGE f s*) (*IMAGE f t*) = *IMAGE f* (*aff_le s t*)

thm AFF_LT_INJECTIVE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *aff_lt* (*IMAGE f s*) (*IMAGE f t*) = *IMAGE f* (*aff_lt s t*)

thm IN_AFFSIGN_TRANSLATION:

$\forall$ (*sgn*::*real* $\Rightarrow$ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *affsign sgn s t v* $\longrightarrow$ *affsign sgn* (*IMAGE* (*vector_add a*) *s*) (*IMAGE* (*vector_add a*) *t*) (*vector_add a v*)

thm AFFSIGN_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*sgn*::*real* $\Rightarrow$ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *affsign sgn* (*IMAGE* (*vector_add a*) *s*) (*IMAGE* (*vector_add a*) *t*) = *IMAGE* (*vector_add a*) (*affsign sgn s t*)

thm AFF_GE_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *aff_ge* (*IMAGE* (*vector_add a*) *s*) (*IMAGE* (*vector_add a*) *t*) = *IMAGE* (*vector_add a*) (*aff_ge s t*)

thm AFF_GT_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *aff_gt* (*IMAGE* (*vector_add a*) *s*) (*IMAGE* (*vector_add a*) *t*) = *IMAGE* (*vector_add a*) (*aff_gt s t*)

thm AFF_LE_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ aff\_le\ (IMAGE\ (vector\_add\ a)\ s)\ (IMAGE\ (vector\_add\ a)\ t) = IMAGE\ (vector\_add\ a)\ (aff\_le\ s\ t)$

thm AFF_LT_TRANSLATION:

$\forall (a::(real,\ ?'a::type)\ cart)\ (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ aff\_lt\ (IMAGE\ (vector\_add\ a)\ s)\ (IMAGE\ (vector\_add\ a)\ t) = IMAGE\ (vector\_add\ a)\ (aff\_lt\ s\ t)$

thm AFF_GE_1_1:

$\forall (x::(real,\ ?'b::type)\ cart)\ (v::(real,\ ?'b::type)\ cart)\ w::?'a::type.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2602::(real,\ ?'b::type)\ cart.\ \exists y::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2602\ (\exists (t1::real)\ t2::real.\ (0::real) \leq t2 \wedge t1 + t2 = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (\%\ t2\ v))\ y)$

thm AFF_GE_1_2:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)) \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%2603::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2603\ (\exists (t1::real)\ (t2::real)\ t3::real.\ (0::real) \leq t2 \wedge (0::real) \leq t3 \wedge t1 + (t2 + t3) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ w)))\ y)$

thm AFF_GE_2_1:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) \longrightarrow aff\_ge\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2604::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2604\ (\exists (t1::real)\ (t2::real)\ t3::real.\ (0::real) \leq t3 \wedge t1 + (t2 + t3) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ w)))\ y)$

thm AFF_GT_1_1:

$\forall (x::(real,\ ?'b::type)\ cart)\ (v::(real,\ ?'b::type)\ cart)\ w::?'a::type.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2605::(real,\ ?'b::type)\ cart.\ \exists y::(real,\ ?'b::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2605\ (\exists (t1::real)\ t2::real.\ (0::real) < t2 \wedge t1 + t2 = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (\%\ t2\ v))\ y)$

thm AFF_GT_1_2:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)) = GSPEC$

$(\lambda GEN\%PVAR\%2606::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2606\ (\exists (t1::real)\ (t2::real)\ t3::real.\ (0::real) < t2 \wedge (0::real) < t3 \wedge t1 + (t2 + t3) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ w)))\ y)$

thm AFF_GT_2_1:

$\forall (x::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) \longrightarrow aff\_gt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2607::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2607\ (\exists (t1::real)\ (t2::real)\ t3::real.\ (0::real) < t3 \wedge t1 + (t2 + t3) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ w)))\ y)$

thm AFF_GT_3_1:

$\forall (v::(real, ?'a::type)\ cart)\ (w::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ DISJOINT\ (INSERT\ v\ (INSERT\ w\ (INSERT\ x\ EMPTY)))\ (INSERT\ y\ EMPTY) \longrightarrow aff\_gt\ (INSERT\ v\ (INSERT\ w\ (INSERT\ x\ EMPTY)))\ (INSERT\ y\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2608::(real, ?'a::type)\ cart.\ \exists z::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2608\ (\exists (t1::real)\ (t2::real)\ (t3::real)\ t4::real.\ (0::real) < t4 \wedge t1 + (t2 + (t3 + t4)) = (1::real) \wedge z = vector\_add\ (\%\ t1\ v)\ (vector\_add\ (\%\ t2\ w)\ (vector\_add\ (\%\ t3\ x)\ (\%\ t4\ y))))\ z)$

thm AFF_LT_1_1:

$\forall (x::(real, ?'a::type)\ cart)\ v::(real, ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) \longrightarrow aff\_lt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2609::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2609\ (\exists (t1::real)\ t2::real.\ t2 < (0::real) \wedge t1 + t2 = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (\%\ t2\ v))\ y)$

thm AFF_LT_2_1:

$\forall (x::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) \longrightarrow aff\_lt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2610::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2610\ (\exists (t1::real)\ (t2::real)\ t3::real.\ t3 < (0::real) \wedge t1 + (t2 + t3) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ w)))\ y)$

thm AFF_GE_1_2_0:

$\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ v \neq vec\ (0::nat) \wedge w \neq vec\ (0::nat) \longrightarrow aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%2611::(real, ?'a::type)\ cart.\ \exists (a::real)\ b::real.\ SETSPEC\ GEN\%PVAR\%2611\ ((0::real) \leq a \wedge (0::real) \leq b)\ (vector\_add\ (\%\ a\ v)\ (\%\ b\ w)))$

thm AFF_GE_1_1_0:

$\forall v::(real, ?'a::type)\ cart.\ v \neq vec\ (0::nat) \longrightarrow aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%2612::(real, ?'a::type)\ cart.\ \exists a::real.\ SETSPEC\ GEN\%PVAR\%2612\ ((0::real) \leq a)\ (\%\ a\ v))$

thm CONVEX_AFFSIGN:

$\forall sgn::real \Rightarrow bool.\ (\forall (x::real)\ (y::real)\ u::real.\ sgn\ x \wedge sgn\ y \wedge (0::real) \leq u \wedge u \leq (1::real) \longrightarrow sgn\ (((1::real) - u) * x + u * y)) \longrightarrow (\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (affsign\ sgn\ s\ t))$

thm CONVEX_AFF_GE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (aff\_ge\ s\ t)$

thm CONVEX_AFF_LE:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (aff\_le\ s\ t)$

thm CONVEX_AFF_GT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (aff\_gt\ s\ t)$

thm CONVEX_AFF_LT:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ convex\ (aff\_lt\ s\ t)$

thm AFFSIGN_SUBSET_AFFINE_HULL:

$\forall (sgn::real \Rightarrow bool)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (affsign\ sgn\ s\ t)\ (hull\ affine\ (HOL\_Light\_Import.UNION\ s\ t))$

thm AFF_GE_SUBSET_AFFINE_HULL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (aff\_ge\ s\ t)\ (hull\ affine\ (HOL\_Light\_Import.UNION\ s\ t))$

thm AFF_LE_SUBSET_AFFINE_HULL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (aff\_le\ s\ t)\ (hull\ affine\ (HOL\_Light\_Import.UNION\ s\ t))$

thm AFF_GT_SUBSET_AFFINE_HULL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (aff\_gt\ s\ t)\ (hull\ affine\ (HOL\_Light\_Import.UNION\ s\ t))$

thm AFF_LT_SUBSET_AFFINE_HULL:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (aff\_lt\ s\ t)\ (hull\ affine\ (HOL\_Light\_Import.UNION\ s\ t))$

thm AFFSIGN_EQ_AFFINE_HULL:

∀ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::*?'a*::*type*. *affsign sgn s EMPTY* = *hull affine s*

thm AFF_GE_EQ_AFFINE_HULL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::*?'a*::*type*. *aff_ge s EMPTY* = *hull affine s*

thm AFF_LE_EQ_AFFINE_HULL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::*?'a*::*type*. *aff_le s EMPTY* = *hull affine s*

thm AFF_GT_EQ_AFFINE_HULL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::*?'a*::*type*. *aff_gt s EMPTY* = *hull affine s*

thm AFF_LT_EQ_AFFINE_HULL:

∀ (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *t*::*?'a*::*type*. *aff_lt s EMPTY* = *hull affine s*

thm AFFSIGN_SUBSET_AFFSIGN:

∀ (*sgn1*::*real* ⇒ *bool*) (*sgn2*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (∀ *x*::*real*. *sgn1 x* ⟶ *sgn2 x*) ⟶ *SUBSET* (*affsign sgn1 s t*) (*affsign sgn2 s t*)

thm AFF_GT_SUBSET_AFF_GE:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET* (*aff_gt s t*) (*aff_ge s t*)

thm AFFSIGN_MONO_LEFT:

∀ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s s'* ⟶ *SUBSET* (*affsign sgn s t*) (*affsign sgn s' t*)

thm AFFSIGN_MONO_SHUFFLE:

∀ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import*.*UNION s' t'* = *HOL_Light_Import*.*UNION s t* ∧ *SUBSET t' t* ⟶ *SUBSET* (*affsign sgn s t*) (*affsign sgn s' t'*)

thm AFF_GT_MONO_LEFT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s s'* ⟶ *SUBSET* (*aff_gt s t*) (*aff_gt s' t*)

thm AFF_GE_MONO_LEFT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s s'* ⟶ *SUBSET* (*aff_ge s t*) (*aff_ge s' t*)

thm AFF_LT_MONO_LEFT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s s'* ⟶ *SUBSET* (*aff_lt s t*) (*aff_lt s' t*)

thm AFF_LE_MONO_LEFT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s s'* ⟶ *SUBSET* (*aff_le s t*) (*aff_le s' t*)

thm AFFSIGN_MONO_RIGHT:

$\forall$ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *sgn* (*0*::*real*) ∧ *SUBSET t t'* ∧ *DISJOINT s t'* ⟶ *SUBSET* (*affsign sgn s t*) (*affsign sgn s t'*)

thm AFF_GE_MONO_RIGHT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t t'* ∧ *DISJOINT s t'* ⟶ *SUBSET* (*aff_ge s t*) (*aff_ge s t'*)

thm AFF_LE_MONO_RIGHT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET t t'* ∧ *DISJOINT s t'* ⟶ *SUBSET* (*aff_le s t*) (*aff_le s t'*)

thm AFFINE_HULL_SUBSET_AFFSIGN:

$\forall$ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *sgn* (*0*::*real*) ∧ *DISJOINT s t* ⟶ *SUBSET* (*hull affine s*) (*affsign sgn s t*)

thm AFFINE_HULL_SUBSET_AFF_GE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *DISJOINT s t* ⟶ *SUBSET* (*hull affine s*) (*aff_ge s t*)

thm AFF_GE_AFF_GT_DECOMP:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ *FINITE* (*?t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ∧ *DISJOINT s ?t* ⟶ *aff_ge s ?t* = *HOL_Light_Import.UNION* (*aff_gt s ?t*) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2616*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *a*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2616* (*IN a ?t*) (*aff_ge s* (*DELETE ?t a*)))))

thm AFFSIGN_SPECIAL_SCALE:

$\forall$ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::*real*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* ∧ *FINITE t* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *t* ∧ ¬ *IN v t* ∧ ¬ *IN* (% *a v*) *t* ∧ (∀ *x*::*real*. *sgn x* ⟶ *sgn* (*x* / *real_of_nat* (*2*::*nat*))) ∧ (∀ (*x*::*real*) *y*::*real*. *sgn x* ∧ *sgn y* ⟶ *sgn* (*x* + *y*)) ∧ (*0*::*real*) < *a* ⟶ *affsign sgn* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) *s*)) *t* = *affsign sgn* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*

thm AFF_GE_SPECIAL_SCALE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::*real*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v t* $\wedge$ $\neg$ *IN* (% *a v*) *t* $\wedge$ (*0*::*real*) $<$ *a* $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) *s*)) *t* = *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*

thm AFF_LE_SPECIAL_SCALE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::*real*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v t* $\wedge$ $\neg$ *IN* (% *a v*) *t* $\wedge$ (*0*::*real*) $<$ *a* $\longrightarrow$ *aff_le* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) *s*)) *t* = *aff_le* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*

thm AFF_GT_SPECIAL_SCALE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::*real*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v t* $\wedge$ $\neg$ *IN* (% *a v*) *t* $\wedge$ (*0*::*real*) $<$ *a* $\longrightarrow$ *aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) *s*)) *t* = *aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*

thm AFF_LT_SPECIAL_SCALE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*a*::*real*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v t* $\wedge$ $\neg$ *IN* (% *a v*) *t* $\wedge$ (*0*::*real*) $<$ *a* $\longrightarrow$ *aff_lt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) *s*)) *t* = *aff_lt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*

thm AFFSIGN_0:

$\forall$ (*sgn*::*real* $\Rightarrow$ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*DIFF s t*) $\longrightarrow$ *affsign sgn s t* = *GSPEC* (λ*GEN%PVAR%2617*::(*real*, *?'a*::*type*) *cart*. $\exists$ *f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *SETSPEC GEN%PVAR%2617* ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x t* $\longrightarrow$ *sgn* (*f x*)) (*vsum* (*HOL_Light_Import.UNION s t*) (λ*v*::(*real*, *?'a*::*type*) *cart*. % (*f v*) *v*)))

thm AFF_GE_0_AFFINE_MULTIPLE_CONVEX:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*DIFF s t*) $\wedge$ *t* $\neq$ *EMPTY* $\longrightarrow$ *aff_ge s t* = *GSPEC* (λ*GEN%PVAR%2618*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) (*c*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2618* (*IN x* (*hull affine* (*DIFF s t*)) $\wedge$ *IN y* (*hull convex t*) $\wedge$ (*0*::*real*) $\leq$ *c*) (*vector_add x* (% *c y*)))

thm AFF_GE_0_MULTIPLE_AFFINE_CONVEX:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*DIFF s t*) $\wedge$ *t* $\neq$ *EMPTY* $\longrightarrow$ *aff_ge s t* = *HOL_Light_Import.UNION* (*hull affine* (*DIFF s t*)) (*GSPEC* (λ*GEN%PVAR%2619*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*c*::*real*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2619* (*IN x* (*hull affine* (*DIFF s t*)) $\wedge$ *IN y* (*hull convex t*) $\wedge$ (*0*::*real*) $\leq$ *c*) (% *c* (*vector_add x y*))))

thm AFF_GE_0_AFFINE_CONVEX_CONE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*DIFF s t*) $\longrightarrow$ *aff_ge s t* = *GSPEC* (λ*GEN%PVAR%2622*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2622* (*IN x* (*hull affine* (*DIFF s t*)) $\wedge$ *IN y* (*hull convex_cone t*)) (*vector_add x y*))

thm AFF_GE_0_N:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *s* $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *s* = *GSPEC* (λ*GEN%PVAR%2623*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2623* ($\exists$ *u*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* $\longrightarrow$ (*0*::*real*) $\leq$ *u x*) $\wedge$ *y* = *vsum s* (λ*x*::(*real*, *?'a*::*type*) *cart*. % (*u x*) *x*)) *y*)

thm AFF_GE_0_CONVEX_HULL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *s* $\neq$ *EMPTY* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *s* $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *s* = *GSPEC* (λ*GEN%PVAR%2624*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*t*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2624* ((*0*::*real*) $\leq$ *t* $\wedge$ *IN y* (*hull convex s*)) (% *t y*))

thm AFF_GE_0_CONVEX_HULL_ALT:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *s* $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *s* = *INSERT* (*vec* (*0*::*nat*)) (*GSPEC* (λ*GEN%PVAR%2625*::(*real*, *?'a*::*type*) *cart*. $\exists$ (*t*::*real*) *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2625* ((*0*::*real*) < *t* $\wedge$ *IN y* (*hull convex s*)) (% *t y*)))

thm AFF_GE_0_CONVEX_CONE_NEGATIONS:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*DIFF s t*) $\longrightarrow$ *aff_ge s t* = *hull convex_cone* (*HOL_Light_Import.UNION s* (*HOL_Light_Import.UNION t* (*IMAGE vector_neg* (*DIFF s t*))))

thm CONVEX_HULL_AFF_GE:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *hull convex s* = *aff_ge EMPTY s*

thm POLYHEDRON_AFF_GE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\longrightarrow$ *polyhedron* (*aff_ge s t*)

thm CLOSED_AFF_GE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *FINITE t* $\longrightarrow$ *HOL_Light_Import.closed* (*aff_ge s t*)

thm CONIC_AFF_GE_0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *s* $\longrightarrow$ *conic* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *s*)

thm HALFLINE_REFL:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *aff_ge* (*INSERT x EMPTY*) (*INSERT x EMPTY*) = *INSERT x EMPTY*

thm HALFLINE_EXPLICIT:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*) = *GSPEC* (λ*GEN%PVAR%2626*::(*real*, *?'a*::*type*) *cart*. ∃ *z*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2626* (∃ (*t1*::*real*) *t2*::*real*. (*0*::*real*) ≤ *t2* ∧ *t1* + *t2* = (*1*::*real*) ∧ *z* = *vector_add* (% *t1 x*) (% *t2 y*)) *z*)

thm HALFLINE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*) = *GSPEC* (λ*GEN%PVAR%2627*::(*real*, *?'a*::*type*) *cart*. ∃ *z*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2627* (∃ *t*≥*0*::*real*. *z* = *vector_add* (% ((*1*::*real*) − *t*) *x*) (% *t y*)) *z*)

thm CLOSED_HALFLINE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.closed* (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*))

thm SEGMENT_SUBSET_HALFLINE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*closed_segment* [(*x*, *y*)]) (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*))

thm ENDS_IN_HALFLINE:

(∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *IN x* (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*))) ∧ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN y* (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*)))

thm HALFLINE_SUBSET_AFFINE_HULL:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*)) (*hull affine* (*INSERT x* (*INSERT y EMPTY*)))

thm HALFLINE_INTER_COMPACT_SEGMENT:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *compact s* ∧ *convex s* ∧ *IN a s* ⟶ (∃ *c*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.INTER* (*aff_ge* (*INSERT a EMPTY*) (*INSERT b EMPTY*)) *s* = *closed_segment* [(*a*, *c*)])

thm DEF_conv0:

conv0 = *affsign sgn_gt EMPTY*

thm conv0:

∀ *S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *conv0 S* = *affsign sgn_gt EMPTY S*

thm CONV0_INJECTIVE_LINEAR_IMAGE:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* $\longrightarrow$ *x* = *y*) $\longrightarrow$ *conv0* (*IMAGE f s*) = *IMAGE f* (*conv0 s*)

thm CONV0_TRANSLATION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *conv0* (*IMAGE* (*vector_add a*) *s*) = *IMAGE* (*vector_add a*) (*conv0 s*)

thm CONV0_SUBSET_CONVEX_HULL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET* (*conv0 s*) (*hull convex s*)

thm CONV0_AFF_GT:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *conv0 s* = *aff_gt EMPTY s*

thm CONVEX_HULL_CONV0_DECOMP:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ *hull convex s* = *HOL_Light_Import.UNION* (*conv0 s*) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%2628*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. $\exists$ *a*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2628* (*IN a s*) (*hull convex* (*DELETE s a*)))))

thm CONVEX_CONV0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *convex* (*conv0 s*)

thm BOUNDED_CONV0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ *bounded* (*conv0 s*)

thm MEASURABLE_CONV0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *bounded s* $\longrightarrow$ *measurable* (*conv0 s*)

thm NEGLIGIBLE_CONVEX_HULL_DIFF_CONV0:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *CARD s* $\leq$ *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) $\longrightarrow$ *negligible* (*DIFF* (*hull convex s*) (*conv0 s*))

thm MEASURE_CONV0_CONVEX_HULL:

$\forall$ *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\wedge$ *CARD s* $\leq$ *dimindex HOL_Light_Import.UNIV* + (*1*::*nat*) $\longrightarrow$ *HOL_Light_Import.measure* (*conv0 s*) = *HOL_Light_Import.measure* (*hull convex s*)

thm DEF_orthonormal:

orthonormal = (λ(*_1947132*::(*real*, *3*) *cart*) (*_1947133*::(*real*, *3*) *cart*) *_1947134*::(*real*, *3*) *cart*. *dot _1947132 _1947132* = (*1*::*real*) $\wedge$ *dot _1947133 _1947133* = (*1*::*real*) $\wedge$ *dot _1947134 _1947134* = (*1*::*real*) $\wedge$ *dot _1947132 _1947133* = (*0*::*real*) $\wedge$ *dot _1947132 _1947134* = (*0*::*real*) $\wedge$ *dot _1947133 _1947134* = (*0*::*real*) $\wedge$ (*0*::*real*) < *dot* (*cross _1947132 _1947133*) *_1947134*)

thm orthonormal:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 = (dot\ e1\ e1 = (1::real) \wedge dot\ e2\ e2 = (1::real) \wedge dot\ e3\ e3 = (1::real) \wedge dot\ e1\ e2 = (0::real) \wedge dot\ e1\ e3 = (0::real) \wedge dot\ e2\ e3 = (0::real) \wedge (0::real) < dot\ (cross\ e1\ e2)\ e3)$

thm ORTHONORMAL_LINEAR_IMAGE:

$\forall f::(real, 3)\ cart \Rightarrow (real, 3)\ cart.\ linear\ f \wedge (\forall x::(real, 3)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \wedge ((2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ (matrix\ f) = (1::real)) \longrightarrow (\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ (f\ e1)\ (f\ e2)\ (f\ e3) = orthonormal\ e1\ e2\ e3)$

thm ORTHONORMAL_PERMUTE:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow orthonormal\ e2\ e3\ e1$

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct0:

$dot\ (basis\ (1::nat))\ (basis\ (1::nat)) = (1::real)$

thm ORTHONORMAL_CROSS:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow cross\ e2\ e3 = e1 \wedge cross\ e3\ e1 = e2 \wedge cross\ e1\ e2 = e3$

thm ORTHONORMAL_IMP_NONZERO:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow e1 \neq vec\ (0::nat) \wedge e2 \neq vec\ (0::nat) \wedge e3 \neq vec\ (0::nat)$

thm ORTHONORMAL_IMP_DISTINCT:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow e1 \neq e2 \wedge e1 \neq e3 \wedge e2 \neq e3$

thm ORTHONORMAL_IMP_INDEPENDENT:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow independent\ (INSERT\ e1\ (INSERT\ e2\ (INSERT\ e3\ EMPTY)))$

thm ORTHONORMAL_IMP_SPANNING:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow span\ (INSERT\ e1\ (INSERT\ e2\ (INSERT\ e3\ EMPTY))) = HOL\_Light\_Import.UNIV$

thm ORTHONORMAL_IMP_INDEPENDENT_EXPLICIT_0:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ (e3::(real, 3)\ cart)\ (t1::real)\ (t2::real)\ t3::real.\ orthonormal\ e1\ e2\ e3 \longrightarrow (vector\_add\ (\%\ t1\ e1)\ (vector\_add\ (\%\ t2\ e2)\ (\%\ t3\ e3)) = vec\ (0::nat)) = (t1 = (0::real) \wedge t2 = (0::real) \wedge t3 = (0::real))$

thm ORTHONORMAL_IMP_INDEPENDENT_EXPLICIT:

$\forall (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ (e3::(real, 3)\ cart)\ (s1::real)\ (s2::real)\ (s3::real)\ (t1::real)\ (t2::real)\ t3::real.\ orthonormal\ e1\ e2\ e3 \longrightarrow (vector\_add\ (\%$

$s1\ e1)\ (vector\_add\ (\%\ s2\ e2)\ (\%\ s3\ e3)) = vector\_add\ (\%\ t1\ e1)\ (vector\_add\ (\%\ t2\ e2)\ (\%\ t3\ e3))) = (s1 = t1 \wedge s2 = t2 \wedge s3 = t3)$

thm DEF_arcV:

$arcV = (\lambda(\_1947854::(real,\ ?'a::type)\ cart)\ (\_1947855::(real,\ ?'a::type)\ cart)\ \_1947856::(real,\ ?'a::type)\ cart.\ acs\ (dot\ (vector\_sub\ \_1947855\ \_1947854)\ (vector\_sub\ \_1947856\ \_1947854) / (vector\_norm\ (vector\_sub\ \_1947855\ \_1947854) * vector\_norm\ (vector\_sub\ \_1947856\ \_1947854))))$

thm arcV:

$\forall (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ arcV\ u\ v\ w = acs\ (dot\ (vector\_sub\ v\ u)\ (vector\_sub\ w\ u) / (vector\_norm\ (vector\_sub\ v\ u) * vector\_norm\ (vector\_sub\ w\ u)))$

thm ARCV_ANGLE:

$\forall (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ arcV\ u\ v\ w = angle\ (v,\ u,\ w)$

thm ARCV_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (a::(real,\ ?'b::type)\ cart)\ (b::(real,\ ?'b::type)\ cart)\ c::(real,\ ?'b::type)\ cart.\ linear\ f \wedge (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow arcV\ (f\ a)\ (f\ b)\ (f\ c) = arcV\ a\ b\ c$

thm ARCV_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ arcV\ (vector\_add\ a\ b)\ (vector\_add\ a\ c)\ (vector\_add\ a\ d) = arcV\ b\ c\ d$

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct2:

$dot\ (basis\ (3::nat))\ (basis\ (3::nat)) = (1::real)$

thm AZIM_EXISTS:

$\forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.\ \exists\ theta \geq 0::real.\ theta < real\_of\_nat\ (2::nat) * pi \wedge (\exists (h1::real)\ h2::real.\ \forall (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \wedge \%\ (distance\ (w,\ v))\ e3 = vector\_sub\ w\ v \wedge w \neq v \longrightarrow (\exists (psi::real)\ (r1::real)\ r2::real.\ vector\_sub\ w1\ v = vector\_add\ (\%\ (r1 * cos\ psi)\ e1)\ (vector\_add\ (\%\ (r1 * sin\ psi)\ e2)\ (\%\ h1\ (vector\_sub\ w\ v))) \wedge vector\_sub\ w2\ v = vector\_add\ (\%\ (r2 * cos\ (psi + theta))\ e1)\ (vector\_add\ (\%\ (r2 * sin\ (psi + theta))\ e2)\ (\%\ h2\ (vector\_sub\ w\ v))) \wedge (\neg\ collinear\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w1\ EMPTY))) \longrightarrow (0::real) < r1) \wedge (\neg\ collinear\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w2\ EMPTY))) \longrightarrow (0::real) < r2)))$

thm azim_spec:

$\exists\ theta::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \Rightarrow real.\ \forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)$

cart. (0::real) ≤ theta v w w1 w2 ∧ theta v w w1 w2 < real_of_nat (2::nat) ∗ pi ∧ (∃(h1::real) h2::real. ∀(e1::(real, 3) cart) (e2::(real, 3) cart) e3::(real, 3) cart. orthonormal e1 e2 e3 ∧ % (distance (w, v)) e3 = vector_sub w v ∧ w ≠ v ⟶ (∃(psi::real) (r1::real) r2::real. vector_sub w1 v = vector_add (% (r1 ∗ cos psi) e1) (vector_add (% (r1 ∗ sin psi) e2) (% h1 (vector_sub w v))) ∧ vector_sub w2 v = vector_add (% (r2 ∗ cos (psi + theta v w w1 w2)) e1) (vector_add (% (r2 ∗ sin (psi + theta v w w1 w2)) e2) (% h2 (vector_sub w v))) ∧ (¬ collinear (INSERT v (INSERT w (INSERT w1 EMPTY))) ⟶ (0::real) < r1) ∧ (¬ collinear (INSERT v (INSERT w (INSERT w2 EMPTY))) ⟶ (0::real) < r2)))

thm DEF_azim:

azim = (λ(_1947959::(real, 3) cart) (_1947960::(real, 3) cart) (_1947961::(real, 3) cart) _1947962::(real, 3) cart. if collinear (INSERT _1947959 (INSERT _1947960 (INSERT _1947961 EMPTY))) ∨ collinear (INSERT _1947959 (INSERT _1947960 (INSERT _1947962 EMPTY))) then 0::real else SOME theta::real. (0::real) ≤ theta ∧ theta < real_of_nat (2::nat) ∗ pi ∧ (∃(h1::real) h2::real. ∀(e1::(real, 3) cart) (e2::(real, 3) cart) e3::(real, 3) cart. orthonormal e1 e2 e3 ∧ % (distance (_1947960, _1947959)) e3 = vector_sub _1947960 _1947959 ∧ _1947960 ≠ _1947959 ⟶ (∃(psi::real) (r1::real) r2::real. vector_sub _1947961 _1947959 = vector_add (% (r1 ∗ cos psi) e1) (vector_add (% (r1 ∗ sin psi) e2) (% h1 (vector_sub _1947960 _1947959))) ∧ vector_sub _1947962 _1947959 = vector_add (% (r2 ∗ cos (psi + theta)) e1) (vector_add (% (r2 ∗ sin (psi + theta)) e2) (% h2 (vector_sub _1947960 _1947959))) ∧ (0::real) < r1 ∧ (0::real) < r2)))

thm azim_def:

∀(w1::(real, 3) cart) (w2::(real, 3) cart) (w::(real, 3) cart) v::(real, 3) cart. azim v w w1 w2 = (if collinear (INSERT v (INSERT w (INSERT w1 EMPTY))) ∨ collinear (INSERT v (INSERT w (INSERT w2 EMPTY))) then 0::real else SOME theta::real. (0::real) ≤ theta ∧ theta < real_of_nat (2::nat) ∗ pi ∧ (∃(h1::real) h2::real. ∀(e1::(real, 3) cart) (e2::(real, 3) cart) e3::(real, 3) cart. orthonormal e1 e2 e3 ∧ % (distance (w, v)) e3 = vector_sub w v ∧ w ≠ v ⟶ (∃(psi::real) (r1::real) r2::real. vector_sub w1 v = vector_add (% (r1 ∗ cos psi) e1) (vector_add (% (r1 ∗ sin psi) e2) (% h1 (vector_sub w v))) ∧ vector_sub w2 v = vector_add (% (r2 ∗ cos (psi + theta)) e1) (vector_add (% (r2 ∗ sin (psi + theta)) e2) (% h2 (vector_sub w v))) ∧ (0::real) < r1 ∧ (0::real) < r2)))

thm azim:

∀(v::(real, 3) cart) (w::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. (0::real) ≤ azim v w w1 w2 ∧ azim v w w1 w2 < real_of_nat (2::nat) ∗ pi ∧ (∃(h1::real) h2::real. ∀(e1::(real, 3) cart) (e2::(real, 3) cart) e3::(real, 3) cart. orthonormal e1 e2 e3 ∧ % (distance (w, v)) e3 = vector_sub w v ∧ w ≠ v ⟶ (∃(psi::real) (r1::real) r2::real. vector_sub w1 v = vector_add (% (r1 ∗ cos psi) e1) (vector_add (% (r1 ∗ sin psi) e2) (% h1 (vector_sub w v)))

$\wedge$ *vector_sub w2 v* = *vector_add* (% (*r2* $*$ *cos* (*psi* + *azim v w w1 w2*)) *e1*) (*vector_add* (% (*r2* $*$ *sin* (*psi* + *azim v w w1 w2*)) *e2*) (% *h2* (*vector_sub w v*))) $\wedge$ ($\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT w1 EMPTY*))) $\longrightarrow$ (*0*::*real*) < *r1*) $\wedge$ ($\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT w2 EMPTY*))) $\longrightarrow$ (*0*::*real*) < *r2*)))

thm AZIM_UNIQUE:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*h1*::*real*) (*h2*::*real*) (*r1*::*real*) (*r2*::*real*) (*e1*::(*real*, *3*) *cart*) (*e2*::(*real*, *3*) *cart*) (*e3*::(*real*, *3*) *cart*) (*psi*::*real*) *theta*::*real*. (*0*::*real*) $\leq$ *theta* $\wedge$ *theta* < *real_of_nat* (*2*::*nat*) $*$ *pi* $\wedge$ *orthonormal e1 e2 e3* $\wedge$ % (*distance* (*w*, *v*)) *e3* = *vector_sub w v* $\wedge$ *w* $\neq$ *v* $\wedge$ (*0*::*real*) < *r1* $\wedge$ (*0*::*real*) < *r2* $\wedge$ *vector_sub w1 v* = *vector_add* (% (*r1* $*$ *cos psi*) *e1*) (*vector_add* (% (*r1* $*$ *sin psi*) *e2*) (% *h1* (*vector_sub w v*))) $\wedge$ *vector_sub w2 v* = *vector_add* (% (*r2* $*$ *cos* (*psi* + *theta*)) *e1*) (*vector_add* (% (*r2* $*$ *sin* (*psi* + *theta*)) *e2*) (% *h2* (*vector_sub w v*))) $\longrightarrow$ *azim v w w1 w2* = *theta*

thm AZIM_TRANSLATION:

$\forall$ (*a*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim* (*vector_add a v*) (*vector_add a w*) (*vector_add a w1*) (*vector_add a w2*) = *azim v w w1 w2*

thm AZIM_LINEAR_IMAGE:

$\forall$ *f*::(*real*, *3*) *cart* $\Rightarrow$ (*real*, *3*) *cart*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *3*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\wedge$ ((*2*::*nat*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *det* (*matrix f*) = (*1*::*real*)) $\longrightarrow$ ($\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim* (*f v*) (*f w*) (*f w1*) (*f w2*) = *azim v w w1 w2*)

thm AZIM_DEGENERATE_conjunct2:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT v* (*INSERT w* (*INSERT w2 EMPTY*))) $\longrightarrow$ *azim v w w1 w2* = (*0*::*real*)

thm AZIM_DEGENERATE_conjunct1:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT v* (*INSERT w* (*INSERT w1 EMPTY*))) $\longrightarrow$ *azim v w w1 w2* = (*0*::*real*)

thm AZIM_DEGENERATE_conjunct0:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *v* = *w* $\longrightarrow$ *azim v w w1 w2* = (*0*::*real*)

thm AZIM_DEGENERATE:

($\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *v* = *w* $\longrightarrow$ *azim v w w1 w2* = (*0*::*real*)) $\wedge$ ($\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT v* (*INSERT w* (*INSERT w1 EMPTY*))) ⟶ *azim v w w1 w2* = (*0*::*real*)) ∧ (∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT v* (*INSERT w* (*INSERT w2 EMPTY*))) ⟶ *azim v w w1 w2* = (*0*::*real*))

thm Trigonometry2.LT_IMP_ABS_REFL:

(*0*::*real*) < (*?a*::*real*) ⟶ |*?a*| = *?a*

thm AZIM_SPECIAL_SCALE:

∀ (*a*::*real*) (*v*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) < *a* ⟶ *azim* (*vec* (*0*::*nat*)) (% *a v*) *w1 w2* = *azim* (*vec* (*0*::*nat*)) *v w1 w2*

thm AZIM_SCALE_ALL:

∀ (*a*::*real*) (*v*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) < *a* ∧ (*0*::*real*) < (*?b*::*real*) ∧ (*0*::*real*) < (*?c*::*real*) ⟶ *azim* (*vec* (*0*::*nat*)) (% *a v*) (% *?b w1*) (% *?c w2*) = *azim* (*vec* (*0*::*nat*)) *v w1 w2*

thm DROPOUT_BASIS_3_conjunct2:

dropout (*3*::*nat*) (*basis* (*3*::*nat*)) = *vec* (*0*::*nat*)

thm DROPOUT_BASIS_3_conjunct1:

dropout (*3*::*nat*) (*basis* (*2*::*nat*)) = *basis* (*2*::*nat*)

thm DROPOUT_BASIS_3_conjunct0:

dropout (*3*::*nat*) (*basis* (*1*::*nat*)) = *basis* (*1*::*nat*)

thm CROSS_BASIS_conjunct5:

cross (*basis* (*1*::*nat*)) (*basis* (*3*::*nat*)) = *vector_neg* (*basis* (*2*::*nat*))

thm CROSS_BASIS_conjunct4:

cross (*basis* (*3*::*nat*)) (*basis* (*1*::*nat*)) = *basis* (*2*::*nat*)

thm CROSS_BASIS_conjunct3:

cross (*basis* (*3*::*nat*)) (*basis* (*2*::*nat*)) = *vector_neg* (*basis* (*1*::*nat*))

thm CROSS_BASIS_conjunct2:

cross (*basis* (*2*::*nat*)) (*basis* (*3*::*nat*)) = *basis* (*1*::*nat*)

thm CROSS_BASIS_conjunct1:

cross (*basis* (*2*::*nat*)) (*basis* (*1*::*nat*)) = *vector_neg* (*basis* (*3*::*nat*))

thm CROSS_BASIS_conjunct0:

cross (*basis* (*1*::*nat*)) (*basis* (*2*::*nat*)) = *basis* (*3*::*nat*)

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct1:

dot (*basis* (*2*::*nat*)) (*basis* (*2*::*nat*)) = (*1*::*real*)

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct3:

dot (*basis* (*1*::*nat*)) (*basis* (*2*::*nat*)) = (*0*::*real*)

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct4:

dot (*basis* (*1*::*nat*)) (*basis* (*3*::*nat*)) = (*0*::*real*)

thm Trigonometry2.ORTHONORMAL_BASIS3_conjunct5:

dot (*basis* (*2*::*nat*)) (*basis* (*3*::*nat*)) = (*0*::*real*)

thm AZIM_ARG:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *azim* (*vec* (*0*::*nat*)) (*basis* (*3*::*nat*)) *x y* = *Arg* (*complex_div* (*dropout* (*3*::*nat*) *y*) (*dropout* (*3*::*nat*) *x*))

thm REAL_CONTINUOUS_AT_AZIM:

∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT v* (*INSERT w* (*INSERT w1* (*INSERT w2 EMPTY*)))) ⟶ *real_continuous* (*azim v w w1*) (*at w2*)

thm AZIM_REFL:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *azim v0 v1 w w* = (*0*::*real*)

thm AZIM_EQ:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT y EMPTY*))) ⟶ (*azim v0 v1 w x* = *azim v0 v1 w y*) = *IN y* (*aff_gt* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT x EMPTY*))

thm AZIM_EQ_ALT:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT y EMPTY*))) ⟶ (*azim v0 v1 w x* = *azim v0 v1 w y*) = *IN x* (*aff_gt* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT y EMPTY*))

thm AZIM_EQ_0:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ⟶ (*azim v0 v1 w x* = (*0*::*real*)) = *IN w* (*aff_gt* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT x EMPTY*))

thm AZIM_EQ_0_ALT:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ⟶ (*azim v0 v1 w x* = (*0*::*real*)) = *IN x* (*aff_gt* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT w EMPTY*))

thm AZIM_EQ_0_GE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ⟶ (*azim v0 v1 w x* = (*0*::*real*)) = *IN w* (*aff_ge* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT x EMPTY*))

thm AZIM_COMPL_EQ_0:

∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) ∧ *azim z w w1 w2* = (*0*::*real*) ⟶ *azim z w w2 w1* = (*0*::*real*)

thm AZIM_COMPL:

∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) ⟶ *azim z w w2 w1* = (*if azim z w w1 w2* = (*0*::*real*) *then 0*::*real else real_of_nat* (*2*::*nat*) ∗ *pi* − *azim z w w1 w2*)

thm AZIM_EQ_PI_SYM:

∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) ⟶ (*azim z w w1 w2* = *pi*) = (*azim z w w2 w1* = *pi*)

thm AZIM_EQ_0_SYM:

∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) ⟶ (*azim z w w1 w2* = (*0*::*real*)) = (*azim z w w2 w1* = (*0*::*real*))

thm AZIM_EQ_0_GE_ALT:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT x EMPTY*))) ⟶ (*azim v0 v1 w x* = (*0*::*real*)) = *IN x* (*aff_ge* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT w EMPTY*))

thm AZIM_EQ_PI:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ x::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ x\ EMPTY))) \longrightarrow (azim\ v0\ v1\ w\ x = pi) = IN\ w\ (aff\_lt\ (INSERT\ v0\ (INSERT\ v1\ EMPTY))\ (INSERT\ x\ EMPTY))$

thm AZIM_EQ_PI_ALT:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ x::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ x\ EMPTY))) \longrightarrow (azim\ v0\ v1\ w\ x = pi) = IN\ x\ (aff\_lt\ (INSERT\ v0\ (INSERT\ v1\ EMPTY))\ (INSERT\ w\ EMPTY))$

thm AZIM_EQ_0_PI_IMP_COPLANAR:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ w2::(real, 3)\ cart.\ azim\ v0\ v1\ w1\ w2 = (0::real) \vee azim\ v0\ v1\ w1\ w2 = pi \longrightarrow coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w1\ (INSERT\ w2\ EMPTY))))$

thm AZIM_EQ_IMP:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ y\ EMPTY))) \wedge IN\ x\ (aff\_gt\ (INSERT\ v0\ (INSERT\ v1\ EMPTY))\ (INSERT\ y\ EMPTY)) \longrightarrow azim\ v0\ v1\ w\ x = azim\ v0\ v1\ w\ y$

thm AZIM_EQ_0_GE_IMP:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ x::(real, 3)\ cart.\ IN\ x\ (aff\_ge\ (INSERT\ v0\ (INSERT\ v1\ EMPTY))\ (INSERT\ w\ EMPTY)) \longrightarrow azim\ v0\ v1\ w\ x = (0::real)$

thm DEF_dihV:

$dihV = (\lambda(\_1950213::(real, ?'a::type)\ cart)\ (\_1950214::(real, ?'a::type)\ cart)\ (\_1950215::(real, ?'a::type)\ cart)\ \_1950216::(real, ?'a::type)\ cart.\ LET\ (\lambda va::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vb::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vc::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vap::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vbp::(real, ?'a::type)\ cart.\ LET\_END\ (arcV\ (vec\ (0::nat))\ vap\ vbp))\ (vector\_sub\ (\%\ (dot\ vc\ vc)\ vb)\ (\%\ (dot\ vb\ vc)\ vc))))\ (vector\_sub\ (\%\ (dot\ vc\ vc)\ va)\ (\%\ (dot\ va\ vc)\ vc))))\ (vector\_sub\ \_1950214\ \_1950213)))\ (vector\_sub\ \_1950216\ \_1950213)))\ (vector\_sub\ \_1950215\ \_1950213))$

thm dihV:

$\forall (w1::(real, ?'a::type)\ cart)\ (w3::(real, ?'a::type)\ cart)\ (w2::(real, ?'a::type)\ cart)\ w0::(real, ?'a::type)\ cart.\ dihV\ w0\ w1\ w2\ w3 = LET\ (\lambda va::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vb::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vc::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vap::(real, ?'a::type)\ cart.\ LET\_END\ (LET\ (\lambda vbp::(real, ?'a::type)\ cart.\ LET\_END\ (arcV\ (vec\ (0::nat))\ vap\ vbp))\ (vector\_sub$

(% (*dot vc vc*) *vb*) (% (*dot vb vc*) *vc*)))) (*vector_sub* (% (*dot vc vc*) *va*) (% (*dot va vc*) *vc*)))) (*vector_sub w1 w0*))) (*vector_sub w3 w0*))) (*vector_sub w2 w0*)

thm DIHV:

dihV (*?w0.0*::(*real*, *?'a*::*type*) *cart*) (*?w1.0*::(*real*, *?'a*::*type*) *cart*) (*?w2.0*::(*real*, *?'a*::*type*) *cart*) (*?w3.0*::(*real*, *?'a*::*type*) *cart*) = *LET* (λ*va*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vb*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vc*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vap*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vbp*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*angle* (*vap*, *vec* (*0*::*nat*), *vbp*))) (*vector_sub* (% (*dot vc vc*) *vb*) (% (*dot vb vc*) *vc*)))) (*vector_sub* (% (*dot vc vc*) *va*) (% (*dot va vc*) *vc*)))) (*vector_sub ?w1.0 ?w0.0*))) (*vector_sub ?w3.0 ?w0.0*))) (*vector_sub ?w2.0 ?w0.0*)

thm DIHV_TRANSLATION_EQ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*w0*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) (*w2*::(*real*, *?'a*::*type*) *cart*) *w3*::(*real*, *?'a*::*type*) *cart*. *dihV* (*vector_add a w0*) (*vector_add a w1*) (*vector_add a w2*) (*vector_add a w3*) = *dihV w0 w1 w2 w3*

thm DIHV_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*w0*::(*real*, *?'b*::*type*) *cart*) (*w1*::(*real*, *?'b*::*type*) *cart*) (*w2*::(*real*, *?'b*::*type*) *cart*) *w3*::(*real*, *?'b*::*type*) *cart*. *linear f* ∧ (∀ *x*::(*real*, *?'b*::*type*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) ⟶ *dihV* (*f w0*) (*f w1*) (*f w2*) (*f w3*) = *dihV w0 w1 w2 w3*

thm DIHV_SPECIAL_SCALE:

∀ (*a*::*real*) (*v*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) *w2*::(*real*, *?'a*::*type*) *cart*. *a* ≠ (*0*::*real*) ⟶ *dihV* (*vec* (*0*::*nat*)) (% *a v*) *w1 w2* = *dihV* (*vec* (*0*::*nat*)) *v w1 w2*

thm DIHV_RANGE:

∀ (*w0*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) (*w2*::(*real*, *?'a*::*type*) *cart*) *w3*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) ≤ *dihV w0 w1 w2 w3* ∧ *dihV w0 w1 w2 w3* ≤ *pi*

thm COS_AZIM_DIHV:

∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT v1 EMPTY*))) ∧ ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT v2 EMPTY*))) ⟶ *cos* (*azim v w v1 v2*) = *cos* (*dihV v w v1 v2*)

thm AZIM_DIHV_SAME:

∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT v1 EMPTY*))) ∧ ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT v2 EMPTY*))) ∧ *azim v w v1 v2* < *pi* ⟶ *azim v w v1 v2* = *dihV v w v1 v2*

thm AZIM_DIHV_COMPL:

$\forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ v2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT v1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT v2 EMPTY*))) $\wedge$ *pi* $\leq$ *azim v w v1 v2* $\longrightarrow$ *azim v w v1 v2* = *real_of_nat* (*2::nat*) $*$ *pi* $-$ *dihV v w v1 v2*

thm AZIM_DIVH:

$\forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ v2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT v1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT v2 EMPTY*))) $\longrightarrow$ *azim v w v1 v2* = (*if azim v w v1 v2* $<$ *pi then dihV v w v1 v2 else real_of_nat* (*2::nat*) $*$ *pi* $-$ *dihV v w v1 v2*)

thm AZIM_DIHV_EQ_0:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) $\longrightarrow$ (*azim v0 v1 w1 w2* = (*0::real*)) = (*dihV v0 v1 w1 w2* = (*0::real*))

thm AZIM_DIHV_EQ_PI:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) $\longrightarrow$ (*azim v0 v1 w1 w2* = *pi*) = (*dihV v0 v1 w1 w2* = *pi*)

thm AZIM_EQ_0_PI_EQ_COPLANAR:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) $\longrightarrow$ (*azim v0 v1 w1 w2* = (*0::real*) $\vee$ *azim v0 v1 w1 w2* = *pi*) = *coplanar* (*INSERT v0* (*INSERT v1* (*INSERT w1* (*INSERT w2 EMPTY*))))

thm DIHV_EQ_0_PI_EQ_COPLANAR:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) $\longrightarrow$ (*dihV v0 v1 w1 w2* = (*0::real*) $\vee$ *dihV v0 v1 w1 w2* = *pi*) = *coplanar* (*INSERT v0* (*INSERT v1* (*INSERT w1* (*INSERT w2 EMPTY*))))

thm DIHV_SYM:

$\forall (v0::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.$ *dihV v0 v1 v3 v2* = *dihV v0 v1 v2 v3*

thm DIHV_NEG:

$\forall (v0::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.$ *dihV* (*vector_neg v0*) (*vector_neg v1*) (*vector_neg v2*) (*vector_neg v3*) = *dihV v0 v1 v2 v3*

thm DIHV_NEG_0:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *dihV* (*vec* (*0*::*nat*)) (*vector_neg v1*) (*vector_neg v2*) (*vector_neg v3*) = *dihV* (*vec* (*0*::*nat*)) *v1 v2 v3*

thm ZENITH_EXISTS:

∀ (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *u* ≠ *v* ∧ *w* ≠ *v* ⟶ (∃ (*u'*::(*real*, *3*) *cart*) (*r*::*real*) (*phi*::*real*) *e3*::(*real*, *3*) *cart*. *phi* = *arcV v u w* ∧ *r* = *distance* (*u*, *v*) ∧ % (*distance* (*w*, *v*)) *e3* = *vector_sub w v* ∧ *dot u' e3* = (*0*::*real*) ∧ *u* = *vector_add v* (*vector_add u'* (% (*r* ∗ *cos phi*) *e3*)))

thm SPHERICAL_COORDINATES:

∀ (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*u'*::(*real*, *3*) *cart*) (*e1*::(*real*, *3*) *cart*) (*e2*::(*real*, *3*) *cart*) (*e3*::(*real*, *3*) *cart*) (*r*::*real*) (*phi*::*real*) *theta*::*real*. ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT v* (*INSERT w* (*INSERT u' EMPTY*))) ∧ *orthonormal e1 e2 e3* ∧ % (*distance* (*w*, *v*)) *e3* = *vector_sub w v* ∧ *IN* (*vector_add v e1*) (*aff_gt* (*INSERT v* (*INSERT w EMPTY*)) (*INSERT u EMPTY*)) ∧ *r* = *distance* (*v*, *u'*) ∧ *phi* = *arcV v u' w* ∧ *theta* = *azim v w u u'* ⟶ *u'* = *vector_add v* (*vector_add* (% (*r* ∗ (*cos theta* ∗ *sin phi*)) *e1*) (*vector_add* (% (*r* ∗ (*sin theta* ∗ *sin phi*)) *e2*) (% (*r* ∗ *cos phi*) *e3*)))

thm DEF_wedge:

wedge = (λ(*_1950679*::(*real*, *3*) *cart*) (*_1950680*::(*real*, *3*) *cart*) (*_1950681*::(*real*, *3*) *cart*) *_1950682*::(*real*, *3*) *cart*. *GSPEC* (λ*GEN%PVAR%2629*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2629* (¬ *collinear* (*INSERT _1950679* (*INSERT _1950680* (*INSERT y EMPTY*))) ∧ (*0*::*real*) < *azim _1950679 _1950680 _1950681 y* ∧ *azim _1950679 _1950680 _1950681 y* < *azim _1950679 _1950680 _1950681 _1950682*) *y*))

thm wedge:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *wedge v0 v1 w1 w2* = *GSPEC* (λ*GEN%PVAR%2629*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2629* (¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT y EMPTY*))) ∧ (*0*::*real*) < *azim v0 v1 w1 y* ∧ *azim v0 v1 w1 y* < *azim v0 v1 w1 w2*) *y*)

thm WEDGE_ALT:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *v0* ≠ *v1* ⟶ *wedge v0 v1 w1 w2* = *GSPEC* (λ*GEN%PVAR%2630*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2630* (¬ *IN y* (*hull affine* (*INSERT v0* (*INSERT v1 EMPTY*))) ∧ (*0*::*real*) < *azim v0 v1 w1 y* ∧ *azim v0 v1 w1 y* < *azim v0 v1 w1 w2*) *y*)

thm WEDGE_TRANSLATION:

∀ (*a*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *wedge* (*vector_add a v*) (*vector_add a w*) (*vector_add a w1*) (*vector_add a w2*) = *IMAGE* (*vector_add a*) (*wedge v w w1 w2*)

thm WEDGE_LINEAR_IMAGE:

∀ *f*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. *linear f* ∧ (∀ *x*::(*real*, *3*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) ∧ ((*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *det* (*matrix f*) = (*1*::*real*)) ⟶ (∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *wedge* (*f v*) (*f w*) (*f w1*) (*f w2*) = *IMAGE f* (*wedge v w w1 w2*))

thm WEDGE_SPECIAL_SCALE:

∀ (*a*::*real*) (*v*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) < *a* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (% *a v*) (*INSERT w2 EMPTY*))) ⟶ *wedge* (*vec* (*0*::*nat*)) (% *a v*) *w1 w2* = *wedge* (*vec* (*0*::*nat*)) *v w1 w2*

thm WEDGE_DEGENERATE:

(∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *z* = *w* ⟶ *wedge z w w1 w2* = *EMPTY*) ∧ (∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) ⟶ *wedge z w w1 w2* = *EMPTY*) ∧ (∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) ⟶ *wedge z w w1 w2* = *EMPTY*)

thm AFF_GT_LEMMA:

∀ (*v1*::*?'b*::*type*) *v2*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < (*?t1.0*::*real*) ∧ *v2* ≠ *vec* (*0*::*nat*) ⟶ *aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (% *?t1.0* (*basis* (*1*::*nat*))) (*INSERT v2 EMPTY*)) = *GSPEC* (λ*GEN%PVAR%2631*::(*real*, *?'a*::*type*) *cart*. ∃ (*a*::*real*) *b*::*real*. *SETSPEC GEN%PVAR%2631* ((*0*::*real*) < *a* ∧ (*0*::*real*) < *b*) (*vector_add* (% *a* (*basis* (*1*::*nat*))) (% *b v2*)))

thm WEDGE_LUNE_GT:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) ∧ (*0*::*real*) < *azim v0 v1 w1 w2* ∧ *azim v0 v1 w1 w2* < *pi* ⟶ *wedge v0 v1 w1 w2* = *aff_gt* (*INSERT v0* (*INSERT v1 EMPTY*)) (*INSERT w1* (*INSERT w2 EMPTY*))

thm WEDGE_LUNE_GE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT w2 EMPTY*))) ∧ (*0*::*real*) < *azim v0 v1 w1 w2* ∧ *azim v0 v1 w1 w2* < *pi* ⟶ *GSPEC* (λ*GEN%PVAR%2648*::(*real*, *3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%2648 ((0::real) ≤ azim v0 v1 w1 x ∧ azim v0 v1 w1 x ≤ azim v0 v1 w1 w2) x) = aff_ge (INSERT v0 (INSERT v1 EMPTY)) (INSERT w1 (INSERT w2 EMPTY))*

thm WEDGE_LUNE:

∀(v0::(real, 3) cart) (v1::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. ¬ coplanar (INSERT v0 (INSERT v1 (INSERT w1 (INSERT w2 EMPTY)))) ∧ azim v0 v1 w1 w2 < pi ⟶ wedge v0 v1 w1 w2 = aff_gt (INSERT v0 (INSERT v1 EMPTY)) (INSERT w1 (INSERT w2 EMPTY))

thm WEDGE_DEGENERATE_conjunct2:

∀(z::(real, 3) cart) (w::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. collinear (INSERT z (INSERT w (INSERT w2 EMPTY))) ⟶ wedge z w w1 w2 = EMPTY

thm WEDGE_DEGENERATE_conjunct1:

∀(z::(real, 3) cart) (w::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. collinear (INSERT z (INSERT w (INSERT w1 EMPTY))) ⟶ wedge z w w1 w2 = EMPTY

thm WEDGE_DEGENERATE_conjunct0:

∀(z::(real, 3) cart) (w::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. z = w ⟶ wedge z w w1 w2 = EMPTY

thm CROSS_EQ_SELF_conjunct1:

∀(x::(real, 3) cart) y::(real, 3) cart. (cross x y = y) = (y = vec (0::nat))

thm CROSS_EQ_SELF_conjunct0:

∀(x::(real, 3) cart) y::(real, 3) cart. (cross x y = x) = (x = vec (0::nat))

thm WEDGE:

wedge (?v1.0::(real, 3) cart) (?v2.0::(real, 3) cart) (?w1.0::(real, 3) cart) (?w2.0::(real, 3) cart) = (if collinear (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?w1.0 EMPTY))) ∨ collinear (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?w2.0 EMPTY))) then EMPTY else LET (λz::(real, 3) cart. LET_END (LET (λu1::(real, 3) cart. LET_END (LET (λu2::(real, 3) cart. LET_END (LET (λn::(real, 3) cart. LET_END (LET (λd::real. LET_END (if IN ?w2.0 (aff_ge (INSERT ?v1.0 (INSERT ?v2.0 EMPTY)) (INSERT ?w1.0 EMPTY)) then EMPTY else if IN ?w2.0 (aff_lt (INSERT ?v1.0 (INSERT ?v2.0 EMPTY)) (INSERT ?w1.0 EMPTY)) then aff_gt (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?w1.0 EMPTY))) (INSERT (vector_add ?v1.0 n) EMPTY) else if (0::real) < d then aff_gt (INSERT ?v1.0 (INSERT ?v2.0 EMPTY)) (INSERT ?w1.0 (INSERT ?w2.0 EMPTY)) else DIFF HOL_Light_Import.UNIV (aff_ge (INSERT ?v1.0 (INSERT ?v2.0 EMPTY)) (INSERT ?w1.0 (INSERT ?w2.0 EMPTY))))) (dot n u2))) (cross z u1))) (vector_sub ?w2.0 ?v1.0))) (vector_sub ?w1.0 ?v1.0))) (vector_sub ?v2.0 ?v1.0))

thm OPEN_WEDGE:

$\forall (z::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.$ $HOL\_Light\_Import.open\ (wedge\ z\ w\ w1\ w2)$

thm DEF_delta_x:

$delta\_x = (\lambda(\_1953948::real)\ (\_1953949::real)\ (\_1953950::real)\ (\_1953951::real)\ (\_1953952::real)\ \_1953953::real.\ \_1953948 * (\_1953951 * (-\ \_1953948 + (\_1953949 + (\_1953950 - \_1953951 + (\_1953952 + \_1953953))))) + (\_1953949 * (\_1953952 * (\_1953948 - \_1953949 + (\_1953950 + (\_1953951 - \_1953952 + \_1953953)))) + (\_1953950 * (\_1953953 * (\_1953948 + (\_1953949 - \_1953950 + (\_1953951 + (\_1953952 - \_1953953))))) - \_1953949 * (\_1953950 * \_1953951) - \_1953948 * (\_1953950 * \_1953952) - \_1953948 * (\_1953949 * \_1953953) - \_1953951 * (\_1953952 * \_1953953))))$

thm delta_x:

$\forall (x3::real)\ (x1::real)\ (x2::real)\ (x4::real)\ (x5::real)\ x6::real.\ delta\_x\ x1\ x2\ x3\ x4\ x5\ x6 = x1 * (x4 * (-\ x1 + (x2 + (x3 - x4 + (x5 + x6))))) + (x2 * (x5 * (x1 - x2 + (x3 + (x4 - x5 + x6)))) + (x3 * (x6 * (x1 + (x2 - x3 + (x4 + (x5 - x6))))) - x2 * (x3 * x4) - x1 * (x3 * x5) - x1 * (x2 * x6) - x4 * (x5 * x6)))$

thm VOLUME_OF_CLOSED_TETRAHEDRON:

$\forall (x1::(real,\ 3)\ cart)\ (x2::(real,\ 3)\ cart)\ (x3::(real,\ 3)\ cart)\ x4::(real,\ 3)\ cart.$ $HOL\_Light\_Import.measure\ (hull\ convex\ (INSERT\ x1\ (INSERT\ x2\ (INSERT\ x3\ (INSERT\ x4\ EMPTY))))) = sqrt\ (delta\_x\ ((distance\ (x1,\ x2))^2)\ ((distance\ (x1,\ x3))^2)\ ((distance\ (x1,\ x4))^2)\ ((distance\ (x3,\ x4))^2)\ ((distance\ (x2,\ x4))^2)\ ((distance\ (x2,\ x3))^2))\ /\ real\_of\_nat\ (12::nat)$

thm VOLUME_OF_TETRAHEDRON:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ v4::(real,\ 3)\ cart.$ $HOL\_Light\_Import.measure\ (conv0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY))))) = LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x14::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda x24::real.\ LET\_END\ (LET\ (\lambda x34::real.\ LET\_END\ (sqrt\ (delta\_x\ x12\ x13\ x14\ x34\ x24\ x23)\ /\ real\_of\_nat\ (12::nat)))\ ((distance\ (v3,\ v4))^2)))\ ((distance\ (v2,\ v4))^2)))\ ((distance\ (v2,\ v3))^2)))\ ((distance\ (v1,\ v4))^2)))\ ((distance\ (v1,\ v3))^2)))\ ((distance\ (v1,\ v2))^2)$

thm AREA_UNIT_CBALL:

$HOL\_Light\_Import.measure\ (cball\ (vec\ (0::nat),\ 1::real)) = pi$

thm AREA_CBALL:

$\forall (z::(real,\ 2)\ cart)\ r::real.\ (0::real) \leq r \longrightarrow HOL\_Light\_Import.measure\ (cball\ (z,\ r)) = pi * r^2$

thm AREA_BALL:

∀ (z::(real, 2) cart) r::real. (0::real) ≤ r ⟶ HOL_Light_Import.measure (ball (z, r)) = pi ∗ r^2

thm VOLUME_CBALL:

∀ (z::(real, 3) cart) r::real. (0::real) ≤ r ⟶ HOL_Light_Import.measure (cball (z, r)) = real_of_nat (4::nat) / real_of_nat (3::nat) ∗ (pi ∗ $r^{3::nat}$)

thm VOLUME_BALL:

∀ (z::(real, 3) cart) r::real. (0::real) ≤ r ⟶ HOL_Light_Import.measure (ball (z, r)) = real_of_nat (4::nat) / real_of_nat (3::nat) ∗ (pi ∗ $r^{3::nat}$)

thm DEF_rconesgn:

rconesgn = (λ(_1954037::real ⇒ real ⇒ bool) (_1954038::(real, ?'a::type) cart) (_1954039::(real, ?'a::type) cart) _1954040::real. GSPEC (λGEN%PVAR%2651::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2651 (_1954037 (dot (vector_sub x _1954038) (vector_sub _1954039 _1954038)) (distance (x, _1954038) ∗ (distance (_1954039, _1954038) ∗ _1954040))) x))

thm rconesgn:

∀ (sgn::real ⇒ real ⇒ bool) (w::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) h::real. rconesgn sgn v w h = GSPEC (λGEN%PVAR%2651::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2651 (sgn (dot (vector_sub x v) (vector_sub w v)) (distance (x, v) ∗ (distance (w, v) ∗ h))) x)

thm rcone_gt:

rcone_gt = rconesgn real_gt

thm rcone_ge:

rcone_ge = rconesgn real_ge

thm rcone_eq:

rcone_eq = rconesgn op =

thm DEF_frustum:

frustum = (λ(_1954069::(real, ?'a::type) cart) (_1954070::(real, ?'a::type) cart) (_1954071::real) (_1954072::real) _1954073::real. GSPEC (λGEN%PVAR%2652::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2652 (rcone_gt _1954069 _1954070 _1954073 y ∧ LET (λd::real. LET_END (LET (λn::real. LET_END (_1954071 ∗ n < d ∧ d < _1954072 ∗ n)) (vector_norm (vector_sub _1954070 _1954069)))) (dot (vector_sub y _1954069) (vector_sub _1954070 _1954069))) y))

thm frustum:

∀ (a::real) (h1::real) (h2::real) (v1::(real, ?'a::type) cart) v0::(real, ?'a::type) cart. frustum v0 v1 h1 h2 a = GSPEC (λGEN%PVAR%2652::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2652 (rcone_gt v0

$v1\ a\ y \wedge LET\ (\lambda d{::}real.\ LET\_END\ (LET\ (\lambda n{::}real.\ LET\_END\ (h1 * n < d \wedge d < h2 * n))\ (vector\_norm\ (vector\_sub\ v1\ v0))))\ (dot\ (vector\_sub\ y\ v0)\ (vector\_sub\ v1\ v0)))\ y)$

thm DEF_frustt:

$frustt = (\lambda(\_1954114{::}(real,\ ?'a{::}type)\ cart)\ \_1954115{::}(real,\ ?'a{::}type)\ cart.\ frustum\ \_1954114\ \_1954115\ (0{::}real))$

thm frustt:

$\forall (v0{::}(real,\ ?'a{::}type)\ cart)\ (v1{::}(real,\ ?'a{::}type)\ cart)\ (h{::}real)\ a{::}real.\ frustt\ v0\ v1\ h\ a = frustum\ v0\ v1\ (0{::}real)\ h\ a$

thm FRUSTUM_DEGENERATE:

$\forall (v0{::}(real,\ ?'a{::}type)\ cart)\ (h1{::}real)\ (h2{::}real)\ a{::}real.\ frustum\ v0\ v0\ h1\ h2\ a = EMPTY$

thm CONVEX_RCONE_GT:

$\forall (v0{::}(real,\ ?'a{::}type)\ cart)\ (v1{::}(real,\ ?'a{::}type)\ cart)\ a{::}real.\ (0{::}real) \leq a \longrightarrow convex\ (rcone\_gt\ v0\ v1\ a)$

thm OPEN_RCONE_GT:

$\forall (v0{::}(real,\ ?'a{::}type)\ cart)\ (v1{::}(real,\ ?'a{::}type)\ cart)\ a{::}real.\ HOL\_Light\_Import.open\ (rcone\_gt\ v0\ v1\ a)$

thm RCONE_GT_NEG:

$\forall (v0{::}(real,\ ?'a{::}type)\ cart)\ (v1{::}(real,\ ?'a{::}type)\ cart)\ a{::}real.\ rcone\_gt\ v0\ v1\ (-\ a) = IMAGE\ (vector\_sub\ (\%\ (real\_of\_nat\ (2{::}nat))\ v0))\ (DIFF\ HOL\_Light\_Import.UNIV\ (rcone\_ge\ v0\ v1\ a))$

thm VOLUME_FRUSTT_STRONG:

$\forall (v0{::}(real,\ 3)\ cart)\ (v1{::}(real,\ 3)\ cart)\ (h{::}real)\ a{::}real.\ (0{::}real) < a \longrightarrow bounded\ (frustt\ v0\ v1\ h\ a) \wedge convex\ (frustt\ v0\ v1\ h\ a) \wedge measurable\ (frustt\ v0\ v1\ h\ a) \wedge HOL\_Light\_Import.measure\ (frustt\ v0\ v1\ h\ a) = (if\ v1 = v0 \vee (1{::}real) \leq a \vee h < (0{::}real)\ then\ 0{::}real\ else\ pi * (((h\ /\ a)^2 - h^2) * (h\ /\ real\_of\_nat\ (3{::}nat))))$

thm VOLUME_FRUSTT:

$\forall (v0{::}(real,\ 3)\ cart)\ (v1{::}(real,\ 3)\ cart)\ (h{::}real)\ a{::}real.\ (0{::}real) < a \longrightarrow measurable\ (frustt\ v0\ v1\ h\ a) \wedge HOL\_Light\_Import.measure\ (frustt\ v0\ v1\ h\ a) = (if\ v1 = v0 \vee (1{::}real) \leq a \vee h < (0{::}real)\ then\ 0{::}real\ else\ pi * (((h\ /\ a)^2 - h^2) * (h\ /\ real\_of\_nat\ (3{::}nat))))$

thm DEF_scale:

$scale = (\lambda(\_1954378{::}(real,\ 3)\ cart)\ \_1954379{::}(real,\ 3)\ cart.\ vector\ [\$\ \_1954378\ (1{::}nat) * \$\ \_1954379\ (1{::}nat),\ \$\ \_1954378\ (2{::}nat) * \$\ \_1954379\ (2{::}nat),\ \$\ \_1954378\ (3{::}nat) * \$\ \_1954379\ (3{::}nat)])$

thm scale:

$\forall (t::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ scale\ t\ u = vector\ [\$\ t\ (1::nat) * \$\ u\ (1::nat),\ \$\ t\ (2::nat) * \$\ u\ (2::nat),\ \$\ t\ (3::nat) * \$\ u\ (3::nat)]$

thm DEF_normball:

$normball = (\lambda(\_1954390::(real,\ ?'a::type)\ cart)\ \_1954391::real.\ GSPEC\ (\lambda GEN\%PVAR\%2657::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2657\ (distance\ (y,\ \_1954390) < \_1954391)\ y))$

thm normball:

$\forall (x::(real,\ ?'a::type)\ cart)\ r::real.\ normball\ x\ r = GSPEC\ (\lambda GEN\%PVAR\%2657::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2657\ (distance\ (y,\ x) < r)\ y)$

thm DEF_ellipsoid:

$ellipsoid = (\lambda(\_1954402::(real,\ 3)\ cart)\ \_1954403::real.\ IMAGE\ (scale\ \_1954402)\ (normball\ (vec\ (0::nat))\ \_1954403))$

thm ellipsoid:

$\forall (t::(real,\ 3)\ cart)\ r::real.\ ellipsoid\ t\ r = IMAGE\ (scale\ t)\ (normball\ (vec\ (0::nat))\ r)$

thm NORMBALL_BALL:

$\forall (z::(real,\ ?'a::type)\ cart)\ r::real.\ normball\ z\ r = ball\ (z,\ r)$

thm MEASURE_SCALE:

$\forall s::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ s \longrightarrow measurable\ (IMAGE\ (scale\ (?t::(real,\ 3)\ cart))\ s) \wedge HOL\_Light\_Import.measure\ (IMAGE\ (scale\ ?t)\ s) = |\$\ ?t\ (1::nat) * (\$\ ?t\ (2::nat) * \$\ ?t\ (3::nat))| * HOL\_Light\_Import.measure\ s$

thm MEASURE_ELLIPSOID:

$\forall (t::(real,\ 3)\ cart)\ r::real.\ (0::real) \le r \longrightarrow measurable\ (ellipsoid\ t\ r) \wedge HOL\_Light\_Import.measure\ (ellipsoid\ t\ r) = |\$\ t\ (1::nat) * (\$\ t\ (2::nat) * \$\ t\ (3::nat))| * (real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * r^{3::nat}))$

thm MEASURABLE_ELLIPSOID:

$\forall (t::(real,\ 3)\ cart)\ r::real.\ measurable\ (ellipsoid\ t\ r)$

thm DEF_conic_cap:

$conic\_cap = (\lambda(\_1954511::(real,\ ?'a::type)\ cart)\ (\_1954512::(real,\ ?'a::type)\ cart)\ (\_1954513::real)\ \_1954514::real.\ HOL\_Light\_Import.INTER\ (normball\ \_1954511\ \_1954513)\ (rcone\_gt\ \_1954511\ \_1954512\ \_1954514))$

thm conic_cap:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ a::real.\ conic\_cap\ v0\ v1\ r\ a = HOL\_Light\_Import.INTER\ (normball\ v0\ r)\ (rcone\_gt\ v0\ v1\ a)$

thm CONIC_CAP_DEGENERATE:

$\forall (v0::(real, ?'a::type)\ cart)\ (r::real)\ a::real.\ conic\_cap\ v0\ v0\ r\ a = EMPTY$

thm BOUNDED_CONIC_CAP:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (r::real)\ a::real.\ bounded\ (conic\_cap\ v0\ v1\ r\ a)$

thm MEASURABLE_CONIC_CAP:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (r::real)\ a::real.\ measurable\ (conic\_cap\ v0\ v1\ r\ a)$

thm VOLUME_CONIC_CAP_STRONG:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (r::real)\ a::real.\ (0::real) < a \longrightarrow bounded\ (conic\_cap\ v0\ v1\ r\ a) \wedge convex\ (conic\_cap\ v0\ v1\ r\ a) \wedge measurable\ (conic\_cap\ v0\ v1\ r\ a) \wedge HOL\_Light\_Import.measure\ (conic\_cap\ v0\ v1\ r\ a) = (if\ v1 = v0 \vee (1::real) \leq a \vee r < (0::real)\ then\ 0::real\ else\ real\_of\_nat\ (2::nat) / real\_of\_nat\ (3::nat) * (pi * (((1::real) - a) * r^{3::nat})))$

thm VOLUME_CONIC_CAP:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (r::real)\ a::real.\ (0::real) < a \longrightarrow measurable\ (conic\_cap\ v0\ v1\ r\ a) \wedge HOL\_Light\_Import.measure\ (conic\_cap\ v0\ v1\ r\ a) = (if\ v1 = v0 \vee (1::real) \leq a \vee r < (0::real)\ then\ 0::real\ else\ real\_of\_nat\ (2::nat) / real\_of\_nat\ (3::nat) * (pi * (((1::real) - a) * r^{3::nat})))$

thm NEGLIGIBLE_CIRCULAR_CONE_0_NONPARALLEL:

$\forall (c::(real, ?'a::type)\ cart)\ k::real.\ c \neq vec\ (0::nat) \wedge k \neq (0::real) \wedge k \neq pi \longrightarrow negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2665::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2665\ (vector\_angle\ c\ x = k)\ x))$

thm NEGLIGIBLE_CIRCULAR_CONE_0:

$\forall (c::(real, ?'a::type)\ cart)\ k::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \wedge c \neq vec\ (0::nat) \longrightarrow negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2668::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2668\ (vector\_angle\ c\ x = k)\ x))$

thm NEGLIGIBLE_CIRCULAR_CONE:

$\forall (a::(real, ?'a::type)\ cart)\ (c::(real, ?'a::type)\ cart)\ k::real.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \wedge c \neq vec\ (0::nat) \longrightarrow negligible\ (INSERT\ a\ (GSPEC\ (\lambda GEN\%PVAR\%2670::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2670\ (vector\_angle\ c\ (vector\_sub\ x\ a) = k)\ x)))$

thm NEGLIGIBLE_RCONE_EQ:

$\forall (w::(real, 3)\ cart)\ (z::(real, 3)\ cart)\ h::real.\ w \neq z \longrightarrow negligible\ (rcone\_eq\ z\ w\ h)$

thm NEGLIGIBLE_ARG_EQ:

$\forall\, t::real.\ negligible\ (GSPEC\ (\lambda GEN\%PVAR\%2673::(real,\ 2)\ cart.\ \exists\, z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2673\ (Arg\ z = t)\ z))$

thm MEASURABLE_CLOSED_SECTOR_LE:

$\forall\, (r::real)\ t::real.\ measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2677::(real,\ 2)\ cart.\ \exists\, z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2677\ (vector\_norm\ z \le r \wedge Arg\ z \le t)\ z))$

thm MEASURABLE_CLOSED_SECTOR_LT:

$\forall\, (r::real)\ t::real.\ measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2680::(real,\ 2)\ cart.\ \exists\, z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2680\ (vector\_norm\ z \le r \wedge Arg\ z < t)\ z))$

thm MEASURABLE_CLOSED_SECTOR_LTE:

$\forall\, (r::real)\ (s::real)\ t::real.\ measurable\ (GSPEC\ (\lambda GEN\%PVAR\%2684::(real,\ 2)\ cart.\ \exists\, z::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2684\ (vector\_norm\ z \le r \wedge s < Arg\ z \wedge Arg\ z \le t)\ z))$

thm MEASURE_CLOSED_SECTOR_LE:

$\forall\, (t::real)\ r::real.\ (0::real) \le r \wedge (0::real) \le t \wedge t \le real\_of\_nat\ (2::nat) * pi \longrightarrow HOL\_Light\_Import.measure\ (GSPEC\ (\lambda GEN\%PVAR\%2697::(real,\ 2)\ cart.\ \exists\, x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2697\ (vector\_norm\ x \le r \wedge Arg\ x \le t)\ x)) = t * (r^2\ /\ real\_of\_nat\ (2::nat))$

thm HAS_MEASURE_OPEN_SECTOR_LT:

$\forall\, (t::real)\ r::real.\ (0::real) \le t \wedge t \le real\_of\_nat\ (2::nat) * pi \longrightarrow has\_measure\ (GSPEC\ (\lambda GEN\%PVAR\%2702::(real,\ 2)\ cart.\ \exists\, x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2702\ (vector\_norm\ x < r \wedge (0::real) < Arg\ x \wedge Arg\ x < t)\ x))\ (if\ (0::real) \le r\ then\ t * (r^2\ /\ real\_of\_nat\ (2::nat))\ else\ (0::real))$

thm MEASURE_OPEN_SECTOR_LT:

$\forall\, (t::real)\ r::real.\ (0::real) \le t \wedge t \le real\_of\_nat\ (2::nat) * pi \longrightarrow HOL\_Light\_Import.measure\ (GSPEC\ (\lambda GEN\%PVAR\%2703::(real,\ 2)\ cart.\ \exists\, x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2703\ (vector\_norm\ x < r \wedge (0::real) < Arg\ x \wedge Arg\ x < t)\ x)) = (if\ (0::real) \le r\ then\ t * (r^2\ /\ real\_of\_nat\ (2::nat))\ else\ (0::real))$

thm HAS_MEASURE_OPEN_SECTOR_LT_GEN:

$\forall\, (w::(real,\ 2)\ cart)\ z::(real,\ 2)\ cart.\ w \ne vec\ (0::nat) \longrightarrow has\_measure\ (GSPEC\ (\lambda GEN\%PVAR\%2704::(real,\ 2)\ cart.\ \exists\, x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2704\ (vector\_norm\ x < (?r::real) \wedge (0::real) < Arg\ (complex\_div\ x\ w) \wedge Arg\ (complex\_div\ x\ w) < Arg\ (complex\_div\ z\ w))\ x))\ (if\ (0::real) \le ?r\ then\ Arg\ (complex\_div\ z\ w) * (?r^2\ /\ real\_of\_nat\ (2::nat))\ else\ (0::real))$

thm MEASURABLE_BALL_WEDGE:

$\forall$ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *measurable* (*HOL_Light_Import.INTER* (*ball* (*z*, *?r*::*real*)) (*wedge z w w1 w2*))

thm VOLUME_BALL_WEDGE:

$\forall$ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*r*::*real*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) $\leq$ *r* $\longrightarrow$ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*z*, *r*)) (*wedge z w w1 w2*)) = *azim z w w1 w2* $*$ (*real_of_nat* (*2*::*nat*) $*$ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*)))

thm HAS_MEASURE_LUNE:

$\forall$ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*r*::*real*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) $\leq$ *r* $\wedge$ *w* $\neq$ *z* $\wedge$ $\neg$ *collinear* (*INSERT z* (*INSERT w* (*INSERT w1 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT z* (*INSERT w* (*INSERT w2 EMPTY*))) $\wedge$ *dihV z w w1 w2* $\neq$ *pi* $\longrightarrow$ *has_measure* (*HOL_Light_Import.INTER* (*ball* (*z*, *r*)) (*aff_gt* (*INSERT z* (*INSERT w EMPTY*)) (*INSERT w1* (*INSERT w2 EMPTY*)))) (*dihV z w w1 w2* $*$ (*real_of_nat* (*2*::*nat*) $*$ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*))))

thm HAS_MEASURE_LUNE_SIMPLE:

$\forall$ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*r*::*real*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. (*0*::*real*) $\leq$ *r* $\wedge$ $\neg$ *coplanar* (*INSERT z* (*INSERT w* (*INSERT w1* (*INSERT w2 EMPTY*)))) $\longrightarrow$ *has_measure* (*HOL_Light_Import.INTER* (*ball* (*z*, *r*)) (*aff_gt* (*INSERT z* (*INSERT w EMPTY*)) (*INSERT w1* (*INSERT w2 EMPTY*)))) (*dihV z w w1 w2* $*$ (*real_of_nat* (*2*::*nat*) $*$ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*))))

thm MEASURABLE_BALL_AFF_GT:

$\forall$ (*z*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *measurable* (*HOL_Light_Import.INTER* (*ball* (*z*, *r*)) (*aff_gt s t*))

thm AFF_GT_SHUFFLE:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *FINITE s* $\wedge$ *FINITE t* $\wedge$ *IN* (*vec* (*0*::*nat*)) *s* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v s* $\wedge$ $\neg$ *IN* (*vector_neg v*) *s* $\wedge$ $\neg$ *IN v t* $\longrightarrow$ *aff_gt* (*INSERT v s*) *t* = *HOL_Light_Import.UNION* (*aff_gt s* (*INSERT v t*)) (*HOL_Light_Import.UNION* (*aff_gt s* (*INSERT* (*vector_neg v*) *t*)) (*aff_gt s t*))

thm MEASURE_BALL_AFF_GT_SHUFFLE_LEMMA:

$\forall$ (*r*::*real*) (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) $\leq$ *r* $\wedge$ *independent* (*INSERT v* (*HOL_Light_Import.UNION* (*DELETE s* (*vec* (*0*::*nat*))) *t*)) $\wedge$ *FINITE s* $\wedge$ *FINITE t* $\wedge$ *CARD* (*HOL_Light_Import.UNION s t*) $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *IN* (*vec* (*0*::*nat*)) *s* $\wedge$ $\neg$ *IN* (*vec* (*0*::*nat*)) *t* $\wedge$ $\neg$ *IN v s* $\wedge$ $\neg$ *IN* (*vector_neg v*) *s* $\wedge$ $\neg$ *IN v t* $\longrightarrow$ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT v s*) *t*)) = *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt s* (*INSERT v t*))) + *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt s* (*INSERT* (*vector_neg v*) *t*)))

thm MEASURE_BALL_AFF_GT_SHUFFLE:

∀ (*r*::*real*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) ≤ *r* ∧ ¬ *IN v* (*HOL_Light_Import.UNION s t*) ∧ *independent* (*INSERT v* (*HOL_Light_Import.UNION s t*)) ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v s*)) *t*)) = *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *s*) (*INSERT v t*))) + *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *s*) (*INSERT* (*vector_neg v*) *t*)))

thm MEASURE_LUNE_DECOMPOSITION:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. (*0*::*real*) ≤ (*?r*::*real*) ∧ ¬ *coplanar* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *?r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))))) + *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *?r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*vector_neg v1*) (*INSERT v2* (*INSERT v3 EMPTY*))))) = *dihV* (*vec* (*0*::*nat*)) *v1 v2 v3* ∗ (*real_of_nat* (*2*::*nat*) ∗ ($?r^{3::nat}$ / *real_of_nat* (*3*::*nat*)))

thm SOLID_TRIANGLE_CONGRUENT_NEG:

∀ (*r*::*real*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*vector_neg v1*) (*INSERT* (*vector_neg v2*) (*INSERT* (*vector_neg v3*) *EMPTY*))))) = *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))))

thm VOLUME_SOLID_TRIANGLE:

∀ (*r*::*real*) (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. (*0*::*real*) < *r* ∧ ¬ *coplanar* (*INSERT v0* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*ball* (*v0*, *r*)) (*aff_gt* (*INSERT v0 EMPTY*) (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))))) = *LET* (λ*a123*::*real*. *LET_END* (*LET* (λ*a231*::*real*. *LET_END* (*LET* (λ*a312*::*real*. *LET_END* ((*a123* + (*a231* + (*a312* − *pi*))) ∗ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*)))) (*dihV v0 v3 v1 v2*))) (*dihV v0 v2 v3 v1*))) (*dihV v0 v1 v2 v3*)

thm MEASURABLE_BOUNDED_INTER_OPEN:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *measurable s* ∧ *bounded s* ∧ *HOL_Light_Import.open t* ⟶ *measurable* (*HOL_Light_Import.INTER s t*)

thm SLICE_SPECIAL_WEDGE:

$\forall$ (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*basis* (*3*::*nat*)) (*INSERT* *w1* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*basis* (*3*::*nat*)) (*INSERT* *w2* *EMPTY*))) ⟶ *slice* (*3*::*nat*) (*?t*::*real*) (*wedge* (*vec* (*0*::*nat*)) (*basis* (*3*::*nat*)) *w1* *w2*) = *GSPEC* (λ*GEN%PVAR%2705*::(*real*, *2*) *cart*. ∃ *z*::(*real*, *2*) *cart*. *SETSPEC* *GEN%PVAR%2705* ((*0*::*real*) < *Arg* (*complex_div* *z* (*dropout* (*3*::*nat*) *w1*)) ∧ *Arg* (*complex_div* *z* (*dropout* (*3*::*nat*) *w1*)) < *Arg* (*complex_div* (*dropout* (*3*::*nat*) *w2*) (*dropout* (*3*::*nat*) *w1*))) *z*)

thm VOLUME_FRUSTT_WEDGE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*h*::*real*) *a*::*real*. (*0*::*real*) < *a* ∧ ¬ *collinear* (*INSERT* *v0* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *v0* (*INSERT* *v1* (*INSERT* *w2* *EMPTY*))) ⟶ *bounded* (*HOL_Light_Import.INTER* (*frustt* *v0* *v1* *h* *a*) (*wedge* *v0* *v1* *w1* *w2*)) ∧ *measurable* (*HOL_Light_Import.INTER* (*frustt* *v0* *v1* *h* *a*) (*wedge* *v0* *v1* *w1* *w2*)) ∧ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*frustt* *v0* *v1* *h* *a*) (*wedge* *v0* *v1* *w1* *w2*)) = (*if* (*1*::*real*) ≤ *a* ∨ *h* < (*0*::*real*) *then* *0*::*real* *else* *azim* *v0* *v1* *w1* *w2* ∗ (((h / a)^2 − h^2) ∗ (h / *real_of_nat* (*6*::*nat*))))

thm VOLUME_CONIC_CAP_WEDGE_WEAK:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*r*::*real*) *a*::*real*. (*0*::*real*) < *a* ∧ ¬ *collinear* (*INSERT* *v0* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *v0* (*INSERT* *v1* (*INSERT* *w2* *EMPTY*))) ⟶ *bounded* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*)) ∧ *measurable* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*)) ∧ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*)) = (*if* (*1*::*real*) ≤ *a* ∨ *r* < (*0*::*real*) *then* *0*::*real* *else* *azim* *v0* *v1* *w1* *w2* / *real_of_nat* (*3*::*nat*) ∗ (((*1*::*real*) − *a*) ∗ $r^{3::nat}$))

thm BOUNDED_CONIC_CAP_WEDGE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*r*::*real*) *a*::*real*. *bounded* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*))

thm MEASURABLE_CONIC_CAP_WEDGE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*r*::*real*) *a*::*real*. *measurable* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*))

thm VOLUME_CONIC_CAP_COMPL:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*r*::*real*) *a*::*real*. (*0*::*real*) ≤ *r* ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* *a*) (*wedge* *v0* *v1* *w1* *w2*)) + *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*conic_cap* *v0* *v1* *r* (− *a*)) (*wedge* *v0* *v1* *w1* *w2*)) = *azim* *v0* *v1* *w1* *w2* ∗ (*real_of_nat* (*2*::*nat*) ∗ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*)))

thm VOLUME_CONIC_CAP_WEDGE_MEDIUM:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ (w2::(real,\ 3)\ cart)\ (r::real)\ a::real.\ (0::real) \leq a \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w2\ EMPTY))) \longrightarrow bounded\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) \wedge measurable\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) = (if\ (1::real) < |a| \vee r < (0::real)\ then\ 0::real\ else\ azim\ v0\ v1\ w1\ w2\ /\ real\_of\_nat\ (3::nat) * (((1::real) - a) * r^{3::nat}))$

thm VOLUME_CONIC_CAP_WEDGE:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ (w2::(real,\ 3)\ cart)\ (r::real)\ a::real.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w2\ EMPTY))) \longrightarrow bounded\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) \wedge measurable\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) = (if\ (1::real) < a \vee r < (0::real)\ then\ 0::real\ else\ azim\ v0\ v1\ w1\ w2\ /\ real\_of\_nat\ (3::nat) * (((1::real) - max\ a\ (-\ (1::real))) * r^{3::nat}))$

thm DEF_cone:

$cone = (\lambda\_1958170::(real,\ ?'a::type)\ cart.\ affsign\ sgn\_ge\ (INSERT\ \_1958170\ EMPTY))$

thm cone:

$\forall (v::(real,\ ?'a::type)\ cart)\ S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ cone\ v\ S = affsign\ sgn\_ge\ (INSERT\ v\ EMPTY)\ S$

thm DEF_cone0:

$cone0 = (\lambda\_1958182::(real,\ ?'a::type)\ cart.\ affsign\ sgn\_gt\ (INSERT\ \_1958182\ EMPTY))$

thm cone0:

$\forall (v::(real,\ ?'a::type)\ cart)\ S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ cone0\ v\ S = affsign\ sgn\_gt\ (INSERT\ v\ EMPTY)\ S$

thm DEF_solid_triangle:

$solid\_triangle = (\lambda(\_1958194::(real,\ ?'a::type)\ cart)\ (\_1958195::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1958196::real.\ HOL\_Light\_Import.INTER\ (normball\ \_1958194\ \_1958196)\ (cone0\ \_1958194\ \_1958195))$

thm solid_triangle:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ solid\_triangle\ v0\ S\ r = HOL\_Light\_Import.INTER\ (normball\ v0\ r)\ (cone0\ v0\ S)$

thm DEF_rect:

rect = (λ(*_1958215*::(*real*, *3*) *cart*) *_1958216*::(*real*, *3*) *cart*. *GSPEC* (λ*GEN%PVAR%2711*::(*real*, *3*) *cart*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2711* ($\forall$ *i*::*nat*. $ *_1958215* *i* < $ *v i* $\wedge$ $ *v i* < $ *_1958216 i*) *v*))

thm rect:

$\forall$ (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *rect a b* = *GSPEC* (λ*GEN%PVAR%2711*::(*real*, *3*) *cart*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2711* ($\forall$ *i*::*nat*. $ *a i* < $ *v i* $\wedge$ $ *v i* < $ *b i*) *v*)

thm RECT_INTERVAL:

$\forall$ (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *rect a b* = *open_interval* (*a*, *b*)

thm RCONE_GE_GT:

rcone_ge (*?z*::(*real*, *?'a*::*type*) *cart*) (*?w*::(*real*, *?'a*::*type*) *cart*) (*?h*::*real*) = *HOL_Light_Import.UNION* (*rcone_gt ?z ?w ?h*) (*GSPEC* (λ*GEN%PVAR%2712*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2712* (*dot* (*vector_sub x ?z*) (*vector_sub ?w ?z*) = *vector_norm* (*vector_sub x ?z*) * (*vector_norm* (*vector_sub ?w ?z*) * *?h*)) *x*))

thm RCONE_GT_GE:

rcone_gt (*?z*::(*real*, *?'a*::*type*) *cart*) (*?w*::(*real*, *?'a*::*type*) *cart*) (*?h*::*real*) = *DIFF* (*rcone_ge ?z ?w ?h*) (*GSPEC* (λ*GEN%PVAR%2713*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2713* (*dot* (*vector_sub x ?z*) (*vector_sub ?w ?z*) = *vector_norm* (*vector_sub x ?z*) * (*vector_norm* (*vector_sub ?w ?z*) * *?h*)) *x*))

thm DEF_sphere:

sphere = (λ*_1958393*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*v*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) < *r* $\wedge$ *_1958393* = *GSPEC* (λ*GEN%PVAR%2714*::(*real*, *3*) *cart*. $\exists$ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2714* (*vector_norm* (*vector_sub w v*) = *r*) *w*))

thm sphere:

$\forall$ *x*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *sphere x* = ($\exists$ (*v*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) < *r* $\wedge$ *x* = *GSPEC* (λ*GEN%PVAR%2714*::(*real*, *3*) *cart*. $\exists$ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2714* (*vector_norm* (*vector_sub w v*) = *r*) *w*))

thm DEF_c_cone:

c_cone = (λ*_1958398*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ *real*. *GSPEC* (λ*GEN%PVAR%2715*::(*real*, *3*) *cart*. $\exists$ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2715* (*dot* (*vector_sub x* (*fst _1958398*)) (*fst* (*snd _1958398*)) = *vector_norm* (*vector_sub x* (*fst _1958398*)) * (*vector_norm* (*fst* (*snd _1958398*)) * *snd* (*snd _1958398*))) *x*))

thm c_cone:

$\forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ r::real.\ c\_cone\ (v,\ w,\ r) = GSPEC\ (\lambda GEN\%PVAR\%2715::(real,\ 3)\ cart.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2715\ (dot\ (vector\_sub\ x\ v)\ w = vector\_norm\ (vector\_sub\ x\ v) * (vector\_norm\ w * r))\ x)$

thm DEF_circular_cone:

$circular\_cone = (\lambda\_1958411::(real,\ 3)\ cart \Rightarrow bool.\ Ex\ (GABS\ (\lambda f::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ \forall (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ GEQ\ (f\ (v,\ w))\ (\exists r::real.\ w \neq vec\ (0::nat) \land \_1958411 = c\_cone\ (v,\ w,\ r)))))$

thm circular_cone:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ circular\_cone\ V = Ex\ (GABS\ (\lambda f::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ \forall (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ GEQ\ (f\ (v,\ w))\ (\exists r::real.\ w \neq vec\ (0::nat) \land V = c\_cone\ (v,\ w,\ r))))$

thm NULLSET_RULES_conjunct0:

$\forall P::(real,\ 3)\ cart \Rightarrow bool.\ plane\ P \lor sphere\ P \lor circular\_cone\ P \longrightarrow negligible\ P$

thm NULLSET_RULES:

$(\forall P::(real,\ 3)\ cart \Rightarrow bool.\ plane\ P \lor sphere\ P \lor circular\_cone\ P \longrightarrow negligible\ P) \land (\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart \Rightarrow bool.\ negligible\ s \land negligible\ t \longrightarrow negligible\ (HOL\_Light\_Import.UNION\ s\ t))$

thm DEF_primitive:

$primitive = (\lambda\_1958420::(real,\ 3)\ cart \Rightarrow bool.\ (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ \_1958420 = solid\_triangle\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))\ r) \lor (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ \_1958420 = conv0\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))))) \lor (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (h::real)\ a::real.\ (0::real) < a \land \_1958420 = HOL\_Light\_Import.INTER\ (frustt\ v0\ v1\ h\ a)\ (wedge\ v0\ v1\ v2\ v3)) \lor (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (r::real)\ c::real.\ \_1958420 = HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ c)\ (wedge\ v0\ v1\ v2\ v3)) \lor (\exists (a::(real,\ 3)\ cart)\ b::(real,\ 3)\ cart.\ \_1958420 = rect\ a\ b) \lor (\exists (t::(real,\ 3)\ cart)\ r::real.\ \_1958420 = ellipsoid\ t\ r) \lor (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ \_1958420 = HOL\_Light\_Import.INTER\ (normball\ v0\ r)\ (wedge\ v0\ v1\ v2\ v3)))$

thm primitive:

$\forall C::(real,\ 3)\ cart \Rightarrow bool.\ primitive\ C = ((\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ C = solid\_triangle\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))\ r) \lor (\exists (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ C = conv0\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))))) \lor (\exists (v0::(real,\ 3)$

cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (h::real) a::real. (0::real) < a ∧ C = HOL_Light_Import.INTER (frustt v0 v1 h a) (wedge v0 v1 v2 v3)) ∨ (∃ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (r::real) c::real. C = HOL_Light_Import.INTER (conic_cap v0 v1 r c) (wedge v0 v1 v2 v3)) ∨ (∃ (a::(real, 3) cart) b::(real, 3) cart. C = rect a b) ∨ (∃ (t::(real, 3) cart) r::real. C = ellipsoid t r) ∨ (∃ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) r::real. C = HOL_Light_Import.INTER (normball v0 r) (wedge v0 v1 v2 v3)))

thm MEASURABLE_RULES_conjunct2:

∀ (X::(real, 3) cart ⇒ bool) t::(real, 3) cart. measurable X ⟶ measurable (IMAGE (scale t) X)

thm MEASURABLE_RULES_conjunct0:

∀ C::(real, 3) cart ⇒ bool. primitive C ⟶ measurable C

thm MEASURABLE_RULES:

(∀ C::(real, 3) cart ⇒ bool. primitive C ⟶ measurable C) ∧ (∀ Z::(real, 3) cart ⇒ bool. negligible Z ⟶ measurable Z) ∧ (∀ (X::(real, 3) cart ⇒ bool) t::(real, 3) cart. measurable X ⟶ measurable (IMAGE (scale t) X)) ∧ (∀ (X::(real, ?'a::type) cart ⇒ bool) v::(real, ?'a::type) cart. measurable X ⟶ measurable (IMAGE (vector_add v) X)) ∧ (∀ (s::(real, 3) cart ⇒ bool) t::(real, 3) cart ⇒ bool. measurable s ∧ measurable t ⟶ measurable (HOL_Light_Import.UNION s t)) ∧ (∀ (s::(real, 3) cart ⇒ bool) t::(real, 3) cart ⇒ bool. measurable s ∧ measurable t ⟶ measurable (HOL_Light_Import.INTER s t)) ∧ (∀ (s::(real, 3) cart ⇒ bool) t::(real, 3) cart ⇒ bool. measurable s ∧ measurable t ⟶ measurable (DIFF s t))

thm DEF_vol_solid_triangle:

vol_solid_triangle = (λ(_1958427::(real, ?'a::type) cart) (_1958428::(real, ?'a::type) cart) (_1958429::(real, ?'a::type) cart) (_1958430::(real, ?'a::type) cart) _1958431::real. LET (λa123::real. LET_END (LET (λa231::real. LET_END (LET (λa312::real. LET_END ((a123 + (a231 + (a312 − pi))) ∗ ($\_1958431^{3::nat}$ / real_of_nat (3::nat)))) (dihV _1958427 _1958430 _1958428 _1958429))) (dihV _1958427 _1958429 _1958430 _1958428))) (dihV _1958427 _1958428 _1958429 _1958430))

thm vol_solid_triangle:

∀ (r::real) (v0::(real, ?'a::type) cart) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. vol_solid_triangle v0 v1 v2 v3 r = LET (λa123::real. LET_END (LET (λa231::real. LET_END (LET (λa312::real. LET_END ((a123 + (a231 + (a312 − pi))) ∗ ($r^{3::nat}$ / real_of_nat (3::nat)))) (dihV v0 v3 v1 v2))) (dihV v0 v2 v3 v1))) (dihV v0 v1 v2 v3)

thm DEF_vol_frustt_wedge:

vol_frustt_wedge = (λ(_1958472::(real, 3) cart) (_1958473::(real, 3) cart) (_1958474::(real, 3) cart) (_1958475::(real, 3) cart) (_1958476::real) _1958477::real. azim _1958472

_1958473 _1958474 _1958475 ∗ (*_1958476*$^{3::nat}$ ∗ (((*1*::*real*) / (*_1958477* ∗ *_1958477*) − (*1*::*real*)) / *real_of_nat* (*6*::*nat*))))

thm vol_frustt_wedge:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*h*::*real*) *a*::*real*. *vol_frustt_wedge v0 v1 v2 v3 h a* = *azim v0 v1 v2 v3* ∗ (*h*$^{3::nat}$ ∗ (((*1*::*real*) / (*a* ∗ *a*) − (*1*::*real*)) / *real_of_nat* (*6*::*nat*)))

thm DEF_vol_conic_cap_wedge:

vol_conic_cap_wedge = (λ(*_1958532*::(*real*, *3*) *cart*) (*_1958533*::(*real*, *3*) *cart*) (*_1958534*::(*real*, *3*) *cart*) (*_1958535*::(*real*, *3*) *cart*) (*_1958536*::*real*) *_1958537*::*real*. *azim _1958532 _1958533 _1958534 _1958535* ∗ (((*1*::*real*) − *_1958537*) ∗ (*_1958536*$^{3::nat}$ / *real_of_nat* (*3*::*nat*))))

thm vol_conic_cap_wedge:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*c*::*real*) *r*::*real*. *vol_conic_cap_wedge v0 v1 v2 v3 r c* = *azim v0 v1 v2 v3* ∗ (((*1*::*real*) − *c*) ∗ (*r*$^{3::nat}$ / *real_of_nat* (*3*::*nat*)))

thm DEF_vol_conv:

vol_conv = (λ(*_1958592*::(*real*, *?'a*::*type*) *cart*) (*_1958593*::(*real*, *?'a*::*type*) *cart*) (*_1958594*::(*real*, *?'a*::*type*) *cart*) *_1958595*::(*real*, *?'a*::*type*) *cart*. *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*sqrt* (*delta_x x12 x13 x14 x34 x24 x23*) / *real_of_nat* (*12*::*nat*))) ((*distance* (*_1958594*, *_1958595*))^2))) ((*distance* (*_1958593*, *_1958595*))^2))) ((*distance* (*_1958593*, *_1958594*))^2))) ((*distance* (*_1958592*, *_1958595*))^2))) ((*distance* (*_1958592*, *_1958594*))^2))) ((*distance* (*_1958592*, *_1958593*))^2))

thm vol_conv:

∀ (*v4*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. *vol_conv v1 v2 v3 v4* = *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*sqrt* (*delta_x x12 x13 x14 x34 x24 x23*) / *real_of_nat* (*12*::*nat*))) ((*distance* (*v3*, *v4*))^2))) ((*distance* (*v2*, *v4*))^2))) ((*distance* (*v2*, *v3*))^2))) ((*distance* (*v1*, *v4*))^2))) ((*distance* (*v1*, *v3*))^2))) ((*distance* (*v1*, *v2*))^2)

thm DEF_vol_rect:

vol_rect = (λ(*_1958624*::(*real*, *?'b*::*type*) *cart*) *_1958625*::(*real*, *?'a*::*type*) *cart*. *if* $ *_1958624* (*1*::*nat*) < $ *_1958625* (*1*::*nat*) ∧ $ *_1958624* (*2*::*nat*) < $ *_1958625* (*2*::*nat*) ∧ $ *_1958624* (*3*::*nat*) < $ *_1958625* (*3*::*nat*) *then* ($ *_1958625* (*3*::*nat*) − $ *_1958624* (*3*::*nat*)) ∗ (($ *_1958625* (*2*::*nat*) − $ *_1958624* (*2*::*nat*)) ∗ ($ *_1958625* (*1*::*nat*) − $ *_1958624* (*1*::*nat*))) *else* (*0*::*real*))

thm vol_rect:

$\forall (b::(real,\ ?'b::type)\ cart)\ a::(real,\ ?'a::type)\ cart.\ vol\_rect\ a\ b = (if\ \$\ a\ (1::nat) < \$\ b\ (1::nat) \wedge \$\ a\ (2::nat) < \$\ b\ (2::nat) \wedge \$\ a\ (3::nat) < \$\ b\ (3::nat)\ then\ (\$\ b\ (3::nat) - \$\ a\ (3::nat)) * ((\$\ b\ (2::nat) - \$\ a\ (2::nat)) * (\$\ b\ (1::nat) - \$\ a\ (1::nat)))\ else\ (0::real))$

thm DEF_vol_ball_wedge:

$vol\_ball\_wedge = (\lambda(\_1958636::(real,\ 3)\ cart)\ (\_1958637::(real,\ 3)\ cart)\ (\_1958638::(real,\ 3)\ cart)\ (\_1958639::(real,\ 3)\ cart)\ \_1958640::real.\ azim\ \_1958636\ \_1958637\ \_1958638\ \_1958639 * (real\_of\_nat\ (2::nat) * (\_1958640^{3::nat}\ /\ real\_of\_nat\ (3::nat))))$

thm vol_ball_wedge:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ vol\_ball\_wedge\ v0\ v1\ v2\ v3\ r = azim\ v0\ v1\ v2\ v3 * (real\_of\_nat\ (2::nat) * (r^{3::nat}\ /\ real\_of\_nat\ (3::nat)))$

thm DEF_SDIFF:

$SDIFF = (\lambda(\_1958681::?'a::type \Rightarrow bool)\ \_1958682::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (DIFF\ \_1958681\ \_1958682)\ (DIFF\ \_1958682\ \_1958681))$

thm SDIFF:

$\forall (Y::?'a::type \Rightarrow bool)\ X::?'a::type \Rightarrow bool.\ SDIFF\ X\ Y = HOL\_Light\_Import.UNION\ (DIFF\ X\ Y)\ (DIFF\ Y\ X)$

thm volume_props_conjunct1:

$\forall Z::(real,\ 3)\ cart \Rightarrow bool.\ negligible\ Z \longrightarrow HOL\_Light\_Import.measure\ Z = (0::real)$

thm volume_props_conjunct2:

$\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ Y::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ X \wedge measurable\ Y \wedge negligible\ (SDIFF\ X\ Y) \longrightarrow HOL\_Light\_Import.measure\ X = HOL\_Light\_Import.measure\ Y$

thm volume_props_conjunct3:

$\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart.\ measurable\ X \wedge measurable\ (IMAGE\ (scale\ t)\ X) \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ (scale\ t)\ X) = |\$\ t\ (1::nat) * (\$\ t\ (2::nat) * \$\ t\ (3::nat))| * HOL\_Light\_Import.measure\ X$

thm volume_props_conjunct4:

$\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ measurable\ X \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ (vector\_add\ v)\ X) = HOL\_Light\_Import.measure\ X$

thm volume_props_conjunct6:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (conv0\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))))) = vol\_conv\ v0\ v1\ v2\ v3$

thm volume_props_conjunct9:

$\forall (a::(real,\ 3)\ cart)\ b::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (rect\ a\ b) = vol\_rect\ a\ b$

thm volume_props:

$(\forall C::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ C \longrightarrow (0::real) \leq HOL\_Light\_Import.measure\ C) \wedge (\forall Z::(real,\ 3)\ cart \Rightarrow bool.\ negligible\ Z \longrightarrow HOL\_Light\_Import.measure\ Z = (0::real)) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ Y::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ X \wedge measurable\ Y \wedge negligible\ (SDIFF\ X\ Y) \longrightarrow HOL\_Light\_Import.measure\ X = HOL\_Light\_Import.measure\ Y) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart.\ measurable\ X \wedge measurable\ (IMAGE\ (scale\ t)\ X) \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ (scale\ t)\ X) = |\$\ t\ (1::nat) * (\$\ t\ (2::nat) * \$\ t\ (3::nat))| * HOL\_Light\_Import.measure\ X) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ measurable\ X \longrightarrow HOL\_Light\_Import.measure\ (IMAGE\ (vector\_add\ v)\ X) = HOL\_Light\_Import.measure\ X) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ (0::real) < r \wedge \neg\ coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow HOL\_Light\_Import.measure\ (solid\_triangle\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))\ r) = vol\_solid\_triangle\ v0\ v1\ v2\ v3\ r) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (conv0\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))))) = vol\_conv\ v0\ v1\ v2\ v3) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (h::real)\ a::real.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \wedge (0::real) \leq h \wedge (0::real) < a \wedge a \leq (1::real) \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (frustt\ v0\ v1\ h\ a)\ (wedge\ v0\ v1\ v2\ v3)) = vol\_frustt\_wedge\ v0\ v1\ v2\ v3\ h\ a) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (r::real)\ c::real.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \wedge (0::real) \leq r \wedge -\ (1::real) \leq c \wedge c \leq (1::real) \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ c)\ (wedge\ v0\ v1\ v2\ v3)) = vol\_conic\_cap\_wedge\ v0\ v1\ v2\ v3\ r\ c) \wedge (\forall (a::(real,\ 3)\ cart)\ b::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (rect\ a\ b) = vol\_rect\ a\ b) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \wedge (0::real) \leq r \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (normball\ v0\ r)\ (wedge\ v0\ v1\ v2\ v3)) = vol\_ball\_wedge\ v0\ v1\ v2\ v3\ r)$

thm POLYHEDRON_COLLINEAR_FACES_STRONG:

$\forall (P::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (f'::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (p::(real,\ ?'a::type)\ cart)\ (q::(real,\ ?'a::type)\ cart)\ (s::real)\ t::real.\ polyhedron\ P \wedge IN\ (vec\ (0::nat))\ (relative\_interior\ P) \wedge face\_of\ f\ P \wedge f \neq P \wedge face\_of\ f'\ P \wedge f' \neq P \wedge IN\ p\ f \wedge IN\ q\ f' \wedge (0::real) < s \wedge (0::real) < t \wedge \%\ s\ p = \%\ t\ q \longrightarrow s = t$

thm POLYHEDRON_COLLINEAR_FACES:

$\forall (P::(real, ?'a::type)\ cart \Rightarrow bool)\ (f::(real, ?'a::type)\ cart \Rightarrow bool)\ (f'::(real, ?'a::type)\ cart \Rightarrow bool)\ (p::(real, ?'a::type)\ cart)\ (q::(real, ?'a::type)\ cart)\ (s::real)\ t::real.\ polyhedron\ P \wedge IN\ (vec\ (0::nat))\ (interior\ P) \wedge face\_of\ f\ P \wedge f \neq P \wedge face\_of\ f'\ P \wedge f' \neq P \wedge IN\ p\ f \wedge IN\ q\ f' \wedge (0::real) < s \wedge (0::real) < t \wedge \%\ s\ p = \%\ t\ q \longrightarrow s = t$

thm DEF_vertices:

$vertices = (\lambda\_1960423::(real, ?'a::type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2716::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2716\ (extreme\_point\_of\ x\ \_1960423)\ x))$

thm vertices:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ vertices\ s = GSPEC\ (\lambda GEN\%PVAR\%2716::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2716\ (extreme\_point\_of\ x\ s)\ x)$

thm DEF_edges:

$edges = (\lambda\_1960428::(real, ?'a::type)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2717::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2717\ (edge\_of\ (closed\_segment\ [(v, w)])\ \_1960428)\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm edges:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ edges\ s = GSPEC\ (\lambda GEN\%PVAR\%2717::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2717\ (edge\_of\ (closed\_segment\ [(v, w)])\ s)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm VERTICES_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ vertices\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (vector\_add\ a)\ (vertices\ s)$

thm VERTICES_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow vertices\ (IMAGE\ f\ s) = IMAGE\ f\ (vertices\ s)$

thm EDGES_TRANSLATION:

$\forall (a::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ edges\ (IMAGE\ (vector\_add\ a)\ s) = IMAGE\ (IMAGE\ (vector\_add\ a))\ (edges\ s)$

thm EDGES_LINEAR_IMAGE:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ s::(real, ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow edges\ (IMAGE\ f\ s) = IMAGE\ (IMAGE\ f)\ (edges\ s)$

thm DEF_ineq:

ineq = (*SOME ineq*::*nat* ⇒ (*real* × *real* × *real*) *list* ⇒ *bool* ⇒ *bool*. ∀ *_1963304*::*nat*. (∀ *c*::*bool*. *ineq _1963304* [] *c* = *c*) ∧ (∀ (*a*::*real*) (*x*::*real*) (*b*::*real*) (*xs*::(*real* × *real* × *real*) *list*) *c*::*bool*. *ineq _1963304* ((*a*, *x*, *b*) # *xs*) *c* = (*a* ≤ *x* ∧ *x* ≤ *b* ⟶ *ineq _1963304 xs c*))) (*60*::*nat*)

thm Sphere.ineq:

(∀ *c*::*bool*. *ineq* [] *c* = *c*) ∧ (∀ (*a*::*real*) (*x*::*real*) (*b*::*real*) (*xs*::(*real* × *real* × *real*) *list*) *c*::*bool*. *ineq* ((*a*, *x*, *b*) # *xs*) *c* = (*a* ≤ *x* ∧ *x* ≤ *b* ⟶ *ineq xs c*))

thm Sphere.SDIFF:

∀ (*Y*::*?'a*::*type* ⇒ *bool*) *X*::*?'a*::*type* ⇒ *bool*. *SDIFF X Y* = *HOL_Light_Import.UNION* (*DIFF X Y*) (*DIFF Y X*)

thm DEF_atn2:

atn2 = (λ*_1963305*::*real* × *real*. *if* |*snd _1963305*| < *fst _1963305 then atn* (*snd _1963305* / *fst _1963305*) *else if* (*0*::*real*) < *snd _1963305 then pi* / *real_of_nat* (*2*::*nat*) − *atn* (*fst _1963305* / *snd _1963305*) *else if snd _1963305* < (*0*::*real*) *then* − (*pi* / *real_of_nat* (*2*::*nat*)) − *atn* (*fst _1963305* / *snd _1963305*) *else pi*)

thm Trigonometry2.atn2:

∀ (*x*::*real*) *y*::*real*. *atn2* (*x*, *y*) = (*if* |*y*| < *x then atn* (*y* / *x*) *else if* (*0*::*real*) < *y then pi* / *real_of_nat* (*2*::*nat*) − *atn* (*x* / *y*) *else if y* < (*0*::*real*) *then* − (*pi* / *real_of_nat* (*2*::*nat*)) − *atn* (*x* / *y*) *else pi*)

thm DEF_abc_of_quadratic:

abc_of_quadratic = (λ*_1963314*::*real* ⇒ *real*. *LET* (λ*c*::*real*. *LET_END* (*LET* (λ*p*::*real*. *LET_END* (*LET* (λ*n*::*real*. *LET_END* ((*p* + *n*) / *real_of_nat* (*2*::*nat*) − *c*, (*p* − *n*) / *real_of_nat* (*2*::*nat*), *c*)) (*_1963314* (− (*1*::*real*))))) (*_1963314* (*1*::*real*)))) (*_1963314* (*0*::*real*)))

thm Sphere.abc_of_quadratic:

∀ *f*::*real* ⇒ *real*. *abc_of_quadratic f* = *LET* (λ*c*::*real*. *LET_END* (*LET* (λ*p*::*real*. *LET_END* (*LET* (λ*n*::*real*. *LET_END* ((*p* + *n*) / *real_of_nat* (*2*::*nat*) − *c*, (*p* − *n*) / *real_of_nat* (*2*::*nat*), *c*)) (*f* (− (*1*::*real*))))) (*f* (*1*::*real*)))) (*f* (*0*::*real*))

thm DEF_quadratic_root_plus:

quadratic_root_plus = (λ*_1963319*::*real* × *real* × *real*. (− *fst* (*snd _1963319*) + *sqrt* ((*fst* (*snd _1963319*))^2 − *real_of_nat* (*4*::*nat*) ∗ (*fst _1963319* ∗ *snd* (*snd _1963319*)))) / (*real_of_nat* (*2*::*nat*) ∗ *fst _1963319*))

thm Sphere.quadratic_root_plus:

∀ (*b*::*real*) (*c*::*real*) *a*::*real*. *quadratic_root_plus* (*a*, *b*, *c*) = (− *b* + *sqrt* (*b*^{2} − *real_of_nat* (*4*::*nat*) ∗ (*a* ∗ *c*))) / (*real_of_nat* (*2*::*nat*) ∗ *a*)

thm Sphere.sqrt8:

$sqrt8 = sqrt\ (real\_of\_nat\ (8::nat))$

thm Sphere.sqrt2:

$sqrt2 = sqrt\ (real\_of\_nat\ (2::nat))$

thm Sphere.sqrt3:

$sqrt3 = sqrt\ (real\_of\_nat\ (3::nat))$

thm Sphere.pi_rt18:

$pi\_rt18 = pi\ /\ sqrt\ (real\_of\_nat\ (18::nat))$

thm DEF_delta_y:

$delta\_y = (\lambda(\_1963332::real)\ (\_1963333::real)\ (\_1963334::real)\ (\_1963335::real)\ (\_1963336::real)\ \_1963337::real.\ delta\_x\ (\_1963332 * \_1963332)\ (\_1963333 * \_1963333)\ (\_1963334 * \_1963334)\ (\_1963335 * \_1963335)\ (\_1963336 * \_1963336)\ (\_1963337 * \_1963337))$

thm Sphere.delta_y:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ delta\_y\ y1\ y2\ y3\ y4\ y5\ y6 = delta\_x\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm DEF_edge_flat:

$edge\_flat = (\lambda(\_1963392::real)\ (\_1963393::real)\ (\_1963394::real)\ (\_1963395::real)\ \_1963396::real.\ sqrt\ (quadratic\_root\_plus\ (abc\_of\_quadratic\ (\lambda x4::real.\ -\ delta\_x\ (\_1963392 * \_1963392)\ (\_1963393 * \_1963393)\ (\_1963394 * \_1963394)\ x4\ (\_1963395 * \_1963395)\ (\_1963396 * \_1963396)))))$

thm Sphere.edge_flat:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y5::real)\ y6::real.\ edge\_flat\ y1\ y2\ y3\ y5\ y6 = sqrt\ (quadratic\_root\_plus\ (abc\_of\_quadratic\ (\lambda x4::real.\ -\ delta\_x\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ x4\ (y5 * y5)\ (y6 * y6))))$

thm DEF_edge_flat2_x:

$edge\_flat2\_x = (\lambda(\_1963437::real)\ (\_1963438::real)\ (\_1963439::real)\ (\_1963440::?'a::type)\ (\_1963441::real)\ \_1963442::real.\ (edge\_flat\ (sqrt\ \_1963437)\ (sqrt\ \_1963438)\ (sqrt\ \_1963439)\ (sqrt\ \_1963441)\ (sqrt\ \_1963442))^2)$

thm Sphere.edge_flat2_x:

$\forall (x4::?'a::type)\ (x1::real)\ (x2::real)\ (x3::real)\ (x5::real)\ x6::real.\ edge\_flat2\_x\ x1\ x2\ x3\ x4\ x5\ x6 = (edge\_flat\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x5)\ (sqrt\ x6))^2$

thm DEF_edge_flat_x:

$edge\_flat\_x = (\lambda(\_1963497::real)\ (\_1963498::real)\ (\_1963499::real)\ (\_1963500::real)\ (\_1963501::real)\ \_1963502::real.\ edge\_flat\ (sqrt\ \_1963497)\ (sqrt\ \_1963498)\ (sqrt\ \_1963499)\ (sqrt\ \_1963501)\ (sqrt\ \_1963502))$

thm Sphere.edge_flat_x:

∀ (*x4*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x5*::*real*) *x6*::*real*. *edge_flat_x x1 x2 x3 x4 x5 x6* = *edge_flat* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x5*) (*sqrt x6*)

thm DEF_delta_x4:

delta_x4 = (λ(*_1963557*::*real*) (*_1963558*::*real*) (*_1963559*::*real*) (*_1963560*::*real*) (*_1963561*::*real*) *_1963562*::*real*. − *_1963558* ∗ *_1963559* − *_1963557* ∗ *_1963560* + (*_1963558* ∗ *_1963561* + (*_1963559* ∗ *_1963562* − *_1963561* ∗ *_1963562* + *_1963557* ∗ (− *_1963557* + (*_1963558* + (*_1963559* − *_1963560* + (*_1963561* + *_1963562*)))))))

thm Trigonometry2.delta_x4:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *delta_x4 x1 x2 x3 x4 x5 x6* = − *x2* ∗ *x3* − *x1* ∗ *x4* + (*x2* ∗ *x5* + (*x3* ∗ *x6* − *x5* ∗ *x6* + *x1* ∗ (− *x1* + (*x2* + (*x3* − *x4* + (*x5* + *x6*))))))

thm DEF_delta_x6:

delta_x6 = (λ(*_1963617*::*real*) (*_1963618*::*real*) (*_1963619*::*real*) (*_1963620*::*real*) (*_1963621*::*real*) *_1963622*::*real*. − *_1963617* ∗ *_1963618* − *_1963619* ∗ *_1963622* + (*_1963617* ∗ *_1963620* + (*_1963618* ∗ *_1963621* − *_1963620* ∗ *_1963621* + *_1963619* ∗ (− *_1963619* + (*_1963617* + (*_1963618* − *_1963622* + (*_1963620* + *_1963621*)))))))

thm Sphere.delta_x6:

∀ (*x3*::*real*) (*x1*::*real*) (*x2*::*real*) (*x6*::*real*) (*x4*::*real*) *x5*::*real*. *delta_x6 x1 x2 x3 x4 x5 x6* = − *x1* ∗ *x2* − *x3* ∗ *x6* + (*x1* ∗ *x4* + (*x2* ∗ *x5* − *x4* ∗ *x5* + *x3* ∗ (− *x3* + (*x1* + (*x2* − *x6* + (*x4* + *x5*))))))

thm DEF_ups_x:

ups_x = (λ(*_1963677*::*real*) (*_1963678*::*real*) *_1963679*::*real*. − *_1963677* ∗ *_1963677* − *_1963678* ∗ *_1963678* − *_1963679* ∗ *_1963679* + (*real_of_nat* (*2*::*nat*) ∗ (*_1963677* ∗ *_1963679*) + (*real_of_nat* (*2*::*nat*) ∗ (*_1963677* ∗ *_1963678*) + *real_of_nat* (*2*::*nat*) ∗ (*_1963678* ∗ *_1963679*))))

thm Trigonometry1.ups_x:

∀ (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *ups_x x1 x2 x6* = − *x1* ∗ *x1* − *x2* ∗ *x2* − *x6* ∗ *x6* + (*real_of_nat* (*2*::*nat*) ∗ (*x1* ∗ *x6*) + (*real_of_nat* (*2*::*nat*) ∗ (*x1* ∗ *x2*) + *real_of_nat* (*2*::*nat*) ∗ (*x2* ∗ *x6*)))

thm DEF_eta_x:

eta_x = (λ(*_1963698*::*real*) (*_1963699*::*real*) *_1963700*::*real*. *sqrt* (*_1963698* ∗ (*_1963699* ∗ *_1963700*) / *ups_x _1963698 _1963699 _1963700*))

thm Sphere.eta_x:

∀ (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *eta_x x1 x2 x3* = *sqrt* (*x1* ∗ (*x2* ∗ *x3*) / *ups_x x1 x2 x3*)

thm DEF_eta_y:

eta_y = (λ(_1963719::real) (_1963720::real) _1963721::real. LET (λx1::real. LET_END (LET (λx2::real. LET_END (LET (λx3::real. LET_END (eta_x x1 x2 x3)) (_1963721 ∗ _1963721))) (_1963720 ∗ _1963720))) (_1963719 ∗ _1963719))

thm Sphere.eta_y:

∀(y3::real) (y2::real) y1::real. eta_y y1 y2 y3 = LET (λx1::real. LET_END (LET (λx2::real. LET_END (LET (λx3::real. LET_END (eta_x x1 x2 x3)) (y3 ∗ y3))) (y2 ∗ y2))) (y1 ∗ y1)

thm DEF_rho_x:

rho_x = (λ(_1963740::real) (_1963741::real) (_1963742::real) (_1963743::real) (_1963744::real) _1963745::real. − _1963740 ∗ (_1963740 ∗ (_1963743 ∗ _1963743)) − _1963741 ∗ (_1963741 ∗ (_1963744 ∗ _1963744)) − _1963742 ∗ (_1963742 ∗ (_1963745 ∗ _1963745)) + (real_of_nat (2::nat) ∗ (_1963740 ∗ (_1963741 ∗ (_1963743 ∗ _1963744))) + (real_of_nat (2::nat) ∗ (_1963740 ∗ (_1963742 ∗ (_1963743 ∗ _1963745))) + real_of_nat (2::nat) ∗ (_1963741 ∗ (_1963742 ∗ (_1963744 ∗ _1963745))))))

thm Sphere.rho_x:

∀(x1::real) (x4::real) (x2::real) (x3::real) (x5::real) x6::real. rho_x x1 x2 x3 x4 x5 x6 = − x1 ∗ (x1 ∗ (x4 ∗ x4)) − x2 ∗ (x2 ∗ (x5 ∗ x5)) − x3 ∗ (x3 ∗ (x6 ∗ x6)) + (real_of_nat (2::nat) ∗ (x1 ∗ (x2 ∗ (x4 ∗ x5))) + (real_of_nat (2::nat) ∗ (x1 ∗ (x3 ∗ (x4 ∗ x6))) + real_of_nat (2::nat) ∗ (x2 ∗ (x3 ∗ (x5 ∗ x6)))))

thm DEF_chi_x:

chi_x = (λ(_1963800::real) (_1963801::real) (_1963802::real) (_1963803::real) (_1963804::real) _1963805::real. − (_1963800 ∗ (_1963803 ∗ _1963803)) + (_1963800 ∗ (_1963803 ∗ _1963804) + (_1963801 ∗ (_1963803 ∗ _1963804) − _1963801 ∗ (_1963804 ∗ _1963804) + (_1963800 ∗ (_1963803 ∗ _1963805) + (_1963802 ∗ (_1963803 ∗ _1963805) + (_1963801 ∗ (_1963804 ∗ _1963805) + (_1963802 ∗ (_1963804 ∗ _1963805) − real_of_nat (2::nat) ∗ (_1963803 ∗ (_1963804 ∗ _1963805)) − _1963802 ∗ (_1963805 ∗ _1963805))))))))

thm Sphere.chi_x:

∀(x1::real) (x2::real) (x4::real) (x5::real) (x3::real) x6::real. chi_x x1 x2 x3 x4 x5 x6 = − (x1 ∗ (x4 ∗ x4)) + (x1 ∗ (x4 ∗ x5) + (x2 ∗ (x4 ∗ x5) − x2 ∗ (x5 ∗ x5) + (x1 ∗ (x4 ∗ x6) + (x3 ∗ (x4 ∗ x6) + (x2 ∗ (x5 ∗ x6) + (x3 ∗ (x5 ∗ x6) − real_of_nat (2::nat) ∗ (x4 ∗ (x5 ∗ x6)) − x3 ∗ (x6 ∗ x6)))))))

thm DEF_dih_x:

dih_x = (λ(_1963860::real) (_1963861::real) (_1963862::real) (_1963863::real) (_1963864::real) _1963865::real. LET (λd_x4::real. LET_END (LET (λd::real. LET_END (pi / real_of_nat (2::nat) + atn2 (sqrt (real_of_nat (4::nat) ∗ (_1963860 ∗ d)), − d_x4))) (delta_x _1963860 _1963861 _1963862 _1963863

_1963864 _1963865))) (*delta_x4 _1963860 _1963861 _1963862 _1963863 _1963864 _1963865*))

thm Sphere.dih_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih_x x1 x2 x3 x4 x5 x6* = *LET* (λ*d_x4*::*real*. *LET_END* (*LET* (λ*d*::*real*. *LET_END* (*pi* / *real_of_nat* (*2*::*nat*) + *atn2* (*sqrt* (*real_of_nat* (*4*::*nat*) ∗ (*x1* ∗ *d*)), − *d_x4*))) (*delta_x x1 x2 x3 x4 x5 x6*))) (*delta_x4 x1 x2 x3 x4 x5 x6*)

thm DEF_dih_y:

dih_y = (λ(*_1963920*::*real*) (*_1963921*::*real*) (*_1963922*::*real*) (*_1963923*::*real*) (*_1963924*::*real*) *_1963925*::*real*. *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real*. ∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *GEQ* (*f* (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*)) (*LET_END* (*dih_x x1 x2 x3 x4 x5 x6*)))) (*_1963920* ∗ *_1963920*, *_1963921* ∗ *_1963921*, *_1963922* ∗ *_1963922*, *_1963923* ∗ *_1963923*, *_1963924* ∗ *_1963924*, *_1963925* ∗ *_1963925*))

thm Sphere.dih_y:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *dih_y y1 y2 y3 y4 y5 y6* = *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real*. ∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *GEQ* (*f* (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*)) (*LET_END* (*dih_x x1 x2 x3 x4 x5 x6*)))) (*y1* ∗ *y1*, *y2* ∗ *y2*, *y3* ∗ *y3*, *y4* ∗ *y4*, *y5* ∗ *y5*, *y6* ∗ *y6*)

thm DEF_dih2_y:

dih2_y = (λ(*_1963980*::*real*) (*_1963981*::*real*) (*_1963982*::*real*) (*_1963983*::*real*) *_1963984*::*real*. *dih_y _1963981 _1963980 _1963982 _1963984 _1963983*)

thm Sphere.dih2_y:

∀ (*y2*::*real*) (*y1*::*real*) (*y3*::*real*) (*y5*::*real*) (*y4*::*real*) *y6*::*real*. *dih2_y y1 y2 y3 y4 y5 y6* = *dih_y y2 y1 y3 y5 y4 y6*

thm DEF_dih3_y:

dih3_y = (λ(*_1964040*::*real*) (*_1964041*::*real*) (*_1964042*::*real*) (*_1964043*::*real*) (*_1964044*::*real*) *_1964045*::*real*. *dih_y _1964042 _1964040 _1964041 _1964045 _1964043 _1964044*)

thm Sphere.dih3_y:

∀ (*y3*::*real*) (*y1*::*real*) (*y2*::*real*) (*y6*::*real*) (*y4*::*real*) *y5*::*real*. *dih3_y y1 y2 y3 y4 y5 y6* = *dih_y y3 y1 y2 y6 y4 y5*

thm DEF_dih2_x:

dih2_x = (λ(*_1964100*::*real*) (*_1964101*::*real*) (*_1964102*::*real*) (*_1964103*::*real*) *_1964104*::*real*. *dih_x _1964101 _1964100 _1964102 _1964104 _1964103*)

thm Sphere.dih2_x:

$\forall (x2::real)\ (x1::real)\ (x3::real)\ (x5::real)\ (x4::real)\ x6::real.\ dih2\_x\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih\_x\ x2\ x1\ x3\ x5\ x4\ x6$

thm DEF_dih3_x:

$dih3\_x = (\lambda(\_1964160::real)\ (\_1964161::real)\ (\_1964162::real)\ (\_1964163::real)\ (\_1964164::real)\ \_1964165::real.\ dih\_x\ \_1964162\ \_1964160\ \_1964161\ \_1964165\ \_1964163\ \_1964164)$

thm Sphere.dih3_x:

$\forall (x3::real)\ (x1::real)\ (x2::real)\ (x6::real)\ (x4::real)\ x5::real.\ dih3\_x\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih\_x\ x3\ x1\ x2\ x6\ x4\ x5$

thm DEF_sol_x:

$sol\_x = (\lambda(\_1964220::real)\ (\_1964221::real)\ (\_1964222::real)\ (\_1964223::real)\ (\_1964224::real)\ \_1964225::real.\ dih\_x\ \_1964220\ \_1964221\ \_1964222\ \_1964223\ \_1964224\ \_1964225 + (dih\_x\ \_1964221\ \_1964222\ \_1964220\ \_1964224\ \_1964225\ \_1964223 + (dih\_x\ \_1964222\ \_1964220\ \_1964221\ \_1964225\ \_1964223\ \_1964224\ -\ pi)))$

thm Sphere.sol_x:

$\forall (x3::real)\ (x1::real)\ (x2::real)\ (x6::real)\ (x4::real)\ x5::real.\ sol\_x\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih\_x\ x1\ x2\ x3\ x4\ x5\ x6\ +\ (dih\_x\ x2\ x3\ x1\ x5\ x6\ x4\ +\ (dih\_x\ x3\ x1\ x2\ x6\ x4\ x5\ -\ pi))$

thm DEF_sol_y:

$sol\_y = (\lambda(\_1964280::real)\ (\_1964281::real)\ (\_1964282::real)\ (\_1964283::real)\ (\_1964284::real)\ \_1964285::real.\ dih\_y\ \_1964280\ \_1964281\ \_1964282\ \_1964283\ \_1964284\ \_1964285 + (dih\_y\ \_1964281\ \_1964282\ \_1964280\ \_1964284\ \_1964285\ \_1964283 + (dih\_y\ \_1964282\ \_1964280\ \_1964281\ \_1964285\ \_1964283\ \_1964284\ -\ pi)))$

thm Sphere.sol_y:

$\forall (y3::real)\ (y1::real)\ (y2::real)\ (y6::real)\ (y4::real)\ y5::real.\ sol\_y\ y1\ y2\ y3\ y4\ y5\ y6\ =\ dih\_y\ y1\ y2\ y3\ y4\ y5\ y6\ +\ (dih\_y\ y2\ y3\ y1\ y5\ y6\ y4\ +\ (dih\_y\ y3\ y1\ y2\ y6\ y4\ y5\ -\ pi))$

thm DEF_interp:

$interp = (\lambda(\_1964340::real)\ (\_1964341::real)\ (\_1964342::real)\ (\_1964343::real)\ \_1964344::real.\ \_1964341\ +\ (\_1964344\ -\ \_1964340) * ((\_1964343\ -\ \_1964341)\ /\ (\_1964342\ -\ \_1964340)))$

thm Sphere.interp:

$\forall (x::real)\ (y2::real)\ (y1::real)\ (x2::real)\ x1::real.\ interp\ x1\ y1\ x2\ y2\ x\ =\ y1\ +\ (x\ -\ x1) * ((y2\ -\ y1)\ /\ (x2\ -\ x1))$

thm Sphere.const1:

const1 = *sol_y* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) / *pi*

thm DEF_ly:

ly = *interp* (*real_of_nat* (*2*::*nat*)) (*1*::*real*) (*DECIMAL* (*252*::*nat*) (*100*::*nat*)) (*0*::*real*)

thm Sphere.ly:

∀ *y*::*real*. *ly y* = *interp* (*real_of_nat* (*2*::*nat*)) (*1*::*real*) (*DECIMAL* (*252*::*nat*) (*100*::*nat*)) (*0*::*real*) *y*

thm DEF_rho:

rho = (λ*_1964390*::*real*. (*1*::*real*) + (*const1* − *const1* ∗ *ly _1964390*))

thm Sphere.rho:

∀ *y*::*real*. *rho y* = (*1*::*real*) + (*const1* − *const1* ∗ *ly y*)

thm DEF_rhazim:

rhazim = (λ(*_1964395*::*real*) (*_1964396*::*real*) (*_1964397*::*real*) (*_1964398*::*real*) (*_1964399*::*real*) *_1964400*::*real*. *rho _1964395* ∗ *dih_y _1964395 _1964396 _1964397 _1964398 _1964399 _1964400*)

thm Sphere.rhazim:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *rhazim y1 y2 y3 y4 y5 y6* = *rho y1* ∗ *dih_y y1 y2 y3 y4 y5 y6*

thm DEF_lnazim:

lnazim = (λ(*_1964455*::*real*) (*_1964456*::*real*) (*_1964457*::*real*) (*_1964458*::*real*) (*_1964459*::*real*) *_1964460*::*real*. *ly _1964455* ∗ *dih_y _1964455 _1964456 _1964457 _1964458 _1964459 _1964460*)

thm Sphere.lnazim:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *lnazim y1 y2 y3 y4 y5 y6* = *ly y1* ∗ *dih_y y1 y2 y3 y4 y5 y6*

thm DEF_taum:

taum = (λ(*_1964515*::*real*) (*_1964516*::*real*) (*_1964517*::*real*) (*_1964518*::*real*) (*_1964519*::*real*) *_1964520*::*real*. *sol_y _1964515 _1964516 _1964517 _1964518 _1964519 _1964520* ∗ ((*1*::*real*) + *const1*) − *const1* ∗ (*lnazim _1964515 _1964516 _1964517 _1964518 _1964519 _1964520* + (*lnazim _1964516 _1964517 _1964515 _1964519 _1964520 _1964518* + *lnazim _1964517 _1964515 _1964516 _1964520 _1964518 _1964519*)))

thm Sphere.taum:

∀ (*y3*::*real*) (*y1*::*real*) (*y2*::*real*) (*y6*::*real*) (*y4*::*real*) *y5*::*real*. *taum y1 y2 y3 y4 y5 y6* = *sol_y y1 y2 y3 y4 y5 y6* ∗ ((*1*::*real*) + *const1*) − *const1* ∗ (*lnazim y1 y2 y3 y4 y5 y6* + (*lnazim y2 y3 y1 y5 y6 y4* + *lnazim y3 y1 y2 y6 y4 y5*))

thm DEF_tauV:

$tauV = (\lambda(\_1964575::(real, 3)\ cart)\ (\_1964576::(real, 3)\ cart)\ \_1964577::(real, 3)\ cart.\ taum\ (vector\_norm\ \_1964575)\ (vector\_norm\ \_1964576)\ (vector\_norm\ \_1964577)\ (distance\ (\_1964576,\ \_1964577))\ (distance\ (\_1964575,\ \_1964577))\ (distance\ (\_1964575,\ \_1964576)))$

thm Sphere.tauV:

$\forall\,(v3::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ v2::(real,\ 3)\ cart.\ tauV\ v1\ v2\ v3 = taum\ (vector\_norm\ v1)\ (vector\_norm\ v2)\ (vector\_norm\ v3)\ (distance\ (v2,\ v3))\ (distance\ (v1,\ v3))\ (distance\ (v1,\ v2))$

thm DEF_node2_y:

$node2\_y = (\lambda(\_1964596::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (\_1964597::?'e::type)\ (\_1964598::?'g::type)\ (\_1964599::?'f::type)\ (\_1964600::?'b::type)\ (\_1964601::?'d::type)\ \_1964602::?'c::type.\ \_1964596\ \_1964598\ \_1964599\ \_1964597\ \_1964601\ \_1964602\ \_1964600)$

thm Sphere.node2_y:

$\forall\,(f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (y2::?'g::type)\ (y3::?'f::type)\ (y1::?'e::type)\ (y5::?'d::type)\ (y6::?'c::type)\ y4::?'b::type.\ node2\_y\ f\ y1\ y2\ y3\ y4\ y5\ y6 = f\ y2\ y3\ y1\ y5\ y6\ y4$

thm DEF_node3_y:

$node3\_y = (\lambda(\_1964673::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (\_1964674::?'f::type)\ (\_1964675::?'e::type)\ (\_1964676::?'g::type)\ (\_1964677::?'c::type)\ (\_1964678::?'b::type)\ \_1964679::?'d::type.\ \_1964673\ \_1964676\ \_1964674\ \_1964675\ \_1964679\ \_1964677\ \_1964678)$

thm Sphere.node3_y:

$\forall\,(f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (y3::?'g::type)\ (y1::?'f::type)\ (y2::?'e::type)\ (y6::?'d::type)\ (y4::?'c::type)\ y5::?'b::type.\ node3\_y\ f\ y1\ y2\ y3\ y4\ y5\ y6 = f\ y3\ y1\ y2\ y6\ y4\ y5$

thm Sphere.rhazim2:

$rhazim2 = node2\_y\ rhazim$

thm Sphere.rhazim3:

$rhazim3 = node3\_y\ rhazim$

thm DEF_dih4_y:

$dih4\_y = (SOME\ dih4\_y::nat \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ \forall\,(\_1965191::nat)\ (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ dih4\_y\ \_1965191\ y1\ y2\ y3\ y4\ y5\ y6 = dih\_y\ y4\ y2\ y6\ y1\ y5\ y3)\ (61::nat)$

thm Sphere.dih4_y:

$dih4\_y$ ($?y1.0$::$real$) ($?y2.0$::$real$) ($?y3.0$::$real$) ($?y4.0$::$real$) ($?y5.0$::$real$) ($?y6.0$::$real$) = dih_y $?y4.0$ $?y2.0$ $?y6.0$ $?y1.0$ $?y5.0$ $?y3.0$

thm DEF_dih5_y:

$dih5\_y$ = ($SOME$ $dih5\_y$::nat ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$. ∀ ($\_1965633$::nat) ($y1$::$real$) ($y2$::$real$) ($y3$::$real$) ($y4$::$real$) ($y5$::$real$) $y6$::$real$. $dih5\_y$ $\_1965633$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$ = dih_y $y5$ $y1$ $y6$ $y2$ $y4$ $y3$) (62::nat)

thm Sphere.dih5_y:

$dih5\_y$ ($?y1.0$::$real$) ($?y2.0$::$real$) ($?y3.0$::$real$) ($?y4.0$::$real$) ($?y5.0$::$real$) ($?y6.0$::$real$) = dih_y $?y5.0$ $?y1.0$ $?y6.0$ $?y2.0$ $?y4.0$ $?y3.0$

thm DEF_dih6_y:

$dih6\_y$ = ($SOME$ $dih6\_y$::nat ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$. ∀ ($\_1966075$::nat) ($y1$::$real$) ($y2$::$real$) ($y3$::$real$) ($y4$::$real$) ($y5$::$real$) $y6$::$real$. $dih6\_y$ $\_1966075$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$ = dih_y $y6$ $y1$ $y5$ $y3$ $y4$ $y2$) (63::nat)

thm Sphere.dih6_y:

$dih6\_y$ ($?y1.0$::$real$) ($?y2.0$::$real$) ($?y3.0$::$real$) ($?y4.0$::$real$) ($?y5.0$::$real$) ($?y6.0$::$real$) = dih_y $?y6.0$ $?y1.0$ $?y5.0$ $?y3.0$ $?y4.0$ $?y2.0$

thm DEF_rhazim4:

$rhazim4$ = ($SOME$ $rhazim4$::nat ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$. ∀ ($\_1966517$::nat) ($y1$::$real$) ($y2$::$real$) ($y3$::$real$) ($y4$::$real$) ($y5$::$real$) $y6$::$real$. $rhazim4$ $\_1966517$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$ = rho $y4$ ∗ $dih4\_y$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$) (64::nat)

thm Sphere.rhazim4:

$rhazim4$ ($?y1.0$::$real$) ($?y2.0$::$real$) ($?y3.0$::$real$) ($?y4.0$::$real$) ($?y5.0$::$real$) ($?y6.0$::$real$) = rho $?y4.0$ ∗ $dih4\_y$ $?y1.0$ $?y2.0$ $?y3.0$ $?y4.0$ $?y5.0$ $?y6.0$

thm DEF_rhazim5:

$rhazim5$ = ($SOME$ $rhazim5$::nat ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$. ∀ ($\_1966959$::nat) ($y1$::$real$) ($y2$::$real$) ($y3$::$real$) ($y4$::$real$) ($y5$::$real$) $y6$::$real$. $rhazim5$ $\_1966959$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$ = rho $y5$ ∗ $dih5\_y$ $y1$ $y2$ $y3$ $y4$ $y5$ $y6$) (65::nat)

thm Sphere.rhazim5:

$rhazim5$ ($?y1.0$::$real$) ($?y2.0$::$real$) ($?y3.0$::$real$) ($?y4.0$::$real$) ($?y5.0$::$real$) ($?y6.0$::$real$) = rho $?y5.0$ ∗ $dih5\_y$ $?y1.0$ $?y2.0$ $?y3.0$ $?y4.0$ $?y5.0$ $?y6.0$

thm DEF_rhazim6:

$rhazim6$ = ($SOME$ $rhazim6$::nat ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$ ⇒ $real$. ∀ ($\_1967401$::nat) ($y1$::$real$) ($y2$::$real$) ($y3$::$real$) ($y4$::$real$) ($y5$::$real$)

*y6::real. rhazim6 _1967401 y1 y2 y3 y4 y5 y6 = rho y6 * dih6_y y1 y2 y3 y4 y5 y6) (66::nat)*

thm Sphere.rhazim6:

*rhazim6 (?y1.0::real) (?y2.0::real) (?y3.0::real) (?y4.0::real) (?y5.0::real) (?y6.0::real) = rho ?y6.0 * dih6_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0*

thm DEF_tauq:

tauq = (λ(_1967402::real) (_1967403::real) (_1967404::real) (_1967405::real) (_1967406::real) (_1967407::real) (_1967408::real) (_1967409::real) _1967410::real. taum _1967402 _1967403 _1967404 _1967405 _1967406 _1967407 + taum _1967408 _1967403 _1967404 _1967405 _1967409 _1967410)

thm Sphere.tauq:

∀ *(y1::real) (y5::real) (y6::real) (y7::real) (y2::real) (y3::real) (y4::real) (y8::real) y9::real. tauq y1 y2 y3 y4 y5 y6 y7 y8 y9 = taum y1 y2 y3 y4 y5 y6 + taum y7 y2 y3 y4 y8 y9*

thm DEF_vol_x:

vol_x = (λ(_1967519::real) (_1967520::real) (_1967521::real) (_1967522::real) (_1967523::real) _1967524::real. sqrt (delta_x _1967519 _1967520 _1967521 _1967522 _1967523 _1967524) / real_of_nat (12::nat))

thm Sphere.vol_x:

∀ *(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. vol_x x1 x2 x3 x4 x5 x6 = sqrt (delta_x x1 x2 x3 x4 x5 x6) / real_of_nat (12::nat)*

thm DEF_arclength:

*arclength = (λ(_1967579::real) (_1967580::real) _1967581::real. pi / real_of_nat (2::nat) + atn2 (sqrt (ups_x (_1967579 * _1967579) (_1967580 * _1967580) (_1967581 * _1967581)), _1967581 * _1967581 − _1967579 * _1967579 − _1967580 * _1967580))*

thm Sphere.arclength:

∀ *(c::real) (a::real) b::real. arclength a b c = pi / real_of_nat (2::nat) + atn2 (sqrt (ups_x (a * a) (b * b) (c * c)), c * c − a * a − b * b)*

thm DEF_volR:

*volR = (λ(_1967600::real) (_1967601::real) _1967602::real. sqrt (_1967600 * (_1967600 * ((_1967601 * _1967601 − _1967600 * _1967600) * (_1967602 * _1967602 − _1967601 * _1967601)))) / real_of_nat (6::nat))*

thm Sphere.volR:

∀ *(a::real) (c::real) b::real. volR a b c = sqrt (a * (a * ((b * b − a * a) * (c * c − b * b)))) / real_of_nat (6::nat)*

thm DEF_solR:

$solR = (\lambda(\_1967621::real)\ (\_1967622::real)\ \_1967623::real.\ real\_of\_nat\ (2::nat) * atn2\ (sqrt\ ((\_1967623 + \_1967622) * (\_1967622 + \_1967621)),\ sqrt\ ((\_1967623 - \_1967622) * (\_1967622 - \_1967621))))$

thm Sphere.solR:

$\forall (c::real)\ (b::real)\ a::real.\ solR\ a\ b\ c = real\_of\_nat\ (2::nat) * atn2\ (sqrt\ ((c + b) * (b + a)),\ sqrt\ ((c - b) * (b - a)))$

thm DEF_dihR:

$dihR = (\lambda(\_1967642::real)\ (\_1967643::real)\ \_1967644::real.\ atn2\ (sqrt\ (\_1967643 * \_1967643 - \_1967642 * \_1967642),\ sqrt\ (\_1967644 * \_1967644 - \_1967643 * \_1967643)))$

thm Sphere.dihR:

$\forall (a::real)\ (c::real)\ b::real.\ dihR\ a\ b\ c = atn2\ (sqrt\ (b * b - a * a),\ sqrt\ (c * c - b * b))$

thm DEF_rad2_x:

$rad2\_x = (\lambda(\_1967663::real)\ (\_1967664::real)\ (\_1967665::real)\ (\_1967666::real)\ (\_1967667::real)\ \_1967668::real.\ rho\_x\ \_1967663\ \_1967664\ \_1967665\ \_1967666\ \_1967667\ \_1967668\ /\ (delta\_x\ \_1967663\ \_1967664\ \_1967665\ \_1967666\ \_1967667\ \_1967668 * real\_of\_nat\ (4::nat)))$

thm Sphere.rad2_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ rad2\_x\ x1\ x2\ x3\ x4\ x5\ x6 = rho\_x\ x1\ x2\ x3\ x4\ x5\ x6\ /\ (delta\_x\ x1\ x2\ x3\ x4\ x5\ x6 * real\_of\_nat\ (4::nat))$

thm Trigonometry.BYFLKYM1:

$aff = hull\ affine$

thm Trigonometry.BYFLKYM3:

$aff\_gt = affsign\ sgn\_gt$

thm Trigonometry.BYFLKYM2:

$aff\_ge = affsign\ sgn\_ge$

thm Trigonometry.BYFLKYM4:

$aff\_lt = affsign\ sgn\_lt$

thm Sphere.aff_le_def:

$aff\_le = affsign\ sgn\_le$

thm DEF_voronoi_open:

$voronoi\_open = (\lambda(\_1967723::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1967724::(real,\ ?'a::type)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%0::(real,\ ?'a::type)\ cart.\ \exists x::(real,$

$?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%0\ (\forall\ w::(real,\ ?'a::type)\ cart.\ \_1967723\ w \land w \neq \_1967724 \longrightarrow distance\ (x,\ \_1967724) < distance\ (x,\ w))\ x))$

thm Geomdetail.voronoi_open:

$\forall\,(S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_open\ S\ v = GSPEC\ (\lambda GEN\%PVAR\%0::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%0\ (\forall\, w::(real,\ ?'a::type)\ cart.\ S\ w \land w \neq v \longrightarrow distance\ (x,\ v) < distance\ (x,\ w))\ x)$

thm DEF_voronoi_closed:

$voronoi\_closed = (\lambda(\_1967735::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_1967736::(real,\ ?'a::type)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%1::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1\ (\forall\, w::(real,\ ?'a::type)\ cart.\ \_1967735\ w \longrightarrow distance\ (x,\ \_1967736) \leq distance\ (x,\ w))\ x))$

thm Sphere.voronoi_closed:

$\forall\,(S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_closed\ S\ v = GSPEC\ (\lambda GEN\%PVAR\%1::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1\ (\forall\, w::(real,\ ?'a::type)\ cart.\ S\ w \longrightarrow distance\ (x,\ v) \leq distance\ (x,\ w))\ x)$

thm DEF_voronoi_set:

$voronoi\_set = (\lambda(\_1967747::(real,\ 3)\ cart \Rightarrow bool)\ \_1967748::(real,\ 3)\ cart \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%2::(real,\ 3)\ cart \Rightarrow bool.\ \exists\, v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2\ (IN\ v\ \_1967748)\ (voronoi\_closed\ \_1967747\ v))))$

thm Sphere.VORONOI_SET:

$\forall\,(W::(real,\ 3)\ cart \Rightarrow bool)\ V::(real,\ 3)\ cart \Rightarrow bool.\ voronoi\_set\ V\ W = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%2::(real,\ 3)\ cart \Rightarrow bool.\ \exists\, v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2\ (IN\ v\ W)\ (voronoi\_closed\ V\ v)))$

thm DEF_voronoi_list:

$voronoi\_list = (\lambda(\_1967759::(real,\ 3)\ cart \Rightarrow bool)\ \_1967760::(real,\ 3)\ cart\ list.\ voronoi\_set\ \_1967759\ (set\_of\_list\ \_1967760))$

thm Pack_concl.BBDTRGC_VORONOI_LIST:

$\forall\,(V::(real,\ 3)\ cart \Rightarrow bool)\ wl::(real,\ 3)\ cart\ list.\ voronoi\_list\ V\ wl = voronoi\_set\ V\ (set\_of\_list\ wl)$

thm DEF_voronoi_nondg:

$voronoi\_nondg = (\lambda(\_1967771::(real,\ 3)\ cart \Rightarrow bool)\ \_1967772::(real,\ 3)\ cart\ list.\ length\ \_1967772 < (5::nat) \land SUBSET\ (set\_of\_list\ \_1967772)\ \_1967771 \land aff\_dim\ (voronoi\_list\ \_1967771\ \_1967772) + int\ (length\ \_1967772) = int\ (4::nat))$

thm Sphere.VORONOI_NONDG:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ voronoi\_nondg\ V\ ul = (length\ ul < (5::nat) \wedge SUBSET\ (set\_of\_list\ ul)\ V \wedge aff\_dim\ (voronoi\_list\ V\ ul) + int\ (length\ ul) = int\ (4::nat))$

thm DEF_initial_sublist:

$initial\_sublist = (\lambda(\_1967783::?'a::type\ list)\ \_1967784::?'a::type\ list.\ \exists\ yl::?'a::type\ list.\ \_1967784 = \_1967783\ @\ yl)$

thm Sphere.INITIAL_SUBLIST:

$\forall (zl::?'a::type\ list)\ xl::?'a::type\ list.\ initial\_sublist\ xl\ zl = (\exists\ yl::?'a::type\ list.\ zl = xl\ @\ yl)$

thm DEF_barV:

$barV = (\lambda(\_1967795::(real,\ 3)\ cart \Rightarrow bool)\ (\_1967796::nat)\ \_1967797::(real,\ 3)\ cart\ list.\ length\ \_1967797 = \_1967796 + (1::nat) \wedge (\forall\ vl::(real,\ 3)\ cart\ list.\ initial\_sublist\ vl\ \_1967797 \wedge (0::nat) < length\ vl \longrightarrow voronoi\_nondg\ \_1967795\ vl))$

thm Pack_concl.NOPZSEH:

$\forall (k::nat)\ (ul::(real,\ 3)\ cart\ list)\ V::(real,\ 3)\ cart \Rightarrow bool.\ barV\ V\ k\ ul = (length\ ul = k + (1::nat) \wedge (\forall\ vl::(real,\ 3)\ cart\ list.\ initial\_sublist\ vl\ ul \wedge (0::nat) < length\ vl \longrightarrow voronoi\_nondg\ V\ vl))$

thm DEF_truncate_simplex:

$truncate\_simplex = (\lambda(\_1967816::nat)\ \_1967817::?'a::type\ list.\ SOME\ vl::?'a::type\ list.\ length\ vl = \_1967816 + (1::nat) \wedge initial\_sublist\ vl\ \_1967817)$

thm Pack_concl.JNRJQSM:

$\forall (j::nat)\ ul::?'a::type\ list.\ truncate\_simplex\ j\ ul = (SOME\ vl::?'a::type\ list.\ length\ vl = j + (1::nat) \wedge initial\_sublist\ vl\ ul)$

thm DEF_omega_list_n:

$omega\_list\_n = (SOME\ omega\_list\_n::nat \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (real,\ 3)\ cart\ list \Rightarrow nat \Rightarrow (real,\ 3)\ cart.\ \forall\ \_1968599::nat.\ (\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ omega\_list\_n\ \_1968599\ V\ ul\ (0::nat) = hd\ ul) \wedge (\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ i::nat.\ omega\_list\_n\ \_1968599\ V\ ul\ (Suc\ i) = closest\_point\ (voronoi\_list\ V\ (truncate\_simplex\ (Suc\ i)\ ul))\ (omega\_list\_n\ \_1968599\ V\ ul\ i)))\ (67::nat)$

thm Sphere.OMEGA_LIST_N:

$omega\_list\_n\ (?V::(real,\ 3)\ cart \Rightarrow bool)\ (?ul::(real,\ 3)\ cart\ list)\ (0::nat) = hd\ ?ul \wedge omega\_list\_n\ ?V\ ?ul\ (Suc\ (?i::nat)) = closest\_point\ (voronoi\_list\ ?V\ (truncate\_simplex\ (Suc\ ?i)\ ?ul))\ (omega\_list\_n\ ?V\ ?ul\ ?i)$

thm DEF_omega_list:

$omega\_list = (\lambda(\_1968600::(real,\ 3)\ cart \Rightarrow bool)\ \_1968601::(real,\ 3)\ cart\ list.\ omega\_list\_n\ \_1968600\ \_1968601\ (length\ \_1968601 - (1::nat)))$

thm Sphere.OMEGA_LIST:

$\forall\,(V::(real,\,3)\ cart \Rightarrow bool)\ ul::(real,\,3)\ cart\ list.\ omega\_list\ V\ ul = omega\_list\_n\ V\ ul\ (length\ ul - (1::nat))$

thm DEF_rogers:

$rogers = (\lambda(\_1968612::(real,\,3)\ cart \Rightarrow bool)\ \_1968613::(real,\,3)\ cart\ list.\ hull\ convex\ (IMAGE\ (omega\_list\_n\ \_1968612\ \_1968613)\ (GSPEC\ (\lambda GEN\%PVAR\%3::nat.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%3\ (j < length\ \_1968613)\ j))))$

thm Sphere.ROGERS:

$\forall\,(V::(real,\,3)\ cart \Rightarrow bool)\ ul::(real,\,3)\ cart\ list.\ rogers\ V\ ul = hull\ convex\ (IMAGE\ (omega\_list\_n\ V\ ul)\ (GSPEC\ (\lambda GEN\%PVAR\%3::nat.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%3\ (j < length\ ul)\ j)))$

thm DEF_line:

$line = (\lambda\_1968624::(real,\,?'a::type)\ cart \Rightarrow bool.\ \exists\,(v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ v \neq w \wedge \_1968624 = hull\ affine\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Trigonometry.SWKFLBJ1:

$\forall\,x::(real,\,?'a::type)\ cart \Rightarrow bool.\ line\ x = (\exists\,(v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ v \neq w \wedge x = hull\ affine\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Sphere.plane:

$\forall\,x::(real,\,?'a::type)\ cart \Rightarrow bool.\ plane\ x = (\exists\,(u::(real,\,?'a::type)\ cart)\ (v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge x = hull\ affine\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm DEF_closed_half_plane:

$closed\_half\_plane = (\lambda\_1968629::(real,\,?'a::type)\ cart \Rightarrow bool.\ \exists\,(u::(real,\,?'a::type)\ cart)\ (v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge \_1968629 = aff\_ge\ (INSERT\ u\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY))$

thm Trigonometry.JLWZFBH2:

$\forall\,x::(real,\,?'a::type)\ cart \Rightarrow bool.\ closed\_half\_plane\ x = (\exists\,(u::(real,\,?'a::type)\ cart)\ (v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge x = aff\_ge\ (INSERT\ u\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY))$

thm DEF_open_half_plane:

$open\_half\_plane = (\lambda\_1968634::(real,\,?'a::type)\ cart \Rightarrow bool.\ \exists\,(u::(real,\,?'a::type)\ cart)\ (v::(real,\,?'a::type)\ cart)\ w::(real,\,?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge \_1968634 = aff\_gt\ (INSERT\ u\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY))$

thm Sphere.open_half_plane:

$\forall\, x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_half\_plane\ x = (\exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ \wedge\ x = aff\_gt\ (INSERT\ u\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY))$

thm DEF_closed_half_space:

$closed\_half\_space = (\lambda\_1968639::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ w'::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w'\ EMPTY))))\ \wedge\ \_1968639 = aff\_ge\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (INSERT\ w'\ EMPTY))$

thm Trigonometry.OAUVFPS1:

$\forall\, x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ closed\_half\_space\ x = (\exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ w'::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w'\ EMPTY))))\ \wedge\ x = aff\_ge\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (INSERT\ w'\ EMPTY))$

thm DEF_open_half_space:

$open\_half\_space = (\lambda\_1968644::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ w'::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w'\ EMPTY))))\ \wedge\ \_1968644 = aff\_gt\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (INSERT\ w'\ EMPTY))$

thm Trigonometry.OAUVFPS2:

$\forall\, x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ open\_half\_space\ x = (\exists\, (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ w'::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ (INSERT\ w'\ EMPTY))))\ \wedge\ x = aff\_gt\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (INSERT\ w'\ EMPTY))$

thm DEF_bis:

$bis = (\lambda(\_1968649::(real,\ ?'a::type)\ cart)\ \_1968650::(real,\ ?'a::type)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%4::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%4\ (distance\ (x,\ \_1968649) = distance\ (x,\ \_1968650))\ x))$

thm Sphere.bis:

$\forall\, (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ bis\ u\ v = GSPEC\ (\lambda GEN\%PVAR\%4::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%4\ (distance\ (x,\ u) = distance\ (x,\ v))\ x)$

thm DEF_bis_le:

bis_le = (λ(*_1968661*::(*real*, *?'a*::*type*) *cart*) *_1968662*::(*real*, *?'a*::*type*) *cart*. *GSPEC* (λ*GEN%PVAR%5*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%5* (*distance* (*x*, *_1968661*) ≤ *distance* (*x*, *_1968662*)) *x*))

thm Sphere.bis_le:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *bis_le u v* = *GSPEC* (λ*GEN%PVAR%5*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%5* (*distance* (*x*, *u*) ≤ *distance* (*x*, *v*)) *x*)

thm DEF_bis_lt:

bis_lt = (λ(*_1968673*::(*real*, *?'a*::*type*) *cart*) *_1968674*::(*real*, *?'a*::*type*) *cart*. *GSPEC* (λ*GEN%PVAR%6*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%6* (*distance* (*x*, *_1968673*) < *distance* (*x*, *_1968674*)) *x*))

thm Sphere.bis_lt:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *bis_lt u v* = *GSPEC* (λ*GEN%PVAR%6*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%6* (*distance* (*x*, *u*) < *distance* (*x*, *v*)) *x*)

thm Sphere.BIS_SYM:

∀ (*p*::(*real*, *?'a*::*type*) *cart*) *q*::(*real*, *?'a*::*type*) *cart*. *bis p q* = *bis q p*

thm DEF_circumcenter:

circumcenter = (λ*_1968697*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SOME v*::(*real*, *?'a*::*type*) *cart*. *hull affine _1968697 v* ∧ (∃ *c*::*real*. ∀ *w*::(*real*, *?'a*::*type*) *cart*. *_1968697 w* ⟶ *c* = *distance* (*v*, *w*)))

thm Sphere.circumcenter:

∀ *S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *circumcenter S* = (*SOME v*::(*real*, *?'a*::*type*) *cart*. *hull affine S v* ∧ (∃ *c*::*real*. ∀ *w*::(*real*, *?'a*::*type*) *cart*. *S w* ⟶ *c* = *distance* (*v*, *w*)))

thm DEF_radV:

radV = (λ*_1968702*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SOME c*::*real*. ∀ *w*::(*real*, *?'a*::*type*) *cart*. *_1968702 w* ⟶ *c* = *distance* (*circumcenter _1968702*, *w*))

thm Collect_geom.radV:

∀ *S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *radV S* = (*SOME c*::*real*. ∀ *w*::(*real*, *?'a*::*type*) *cart*. *S w* ⟶ *c* = *distance* (*circumcenter S*, *w*))

thm DEF_orientation:

orientation = (λ(*_1968707*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*_1968708*::(*real*, *?'a*::*type*) *cart*) *_1968709*::*real* ⇒ *bool*. *affsign _1968709* (*DIFF _1968707* (*INSERT _1968708 EMPTY*)) (*INSERT _1968708 EMPTY*) (*circumcenter _1968707*))

thm Sphere.orientation:

∀ (*sgn*::*real* ⇒ *bool*) (*v*::(*real*, *?'a*::*type*) *cart*) *S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *orientation S v sgn* = *affsign sgn* (*DIFF S* (*INSERT v EMPTY*)) (*INSERT v EMPTY*) (*circumcenter S*)

thm Sphere.arcV:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. *arcV u v w* = *acs* (*dot* (*vector_sub v u*) (*vector_sub w u*) / (*vector_norm* (*vector_sub v u*) ∗ *vector_norm* (*vector_sub w u*)))

thm Sphere.dihV:

∀ (*w1*::(*real*, *?'a*::*type*) *cart*) (*w3*::(*real*, *?'a*::*type*) *cart*) (*w2*::(*real*, *?'a*::*type*) *cart*) *w0*::(*real*, *?'a*::*type*) *cart*. *dihV w0 w1 w2 w3* = *LET* (λ*va*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vb*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vc*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vap*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*vbp*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*arcV* (*vec* (*0*::*nat*)) *vap vbp*)) (*vector_sub* (% (*dot vc vc*) *vb*) (% (*dot vb vc*) *vc*)))) (*vector_sub* (% (*dot vc vc*) *va*) (% (*dot va vc*) *vc*)))) (*vector_sub w1 w0*))) (*vector_sub w3 w0*))) (*vector_sub w2 w0*)

thm DEF_ylist:

ylist = (λ(*_1968728*::(*real*, *?'a*::*type*) *cart*) (*_1968729*::(*real*, *?'a*::*type*) *cart*) (*_1968730*::(*real*, *?'a*::*type*) *cart*) *_1968731*::(*real*, *?'a*::*type*) *cart*. (*distance* (*_1968728*, *_1968729*), *distance* (*_1968728*, *_1968730*), *distance* (*_1968728*, *_1968731*), *distance* (*_1968730*, *_1968731*), *distance* (*_1968729*, *_1968731*), *distance* (*_1968729*, *_1968730*)))

thm Sphere.ylist:

∀ (*w0*::(*real*, *?'a*::*type*) *cart*) (*w3*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) *w2*::(*real*, *?'a*::*type*) *cart*. *ylist w0 w1 w2 w3* = (*distance* (*w0*, *w1*), *distance* (*w0*, *w2*), *distance* (*w0*, *w3*), *distance* (*w2*, *w3*), *distance* (*w1*, *w3*), *distance* (*w1*, *w2*))

thm DEF_xlist:

xlist = (λ(*_1968760*::(*real*, *?'a*::*type*) *cart*) (*_1968761*::(*real*, *?'a*::*type*) *cart*) (*_1968762*::(*real*, *?'a*::*type*) *cart*) *_1968763*::(*real*, *?'a*::*type*) *cart*. *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real* × *real* × *real* × *real* × *real* × *real*. ∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *GEQ* (*f* (*y1*, *y2*, *y3*, *y4*, *y5*, *y6*)) (*LET_END* ($y1^2$, $y2^2$, $y3^2$, $y4^2$, $y5^2$, $y6^2$)))) (*ylist _1968760 _1968761 _1968762 _1968763*))

thm Sphere.xlist:

∀ (*w0*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) (*w2*::(*real*, *?'a*::*type*) *cart*) *w3*::(*real*, *?'a*::*type*) *cart*. *xlist w0 w1 w2 w3* = *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real* × *real* × *real* × *real* × *real* × *real*. ∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *GEQ* (*f* (*y1*, *y2*, *y3*, *y4*, *y5*, *y6*)) (*LET_END* (*y1*^{2}, *y2*^{2}, *y3*^{2}, *y4*^{2}, *y5*^{2}, *y6*^{2})))) (*ylist w0 w1 w2 w3*)

thm DEF_euler_p:

euler_p = (λ(*_1968792*::(*real*, *?'a*::*type*) *cart*) (*_1968793*::(*real*, *?'a*::*type*) *cart*) (*_1968794*::(*real*, *?'a*::*type*) *cart*) *_1968795*::(*real*, *?'a*::*type*) *cart*. *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real*. ∀(*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *GEQ* (*f* (*y1*, *y2*, *y3*, *y4*, *y5*, *y6*)) (*LET_END* (*LET* (λ*w1*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*w2*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*w3*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*y1* ∗ (*y2* ∗ *y3*) + (*y1* ∗ *dot w2 w3* + (*y2* ∗ *dot w3 w1* + *y3* ∗ *dot w1 w2*)))) (*vector_sub _1968795 _1968792*))) (*vector_sub _1968794 _1968792*))) (*vector_sub _1968793 _1968792*))))) (*ylist _1968792 _1968793 _1968794 _1968795*))

thm Sphere.euler_p:

∀(*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *euler_p v0 v1 v2 v3* = *LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *real*. ∀(*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *GEQ* (*f* (*y1*, *y2*, *y3*, *y4*, *y5*, *y6*)) (*LET_END* (*LET* (λ*w1*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*w2*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*LET* (λ*w3*::(*real*, *?'a*::*type*) *cart*. *LET_END* (*y1* ∗ (*y2* ∗ *y3*) + (*y1* ∗ *dot w2 w3* + (*y2* ∗ *dot w3 w1* + *y3* ∗ *dot w1 w2*)))) (*vector_sub v3 v0*))) (*vector_sub v2 v0*))) (*vector_sub v1 v0*))))) (*ylist v0 v1 v2 v3*)

thm DEF_beta:

beta = (λ(*_1968824*::*real*) *_1968825*::*real*. *LET* (λ*arg*::*real*. *LET_END* (*acs* (*sqrt arg*))) ((*cos _1968824* ∗ *cos _1968824* − *cos _1968825* ∗ *cos _1968825*) / ((*1*::*real*) − *cos _1968825* ∗ *cos _1968825*)))

thm Trigonometry2.beta:

∀(*psi*::*real*) *theta*::*real*. *beta psi theta* = *LET* (λ*arg*::*real*. *LET_END* (*acs* (*sqrt arg*))) ((*cos psi* ∗ *cos psi* − *cos theta* ∗ *cos theta*) / ((*1*::*real*) − *cos theta* ∗ *cos theta*))

thm DEF_radius:

radius = (λ*_1968836*::*real* × *real*. *sqrt* ((*fst _1968836*)^2 + (*snd _1968836*)^2))

thm Sphere.radius:

∀(*x*::*real*) *y*::*real*. *radius* (*x*, *y*) = *sqrt* (x^2 + y^2)

thm DEF_cyclic_set:

cyclic_set = (λ(*_1968845*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*_1968846*::(*real*, *?'a*::*type*) *cart*) *_1968847*::(*real*, *?'a*::*type*) *cart*. *_1968846* ≠ *_1968847* ∧ *FINITE _1968845* ∧ (∀(*p*::(*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *?'a*::*type*) *cart*) *h*::*real*. *_1968845 p* ∧ *_1968845 q* ∧ *p* = *vector_add q* (% *h* (*vector_sub _1968846 _1968847*)) ⟶ *p*

= q) ∧ HOL_Light_Import.INTER _1968845 (hull affine (INSERT _1968846 (INSERT _1968847 EMPTY))) = EMPTY)

thm Trigonometry2.cyclic_set:

∀ (W::(real, ?'a::type) cart ⇒ bool) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. cyclic_set W v w = (v ≠ w ∧ FINITE W ∧ (∀ (p::(real, ?'a::type) cart) (q::(real, ?'a::type) cart) h::real. W p ∧ W q ∧ p = vector_add q (% h (vector_sub v w)) ⟶ p = q) ∧ HOL_Light_Import.INTER W (hull affine (INSERT v (INSERT w EMPTY))) = EMPTY)

thm DEF_projection:

projection = (λ(_1968866::(real, ?'a::type) cart) _1968867::(real, ?'a::type) cart. vector_sub _1968867 (% (dot _1968867 _1968866 / dot _1968866 _1968866) _1968866))

thm Sphere.projection:

∀ (x::(real, ?'a::type) cart) e::(real, ?'a::type) cart. projection e x = vector_sub x (% (dot x e / dot e e) e)

thm DEF_azim_cycle:

azim_cycle = (λ(_1968878::(real, 3) cart ⇒ bool) (_1968879::(real, 3) cart) (_1968880::(real, 3) cart) _1968881::(real, 3) cart. if SUBSET _1968878 (INSERT _1968881 EMPTY) then _1968881 else SOME u::(real, 3) cart. u ≠ _1968881 ∧ _1968878 u ∧ (∀ q::(real, 3) cart. q ≠ _1968881 ∧ _1968878 q ⟶ azim _1968879 _1968880 _1968881 u < azim _1968879 _1968880 _1968881 q ∨ azim _1968879 _1968880 _1968881 u = azim _1968879 _1968880 _1968881 q ∧ vector_norm (projection (vector_sub _1968880 _1968879) (vector_sub u _1968879)) ≤ vector_norm (projection (vector_sub _1968880 _1968879) (vector_sub q _1968879))))

thm Sphere.azim_cycle:

∀ (W::(real, 3) cart ⇒ bool) (p::(real, 3) cart) (w::(real, 3) cart) v::(real, 3) cart. azim_cycle W v w p = (if SUBSET W (INSERT p EMPTY) then p else SOME u::(real, 3) cart. u ≠ p ∧ W u ∧ (∀ q::(real, 3) cart. q ≠ p ∧ W q ⟶ azim v w p u < azim v w p q ∨ azim v w p u = azim v w p q ∧ vector_norm (projection (vector_sub w v) (vector_sub u v)) ≤ vector_norm (projection (vector_sub w v) (vector_sub q v))))

thm DEF_packing:

packing = (λ_1968910::(real, 3) cart ⇒ bool. ∀ (u::(real, 3) cart) v::(real, 3) cart. _1968910 u ∧ _1968910 v ∧ u ≠ v ⟶ real_of_nat (2::nat) ≤ distance (u, v))

thm Sphere.packing:

∀ S::(real, 3) cart ⇒ bool. packing S = (∀ (u::(real, 3) cart) v::(real, 3) cart. S u ∧ S v ∧ u ≠ v ⟶ real_of_nat (2::nat) ≤ distance (u, v))

thm Sphere.packing_lt:

packing (?V::(real, 3) cart ⇒ bool) = (∀ (u::(real, 3) cart) v::(real, 3) cart. IN u ?V ∧ IN v ?V ∧ distance (u, v) < real_of_nat (2::nat) ⟶ u = v)

thm DEF_saturated:

saturated = (λ_1968959::(real, ?'a::type) cart ⇒ bool. ∀ x::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. IN y _1968959 ∧ distance (x, y) < real_of_nat (2::nat))

thm Pack1.saturated:

∀ S::(real, ?'a::type) cart ⇒ bool. saturated S = (∀ x::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. IN y S ∧ distance (x, y) < real_of_nat (2::nat))

thm Sphere.sphere:

∀ x::(real, 3) cart ⇒ bool. sphere x = (∃ (v::(real, 3) cart) r::real. (0::real) < r ∧ x = GSPEC (λGEN%PVAR%7::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%7 (vector_norm (vector_sub w v) = r) w))

thm Sphere.c_cone:

∀ (v::(real, 3) cart) (w::(real, 3) cart) r::real. c_cone (v, w, r) = GSPEC (λGEN%PVAR%8::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%8 (dot (vector_sub x v) w = vector_norm (vector_sub x v) ∗ (vector_norm w ∗ r)) x)

thm DEF_null_equiv:

null_equiv = (λ_1968964::((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool). ∃ B::(real, 3) cart ⇒ bool. negligible B ∧ SUBSET (HOL_Light_Import.UNION (DIFF (fst _1968964) (snd _1968964)) (DIFF (snd _1968964) (fst _1968964))) B)

thm Vol1.null_equiv:

∀ (t::(real, 3) cart ⇒ bool) s::(real, 3) cart ⇒ bool. null_equiv (s, t) = (∃ B::(real, 3) cart ⇒ bool. negligible B ∧ SUBSET (HOL_Light_Import.UNION (DIFF s t) (DIFF t s)) B)

thm DEF_radial:

radial = (λ(_1968973::real) (_1968974::(real, ?'a::type) cart) _1968975::(real, ?'a::type) cart ⇒ bool. SUBSET _1968975 (ball (_1968974, _1968973)) ∧ (∀ u::(real, ?'a::type) cart. IN (vector_add _1968974 u) _1968975 ⟶ (∀ t::real. (0::real) < t ∧ t ∗ vector_norm u < _1968973 ⟶ IN (vector_add _1968974 (% t u)) _1968975)))

thm Sphere.radial:

∀ (r::real) (x::(real, ?'a::type) cart) C::(real, ?'a::type) cart ⇒ bool. radial r x C = (SUBSET C (ball (x, r)) ∧ (∀ u::(real, ?'a::type) cart. IN (vector_add x

u) *C* ⟶ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* ∗ *vector_norm u* < *r* ⟶ *IN* (*vector_add x* (% *t u*)) *C*)))

thm DEF_eventually_radial:

eventually_radial = (λ(*_1968994*::(*real*, *?'a*::*type*) *cart*) *_1968995*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *r*>*0*::*real*. *radial r _1968994* (*HOL_Light_Import.INTER _1968995* (*ball* (*_1968994*, *r*))))

thm Sphere.eventually_radial:

∀ (*C*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *eventually_radial x C* = (∃ *r*>*0*::*real*. *radial r x* (*HOL_Light_Import.INTER C* (*ball* (*x*, *r*))))

thm Sphere.rconesgn:

∀ (*sgn*::*real* ⇒ *real* ⇒ *bool*) (*w*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *h*::*real*. *rconesgn sgn v w h* = *GSPEC* (λ*GEN%PVAR%9*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%9* (*sgn* (*dot* (*vector_sub x v*) (*vector_sub w v*)) (*distance* (*x*, *v*) ∗ (*distance* (*w*, *v*) ∗ *h*))) *x*)

thm Sphere.rcone_ge:

rcone_ge = *rconesgn real_ge*

thm Sphere.rcone_gt:

rcone_gt = *rconesgn real_gt*

thm Sphere.rcone_lt:

rcone_lt = *rconesgn op* <

thm Sphere.rcone_eq:

rcone_eq = *rconesgn op* =

thm Sphere.vol_solid_triangle:

∀ (*r*::*real*) (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *vol_solid_triangle v0 v1 v2 v3 r* = *LET* (λ*a123*::*real*. *LET_END* (*LET* (λ*a231*::*real*. *LET_END* (*LET* (λ*a312*::*real*. *LET_END* ((*a123* + (*a231* + (*a312* − *pi*))) ∗ ($r^{3::nat}$ / *real_of_nat* (*3*::*nat*)))) (*dihV v0 v3 v1 v2*))) (*dihV v0 v2 v3 v1*))) (*dihV v0 v1 v2 v3*)

thm Sphere.vol_conv:

∀ (*v4*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. *vol_conv v1 v2 v3 v4* = *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*sqrt* (*delta_x x12 x13 x14 x34 x24 x23*) / *real_of_nat* (*12*::*nat*))) ((*distance* (*v3*, *v4*))^2))) ((*distance* (*v2*, *v4*))^2))) ((*distance* (*v2*, *v3*))^2))) ((*distance* (*v1*, *v4*))^2))) ((*distance* (*v1*, *v3*))^2))) ((*distance* (*v1*, *v2*))^2)

thm Sphere.vol_rect:

∀(b::(real, ?'b::type) cart) a::(real, ?'a::type) cart. vol_rect a b = (if $ a (1::nat) < $ b (1::nat) ∧ $ a (2::nat) < $ b (2::nat) ∧ $ a (3::nat) < $ b (3::nat) then ($ b (3::nat) − $ a (3::nat)) ∗ (($ b (2::nat) − $ a (2::nat)) ∗ ($ b (1::nat) − $ a (1::nat))) else (0::real))

thm DEF_ortho0:

ortho0 = (λ(_1969006::(real, ?'a::type) cart) (_1969007::(real, ?'a::type) cart) (_1969008::(real, ?'a::type) cart) _1969009::(real, ?'a::type) cart. conv0 (INSERT _1969006 (INSERT (vector_add _1969006 _1969007) (INSERT (vector_add _1969006 (vector_add _1969007 _1969008)) (INSERT (vector_add _1969006 (vector_add _1969007 (vector_add _1969008 _1969009))) EMPTY)))))

thm Sphere.ortho0:

∀(x::(real, ?'a::type) cart) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. ortho0 x v1 v2 v3 = conv0 (INSERT x (INSERT (vector_add x v1) (INSERT (vector_add x (vector_add v1 v2)) (INSERT (vector_add x (vector_add v1 (vector_add v2 v3))) EMPTY))))

thm DEF_make_point:

make_point = (λ(_1969038::(real, 3) cart) (_1969039::(real, 3) cart) (_1969040::(real, 3) cart) (_1969041::(real, 3) cart) (_1969042::real) (_1969043::real) _1969044::real. SOME v::(real, 3) cart. aff_ge (INSERT _1969038 (INSERT _1969039 (INSERT _1969040 EMPTY))) (INSERT _1969041 EMPTY) v ∧ _1969042 = distance (_1969038, v) ∧ _1969043 = distance (_1969039, v) ∧ _1969044 = distance (_1969040, v))

thm Sphere.make_point:

∀(w::(real, 3) cart) (r1::real) (v1::(real, 3) cart) (r2::real) (v2::(real, 3) cart) (r3::real) v3::(real, 3) cart. make_point v1 v2 v3 w r1 r2 r3 = (SOME v::(real, 3) cart. aff_ge (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT w EMPTY) v ∧ r1 = distance (v1, v) ∧ r2 = distance (v2, v) ∧ r3 = distance (v3, v))

thm DEF_abc_param:

abc_param = (λ(_1969115::(real, ?'b::type) cart) (_1969116::(real, ?'b::type) cart) (_1969117::(real, ?'b::type) cart) _1969118::?'a::type. LET (λa::real. LET_END (LET (λb::real. LET_END (a, b, _1969118)) (radV (INSERT _1969115 (INSERT _1969116 (INSERT _1969117 EMPTY)))))) ((1::real) / real_of_nat (2::nat) ∗ distance (_1969115, _1969116)))

thm Sphere.abc_param:

∀(c::?'b::type) (v2::(real, ?'a::type) cart) (v0::(real, ?'a::type) cart) v1::(real, ?'a::type) cart. abc_param v0 v1 v2 c = LET (λa::real. LET_END (LET (λb::real. LET_END (a, b, c)) (radV (INSERT v0 (INSERT v1 (INSERT v2 EMPTY)))))) ((1::real) / real_of_nat (2::nat) ∗ distance (v0, v1))

thm DEF_res:

$res = (\lambda(\_1969147::?'a::type \Rightarrow ?'a::type)\ (\_1969148::?'a::type \Rightarrow bool)\ \_1969149::?'a::type.$ $if\ IN\ \_1969149\ \_1969148\ then\ \_1969147\ \_1969149\ else\ \_1969149)$

thm Hypermap.res:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ res\ f\ s\ x = (if\ IN\ x\ s\ then\ f\ x\ else\ x)$

thm DEF_regular_spherical_polygon_area:

$regular\_spherical\_polygon\_area = (\lambda(\_1969168::real)\ \_1969169::real.\ real\_of\_nat\ (2::nat) * pi - real\_of\_nat\ (2::nat) * (\_1969169 * asn\ (\_1969168 * sin\ (pi\ /\ \_1969169))))$

thm Sphere.regular_spherical_polygon_area:

$\forall (ca::real)\ k::real.\ regular\_spherical\_polygon\_area\ ca\ k = real\_of\_nat\ (2::nat) * pi - real\_of\_nat\ (2::nat) * (k * asn\ (ca * sin\ (pi\ /\ k)))$

thm Sphere.h0:

$h0 = DECIMAL\ (126::nat)\ (100::nat)$

thm Sphere.sol0:

$sol0 = real\_of\_nat\ (3::nat) * acs\ ((1::real)\ /\ real\_of\_nat\ (3::nat)) - pi$

thm Sphere.tau0:

$tau0 = real\_of\_nat\ (4::nat) * pi - real\_of\_nat\ (20::nat) * sol0$

thm Sphere.mm1:

$mm1 = sol0 * (sqrt\ (real\_of\_nat\ (8::nat))\ /\ tau0)$

thm Sphere.mm2:

$mm2 = (real\_of\_nat\ (6::nat) * sol0 - pi) * (sqrt\ (real\_of\_nat\ (2::nat))\ /\ (real\_of\_nat\ (6::nat) * tau0))$

thm Pack_defs.hplus:

$hplus = DECIMAL\ (13254::nat)\ (10000::nat)$

thm DEF_h0cut:

$h0cut = (\lambda\_1969180::real.\ if\ \_1969180 \leq real\_of\_nat\ (2::nat) * h0\ then\ 1::real\ else\ (0::real))$

thm Functional_equation.h0cut:

$\forall y::real.\ h0cut\ y = (if\ y \leq real\_of\_nat\ (2::nat) * h0\ then\ 1::real\ else\ (0::real))$

thm DEF_marchal_quartic:

$marchal\_quartic = (\lambda\_1969185::real.\ (sqrt\ (real\_of\_nat\ (2::nat)) - \_1969185) * ((\_1969185 - hplus) * ((real\_of\_nat\ (9::nat) * \_1969185^2 - real\_of\_nat$

$(17::nat) * \_1969185 + real\_of\_nat\ (3::nat)) / ((sqrt\ (real\_of\_nat\ (2::nat)) - (1::real)) * (real\_of\_nat\ (5::nat) * (hplus - (1::real)))))))$

thm Sphere.marchal_quartic:

$\forall h::real.\ marchal\_quartic\ h = (sqrt\ (real\_of\_nat\ (2::nat)) - h) * ((h - hplus) * ((real\_of\_nat\ (9::nat) * h^2 - real\_of\_nat\ (17::nat) * h + real\_of\_nat\ (3::nat)) / ((sqrt\ (real\_of\_nat\ (2::nat)) - (1::real)) * (real\_of\_nat\ (5::nat) * (hplus - (1::real))))))$

thm DEF_lmfun:

$lmfun = (\lambda\_1969190::real.\ if\ \_1969190 \leq h0\ then\ (h0 - \_1969190) / (h0 - (1::real))\ else\ (0::real))$

thm Pack_defs.lmfun:

$\forall h::real.\ lmfun\ h = (if\ h \leq h0\ then\ (h0 - h) / (h0 - (1::real))\ else\ (0::real))$

thm DEF_lfun:

$lfun = (\lambda\_1969195::real.\ (h0 - \_1969195) / (h0 - (1::real)))$

thm Sphere.lfun:

$\forall h::real.\ lfun\ h = (h0 - h) / (h0 - (1::real))$

thm DEF_flat_term:

$flat\_term = (\lambda\_1969200::real.\ sol0 * ((\_1969200 - real\_of\_nat\ (2::nat) * h0) / (real\_of\_nat\ (2::nat) * h0 - real\_of\_nat\ (2::nat))))$

thm Sphere.flat_term:

$\forall y::real.\ flat\_term\ y = sol0 * ((y - real\_of\_nat\ (2::nat) * h0) / (real\_of\_nat\ (2::nat) * h0 - real\_of\_nat\ (2::nat)))$

thm Sphere.hminus:

$hminus = (SOME\ x::real.\ DECIMAL\ (12::nat)\ (10::nat) \leq x \wedge x < DECIMAL\ (13::nat)\ (10::nat) \wedge marchal\_quartic\ x = lmfun\ x)$

thm DEF_y_of_x:

$y\_of\_x = (\lambda(\_1969205::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow ?'a::type)\ (\_1969206::real)\ (\_1969207::real)\ (\_1969208::real)\ (\_1969209::real)\ (\_1969210::real)\ \_1969211::real.\ \_1969205\ (\_1969206 * \_1969206)\ (\_1969207 * \_1969207)\ (\_1969208 * \_1969208)\ (\_1969209 * \_1969209)\ (\_1969210 * \_1969210)\ (\_1969211 * \_1969211))$

thm Sphere.y_of_x:

$\forall (fx::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow ?'a::type)\ (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ y\_of\_x\ fx\ y1\ y2\ y3\ y4\ y5\ y6 = fx\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Sphere.rad2_y:

rad2_y = y_of_x rad2_x

thm Sphere.delta4_y:

delta4_y = y_of_x delta_x4

thm Sphere.vol_y:

vol_y = y_of_x vol_x

thm DEF_vol4f:

vol4f = (λ(_1969282::real) (_1969283::real) (_1969284::real) (_1969285::real) (_1969286::real) (_1969287::real) _1969288::real ⇒ real. real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (sol_y _1969282 _1969283 _1969284 _1969285 _1969286 _1969287 + (sol_y _1969282 _1969286 _1969287 _1969285 _1969283 _1969284 + (sol_y _1969285 _1969286 _1969284 _1969282 _1969283 _1969287 + sol_y _1969285 _1969283 _1969287 _1969282 _1969286 _1969284))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (_1969288 (_1969282 / real_of_nat (2::nat)) ∗ dih_y _1969282 _1969283 _1969284 _1969285 _1969286 _1969287 + (_1969288 (_1969283 / real_of_nat (2::nat)) ∗ dih_y _1969283 _1969284 _1969282 _1969286 _1969287 _1969285 + (_1969288 (_1969284 / real_of_nat (2::nat)) ∗ dih_y _1969284 _1969282 _1969283 _1969287 _1969285 _1969286 + (_1969288 (_1969285 / real_of_nat (2::nat)) ∗ dih_y _1969285 _1969284 _1969286 _1969282 _1969287 _1969283 + (_1969288 (_1969286 / real_of_nat (2::nat)) ∗ dih_y _1969286 _1969282 _1969287 _1969283 _1969285 _1969284 + _1969288 (_1969287 / real_of_nat (2::nat)) ∗ dih_y _1969287 _1969282 _1969286 _1969284 _1969285 _1969283))))))

thm Sphere.vol4f:

∀ (f::real ⇒ real) (y6::real) (y1::real) (y5::real) (y3::real) (y4::real) y2::real. vol4f y1 y2 y3 y4 y5 y6 f = real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (sol_y y1 y2 y3 y4 y5 y6 + (sol_y y1 y5 y6 y4 y2 y3 + (sol_y y4 y5 y3 y1 y2 y6 + sol_y y4 y2 y6 y1 y5 y3))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (f (y1 / real_of_nat (2::nat)) ∗ dih_y y1 y2 y3 y4 y5 y6 + (f (y2 / real_of_nat (2::nat)) ∗ dih_y y2 y3 y1 y5 y6 y4 + (f (y3 / real_of_nat (2::nat)) ∗ dih_y y3 y1 y2 y6 y4 y5 + (f (y4 / real_of_nat (2::nat)) ∗ dih_y y4 y3 y5 y1 y6 y2 + (f (y5 / real_of_nat (2::nat)) ∗ dih_y y5 y1 y6 y2 y4 y3 + f (y6 / real_of_nat (2::nat)) ∗ dih_y y6 y1 y5 y3 y4 y2)))))

thm DEF_gamma4f:

gamma4f = (λ(_1969359::real) (_1969360::real) (_1969361::real) (_1969362::real) (_1969363::real) (_1969364::real) _1969365::real ⇒ real. vol_y _1969359 _1969360 _1969361 _1969362 _1969363 _1969364 − vol4f _1969359 _1969360 _1969361 _1969362 _1969363 _1969364 _1969365)

thm Sphere.gamma4f:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) f::real ⇒ real. gamma4f y1 y2 y3 y4 y5 y6 f = vol_y y1 y2 y3 y4 y5 y6 − vol4f y1 y2 y3 y4 y5 y6 f

thm DEF_gamma4fgcy:

gamma4fgcy = gamma4f

thm Sphere.gamma4fgcy:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) *f*::*real* ⇒ *real*. *gamma4fgcy y1 y2 y3 y4 y5 y6 f* = *gamma4f y1 y2 y3 y4 y5 y6 f*

thm DEF_vol3r:

vol3r = (λ(*_1969513*::*real*) (*_1969514*::*real*) (*_1969515*::*real*) *_1969516*::*real*. *vol_y _1969516 _1969516 _1969516 _1969513 _1969514 _1969515*)

thm Sphere.vol3r:

∀ (*r*::*real*) (*y1*::*real*) (*y2*::*real*) *y3*::*real*. *vol3r y1 y2 y3 r* = *vol_y r r r y1 y2 y3*

thm DEF_vol3f:

vol3f = (λ(*_1969545*::*real*) (*_1969546*::*real*) (*_1969547*::*real*) (*_1969548*::*real*) *_1969549*::*real* ⇒ *real*. *real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ (*sol_y _1969545 _1969546 _1969548 _1969548 _1969548 _1969547* + (*sol_y _1969546 _1969547 _1969548 _1969548 _1969548 _1969545* + *sol_y _1969547 _1969545 _1969548 _1969548 _1969548 _1969546*)) − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*_1969549* (*_1969545* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y _1969545 _1969546 _1969548 _1969548 _1969548 _1969547* + (*_1969549* (*_1969546* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y _1969546 _1969547 _1969548 _1969548 _1969548 _1969545* + *_1969549* (*_1969547* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y _1969547 _1969545 _1969548 _1969548 _1969548 _1969546*)))

thm Sphere.vol3f:

∀ (*f*::*real* ⇒ *real*) (*y3*::*real*) (*y1*::*real*) (*r*::*real*) *y2*::*real*. *vol3f y1 y2 y3 r f* = *real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ (*sol_y y1 y2 r r r y3* + (*sol_y y2 y3 r r r y1* + *sol_y y3 y1 r r r y2*)) − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*f* (*y1* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y y1 y2 r r r y3* + (*f* (*y2* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y y2 y3 r r r y1* + *f* (*y3* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y y3 y1 r r r y2*))

thm DEF_gamma3f:

gamma3f = (λ(*_1969590*::*real*) (*_1969591*::*real*) (*_1969592*::*real*) (*_1969593*::*real*) *_1969594*::*real* ⇒ *real*. *vol3r _1969590 _1969591 _1969592 _1969593* − *vol3f _1969590 _1969591 _1969592 _1969593 _1969594*)

thm Sphere.gamma3f:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*r*::*real*) *f*::*real* ⇒ *real*. *gamma3f y1 y2 y3 r f* = *vol3r y1 y2 y3 r* − *vol3f y1 y2 y3 r f*

thm DEF_vol2r:

vol2r = (λ(*_1969635*::*real*) *_1969636*::*real*. *real_of_nat* (*2*::*nat*) ∗ (*pi* ∗ ((*_1969636* ∗ *_1969636* − (*_1969635* / *real_of_nat* (*2*::*nat*))^2) / *real_of_nat* (*3*::*nat*))))

thm Sphere.vol2r:

$\forall (r::real)\ y::real.\ vol2r\ y\ r = real\_of\_nat\ (2::nat) * (pi * ((r * r - (y / real\_of\_nat\ (2::nat))^2) / real\_of\_nat\ (3::nat)))$

thm DEF_vol2f:

$vol2f = (\lambda(\_1969647::real)\ (\_1969648::real)\ \_1969649::real \Rightarrow real.\ real\_of\_nat\ (2::nat) * (mm1 / pi) * (real\_of\_nat\ (2::nat) * (pi * ((1::real) - \_1969647 / (\_1969648 * real\_of\_nat\ (2::nat))))) - real\_of\_nat\ (8::nat) * (mm2 / pi) * (real\_of\_nat\ (2::nat) * (pi * \_1969649\ (\_1969647 / real\_of\_nat\ (2::nat)))))$

thm Sphere.vol2f:

$\forall (r::real)\ (f::real \Rightarrow real)\ y::real.\ vol2f\ y\ r\ f = real\_of\_nat\ (2::nat) * (mm1 / pi) * (real\_of\_nat\ (2::nat) * (pi * ((1::real) - y / (r * real\_of\_nat\ (2::nat))))) - real\_of\_nat\ (8::nat) * (mm2 / pi) * (real\_of\_nat\ (2::nat) * (pi * f\ (y / real\_of\_nat\ (2::nat))))$

thm DEF_norm2hh:

$norm2hh = (\lambda(\_1969668::real)\ (\_1969669::real)\ (\_1969670::real)\ (\_1969671::real)\ (\_1969672::real)\ \_1969673::real.\ (\_1969668 - hminus - hplus)^2 + ((\_1969669 - real\_of\_nat\ (2::nat))^2 + ((\_1969670 - real\_of\_nat\ (2::nat))^2 + ((\_1969671 - real\_of\_nat\ (2::nat))^2 + ((\_1969672 - real\_of\_nat\ (2::nat))^2 + (\_1969673 - real\_of\_nat\ (2::nat))^2)))))$

thm Sphere.norm2hh:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ norm2hh\ y1\ y2\ y3\ y4\ y5\ y6 = (y1 - hminus - hplus)^2 + ((y2 - real\_of\_nat\ (2::nat))^2 + ((y3 - real\_of\_nat\ (2::nat))^2 + ((y4 - real\_of\_nat\ (2::nat))^2 + ((y5 - real\_of\_nat\ (2::nat))^2 + (y6 - real\_of\_nat\ (2::nat))^2))))$

thm DEF_bump:

$bump = (\lambda\_1969728::real.\ DECIMAL\ (5::nat)\ (1000::nat) * ((1::real) - (\_1969728 - h0)^2 / (hplus - h0)^2))$

thm Pack_defs.bump:

$\forall h::real.\ bump\ h = DECIMAL\ (5::nat)\ (1000::nat) * ((1::real) - (h - h0)^2 / (hplus - h0)^2)$

thm DEF_critical_edge_y:

$critical\_edge\_y = (\lambda\_1969733::real.\ real\_of\_nat\ (2::nat) * hminus \leq \_1969733 \land \_1969733 \leq real\_of\_nat\ (2::nat) * hplus)$

thm Pack_defs.critical_edge_y:

$\forall y::real.\ critical\_edge\_y\ y = (real\_of\_nat\ (2::nat) * hminus \leq y \land y \leq real\_of\_nat\ (2::nat) * hplus)$

thm DEF_beta_bumpA_y:

beta_bumpA_y = (λ(*_1969738*::*real*) (*_1969739*::*real*) (*_1969740*::*real*) (*_1969741*::*real*) (*_1969742*::*real*) *_1969743*::*real*. (*if critical_edge_y _1969738 then 1*::*real else* (*0*::*real*)) ∗ ((*if _1969739* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if _1969740* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if critical_edge_y _1969741 then 1*::*real else* (*0*::*real*)) ∗ ((*if _1969742* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if _1969743* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ (*bump* (*_1969738* / *real_of_nat* (*2*::*nat*)) − *bump* (*_1969741* / *real_of_nat* (*2*::*nat*)))))))))

thm Sphere.beta_bumpA_y:

∀ (*y2*::*real*) (*y3*::*real*) (*y5*::*real*) (*y6*::*real*) (*y1*::*real*) *y4*::*real*. *beta_bumpA_y y1 y2 y3 y4 y5 y6* = (*if critical_edge_y y1 then 1*::*real else* (*0*::*real*)) ∗ ((*if y2* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if y3* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if critical_edge_y y4 then 1*::*real else* (*0*::*real*)) ∗ ((*if y5* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ ((*if y6* < *real_of_nat* (*2*::*nat*) ∗ *hminus then 1*::*real else* (*0*::*real*)) ∗ (*bump* (*y1* / *real_of_nat* (*2*::*nat*)) − *bump* (*y4* / *real_of_nat* (*2*::*nat*))))))))

thm DEF_beta_bump_force_y:

beta_bump_force_y = (λ(*_1969798*::*real*) (*_1969799*::*?'d*::*type*) (*_1969800*::*?'c*::*type*) (*_1969801*::*real*) (*_1969802*::*?'b*::*type*) *_1969803*::*?'a*::*type*. *bump* (*_1969798* / *real_of_nat* (*2*::*nat*)) − *bump* (*_1969801* / *real_of_nat* (*2*::*nat*)))

thm Sphere.beta_bump_force_y:

∀ (*y2*::*?'d*::*type*) (*y3*::*?'c*::*type*) (*y5*::*?'b*::*type*) (*y6*::*?'a*::*type*) (*y1*::*real*) *y4*::*real*. *beta_bump_force_y y1 y2 y3 y4 y5 y6* = *bump* (*y1* / *real_of_nat* (*2*::*nat*)) − *bump* (*y4* / *real_of_nat* (*2*::*nat*))

thm DEF_wtcount3_y:

wtcount3_y = (λ(*_1969858*::*real*) (*_1969859*::*real*) *_1969860*::*real*. (*if critical_edge_y _1969858 then 1*::*nat else* (*0*::*nat*)) + ((*if critical_edge_y _1969859 then 1*::*nat else* (*0*::*nat*)) + (*if critical_edge_y _1969860 then 1*::*nat else* (*0*::*nat*))))

thm Sphere.wtcount3_y:

∀ (*y1*::*real*) (*y2*::*real*) *y3*::*real*. *wtcount3_y y1 y2 y3* = (*if critical_edge_y y1 then 1*::*nat else* (*0*::*nat*)) + ((*if critical_edge_y y2 then 1*::*nat else* (*0*::*nat*)) + (*if critical_edge_y y3 then 1*::*nat else* (*0*::*nat*)))

thm DEF_wtcount6_y:

wtcount6_y = (λ(*_1969879*::*real*) (*_1969880*::*real*) (*_1969881*::*real*) (*_1969882*::*real*) (*_1969883*::*real*) *_1969884*::*real*. *wtcount3_y _1969879 _1969880 _1969881* + *wtcount3_y _1969882 _1969883 _1969884*)

thm Pack_defs.wtcount6_y:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *wtcount6_y y1 y2 y3 y4 y5 y6* = *wtcount3_y y1 y2 y3* + *wtcount3_y y4 y5 y6*

thm Sphere.machine_eps:

machine_eps = (*0*::*real*)

thm Sphere.a_spine5:

a_spine5 = *DECIMAL* (*560305*::*nat*) (*10000000*::*nat*)

thm Sphere.b_spine5:

b_spine5 = − *DECIMAL* (*445813*::*nat*) (*10000000*::*nat*)

thm Sphere.beta_bump_lb:

beta_bump_lb = − *DECIMAL* (*5*::*nat*) (*1000*::*nat*)

thm DEF_gamma23f:

gamma23f = (λ(*_1969939*::*real*) (*_1969940*::*real*) (*_1969941*::*real*) (*_1969942*::*real*) (*_1969943*::*real*) (*_1969944*::*real*) (*_1969945*::*nat*) (*_1969946*::*nat*) (*_1969947*::*real*) *_1969948*::*real* ⇒ *real*. *gamma3f _1969939 _1969940 _1969944 _1969947 _1969948* / *real_of_nat _1969945* + (*gamma3f _1969939 _1969941 _1969943 _1969947 _1969948* / *real_of_nat _1969946* + (*dih_y _1969939 _1969940 _1969941 _1969942 _1969943 _1969944* − *dih_y _1969939 _1969940 _1969947 _1969947 _1969947 _1969944* − *dih_y _1969939 _1969941 _1969947 _1969947 _1969947 _1969943*) ∗ ((*vol2r _1969939 _1969947* − *vol2f _1969939 _1969947 _1969948*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))))

thm Nonlinear_lemma.gamma23f:

∀ (*w1*::*nat*) (*w2*::*nat*) (*y4*::*real*) (*y2*::*real*) (*y6*::*real*) (*y3*::*real*) (*y5*::*real*) (*y1*::*real*) (*r*::*real*) *f*::*real* ⇒ *real*. *gamma23f y1 y2 y3 y4 y5 y6 w1 w2 r f* = *gamma3f y1 y2 y6 r f* / *real_of_nat w1* + (*gamma3f y1 y3 y5 r f* / *real_of_nat w2* + (*dih_y y1 y2 y3 y4 y5 y6* − *dih_y y1 y2 r r r y6* − *dih_y y1 y3 r r r y5*) ∗ ((*vol2r y1 r* − *vol2f y1 r f*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm DEF_gamma23f_126_03:

gamma23f_126_03 = (λ(*_1970079*::*real*) (*_1970080*::*real*) (*_1970081*::*real*) (*_1970082*::*real*) (*_1970083*::*real*) (*_1970084*::*real*) (*_1970085*::*nat*) (*_1970086*::*real*) *_1970087*::*real* ⇒ *real*. *gamma3f _1970079 _1970080 _1970084 _1970086 _1970087* / *real_of_nat _1970085* + (*dih_y _1970079 _1970080 _1970081 _1970082 _1970083 _1970084* − *dih_y _1970079 _1970080 _1970086 _1970086 _1970086 _1970084* − *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r _1970079 _1970086* − *vol2f _1970079 _1970086 _1970087*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Sphere.gamma23f_126_03:

∀ (*w1*::*nat*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y2*::*real*) (*y6*::*real*) (*y1*::*real*) (*r*::*real*) *f*::*real* ⇒ *real*. *gamma23f_126_03 y1 y2 y3 y4 y5 y6 w1 r f* = *gamma3f y1 y2 y6 r f* / *real_of_nat w1* + (*dih_y y1 y2 y3 y4 y5 y6* − *dih_y y1 y2 r r r y6* − *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r y1 r* − *vol2f y1 r f*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))

thm DEF_gamma23f_red_03:

$gamma23f\_red\_03 = (\lambda(\_1970196::real)\ (\_1970197::real)\ (\_1970198::real)\ (\_1970199::real)\ (\_1970200::real)\ (\_1970201::real)\ (\_1970202::real)\ \_1970203::real \Rightarrow real.\ (dih\_y\ \_1970196\ \_1970197\ \_1970198\ \_1970199\ \_1970200\ \_1970201 - real\_of\_nat\ (2::nat) * DECIMAL\ (3::nat)\ (100::nat)) * ((vol2r\ \_1970196\ \_1970202 - vol2f\ \_1970196\ \_1970202\ \_1970203)\ /\ (real\_of\_nat\ (2::nat) * pi)))$

thm Nonlin_def.gamma23f_red_03:

$\forall (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ (y6::real)\ (y1::real)\ (r::real)\ f::real \Rightarrow real.\ gamma23f\_red\_03\ y1\ y2\ y3\ y4\ y5\ y6\ r\ f = (dih\_y\ y1\ y2\ y3\ y4\ y5\ y6 - real\_of\_nat\ (2::nat) * DECIMAL\ (3::nat)\ (100::nat)) * ((vol2r\ y1\ r - vol2f\ y1\ r\ f)\ /\ (real\_of\_nat\ (2::nat) * pi))$

thm DEF_pathL:

$pathL = (\lambda\_1970292::real \times real.\ (fst\ \_1970292, (fst\ \_1970292 + snd\ \_1970292)\ /\ real\_of\_nat\ (2::nat)))$

thm Sphere.pathL:

$\forall (a::real)\ b::real.\ pathL\ (a,\ b) = (a,\ (a + b)\ /\ real\_of\_nat\ (2::nat))$

thm DEF_pathR:

$pathR = (\lambda\_1970301::real \times real.\ ((fst\ \_1970301 + snd\ \_1970301)\ /\ real\_of\_nat\ (2::nat),\ snd\ \_1970301))$

thm Sphere.pathR:

$\forall (a::real)\ b::real.\ pathR\ (a,\ b) = ((a + b)\ /\ real\_of\_nat\ (2::nat),\ b)$

thm DEF_rotate2:

$rotate2 = (\lambda(\_1970310::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (\_1970311::?'e::type)\ (\_1970312::?'g::type)\ (\_1970313::?'f::type)\ (\_1970314::?'b::type)\ (\_1970315::?'d::type)\ \_1970316::?'c::type.\ \_1970310\ \_1970312\ \_1970313\ \_1970311\ \_1970315\ \_1970316\ \_1970314)$

thm Sphere.rotate2:

$\forall (f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (x2::?'g::type)\ (x3::?'f::type)\ (x1::?'e::type)\ (x5::?'d::type)\ (x6::?'c::type)\ x4::?'b::type.\ rotate2\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x2\ x3\ x1\ x5\ x6\ x4$

thm DEF_rotate3:

$rotate3 = (\lambda(\_1970387::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (\_1970388::?'f::type)\ (\_1970389::?'e::type)\ (\_1970390::?'g::type)\ (\_1970391::?'c::type)\ (\_1970392::?'b::type)\ \_1970393::?'d::type.\ \_1970387\ \_1970390\ \_1970388\ \_1970389\ \_1970393\ \_1970391\ \_1970392)$

thm Sphere.rotate3:

$\forall (f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(x3::?'g::type)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x6::?'d::type)$ $(x4::?'c::type)$ $x5::?'b::type.$ $rotate3\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x3\ x1\ x2\ x6\ x4\ x5$

thm DEF_rotate4:

$rotate4 = (\lambda(\_1970464::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(\_1970465::?'d::type)$ $(\_1970466::?'f::type)$ $(\_1970467::?'b::type)$ $(\_1970468::?'g::type)$ $(\_1970469::?'c::type)$ $\_1970470::?'e::type.$ $\_1970464\ \_1970468\ \_1970466\ \_1970470\ \_1970465\ \_1970469\ \_1970467)$

thm Sphere.rotate4:

$\forall (f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(x4::?'g::type)$ $(x2::?'f::type)$ $(x6::?'e::type)$ $(x1::?'d::type)$ $(x5::?'c::type)$ $x3::?'b::type.$ $rotate4\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x4\ x2\ x6\ x1\ x5\ x3$

thm DEF_rotate5:

$rotate5 = (\lambda(\_1970541::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(\_1970542::?'b::type)$ $(\_1970543::?'d::type)$ $(\_1970544::?'f::type)$ $(\_1970545::?'e::type)$ $(\_1970546::?'g::type)$ $\_1970547::?'c::type.$ $\_1970541\ \_1970546\ \_1970544\ \_1970545\ \_1970543\ \_1970547\ \_1970542)$

thm Sphere.rotate5:

$\forall (f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(x5::?'g::type)$ $(x3::?'f::type)$ $(x4::?'e::type)$ $(x2::?'d::type)$ $(x6::?'c::type)$ $x1::?'b::type.$ $rotate5\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x5\ x3\ x4\ x2\ x6\ x1$

thm DEF_rotate6:

$rotate6 = (\lambda(\_1970618::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(\_1970619::?'f::type)$ $(\_1970620::?'b::type)$ $(\_1970621::?'d::type)$ $(\_1970622::?'c::type)$ $(\_1970623::?'e::type)$ $\_1970624::?'g::type.$ $\_1970618\ \_1970624\ \_1970619\ \_1970623\ \_1970621\ \_1970622\ \_1970620)$

thm Sphere.rotate6:

$\forall (f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $(x6::?'g::type)$ $(x1::?'f::type)$ $(x5::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $x2::?'b::type.$ $rotate6\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x6\ x1\ x5\ x3\ x4\ x2$

thm DEF_norm2hh_x:

$norm2hh\_x = (\lambda(\_1970695::real)$ $(\_1970696::real)$ $(\_1970697::real)$ $(\_1970698::real)$ $(\_1970699::real)$ $\_1970700::real.$ $norm2hh\ (sqrt\ \_1970695)\ (sqrt\ \_1970696)\ (sqrt\ \_1970697)\ (sqrt\ \_1970698)\ (sqrt\ \_1970699)\ (sqrt\ \_1970700))$

thm Sphere.norm2hh_x:

$\forall (x1::real)$ $(x2::real)$ $(x3::real)$ $(x4::real)$ $(x5::real)$ $x6::real.$ $norm2hh\_x\ x1\ x2\ x3\ x4\ x5\ x6 = norm2hh\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6)$

thm DEF_rhazim_x:

rhazim_x = (λ(*_1970755*::*real*) (*_1970756*::*real*) (*_1970757*::*real*) (*_1970758*::*real*) (*_1970759*::*real*) *_1970760*::*real*. *rhazim* (*sqrt* *_1970755*) (*sqrt* *_1970756*) (*sqrt* *_1970757*) (*sqrt* *_1970758*) (*sqrt* *_1970759*) (*sqrt* *_1970760*))

thm Sphere.rhazim_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *rhazim_x x1 x2 x3 x4 x5 x6* = *rhazim* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_rhazim2_x:

rhazim2_x = (λ(*_1970815*::*real*) (*_1970816*::*real*) (*_1970817*::*real*) (*_1970818*::*real*) (*_1970819*::*real*) *_1970820*::*real*. *rhazim2* (*sqrt* *_1970815*) (*sqrt* *_1970816*) (*sqrt* *_1970817*) (*sqrt* *_1970818*) (*sqrt* *_1970819*) (*sqrt* *_1970820*))

thm Sphere.rhazim2_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *rhazim2_x x1 x2 x3 x4 x5 x6* = *rhazim2* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_rhazim3_x:

rhazim3_x = (λ(*_1970875*::*real*) (*_1970876*::*real*) (*_1970877*::*real*) (*_1970878*::*real*) (*_1970879*::*real*) *_1970880*::*real*. *rhazim3* (*sqrt* *_1970875*) (*sqrt* *_1970876*) (*sqrt* *_1970877*) (*sqrt* *_1970878*) (*sqrt* *_1970879*) (*sqrt* *_1970880*))

thm Sphere.rhazim3_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *rhazim3_x x1 x2 x3 x4 x5 x6* = *rhazim3* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_dih4_x:

dih4_x = (λ(*_1970935*::*real*) (*_1970936*::*real*) (*_1970937*::*real*) (*_1970938*::*real*) (*_1970939*::*real*) *_1970940*::*real*. *dih4_y* (*sqrt* *_1970935*) (*sqrt* *_1970936*) (*sqrt* *_1970937*) (*sqrt* *_1970938*) (*sqrt* *_1970939*) (*sqrt* *_1970940*))

thm Sphere.dih4_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih4_x x1 x2 x3 x4 x5 x6* = *dih4_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_dih5_x:

dih5_x = (λ(*_1970995*::*real*) (*_1970996*::*real*) (*_1970997*::*real*) (*_1970998*::*real*) (*_1970999*::*real*) *_1971000*::*real*. *dih5_y* (*sqrt* *_1970995*) (*sqrt* *_1970996*) (*sqrt* *_1970997*) (*sqrt* *_1970998*) (*sqrt* *_1970999*) (*sqrt* *_1971000*))

thm Sphere.dih5_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih5_x x1 x2 x3 x4 x5 x6* = *dih5_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_dih6_x:

$dih6\_x = (\lambda(\_1971055::real)\ (\_1971056::real)\ (\_1971057::real)\ (\_1971058::real)\ (\_1971059::real)\ \_1971060::real.\ dih6\_y\ (sqrt\ \_1971055)\ (sqrt\ \_1971056)\ (sqrt\ \_1971057)\ (sqrt\ \_1971058)\ (sqrt\ \_1971059)\ (sqrt\ \_1971060))$

thm Sphere.dih6_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ dih6\_x\ x1\ x2\ x3\ x4\ x5\ x6 = dih6\_y\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6)$

thm DEF_gcy:

$gcy = (\lambda\_1971115::real.\ real\_of\_nat\ (4::nat) * (mm1\ /\ pi) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * lmfun\ (\_1971115\ /\ real\_of\_nat\ (2::nat)))$

thm Sphere.gcy:

$\forall y::real.\ gcy\ y = real\_of\_nat\ (4::nat) * (mm1\ /\ pi) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * lmfun\ (y\ /\ real\_of\_nat\ (2::nat))$

thm DEF_gchi:

$gchi = (\lambda\_1971120::real.\ real\_of\_nat\ (4::nat) * (mm1\ /\ pi) - real\_of\_nat\ (504::nat) * (mm2\ /\ pi)\ /\ real\_of\_nat\ (13::nat) + real\_of\_nat\ (200::nat) * (\_1971120 * (mm2\ /\ pi))\ /\ real\_of\_nat\ (13::nat))$

thm Sphere.gchi:

$\forall y::real.\ gchi\ y = real\_of\_nat\ (4::nat) * (mm1\ /\ pi) - real\_of\_nat\ (504::nat) * (mm2\ /\ pi)\ /\ real\_of\_nat\ (13::nat) + real\_of\_nat\ (200::nat) * (y * (mm2\ /\ pi))\ /\ real\_of\_nat\ (13::nat)$

thm DEF_gchi1_x:

$gchi1\_x = (\lambda(\_1971125::real)\ (\_1971126::real)\ (\_1971127::real)\ (\_1971128::real)\ (\_1971129::real)\ \_1971130::real.\ gchi\ (sqrt\ \_1971125) * dih\_x\ \_1971125\ \_1971126\ \_1971127\ \_1971128\ \_1971129\ \_1971130)$

thm Sphere.gchi1_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gchi1\_x\ x1\ x2\ x3\ x4\ x5\ x6 = gchi\ (sqrt\ x1) * dih\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_gchi2_x:

$gchi2\_x = (\lambda(\_1971185::real)\ (\_1971186::real)\ (\_1971187::real)\ (\_1971188::real)\ (\_1971189::real)\ \_1971190::real.\ gchi\ (sqrt\ \_1971186) * dih2\_x\ \_1971185\ \_1971186\ \_1971187\ \_1971188\ \_1971189\ \_1971190)$

thm Sphere.gchi2_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gchi2\_x\ x1\ x2\ x3\ x4\ x5\ x6 = gchi\ (sqrt\ x2) * dih2\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_gchi3_x:

gchi3_x = (λ(*_1971245*::*real*) (*_1971246*::*real*) (*_1971247*::*real*) (*_1971248*::*real*) (*_1971249*::*real*) *_1971250*::*real*. *gchi* (*sqrt* *_1971247*) ∗ *dih3_x* *_1971245* *_1971246* *_1971247* *_1971248* *_1971249* *_1971250*)

thm Sphere.gchi3_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gchi3_x x1 x2 x3 x4 x5 x6* = *gchi* (*sqrt x3*) ∗ *dih3_x x1 x2 x3 x4 x5 x6*

thm DEF_gchi4_x:

gchi4_x = (λ(*_1971305*::*real*) (*_1971306*::*real*) (*_1971307*::*real*) (*_1971308*::*real*) (*_1971309*::*real*) *_1971310*::*real*. *gchi* (*sqrt* *_1971308*) ∗ *dih4_x* *_1971305* *_1971306* *_1971307* *_1971308* *_1971309* *_1971310*)

thm Sphere.gchi4_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gchi4_x x1 x2 x3 x4 x5 x6* = *gchi* (*sqrt x4*) ∗ *dih4_x x1 x2 x3 x4 x5 x6*

thm DEF_gchi5_x:

gchi5_x = (λ(*_1971365*::*real*) (*_1971366*::*real*) (*_1971367*::*real*) (*_1971368*::*real*) (*_1971369*::*real*) *_1971370*::*real*. *gchi* (*sqrt* *_1971369*) ∗ *dih5_x* *_1971365* *_1971366* *_1971367* *_1971368* *_1971369* *_1971370*)

thm Sphere.gchi5_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gchi5_x x1 x2 x3 x4 x5 x6* = *gchi* (*sqrt x5*) ∗ *dih5_x x1 x2 x3 x4 x5 x6*

thm DEF_gchi6_x:

gchi6_x = (λ(*_1971425*::*real*) (*_1971426*::*real*) (*_1971427*::*real*) (*_1971428*::*real*) (*_1971429*::*real*) *_1971430*::*real*. *gchi* (*sqrt* *_1971430*) ∗ *dih6_x* *_1971425* *_1971426* *_1971427* *_1971428* *_1971429* *_1971430*)

thm Sphere.gchi6_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gchi6_x x1 x2 x3 x4 x5 x6* = *gchi* (*sqrt x6*) ∗ *dih6_x x1 x2 x3 x4 x5 x6*

thm DEF_ldih_x:

ldih_x = (λ(*_1971485*::*real*) (*_1971486*::*real*) (*_1971487*::*real*) (*_1971488*::*real*) (*_1971489*::*real*) *_1971490*::*real*. *lfun* (*sqrt* *_1971485* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x* *_1971485* *_1971486* *_1971487* *_1971488* *_1971489* *_1971490*)

thm Sphere.ldih_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih_x x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x1* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x x1 x2 x3 x4 x5 x6*

thm DEF_ldih2_x:

$ldih2\_x = (\lambda(\_1971545::real)\ (\_1971546::real)\ (\_1971547::real)\ (\_1971548::real)\ (\_1971549::real)\ \_1971550::real.\ lfun\ (sqrt\ \_1971546\ /\ real\_of\_nat\ (2::nat)) * dih2\_x\ \_1971545\ \_1971546\ \_1971547\ \_1971548\ \_1971549\ \_1971550)$

thm Sphere.ldih2_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih2\_x\ x1\ x2\ x3\ x4\ x5\ x6 = lfun\ (sqrt\ x2\ /\ real\_of\_nat\ (2::nat)) * dih2\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_ldih3_x:

$ldih3\_x = (\lambda(\_1971605::real)\ (\_1971606::real)\ (\_1971607::real)\ (\_1971608::real)\ (\_1971609::real)\ \_1971610::real.\ lfun\ (sqrt\ \_1971607\ /\ real\_of\_nat\ (2::nat)) * dih3\_x\ \_1971605\ \_1971606\ \_1971607\ \_1971608\ \_1971609\ \_1971610)$

thm Sphere.ldih3_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih3\_x\ x1\ x2\ x3\ x4\ x5\ x6 = lfun\ (sqrt\ x3\ /\ real\_of\_nat\ (2::nat)) * dih3\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_ldih6_x:

$ldih6\_x = (\lambda(\_1971665::real)\ (\_1971666::real)\ (\_1971667::real)\ (\_1971668::real)\ (\_1971669::real)\ \_1971670::real.\ lfun\ (sqrt\ \_1971670\ /\ real\_of\_nat\ (2::nat)) * dih6\_x\ \_1971665\ \_1971666\ \_1971667\ \_1971668\ \_1971669\ \_1971670)$

thm Sphere.ldih6_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih6\_x\ x1\ x2\ x3\ x4\ x5\ x6 = lfun\ (sqrt\ x6\ /\ real\_of\_nat\ (2::nat)) * dih6\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_matan:

$matan = (\lambda\_1971725::real.\ if\ \_1971725 = (0::real)\ then\ 1::real\ else\ if\ (0::real) < \_1971725\ then\ atn\ (sqrt\ \_1971725)\ /\ sqrt\ \_1971725\ else\ log\ (((1::real) + sqrt\ (-\ \_1971725))\ /\ ((1::real) - sqrt\ (-\ \_1971725)))\ /\ (real\_of\_nat\ (2::nat) * sqrt\ (-\ \_1971725)))$

thm Sphere.matan:

$\forall x::real.\ matan\ x = (if\ x = (0::real)\ then\ 1::real\ else\ if\ (0::real) < x\ then\ atn\ (sqrt\ x)\ /\ sqrt\ x\ else\ log\ (((1::real) + sqrt\ (-\ x))\ /\ ((1::real) - sqrt\ (-\ x)))\ /\ (real\_of\_nat\ (2::nat) * sqrt\ (-\ x)))$

thm DEF_sol_euler_x:

$sol\_euler\_x = (\lambda(\_1971730::real)\ (\_1971731::real)\ (\_1971732::real)\ (\_1971733::real)\ (\_1971734::real)\ \_1971735::real.\ LET\ (\lambda a::real.\ LET\_END\ (real\_of\_nat\ (2::nat) * atn2\ (real\_of\_nat\ (2::nat) * a,\ sqrt\ (delta\_x\ \_1971730\ \_1971731\ \_1971732\ \_1971733\ \_1971734\ \_1971735))))\ (sqrt\ (\_1971730 * (\_1971731 * \_1971732)) + (sqrt\ \_1971730 * ((\_1971731 + (\_1971732 - \_1971733))\ /\ real\_of\_nat\ (2::nat)) + (sqrt\ \_1971731 * ((\_1971730 + (\_1971732 - \_1971734))\ /\ real\_of\_nat\ (2::nat)) + sqrt\ \_1971732 * ((\_1971730 + (\_1971731 - \_1971735))\ /\ real\_of\_nat\ (2::nat))))))$

thm Sphere.sol_euler_x:

$\forall$ (*x4*::*real*) (*x5*::*real*) (*x3*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *sol_euler_x x1 x2 x3 x4 x5 x6* = *LET* (λ*a*::*real*. *LET_END* (*real_of_nat* (*2*::*nat*) ∗ *atn2* (*real_of_nat* (*2*::*nat*) ∗ *a*, *sqrt* (*delta_x x1 x2 x3 x4 x5 x6*)))) (*sqrt* (*x1* ∗ (*x2* ∗ *x3*)) + (*sqrt x1* ∗ ((*x2* + (*x3* − *x4*)) / *real_of_nat* (*2*::*nat*)) + (*sqrt x2* ∗ ((*x1* + (*x3* − *x5*)) / *real_of_nat* (*2*::*nat*)) + *sqrt x3* ∗ ((*x1* + (*x2* − *x6*)) / *real_of_nat* (*2*::*nat*)))))

thm DEF_sol_euler_x_div_sqrtdelta:

sol_euler_x_div_sqrtdelta = (λ(*_1971790*::*real*) (*_1971791*::*real*) (*_1971792*::*real*) (*_1971793*::*real*) (*_1971794*::*real*) *_1971795*::*real*. *LET* (λ*a*::*real*. *LET_END* (*matan* (*delta_x _1971790 _1971791 _1971792 _1971793 _1971794 _1971795* / (*real_of_nat* (*4*::*nat*) ∗ a^2)) / *a*)) (*sqrt* (*_1971790* ∗ (*_1971791* ∗ *_1971792*)) + (*sqrt _1971790* ∗ ((*_1971791* + (*_1971792* − *_1971793*)) / *real_of_nat* (*2*::*nat*)) + (*sqrt _1971791* ∗ ((*_1971790* + (*_1971792* − *_1971794*)) / *real_of_nat* (*2*::*nat*)) + *sqrt _1971792* ∗ ((*_1971790* + (*_1971791* − *_1971795*)) / *real_of_nat* (*2*::*nat*))))))

thm Sphere.sol_euler_x_div_sqrtdelta:

$\forall$ (*x4*::*real*) (*x5*::*real*) (*x3*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *sol_euler_x_div_sqrtdelta x1 x2 x3 x4 x5 x6* = *LET* (λ*a*::*real*. *LET_END* (*matan* (*delta_x x1 x2 x3 x4 x5 x6* / (*real_of_nat* (*4*::*nat*) ∗ a^2)) / *a*)) (*sqrt* (*x1* ∗ (*x2* ∗ *x3*)) + (*sqrt x1* ∗ ((*x2* + (*x3* − *x4*)) / *real_of_nat* (*2*::*nat*)) + (*sqrt x2* ∗ ((*x1* + (*x3* − *x5*)) / *real_of_nat* (*2*::*nat*)) + *sqrt x3* ∗ ((*x1* + (*x2* − *x6*)) / *real_of_nat* (*2*::*nat*)))))

thm Sphere.sol_euler246_x_div_sqrtdelta:

sol_euler246_x_div_sqrtdelta = *rotate4 sol_euler_x_div_sqrtdelta*

thm Sphere.sol_euler345_x_div_sqrtdelta:

sol_euler345_x_div_sqrtdelta = *rotate5 sol_euler_x_div_sqrtdelta*

thm Sphere.sol_euler156_x_div_sqrtdelta:

sol_euler156_x_div_sqrtdelta = *rotate6 sol_euler_x_div_sqrtdelta*

thm DEF_dih_x_div_sqrtdelta_posbranch:

dih_x_div_sqrtdelta_posbranch = (λ(*_1971850*::*real*) (*_1971851*::*real*) (*_1971852*::*real*) (*_1971853*::*real*) (*_1971854*::*real*) *_1971855*::*real*. *LET* (λ*d_x4*::*real*. *LET_END* (*LET* (λ*d*::*real*. *LET_END* (*sqrt* (*real_of_nat* (*4*::*nat*) ∗ *_1971850*) / *d_x4* ∗ *matan* (*real_of_nat* (*4*::*nat*) ∗ (*_1971850* ∗ *d*) / d_x4^2))) (*delta_x _1971850 _1971851 _1971852 _1971853 _1971854 _1971855*))) (*delta_x4 _1971850 _1971851 _1971852 _1971853 _1971854 _1971855*))

thm Sphere.dih_x_div_sqrtdelta_posbranch:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *LET* (λ*d_x4*::*real*. *LET_END* (*LET* (λ*d*::*real*. *LET_END* (*sqrt* (*real_of_nat* (*4*::*nat*) ∗ *x1*) / *d_x4* ∗ *matan* (*real_of_nat* (*4*::*nat*) ∗ (*x1* ∗ *d*) / *d_x4*^{2}))) (*delta_x x1 x2 x3 x4 x5 x6*))) (*delta_x4 x1 x2 x3 x4 x5 x6*)

thm DEF_ldih_x_div_sqrtdelta_posbranch:

ldih_x_div_sqrtdelta_posbranch = (λ(*_1971910*::*real*) (*_1971911*::*real*) (*_1971912*::*real*) (*_1971913*::*real*) (*_1971914*::*real*) *_1971915*::*real*. *lfun* (*sqrt _1971910* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x_div_sqrtdelta_posbranch _1971910 _1971911 _1971912 _1971913 _1971914 _1971915*)

thm Sphere.ldih_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x1* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm Sphere.ldih2_x_div_sqrtdelta_posbranch:

ldih2_x_div_sqrtdelta_posbranch = *rotate2 ldih_x_div_sqrtdelta_posbranch*

thm Sphere.ldih3_x_div_sqrtdelta_posbranch:

ldih3_x_div_sqrtdelta_posbranch = *rotate3 ldih_x_div_sqrtdelta_posbranch*

thm Functional_equation.functional_ldih5_x_div_sqrtdelta_posbranch:

ldih5_x_div_sqrtdelta_posbranch = *rotate5 ldih_x_div_sqrtdelta_posbranch*

thm Sphere.ldih6_x_div_sqrtdelta_posbranch:

ldih6_x_div_sqrtdelta_posbranch = *rotate6 ldih_x_div_sqrtdelta_posbranch*

thm Sphere.dih3_x_div_sqrtdelta_posbranch:

dih3_x_div_sqrtdelta_posbranch = *rotate3 dih_x_div_sqrtdelta_posbranch*

thm Sphere.dih4_x_div_sqrtdelta_posbranch:

dih4_x_div_sqrtdelta_posbranch = *rotate4 dih_x_div_sqrtdelta_posbranch*

thm Sphere.dih5_x_div_sqrtdelta_posbranch:

dih5_x_div_sqrtdelta_posbranch = *rotate5 dih_x_div_sqrtdelta_posbranch*

thm DEF_lmdih_x_div_sqrtdelta_posbranch:

lmdih_x_div_sqrtdelta_posbranch = (λ(*_1971970*::*real*) (*_1971971*::*real*) (*_1971972*::*real*) (*_1971973*::*real*) (*_1971974*::*real*) *_1971975*::*real*. *lmfun* (*sqrt _1971970* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x_div_sqrtdelta_posbranch _1971970 _1971971 _1971972 _1971973 _1971974 _1971975*)

thm Sphere.lmdih_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *lmdih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *lmfun* (*sqrt x1* / *real_of_nat* (*2*::*nat*)) ∗ *dih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_lmdih2_x_div_sqrtdelta_posbranch:

lmdih2_x_div_sqrtdelta_posbranch = rotate2 lmdih_x_div_sqrtdelta_posbranch

thm Sphere.lmdih2_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *lmdih2_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *rotate2 lmdih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_lmdih3_x_div_sqrtdelta_posbranch:

lmdih3_x_div_sqrtdelta_posbranch = rotate3 lmdih_x_div_sqrtdelta_posbranch

thm Sphere.lmdih3_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *lmdih3_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *rotate3 lmdih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_lmdih5_x_div_sqrtdelta_posbranch:

lmdih5_x_div_sqrtdelta_posbranch = rotate5 lmdih_x_div_sqrtdelta_posbranch

thm Sphere.lmdih5_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *lmdih5_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *rotate5 lmdih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_lmdih6_x_div_sqrtdelta_posbranch:

lmdih6_x_div_sqrtdelta_posbranch = rotate6 lmdih_x_div_sqrtdelta_posbranch

thm Sphere.lmdih6_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *lmdih6_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *rotate6 lmdih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_taum_y1:

taum_y1 = (λ(*_1972270*::*real*) (*_1972271*::*real*) (*_1972272*::*real*) (*_1972273*::*real*) (*_1972274*::*real*) (*_1972275*::*real*) (*_1972276*::*real*) *_1972277*::*real*. *taum* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *_1972270 _1972271 _1972272*)

thm Sphere.taum_y1:

∀ (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*a*::*real*) (*b*::*real*) *y1*::*real*. *taum_y1 a b y1 y2 y3 y4 y5 y6* = *taum* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *a b y1*

thm DEF_taum_y2:

taum_y2 = (λ(*_1972366*::*real*) (*_1972367*::*real*) (*_1972368*::*real*) (*_1972369*::*real*) (*_1972370*::*real*) (*_1972371*::*real*) (*_1972372*::*real*) *_1972373*::*real*. *taum* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *_1972366 _1972367 _1972369*)

thm Sphere.taum_y2:

∀ (*y1*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*a*::*real*) (*b*::*real*) *y2*::*real*. *taum_y2 a b y1 y2 y3 y4 y5 y6* = *taum* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *a b y2*

thm DEF_taum_y1_y2:

taum_y1_y2 = (λ(_1972462::real) (_1972463::real) (_1972464::real) (_1972465::real) (_1972466::real) (_1972467::real) _1972468::real. taum (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) _1972462 _1972463 _1972464)

thm Sphere.taum_y1_y2:

∀ (y3::real) (y4::real) (y5::real) (y6::real) (a::real) (y1::real) y2::real. taum_y1_y2 a y1 y2 y3 y4 y5 y6 = taum (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) a y1 y2

thm DEF_taum_x1:

taum_x1 = (λ(_1972539::real) (_1972540::real) (_1972541::real) (_1972542::real) (_1972543::real) (_1972544::real) (_1972545::real) _1972546::real. taum_y1 _1972539 _1972540 (sqrt _1972541) (sqrt _1972542) (sqrt _1972543) (sqrt _1972544) (sqrt _1972545) (sqrt _1972546))

thm Sphere.taum_x1:

∀ (a::real) (b::real) (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. taum_x1 a b x1 x2 x3 x4 x5 x6 = taum_y1 a b (sqrt x1) (sqrt x2) (sqrt x3) (sqrt x4) (sqrt x5) (sqrt x6)

thm DEF_taum_x2:

taum_x2 = (λ(_1972635::real) (_1972636::real) (_1972637::real) (_1972638::real) (_1972639::real) (_1972640::real) (_1972641::real) _1972642::real. taum_y2 _1972635 _1972636 (sqrt _1972637) (sqrt _1972638) (sqrt _1972639) (sqrt _1972640) (sqrt _1972641) (sqrt _1972642))

thm Sphere.taum_x2:

∀ (a::real) (b::real) (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. taum_x2 a b x1 x2 x3 x4 x5 x6 = taum_y2 a b (sqrt x1) (sqrt x2) (sqrt x3) (sqrt x4) (sqrt x5) (sqrt x6)

thm DEF_taum_x1_x2:

taum_x1_x2 = (λ(_1972731::real) (_1972732::real) (_1972733::real) (_1972734::real) (_1972735::real) (_1972736::real) _1972737::real. taum_y1_y2 _1972731 (sqrt _1972732) (sqrt _1972733) (sqrt _1972734) (sqrt _1972735) (sqrt _1972736) (sqrt _1972737))

thm Sphere.taum_x1_x2:

∀ (a::real) (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. taum_x1_x2 a x1 x2 x3 x4 x5 x6 = taum_y1_y2 a (sqrt x1) (sqrt x2) (sqrt x3) (sqrt x4) (sqrt x5) (sqrt x6)

thm DEF_arclength_y1:

arclength_y1 = (λ(_1972808::real) (_1972809::real) (_1972810::real) (_1972811::real) (_1972812::real) (_1972813::real) (_1972814::real) _1972815::real. arclength _1972810 _1972808 _1972809)

thm Sphere.arclength_y1:

$\forall$ (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*y1*::*real*) (*a*::*real*) *b*::*real*. *arclength_y1 a b y1 y2 y3 y4 y5 y6* = *arclength y1 a b*

thm DEF_arclength_y2:

arclength_y2 = (λ(*_1972904*::*real*) (*_1972905*::*real*) (*_1972906*::*real*) (*_1972907*::*real*) (*_1972908*::*real*) (*_1972909*::*real*) (*_1972910*::*real*) *_1972911*::*real*. *arclength _1972907 _1972904 _1972905*)

thm Sphere.arclength_y2:

$\forall$ (*y1*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*y2*::*real*) (*a*::*real*) *b*::*real*. *arclength_y2 a b y1 y2 y3 y4 y5 y6* = *arclength y2 a b*

thm DEF_arclength_x1:

arclength_x1 = (λ(*_1973000*::*real*) (*_1973001*::*real*) (*_1973002*::*real*) (*_1973003*::*real*) (*_1973004*::*real*) (*_1973005*::*real*) (*_1973006*::*real*) *_1973007*::*real*. *arclength_y1 _1973000 _1973001* (*sqrt _1973002*) (*sqrt _1973003*) (*sqrt _1973004*) (*sqrt _1973005*) (*sqrt _1973006*) (*sqrt _1973007*))

thm Sphere.arclength_x1:

$\forall$ (*a*::*real*) (*b*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *arclength_x1 a b x1 x2 x3 x4 x5 x6* = *arclength_y1 a b* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_arclength_x2:

arclength_x2 = (λ(*_1973096*::*real*) (*_1973097*::*real*) (*_1973098*::*real*) (*_1973099*::*real*) (*_1973100*::*real*) (*_1973101*::*real*) (*_1973102*::*real*) *_1973103*::*real*. *arclength_y2 _1973096 _1973097* (*sqrt _1973098*) (*sqrt _1973099*) (*sqrt _1973100*) (*sqrt _1973101*) (*sqrt _1973102*) (*sqrt _1973103*))

thm Sphere.arclength_x2:

$\forall$ (*a*::*real*) (*b*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *arclength_x2 a b x1 x2 x3 x4 x5 x6* = *arclength_y2 a b* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm Sphere.arc_hhn:

arc_hhn = *arclength* (*real_of_nat* (*2*::*nat*) * *h0*) (*real_of_nat* (*2*::*nat*) * *h0*) (*real_of_nat* (*2*::*nat*))

thm DEF_asn797k:

asn797k = (λ(*_1973192*::*real*) (*_1973193*::*?'e*::*type*) (*_1973194*::*?'d*::*type*) (*_1973195*::*?'c*::*type*) (*_1973196*::*?'b*::*type*) *_1973197*::*?'a*::*type*. *_1973192* * *asn* (*cos* (*DECIMAL* (*797*::*nat*) (*1000*::*nat*)) * *sin* (*pi* / *_1973192*)))

thm Sphere.asn797k:

∀ (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) (*x6*::*?'a*::*type*) *k*::*real*. *asn797k k x2 x3 x4 x5 x6* = *k* ∗ *asn* (*cos* (*DECIMAL* (*797*::*nat*) (*1000*::*nat*)) ∗ *sin* (*pi* / *k*))

thm DEF_asnFnhk:

asnFnhk = (λ(*_1973252*::*real*) (*_1973253*::*real*) (*_1973254*::*?'d*::*type*) (*_1973255*::*?'c*::*type*) (*_1973256*::*?'b*::*type*) *_1973257*::*?'a*::*type*. *_1973253* ∗ *asn* ((*_1973252* ∗ (*sqrt3* / *DECIMAL* (*40*::*nat*) (*10*::*nat*)) + *sqrt* ((*1*::*real*) − (*_1973252* / *real_of_nat* (*2*::*nat*))^2) / *real_of_nat* (*2*::*nat*)) ∗ *sin* (*pi* / *_1973253*)))

thm Sphere.asnFnhk:

∀ (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) (*x6*::*?'a*::*type*) (*h*::*real*) *k*::*real*. *asnFnhk h k x3 x4 x5 x6* = *k* ∗ *asn* ((*h* ∗ (*sqrt3* / *DECIMAL* (*40*::*nat*) (*10*::*nat*)) + *sqrt* ((*1*::*real*) − (*h* / *real_of_nat* (*2*::*nat*))^2) / *real_of_nat* (*2*::*nat*)) ∗ *sin* (*pi* / *k*))

thm DEF_lfun_y1:

lfun_y1 = (λ(*_1973312*::*real*) (*_1973313*::*real*) (*_1973314*::*real*) (*_1973315*::*real*) (*_1973316*::*real*) *_1973317*::*real*. *lfun _1973312*)

thm Sphere.lfun_y1:

∀ (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) *y1*::*real*. *lfun_y1 y1 y2 y3 y4 y5 y6* = *lfun y1*

thm DEF_acs_sqrt_x1_d4:

acs_sqrt_x1_d4 = (λ(*_1973372*::*real*) (*_1973373*::*real*) (*_1973374*::*real*) (*_1973375*::*real*) (*_1973376*::*real*) *_1973377*::*real*. *acs* (*sqrt _1973372* / *real_of_nat* (*4*::*nat*)))

thm Sphere.acs_sqrt_x1_d4:

∀ (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) *x1*::*real*. *acs_sqrt_x1_d4 x1 x2 x3 x4 x5 x6* = *acs* (*sqrt x1* / *real_of_nat* (*4*::*nat*))

thm DEF_acs_sqrt_x2_d4:

acs_sqrt_x2_d4 = (λ(*_1973432*::*real*) (*_1973433*::*real*) (*_1973434*::*real*) (*_1973435*::*real*) (*_1973436*::*real*) *_1973437*::*real*. *acs* (*sqrt _1973433* / *real_of_nat* (*4*::*nat*)))

thm Sphere.acs_sqrt_x2_d4:

∀ (*x1*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) *x2*::*real*. *acs_sqrt_x2_d4 x1 x2 x3 x4 x5 x6* = *acs* (*sqrt x2* / *real_of_nat* (*4*::*nat*))

thm DEF_arclength_x_123:

arclength_x_123 = (λ(*_1973492*::*real*) (*_1973493*::*real*) (*_1973494*::*real*) (*_1973495*::*real*) (*_1973496*::*real*) *_1973497*::*real*. *arclength* (*sqrt _1973492*) (*sqrt _1973493*) (*sqrt _1973494*))

thm Sphere.arclength_x_123:

∀ (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *arclength_x_123 x1 x2 x3 x4 x5 x6* = *arclength* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*)

thm DEF_tame_table_d:

tame_table_d = (λ(*_1973552*::*nat*) *_1973553*::*nat*. *if* (*3*::*nat*) < *_1973552* + (*2*::*nat*) ∗ *_1973553 then DECIMAL* (*103*::*nat*) (*1000*::*nat*) ∗ (*real_of_nat* (*2*::*nat*) − *real_of_nat _1973553*) + *DECIMAL* (*2759*::*nat*) (*10000*::*nat*) ∗ (*real_of_nat _1973552* + (*real_of_nat* (*2*::*nat*) ∗ *real_of_nat _1973553* − *real_of_nat* (*4*::*nat*))) *else DECIMAL* (*0*::*nat*) (*10*::*nat*))

thm Sphere.tame_table_d:

∀ (*r*::*nat*) *s*::*nat*. *tame_table_d r s* = (*if* (*3*::*nat*) < *r* + (*2*::*nat*) ∗ *s then DECIMAL* (*103*::*nat*) (*1000*::*nat*) ∗ (*real_of_nat* (*2*::*nat*) − *real_of_nat s*) + *DECIMAL* (*2759*::*nat*) (*10000*::*nat*) ∗ (*real_of_nat r* + (*real_of_nat* (*2*::*nat*) ∗ *real_of_nat s* − *real_of_nat* (*4*::*nat*))) *else DECIMAL* (*0*::*nat*) (*10*::*nat*))

thm Sphere.ydodec:

ydodec = (*SOME y*::*real*. *sol_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *y y y* = *pi* / *DECIMAL* (*50*::*nat*) (*10*::*nat*))

thm DEF_fdodec:

fdodec = (λ(*_1973564*::*real*) (*_1973565*::*real*) *_1973566*::*real*. *DECIMAL* (*60*::*nat*) (*10*::*nat*) ∗ *volR* (*DECIMAL* (*10*::*nat*) (*10*::*nat*)) (*eta_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *_1973564*) (*sqrt* (*rad2_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *_1973564 _1973564 _1973564*)) + (*_1973565* ∗ *sol_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *_1973564 _1973564 _1973564* + *DECIMAL* (*30*::*nat*) (*10*::*nat*) ∗ (*_1973566* ∗ *dih_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *_1973564 _1973564 _1973564*)))

thm Sphere.fdodec:

∀ (*a*::*real*) (*b*::*real*) *y*::*real*. *fdodec y a b* = *DECIMAL* (*60*::*nat*) (*10*::*nat*) ∗ *volR* (*DECIMAL* (*10*::*nat*) (*10*::*nat*)) (*eta_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *y*) (*sqrt* (*rad2_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *y y y*)) + (*a* ∗ *sol_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *y y y* + *DECIMAL* (*30*::*nat*) (*10*::*nat*) ∗ (*b* ∗ *dih_y* (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) (*DECIMAL* (*20*::*nat*) (*10*::*nat*)) *y y y*))

thm DEF_dfdodec:

dfdodec = (λ(*_1973585*::*real*) *_1973586*::*real*. *SOME d*::*real*. *has_real_derivative* (λ*t*::*real*. *fdodec t _1973585 _1973586*) *d* (*atreal ydodec*))

thm Sphere.dfdodec:

$\forall (a::real)\ b::real.\ dfdodec\ a\ b = (SOME\ d::real.\ has\_real\_derivative\ (\lambda t::real.\ fdodec\ t\ a\ b)\ d\ (atreal\ ydodec))$

thm Sphere.abdodec:

$abdodec = (SOME\ ab::real \times real.\ fdodec\ ydodec\ (fst\ ab)\ (snd\ ab) = (0::real) \wedge dfdodec\ (fst\ ab)\ (snd\ ab) = (0::real))$

thm Sphere.adodec:

$adodec = fst\ abdodec$

thm Sphere.bdodec:

$bdodec = snd\ abdodec$

thm DEF_surfR:

$surfR = (\lambda(\_1973597::real)\ (\_1973598::real)\ \_1973599::real.\ DECIMAL\ (30::nat)\ (10::nat) * (volR\ \_1973597\ \_1973598\ \_1973599\ /\ \_1973597))$

thm Sphere.surfR:

$\forall (b::real)\ (c::real)\ a::real.\ surfR\ a\ b\ c = DECIMAL\ (30::nat)\ (10::nat) * (volR\ a\ b\ c\ /\ a)$

thm DEF_surfRy:

$surfRy = (\lambda(\_1973618::real)\ (\_1973619::real)\ \_1973620::real.\ surfR\ (\_1973618\ /\ real\_of\_nat\ (2::nat))\ (eta\_y\ \_1973618\ \_1973619\ \_1973620))$

thm Sphere.surfRy:

$\forall (y1::real)\ (y2::real)\ (y6::real)\ c::real.\ surfRy\ y1\ y2\ y6\ c = surfR\ (y1\ /\ real\_of\_nat\ (2::nat))\ (eta\_y\ y1\ y2\ y6)\ c$

thm DEF_surfRdyc2:

$surfRdyc2 = (\lambda(\_1973650::real)\ (\_1973651::real)\ (\_1973652::real)\ \_1973653::real.\ surfRy\ \_1973650\ \_1973651\ \_1973652\ (sqrt\ \_1973653) + surfRy\ \_1973651\ \_1973650\ \_1973652\ (sqrt\ \_1973653))$

thm Sphere.surfRdyc2:

$\forall (y2::real)\ (y1::real)\ (y6::real)\ c2::real.\ surfRdyc2\ y1\ y2\ y6\ c2 = surfRy\ y1\ y2\ y6\ (sqrt\ c2) + surfRy\ y2\ y1\ y6\ (sqrt\ c2)$

thm DEF_surfy:

$surfy = (\lambda(\_1973682::real)\ (\_1973683::real)\ (\_1973684::real)\ (\_1973685::real)\ (\_1973686::real)\ \_1973687::real.\ LET\ (\lambda c::real.\ LET\_END\ (surfRy\ \_1973682\ \_1973683\ \_1973687\ c + (surfRy\ \_1973683\ \_1973682\ \_1973687\ c + (surfRy\ \_1973683\ \_1973684\ \_1973685\ c + (surfRy\ \_1973684\ \_1973683\ \_1973685\ c + (surfRy\ \_1973684\ \_1973682\ \_1973686\ c + surfRy\ \_1973682\ \_1973684\ \_1973686\ c))))))\ (sqrt\ (rad2\_y\ \_1973682\ \_1973683\ \_1973684\ \_1973685\ \_1973686\ \_1973687)))$

thm Sphere.surfy:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *surfy y1 y2 y3 y4 y5 y6* = *LET* (λ*c*::*real*. *LET_END* (*surfRy y1 y2 y6 c* + (*surfRy y2 y1 y6 c* + (*surfRy y2 y3 y4 c* + (*surfRy y3 y2 y4 c* + (*surfRy y3 y1 y5 c* + *surfRy y1 y3 y5 c*)))))) (*sqrt* (*rad2_y y1 y2 y3 y4 y5 y6*))

thm DEF_surf_x:

surf_x = (λ(*_1973742*::*real*) (*_1973743*::*real*) (*_1973744*::*real*) (*_1973745*::*real*) (*_1973746*::*real*) *_1973747*::*real*. *surfy* (*sqrt _1973742*) (*sqrt _1973743*) (*sqrt _1973744*) (*sqrt _1973745*) (*sqrt _1973746*) (*sqrt _1973747*))

thm Sphere.surf_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *surf_x x1 x2 x3 x4 x5 x6* = *surfy* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_surfR126d:

surfR126d = (λ(*_1973802*::*real*) (*_1973803*::*real*) (*_1973804*::*real*) (*_1973805*::*real*) (*_1973806*::*real*) (*_1973807*::*real*) *_1973808*::*real*. *surfRdyc2* (*sqrt _1973803*) (*sqrt _1973804*) (*sqrt _1973808*) *_1973802*)

thm Sphere.surfR126d:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) (*x6*::*real*) *c2*::*real*. *surfR126d c2 x1 x2 x3 x4 x5 x6* = *surfRdyc2* (*sqrt x1*) (*sqrt x2*) (*sqrt x6*) *c2*

thm DEF_eta2_126:

eta2_126 = (λ(*_1973879*::*real*) (*_1973880*::*real*) (*_1973881*::*real*) (*_1973882*::*real*) (*_1973883*::*real*) *_1973884*::*real*. (*eta_y* (*sqrt _1973879*) (*sqrt _1973880*) (*sqrt _1973884*))^2)

thm Sphere.eta2_126:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *eta2_126 x1 x2 x3 x4 x5 x6* = (*eta_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x6*))^2

thm DEF_eta2_135:

eta2_135 = (λ(*_1973939*::*real*) (*_1973940*::*real*) (*_1973941*::*real*) (*_1973942*::*real*) (*_1973943*::*real*) *_1973944*::*real*. (*eta_y* (*sqrt _1973939*) (*sqrt _1973941*) (*sqrt _1973943*))^2)

thm Sphere.eta2_135:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *eta2_135 x1 x2 x3 x4 x5 x6* = (*eta_y* (*sqrt x1*) (*sqrt x3*) (*sqrt x5*))^2

thm DEF_eta2_456:

eta2_456 = (λ(*_1973999*::*real*) (*_1974000*::*real*) (*_1974001*::*real*) (*_1974002*::*real*) (*_1974003*::*real*) *_1974004*::*real*. (*eta_y* (*sqrt _1974002*) (*sqrt _1974003*) (*sqrt _1974004*))^2)

thm Sphere.eta2_456:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *eta2_456 x1 x2 x3 x4 x5 x6* = (*eta_y* (*sqrt x4*) (*sqrt x5*) (*sqrt x6*))^2

thm DEF_num1:

num1 = (λ(*_1974059*::*real*) (*_1974060*::*real*) (*_1974061*::*real*) (*_1974062*::*real*) (*_1974063*::*real*) *_1974064*::*real*. − *real_of_nat* (*4*::*nat*) ∗ (*_1974062*^{2} ∗ *_1974059* + (*real_of_nat* (*8*::*nat*) ∗ ((*_1974063* − *_1974064*) ∗ (*_1974060* − *_1974061*)) − *_1974062* ∗ (*real_of_nat* (*16*::*nat*) ∗ *_1974059* + ((*_1974063* − *real_of_nat* (*8*::*nat*)) ∗ *_1974060* + (*_1974064* − *real_of_nat* (*8*::*nat*)) ∗ *_1974061*)))))

thm Sphere.num1:

$\forall$ (*a2*::*real*) (*e1*::*real*) (*b2*::*real*) (*e2*::*real*) (*c2*::*real*) *e3*::*real*. *num1 e1 e2 e3 a2 b2 c2* = − *real_of_nat* (*4*::*nat*) ∗ (*a2*^{2} ∗ *e1* + (*real_of_nat* (*8*::*nat*) ∗ ((*b2* − *c2*) ∗ (*e2* − *e3*)) − *a2* ∗ (*real_of_nat* (*16*::*nat*) ∗ *e1* + ((*b2* − *real_of_nat* (*8*::*nat*)) ∗ *e2* + (*c2* − *real_of_nat* (*8*::*nat*)) ∗ *e3*))))

thm DEF_num2:

num2 = (λ(*_1974119*::*real*) (*_1974120*::*real*) (*_1974121*::*real*) (*_1974122*::*real*) (*_1974123*::*real*) *_1974124*::*real*. *real_of_nat* (*8*::*nat*) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*_1974122*$^{5::nat}$ ∗ *_1974119*) + (− *real_of_nat* (*256*::*nat*) ∗ ((*_1974123* + − (*1*::*real*) ∗ *_1974124*)$^{3::nat}$ ∗ (*_1974120* + − (*1*::*real*) ∗ *_1974121*)) + (− (*1*::*real*) ∗ (*_1974122*$^{3::nat}$ ∗ (*real_of_nat* (*2*::*nat*) ∗ ((− *real_of_nat* (*256*::*nat*) + (*_1974123*^{2} + (− *real_of_nat* (*2*::*nat*) ∗ (*_1974123* ∗ *_1974124*) + *_1974124*^{2}))) ∗ *_1974119*) + ((*_1974123*^{2} ∗ (− *real_of_nat* (*8*::*nat*) + *_1974124*) + (− *real_of_nat* (*16*::*nat*) ∗ (*_1974123* ∗ (*real_of_nat* (*3*::*nat*) + *_1974124*)) + *real_of_nat* (*16*::*nat*) ∗ (*real_of_nat* (*16*::*nat*) + *real_of_nat* (*9*::*nat*) ∗ *_1974124*))) ∗ *_1974120* + (*_1974123* ∗ (*real_of_nat* (*144*::*nat*) + (− *real_of_nat* (*16*::*nat*) ∗ *_1974124* + *_1974124*^{2})) + − *real_of_nat* (*8*::*nat*) ∗ (− *real_of_nat* (*32*::*nat*) + (*real_of_nat* (*6*::*nat*) ∗ *_1974124* + *_1974124*^{2}))) ∗ *_1974121*))) + (*_1974122*$^{4::nat}$ ∗ (− *real_of_nat* (*64*::*nat*) ∗ *_1974119* + − *real_of_nat* (*6*::*nat*) ∗ ((− *real_of_nat* (*8*::*nat*) + *_1974123*) ∗ *_1974120* + (− *real_of_nat* (*8*::*nat*) + *_1974124*) ∗ *_1974121*)) + (− *real_of_nat* (*2*::*nat*) ∗ (*_1974122*^{2} ∗ ((*_1974123* + − (*1*::*real*) ∗ *_1974124*) ∗ (*_1974123*^{2} ∗ *_1974120* + (*real_of_nat* (*8*::*nat*) ∗ (*_1974124* ∗ (*real_of_nat* (*4*::*nat*) ∗ *_1974119* + (*real_of_nat* (*9*::*nat*) ∗ *_1974120* + − *real_of_nat* (*7*::*nat*) ∗ *_1974121*))) + (*real_of_nat* (*384*::*nat*) ∗ (*_1974120* + − (*1*::*real*) ∗ *_1974121*) + (− (*1*::*real*) ∗ (*_1974124*^{2} ∗ *_1974121*) + *_1974123* ∗ (− *real_of_nat* (*32*::*nat*) ∗ *_1974119* + ((*real_of_nat* (*56*::*nat*) + − *real_of_nat* (*9*::*nat*) ∗ *_1974124*) ∗ *_1974120* + *real_of_nat* (*9*::*nat*) ∗ ((− *real_of_nat* (*8*::*nat*) + *_1974124*) ∗ *_1974121*))))))))) + *real_of_nat* (*16*::*nat*) ∗ (*_1974122* ∗ ((*_1974123* + − (*1*::*real*) ∗ *_1974124*) ∗ (*_1974123*^{2} ∗ (*_1974120* + − *real_of_nat* (*3*::*nat*) ∗ *_1974121*) + (− *real_of_nat* (*4*::*nat*) ∗ (*_1974123* ∗ (*real_of_nat* (*8*::*nat*) ∗ *_1974119* + ((− *real_of_nat* (*20*::*nat*) + *real_of_nat* (*3*::*nat*) ∗ *_1974124*) ∗ *_1974120* + − *real_of_nat* (*3*::*nat*) ∗ ((− *real_of_nat* (*4*::*nat*) + *_1974124*) ∗ *_1974121*)))) + *_1974124* ∗ (*real_of_nat* (*32*::*nat*) ∗ *_1974119* + (*real_of_nat* (*3*::*nat*) ∗ ((*real_of_nat* (*16*::*nat*) + *_1974124*) ∗ *_1974120*) + − (*1*::*real*) ∗ ((*real_of_nat* (*80*::*nat*) + *_1974124*) ∗ *_1974121*)))))))))))))

thm Sphere.num2:

∀ (*a2*::*real*) (*b2*::*real*) (*e1*::*real*) (*e2*::*real*) (*c2*::*real*) *e3*::*real*. *num2 e1 e2 e3 a2 b2 c2* = *real_of_nat* (*8*::*nat*) ∗ (*real_of_nat* (*2*::*nat*) ∗ ($a2^{5::nat}$ ∗ *e1*) + (− *real_of_nat* (*256*::*nat*) ∗ ((*b2* + − (*1*::*real*) ∗ *c2*)$^{3::nat}$ ∗ (*e2* + − (*1*::*real*) ∗ *e3*)) + (− (*1*::*real*) ∗ ($a2^{3::nat}$ ∗ (*real_of_nat* (*2*::*nat*) ∗ ((− *real_of_nat* (*256*::*nat*) + ($b2^2$ + (− *real_of_nat* (*2*::*nat*) ∗ (*b2* ∗ *c2*) + $c2^2$))) ∗ *e1*) + (($b2^2$ ∗ (− *real_of_nat* (*8*::*nat*) + *c2*) + (− *real_of_nat* (*16*::*nat*) ∗ (*b2* ∗ (*real_of_nat* (*3*::*nat*) + *c2*)) + *real_of_nat* (*16*::*nat*) ∗ (*real_of_nat* (*16*::*nat*) + *real_of_nat* (*9*::*nat*) ∗ *c2*))) ∗ *e2* + (*b2* ∗ (*real_of_nat* (*144*::*nat*) + (− *real_of_nat* (*16*::*nat*) ∗ *c2* + $c2^2$)) + − *real_of_nat* (*8*::*nat*) ∗ (− *real_of_nat* (*32*::*nat*) + (*real_of_nat* (*6*::*nat*) ∗ *c2* + $c2^2$))) ∗ *e3*))) + ($a2^{4::nat}$ ∗ (− *real_of_nat* (*64*::*nat*) ∗ *e1* + − *real_of_nat* (*6*::*nat*) ∗ ((− *real_of_nat* (*8*::*nat*) + *b2*) ∗ *e2* + (− *real_of_nat* (*8*::*nat*) + *c2*) ∗ *e3*)) + (− *real_of_nat* (*2*::*nat*) ∗ ($a2^2$ ∗ ((*b2* + − (*1*::*real*) ∗ *c2*) ∗ ($b2^2$ ∗ *e2* + (*real_of_nat* (*8*::*nat*) ∗ (*c2* ∗ (*real_of_nat* (*4*::*nat*) ∗ *e1* + (*real_of_nat* (*9*::*nat*) ∗ *e2* + − *real_of_nat* (*7*::*nat*) ∗ *e3*))) + (*real_of_nat* (*384*::*nat*) ∗ (*e2* + − (*1*::*real*) ∗ *e3*) + (− (*1*::*real*) ∗ ($c2^2$ ∗ *e3*) + *b2* ∗ (− *real_of_nat* (*32*::*nat*) ∗ *e1* + ((*real_of_nat* (*56*::*nat*) + − *real_of_nat* (*9*::*nat*) ∗ *c2*) ∗ *e2* + *real_of_nat* (*9*::*nat*) ∗ ((− *real_of_nat* (*8*::*nat*) + *c2*) ∗ *e3*))))))))) + *real_of_nat* (*16*::*nat*) ∗ (*a2* ∗ ((*b2* + − (*1*::*real*) ∗ *c2*) ∗ ($b2^2$ ∗ (*e2* + − *real_of_nat* (*3*::*nat*) ∗ *e3*) + (− *real_of_nat* (*4*::*nat*) ∗ (*b2* ∗ (*real_of_nat* (*8*::*nat*) ∗ *e1* + ((− *real_of_nat* (*20*::*nat*) + *real_of_nat* (*3*::*nat*) ∗ *c2*) ∗ *e2* + − *real_of_nat* (*3*::*nat*) ∗ ((− *real_of_nat* (*4*::*nat*) + *c2*) ∗ *e3*)))) + *c2* ∗ (*real_of_nat* (*32*::*nat*) ∗ *e1* + (*real_of_nat* (*3*::*nat*) ∗ ((*real_of_nat* (*16*::*nat*) + *c2*) ∗ *e2*) + − (*1*::*real*) ∗ ((*real_of_nat* (*80*::*nat*) + *c2*) ∗ *e3*)))))))))))

thm DEF_rat1:

rat1 = (λ(*_1974179*::*real*) (*_1974180*::*real*) (*_1974181*::*real*) (*_1974182*::*real*) (*_1974183*::*real*) *_1974184*::*real*. *num1 _1974179 _1974180 _1974181 _1974182 _1974183 _1974184* / (*sqrt* (*delta_x* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) *_1974182 _1974183 _1974184*) ∗ (*sqrt _1974182* ∗ (*real_of_nat* (*16*::*nat*) − *_1974182*))))

thm Sphere.rat1:

∀ (*e1*::*real*) (*e2*::*real*) (*e3*::*real*) (*b2*::*real*) (*c2*::*real*) *a2*::*real*. *rat1 e1 e2 e3 a2 b2 c2* = *num1 e1 e2 e3 a2 b2 c2* / (*sqrt* (*delta_x* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) *a2 b2 c2*) ∗ (*sqrt a2* ∗ (*real_of_nat* (*16*::*nat*) − *a2*)))

thm DEF_rat2:

rat2 = (λ(*_1974239*::*real*) (*_1974240*::*real*) (*_1974241*::*real*) (*_1974242*::*real*) (*_1974243*::*real*) *_1974244*::*real*. *num2 _1974239 _1974240 _1974241 _1974242 _1974243 _1974244* / ((*sqrt* (*delta_x* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) *_1974242 _1974243 _1974244*))$^{3::nat}$ ∗ (*_1974242* ∗ (*real_of_nat* (*16*::*nat*) − *_1974242*)^2)))

thm Sphere.rat2:

∀ (e1::real) (e2::real) (e3::real) (b2::real) (c2::real) a2::real. rat2 e1 e2 e3 a2 b2 c2 = num2 e1 e2 e3 a2 b2 c2 / ((sqrt (delta_x (real_of_nat (4::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) a2 b2 c2))$^{3::nat}$ ∗ (a2 ∗ (real_of_nat (16::nat) − a2)^2))

thm DEF_num_combo1:

num_combo1 = (λ(_1974299::real) (_1974300::real) (_1974301::real) (_1974302::real) (_1974303::real) _1974304::real. real_of_nat (2::nat) / real_of_nat (25::nat) ∗ (− real_of_nat (2::nat) ∗ (_1974302$^{5::nat}$ ∗ _1974299) + (real_of_nat (256::nat) ∗ ((_1974303 + − (1::real) ∗ _1974304)$^{3::nat}$ ∗ (_1974300 + − (1::real) ∗ _1974301)) + (_1974302$^{3::nat}$ ∗ (real_of_nat (2::nat) ∗ ((− real_of_nat (256::nat) + (_1974303^2 + (− real_of_nat (2::nat) ∗ (_1974303 ∗ _1974304) + _1974304^2))) ∗ _1974299) + ((_1974303^2 ∗ (− real_of_nat (8::nat) + _1974304) + (− real_of_nat (16::nat) ∗ (_1974303 ∗ (real_of_nat (3::nat) + _1974304)) + real_of_nat (16::nat) ∗ (real_of_nat (16::nat) + real_of_nat (9::nat) ∗ _1974304))) ∗ _1974300 + (_1974303 ∗ (real_of_nat (144::nat) + (− real_of_nat (16::nat) ∗ _1974304 + _1974304^2)) + − real_of_nat (8::nat) ∗ (− real_of_nat (32::nat) + (real_of_nat (6::nat) ∗ _1974304 + _1974304^2))) ∗ _1974301)) + (real_of_nat (2::nat) ∗ (_1974302$^{4::nat}$ ∗ (real_of_nat (32::nat) ∗ _1974299 + real_of_nat (3::nat) ∗ ((− real_of_nat (8::nat) + _1974303) ∗ _1974300 + (− real_of_nat (8::nat) + _1974304) ∗ _1974301))) + (real_of_nat (200::nat) ∗ (_1974302^2 ∗ _1974299 + (real_of_nat (8::nat) ∗ ((_1974303 + − (1::real) ∗ _1974304) ∗ (_1974300 + − (1::real) ∗ _1974301)) + − (1::real) ∗ (_1974302 ∗ (real_of_nat (16::nat) ∗ _1974299 + ((− real_of_nat (8::nat) + _1974303) ∗ _1974300 + (− real_of_nat (8::nat) + _1974304) ∗ _1974301)))))^2 + (real_of_nat (2::nat) ∗ (_1974302^2 ∗ ((_1974303 + − (1::real) ∗ _1974304) ∗ (_1974303^2 ∗ _1974300 + (real_of_nat (8::nat) ∗ (_1974304 ∗ (real_of_nat (4::nat) ∗ _1974299 + (real_of_nat (9::nat) ∗ _1974300 + − real_of_nat (7::nat) ∗ _1974301))) + (real_of_nat (384::nat) ∗ (_1974300 + − (1::real) ∗ _1974301) + (− (1::real) ∗ (_1974304^2 ∗ _1974301) + _1974303 ∗ (− real_of_nat (32::nat) ∗ _1974299 + ((real_of_nat (56::nat) + − real_of_nat (9::nat) ∗ _1974304) ∗ _1974300 + real_of_nat (9::nat) ∗ ((− real_of_nat (8::nat) + _1974304) ∗ _1974301))))))))) + − real_of_nat (16::nat) ∗ (_1974302 ∗ ((_1974303 + − (1::real) ∗ _1974304) ∗ (_1974303^2 ∗ (_1974300 + − real_of_nat (3::nat) ∗ _1974301) + (− real_of_nat (4::nat) ∗ (_1974303 ∗ (real_of_nat (8::nat) ∗ _1974299 + ((− real_of_nat (20::nat) + real_of_nat (3::nat) ∗ _1974304) ∗ _1974300 + − real_of_nat (3::nat) ∗ ((− real_of_nat (4::nat) + _1974304) ∗ _1974301)))) + _1974304 ∗ (real_of_nat (32::nat) ∗ _1974299 + (real_of_nat (3::nat) ∗ ((real_of_nat (16::nat) + _1974304) ∗ _1974300) + − (1::real) ∗ ((real_of_nat (80::nat) + _1974304) ∗ _1974301))))))))))))))

thm Sphere.num_combo1:

∀ (a2::real) (b2::real) (e1::real) (e2::real) (c2::real) e3::real. num_combo1 e1 e2 e3 a2 b2 c2 = real_of_nat (2::nat) / real_of_nat (25::nat) ∗ (− real_of_nat

(2::nat) ∗ (a2$^{5::nat}$ ∗ e1) + (real_of_nat (256::nat) ∗ ((b2 + − (1::real) ∗ c2)$^{3::nat}$ ∗ (e2 + − (1::real) ∗ e3)) + (a2$^{3::nat}$ ∗ (real_of_nat (2::nat) ∗ ((− real_of_nat (256::nat) + (b2^2 + (− real_of_nat (2::nat) ∗ (b2 ∗ c2) + c2^2))) ∗ e1) + ((b2^2 ∗ (− real_of_nat (8::nat) + c2) + (− real_of_nat (16::nat) ∗ (b2 ∗ (real_of_nat (3::nat) + c2)) + real_of_nat (16::nat) ∗ (real_of_nat (16::nat) + real_of_nat (9::nat) ∗ c2))) ∗ e2 + (b2 ∗ (real_of_nat (144::nat) + (− real_of_nat (16::nat) ∗ c2 + c2^2)) + − real_of_nat (8::nat) ∗ (− real_of_nat (32::nat) + (real_of_nat (6::nat) ∗ c2 + c2^2))) ∗ e3)) + (real_of_nat (2::nat) ∗ (a2$^{4::nat}$ ∗ (real_of_nat (32::nat) ∗ e1 + real_of_nat (3::nat) ∗ ((− real_of_nat (8::nat) + b2) ∗ e2 + (− real_of_nat (8::nat) + c2) ∗ e3))) + (real_of_nat (200::nat) ∗ (a2^2 ∗ e1 + (real_of_nat (8::nat) ∗ ((b2 + − (1::real) ∗ c2) ∗ (e2 + − (1::real) ∗ e3)) + − (1::real) ∗ (a2 ∗ (real_of_nat (16::nat) ∗ e1 + ((− real_of_nat (8::nat) + b2) ∗ e2 + (− real_of_nat (8::nat) + c2) ∗ e3)))))^2 + (real_of_nat (2::nat) ∗ (a2^2 ∗ ((b2 + − (1::real) ∗ c2) ∗ (b2^2 ∗ e2 + (real_of_nat (8::nat) ∗ (c2 ∗ (real_of_nat (4::nat) ∗ e1 + (real_of_nat (9::nat) ∗ e2 + − real_of_nat (7::nat) ∗ e3))) + (real_of_nat (384::nat) ∗ (e2 + − (1::real) ∗ e3) + (− (1::real) ∗ (c2^2 ∗ e3) + b2 ∗ (− real_of_nat (32::nat) ∗ e1 + ((real_of_nat (56::nat) + − real_of_nat (9::nat) ∗ c2) ∗ e2 + real_of_nat (9::nat) ∗ ((− real_of_nat (8::nat) + c2) ∗ e3))))))))) + − real_of_nat (16::nat) ∗ (a2 ∗ ((b2 + − (1::real) ∗ c2) ∗ (b2^2 ∗ (e2 + − real_of_nat (3::nat) ∗ e3) + (− real_of_nat (4::nat) ∗ (b2 ∗ (real_of_nat (8::nat) ∗ e1 + ((− real_of_nat (20::nat) + real_of_nat (3::nat) ∗ c2) ∗ e2 + − real_of_nat (3::nat) ∗ ((− real_of_nat (4::nat) + c2) ∗ e3)))) + c2 ∗ (real_of_nat (32::nat) ∗ e1 + (real_of_nat (3::nat) ∗ ((real_of_nat (16::nat) + c2) ∗ e2) + − (1::real) ∗ ((real_of_nat (80::nat) + c2) ∗ e3)))))))))))))

thm DEF_flat_term_x:

flat_term_x = (λ_1974359::real. flat_term (sqrt _1974359))

thm Sphere.flat_term_x:

∀ x::real. flat_term_x x = flat_term (sqrt x)

thm DEF_taum_x:

taum_x = (λ(_1974364::real) (_1974365::real) (_1974366::real) (_1974367::real) (_1974368::real) _1974369::real. rhazim_x _1974364 _1974365 _1974366 _1974367 _1974368 _1974369 + (rhazim2_x _1974364 _1974365 _1974366 _1974367 _1974368 _1974369 + (rhazim3_x _1974364 _1974365 _1974366 _1974367 _1974368 _1974369 − ((1::real) + const1) ∗ pi)))

thm Sphere.taum_x:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. taum_x x1 x2 x3 x4 x5 x6 = rhazim_x x1 x2 x3 x4 x5 x6 + (rhazim2_x x1 x2 x3 x4 x5 x6 + (rhazim3_x x1 x2 x3 x4 x5 x6 − ((1::real) + const1) ∗ pi))

thm DEF_sqp:

sqp = (λ_1974424::real. if _1974424 < (1::real) then real_of_nat (3::nat) / real_of_nat (8::nat) + (((1::real) − _1974424)$^{3::nat}$ ∗ (− DECIMAL (25::nat)

(100::nat) + DECIMAL (7::nat) (10::nat) ∗ _1974424) + (real_of_nat (3::nat) ∗ (_1974424 / real_of_nat (4::nat)) − _1974424 ∗ (_1974424 / real_of_nat (8::nat)))) else sqrt _1974424)

thm Sphere.sqp:

∀ x::real. sqp x = (if x < (1::real) then real_of_nat (3::nat) / real_of_nat (8::nat) + (((1::real) − x)$^{3::nat}$ ∗ (− DECIMAL (25::nat) (100::nat) + DECIMAL (7::nat) (10::nat) ∗ x) + (real_of_nat (3::nat) ∗ (x / real_of_nat (4::nat)) − x ∗ (x / real_of_nat (8::nat)))) else sqrt x)

thm DEF_sqn:

sqn = (λ_1974429::real. if _1974429 < (1::real) then DECIMAL (375::nat) (1000::nat) + (real_of_nat (3::nat) ∗ _1974429 / real_of_nat (4::nat) − _1974429^2 / real_of_nat (8::nat) − DECIMAL (3::nat) (10::nat) ∗ (((1::real) − _1974429)$^{3::nat}$ ∗ _1974429^2) + ((1::real) − _1974429)$^{3::nat}$ ∗ (− DECIMAL (375::nat) (1000::nat) + DECIMAL (13::nat) (10::nat) ∗ (((1::real) − _1974429) ∗ _1974429))) else sqrt _1974429)

thm Sphere.sqn:

∀ x::real. sqn x = (if x < (1::real) then DECIMAL (375::nat) (1000::nat) + (real_of_nat (3::nat) ∗ x / real_of_nat (4::nat) − x^2 / real_of_nat (8::nat) − DECIMAL (3::nat) (10::nat) ∗ (((1::real) − x)$^{3::nat}$ ∗ x^2) + ((1::real) − x)$^{3::nat}$ ∗ (− DECIMAL (375::nat) (1000::nat) + DECIMAL (13::nat) (10::nat) ∗ (((1::real) − x) ∗ x))) else sqrt x)

thm DEF_upper_dih_x:

upper_dih_x = (λ(_1974434::real) (_1974435::real) (_1974436::real) (_1974437::real) (_1974438::real) _1974439::real. LET (λd::real. LET_END (LET (λd4::real. LET_END (real_of_nat (2::nat) ∗ (sqrt _1974434 ∗ (sqp d ∗ (matan (real_of_nat (4::nat) ∗ (_1974434 ∗ (d / d4^2))) / d4))))) (delta_x4 _1974434 _1974435 _1974436 _1974437 _1974438 _1974439))) (delta_x _1974434 _1974435 _1974436 _1974437 _1974438 _1974439))

thm Sphere.upper_dih_x:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. upper_dih_x x1 x2 x3 x4 x5 x6 = LET (λd::real. LET_END (LET (λd4::real. LET_END (real_of_nat (2::nat) ∗ (sqrt x1 ∗ (sqp d ∗ (matan (real_of_nat (4::nat) ∗ (x1 ∗ (d / d4^2))) / d4))))) (delta_x4 x1 x2 x3 x4 x5 x6))) (delta_x x1 x2 x3 x4 x5 x6)

thm Sphere.upper_dih_y:

upper_dih_y = y_of_x upper_dih_x

thm DEF_gamma3f_135_n:

gamma3f_135_n = (λ(_1974494::real) (_1974495::real) (_1974496::real) (_1974497::real) (_1974498::real) _1974499::real. sqn (delta_y _1974494 _1974495 _1974496

_1974497 _1974498 _1974499) ∗ ((1::real) / real_of_nat (12::nat) − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (y_of_x sol_euler_x_div_sqrtdelta _1974494 _1974495 _1974496 _1974497 _1974498 _1974499 + (y_of_x sol_euler156_x_div_sqrtdelta _1974494 _1974495 _1974496 _1974497 _1974498 _1974499 + y_of_x sol_euler345_x_div_sqrtdelta _1974494 _1974495 _1974496 _1974497 _1974498 _1974499)) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (y_of_x lmdih_x_div_sqrtdelta_posbranch _1974494 _1974495 _1974496 _1974497 _1974498 _1974499 + (y_of_x lmdih3_x_div_sqrtdelta_posbranch _1974494 _1974495 _1974496 _1974497 _1974498 _1974499 + y_of_x lmdih5_x_div_sqrtdelta_posbranch _1974494 _1974495 _1974496 _1974497 _1974498 _1974499)))))

thm Sphere.gamma3f_135_n:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. gamma3f_135_n y1 y2 y3 y4 y5 y6 = sqn (delta_y y1 y2 y3 y4 y5 y6) ∗ ((1::real) / real_of_nat (12::nat) − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (y_of_x sol_euler_x_div_sqrtdelta y1 y2 y3 y4 y5 y6 + (y_of_x sol_euler156_x_div_sqrtdelta y1 y2 y3 y4 y5 y6 + y_of_x sol_euler345_x_div_sqrtdelta y1 y2 y3 y4 y5 y6)) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (y_of_x lmdih_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6 + (y_of_x lmdih3_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6 + y_of_x lmdih5_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6))))

thm DEF_gamma3f_126_n:

gamma3f_126_n = (λ(_1974554::real) (_1974555::real) (_1974556::real) (_1974557::real) (_1974558::real) _1974559::real. sqn (delta_y _1974554 _1974555 _1974556 _1974557 _1974558 _1974559) ∗ ((1::real) / real_of_nat (12::nat) − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (y_of_x sol_euler_x_div_sqrtdelta _1974554 _1974555 _1974556 _1974557 _1974558 _1974559 + (y_of_x sol_euler246_x_div_sqrtdelta _1974554 _1974555 _1974556 _1974557 _1974558 _1974559 + y_of_x sol_euler156_x_div_sqrtdelta _1974554 _1974555 _1974556 _1974557 _1974558 _1974559)) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (y_of_x lmdih_x_div_sqrtdelta_posbranch _1974554 _1974555 _1974556 _1974557 _1974558 _1974559 + (y_of_x lmdih2_x_div_sqrtdelta_posbranch _1974554 _1974555 _1974556 _1974557 _1974558 _1974559 + y_of_x lmdih6_x_div_sqrtdelta_posbranch _1974554 _1974555 _1974556 _1974557 _1974558 _1974559)))))

thm Sphere.gamma3f_126_n:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. gamma3f_126_n y1 y2 y3 y4 y5 y6 = sqn (delta_y y1 y2 y3 y4 y5 y6) ∗ ((1::real) / real_of_nat (12::nat) − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (y_of_x sol_euler_x_div_sqrtdelta y1 y2 y3 y4 y5 y6 + (y_of_x sol_euler246_x_div_sqrtdelta y1 y2 y3 y4 y5 y6 + y_of_x sol_euler156_x_div_sqrtdelta y1 y2 y3 y4 y5 y6)) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (y_of_x lmdih_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6 + (y_of_x lmdih2_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6 + y_of_x lmdih6_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6))))

thm DEF_gamma23f_n:

gamma23f_n = (λ(_1974614::real) (_1974615::real) (_1974616::real) (_1974617::real) (_1974618::real) (_1974619::real) (_1974620::nat) (_1974621::nat) (_1974622::real)

_1974623::real ⇒ real. gamma3f_126_n _1974614 _1974615 sqrt2 sqrt2 sqrt2 _1974619 / real_of_nat _1974620 + (gamma3f_135_n _1974614 sqrt2 _1974616 sqrt2 _1974618 sqrt2 / real_of_nat _1974621 + (dih_y _1974614 _1974615 _1974616 _1974617 _1974618 _1974619 − upper_dih_y _1974614 _1974615 _1974622 _1974622 _1974622 _1974619 − upper_dih_y _1974614 _1974616 _1974622 _1974622 _1974622 _1974618) ∗ ((vol2r _1974614 _1974622 − vol2f _1974614 _1974622 _1974623) / (real_of_nat (2::nat) ∗ pi))))

thm Sphere.gamma23f_n:

∀ (w1::nat) (w2::nat) (y4::real) (y2::real) (y6::real) (y3::real) (y5::real) (y1::real) (r::real) f::real ⇒ real. gamma23f_n y1 y2 y3 y4 y5 y6 w1 w2 r f = gamma3f_126_n y1 y2 sqrt2 sqrt2 sqrt2 y6 / real_of_nat w1 + (gamma3f_135_n y1 sqrt2 y3 sqrt2 y5 sqrt2 / real_of_nat w2 + (dih_y y1 y2 y3 y4 y5 y6 − upper_dih_y y1 y2 r r r y6 − upper_dih_y y1 y3 r r r y5) ∗ ((vol2r y1 r − vol2f y1 r f) / (real_of_nat (2::nat) ∗ pi)))

thm DEF_gamma23f_126_03_n:

gamma23f_126_03_n = (λ(_1974754::real) (_1974755::real) (_1974756::real) (_1974757::real) (_1974758::real) (_1974759::real) (_1974760::nat) (_1974761::real) _1974762::real ⇒ real. gamma3f_126_n _1974754 _1974755 sqrt2 sqrt2 sqrt2 _1974759 / real_of_nat _1974760 + (dih_y _1974754 _1974755 _1974756 _1974757 _1974758 _1974759 − upper_dih_y _1974754 _1974755 _1974761 _1974761 _1974761 _1974759 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r _1974754 _1974761 − vol2f _1974754 _1974761 _1974762) / (real_of_nat (2::nat) ∗ pi)))

thm Sphere.gamma23f_126_03_n:

∀ (w1::nat) (y3::real) (y4::real) (y5::real) (y2::real) (y6::real) (y1::real) (r::real) f::real ⇒ real. gamma23f_126_03_n y1 y2 y3 y4 y5 y6 w1 r f = gamma3f_126_n y1 y2 sqrt2 sqrt2 sqrt2 y6 / real_of_nat w1 + (dih_y y1 y2 y3 y4 y5 y6 − upper_dih_y y1 y2 r r r y6 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r y1 r − vol2f y1 r f) / (real_of_nat (2::nat) ∗ pi))

thm DEF_eulerA_x:

eulerA_x = (λ(_1974871::real) (_1974872::real) (_1974873::real) (_1974874::real) (_1974875::real) _1974876::real. sqrt _1974871 ∗ (sqrt _1974872 ∗ sqrt _1974873) + (sqrt _1974871 ∗ ((_1974872 + (_1974873 − _1974874)) / real_of_nat (2::nat)) + (sqrt _1974872 ∗ ((_1974871 + (_1974873 − _1974875)) / real_of_nat (2::nat)) + sqrt _1974873 ∗ ((_1974871 + (_1974872 − _1974876)) / real_of_nat (2::nat)))))

thm Sphere.eulerA_x:

∀ (x4::real) (x5::real) (x3::real) (x1::real) (x2::real) x6::real. eulerA_x x1 x2 x3 x4 x5 x6 = sqrt x1 ∗ (sqrt x2 ∗ sqrt x3) + (sqrt x1 ∗ ((x2 + (x3 − x4)) / real_of_nat (2::nat)) + (sqrt x2 ∗ ((x1 + (x3 − x5)) / real_of_nat (2::nat)) + sqrt x3 ∗ ((x1 + (x2 − x6)) / real_of_nat (2::nat))))

thm DEF_euler_3flat_x:

euler_3flat_x = (λ(*_1974931*::*real*) (*_1974932*::*real*) (*_1974933*::*real*) (*_1974934*::*real*) (*_1974935*::*real*) *_1974936*::*real*. *LET* (λ*x5*::*real*. *LET_END* (*LET* (λ*x6*::*real*. *LET_END* (*LET* (λ*x4*::*real*. *LET_END* (*eulerA_x _1974931 _1974932 _1974933 x4 x5 x6*)) (*edge_flat2_x _1974934 _1974932 _1974933* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x _1974936 _1974931 _1974932* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x _1974935 _1974931 _1974933* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))

thm Sphere.euler_3flat_x:

∀ (*x23*::*real*) (*x12*::*real*) (*x2*::*real*) (*x13*::*real*) (*x1*::*real*) *x3*::*real*. *euler_3flat_x x1 x2 x3 x23 x13 x12* = *LET* (λ*x5*::*real*. *LET_END* (*LET* (λ*x6*::*real*. *LET_END* (*LET* (λ*x4*::*real*. *LET_END* (*eulerA_x x1 x2 x3 x4 x5 x6*)) (*edge_flat2_x x23 x2 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm DEF_euler_2flat_x:

euler_2flat_x = (λ(*_1974991*::*real*) (*_1974992*::*real*) (*_1974993*::*real*) (*_1974994*::*real*) (*_1974995*::*real*) *_1974996*::*real*. *LET* (λ*x5*::*real*. *LET_END* (*LET* (λ*x6*::*real*. *LET_END* (*eulerA_x _1974991 _1974992 _1974993 _1974994 x5 x6*)) (*edge_flat2_x _1974996 _1974991 _1974992* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x _1974995 _1974991 _1974993* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))

thm Sphere.euler_2flat_x:

∀ (*x4*::*real*) (*x12*::*real*) (*x2*::*real*) (*x13*::*real*) (*x1*::*real*) *x3*::*real*. *euler_2flat_x x1 x2 x3 x4 x13 x12* = *LET* (λ*x5*::*real*. *LET_END* (*LET* (λ*x6*::*real*. *LET_END* (*eulerA_x x1 x2 x3 x4 x5 x6*)) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))) (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm DEF_euler_1flat_x:

euler_1flat_x = (λ(*_1975051*::*real*) (*_1975052*::*real*) (*_1975053*::*real*) (*_1975054*::*real*) (*_1975055*::*real*) *_1975056*::*real*. *LET* (λ*x6*::*real*. *LET_END* (*eulerA_x _1975051 _1975052 _1975053 _1975054 _1975055 x6*)) (*edge_flat2_x _1975056 _1975051 _1975052* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))))

thm Sphere.euler_1flat_x:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x12*::*real*) (*x1*::*real*) *x2*::*real*. *euler_1flat_x x1 x2 x3 x4 x5 x12* = *LET* (λ*x6*::*real*. *LET_END* (*eulerA_x x1 x2 x3 x4 x5 x6*)) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm DEF_taum_3flat_x:

taum_3flat_x = (λ(*_1975111*::*real*) (*_1975112*::*real*) (*_1975113*::*real*) (*_1975114*::*real*) (*_1975115*::*real*) *_1975116*::*real*. *LET* (λ*x5*::*real*. *LET_END* (*LET* (λ*x6*::*real*. *LET_END* (*LET* (λ*x4*::*real*. *LET_END* (*taum_x _1975111 _1975112 _1975113* *x4 x5 x6 + (flat_term_x _1975116 + (flat_term_x _1975114 + flat_term_x _1975115)))) (edge_flat2_x _1975114 _1975112 _1975113 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x _1975116 _1975111 _1975112 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x _1975115 _1975111 _1975113 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))*

thm Sphere.taum_3flat_x:

∀(x23::real) (x12::real) (x2::real) (x13::real) (x1::real) x3::real. taum_3flat_x x1 x2 x3 x23 x13 x12 = LET (λx5::real. LET_END (LET (λx6::real. LET_END (LET (λx4::real. LET_END (taum_x x1 x2 x3 x4 x5 x6 + (flat_term_x x12 + (flat_term_x x23 + flat_term_x x13)))) (edge_flat2_x x23 x2 x3 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x x12 x1 x2 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x x13 x1 x3 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat)))

thm DEF_taum_2flat_x:

taum_2flat_x = (λ(_1975171::real) (_1975172::real) (_1975173::real) (_1975174::real) (_1975175::real) _1975176::real. LET (λx5::real. LET_END (LET (λx6::real. LET_END (taum_x _1975171 _1975172 _1975173 _1975174 x5 x6 + (flat_term_x _1975176 + flat_term_x _1975175))) (edge_flat2_x _1975176 _1975171 _1975172 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x _1975175 _1975171 _1975173 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))

thm Sphere.taum_2flat_x:

∀(x4::real) (x12::real) (x2::real) (x13::real) (x1::real) x3::real. taum_2flat_x x1 x2 x3 x4 x13 x12 = LET (λx5::real. LET_END (LET (λx6::real. LET_END (taum_x x1 x2 x3 x4 x5 x6 + (flat_term_x x12 + flat_term_x x13))) (edge_flat2_x x12 x1 x2 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))) (edge_flat2_x x13 x1 x3 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat)))

thm DEF_taum_1flat_x:

taum_1flat_x = (λ(_1975231::real) (_1975232::real) (_1975233::real) (_1975234::real) (_1975235::real) _1975236::real. LET (λx6::real. LET_END (taum_x _1975231 _1975232 _1975233 _1975234 _1975235 x6 + flat_term_x _1975236)) (edge_flat2_x _1975236 _1975231 _1975232 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat))))

thm Sphere.taum_1flat_x:

∀(x3::real) (x4::real) (x5::real) (x12::real) (x1::real) x2::real. taum_1flat_x x1 x2 x3 x4 x5 x12 = LET (λx6::real. LET_END (taum_x x1 x2 x3 x4 x5 x6 + flat_term_x x12)) (edge_flat2_x x12 x1 x2 (0::real) (real_of_nat (4::nat)) (real_of_nat (4::nat)))

thm DEF_delta4_squared_x:

delta4_squared_x = (λ(_1975291::real) (_1975292::real) (_1975293::real) (_1975294::real) (_1975295::real) _1975296::real. (delta_x4 _1975291 _1975292 _1975293 _1975294 _1975295 _1975296)^2)

thm Sphere.delta4_squared_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ delta4\_squared\_x\ x1\ x2\ x3\ x4\ x5\ x6 = (delta\_x4\ x1\ x2\ x3\ x4\ x5\ x6)^2$

thm Sphere.delta4_squared_y:

$delta4\_squared\_y = y\_of\_x\ delta4\_squared\_x$

thm DEF_x1_delta_x:

$x1\_delta\_x = (\lambda(\_1975351::real)\ (\_1975352::real)\ (\_1975353::real)\ (\_1975354::real)\ (\_1975355::real)\ \_1975356::real.\ \_1975351 * delta\_x\ \_1975351\ \_1975352\ \_1975353\ \_1975354\ \_1975355\ \_1975356)$

thm Sphere.x1_delta_x:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ x1\_delta\_x\ x1\ x2\ x3\ x4\ x5\ x6 = x1 * delta\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm Sphere.x1_delta_y:

$x1\_delta\_y = y\_of\_x\ x1\_delta\_x$

thm DEF_delta_126_x:

$delta\_126\_x = (\lambda(\_1975411::real)\ (\_1975412::real)\ (\_1975413::real)\ (\_1975414::real)\ (\_1975415::real)\ (\_1975416::real)\ (\_1975417::real)\ \_1975418::real.\ delta\_x\ \_1975414\ \_1975415\ \_1975411\ \_1975412\ \_1975413)$

thm Sphere.delta_126_x:

$\forall (x3::real)\ (x4::real)\ (x5::real)\ (x1::real)\ (x2::real)\ (x3s::real)\ (x4s::real)\ (x5s::real)\ x6::real.\ delta\_126\_x\ x3s\ x4s\ x5s\ x1\ x2\ x3\ x4\ x5\ x6 = delta\_x\ x1\ x2\ x3s\ x4s\ x5s\ x6$

thm DEF_delta_234_x:

$delta\_234\_x = (\lambda(\_1975528::real)\ (\_1975529::real)\ (\_1975530::real)\ (\_1975531::real)\ (\_1975532::real)\ (\_1975533::real)\ (\_1975534::real)\ (\_1975535::real)\ \_1975536::real.\ delta\_x\ \_1975528\ \_1975532\ \_1975533\ \_1975534\ \_1975529\ \_1975530)$

thm Sphere.delta_234_x:

$\forall (x1::real)\ (x5::real)\ (x6::real)\ (x1s::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5s::real)\ x6s::real.\ delta\_234\_x\ x1s\ x5s\ x6s\ x1\ x2\ x3\ x4\ x5\ x6 = delta\_x\ x1s\ x2\ x3\ x4\ x5s\ x6s$

thm DEF_delta_135_x:

$delta\_135\_x = (\lambda(\_1975645::real)\ (\_1975646::real)\ (\_1975647::real)\ (\_1975648::real)\ (\_1975649::real)\ (\_1975650::real)\ (\_1975651::real)\ (\_1975652::real)\ \_1975653::real.\ delta\_x\ \_1975648\ \_1975645\ \_1975650\ \_1975646\ \_1975652\ \_1975647)$

thm Sphere.delta_135_x:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x2s*::*real*) (*x3*::*real*) (*x4s*::*real*) (*x5*::*real*) *x6s*::*real*. *delta_135_x x2s x4s x6s x1 x2 x3 x4 x5 x6* = *delta_x x1 x2s x3 x4s x5 x6s*

thm DEF_delta_pent_x:

delta_pent_x = (λ(*_1975762*::*real*) (*_1975763*::*real*) (*_1975764*::*real*) (*_1975765*::*real*) (*_1975766*::*real*) *_1975767*::*real*. *delta_x _1975762 _1975763 _1975767* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) ((*DECIMAL* (*324*::*nat*) (*100*::*nat*))^2))

thm Sphere.delta_pent_x:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *delta_pent_x x1 x2 x3 x4 x5 x6* = *delta_x x1 x2 x6* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) ((*DECIMAL* (*324*::*nat*) (*100*::*nat*))^2)

thm DEF_delta_sub1_x:

delta_sub1_x = (λ(*_1975822*::*real*) *_1975823*::*real*. *delta_x _1975822*)

thm Sphere.delta_sub1_x:

∀ (*x1*::*real*) (*x1s*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *delta_sub1_x x1s x1 x2 x3 x4 x5 x6* = *delta_x x1s x2 x3 x4 x5 x6*

thm DEF_taum_sub1_x:

taum_sub1_x = (λ(*_1975899*::*real*) *_1975900*::*real*. *taum_x _1975899*)

thm Sphere.taum_sub1_x:

∀ (*x1*::*real*) (*x1s*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *taum_sub1_x x1s x1 x2 x3 x4 x5 x6* = *taum_x x1s x2 x3 x4 x5 x6*

thm DEF_taum_sub246_x:

taum_sub246_x = (λ(*_1975976*::*real*) (*_1975977*::*real*) (*_1975978*::*real*) (*_1975979*::*real*) (*_1975980*::*real*) (*_1975981*::*real*) (*_1975982*::*real*) (*_1975983*::*real*) *_1975984*::*real*. *taum_x _1975979 _1975976 _1975981 _1975977 _1975983 _1975978*)

thm Sphere.taum_sub246_x:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x2s*::*real*) (*x3*::*real*) (*x4s*::*real*) (*x5*::*real*) *x6s*::*real*. *taum_sub246_x x2s x4s x6s x1 x2 x3 x4 x5 x6* = *taum_x x1 x2s x3 x4s x5 x6s*

thm DEF_taum_sub345_x:

taum_sub345_x = (λ(*_1976093*::*real*) (*_1976094*::*real*) (*_1976095*::*real*) (*_1976096*::*real*) (*_1976097*::*real*) (*_1976098*::*real*) (*_1976099*::*real*) *_1976100*::*real*. *taum_x _1976096 _1976097 _1976093 _1976094 _1976095*)

thm Sphere.taum_sub345_x:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3s*::*real*) (*x4s*::*real*) (*x5s*::*real*) *x6*::*real*. *taum_sub345_x x3s x4s x5s x1 x2 x3 x4 x5 x6* = *taum_x x1 x2 x3s x4s x5s x6*

thm DEF_rhazim_x_div_sqrtdelta_posbranch:

rhazim_x_div_sqrtdelta_posbranch = (λ(*_1976210*::*real*) (*_1976211*::*real*) (*_1976212*::*real*) (*_1976213*::*real*) (*_1976214*::*real*) *_1976215*::*real*. *rho* (*sqrt _1976210*) ∗ *dih_x_div_sqrtdelta_posbranch _1976210 _1976211 _1976212 _1976213 _1976214 _1976215*)

thm Sphere.rhazim_x_div_sqrtdelta_posbranch:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *rhazim_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* = *rho* (*sqrt x1*) ∗ *dih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm Functional_equation.functional_rhazim2_x_div_sqrt_delta_posbranch:

rhazim2_x_div_sqrtdelta_posbranch = *rotate2 rhazim_x_div_sqrtdelta_posbranch*

thm Sphere.rhazim3_x_div_sqrtdelta_posbranch:

rhazim3_x_div_sqrtdelta_posbranch = *rotate3 rhazim_x_div_sqrtdelta_posbranch*

thm DEF_tau_residual_x:

tau_residual_x = (λ(*_1976270*::*real*) (*_1976271*::*real*) (*_1976272*::*real*) (*_1976273*::*real*) (*_1976274*::*real*) *_1976275*::*real*. *rhazim_x_div_sqrtdelta_posbranch _1976270 _1976271 _1976272 _1976273 _1976274 _1976275* + (*rhazim2_x_div_sqrtdelta_posbranch _1976270 _1976271 _1976272 _1976273 _1976274 _1976275* + *rhazim3_x_div_sqrtdelta_posbranch _1976270 _1976271 _1976272 _1976273 _1976274 _1976275*))

thm Sphere.tau_residual_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *tau_residual_x x1 x2 x3 x4 x5 x6* = *rhazim_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* + (*rhazim2_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6* + *rhazim3_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*)

thm DEF_delta_y_LC:

delta_y_LC = *delta_y*

thm Sphere.delta_y_LC:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *delta_y_LC y1 y2 y3 y4 y5 y6* = *delta_y y1 y2 y3 y4 y5 y6*

thm DEF_ell_uvx:

ell_uvx = (λ(*_1976390*::*real*) (*_1976391*::*real*) (*_1976392*::*real*) (*_1976393*::*real*) (*_1976394*::*real*) *_1976395*::*real*. *LET* (λ*et2*::*real*. *LET_END* (*sqrt* (*et2* − *_1976390* / *real_of_nat* (*4*::*nat*)) + *sqrt* (*et2* − *_1976391* / *real_of_nat* (*4*::*nat*)))) ((*eta_x _1976390 _1976391 _1976392*)^2))

thm Sphere.ell_uvx:

∀ (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *ell_uvx x1 x2 x3 x4 x5 x6* = *LET* (λ*et2*::*real*. *LET_END* (*sqrt* (*et2* − *x1* / *real_of_nat* (*4*::*nat*)) + *sqrt* (*et2* − *x2* / *real_of_nat* (*4*::*nat*)))) ((*eta_x x1 x2 x3*)^2)

thm DEF_ell_vx2:

ell_vx2 = (λ(*_1976450*::*real*) (*_1976451*::*real*) (*_1976452*::*real*) (*_1976453*::*real*) (*_1976454*::*real*) *_1976455*::*real*. *LET* (λ*et2*::*real*. *LET_END* (*sqrt* (*et2* − *_1976451* / *real_of_nat* (*4*::*nat*)))) ((*eta_x* *_1976450* *_1976451* *_1976452*)^2))

thm Sphere.ell_vx2:

∀ (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *ell_vx2 x1 x2 x3 x4 x5 x6* = *LET* (λ*et2*::*real*. *LET_END* (*sqrt* (*et2* − *x2* / *real_of_nat* (*4*::*nat*)))) ((*eta_x x1 x2 x3*)^2)

thm DEF_voronoi_trg:

voronoi_trg = (λ(*_1976510*::(*real*, *?'a*::*type*) *cart*) *_1976511*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%10*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%10* (∀ *w*::(*real*, *?'a*::*type*) *cart*. *_1976511* *w* ∧ *w* ≠ *_1976510* ⟶ *distance* (*x*, *_1976510*) < *distance* (*x*, *w*)) *x*))

thm Geomdetail.voronoi_trg:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *voronoi_trg v S* = *GSPEC* (λ*GEN%PVAR%10*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%10* (∀ *w*::(*real*, *?'a*::*type*) *cart*. *S w* ∧ *w* ≠ *v* ⟶ *distance* (*x*, *v*) < *distance* (*x*, *w*)) *x*)

thm DEF_conv0_2:

conv0_2 = *conv0*

thm Geomdetail.conv0_2:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *conv0_2 s* = *conv0 s*

thm Geomdetail.convex:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *convex s* = (∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*u*::*real*) *v*::*real*. *IN x s* ∧ *IN y s* ∧ (*0*::*real*) ≤ *u* ∧ (*0*::*real*) ≤ *v* ∧ *u* + *v* = (*1*::*real*) ⟶ *IN* (*vector_add* (% *u x*) (% *v y*)) *s*)

thm Geomdetail.aff:

aff = *hull affine*

thm DEF_conv_trg:

conv_trg = *hull convex*

thm Geomdetail.conv_trg:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *conv_trg s* = *hull convex s*

thm DEF_border:

border = (λ*_1976532*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%11*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%11* (∀ *ep*::*real*. (*0*::*real*) < *ep* ∧ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ *IN b*

$\_1976532 \wedge distance\ (b,\ x) < ep \wedge IN\ a\ \_1976532 \wedge distance\ (a,\ x) < ep))\ x))$

thm Geomdetail.border:

$\forall\, s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ border\ s = GSPEC\ (\lambda GEN\%PVAR\%11::(real,\ ?'a::type)\ cart.\ \exists\, x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%11\ (\forall\, ep::real.\ (0::real) < ep \wedge (\exists\, (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ \neg\ IN\ b\ s \wedge distance\ (b,\ x) < ep \wedge IN\ a\ s \wedge distance\ (a,\ x) < ep))\ x)$

thm DEF_packing_trg:

$packing\_trg = (\lambda\_1976537::(real,\ 3)\ cart \Rightarrow bool.\ \forall\, (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ \_1976537\ x \wedge \_1976537\ y \wedge x \neq y \longrightarrow real\_of\_nat\ (2::nat) \leq distance\ (x,\ y))$

thm Geomdetail.packing_trg:

$\forall\, s::(real,\ 3)\ cart \Rightarrow bool.\ packing\_trg\ s = (\forall\, (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ s\ x \wedge s\ y \wedge x \neq y \longrightarrow real\_of\_nat\ (2::nat) \leq distance\ (x,\ y))$

thm DEF_center_pac:

$center\_pac = (\lambda(\_1976542::(real,\ 3)\ cart \Rightarrow bool)\ \_1976543::(real,\ 3)\ cart.\ packing\_trg\ \_1976542 \wedge \_1976542\ \_1976543)$

thm Geomdetail.center_pac:

$\forall\, (s::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ center\_pac\ s\ v = (packing\_trg\ s \wedge s\ v)$

thm DEF_centered_pac:

$centered\_pac = (\lambda(\_1976554::(real,\ 3)\ cart \Rightarrow bool)\ \_1976555::(real,\ 3)\ cart.\ packing\ \_1976554 \wedge IN\ \_1976555\ \_1976554)$

thm Geomdetail.centered_pac:

$\forall\, (v::(real,\ 3)\ cart)\ s::(real,\ 3)\ cart \Rightarrow bool.\ centered\_pac\ s\ v = (packing\ s \wedge IN\ v\ s)$

thm DEF_d3:

$d3 = (\lambda(\_1976566::(real,\ 3)\ cart)\ \_1976567::(real,\ 3)\ cart.\ distance\ (\_1976566,\ \_1976567))$

thm Geomdetail.d3:

$\forall\, (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ d3\ x\ y = distance\ (x,\ y)$

thm DEF_voronoi2:

$voronoi2 = (\lambda(\_1976578::(real,\ 3)\ cart)\ \_1976579::(real,\ 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%12::(real,\ 3)\ cart.\ \exists\, x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%12\ (IN\ x\ (voronoi\_trg\ \_1976578\ \_1976579) \wedge d3\ x\ \_1976578 < real\_of\_nat\ (2::nat))\ x))$

thm Geomdetail.voronoi2:

$\forall (S::(real, 3)\ cart \Rightarrow bool)\ v::(real, 3)\ cart.\ voronoi2\ v\ S = GSPEC\ (\lambda GEN\%PVAR\%12::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%12\ (IN\ x\ (voronoi\_trg\ v\ S) \wedge d3\ x\ v < real\_of\_nat\ (2::nat))\ x)$

thm DEF_voro2:

$voro2 = (\lambda(\_1976590::(real, 3)\ cart)\ \_1976591::(real, 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%13::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%13\ (IN\ x\ (voronoi\_open\ \_1976591\ \_1976590) \wedge d3\ x\ \_1976590 < real\_of\_nat\ (2::nat))\ x))$

thm Geomdetail.voro2:

$\forall (s::(real, 3)\ cart \Rightarrow bool)\ v::(real, 3)\ cart.\ voro2\ v\ s = GSPEC\ (\lambda GEN\%PVAR\%13::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%13\ (IN\ x\ (voronoi\_open\ s\ v) \wedge d3\ x\ v < real\_of\_nat\ (2::nat))\ x)$

thm Geomdetail.t0:

$t0 = DECIMAL\ (1255::nat)\ (1000::nat)$

thm DEF_quasi_tri:

$quasi\_tri = (\lambda(\_1976602::(real, 3)\ cart \Rightarrow bool)\ \_1976603::(real, 3)\ cart \Rightarrow bool.\ packing\ \_1976603 \wedge SUBSET\ \_1976602\ \_1976603 \wedge (\exists (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ c::(real, 3)\ cart.\ \neg\ (a = b \vee b = c \vee c = a) \wedge INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)) = \_1976602) \wedge (\forall (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ IN\ x\ \_1976602 \wedge IN\ y\ \_1976602 \wedge x \neq y \longrightarrow d3\ x\ y \leq real\_of\_nat\ (2::nat) * t0))$

thm Geomdetail.quasi_tri:

$\forall (s::(real, 3)\ cart \Rightarrow bool)\ tri::(real, 3)\ cart \Rightarrow bool.\ quasi\_tri\ tri\ s = (packing\ s \wedge SUBSET\ tri\ s \wedge (\exists (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ c::(real, 3)\ cart.\ \neg\ (a = b \vee b = c \vee c = a) \wedge INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)) = tri) \wedge (\forall (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ IN\ x\ tri \wedge IN\ y\ tri \wedge x \neq y \longrightarrow d3\ x\ y \leq real\_of\_nat\ (2::nat) * t0))$

thm DEF_geomdetail′simplex:

$geomdetail'simplex = (\lambda(\_1976614::(real, 3)\ cart \Rightarrow bool)\ \_1976615::(real, 3)\ cart \Rightarrow bool.\ packing\ \_1976615 \wedge SUBSET\ \_1976614\ \_1976615 \wedge (\exists (v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ (v3::(real, 3)\ cart)\ v4::(real, 3)\ cart.\ \neg\ (v1 = v2 \vee v3 = v4) \wedge HOL\_Light\_Import.INTER\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))\ (INSERT\ v3\ (INSERT\ v4\ EMPTY)) = EMPTY \wedge INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY))) = \_1976614))$

thm Geomdetail.simplex:

$\forall (s::(real, 3)\ cart \Rightarrow bool)\ d::(real, 3)\ cart \Rightarrow bool.\ geomdetail'simplex\ d\ s = (packing\ s \wedge SUBSET\ d\ s \wedge (\exists (v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ (v3::(real,$

3) cart) v4::(real, 3) cart. ¬ (v1 = v2 ∨ v3 = v4) ∧ HOL_Light_Import.INTER (INSERT v1 (INSERT v2 EMPTY)) (INSERT v3 (INSERT v4 EMPTY)) = EMPTY ∧ INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))) = d))

thm DEF_quasi_reg_tet:

quasi_reg_tet = (λ(_1976626::(real, 3) cart ⇒ bool) _1976627::(real, 3) cart ⇒ bool. geomdetail′simplex _1976626 _1976627 ∧ (∀ (v::(real, 3) cart) w::(real, 3) cart. IN v _1976626 ∧ IN w _1976626 ∧ v ≠ w ⟶ d3 v w ≤ real_of_nat (2::nat) ∗ t0))

thm Geomdetail.quasi_reg_tet:

∀ (s::(real, 3) cart ⇒ bool) x::(real, 3) cart ⇒ bool. quasi_reg_tet x s = (geomdetail′simplex x s ∧ (∀ (v::(real, 3) cart) w::(real, 3) cart. IN v x ∧ IN w x ∧ v ≠ w ⟶ d3 v w ≤ real_of_nat (2::nat) ∗ t0))

thm DEF_quarter:

quarter = (λ(_1976638::(real, 3) cart ⇒ bool) _1976639::(real, 3) cart ⇒ bool. packing _1976639 ∧ geomdetail′simplex _1976638 _1976639 ∧ (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v _1976638 ∧ IN w _1976638 ∧ real_of_nat (2::nat) ∗ t0 ≤ d3 v w ∧ d3 v w ≤ sqrt (real_of_nat (8::nat)) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x _1976638 ∧ IN y _1976638 ∧ INSERT x (INSERT y EMPTY) ≠ INSERT v (INSERT w EMPTY) ⟶ d3 x y ≤ real_of_nat (2::nat) ∗ t0)))

thm Geomdetail.quarter:

∀ (s::(real, 3) cart ⇒ bool) q::(real, 3) cart ⇒ bool. quarter q s = (packing s ∧ geomdetail′simplex q s ∧ (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v q ∧ IN w q ∧ real_of_nat (2::nat) ∗ t0 ≤ d3 v w ∧ d3 v w ≤ sqrt (real_of_nat (8::nat)) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x q ∧ IN y q ∧ INSERT x (INSERT y EMPTY) ≠ INSERT v (INSERT w EMPTY) ⟶ d3 x y ≤ real_of_nat (2::nat) ∗ t0)))

thm DEF_diagonal:

diagonal = (λ(_1976650::(real, 3) cart) (_1976651::(real, 3) cart) (_1976652::(real, 3) cart ⇒ bool) _1976653::(real, 3) cart ⇒ bool. quarter _1976652 _1976653 ∧ SUBSET (INSERT _1976650 (INSERT _1976651 EMPTY)) _1976652 ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x _1976652 ∧ IN y _1976652 ⟶ d3 x y ≤ d3 _1976650 _1976651))

thm Geomdetail.diagonal:

∀ (s::(real, 3) cart ⇒ bool) (q::(real, 3) cart ⇒ bool) (d1::(real, 3) cart) d2::(real, 3) cart. diagonal d1 d2 q s = (quarter q s ∧ SUBSET (INSERT d1 (INSERT d2 EMPTY)) q ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x q ∧ IN y q ⟶ d3 x y ≤ d3 d1 d2))

thm DEF_strict_qua:

strict_qua = (λ(_1976682::(real, 3) cart ⇒ bool) _1976683::(real, 3) cart ⇒ bool. quarter _1976682 _1976683 ∧ (∃(x::(real, 3) cart) y::(real, 3) cart. IN x _1976682 ∧ IN y _1976682 ∧ real_of_nat (2::nat) ∗ t0 < d3 x y ∧ d3 x y < sqrt (real_of_nat (8::nat))))

thm Geomdetail.strict_qua:

∀(s::(real, 3) cart ⇒ bool) d::(real, 3) cart ⇒ bool. strict_qua d s = (quarter d s ∧ (∃(x::(real, 3) cart) y::(real, 3) cart. IN x d ∧ IN y d ∧ real_of_nat (2::nat) ∗ t0 < d3 x y ∧ d3 x y < sqrt (real_of_nat (8::nat))))

thm DEF_strict_qua2:

strict_qua2 = (λ(_1976694::(real, 3) cart ⇒ bool) (_1976695::(real, 3) cart ⇒ bool) _1976696::(real, 3) cart ⇒ bool. quarter _1976694 _1976696 ∧ SUBSET _1976695 _1976694 ∧ (∃(x::(real, 3) cart) y::(real, 3) cart. x ≠ y ∧ _1976695 = INSERT x (INSERT y EMPTY) ∧ real_of_nat (2::nat) ∗ t0 < d3 x y ∧ d3 x y < sqrt (real_of_nat (8::nat))))

thm Geomdetail.strict_qua2:

∀(s::(real, 3) cart ⇒ bool) (d::(real, 3) cart ⇒ bool) ch::(real, 3) cart ⇒ bool. strict_qua2 d ch s = (quarter d s ∧ SUBSET ch d ∧ (∃(x::(real, 3) cart) y::(real, 3) cart. x ≠ y ∧ ch = INSERT x (INSERT y EMPTY) ∧ real_of_nat (2::nat) ∗ t0 < d3 x y ∧ d3 x y < sqrt (real_of_nat (8::nat))))

thm DEF_quartered_oct:

quartered_oct = (λ(_1976715::(real, 3) cart) (_1976716::(real, 3) cart) (_1976717::(real, 3) cart) (_1976718::(real, 3) cart) (_1976719::(real, 3) cart) (_1976720::(real, 3) cart) _1976721::(real, 3) cart ⇒ bool. packing _1976721 ∧ (real_of_nat (2::nat) ∗ t0 < distance (_1976715, _1976716) ∧ distance (_1976715, _1976716) < sqrt (real_of_nat (8::nat))) ∧ (∀x::(real, 3) cart. IN x (INSERT _1976717 (INSERT _1976718 (INSERT _1976719 (INSERT _1976720 EMPTY)))) ⟶ distance (x, _1976715) ≤ real_of_nat (2::nat) ∗ t0 ∧ distance (x, _1976716) ≤ real_of_nat (2::nat) ∗ t0) ∧ SUBSET (INSERT _1976715 (INSERT _1976716 (INSERT _1976717 (INSERT _1976718 (INSERT _1976719 (INSERT _1976720 EMPTY)))))) _1976721 ∧ (real_of_nat (2::nat) ≤ distance (_1976717, _1976718) ∧ distance (_1976717, _1976718) ≤ real_of_nat (2::nat) ∗ t0) ∧ (real_of_nat (2::nat) ≤ distance (_1976718, _1976719) ∧ distance (_1976718, _1976719) ≤ real_of_nat (2::nat) ∗ t0) ∧ (real_of_nat (2::nat) ≤ distance (_1976719, _1976720) ∧ distance (_1976719, _1976720) ≤ real_of_nat (2::nat) ∗ t0) ∧ real_of_nat (2::nat) ≤ distance (_1976720, _1976717) ∧ distance (_1976720, _1976717) ≤ real_of_nat (2::nat) ∗ t0)

thm Geomdetail.quartered_oct:

∀(v::(real, 3) cart) (w::(real, 3) cart) (s::(real, 3) cart ⇒ bool) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v1::(real, 3) cart. quartered_oct v w v1 v2 v3 v4 s = (packing s ∧ (real_of_nat (2::nat) ∗ t0 < distance (v, w) ∧ distance (v, w) < sqrt (real_of_nat (8::nat))) ∧ (∀x::(real, 3) cart. IN x

(INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ distance (x, v) ≤ real_of_nat (2::nat) ∗ t0 ∧ distance (x, w) ≤ real_of_nat (2::nat) ∗ t0) ∧ SUBSET (INSERT v (INSERT w (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))))) s ∧ (real_of_nat (2::nat) ≤ distance (v1, v2) ∧ distance (v1, v2) ≤ real_of_nat (2::nat) ∗ t0) ∧ (real_of_nat (2::nat) ≤ distance (v2, v3) ∧ distance (v2, v3) ≤ real_of_nat (2::nat) ∗ t0) ∧ (real_of_nat (2::nat) ≤ distance (v3, v4) ∧ distance (v3, v4) ≤ real_of_nat (2::nat) ∗ t0) ∧ real_of_nat (2::nat) ≤ distance (v4, v1) ∧ distance (v4, v1) ≤ real_of_nat (2::nat) ∗ t0)

thm DEF_adjacent_pai:

adjacent_pai = (λ(_1976792::(real, 3) cart) (_1976793::(real, 3) cart) (_1976794::(real, 3) cart) (_1976795::(real, 3) cart) (_1976796::(real, 3) cart) _1976797::(real, 3) cart ⇒ bool. strict_qua2 (INSERT _1976792 (INSERT _1976793 (INSERT _1976794 (INSERT _1976795 EMPTY)))) (INSERT _1976793 (INSERT _1976794 EMPTY)) _1976797 ∧ strict_qua2 (INSERT _1976792 (INSERT _1976793 (INSERT _1976794 (INSERT _1976796 EMPTY)))) (INSERT _1976793 (INSERT _1976794 EMPTY)) _1976797 ∧ HOL_Light_Import.INTER (conv0 (INSERT _1976792 (INSERT _1976793 (INSERT _1976794 (INSERT _1976795 EMPTY))))) (conv0 (INSERT _1976792 (INSERT _1976793 (INSERT _1976794 (INSERT _1976796 EMPTY))))) = EMPTY)

thm Geomdetail.adjacent_pai:

∀ (s::(real, 3) cart ⇒ bool) (v2::(real, 3) cart) (v::(real, 3) cart) (v1::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. adjacent_pai v v1 v3 v2 v4 s = (strict_qua2 (INSERT v (INSERT v1 (INSERT v3 (INSERT v2 EMPTY)))) (INSERT v1 (INSERT v3 EMPTY)) s ∧ strict_qua2 (INSERT v (INSERT v1 (INSERT v3 (INSERT v4 EMPTY)))) (INSERT v1 (INSERT v3 EMPTY)) s ∧ HOL_Light_Import.INTER (conv0 (INSERT v (INSERT v1 (INSERT v3 (INSERT v2 EMPTY))))) (conv0 (INSERT v (INSERT v1 (INSERT v3 (INSERT v4 EMPTY))))) = EMPTY)

thm DEF_conflicting_dia:

conflicting_dia = (λ(_1976852::(real, 3) cart) (_1976853::(real, 3) cart) (_1976854::(real, 3) cart) (_1976855::(real, 3) cart) (_1976856::(real, 3) cart) _1976857::(real, 3) cart ⇒ bool. adjacent_pai _1976852 _1976853 _1976854 _1976855 _1976856 _1976857 ∧ adjacent_pai _1976852 _1976855 _1976856 _1976853 _1976854 _1976857)

thm Geomdetail.conflicting_dia:

∀ (v::(real, 3) cart) (v2::(real, 3) cart) (v4::(real, 3) cart) (v1::(real, 3) cart) (v3::(real, 3) cart) s::(real, 3) cart ⇒ bool. conflicting_dia v v1 v3 v2 v4 s = (adjacent_pai v v1 v3 v2 v4 s ∧ adjacent_pai v v2 v4 v1 v3 s)

thm DEF_interior_pos:

interior_pos = (λ(_1976912::(real, 3) cart) (_1976913::(real, 3) cart) (_1976914::(real, 3) cart) (_1976915::(real, 3) cart) (_1976916::(real, 3) cart) _1976917::(real,

$3)$ *cart* ⇒ *bool*. *conflicting_dia _1976912 _1976913 _1976914 _1976915 _1976916 _1976917* ∧ *HOL_Light_Import.INTER* (*conv0* (*INSERT _1976913* (*INSERT _1976914 EMPTY*))) (*conv0* (*INSERT _1976912* (*INSERT _1976915* (*INSERT _1976916 EMPTY*)))) ≠ *EMPTY*)

thm Geomdetail.interior_pos:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*v1*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *interior_pos v v1 v3 v2 v4 s* = (*conflicting_dia v v1 v3 v2 v4 s* ∧ *HOL_Light_Import.INTER* (*conv0* (*INSERT v1* (*INSERT v3 EMPTY*))) (*conv0* (*INSERT v* (*INSERT v2* (*INSERT v4 EMPTY*)))) ≠ *EMPTY*)

thm DEF_isolated_qua:

isolated_qua = (λ(*_1976972*::(*real*, *3*) *cart* ⇒ *bool*) *_1976973*::(*real*, *3*) *cart* ⇒ *bool*. *quarter _1976972 _1976973* ∧ ¬ (∃ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *_1976972* = *INSERT v* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *adjacent_pai v v1 v2 v3 v4 _1976973*))

thm Geomdetail.isolated_qua:

∀ (*q*::(*real*, *3*) *cart* ⇒ *bool*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *isolated_qua q s* = (*quarter q s* ∧ ¬ (∃ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *q* = *INSERT v* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *adjacent_pai v v1 v2 v3 v4 s*))

thm DEF_isolated_pai:

isolated_pai = (λ(*_1976984*::(*real*, *3*) *cart* ⇒ *bool*) (*_1976985*::(*real*, *3*) *cart* ⇒ *bool*) *_1976986*::(*real*, *3*) *cart* ⇒ *bool*. *isolated_qua _1976984 _1976986* ∧ *isolated_qua _1976985 _1976986* ∧ *HOL_Light_Import.INTER* (*conv0 _1976984*) (*conv0 _1976985*) ≠ *EMPTY*)

thm Geomdetail.isolated_pai:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*q1*::(*real*, *3*) *cart* ⇒ *bool*) *q2*::(*real*, *3*) *cart* ⇒ *bool*. *isolated_pai q1 q2 s* = (*isolated_qua q1 s* ∧ *isolated_qua q2 s* ∧ *HOL_Light_Import.INTER* (*conv0 q1*) (*conv0 q2*) ≠ *EMPTY*)

thm DEF_anchor:

anchor = (λ(*_1977005*::(*real*, *3*) *cart*) (*_1977006*::(*real*, *3*) *cart*) (*_1977007*::(*real*, *3*) *cart*) *_1977008*::(*real*, *3*) *cart* ⇒ *bool*. *SUBSET* (*INSERT _1977005* (*INSERT _1977006* (*INSERT _1977007 EMPTY*))) *_1977008* ∧ *d3 _1977006 _1977007* ≤ *sqrt* (*real_of_nat* (*8*::*nat*)) ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* ≤ *d3 _1977006 _1977007* ∧ *d3 _1977005 _1977006* ≤ *real_of_nat* (*2*::*nat*) ∗ *t0* ∧ *d3 _1977005 _1977007* ≤ *real_of_nat* (*2*::*nat*) ∗ *t0*)

thm Geomdetail.anchor:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*v1*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. *anchor v v1 v2 s* = (*SUBSET* (*INSERT v* (*INSERT v1* (*INSERT v2*

EMPTY))) *s* ∧ *d3 v1 v2* ≤ *sqrt* (*real_of_nat* (*8*::*nat*)) ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* ≤ *d3 v1 v2* ∧ *d3 v v1* ≤ *real_of_nat* (*2*::*nat*) ∗ *t0* ∧ *d3 v v2* ≤ *real_of_nat* (*2*::*nat*) ∗ *t0*)

thm DEF_anchor_points:

anchor_points = (λ(*_1977037*::(*real*, *3*) *cart*) (*_1977038*::(*real*, *3*) *cart*) *_1977039*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%14*::(*real*, *3*) *cart*. ∃ *t*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%14* (*real_of_nat* (*2*::*nat*) ∗ *t0* ≤ *d3 _1977037 _1977038* ∧ *anchor t _1977037 _1977038 _1977039*) *t*))

thm Geomdetail.anchor_points:

∀ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *anchor_points v w s* = *GSPEC* (λ*GEN%PVAR%14*::(*real*, *3*) *cart*. ∃ *t*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%14* (*real_of_nat* (*2*::*nat*) ∗ *t0* ≤ *d3 v w* ∧ *anchor t v w s*) *t*)

thm DEF_Q_SYS:

Q_SYS = (λ*_1977058*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%17*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *q*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%17* (*quasi_reg_tet q _1977058* ∨ *strict_qua q _1977058* ∧ (∃ (*c*::(*real*, *3*) *cart*) *d*::(*real*, *3*) *cart*. ∀ *qq*::(*real*, *3*) *cart* ⇒ *bool*. *IN c q* ∧ *IN d q* ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 c d* ∧ (*quasi_reg_tet qq _1977058* ∨ *strict_qua qq _1977058*) ∧ *HOL_Light_Import.INTER* (*conv0* (*INSERT c* (*INSERT d EMPTY*))) (*conv0 qq*) = *EMPTY*) ∨ *strict_qua q _1977058* ∧ ((*4*::*nat*) < *CARD* (*GSPEC* (λ*GEN%PVAR%15*::(*real*, *3*) *cart*. ∃ *t*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%15* (∃ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v q* ∧ *IN w q* ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 v w* ∧ *anchor t v w _1977058*) *t*)) ∨ *CARD* (*GSPEC* (λ*GEN%PVAR%16*::(*real*, *3*) *cart*. ∃ *t*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%16* (∃ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v q* ∧ *IN w q* ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 v w* ∧ *anchor t v w _1977058*) *t*)) = (*4*::*nat*) ∧ ¬ (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v q* ∧ *IN w q* ∧ *SUBSET* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) (*anchor_points v w _1977058*) ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 v w* ∧ *quartered_oct v w v1 v2 v3 v4 _1977058*)) ∨ (∃ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *q* = *INSERT v* (*INSERT w* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ *quartered_oct v w v1 v2 v3 v4 _1977058*) ∨ (∃ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. (*q* = *INSERT v* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∨ *q* = *INSERT v* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*)))) ∧ *interior_pos v v1 v3 v2 v4 _1977058* ∧ *anchor_points v1 v3 _1977058* = *INSERT v* (*INSERT v2* (*INSERT v4 EMPTY*)) ∧ *anchor_points v2 v4 _1977058* = *INSERT v* (*INSERT v1* (*INSERT v3 EMPTY*)))) *q*))

thm Geomdetail.Q_SYS:

∀ *s*::(*real*, *3*) *cart* ⇒ *bool*. *Q_SYS s* = *GSPEC* (λ*GEN%PVAR%17*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *q*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%17* (*quasi_reg_tet* q s ∨ strict_qua q s ∧ (∃ (c::(real, 3) cart) d::(real, 3) cart. ∀ qq::(real, 3) cart ⇒ bool. IN c q ∧ IN d q ∧ real_of_nat (2::nat) ∗ t0 < d3 c d ∧ (quasi_reg_tet qq s ∨ strict_qua qq s) ∧ HOL_Light_Import.INTER (conv0 (INSERT c (INSERT d EMPTY))) (conv0 qq) = EMPTY) ∨ strict_qua q s ∧ ((4::nat) < CARD (GSPEC (λGEN%PVAR%15::(real, 3) cart. ∃ t::(real, 3) cart. SETSPEC GEN%PVAR%15 (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v q ∧ IN w q ∧ real_of_nat (2::nat) ∗ t0 < d3 v w ∧ anchor t v w s) t)) ∨ CARD (GSPEC (λGEN%PVAR%16::(real, 3) cart. ∃ t::(real, 3) cart. SETSPEC GEN%PVAR%16 (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v q ∧ IN w q ∧ real_of_nat (2::nat) ∗ t0 < d3 v w ∧ anchor t v w s) t)) = (4::nat) ∧ ¬ (∃ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) (v::(real, 3) cart) w::(real, 3) cart. IN v q ∧ IN w q ∧ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) (anchor_points v w s) ∧ real_of_nat (2::nat) ∗ t0 < d3 v w ∧ quartered_oct v w v1 v2 v3 v4 s)) ∨ (∃ (v::(real, 3) cart) (w::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. q = INSERT v (INSERT w (INSERT v1 (INSERT v2 EMPTY))) ∧ quartered_oct v w v1 v2 v3 v4 s) ∨ (∃ (v::(real, 3) cart) (v1::(real, 3) cart) (v3::(real, 3) cart) (v2::(real, 3) cart) v4::(real, 3) cart. (q = INSERT v (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∨ q = INSERT v (INSERT v1 (INSERT v3 (INSERT v4 EMPTY)))) ∧ interior_pos v v1 v3 v2 v4 s ∧ anchor_points v1 v3 s = INSERT v (INSERT v2 (INSERT v4 EMPTY)) ∧ anchor_points v2 v4 s = INSERT v (INSERT v1 (INSERT v3 EMPTY)))) q)

thm DEF_barrier:

barrier = (λ_1977063::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%18::(real, 3) cart ⇒ bool. ∃ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. SETSPEC GEN%PVAR%18 (quasi_tri (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) _1977063 ∨ (∃ v4::(real, 3) cart. IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS _1977063))) (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))))

thm Geomdetail.barrier:

∀ s::(real, 3) cart ⇒ bool. barrier s = GSPEC (λGEN%PVAR%18::(real, 3) cart ⇒ bool. ∃ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. SETSPEC GEN%PVAR%18 (quasi_tri (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s ∨ (∃ v4::(real, 3) cart. IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS s))) (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))))

thm DEF_obstructed:

obstructed = (λ(_1977068::(real, 3) cart) (_1977069::(real, 3) cart) _1977070::(real, 3) cart ⇒ bool. ∃ bar::(real, 3) cart ⇒ bool. IN bar (barrier _1977070) ∧ HOL_Light_Import.INTER (conv0_2 (INSERT _1977068 (INSERT _1977069 EMPTY))) (conv_trg bar) ≠ EMPTY)

thm Geomdetail.obstructed:

$\forall$ (*s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *obstructed x y s* = ($\exists$ *bar*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN bar* (*barrier s*) $\wedge$ *HOL_Light_Import.INTER* (*conv0_2* (*INSERT x* (*INSERT y EMPTY*))) (*conv_trg bar*) $\neq$ *EMPTY*)

thm DEF_conv:

conv = *affsign sgn_ge EMPTY*

thm Collect_geom.conv:

$\forall$ *S*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *conv S* = *affsign sgn_ge EMPTY S*

thm DEF_obstruct:

obstruct = (λ(*_1977094*::(*real*, *3*) *cart*) (*_1977095*::(*real*, *3*) *cart*) *_1977096*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ *bar*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN bar* (*barrier _1977096*) $\wedge$ *HOL_Light_Import.INTER* (*conv0* (*INSERT _1977094* (*INSERT _1977095* *EMPTY*))) (*conv bar*) $\neq$ *EMPTY*)

thm Geomdetail.obstruct:

$\forall$ (*s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *obstruct x y s* = ($\exists$ *bar*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN bar* (*barrier s*) $\wedge$ *HOL_Light_Import.INTER* (*conv0* (*INSERT x* (*INSERT y EMPTY*))) (*conv bar*) $\neq$ *EMPTY*)

thm DEF_unobstructed:

unobstructed = (λ(*_1977115*::(*real*, *3*) *cart*) (*_1977116*::(*real*, *3*) *cart*) *_1977117*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\neg$ *obstructed _1977115 _1977116 _1977117*)

thm Geomdetail.unobstructed:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *unobstructed x y s* = ($\neg$ *obstructed x y s*)

thm DEF_VC_trg:

VC_trg = (λ(*_1977136*::(*real*, *3*) *cart*) *_1977137*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *GSPEC* (λ*GEN%PVAR%19*::(*real*, *3*) *cart*. $\exists$ *z*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%19* (*d3 _1977136 z* < *real_of_nat* (*2*::*nat*) $\wedge$ $\neg$ *obstructed _1977136 z _1977137* $\wedge$ ($\forall$ *y*::(*real*, *3*) *cart*. *IN y _1977137* $\wedge$ *_1977136* $\neq$ *y* $\wedge$ $\neg$ *obstructed z y _1977137* $\longrightarrow$ *d3 _1977136 z* < *d3 y z*)) *z*))

thm Geomdetail.VC_trg:

$\forall$ (*s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *x*::(*real*, *3*) *cart*. *VC_trg x s* = *GSPEC* (λ*GEN%PVAR%19*::(*real*, *3*) *cart*. $\exists$ *z*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%19* (*d3 x z* < *real_of_nat* (*2*::*nat*) $\wedge$ $\neg$ *obstructed x z s* $\wedge$ ($\forall$ *y*::(*real*, *3*) *cart*. *IN y s* $\wedge$ *x* $\neq$ *y* $\wedge$ $\neg$ *obstructed z y s* $\longrightarrow$ *d3 x z* < *d3 y z*)) *z*)

thm DEF_VC_INFI_trg:

VC_INFI_trg = (λ*_1977148*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *GSPEC* (λ*GEN%PVAR%20*::(*real*, *3*) *cart*. $\exists$ *z*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%20* ($\forall$ *x*::(*real*, *3*) *cart*. *IN x _1977148* $\wedge$ $\neg$ *IN z* (*VC_trg x _1977148*)) *z*))

thm Geomdetail.VC_INFI_trg:

∀ s::(real, 3) cart ⇒ bool. VC_INFI_trg s = GSPEC (λGEN%PVAR%20::(real, 3) cart. ∃ z::(real, 3) cart. SETSPEC GEN%PVAR%20 (∀ x::(real, 3) cart. IN x s ∧ ¬ IN z (VC_trg x s)) z)

thm DEF_lambda_x:

lambda_x = (λ(_1977153::(real, 3) cart) _1977154::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%21::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%21 (IN w _1977154 ∧ d3 w _1977153 < real_of_nat (2::nat) ∧ ¬ obstructed w _1977153 _1977154) w))

thm Geomdetail.lambda_x:

∀ (x::(real, 3) cart) s::(real, 3) cart ⇒ bool. lambda_x x s = GSPEC (λGEN%PVAR%21::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%21 (IN w s ∧ d3 w x < real_of_nat (2::nat) ∧ ¬ obstructed w x s) w)

thm DEF_lambda_y:

lambda_y = (λ(_1977165::(real, 3) cart) _1977166::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%22::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%22 (IN w _1977166 ∧ d3 w _1977165 < real_of_nat (2::nat) ∧ ¬ obstruct w _1977165 _1977166) w))

thm Geomdetail.lambda_y:

∀ (y::(real, 3) cart) s::(real, 3) cart ⇒ bool. lambda_y y s = GSPEC (λGEN%PVAR%22::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%22 (IN w s ∧ d3 w y < real_of_nat (2::nat) ∧ ¬ obstruct w y s) w)

thm DEF_VC:

VC = (λ(_1977177::(real, 3) cart) _1977178::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%23::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%23 (IN _1977177 (lambda_x x _1977178) ∧ (∀ w::(real, 3) cart. IN w (lambda_x x _1977178) ∧ w ≠ _1977177 ⟶ d3 x _1977177 < d3 x w)) x))

thm Geomdetail.VC:

∀ (s::(real, 3) cart ⇒ bool) v::(real, 3) cart. VC v s = GSPEC (λGEN%PVAR%23::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%23 (IN v (lambda_x x s) ∧ (∀ w::(real, 3) cart. IN w (lambda_x x s) ∧ w ≠ v ⟶ d3 x v < d3 x w)) x)

thm DEF_VC_INFI:

VC_INFI = (λ_1977189::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%24::(real, 3) cart. ∃ z::(real, 3) cart. SETSPEC GEN%PVAR%24 (∀ x::(real, 3) cart. ¬ IN z (VC x _1977189)) z))

thm Geomdetail.VC_INFI:

∀ s::(real, 3) cart ⇒ bool. VC_INFI s = GSPEC (λGEN%PVAR%24::(real, 3) cart. ∃ z::(real, 3) cart. SETSPEC GEN%PVAR%24 (∀ x::(real, 3) cart. ¬ IN z (VC x s)) z)

thm Geomdetail.trg_sub_bo:

$SUBSET\ (?A::?'a::type \Rightarrow bool)\ (?B::?'a::type \Rightarrow bool) = (\forall x::?'a::type.\ ?A\ x \longrightarrow ?B\ x)$

thm Geomdetail.trg_sub_se:

$SUBSET\ (?A::?'a::type \Rightarrow bool)\ (?B::?'a::type \Rightarrow bool) = (\forall x::?'a::type.\ IN\ x\ ?A \longrightarrow IN\ x\ ?B)$

thm Geomdetail.trg_setbool:

$IN\ (?x::?'a::type)\ (?A::?'a::type \Rightarrow bool) = ?A\ ?x$

thm Geomdetail.trg_boolset:

$(?A::?'a::type \Rightarrow bool)\ (?x::?'a::type) = IN\ ?x\ ?A$

thm Geomdetail.trg_in_union:

$IN\ (?x::?'a::type)\ (HOL\_Light\_Import.UNION\ (?A::?'a::type \Rightarrow bool)\ (?B::?'a::type \Rightarrow bool)) = (IN\ ?x\ ?A \lor IN\ ?x\ ?B)$

thm Geomdetail.not_in_set3:

$(\neg\ IN\ (?x::?'a::type)\ (GSPEC\ (\lambda GEN\%PVAR\%25::?'a::type.\ \exists z::?'a::type.\ SETSPEC\ GEN\%PVAR\%25\ ((?A::?'a::type \Rightarrow bool)\ z \land (?B::?'a::type \Rightarrow bool)\ z \land (?C::?'a::type \Rightarrow bool)\ z)\ z))) = (\neg\ ?A\ ?x \lor \neg\ ?B\ ?x \lor \neg\ ?C\ ?x)$

thm Collect_geom.trg_d3_sym:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ d3\ x\ y = d3\ y\ x$

thm Geomdetail.affine_hull_e:

$hull\ affine\ EMPTY = EMPTY$

thm Geomdetail.wlog:

$(\forall (a::?'a::type)\ b::?'a::type.\ (?P::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ a\ b = ?P\ b\ a \land ((?Q::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ a\ b \lor ?Q\ b\ a)) \longrightarrow (\exists (a::?'a::type)\ b::?'a::type.\ ?P\ a\ b) = (\exists (a::?'a::type)\ b::?'a::type.\ ?P\ a\ b \land ?Q\ a\ b)$

thm Geomdetail.wlog_real:

$(\forall (a::real)\ b::real.\ (?P::real \Rightarrow real \Rightarrow bool)\ a\ b = ?P\ b\ a) \longrightarrow (\exists (a::real)\ b::real.\ ?P\ a\ b) = (\exists (a::real)\ b::real.\ ?P\ a\ b \land a \leq b)$

thm Geomdetail.dkdx:

$\forall P::real \Rightarrow real \Rightarrow ?'d::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (a::real)\ (b::real)\ (u::?'d::type)\ (v::?'d::type)\ (m::?'c::type)\ (n::?'b::type)\ p::?'a::type.\ P\ a\ b\ u\ v\ m\ n\ p = P\ b\ a\ v\ u\ m\ n\ p) \longrightarrow (\exists (a::real)\ (b::real)\ (u::?'d::type)\ (v::?'d::type)\ (m::?'c::type)\ (n::?'b::type)\ p::?'a::type.\ P\ a\ b\ u\ v\ m\ n\ p) = (\exists (a::real)\ (b::real)\ (u::?'d::type)\ (v::?'d::type)\ (m::?'c::type)\ (n::?'b::type)\ p::?'a::type.\ P\ a\ b\ u\ v\ m\ n\ p \land a \leq b)$

thm Geomdetail.AFFINE_HULL_SINGLE:

$\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*INSERT x EMPTY*) = *INSERT x EMPTY*

thm Geomdetail.usefull:

(($\forall$ *x*::*?'a*::*type*. (*?a*::*?'a*::*type* $\Rightarrow$ *bool*) *x* $\longrightarrow$ (*?b*::*?'a*::*type* $\Rightarrow$ *bool*) *x*) $\longrightarrow$ ($\exists$ *x*::*?'a*::*type*. *?a x*) $\longrightarrow$ (*?c*::*bool*) $\longrightarrow$ (*?d*::*bool*)) = (($\forall$ *x*::*?'a*::*type*. *?a x* $\longrightarrow$ *?b x*) $\longrightarrow$ ($\exists$ *x*::*?'a*::*type*. *?a x* $\wedge$ *?b x*) $\longrightarrow$ *?c* $\longrightarrow$ *?d*)

thm Geomdetail.v1_in_convex3:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *IN v1* (*GSPEC* (λ*GEN%PVAR%26*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%26* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*))

thm Geomdetail.v3_in_convex3:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *IN v3* (*GSPEC* (λ*GEN%PVAR%27*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%27* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*))

thm Geomdetail.v1v2v3_in_convex3:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *IN v1* (*GSPEC* (λ*GEN%PVAR%28*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%28* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*)) $\wedge$ *IN v2* (*GSPEC* (λ*GEN%PVAR%29*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%29* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*)) $\wedge$ *IN v3* (*GSPEC* (λ*GEN%PVAR%30*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%30* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*))

thm Geomdetail.convex3:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. *convex* (*GSPEC* (λ*GEN%PVAR%31*::(*real*, *?'a*::*type*) *cart*. $\exists$ *t*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%31* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *t* = *vector_add* (% *a v1*) (*vector_add* (% *b v2*) (% *c v3*))) *t*))

thm Geomdetail.INTERS_SUBSET:

$\forall$ (*P*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *t0*::*?'a*::*type* $\Rightarrow$ *bool*. *P t0* $\longrightarrow$ *SUBSET* (*INTERS* (*GSPEC* (λ*GEN%PVAR%32*::*?'a*::*type* $\Rightarrow$ *bool*. $\exists$ *t*::*?'a*::*type* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%32* (*P t*) *t*))) *t0*

thm Geomdetail.SUM_TWO_RATIO:

((?a::real) + (?b::real) ≠ (0::real)) = (?a / (?a + ?b) + ?b / (?a + ?b) = (1::real))

thm Geomdetail.minconvex3:

∀ (t::(real, ?'a::type) cart ⇒ bool) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. convex t ∧ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) t ⟶ (∀ (a::real) (b::real) c::real. (0::real) ≤ a ∧ (0::real) ≤ b ∧ (0::real) ≤ c ∧ a + (b + c) = (1::real) ⟶ IN (vector_add (% a v1) (vector_add (% b v2) (% c v3))) t)

thm Geomdetail.OTHER_CONVEX_CONDI:

∀ s::(real, ?'a::type) cart ⇒ bool. convex s = (∀ (a::real) (b::real) (c::real) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s ∧ (0::real) ≤ a ∧ (0::real) ≤ b ∧ (0::real) ≤ c ∧ a + (b + c) = (1::real) ⟶ IN (vector_add (% a v1) (vector_add (% b v2) (% c v3))) s)

thm Geomdetail.convex3_in_inters:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. SUBSET (GSPEC (λGEN%PVAR%33::(real, ?'a::type) cart. ∃ t::(real, ?'a::type) cart. SETSPEC GEN%PVAR%33 (∃ (a::real) (b::real) c::real. (0::real) ≤ a ∧ (0::real) ≤ b ∧ (0::real) ≤ c ∧ a + (b + c) = (1::real) ∧ t = vector_add (% a v1) (vector_add (% b v2) (% c v3))) t)) (INTERS (GSPEC (λGEN%PVAR%34::(real, ?'a::type) cart ⇒ bool. ∃ t::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%34 (convex t ∧ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) t) t)))

thm Geomdetail.simpl_conv3:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) = GSPEC (λGEN%PVAR%36::(real, ?'a::type) cart. ∃ t::(real, ?'a::type) cart. SETSPEC GEN%PVAR%36 (∃ (a::real) (b::real) c::real. (0::real) ≤ a ∧ (0::real) ≤ b ∧ (0::real) ≤ c ∧ a + (b + c) = (1::real) ∧ t = vector_add (% a v1) (vector_add (% b v2) (% c v3))) t)

thm DEF_near:

near = (λ(_1978220::(real, ?'a::type) cart) (_1978221::real) _1978222::(real, ?'a::type) cart ⇒ bool. GSPEC (λGEN%PVAR%37::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%37 (IN x _1978222 ∧ distance (x, _1978220) < _1978221) x))

thm Geomdetail.near:

∀ (s::(real, ?'a::type) cart ⇒ bool) (v0::(real, ?'a::type) cart) r::real. near v0 r s = GSPEC (λGEN%PVAR%37::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%37 (IN x s ∧ distance (x, v0) < r) x)

thm DEF_near2t0:

near2t0 = (λ(_1978241::(real, ?'a::type) cart) _1978242::(real, ?'a::type) cart ⇒ bool. GSPEC (λGEN%PVAR%38::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%38 (IN x _1978242 ∧ distance (_1978241, x) < real_of_nat (2::nat) ∗ t0) x))

thm Geomdetail.near2t0:

∀ (s::(real, ?'a::type) cart ⇒ bool) v0::(real, ?'a::type) cart. near2t0 v0 s = GSPEC (λGEN%PVAR%38::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%38 (IN x s ∧ distance (v0, x) < real_of_nat (2::nat) ∗ t0) x)

thm DEF_e_plane:

e_plane = (λ(_1978253::(real, ?'a::type) cart) (_1978254::(real, ?'a::type) cart) _1978255::(real, ?'a::type) cart. _1978253 ≠ _1978254 ∧ distance (_1978253, _1978255) = distance (_1978254, _1978255))

thm Geomdetail.e_plane:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) x::(real, ?'a::type) cart. e_plane a b x = (a ≠ b ∧ distance (a, x) = distance (b, x))

thm DEF_min_dist:

min_dist = (λ(_1978274::(real, 3) cart) _1978275::(real, 3) cart ⇒ bool. (∃ u::(real, 3) cart. IN u _1978275 ∧ (∀ w::(real, 3) cart. IN w _1978275 ∧ u ≠ w ⟶ distance (u, _1978274) < distance (w, _1978274))) ∨ (∃ (u::(real, 3) cart) v::(real, 3) cart. IN u _1978275 ∧ IN v _1978275 ∧ u ≠ v ∧ distance (u, _1978274) = distance (v, _1978274) ∧ (∀ w::(real, 3) cart. IN w _1978275 ⟶ distance (u, _1978274) ≤ distance (w, _1978274))))

thm Geomdetail.min_dist:

∀ (s::(real, 3) cart ⇒ bool) x::(real, 3) cart. min_dist x s = ((∃ u::(real, 3) cart. IN u s ∧ (∀ w::(real, 3) cart. IN w s ∧ u ≠ w ⟶ distance (u, x) < distance (w, x))) ∨ (∃ (u::(real, 3) cart) v::(real, 3) cart. IN u s ∧ IN v s ∧ u ≠ v ∧ distance (u, x) = distance (v, x) ∧ (∀ w::(real, 3) cart. IN w s ⟶ distance (u, x) ≤ distance (w, x))))

thm Geomdetail.SET3_SUBSET:

∀ (a::?'a::type) (b::?'a::type) c::?'a::type. SUBSET (INSERT a (INSERT b (INSERT c EMPTY))) (?s::?'a::type ⇒ bool) = (IN a ?s ∧ IN b ?s ∧ IN c ?s)

thm Geomdetail.FINITE6:

∀ (a::?'b::type) (b::?'b::type) (c::?'b::type) (d::?'b::type) (e::?'b::type) f::?'b::type. FINITE EMPTY ∧ FINITE (INSERT a EMPTY) ∧ FINITE (INSERT a (INSERT b EMPTY)) ∧ FINITE (INSERT a (INSERT b (INSERT c EMPTY)))

$\wedge$ *FINITE* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*)))) $\wedge$ *FINITE* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d* (*INSERT e EMPTY*))))) $\wedge$ *FINITE* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d* (*INSERT e* (*INSERT f EMPTY*))))))

thm Geomdetail.CONV_EM:

conv EMPTY = *EMPTY*

thm Geomdetail.CONV_SING:

$\forall$ *u*::(*real*, *?'a*::*type*) *cart*. *conv* (*INSERT u EMPTY*) = *INSERT u EMPTY*

thm Geomdetail.NOV9:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *x* = *y* $\wedge$ *y* = *z* $\longrightarrow$ *conv* (*INSERT y* (*INSERT z EMPTY*)) = *GSPEC* (λ*GEN%PVAR%39*::(*real*, *?'a*::*type*) *cart*. $\exists$ *w*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%39* ($\exists$ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ (*0*::*real*) $\leq$ *c* $\wedge$ *a* + (*b* + *c*) = (*1*::*real*) $\wedge$ *w* = *vector_add* (% *a y*) (*vector_add* (% *b y*) (% *c z*))) *w*)

thm Geomdetail.IN_ACT_SING:

$\forall$ (*a*::*?'a*::*type*) *x*::*?'a*::*type*. *INSERT a EMPTY x* = (*a* = *x*) $\wedge$ *IN x* (*INSERT a EMPTY*) = (*x* = *a*) $\wedge$ *INSERT a EMPTY a*

thm Geomdetail.NOV10:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *x* = *y* $\longrightarrow$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. (*y* = *x*) = ($\exists$ (*a*::*real*) *b*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *a y*) (% *b y*)))

thm Geomdetail.IN_SET2:

$\forall$ (*a*::*?'a*::*type*) (*b*::*?'a*::*type*) *x*::*?'a*::*type*. *IN x* (*INSERT a* (*INSERT b EMPTY*)) = (*x* = *a* $\vee$ *x* = *b*) $\wedge$ *INSERT a* (*INSERT b EMPTY*) *x* = (*x* = *a* $\vee$ *x* = *b*)

thm Geomdetail.VSUM_DIS2:

$\forall$ (*x*::*?'b*::*type*) (*y*::*?'b*::*type*) *f*::*?'b*::*type* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *x* $\neq$ *y* $\longrightarrow$ *vsum* (*INSERT x* (*INSERT y EMPTY*)) *f* = *vector_add* (*f x*) (*f y*)

thm Geomdetail.SUM_DIS2:

$\forall$ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *f*::*?'a*::*type* $\Rightarrow$ *real*. *x* $\neq$ *y* $\longrightarrow$ *sum* (*INSERT x* (*INSERT y EMPTY*)) *f* = *f x* + *f y*

thm Collect_geom.TRUONG_LEMMA:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *x'*::(*real*, *?'a*::*type*) *cart*. ($\exists$ *f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*. *x'* = *vector_add* (% (*f x*) *x*) (% (*f y*) *y*) $\wedge$ ((*0*::*real*) $\leq$ *f x* $\wedge$ (*0*::*real*) $\leq$ *f y*) $\wedge$ *f x* + *f y* = (*1*::*real*)) = ($\exists$ (*a*::*real*) *b*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *x'* = *vector_add* (% *a x*) (% *b y*))

thm Geomdetail.NOV11:

$\forall z::(real, ?'a::type)\ cart.\ INSERT\ z\ EMPTY = GSPEC\ (\lambda GEN\%PVAR\%40::(real, ?'a::type)\ cart.\ \exists w::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%40\ (\exists (a::real)\ (b::real)\ c::real.\ (0::real) \le a \wedge (0::real) \le b \wedge (0::real) \le c \wedge a + (b + c) = (1::real) \wedge w = vector\_add\ (\%\ a\ z)\ (vector\_add\ (\%\ b\ z)\ (\%\ c\ z)))\ w)$

thm Geomdetail.CONV2_CONV3:

$\forall (x'::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ (\exists (a::real)\ b::real.\ (0::real) \le a \wedge (0::real) \le b \wedge a + b = (1::real) \wedge x' = vector\_add\ (\%\ a\ y)\ (\%\ b\ z)) \longrightarrow (\exists (a::real)\ (b::real)\ c::real.\ (0::real) \le a \wedge (0::real) \le b \wedge (0::real) \le c \wedge a + (b + c) = (1::real) \wedge x' = vector\_add\ (\%\ a\ y)\ (vector\_add\ (\%\ b\ y)\ (\%\ c\ z)))$

thm Geomdetail.CONV3_CONV2:

$\forall (x'::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ (\exists (a::real)\ (b::real)\ c::real.\ (0::real) \le a \wedge (0::real) \le b \wedge (0::real) \le c \wedge a + (b + c) = (1::real) \wedge x' = vector\_add\ (\%\ a\ y)\ (vector\_add\ (\%\ b\ y)\ (\%\ c\ z))) \longrightarrow (\exists (a::real)\ b::real.\ (0::real) \le a \wedge (0::real) \le b \wedge a + b = (1::real) \wedge x' = vector\_add\ (\%\ a\ y)\ (\%\ b\ z))$

thm Geomdetail.CONV_SET2:

$\forall (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ conv\ (INSERT\ x\ (INSERT\ y\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%41::(real, ?'a::type)\ cart.\ \exists w::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%41\ (\exists (a::real)\ b::real.\ (0::real) \le a \wedge (0::real) \le b \wedge a + b = (1::real) \wedge w = vector\_add\ (\%\ a\ x)\ (\%\ b\ y))\ w)$

thm Geomdetail.CONV3_A_EQ:

$\forall (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ z::(real, ?'a::type)\ cart.\ x = y \vee y = z \vee z = x \longrightarrow conv\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%42::(real, ?'a::type)\ cart.\ \exists w::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%42\ (\exists (a::real)\ (b::real)\ c::real.\ (0::real) \le a \wedge (0::real) \le b \wedge (0::real) \le c \wedge a + (b + c) = (1::real) \wedge w = vector\_add\ (\%\ a\ x)\ (vector\_add\ (\%\ b\ y)\ (\%\ c\ z)))\ w)$

thm Geomdetail.VSUM_DIS3:

$\forall (x::?'b::type)\ (y::?'b::type)\ (z::?'b::type)\ f::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ \neg\ (x = y \vee y = z \vee z = x) \longrightarrow vsum\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY)))\ f = vector\_add\ (f\ x)\ (vector\_add\ (f\ y)\ (f\ z))$

thm Geomdetail.SUM_DIS3:

$\forall (x::?'a::type)\ (y::?'a::type)\ (z::?'a::type)\ f::?'a::type \Rightarrow real.\ \neg\ (x = y \vee y = z \vee z = x) \longrightarrow sum\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY)))\ f = f\ x + (f\ y + f\ z)$

thm Geomdetail.CONV_SET3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.$ $conv\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%43::(real,$ $?'a::type)\ cart.\ \exists\ w::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%43\ (\exists\ (a::real)$ $(b::real)\ c::real.\ (0::real) \leq a \wedge (0::real) \leq b \wedge (0::real) \leq c \wedge a + (b + c) =$ $(1::real) \wedge w = vector\_add\ (\%\ a\ x)\ (vector\_add\ (\%\ b\ y)\ (\%\ c\ z)))\ w)$

thm Geomdetail.CONV3_EQ:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.$ $conv\_trg\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = conv\ (INSERT\ x$ $(INSERT\ y\ (INSERT\ z\ EMPTY)))$

thm Geomdetail.CONV_BAR_EQ:

$\forall (bar::(real,\ 3)\ cart \Rightarrow bool)\ s::(real,\ 3)\ cart \Rightarrow bool.\ IN\ bar\ (barrier\ s) \longrightarrow$ $conv\ bar = conv\_trg\ bar$

thm Geomdetail.OBSTRUCT_EQ:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ s::(real,\ 3)\ cart \Rightarrow bool.\ obstruct\ x\ y\ s$ $= obstructed\ x\ y\ s$

thm Geomdetail.CARD_CLAUSES_IMP:

$\forall (a::?'a::type)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (INSERT\ a\ s) \leq Suc$ $(CARD\ s) \wedge (IN\ a\ s \longrightarrow CARD\ (INSERT\ a\ s) = CARD\ s) \wedge (\neg\ IN\ a\ s \longrightarrow$ $CARD\ (INSERT\ a\ s) = Suc\ (CARD\ s))$

thm Geomdetail.CARD_SING:

$\forall\ a::?'a::type.\ CARD\ (INSERT\ a\ EMPTY) = (1::nat)$

thm Geomdetail.CARD_SET2:

$\forall (a::?'a::type)\ b::?'a::type.\ (CARD\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = (2::nat))$ $= (a \neq b) \wedge (CARD\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = (1::nat)) = (a =$ $b)$

thm Geomdetail.CARD_EQUATION:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow CARD$ $(HOL\_Light\_Import.UNION\ s\ t) + CARD\ (HOL\_Light\_Import.INTER\ s\ t) =$ $CARD\ s + CARD\ t$

thm Geomdetail.CARD_DISJOINT:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow (CARD\ s$ $+ CARD\ t = CARD\ (HOL\_Light\_Import.UNION\ s\ t)) = (HOL\_Light\_Import.INTER$ $s\ t = EMPTY)$

thm Geomdetail.CARD2:

$\forall (a::?'a::type)\ b::?'a::type.\ CARD\ (INSERT\ a\ (INSERT\ b\ EMPTY)) \leq (2::nat)$ $\wedge (CARD\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = (2::nat)) = (a \neq b)$

thm Geomdetail.CARD3:

$\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \leq (3::nat) \wedge (CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) = (3::nat)) = (\neg\ (a = b \vee b = c \vee c = a))$

thm Geomdetail.CARD4:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \leq (4::nat) \wedge (CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) = (4::nat)) = (\neg\ IN\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))) \wedge \neg\ (b = c \vee c = d \vee d = b))$

thm Geomdetail.CARD5:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ (d::?'a::type)\ e::?'a::type.\ CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ (INSERT\ e\ EMPTY))))) \leq (5::nat) \wedge (CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ (INSERT\ e\ EMPTY))))) = (5::nat)) = (\neg\ IN\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ (INSERT\ e\ EMPTY)))) \wedge \neg\ IN\ b\ (INSERT\ c\ (INSERT\ d\ (INSERT\ e\ EMPTY))) \wedge \neg\ (c = d \vee d = e \vee e = c))$

thm Geomdetail.set_3elements:

$(\exists (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ \neg\ (a = b \vee b = c \vee c = a) \wedge INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)) = INSERT\ (?v1.0::?'a::type)\ (INSERT\ (?v2.0::?'a::type)\ (INSERT\ (?v3.0::?'a::type)\ EMPTY))) = (\neg\ (?v1.0 = ?v2.0 \vee ?v2.0 = ?v3.0 \vee ?v3.0 = ?v1.0))$

thm Geomdetail.db_t0_conjunct0:

$real\_of\_nat\ (2::nat) * t0 = real\_of\_nat\ (2::nat) * DECIMAL\ (1255::nat)\ (1000::nat)$

thm Geomdetail.db_t0_conjunct2:

$real\_of\_nat\ (2::nat) * DECIMAL\ (1255::nat)\ (1000::nat) = DECIMAL\ (251::nat)\ (100::nat)$

thm Geomdetail.db_t0:

$real\_of\_nat\ (2::nat) * t0 = real\_of\_nat\ (2::nat) * DECIMAL\ (1255::nat)\ (1000::nat) \wedge real\_of\_nat\ (2::nat) * t0 = DECIMAL\ (251::nat)\ (100::nat) \wedge real\_of\_nat\ (2::nat) * DECIMAL\ (1255::nat)\ (1000::nat) = DECIMAL\ (251::nat)\ (100::nat)$

thm Geomdetail.without_lost:

$\forall (P::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ (\forall (a::?'a::type)\ b::?'a::type.\ P\ a\ b = P\ b\ a) \wedge (\exists (a::?'a::type)\ b::?'a::type.\ P\ a\ b \wedge x = a) \longrightarrow (\exists (a::?'a::type)\ b::?'a::type.\ P\ a\ b \wedge (x = a \vee x = b))$

thm Geomdetail.condi_of_wlofg:

$(\forall (a::?'a::type)\ b::?'a::type.\ (?P::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ a\ b = ?P\ b\ a) \longrightarrow (\exists (a::?'a::type)\ b::?'a::type.\ ?P\ a\ b \wedge ((?x::?'a::type) = a \vee ?x = b)) = (\exists (a::?'a::type)\ b::?'a::type.\ ?P\ a\ b \wedge ?x = a)$

thm Geomdetail.CARD_SET_OF_LIST_LE:

∀ *l::?'a::type list. CARD (set_of_list l) ≤ length l*

thm Geomdetail.HAS_SIZE_SET_OF_LIST:

∀ *l::?'a::type list. HAS_SIZE (set_of_list l) (length l) = PAIRWISE op ≠ l*

thm Geomdetail.HAS_SIZE_SET_OF_LIST_4:

∀ *(a::?'a::type) (b::?'a::type) (c::?'a::type) d::?'a::type. HAS_SIZE (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) (4::nat) = PAIRWISE op ≠ [a, b, c, d]*

thm Geomdetail.SET_OF_4:

∀ *(a::?'a::type) (b::?'a::type) (c::?'a::type) d::?'a::type. (∃ (v1::?'a::type) (v2::?'a::type) (v3::?'a::type) v4::?'a::type. ¬ (v1 = v2 ∨ v3 = v4) ∧ HOL_Light_Import.INTER (INSERT v1 (INSERT v2 EMPTY)) (INSERT v3 (INSERT v4 EMPTY)) = EMPTY ∧ INSERT a (INSERT b (INSERT c (INSERT d EMPTY))) = INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) = (¬ (a = b ∨ a = c ∨ a = d ∨ b = c ∨ b = d ∨ c = d))*

thm Geomdetail.def_simplex:

∀ *(s::(real, 3) cart ⇒ bool) (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. geomdetail'simplex (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) s = (packing s ∧ SUBSET (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) s ∧ ¬ (a = b ∨ a = c ∨ a = d ∨ b = c ∨ b = d ∨ c = d))*

thm Geomdetail.strict_qua2_imply_strict_qua:

∀ *(q::(real, 3) cart ⇒ bool) (d::(real, 3) cart ⇒ bool) s::(real, 3) cart ⇒ bool. strict_qua2 q d s ⟶ strict_qua q s*

thm Geomdetail.strict_qua2_interior_pos:

∀ *(s::(real, 3) cart ⇒ bool) (v::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. interior_pos v v1 v3 v2 v4 s ⟶ strict_qua2 (INSERT v (INSERT v1 (INSERT v3 (INSERT v2 EMPTY)))) (INSERT v1 (INSERT v3 EMPTY)) s ∧ strict_qua2 (INSERT v (INSERT v1 (INSERT v3 (INSERT v4 EMPTY)))) (INSERT v1 (INSERT v3 EMPTY)) s ∧ strict_qua2 (INSERT v (INSERT v2 (INSERT v4 (INSERT v1 EMPTY)))) (INSERT v2 (INSERT v4 EMPTY)) s ∧ strict_qua2 (INSERT v (INSERT v2 (INSERT v4 (INSERT v3 EMPTY)))) (INSERT v2 (INSERT v4 EMPTY)) s*

thm Geomdetail.RELATE_Q_SYS:

∀ *(q::(real, 3) cart ⇒ bool) s::(real, 3) cart ⇒ bool. (∃ (v::(real, 3) cart) (v1::(real, 3) cart) (v3::(real, 3) cart) (v2::(real, 3) cart) v4::(real, 3) cart. (q = INSERT v (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∨ q = INSERT v (INSERT v1 (INSERT v3 (INSERT v4 EMPTY)))) ∧ interior_pos v v1 v3 v2* *v4 s* ∧ *anchor_points v1 v3 s* = *INSERT v* (*INSERT v2* (*INSERT v4 EMPTY*)) ∧ *anchor_points v2 v4 s* = *INSERT v* (*INSERT v1* (*INSERT v3 EMPTY*))) ⟶ *strict_qua q s*

thm Geomdetail.strict_qua_in_oct:

∀ (*q*::(*real*, *3*) *cart* ⇒ *bool*) *s*::(*real*, *3*) *cart* ⇒ *bool*. (∃ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *q* = *INSERT v* (*INSERT w* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ *quartered_oct v w v1 v2 v3 v4 s*) ⟶ *strict_qua q s*

thm Geomdetail.WHEN_IN_Q_SYS:

∀ (*q*::(*real*, *3*) *cart* ⇒ *bool*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *IN q* (*Q_SYS s*) ⟶ *quasi_reg_tet q s* ∨ *strict_qua q s* ∨ (∃ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. (*q* = *INSERT v* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∨ *q* = *INSERT v* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*)))) ∧ *interior_pos v v1 v3 v2 v4 s* ∧ *anchor_points v1 v3 s* = *INSERT v* (*INSERT v2* (*INSERT v4 EMPTY*)) ∧ *anchor_points v2 v4 s* = *INSERT v* (*INSERT v1* (*INSERT v3 EMPTY*)))

thm Geomdetail.CASES_OF_Q_SYS:

∀ (*q*::(*real*, *3*) *cart* ⇒ *bool*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *IN q* (*Q_SYS s*) ⟶ *quasi_reg_tet q s* ∨ *strict_qua q s*

thm Geomdetail.simplex_interior_pos:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *interior_pos v v1 v3 v2 v4 s* ⟶ *geomdetail'simplex* (*INSERT v* (*INSERT v1* (*INSERT v3* (*INSERT v2 EMPTY*)))) *s* ∧ *geomdetail'simplex* (*INSERT v* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*)))) *s* ∧ *geomdetail'simplex* (*INSERT v* (*INSERT v2* (*INSERT v4* (*INSERT v1 EMPTY*)))) *s* ∧ *geomdetail'simplex* (*INSERT v* (*INSERT v2* (*INSERT v4* (*INSERT v3 EMPTY*)))) *s*

thm Geomdetail.QUA_TET_IMPLY_QUA_TRI:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. (∃ *v4*::(*real*, *3*) *cart*. *quasi_reg_tet* (*HOL_Light_Import.UNION* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) (*INSERT v4 EMPTY*)) *s*) ⟶ *quasi_tri* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) *s*

thm Geomdetail.PAIR_D3:

∀ (*i*::(*real*, *3*) *cart*) (*j*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *INSERT u* (*INSERT w EMPTY*) = *INSERT i* (*INSERT j EMPTY*) ⟶ *d3 i j* = *d3 u w*

thm Geomdetail.PAIR_DIST:

∀ (*i*::(*real*, *?'a*::*type*) *cart*) (*j*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *INSERT u* (*INSERT w EMPTY*) = *INSERT i* (*INSERT j EMPTY*) ⟶ *distance* (*i*, *j*) = *distance* (*u*, *w*)

thm Geomdetail.DIAGONAL_QUA:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (q::(real, 3) cart ⇒ bool) s::(real, 3) cart ⇒ bool. IN a q ∧ IN b q ∧ real_of_nat (2::nat) ∗ t0 < d3 a b ∧ quarter q s ⟶ diagonal a b q s

thm Geomdetail.NOV2:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) v4::(real, 3) cart. (∃ (i::(real, 3) cart) j::(real, 3) cart. IN i (INSERT a (INSERT b (INSERT c EMPTY))) ∧ IN j (INSERT a (INSERT b (INSERT c EMPTY))) ∧ i ≠ j ∧ DECIMAL (251::nat) (100::nat) < d3 i j) ∧ (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v (INSERT a (INSERT b (INSERT c (INSERT v4 EMPTY)))) ∧ IN w (INSERT a (INSERT b (INSERT c (INSERT v4 EMPTY)))) ∧ DECIMAL (251::nat) (100::nat) ≤ d3 v w ∧ d3 v w ≤ sqrt (real_of_nat (8::nat)) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x (INSERT a (INSERT b (INSERT c (INSERT v4 EMPTY)))) ∧ IN y (INSERT a (INSERT b (INSERT c (INSERT v4 EMPTY)))) ∧ INSERT x (INSERT y EMPTY) ≠ INSERT v (INSERT w EMPTY) ⟶ d3 x y ≤ DECIMAL (251::nat) (100::nat))) ⟶ (∃ (i::(real, 3) cart) j::(real, 3) cart. IN i (INSERT a (INSERT b (INSERT c EMPTY))) ∧ IN j (INSERT a (INSERT b (INSERT c EMPTY))) ∧ DECIMAL (251::nat) (100::nat) < d3 i j ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x (INSERT a (INSERT b (INSERT c EMPTY))) ∧ IN y (INSERT a (INSERT b (INSERT c EMPTY))) ∧ INSERT x (INSERT y EMPTY) ≠ INSERT i (INSERT j EMPTY) ⟶ d3 x y ≤ DECIMAL (251::nat) (100::nat)))

thm Geomdetail.db_t0_conjunct1:

real_of_nat (2::nat) ∗ t0 = DECIMAL (251::nat) (100::nat)

thm Geomdetail.TRIANGLE_IN_STRICT_QUA:

∀ (s::(real, 3) cart ⇒ bool) (a::(real, 3) cart) (b::(real, 3) cart) c::(real, 3) cart. (∃ v4::(real, 3) cart. strict_qua (HOL_Light_Import.UNION (INSERT a (INSERT b (INSERT c EMPTY))) (INSERT v4 EMPTY)) s) ⟶ quasi_tri (INSERT a (INSERT b (INSERT c EMPTY))) s ∨ (∃ (aa::(real, 3) cart) bb::(real, 3) cart. SUBSET (INSERT aa (INSERT bb EMPTY)) (INSERT a (INSERT b (INSERT c EMPTY))) ∧ DECIMAL (251::nat) (100::nat) < distance (aa, bb) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. INSERT x (INSERT y EMPTY) ≠ INSERT aa (INSERT bb EMPTY) ∧ SUBSET (INSERT x (INSERT y EMPTY)) (INSERT a (INSERT b (INSERT c EMPTY))) ⟶ distance (x, y) ≤ DECIMAL (251::nat) (100::nat)))

thm Geomdetail.xxii:

real_of_nat (2::nat) ∗ DECIMAL (1255::nat) (1000::nat) < sqrt (real_of_nat (8::nat))

thm Geomdetail.A_LEMMA:

∀ (s::(real, 3) cart ⇒ bool) (v0::(real, 3) cart) (v1::(real, 3) cart) v2::(real, 3) cart. quasi_tri (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) s ⟶

(∀ (xx::(real, 3) cart) yy::(real, 3) cart. xx ≠ yy ∧ SUBSET (INSERT xx (INSERT yy EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ real_of_nat (2::nat) ≤ distance (xx, yy) ∧ distance (xx, yy) ≤ sqrt (real_of_nat (8::nat))) ∧ ((∀ (aa::(real, 3) cart) bb::(real, 3) cart. IN aa (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ∧ IN bb (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ distance (aa, bb) ≤ DECIMAL (251::nat) (100::nat)) ∨ (∃ (aa::(real, 3) cart) bb::(real, 3) cart. SUBSET (INSERT aa (INSERT bb EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ∧ DECIMAL (251::nat) (100::nat) < distance (aa, bb) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. INSERT x (INSERT y EMPTY) ≠ INSERT aa (INSERT bb EMPTY) ∧ SUBSET (INSERT x (INSERT y EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ distance (x, y) ≤ DECIMAL (251::nat) (100::nat))))

thm Geomdetail.NOV1:

∀ (s::(real, 3) cart ⇒ bool) (v0::(real, 3) cart) (v1::(real, 3) cart) v2::(real, 3) cart. (∃ v4::(real, 3) cart. quasi_reg_tet (HOL_Light_Import.UNION (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) (INSERT v4 EMPTY)) s) ⟶ (∀ (xx::(real, 3) cart) yy::(real, 3) cart. xx ≠ yy ∧ SUBSET (INSERT xx (INSERT yy EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ real_of_nat (2::nat) ≤ distance (xx, yy) ∧ distance (xx, yy) ≤ sqrt (real_of_nat (8::nat))) ∧ ((∀ (aa::(real, 3) cart) bb::(real, 3) cart. IN aa (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ∧ IN bb (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ distance (aa, bb) ≤ DECIMAL (251::nat) (100::nat)) ∨ (∃ (aa::(real, 3) cart) bb::(real, 3) cart. SUBSET (INSERT aa (INSERT bb EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ∧ DECIMAL (251::nat) (100::nat) < distance (aa, bb) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. INSERT x (INSERT y EMPTY) ≠ INSERT aa (INSERT bb EMPTY) ∧ SUBSET (INSERT x (INSERT y EMPTY)) (INSERT v1 (INSERT v2 (INSERT v0 EMPTY))) ⟶ distance (x, y) ≤ DECIMAL (251::nat) (100::nat))))

thm Geomdetail.DOT_SUB_ADD:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. dot b b − dot a a = dot (vector_sub b a) (vector_add b a)

thm Geomdetail.DIST_LT_HALF_PLANE:

∀ (x::(real, ?'a::type) cart) (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. (distance (x, a) < distance (x, b)) = ((0::real) < dot (vector_sub a b) (vector_sub (% (real_of_nat (2::nat)) x) (vector_add a b)))

thm Geomdetail.OR_IMP_EX:

∀ (a::bool) (b::bool) c::bool. (a ∨ b ⟶ c) = ((a ⟶ c) ∧ (b ⟶ c))

thm Geomdetail.HALF_PLANE_CONV:

∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. convex (GSPEC (λGEN%PVAR%44::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%44 (distance (x, a) < distance (x, b)) x))

thm Geomdetail.HALF_PLANE_CONV_EP:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (u::real)\ v::real.\ distance\ (x,\ a) < distance\ (x,\ b) \wedge distance\ (y,\ a) < distance\ (y,\ b) \wedge (0::real) \leq u \wedge (0::real) \leq v \wedge u + v = (1::real) \longrightarrow distance\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y),\ a) < distance\ (vector\_add\ (\%\ u\ x)\ (\%\ v\ y),\ b)$

thm Geomdetail.VORONOI_CONV:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ convex\ (voronoi\_open\ s\ v)$

thm Geomdetail.CONVEX_IM_CONV2_SU:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ convex\ s \wedge IN\ u\ s \wedge IN\ v\ s \longrightarrow SUBSET\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ s$

thm Geomdetail.U_IN_CONV2:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ IN\ u\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY)))$

thm Geomdetail.UV_IN_CONV2:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ IN\ u\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY))) \wedge IN\ v\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY)))$

thm Geomdetail.CONVEX_AS_CONV2:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ s \longrightarrow (\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ (IN\ u\ s \wedge IN\ v\ s) = SUBSET\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ s)$

thm Geomdetail.CONV0_SING:

$\forall x::(real,\ ?'a::type)\ cart.\ conv0\ (INSERT\ x\ EMPTY) = INSERT\ x\ EMPTY$

thm Geomdetail.NOV10′:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ x = y \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ (y = x) = (\exists (a::real)\ b::real.\ (0::real) < a \wedge (0::real) < b \wedge a + b = (1::real) \wedge x = vector\_add\ (\%\ a\ y)\ (\%\ b\ y)))$

thm Collect_geom.CONV0_SET2:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ conv0\ (INSERT\ x\ (INSERT\ y\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%45::(real,\ ?'a::type)\ cart.\ \exists w::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%45\ (\exists (a::real)\ b::real.\ (0::real) < a \wedge (0::real) < b \wedge a + b = (1::real) \wedge w = vector\_add\ (\%\ a\ x)\ (\%\ b\ y))\ w)$

thm Geomdetail.CONV02_SU_CONV2:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ SUBSET\ (conv0\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (conv\ (INSERT\ u\ (INSERT\ v\ EMPTY)))$

thm Geomdetail.CONVEX_IM_CONV02_SU:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ convex\ s \wedge IN\ u\ s \wedge IN\ v\ s \longrightarrow SUBSET\ (conv0\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ s$

thm Geomdetail.PAIR_EQ_EXPAND:

$(INSERT\ (?a::?'a::type)\ (INSERT\ (?b::?'a::type)\ EMPTY) = INSERT\ (?c::?'a::type)\ (INSERT\ (?d::?'a::type)\ EMPTY)) = (?a = ?c \wedge ?b = ?d \vee ?a = ?d \wedge ?b = ?c)$

thm Geomdetail.IN_SET3:

$IN\ (?x::?'a::type)\ (INSERT\ (?a::?'a::type)\ (INSERT\ (?b::?'a::type)\ (INSERT\ (?c::?'a::type)\ EMPTY))) = (?x = ?a \vee ?x = ?b \vee ?x = ?c)$

thm Geomdetail.IN_SET4:

$IN\ (?x::?'a::type)\ (INSERT\ (?a::?'a::type)\ (INSERT\ (?b::?'a::type)\ (INSERT\ (?c::?'a::type)\ (INSERT\ (?d::?'a::type)\ EMPTY)))) = (?x = ?a \vee ?x = ?b \vee ?x = ?c \vee ?x = ?d)$

thm Geomdetail.FOUR_IN:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ IN\ a\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \wedge IN\ b\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \wedge IN\ c\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \wedge IN\ d\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))))$

thm Geomdetail.DIA_OF_QUARTER:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (q::(real,\ 3)\ cart \Rightarrow bool)\ s::(real,\ 3)\ cart \Rightarrow bool.\ diagonal\ a\ b\ q\ s \longrightarrow real\_of\_nat\ (2::nat) * t0 \leq d3\ a\ b \wedge d3\ a\ b \leq sqrt\ (real\_of\_nat\ (8::nat))$

thm Geomdetail.D3_REFL:

$\forall x::(real,\ 3)\ cart.\ d3\ x\ x = (0::real)$

thm Geomdetail.db_t0_sq8:

$DECIMAL\ (251::nat)\ (100::nat) < sqrt\ (real\_of\_nat\ (8::nat))$

thm Geomdetail.SUB_PACKING:

$\forall (sub::(real,\ 3)\ cart \Rightarrow bool)\ s::(real,\ 3)\ cart \Rightarrow bool.\ packing\ s \wedge SUBSET\ sub\ s \longrightarrow (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ IN\ x\ sub \wedge IN\ y\ sub \wedge x \neq y \longrightarrow real\_of\_nat\ (2::nat) \leq d3\ x\ y)$

thm Geomdetail.SHORT_EDGES:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ d3\ c\ a \leq real\_of\_nat\ (2::nat) * t0 \wedge d3\ c\ b \leq real\_of\_nat\ (2::nat) * t0 \wedge (\forall aa::(real,\ 3)\ cart.\ IN\ aa\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \longrightarrow d3\ aa\ w \leq real\_of\_nat\ (2::nat) * t0) \longrightarrow (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ IN\ x$

(*INSERT a* (*INSERT b* (*INSERT c* (*INSERT w EMPTY*)))) ∧ *IN y* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT w EMPTY*)))) ∧ *INSERT x* (*INSERT y EMPTY*) ≠ *INSERT a* (*INSERT b EMPTY*) ⟶ *d3 x y* ≤ *real_of_nat* (*2*::*nat*) ∗ *t0*)

thm Geomdetail.IN_Q_IMP_BAR:

IN (*INSERT* (*?v0.0*::(*real*, *3*) *cart*) (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*)))) (*Q_SYS* (*?s*::(*real*, *3*) *cart* ⇒ *bool*)) ⟶ *IN* (*INSERT ?v0.0* (*INSERT ?v1.0* (*INSERT ?v2.0* *EMPTY*))) (*barrier ?s*)

thm Geomdetail.v_near2t0_v:

IN (*?w*::(*real*, *?'a*::*type*) *cart*) (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⟶ *IN ?w* (*near2t0 ?w ?s*)

thm Geomdetail.in_VC:

∀ (*w*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) *x*::(*real*, *3*) *cart*. *IN w s* ∧ *distance* (*w*, *x*) < *real_of_nat* (*2*::*nat*) ∧ (∀ *w'*::(*real*, *3*) *cart*. *s w'* ∧ *w'* ≠ *w* ⟶ *distance* (*x*, *w*) < *distance* (*x*, *w'*)) ∧ ¬ *obstructed w x s* ⟶ *IN x* (*VC w s*)

thm Geomdetail.VC_DISJOINT:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. (*a* ≠ *b* ⟶ *HOL_Light_Import.INTER* (*VC a s*) (*VC b s*) = *EMPTY*) ∧ *HOL_Light_Import.INTER* (*VC a s*) (*VC_INFI s*) = *EMPTY*

thm Geomdetail.DIAGONAL_STRICT_QUA:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*q*::(*real*, *3*) *cart* ⇒ *bool*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *IN a q* ∧ *IN b q* ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 a b* ∧ *strict_qua q s* ⟶ *diagonal a b q s*

thm Geomdetail.DIAGONAL_BARRIER:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *bar*::(*real*, *3*) *cart* ⇒ *bool*. *IN v1 bar* ∧ *IN v2 bar* ∧ *IN bar* (*barrier s*) ∧ *real_of_nat* (*2*::*nat*) ∗ *t0* < *d3 v1 v2* ⟶ (∃ *w*::(*real*, *3*) *cart*. *diagonal v1 v2* (*INSERT w bar*) *s*)

thm Geomdetail.quasi_tri_case:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ¬ *IN x* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *quasi_tri* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *s* ∧ *HOL_Light_Import.INTER* (*conv0* (*INSERT x* (*INSERT y EMPTY*))) (*conv_trg* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ≠ *EMPTY* ∧ *distance* (*x*, *y*) < *t0*) ⟶ (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *packing s* ∧ ¬ (*v1* = *v2* ∨ *v2* = *v3* ∨ *v3* = *v1*) ∧ *distance* (*v1*, *v2*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v2*, *v3*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v3*, *v1*) < *sqrt* (*real_of_nat* (*8*::*nat*)) ∧ *SUBSET* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *s* ∧ ¬ *IN x* (*INSERT*

$v1$ (INSERT $v2$ (INSERT $v3$ EMPTY))) $\wedge$ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT $v1$ (INSERT $v2$ (INSERT $v3$ EMPTY)))) $\neq$ EMPTY $\wedge$ distance $(x, y) < t0$)

thm Geomdetail.OCT23:

$\forall$ (s::(real, 3) cart $\Rightarrow$ bool) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS s) $\longrightarrow$ packing s $\wedge$ $\neg$ ($v1 = v2 \vee v2 = v3 \vee v3 = v1$) $\wedge$ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s

thm Geomdetail.OCT24:

$\forall$ (s::(real, 3) cart $\Rightarrow$ bool) (x::(real, 3) cart) y::(real, 3) cart. ($\exists$ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. $\neg$ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) $\wedge$ ($\exists$ v4::(real, 3) cart. ($\forall$ (vv1::(real, 3) cart) (vv2::(real, 3) cart) vv3::(real, 3) cart. INSERT vv1 (INSERT vv2 (INSERT vv3 EMPTY)) = INSERT v1 (INSERT v2 (INSERT v3 EMPTY)) $\longrightarrow$ $\neg$ DECIMAL (251::nat) (100::nat) < distance (vv3, vv1)) $\wedge$ IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS s)) $\wedge$ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) $\neq$ EMPTY $\wedge$ distance $(x, y) < t0$) $\longrightarrow$ ($\exists$ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. packing s $\wedge$ $\neg$ ($v1 = v2 \vee v2 = v3 \vee v3 = v1$) $\wedge$ distance (v1, v2) $\leq$ DECIMAL (251::nat) (100::nat) $\wedge$ distance (v2, v3) $\leq$ DECIMAL (251::nat) (100::nat) $\wedge$ distance (v3, v1) < sqrt (real_of_nat (8::nat)) $\wedge$ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s $\wedge$ $\neg$ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) $\wedge$ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) $\neq$ EMPTY $\wedge$ distance $(x, y) < t0$)

thm Geomdetail.quasi_reg_tet_case:

$\forall$ (s::(real, 3) cart $\Rightarrow$ bool) (x::(real, 3) cart) y::(real, 3) cart. ($\exists$ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. ($\exists$ (v4::(real, 3) cart) (vv1::(real, 3) cart) (vv2::(real, 3) cart) vv3::(real, 3) cart. INSERT vv1 (INSERT vv2 (INSERT vv3 EMPTY)) = INSERT v1 (INSERT v2 (INSERT v3 EMPTY)) $\wedge$ DECIMAL (251::nat) (100::nat) < distance (vv3, vv1) $\wedge$ IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS s) $\wedge$ quasi_reg_tet (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) s) $\wedge$ $\neg$ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) $\wedge$ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) $\neq$ EMPTY) $\longrightarrow$ ($\exists$ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. packing s $\wedge$ $\neg$ ($v1 = v2 \vee v2 = v3 \vee v3 = v1$) $\wedge$ distance (v1, v2) $\leq$ DECIMAL (251::nat) (100::nat) $\wedge$ distance (v2, v3) $\leq$ DECIMAL (251::nat) (100::nat) $\wedge$ distance (v3, v1) < sqrt (real_of_nat (8::nat)) $\wedge$ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s $\wedge$ $\neg$ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∧ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ≠ EMPTY)

thm Geomdetail.DIST_PAIR_LEMMA:

INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) EMPTY) = INSERT (?c::(real, ?'a::type) cart) (INSERT (?d::(real, ?'a::type) cart) EMPTY) ⟶ distance (?a, ?b) = distance (?c, ?d)

thm Geomdetail.OTHER_LEMMA:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. (∃ (v::(real, 3) cart) w::(real, 3) cart. IN v (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ IN w (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ real_of_nat (2::nat) ∗ t0 ≤ d3 v w ∧ d3 v w ≤ sqrt (real_of_nat (8::nat)) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ IN y (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ ¬ (x = v ∧ y = w ∨ x = w ∧ y = v) ⟶ d3 x y ≤ real_of_nat (2::nat) ∗ t0)) ∧ (∃ (x::(real, 3) cart) y::(real, 3) cart. IN x (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ IN y (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ real_of_nat (2::nat) ∗ t0 < d3 x y ∧ d3 x y < sqrt (real_of_nat (8::nat))) ∧ real_of_nat (2::nat) ∗ t0 < d3 d b ⟶ real_of_nat (2::nat) ∗ t0 < d3 d b ∧ d3 d b < sqrt (real_of_nat (8::nat)) ∧ (∀ (x::(real, 3) cart) y::(real, 3) cart. IN x (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ IN y (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) ∧ INSERT x (INSERT y EMPTY) ≠ INSERT d (INSERT b EMPTY) ⟶ d3 x y ≤ real_of_nat (2::nat) ∗ t0)

thm Geomdetail.hard_case:

∀ (s::(real, 3) cart ⇒ bool) (x::(real, 3) cart) y::(real, 3) cart. (∃ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. (∃ (v4::(real, 3) cart) (vv1::(real, 3) cart) (vv2::(real, 3) cart) vv3::(real, 3) cart. INSERT vv1 (INSERT vv2 (INSERT vv3 EMPTY)) = INSERT v1 (INSERT v2 (INSERT v3 EMPTY)) ∧ DECIMAL (251::nat) (100::nat) < distance (vv3, vv1) ∧ IN (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) (Q_SYS s) ∧ strict_qua (HOL_Light_Import.UNION (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT v4 EMPTY)) s) ∧ ¬ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∧ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ≠ EMPTY) ⟶ (∃ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. packing s ∧ ¬ (v1 = v2 ∨ v2 = v3 ∨ v3 = v1) ∧ distance (v1, v2) ≤ DECIMAL (251::nat) (100::nat) ∧ distance (v2, v3) ≤ DECIMAL (251::nat) (100::nat) ∧ distance (v3, v1) < sqrt (real_of_nat (8::nat)) ∧ SUBSET (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) s ∧ ¬ IN x (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∧ HOL_Light_Import.INTER (conv0 (INSERT x (INSERT y EMPTY))) (conv_trg (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ≠ EMPTY)

thm Geomdetail.OCTOR23:

(∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ¬ *IN* (*?x*::(*real*, *3*) *cart*) (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ (∃ (*v4*::(*real*, *3*) *cart*) (*vv1*::(*real*, *3*) *cart*) (*vv2*::(*real*, *3*) *cart*) *vv3*::(*real*, *3*) *cart*. *INSERT vv1* (*INSERT vv2* (*INSERT vv3 EMPTY*)) = *INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)) ∧ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *distance* (*vv3*, *vv1*) ∧ *IN* (*HOL_Light_Import*.*UNION* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT v4 EMPTY*)) (*Q_SYS* (*?s*::(*real*, *3*) *cart* ⇒ *bool*))) ∧ *HOL_Light_Import*.*INTER* (*conv0* (*INSERT ?x* (*INSERT* (*?y*::(*real*, *3*) *cart*) *EMPTY*))) (*conv_trg* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ≠ *EMPTY* ∧ *distance* (*?x*, *?y*) < *t0*) ⟶ (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *packing ?s* ∧ ¬ (*v1* = *v2* ∨ *v2* = *v3* ∨ *v3* = *v1*) ∧ *distance* (*v1*, *v2*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v2*, *v3*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v3*, *v1*) < *sqrt* (*real_of_nat* (*8*::*nat*)) ∧ *SUBSET* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *?s* ∧ ¬ *IN ?x* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *HOL_Light_Import*.*INTER* (*conv0* (*INSERT ?x* (*INSERT ?y EMPTY*))) (*conv_trg* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ≠ *EMPTY* ∧ *distance* (*?x*, *?y*) < *t0*)

thm Geomdetail.OCTO23:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ¬ *IN x* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ (*quasi_tri* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *s* ∨ (∃ *v4*::(*real*, *3*) *cart*. *IN* (*HOL_Light_Import*.*UNION* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT v4 EMPTY*)) (*Q_SYS s*))) ∧ *HOL_Light_Import*.*INTER* (*conv0* (*INSERT x* (*INSERT y EMPTY*))) (*conv_trg* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ≠ *EMPTY* ∧ *distance* (*x*, *y*) < *t0*) ⟶ (∃ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *packing s* ∧ ¬ (*v1* = *v2* ∨ *v2* = *v3* ∨ *v3* = *v1*) ∧ *distance* (*v1*, *v2*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v2*, *v3*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *distance* (*v3*, *v1*) < *sqrt* (*real_of_nat* (*8*::*nat*)) ∧ *SUBSET* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *s* ∧ ¬ *IN x* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *HOL_Light_Import*.*INTER* (*conv0* (*INSERT x* (*INSERT y EMPTY*))) (*conv_trg* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ≠ *EMPTY* ∧ *distance* (*x*, *y*) < *t0*)

thm Collect_geom.simp_def2_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) ⟶ *aff_gt* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%59*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%59* (∃ (*ta*::*real*) (*tb*::*real*) *t*::*real*. *ta* + (*tb* + *t*) = (*1*::*real*) ∧ (*0*::*real*) < *t* ∧ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (% *t v0*))) *x*) ∧ *aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%60*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*.

SETSPEC GEN%PVAR%60 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ (0::real) ≤ t ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x)

thm Collect_geom.simp_def2:

(∀(a::(real, ?'d::type) cart) (b::(real, ?'d::type) cart) v0::(real, ?'d::type) cart. DISJOINT (INSERT a (INSERT b EMPTY)) (INSERT v0 EMPTY) ⟶ aff_gt (INSERT a (INSERT b EMPTY)) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%59::(real, ?'d::type) cart. ∃x::(real, ?'d::type) cart. SETSPEC GEN%PVAR%59 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ (0::real) < t ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x) ∧ aff_ge (INSERT a (INSERT b EMPTY)) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%60::(real, ?'d::type) cart. ∃x::(real, ?'d::type) cart. SETSPEC GEN%PVAR%60 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ (0::real) ≤ t ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x)) ∧ (∀(x::(real, ?'c::type) cart) (y::(real, ?'c::type) cart) z::(real, ?'c::type) cart. conv0 (INSERT x (INSERT y (INSERT z EMPTY))) = GSPEC (λGEN%PVAR%61::(real, ?'c::type) cart. ∃t::(real, ?'c::type) cart. SETSPEC GEN%PVAR%61 (∃(a::real) (b::real) c::real. a + (b + c) = (1::real) ∧ (0::real) < a ∧ (0::real) < b ∧ (0::real) < c ∧ t = vector_add (% a x) (vector_add (% b y) (% c z))) t)) ∧ (∀(x::(real, ?'b::type) cart) (y::(real, ?'b::type) cart) z::(real, ?'b::type) cart. hull affine (INSERT x (INSERT y (INSERT z EMPTY))) = GSPEC (λGEN%PVAR%62::(real, ?'b::type) cart. ∃t::(real, ?'b::type) cart. SETSPEC GEN%PVAR%62 (∃(a::real) (b::real) c::real. a + (b + c) = (1::real) ∧ t = vector_add (% a x) (vector_add (% b y) (% c z))) t)) ∧ (∀(v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. DISJOINT (INSERT v2 (INSERT v3 EMPTY)) (INSERT v1 EMPTY) ⟶ aff_lt (INSERT v2 (INSERT v3 EMPTY)) (INSERT v1 EMPTY) = GSPEC (λGEN%PVAR%63::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%63 (∃(t2::real) (t3::real) t1::real. t2 + (t3 + t1) = (1::real) ∧ t1 < (0::real) ∧ x = vector_add (% t2 v2) (vector_add (% t3 v3) (% t1 v1))) x))

thm DEF_cayleyR:

cayleyR = (λ(_2055024::real) (_2055025::real) (_2055026::real) (_2055027::real) (_2055028::real) (_2055029::real) (_2055030::real) (_2055031::real) (_2055032::real) _2055033::real. − (_2055026 ∗ (_2055026 ∗ (_2055028 ∗ _2055028))) + (real_of_nat (2::nat) ∗ (_2055026 ∗ (_2055027 ∗ (_2055028 ∗ _2055028))) − _2055027 ∗ (_2055027 ∗ (_2055028 ∗ _2055028)) + (real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055026 ∗ (_2055028 ∗ _2055029))) − real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055027 ∗ (_2055028 ∗ _2055029))) − real_of_nat (2::nat) ∗ (_2055026 ∗ (_2055027 ∗ (_2055028 ∗ _2055029))) + (real_of_nat (2::nat) ∗ (_2055027 ∗ (_2055027 ∗ (_2055028 ∗ _2055029))) − _2055025 ∗ (_2055025 ∗ (_2055029 ∗ _2055029)) + (real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055027 ∗ (_2055029 ∗ _2055029))) − _2055027 ∗ (_2055027 ∗ (_2055029 ∗ _2055029)) − real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055026 ∗ (_2055028 ∗ _2055030))) + (real_of_nat (2::nat) ∗ (_2055026 ∗ (_2055026 ∗ (_2055028 ∗ _2055030))) + (real_of_nat

$(2::nat) * (\_2055025 * (\_2055027 * (\_2055028 * \_2055030))) - real\_of\_nat$
$(2::nat) * (\_2055026 * (\_2055027 * (\_2055028 * \_2055030))) + (real\_of\_nat$
$(2::nat) * (\_2055025 * (\_2055025 * (\_2055029 * \_2055030))) - real\_of\_nat$
$(2::nat) * (\_2055025 * (\_2055026 * (\_2055029 * \_2055030))) - real\_of\_nat$
$(2::nat) * (\_2055025 * (\_2055027 * (\_2055029 * \_2055030))) + (real\_of\_nat$
$(2::nat) * (\_2055026 * (\_2055027 * (\_2055029 * \_2055030))) - \_2055025 *$
$(\_2055025 * (\_2055030 * \_2055030)) + (real\_of\_nat (2::nat) * (\_2055025 *$
$(\_2055026 * (\_2055030 * \_2055030))) - \_2055026 * (\_2055026 * (\_2055030$
$* \_2055030)) + (real\_of\_nat (2::nat) * (\_2055024 * (\_2055026 * (\_2055028$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055027 * (\_2055028$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055027 * (\_2055028$
$* \_2055031))) + (real\_of\_nat (2::nat) * (\_2055027 * (\_2055027 * (\_2055028$
$* \_2055031))) + (real\_of\_nat (2::nat) * (\_2055024 * (\_2055025 * (\_2055029$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055027 * (\_2055029$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055025 * (\_2055027 * (\_2055029$
$* \_2055031))) + (real\_of\_nat (2::nat) * (\_2055027 * (\_2055027 * (\_2055029$
$* \_2055031))) + (real\_of\_nat (4::nat) * (\_2055027 * (\_2055028 * (\_2055029$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055025 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055026 * (\_2055030$
$* \_2055031))) + (real\_of\_nat (4::nat) * (\_2055025 * (\_2055026 * (\_2055030$
$* \_2055031))) + (real\_of\_nat (4::nat) * (\_2055024 * (\_2055027 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055025 * (\_2055027 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055027 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055028 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055027 * (\_2055028 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055025 * (\_2055029 * (\_2055030$
$* \_2055031))) - real\_of\_nat (2::nat) * (\_2055027 * (\_2055029 * (\_2055030$
$* \_2055031))) + (real\_of\_nat (2::nat) * (\_2055025 * (\_2055030 * (\_2055030$
$* \_2055031))) + (real\_of\_nat (2::nat) * (\_2055026 * (\_2055030 * (\_2055030 *$
$\_2055031))) - \_2055024 * (\_2055024 * (\_2055031 * \_2055031)) + (real\_of\_nat$
$(2::nat) * (\_2055024 * (\_2055027 * (\_2055031 * \_2055031))) - \_2055027 *$
$(\_2055027 * (\_2055031 * \_2055031)) + (real\_of\_nat (2::nat) * (\_2055024 *$
$(\_2055030 * (\_2055031 * \_2055031))) + (real\_of\_nat (2::nat) * (\_2055027 *$
$(\_2055030 * (\_2055031 * \_2055031))) - \_2055030 * (\_2055030 * (\_2055031$
$* \_2055031)) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055026 * (\_2055028$
$* \_2055032))) + (real\_of\_nat (2::nat) * (\_2055026 * (\_2055026 * (\_2055028$
$* \_2055032))) + (real\_of\_nat (2::nat) * (\_2055024 * (\_2055027 * (\_2055028$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055027 * (\_2055028$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055025 * (\_2055029$
$* \_2055032))) + (real\_of\_nat (4::nat) * (\_2055024 * (\_2055026 * (\_2055029$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055025 * (\_2055026 * (\_2055029$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055024 * (\_2055027 * (\_2055029$
$* \_2055032))) + (real\_of\_nat (4::nat) * (\_2055025 * (\_2055027 * (\_2055029$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055027 * (\_2055029$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055026 * (\_2055028 * (\_2055029$
$* \_2055032))) - real\_of\_nat (2::nat) * (\_2055027 * (\_2055028 * (\_2055029$

$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055029 * (\_2055029$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055027 * (\_2055029 * (\_2055029$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055025 * (\_2055030$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055026 * (\_2055030$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055026 * (\_2055030$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055026 * (\_2055030$
$* \_2055032))) + (real\_of\_nat\ (4::nat) * (\_2055026 * (\_2055028 * (\_2055030$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055029 * (\_2055030$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055029 * (\_2055030$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055024 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055026 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055027 * (\_2055031$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055027 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055029 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055027 * (\_2055029 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055030 * (\_2055031$
$* \_2055032))) - real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055030 * (\_2055031$
$* \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055029 * (\_2055030 * (\_2055031 *$
$\_2055032))) - \_2055024 * (\_2055024 * (\_2055032 * \_2055032)) + (real\_of\_nat$
$(2::nat) * (\_2055024 * (\_2055026 * (\_2055032 * \_2055032))) - \_2055026 *$
$(\_2055026 * (\_2055032 * \_2055032)) + (real\_of\_nat\ (2::nat) * (\_2055024 *$
$(\_2055029 * (\_2055032 * \_2055032))) + (real\_of\_nat\ (2::nat) * (\_2055026 *$
$(\_2055029 * (\_2055032 * \_2055032))) - \_2055029 * (\_2055029 * (\_2055032$
$* \_2055032)) + (real\_of\_nat\ (4::nat) * (\_2055024 * (\_2055025 * (\_2055028$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055026 * (\_2055028$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055026 * (\_2055028$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055027 * (\_2055028$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055027 * (\_2055028$
$* \_2055033))) + (real\_of\_nat\ (4::nat) * (\_2055026 * (\_2055027 * (\_2055028$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055028 * (\_2055028$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055027 * (\_2055028 * (\_2055028$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055025 * (\_2055029$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055025 * (\_2055029$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055027 * (\_2055029$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055027 * (\_2055029$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055028 * (\_2055029$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055027 * (\_2055028 * (\_2055029$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055025 * (\_2055030$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055025 * (\_2055030$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055026 * (\_2055030$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055026 * (\_2055030$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055025 * (\_2055028 * (\_2055030$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055026 * (\_2055028 * (\_2055030$
$* \_2055033))) + (real\_of\_nat\ (4::nat) * (\_2055025 * (\_2055029 * (\_2055030$
$* \_2055033))) + (real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055024 * (\_2055031$
$* \_2055033))) - real\_of\_nat\ (2::nat) * (\_2055024 * (\_2055025 * (\_2055031$

∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055027 ∗ (_2055031 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055027 ∗ (_2055031 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055028 ∗ (_2055031 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055027 ∗ (_2055028 ∗ (_2055031 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055030 ∗ (_2055031 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055030 ∗ (_2055031 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055028 ∗ (_2055030 ∗ (_2055031 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055024 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055025 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055026 ∗ (_2055032 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055026 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055028 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055026 ∗ (_2055028 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055029 ∗ (_2055032 ∗ _2055033))) − real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055029 ∗ (_2055032 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055028 ∗ (_2055029 ∗ (_2055032 ∗ _2055033))) + (real_of_nat (4::nat) ∗ (_2055024 ∗ (_2055031 ∗ (_2055032 ∗ _2055033))) − _2055024 ∗ (_2055024 ∗ (_2055033 ∗ _2055033)) + (real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055025 ∗ (_2055033 ∗ _2055033))) − _2055025 ∗ (_2055025 ∗ (_2055033 ∗ _2055033)) + (real_of_nat (2::nat) ∗ (_2055024 ∗ (_2055028 ∗ (_2055033 ∗ _2055033))) + (real_of_nat (2::nat) ∗ (_2055025 ∗ (_2055028 ∗ (_2055033 ∗ _2055033))) − _2055028 ∗ (_2055028 ∗ (_2055033 ∗ _2055033))

thm Collect_geom2.cayleyR:

∀ (x15::real) (x25::real) (x14::real) (x24::real) (x34::real) (x35::real) (x12::real) (x13::real) (x23::real) x45::real. cayleyR x12 x13 x14 x15 x23 x24 x25 x34 x35 x45 = − (x14 ∗ (x14 ∗ (x23 ∗ x23))) + (real_of_nat (2::nat) ∗ (x14 ∗ (x15 ∗ (x23 ∗ x23))) − x15 ∗ (x15 ∗ (x23 ∗ x23)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ (x23 ∗ x24))) − real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ (x23 ∗ x24))) − real_of_nat (2::nat) ∗ (x14 ∗ (x15 ∗ (x23 ∗ x24))) + (real_of_nat (2::nat) ∗ (x15 ∗ (x15 ∗ (x23 ∗ x24))) − x13 ∗ (x13 ∗ (x24 ∗ x24)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ (x24 ∗ x24))) − x15 ∗ (x15 ∗ (x24 ∗ x24)) − real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ (x23 ∗ x25))) + (real_of_nat (2::nat) ∗ (x14 ∗ (x14 ∗ (x23 ∗ x25))) + (real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ (x23 ∗ x25))) − real_of_nat (2::nat) ∗ (x14 ∗ (x15 ∗ (x23 ∗ x25))) + (real_of_nat (2::nat) ∗ (x13 ∗ (x13 ∗ (x24 ∗ x25))) − real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ (x24 ∗ x25))) − real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ (x24 ∗ x25))) + (real_of_nat (2::nat) ∗ (x14 ∗ (x15 ∗ (x24 ∗ x25))) − x13 ∗ (x13 ∗ (x25 ∗ x25)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ (x25 ∗ x25))) − x14 ∗ (x14 ∗ (x25 ∗ x25)) + (real_of_nat (2::nat) ∗ (x12 ∗ (x14 ∗ (x23 ∗ x34))) − real_of_nat (2::nat) ∗ (x12 ∗ (x15 ∗ (x23 ∗ x34))) − real_of_nat (2::nat) ∗ (x14 ∗ (x15 ∗ (x23 ∗ x34))) + (real_of_nat (2::nat) ∗ (x15 ∗ (x15 ∗ (x23 ∗ x34))) + (real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ (x24 ∗ x34))) − real_of_nat (2::nat) ∗ (x12 ∗ (x15 ∗ (x24 ∗ x34))) − real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ (x24 ∗ x34))) + (real_of_nat (2::nat) ∗ (x15 ∗ (x15 ∗ (x24 ∗ x34))) +

(real_of_nat (4::nat) * (x15 * (x23 * (x24 * x34))) − real_of_nat (2::nat) *
(x12 * (x13 * (x25 * x34))) − real_of_nat (2::nat) * (x12 * (x14 * (x25 *
x34))) + (real_of_nat (4::nat) * (x13 * (x14 * (x25 * x34))) + (real_of_nat
(4::nat) * (x12 * (x15 * (x25 * x34))) − real_of_nat (2::nat) * (x13 * (x15
* (x25 * x34))) − real_of_nat (2::nat) * (x14 * (x15 * (x25 * x34))) −
real_of_nat (2::nat) * (x14 * (x23 * (x25 * x34))) − real_of_nat (2::nat)
* (x15 * (x23 * (x25 * x34))) − real_of_nat (2::nat) * (x13 * (x24 * (x25 *
x34))) − real_of_nat (2::nat) * (x15 * (x24 * (x25 * x34))) + (real_of_nat
(2::nat) * (x13 * (x25 * (x25 * x34))) + (real_of_nat (2::nat) * (x14 * (x25
* (x25 * x34))) − x12 * (x12 * (x34 * x34)) + (real_of_nat (2::nat) * (x12
* (x15 * (x34 * x34))) − x15 * (x15 * (x34 * x34)) + (real_of_nat (2::nat)
* (x12 * (x25 * (x34 * x34))) + (real_of_nat (2::nat) * (x15 * (x25 * (x34
* x34))) − x25 * (x25 * (x34 * x34)) − real_of_nat (2::nat) * (x12 * (x14
* (x23 * x35))) + (real_of_nat (2::nat) * (x14 * (x14 * (x23 * x35))) +
(real_of_nat (2::nat) * (x12 * (x15 * (x23 * x35))) − real_of_nat (2::nat) *
(x14 * (x15 * (x23 * x35))) − real_of_nat (2::nat) * (x12 * (x13 * (x24 *
x35))) + (real_of_nat (4::nat) * (x12 * (x14 * (x24 * x35))) − real_of_nat
(2::nat) * (x13 * (x14 * (x24 * x35))) − real_of_nat (2::nat) * (x12 * (x15
* (x24 * x35))) + (real_of_nat (4::nat) * (x13 * (x15 * (x24 * x35))) −
real_of_nat (2::nat) * (x14 * (x15 * (x24 * x35))) − real_of_nat (2::nat) *
(x14 * (x23 * (x24 * x35))) − real_of_nat (2::nat) * (x15 * (x23 * (x24 *
x35))) + (real_of_nat (2::nat) * (x13 * (x24 * (x24 * x35))) + (real_of_nat
(2::nat) * (x15 * (x24 * (x24 * x35))) + (real_of_nat (2::nat) * (x12 * (x13
* (x25 * x35))) − real_of_nat (2::nat) * (x12 * (x14 * (x25 * x35))) −
real_of_nat (2::nat) * (x13 * (x14 * (x25 * x35))) + (real_of_nat (2::nat)
* (x14 * (x14 * (x25 * x35))) + (real_of_nat (4::nat) * (x14 * (x23 * (x25
* x35))) − real_of_nat (2::nat) * (x13 * (x24 * (x25 * x35))) − real_of_nat
(2::nat) * (x14 * (x24 * (x25 * x35))) + (real_of_nat (2::nat) * (x12 * (x12
* (x34 * x35))) − real_of_nat (2::nat) * (x12 * (x14 * (x34 * x35))) −
real_of_nat (2::nat) * (x12 * (x15 * (x34 * x35))) + (real_of_nat (2::nat)
* (x14 * (x15 * (x34 * x35))) − real_of_nat (2::nat) * (x12 * (x24 * (x34
* x35))) − real_of_nat (2::nat) * (x15 * (x24 * (x34 * x35))) − real_of_nat
(2::nat) * (x12 * (x25 * (x34 * x35))) − real_of_nat (2::nat) * (x14 * (x25
* (x34 * x35))) + (real_of_nat (2::nat) * (x24 * (x25 * (x34 * x35))) −
x12 * (x12 * (x35 * x35)) + (real_of_nat (2::nat) * (x12 * (x14 * (x35 *
x35))) − x14 * (x14 * (x35 * x35)) + (real_of_nat (2::nat) * (x12 * (x24 *
(x35 * x35))) + (real_of_nat (2::nat) * (x14 * (x24 * (x35 * x35))) − x24
* (x24 * (x35 * x35)) + (real_of_nat (4::nat) * (x12 * (x13 * (x23 * x45)))
− real_of_nat (2::nat) * (x12 * (x14 * (x23 * x45))) − real_of_nat (2::nat)
* (x13 * (x14 * (x23 * x45))) − real_of_nat (2::nat) * (x12 * (x15 * (x23 *
x45))) − real_of_nat (2::nat) * (x13 * (x15 * (x23 * x45))) + (real_of_nat
(4::nat) * (x14 * (x15 * (x23 * x45))) + (real_of_nat (2::nat) * (x14 * (x23
* (x23 * x45))) + (real_of_nat (2::nat) * (x15 * (x23 * (x23 * x45))) −
real_of_nat (2::nat) * (x12 * (x13 * (x24 * x45))) + (real_of_nat (2::nat) *
(x13 * (x13 * (x24 * x45))) + (real_of_nat (2::nat) * (x12 * (x15 * (x24 *
x45))) − real_of_nat (2::nat) * (x13 * (x15 * (x24 * x45))) − real_of_nat

$(2::nat) * (x13 * (x23 * (x24 * x45))) - real\_of\_nat\ (2::nat) * (x15 * (x23 * (x24 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x13 * (x25 * x45))) + (real\_of\_nat\ (2::nat) * (x13 * (x13 * (x25 * x45))) + (real\_of\_nat\ (2::nat) * (x12 * (x14 * (x25 * x45))) - real\_of\_nat\ (2::nat) * (x13 * (x14 * (x25 * x45))) - real\_of\_nat\ (2::nat) * (x13 * (x23 * (x25 * x45))) - real\_of\_nat\ (2::nat) * (x14 * (x23 * (x25 * x45))) + (real\_of\_nat\ (4::nat) * (x13 * (x24 * (x25 * x45))) + (real\_of\_nat\ (2::nat) * (x12 * (x12 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x13 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x15 * (x34 * x45))) + (real\_of\_nat\ (2::nat) * (x13 * (x15 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x23 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x15 * (x23 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x25 * (x34 * x45))) - real\_of\_nat\ (2::nat) * (x13 * (x25 * (x34 * x45))) + (real\_of\_nat\ (2::nat) * (x23 * (x25 * (x34 * x45))) + (real\_of\_nat\ (2::nat) * (x12 * (x12 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x13 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x14 * (x35 * x45))) + (real\_of\_nat\ (2::nat) * (x13 * (x14 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x23 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x14 * (x23 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x12 * (x24 * (x35 * x45))) - real\_of\_nat\ (2::nat) * (x13 * (x24 * (x35 * x45))) + (real\_of\_nat\ (2::nat) * (x23 * (x24 * (x35 * x45))) + (real\_of\_nat\ (4::nat) * (x12 * (x34 * (x35 * x45))) - x12 * (x12 * (x45 * x45)) + (real\_of\_nat\ (2::nat) * (x12 * (x13 * (x45 * x45))) - x13 * (x13 * (x45 * x45)) + (real\_of\_nat\ (2::nat) * (x12 * (x23 * (x45 * x45))) + (real\_of\_nat\ (2::nat) * (x13 * (x23 * (x45 * x45))) - x23 * (x23 * (x45 * x45))$

thm DEF_muR:

$muR = (\lambda(\_2055182::real)\ (\_2055183::real)\ (\_2055184::real)\ (\_2055185::real)\ (\_2055186::real)\ (\_2055187::real)\ (\_2055188::real)\ (\_2055189::real)\ \_2055190::real.\ cayleyR\ (\_2055187 * \_2055187)\ (\_2055186 * \_2055186)\ (\_2055182 * \_2055182)\ (\_2055188 * \_2055188)\ (\_2055185 * \_2055185)\ (\_2055183 * \_2055183)\ (\_2055189 * \_2055189)\ (\_2055184 * \_2055184)\ (\_2055190 * \_2055190))$

thm DEF_enclosed:

$enclosed = (\lambda(\_2055322::real)\ (\_2055323::real)\ (\_2055324::real)\ (\_2055325::real)\ (\_2055326::real)\ (\_2055327::real)\ (\_2055328::real)\ (\_2055329::real)\ \_2055330::real.\ sqrt\ (quadratic\_root\_plus\ (abc\_of\_quadratic\ (muR\ \_2055322\ \_2055323\ \_2055324\ \_2055325\ \_2055326\ \_2055327\ \_2055328\ \_2055329\ \_2055330))))$

thm Trigonometry.RPFVZDI:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ (y6::real)\ (y7::real)\ (y8::real)\ y9::real.\ enclosed\ y1\ y2\ y3\ y4\ y5\ y6\ y7\ y8\ y9 = sqrt\ (quadratic\_root\_plus\ (abc\_of\_quadratic\ (muR\ y1\ y2\ y3\ y4\ y5\ y6\ y7\ y8\ y9)))$

thm DEF_quad_cross_diag2_x:

$quad\_cross\_diag2\_x = (\lambda(\_2055439::real)\ (\_2055440::real)\ (\_2055441::real)\ (\_2055442::real)\ (\_2055443::real)\ (\_2055444::real)\ (\_2055445::real)\ (\_2055446::real)$

_2055447::real. enclosed (sqrt _2055439) (sqrt _2055443) (sqrt _2055444) (sqrt _2055442) (sqrt _2055440) (sqrt _2055441) (sqrt _2055445) (sqrt _2055446) (sqrt _2055447))

thm Collect_geom.d3:

∀ (v::(real, 3) cart) w::(real, 3) cart. d3 v w = distance (v, w)

thm DEF_rho_ij:

rho_ij = (λ(_2055556::real) (_2055557::real) (_2055558::real) (_2055559::real) (_2055560::real) _2055561::real. − (_2055558 ∗ (_2055558 ∗ (_2055559 ∗ _2055559))) − _2055557 ∗ (_2055557 ∗ (_2055560 ∗ _2055560)) − _2055556 ∗ (_2055556 ∗ (_2055561 ∗ _2055561)) + real_of_nat (2::nat) ∗ (_2055556 ∗ (_2055558 ∗ (_2055559 ∗ _2055561)) + (_2055556 ∗ (_2055557 ∗ (_2055560 ∗ _2055561)) + _2055557 ∗ (_2055558 ∗ (_2055559 ∗ _2055560)))))

thm Collect_geom.rho_ij:

∀ (x12::real) (x34::real) (x13::real) (x14::real) (x23::real) x24::real. rho_ij x12 x13 x14 x23 x24 x34 = − (x14 ∗ (x14 ∗ (x23 ∗ x23))) − x13 ∗ (x13 ∗ (x24 ∗ x24)) − x12 ∗ (x12 ∗ (x34 ∗ x34)) + real_of_nat (2::nat) ∗ (x12 ∗ (x14 ∗ (x23 ∗ x34)) + (x12 ∗ (x13 ∗ (x24 ∗ x34)) + x13 ∗ (x14 ∗ (x23 ∗ x24))))

thm DEF_chi:

chi = (λ(_2055616::real) (_2055617::real) (_2055618::real) (_2055619::real) (_2055620::real) _2055621::real. _2055617 ∗ (_2055619 ∗ _2055620) + (_2055618 ∗ (_2055619 ∗ _2055620) + (_2055616 ∗ (_2055619 ∗ _2055621) + (_2055618 ∗ (_2055619 ∗ _2055621) + (_2055616 ∗ (_2055620 ∗ _2055621) + (_2055617 ∗ (_2055620 ∗ _2055621) − real_of_nat (2::nat) ∗ (_2055619 ∗ (_2055620 ∗ _2055621)) − _2055616 ∗ (_2055621 ∗ _2055621) − _2055618 ∗ (_2055619 ∗ _2055619) − _2055617 ∗ (_2055620 ∗ _2055620)))))))

thm Collect_geom.chi:

∀ (x12::real) (x34::real) (x14::real) (x23::real) (x13::real) x24::real. chi x12 x13 x14 x23 x24 x34 = x13 ∗ (x23 ∗ x24) + (x14 ∗ (x23 ∗ x24) + (x12 ∗ (x23 ∗ x34) + (x14 ∗ (x23 ∗ x34) + (x12 ∗ (x24 ∗ x34) + (x13 ∗ (x24 ∗ x34) − real_of_nat (2::nat) ∗ (x23 ∗ (x24 ∗ x34)) − x12 ∗ (x34 ∗ x34) − x14 ∗ (x23 ∗ x23) − x13 ∗ (x24 ∗ x24))))))

thm DEF_delta:

delta = (λ(_2055676::real) (_2055677::real) (_2055678::real) (_2055679::real) (_2055680::real) _2055681::real. − (_2055676 ∗ (_2055677 ∗ _2055679)) − _2055676 ∗ (_2055678 ∗ _2055680) − _2055677 ∗ (_2055678 ∗ _2055681) − _2055679 ∗ (_2055680 ∗ _2055681) + (_2055676 ∗ (_2055681 ∗ (− _2055676 + (_2055677 + (_2055678 + (_2055679 + (_2055680 − _2055681)))))) + (_2055677 ∗ (_2055680 ∗ (_2055676 − _2055677 + (_2055678 + (_2055679 − _2055680 + _2055681)))) + _2055678 ∗ (_2055679 ∗ (_2055676 + (_2055677 − _2055678 − _2055679 + (_2055680 + _2055681)))))))

thm Collect_geom.delta:

$\forall (x12::real)\ (x13::real)\ (x14::real)\ (x23::real)\ (x24::real)\ x34::real.\ delta\ x12\ x13\ x14\ x23\ x24\ x34 = -\ (x12 * (x13 * x23)) - x12 * (x14 * x24) - x13 * (x14 * x34) - x23 * (x24 * x34) + (x12 * (x34 * (-\ x12 + (x13 + (x14 + (x23 + (x24 - x34)))))) + (x13 * (x24 * (x12 - x13 + (x14 + (x23 - x24 + x34)))) + x14 * (x23 * (x12 + (x13 - x14 - x23 + (x24 + x34))))))$

thm DEF_eta_v:

$eta\_v = (\lambda(\_2055736::(real,\ ?'a::type)\ cart)\ (\_2055737::(real,\ ?'a::type)\ cart)\ \_2055738::(real,\ ?'a::type)\ cart.\ LET\ (\lambda e1::real.\ LET\_END\ (LET\ (\lambda e2::real.\ LET\_END\ (LET\ (\lambda e3::real.\ LET\_END\ (e1 * (e2 * (e3\ /\ sqrt\ (ups\_x\ (e1^2)\ (e2^2)\ (e3^2)))))))\ (distance\ (\_2055737,\ \_2055736))))\ (distance\ (\_2055736,\ \_2055738))))\ (distance\ (\_2055737,\ \_2055738)))$

thm Collect_geom.eta_v:

$\forall (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.\ eta\_v\ v1\ v2\ v3 = LET\ (\lambda e1::real.\ LET\_END\ (LET\ (\lambda e2::real.\ LET\_END\ (LET\ (\lambda e3::real.\ LET\_END\ (e1 * (e2 * (e3\ /\ sqrt\ (ups\_x\ (e1^2)\ (e2^2)\ (e3^2)))))))\ (distance\ (v2,\ v1))))\ (distance\ (v1,\ v3))))\ (distance\ (v2,\ v3))$

thm DEF_max_real:

$max\_real = (\lambda(\_2055757::real)\ \_2055758::real.\ if\ \_2055758 < \_2055757\ then\ \_2055757\ else\ \_2055758)$

thm Collect_geom.max_real:

$\forall (x::real)\ y::real.\ max\_real\ x\ y = (if\ y < x\ then\ x\ else\ y)$

thm DEF_max_real3:

$max\_real3 = (\lambda(\_2055769::real)\ \_2055770::real.\ max\_real\ (max\_real\ \_2055769\ \_2055770))$

thm Collect_geom.max_real3:

$\forall (x::real)\ (y::real)\ z::real.\ max\_real3\ x\ y\ z = max\_real\ (max\_real\ x\ y)\ z$

thm DEF_ups_x_pow2:

$ups\_x\_pow2 = (\lambda(\_2055790::real)\ (\_2055791::real)\ \_2055792::real.\ ups\_x\ (\_2055790 * \_2055790)\ (\_2055791 * \_2055791)\ (\_2055792 * \_2055792))$

thm Collect_geom.ups_x_pow2:

$\forall (x::real)\ (y::real)\ z::real.\ ups\_x\_pow2\ x\ y\ z = ups\_x\ (x * x)\ (y * y)\ (z * z)$

thm DEF_plane_norm:

$plane\_norm = (\lambda\_2055811::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists (n::(real,\ ?'a::type)\ cart)\ v0::(real,\ ?'a::type)\ cart.\ n \neq vec\ (0::nat) \wedge \_2055811 = GSPEC\ (\lambda GEN\%PVAR\%64::(real,\ ?'a::type)\ cart.\ \exists\ v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%64\ (dot\ n\ (vector\_sub\ v\ v0) = (0::real))\ v))$

thm Collect_geom.plane_norm:

$\forall p$::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *plane_norm p* = ($\exists$ (*n*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. *n* $\neq$ *vec* (*0*::*nat*) $\wedge$ *p* = *GSPEC* (λ*GEN%PVAR%64*::(*real*, *?'a*::*type*) *cart*. $\exists$ *v*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%64* (*dot n* (*vector_sub v v0*) = (*0*::*real*)) *v*))

thm DEF_delta_x34:

delta_x34 = (λ(*_2055816*::*real*) (*_2055817*::*real*) (*_2055818*::*real*) (*_2055819*::*real*) (*_2055820*::*real*) *_2055821*::*real*. − *real_of_nat* (*2*::*nat*) ∗ (*_2055816* ∗ *_2055821*) + (− *_2055817* ∗ *_2055818* + (− *_2055819* ∗ *_2055820* + (*_2055817* ∗ *_2055820* + (*_2055818* ∗ *_2055819* + (− *_2055816* ∗ *_2055816* + (*_2055816* ∗ *_2055818* + (*_2055816* ∗ *_2055820* + (*_2055816* ∗ *_2055817* + *_2055816* ∗ *_2055819*)))))))))

thm Collect_geom.delta_x34:

$\forall$ (*x34*::*real*) (*x14*::*real*) (*x24*::*real*) (*x13*::*real*) (*x12*::*real*) *x23*::*real*. *delta_x34 x12 x13 x14 x23 x24 x34* = − *real_of_nat* (*2*::*nat*) ∗ (*x12* ∗ *x34*) + (− *x13* ∗ *x14* + (− *x23* ∗ *x24* + (*x13* ∗ *x24* + (*x14* ∗ *x23* + (− *x12* ∗ *x12* + (*x12* ∗ *x14* + (*x12* ∗ *x24* + (*x12* ∗ *x13* + *x12* ∗ *x23*))))))))

thm DEF_plane_3p:

plane_3p = (λ(*_2055876*::(*real*, *3*) *cart*) (*_2055877*::(*real*, *3*) *cart*) *_2055878*::(*real*, *3*) *cart*. *GSPEC* (λ*GEN%PVAR%65*::(*real*, *3*) *cart*. $\exists$ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%65* ($\neg$ *collinear* (*INSERT _2055876* (*INSERT _2055877* (*INSERT _2055878 EMPTY*))) $\wedge$ ($\exists$ (*ta*::*real*) (*tb*::*real*) *tc*::*real*. *ta* + (*tb* + *tc*) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *ta _2055876*) (*vector_add* (% *tb _2055877*) (% *tc _2055878*)))) *x*))

thm Collect_geom.plane_3p:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) *c*::(*real*, *3*) *cart*. *plane_3p a b c* = *GSPEC* (λ*GEN%PVAR%65*::(*real*, *3*) *cart*. $\exists$ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%65* ($\neg$ *collinear* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) $\wedge$ ($\exists$ (*ta*::*real*) (*tb*::*real*) *tc*::*real*. *ta* + (*tb* + *tc*) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (% *tc c*)))) *x*)

thm DEF_cm3_ups_x:

cm3_ups_x = (λ(*_2055897*::(*real*, *3*) *cart*) (*_2055898*::(*real*, *3*) *cart*) *_2055899*::(*real*, *3*) *cart*. ($ (*vector_sub _2055898 _2055897*) (*2*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*3*::*nat*) − $ (*vector_sub _2055898 _2055897*) (*3*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*2*::*nat*))^2 + (($ (*vector_sub _2055898 _2055897*) (*3*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*1*::*nat*) − $ (*vector_sub _2055898 _2055897*) (*1*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*3*::*nat*))^2 + ($ (*vector_sub _2055898 _2055897*) (*1*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*2*::*nat*) − $ (*vector_sub _2055898 _2055897*) (*2*::*nat*) ∗ $ (*vector_sub _2055899 _2055897*) (*1*::*nat*))^2))

thm Collect_geom.cm3_ups_x:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *cm3_ups_x v1 v2 v3* = ($ (*vector_sub v2 v1*) (*2*::*nat*) $*$ $ (*vector_sub v3 v1*) (*3*::*nat*) $-$ $ (*vector_sub v2 v1*) (*3*::*nat*) $*$ $ (*vector_sub v3 v1*) (*2*::*nat*))^2 + (($ (*vector_sub v2 v1*) (*3*::*nat*) $*$ $ (*vector_sub v3 v1*) (*1*::*nat*) $-$ $ (*vector_sub v2 v1*) (*1*::*nat*) $*$ $ (*vector_sub v3 v1*) (*3*::*nat*))^2 + ($ (*vector_sub v2 v1*) (*1*::*nat*) $*$ $ (*vector_sub v3 v1*) (*2*::*nat*) $-$ $ (*vector_sub v2 v1*) (*2*::*nat*) $*$ $ (*vector_sub v3 v1*) (*1*::*nat*))^2)

thm Collect_geom.lemma_cm3:

$\forall$ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. $ (*vector_sub x y*) (*1*::*nat*) = $ *x* (*1*::*nat*) $-$ $ *y* (*1*::*nat*) $\wedge$ $ (*vector_sub x y*) (*2*::*nat*) = $ *x* (*2*::*nat*) $-$ $ *y* (*2*::*nat*) $\wedge$ $ (*vector_sub x y*) (*3*::*nat*) = $ *x* (*3*::*nat*) $-$ $ *y* (*3*::*nat*)

thm Collect_geom.lemma7:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *cm3_ups_x v1 v2 v3* = *ups_x* ((*vector_norm* (*vector_sub v1 v2*))^2) ((*vector_norm* (*vector_sub v2 v3*))^2) ((*vector_norm* (*vector_sub v3 v1*))^2) / *real_of_nat* (*4*::*nat*)

thm Collect_geom.lemma8:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. (*0*::*real*) $\leq$ *ups_x* ((*vector_norm* (*vector_sub v1 v2*))^2) ((*vector_norm* (*vector_sub v2 v3*))^2) ((*vector_norm* (*vector_sub v3 v1*))^2)

thm Collect_geom.SUB_SUM_SUB:

(*?a*::*real*) $-$ ((*?b*::*real*) + (*?c*::*real*)) = *?a* $-$ *?b* $-$ *?c* $\wedge$ *?a* $-$ (*?b* $-$ *?c*) = *?a* $-$ *?b* + *?c*

thm Collect_geom.JVUNDLC:

$\forall$ (*a*::*real*) (*b*::*real*) (*c*::*real*) *s*::*real*. *s* = (*a* + (*b* + *c*)) / *real_of_nat* (*2*::*nat*) $\longrightarrow$ *real_of_nat* (*16*::*nat*) $*$ (*s* $*$ ((*s* $-$ *a*) $*$ ((*s* $-$ *b*) $*$ (*s* $-$ *c*)))) = *ups_x* (a^2) (b^2) (c^2)

thm Collect_geom.IN_ACT_SING:

$\forall$ (*a*::*?'a*::*type*) *x*::*?'a*::*type*. *INSERT a EMPTY x* = (*a* = *x*) $\wedge$ *IN x* (*INSERT a EMPTY*) = (*x* = *a*) $\wedge$ *INSERT a EMPTY a*

thm Collect_geom.IN_SET2:

$\forall$ (*a*::*?'a*::*type*) (*b*::*?'a*::*type*) *x*::*?'a*::*type*. *IN x* (*INSERT a* (*INSERT b EMPTY*)) = (*x* = *a* $\vee$ *x* = *b*) $\wedge$ *INSERT a* (*INSERT b EMPTY*) *x* = (*x* = *a* $\vee$ *x* = *b*)

thm Collect_geom.SUM_DIS2:

$\forall$ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *f*::*?'a*::*type* $\Rightarrow$ *real*. *x* $\neq$ *y* $\longrightarrow$ *sum* (*INSERT x* (*INSERT y EMPTY*)) *f* = *f x* + *f y*

thm Collect_geom.VSUM_DIS2:

$\forall$ (*x*::*?'b*::*type*) (*y*::*?'b*::*type*) *f*::*?'b*::*type* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*. *x* $\neq$ *y* $\longrightarrow$ *vsum* (*INSERT x* (*INSERT y EMPTY*)) *f* = *vector_add* (*f x*) (*f y*)

thm Collect_geom.NOV10:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *x* = *y* $\longrightarrow$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. (*y* = *x*) = ($\exists$ (*a*::*real*) *b*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *a y*) (% *b y*)))

thm Collect_geom.CONV_SET2:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *conv* (*INSERT x* (*INSERT y EMPTY*)) = *GSPEC* (λ*GEN%PVAR%66*::(*real*, *?'a*::*type*) *cart*. $\exists$ *w*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%66* ($\exists$ (*a*::*real*) *b*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *w* = *vector_add* (% *a x*) (% *b y*)) *w*)

thm Collect_geom.LE_OF_ZPGPXNN:

$\forall$ (*a*::*real*) (*b*::*real*) (*v*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) $\leq$ *a* $\wedge$ (*0*::*real*) $\leq$ *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *v* = *vector_add* (% *a v1*) (% *b v2*) $\longrightarrow$ *distance* (*v*, *v1*) + *distance* (*v*, *v2*) = *distance* (*v1*, *v2*)

thm Collect_geom.LENGTH_EQUA:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. *IN v* (*conv* (*INSERT v1* (*INSERT v2 EMPTY*))) $\longrightarrow$ *distance* (*v*, *v1*) + *distance* (*v*, *v2*) = *distance* (*v1*, *v2*)

thm Collect_geom.ZPGPXNN:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *distance* (*v1*, *v2*) < *distance* (*v*, *v1*) + *distance* (*v*, *v2*) $\longrightarrow$ $\neg$ *IN v* (*conv* (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Collect_geom.REDUCE_T2:

$\forall$ (*P*::*?'b*::*type* $\Rightarrow$ *real* $\Rightarrow$ *?'b*::*type* $\Rightarrow$ *real* $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *real* $\Rightarrow$ *bool*) *Q*::*?'b*::*type* $\Rightarrow$ *?'b*::*type* $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *bool*. ($\forall$ (*v1*::*?'b*::*type*) (*v2*::*?'b*::*type*) (*v3*::*?'a*::*type*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. *P v1 t1 v2 t2 v3 t3* = *P v2 t2 v1 t1 v3 t3*) $\wedge$ ($\forall$ (*v1*::*?'b*::*type*) (*v2*::*?'b*::*type*) *v3*::*?'a*::*type*. *Q v1 v2 v3* = *Q v2 v1 v3*) $\wedge$ ($\forall$ (*v1*::*?'b*::*type*) (*v2*::*?'b*::*type*) (*v3*::*?'a*::*type*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. $\neg$ (*t1* = (*0*::*real*) $\wedge$ *t3* = (*0*::*real*)) $\wedge$ *P v1 t1 v2 t2 v3 t3* $\longrightarrow$ *Q v1 v2 v3*) $\longrightarrow$ ($\forall$ (*v1*::*?'b*::*type*) (*v2*::*?'b*::*type*) (*v3*::*?'a*::*type*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. $\neg$ (*t1* = (*0*::*real*) $\wedge$ *t2* = (*0*::*real*) $\wedge$ *t3* = (*0*::*real*)) $\wedge$ *P v1 t1 v2 t2 v3 t3* $\longrightarrow$ *Q v1 v2 v3*)

thm Collect_geom.VEC_PER2_3_conjunct1:

vector_add (*?a*::(*real*, *?'a*::*type*) *cart*) (*vector_add* (*?b*::(*real*, *?'a*::*type*) *cart*) (*?c*::(*real*, *?'a*::*type*) *cart*)) = *vector_add ?c* (*vector_add ?b ?a*)

thm Collect_geom.VEC_PER2_3_conjunct0:

vector_add (*?a*::(*real*, *?'a*::*type*) *cart*) (*vector_add* (*?b*::(*real*, *?'a*::*type*) *cart*) (*?c*::(*real*, *?'a*::*type*) *cart*)) = *vector_add ?b* (*vector_add ?a ?c*)

thm Collect_geom.VEC_PER2_3:

vector_add (?a::(real, ?'a::type) cart) (vector_add (?b::(real, ?'a::type) cart) (?c::(real, ?'a::type) cart)) = vector_add ?b (vector_add ?a ?c) ∧ vector_add ?a (vector_add ?b ?c) = vector_add ?c (vector_add ?b ?a)

thm Collect_geom.PER2_IN3:

INSERT (?a::?'a::type) (INSERT (?b::?'a::type) (INSERT (?c::?'a::type) EMPTY)) = INSERT ?b (INSERT ?a (INSERT ?c EMPTY)) ∧ INSERT ?a (INSERT ?b (INSERT ?c EMPTY)) = INSERT ?c (INSERT ?b (INSERT ?a EMPTY))

thm Collect_geom.REDUCE_T3:

∀(P::?'c::type ⇒ real ⇒ ?'b::type ⇒ ?'a::type ⇒ ?'c::type ⇒ real ⇒ bool) Q::?'c::type ⇒ ?'b::type ⇒ ?'c::type ⇒ bool. (∀(v1::?'c::type) (v2::?'b::type) (v3::?'c::type) (t1::real) (t2::?'a::type) t3::real. P v1 t1 v2 t2 v3 t3 = P v3 t3 v2 t2 v1 t1) ∧ (∀(v1::?'c::type) (v2::?'b::type) v3::?'c::type. Q v1 v2 v3 = Q v3 v2 v1) ∧ (∀(v1::?'c::type) (v2::?'b::type) (v3::?'c::type) (t1::real) (t2::?'a::type) t3::real. t1 ≠ (0::real) ∧ P v1 t1 v2 t2 v3 t3 ⟶ Q v1 v2 v3) ⟶ (∀(v1::?'c::type) (v2::?'b::type) (v3::?'c::type) (t1::real) (t2::?'a::type) t3::real. ¬ (t1 = (0::real) ∧ t3 = (0::real)) ∧ P v1 t1 v2 t2 v3 t3 ⟶ Q v1 v2 v3)

thm Collect_geom.PAIR_EQ_EXPAND:

(INSERT (?a::?'a::type) (INSERT (?b::?'a::type) EMPTY) = INSERT (?c::?'a::type) (INSERT (?d::?'a::type) EMPTY)) = (?a = ?c ∧ ?b = ?d ∨ ?a = ?d ∧ ?b = ?c)

thm Collect_geom.IN_SET3:

IN (?x::?'a::type) (INSERT (?a::?'a::type) (INSERT (?b::?'a::type) (INSERT (?c::?'a::type) EMPTY))) = (?x = ?a ∨ ?x = ?b ∨ ?x = ?c)

thm Collect_geom.IN_SET4:

IN (?x::?'a::type) (INSERT (?a::?'a::type) (INSERT (?b::?'a::type) (INSERT (?c::?'a::type) (INSERT (?d::?'a::type) EMPTY)))) = (?x = ?a ∨ ?x = ?b ∨ ?x = ?c ∨ ?x = ?d)

thm Collect_geom.PER2_IN3_conjunct1:

INSERT (?a::?'a::type) (INSERT (?b::?'a::type) (INSERT (?c::?'a::type) EMPTY)) = INSERT ?c (INSERT ?b (INSERT ?a EMPTY))

thm Collect_geom.PER2_IN3_conjunct0:

INSERT (?a::?'a::type) (INSERT (?b::?'a::type) (INSERT (?c::?'a::type) EMPTY)) = INSERT ?b (INSERT ?a (INSERT ?c EMPTY))

thm Collect_geom.SGFCDZO:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (t1::real) (t2::real) t3::real. vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) = vec (0::nat)

$\wedge$ *t1* + (*t2* + *t3*) = (*0*::*real*) $\wedge$ $\neg$ (*t1* = (*0*::*real*) $\wedge$ *t2* = (*0*::*real*) $\wedge$ *t3* = (*0*::*real*)) $\longrightarrow$ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))

thm Collect_geom.RHUFIIB:

$\forall$ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *rho_ij x12 x13 x14 x23 x24 x34* * *ups_x x34 x24 x23* = (*chi x12 x13 x14 x23 x24 x34*)^2 + *real_of_nat* (*4*::*nat*) * (*delta x12 x13 x14 x23 x24 x34* * (*x34* * (*x24* * *x23*)))

thm Collect_geom.RIGHT_END_POINT:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*aa*::(*real*, *?'a*::*type*) *cart*) *bb*::(*real*, *?'a*::*type*) *cart*. ($\exists$ (*a*::*real*) *b*::*real*. (*0*::*real*) < *a* $\wedge$ *b* = (*0*::*real*) $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *a aa*) (% *b bb*)) = (*x* = *aa*)

thm Collect_geom.LEFT_END_POINT:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*aa*::(*real*, *?'a*::*type*) *cart*) *bb*::(*real*, *?'a*::*type*) *cart*. ($\exists$ (*a*::*real*) *b*::*real*. *a* = (*0*::*real*) $\wedge$ (*0*::*real*) < *b* $\wedge$ *a* + *b* = (*1*::*real*) $\wedge$ *x* = *vector_add* (% (*0*::*real*) *aa*) (% (*1*::*real*) *bb*)) = (*x* = *bb*)

thm Trigonometry2.REAL_LE_EQ_OR_LT:

((*0*::*real*) $\leq$ (*?a*::*real*)) = (*?a* = (*0*::*real*) $\vee$ (*0*::*real*) < *?a*)

thm Collect_geom.CONV_CONV0:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN x* (*conv* (*INSERT a* (*INSERT b EMPTY*))) = (*x* = *a* $\vee$ *x* = *b* $\vee$ *IN x* (*conv0* (*INSERT a* (*INSERT b EMPTY*))))

thm Collect_geom.MAEWNPU:

$\exists$ (*b*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*) *c*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*. $\forall$ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *delta x12 x13 x14 x23 x24 x34* = $-$ *x12* * *x34*^{2} + (*b x12 x13 x14 x23 x24* * *x34* + *c x12 x13 x14 x23 x24*)

thm DEF_b_coef:

b_coef = (*SOME b*::*nat* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*. $\forall$ *_2063358*::*nat*. $\exists$ *c*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*. $\forall$ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *delta x12 x13 x14 x23 x24 x34* = $-$ *x12* * *x34*^{2} + (*b _2063358 x12 x13 x14 x23 x24* * *x34* + *c x12 x13 x14 x23 x24*)) (*68*::*nat*)

thm DEF_c_coef:

c_coef = (*SOME c*::*nat* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*. $\forall$ (*_2063359*::*nat*) (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *delta x12 x13 x14 x23 x24 x34* = $-$ *x12* * *x34*^{2} + (*b_coef x12 x13 x14 x23 x24* * *x34* + *c _2063359 x12 x13 x14 x23 x24*)) (*69*::*nat*)

thm Collect_geom.DELTA_COEFS:

∀ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *delta x12 x13 x14 x23 x24 x34* = − *x12* ∗ *x34*^{2} + (*b_coef x12 x13 x14 x23 x24* ∗ *x34* + *c_coef x12 x13 x14 x23 x24*)

thm Collect_geom.DELTA_X34:

∀ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *delta x12 x13 x14 x23 x24 x34* = − *x12* ∗ *x34*^{2} + ((− *x13* ∗ *x14* + (− *x23* ∗ *x24* + (*x13* ∗ *x24* + (*x14* ∗ *x23* + (− *x12* ∗ *x12* + (*x12* ∗ *x14* + (*x12* ∗ *x24* + (*x12* ∗ *x13* + *x12* ∗ *x23*)))))))) ∗ *x34* + (*x14* ∗ *x23* ∗ *x12* + (*x14* ∗ *x23* ∗ *x13* + (− *x14* ∗ *x23* ∗ *x14* + (− *x14* ∗ *x23* ∗ *x23* + (*x14* ∗ *x23* ∗ *x24* + (− *x12* ∗ *x13* ∗ *x23* + (− *x12* ∗ *x14* ∗ *x24* + (*x13* ∗ *x24* ∗ *x12* + (− *x13* ∗ *x24* ∗ *x13* + (*x13* ∗ *x24* ∗ *x14* + (*x13* ∗ *x24* ∗ *x23* + − *x13* ∗ *x24* ∗ *x24*))))))))))))

thm Collect_geom.C_COEF_FORMULA:

∀ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) *x24*::*real*. *c_coef x12 x13 x14 x23 x24* = *x14* ∗ *x23* ∗ *x12* + (*x14* ∗ *x23* ∗ *x13* + (− *x14* ∗ *x23* ∗ *x14* + (− *x14* ∗ *x23* ∗ *x23* + (*x14* ∗ *x23* ∗ *x24* + (− *x12* ∗ *x13* ∗ *x23* + (− *x12* ∗ *x14* ∗ *x24* + (*x13* ∗ *x24* ∗ *x12* + (− *x13* ∗ *x24* ∗ *x13* + (*x13* ∗ *x24* ∗ *x14* + (*x13* ∗ *x24* ∗ *x23* + − *x13* ∗ *x24* ∗ *x24*))))))))))

thm Collect_geom2.b_coef:

∀ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) *x24*::*real*. *b_coef x12 x13 x14 x23 x24* = − *x13* ∗ *x14* + (− *x23* ∗ *x24* + (*x13* ∗ *x24* + (*x14* ∗ *x23* + (− *x12* ∗ *x12* + (*x12* ∗ *x14* + (*x12* ∗ *x24* + (*x12* ∗ *x13* + *x12* ∗ *x23*))))))) ∧ *c_coef x12 x13 x14 x23 x24* = *x14* ∗ *x23* ∗ *x12* + (*x14* ∗ *x23* ∗ *x13* + (− *x14* ∗ *x23* ∗ *x14* + (− *x14* ∗ *x23* ∗ *x23* + (*x14* ∗ *x23* ∗ *x24* + (− *x12* ∗ *x13* ∗ *x23* + (− *x12* ∗ *x14* ∗ *x24* + (*x13* ∗ *x24* ∗ *x12* + (− *x13* ∗ *x24* ∗ *x13* + (*x13* ∗ *x24* ∗ *x14* + (*x13* ∗ *x24* ∗ *x23* + − *x13* ∗ *x24* ∗ *x24*))))))))))

thm Collect_geom.AGBWHRD:

∀ (*x12*::*?'e*::*type*) (*x13*::*?'d*::*type*) (*x14*::*?'c*::*type*) (*x23*::*?'b*::*type*) (*x24*::*?'a*::*type*) (*x12a*::*real*) (*x13a*::*real*) (*x14a*::*real*) (*x23a*::*real*) *x24a*::*real*. (*b_coef x12a x13a x14a x23a x24a*)^2 + *real_of_nat* (*4*::*nat*) ∗ (*x12a* ∗ *c_coef x12a x13a x14a x23a x24a*) = *ups_x x12a x23a x13a* ∗ *ups_x x12a x24a x14a*

thm Collect_geom.COLLINEAR_EX:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *collinear* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))) = (∃ (*a*::*real*) (*b*::*real*) *c*::*real*. *a* + (*b* + *c*) = (*0*::*real*) ∧ ¬ (*a* = (*0*::*real*) ∧ *b* = (*0*::*real*) ∧ *c* = (*0*::*real*)) ∧ *vector_add* (% *a x*) (*vector_add* (% *b y*) (% *c z*)) = *vec* (*0*::*nat*))

thm Collect_geom.MAX_REAL_LESS_EX:

∀ (*x*::*real*) (*y*::*real*) *a*::*real*. (*max_real x y* ≤ *a*) = (*x* ≤ *a* ∧ *y* ≤ *a*)

thm Collect_geom.MAX_REAL3_LESS_EX:

∀ (*x*::*real*) (*y*::*real*) (*z*::*real*) *a*::*real*. (*max_real3 x y z* ≤ *a*) = (*x* ≤ *a* ∧ *y* ≤ *a* ∧ *z* ≤ *a*)

thm Collect_geom.PER_MUL3:

$(?a::real) * ((?b::real) * (?c::real)) = ?b * (?a * ?c) \wedge ?a * (?b * ?c) = ?a * (?c * ?b)$

thm Collect_geom.ETA_X_SYM:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge (0::real) \leq z \wedge (0::real) \leq ups\_x\ x\ y\ z \longrightarrow eta\_x\ x\ y\ z = eta\_x\ y\ x\ z \wedge eta\_x\ x\ y\ z = eta\_x\ x\ z\ y$

thm Collect_geom.ETA_Y_SYM:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq ups\_x\ (x * x)\ (y * y)\ (z * z) \longrightarrow eta\_y\ x\ y\ z = eta\_y\ y\ x\ z \wedge eta\_y\ x\ y\ z = eta\_y\ x\ z\ y$

thm Collect_geom.ETA_Y_SYMM:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq ups\_x\ (x * x)\ (y * y)\ (z * z) \longrightarrow eta\_y\ x\ y\ z = eta\_y\ x\ z\ y \wedge eta\_y\ x\ y\ z = eta\_y\ y\ x\ z \wedge eta\_y\ x\ y\ z = eta\_y\ z\ x\ y \wedge eta\_y\ x\ y\ z = eta\_y\ y\ z\ x \wedge eta\_y\ x\ y\ z = eta\_y\ z\ y\ x$

thm Collect_geom.IMPLY_POS:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq ups\_x\ (x * x)\ (y * y)\ (z * z) \longrightarrow (0::real) \leq z * z * (x * x * (y * y)) / ups\_x\ (z * z)\ (x * x)\ (y * y) \wedge (0::real) \leq x * x * (y * y * (z * z)) / ups\_x\ (x * x)\ (y * y)\ (z * z) \wedge (0::real) \leq y * y * (z * z * (x * x)) / ups\_x\ (y * y)\ (z * z)\ (x * x)$

thm Collect_geom.POW2_COND:

$\forall (a::real)\ b::real.\ (0::real) \leq a \wedge (0::real) \leq b \longrightarrow (a \leq b) = (a^2 \leq b^2)$

thm Collect_geom.TRUONGG:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < ups\_x\_pow2\ z\ x\ y \longrightarrow z * z * (x * x * (y * y)) / ups\_x\ (z * z)\ (x * x)\ (y * y) - z^2 / real\_of\_nat\ (4::nat) = z^2 * (z^2 - x^2 - y^2)^2 / (real\_of\_nat\ (4::nat) * ups\_x\_pow2\ z\ x\ y)$

thm Collect_geom.RE_TRUONGG:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < ups\_x\_pow2\ z\ x\ y \longrightarrow z * z * (x * x * (y * y)) / ups\_x\_pow2\ z\ x\ y - z^2 / real\_of\_nat\ (4::nat) = z^2 * (z^2 - x^2 - y^2)^2 / (real\_of\_nat\ (4::nat) * ups\_x\_pow2\ z\ x\ y)$

thm Collect_geom.HVXIKHW:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge (0::real) \leq z \wedge (0::real) < ups\_x\_pow2\ x\ y\ z \longrightarrow max\_real3\ x\ y\ z / real\_of\_nat\ (2::nat) \leq eta\_y\ x\ y\ z$

thm Collect_geom.EXISTS_INV:

$((?a::real) \neq (0::real)) = (?a * ((1::real) / ?a) = (1::real) \wedge (1::real) / ?a * ?a = (1::real))$

thm Collect_geom.REAL_LT_RDIV_0:

$\forall (y::real)\ z::real.\ (0::real) < z \longrightarrow ((0::real) < y\ /\ z) = ((0::real) < y)$

thm Collect_geom.POS_EQ_INV_POS:

$\forall x::real.\ ((0::real) < x) = ((0::real) < (1::real)\ /\ x)$

thm Collect_geom.MIDDLE_POINT:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \longrightarrow IN\ x\ (conv\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \lor IN\ y\ (conv\ (INSERT\ x\ (INSERT\ z\ EMPTY))) \lor IN\ z\ (conv\ (INSERT\ x\ (INSERT\ y\ EMPTY)))$

thm Collect_geom.IN_CONV_COLLINEAR:

$\forall (v::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ v2::(real,\ 3)\ cart.\ IN\ v\ (conv\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \longrightarrow collinear\ (INSERT\ v\ (INSERT\ v1\ (INSERT\ v2\ EMPTY)))$

thm Collect_geom.PER_SET3:

$INSERT\ (?a::?'a::type)\ (INSERT\ (?b::?'a::type)\ (INSERT\ (?c::?'a::type)\ EMPTY)) = INSERT\ ?a\ (INSERT\ ?c\ (INSERT\ ?b\ EMPTY)) \land INSERT\ ?a\ (INSERT\ ?b\ (INSERT\ ?c\ EMPTY)) = INSERT\ ?b\ (INSERT\ ?a\ (INSERT\ ?c\ EMPTY)) \land INSERT\ ?a\ (INSERT\ ?b\ (INSERT\ ?c\ EMPTY)) = INSERT\ ?c\ (INSERT\ ?a\ (INSERT\ ?b\ EMPTY)) \land INSERT\ ?a\ (INSERT\ ?b\ (INSERT\ ?c\ EMPTY)) = INSERT\ ?b\ (INSERT\ ?c\ (INSERT\ ?a\ EMPTY)) \land INSERT\ ?a\ (INSERT\ ?b\ (INSERT\ ?c\ EMPTY)) = INSERT\ ?c\ (INSERT\ ?b\ (INSERT\ ?a\ EMPTY))$

thm Collect_geom.COLLINERA_AS_IN_CONV2:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = (IN\ x\ (conv\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \lor IN\ y\ (conv\ (INSERT\ x\ (INSERT\ z\ EMPTY))) \lor IN\ z\ (conv\ (INSERT\ x\ (INSERT\ y\ EMPTY))))$

thm Collect_geom.LENGTH_EQ_EX:

$\forall (v::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ v2::(real,\ ?'a::type)\ cart.\ (distance\ (v1,\ v) + distance\ (v,\ v2) = distance\ (v1,\ v2)) = (\neg\ distance\ (v1,\ v2) < distance\ (v1,\ v) + distance\ (v,\ v2))$

thm Collect_geom.BETWEEN_IFF_IN_CONVEX_HULL:

$\forall (v::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ v2::(real,\ ?'a::type)\ cart.\ (distance\ (v1,\ v) + distance\ (v,\ v2) = distance\ (v1,\ v2)) = IN\ v\ (hull\ convex\ (INSERT\ v1\ (INSERT\ v2\ EMPTY)))$

thm Collect_geom.BETWEEN_IMP_IN_CONVEX_HULL:

$\forall (v::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ v2::(real,\ ?'a::type)\ cart.\ distance\ (v1,\ v) + distance\ (v,\ v2) = distance\ (v1,\ v2) \longrightarrow (\exists (a::real)\ b::real.$

$(0::real) \leq a \wedge (0::real) \leq b \wedge a + b = (1::real) \wedge v = vector\_add\ (\%\ a\ v1)\ (\%\ b\ v2))$

thm Collect_geom.PRE_HER:

$\forall (x::real)\ (y::real)\ z::real.\ ups\_x\_pow2\ x\ y\ z = real\_of\_nat\ (16::nat) * ((x + (y + z))\ /\ real\_of\_nat\ (2::nat) * (((x + (y + z))\ /\ real\_of\_nat\ (2::nat) - x) * (((x + (y + z))\ /\ real\_of\_nat\ (2::nat) - y) * ((x + (y + z))\ /\ real\_of\_nat\ (2::nat) - z))))$

thm Collect_geom.PRE_HE:

$\forall (x::real)\ (y::real)\ z::real.\ LET\ (\lambda p::real.\ LET\_END\ (ups\_x\_pow2\ x\ y\ z = real\_of\_nat\ (16::nat) * (p * ((p - x) * ((p - y) * (p - z)))))) ((x + (y + z))\ /\ real\_of\_nat\ (2::nat))$

thm Collect_geom.TRIVIVAL_LE:

$\forall (v1::(real, ?'a::type)\ cart)\ (v2::(real, ?'a::type)\ cart)\ v3::(real, ?'a::type)\ cart.\ \neg\ (v2 = v3 \wedge v1 = v2) \longrightarrow distance\ (v1, v2) + (distance\ (v1, v3) + distance\ (v2, v3)) \neq (0::real)$

thm Collect_geom.MID_COND:

$\forall (v::(real, ?'a::type)\ cart)\ (v1::(real, ?'a::type)\ cart)\ v2::(real, ?'a::type)\ cart.\ IN\ v\ (conv\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) = (distance\ (v1, v) + distance\ (v, v2) = distance\ (v1, v2))$

thm Collect_geom.FHFMKIY:

$\forall (v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ (v3::(real, 3)\ cart)\ (x12::real)\ (x13::real)\ x23::real.\ x12 = (distance\ (v1, v2))^2 \wedge x13 = (distance\ (v1, v3))^2 \wedge x23 = (distance\ (v2, v3))^2 \longrightarrow collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) = (ups\_x\ x12\ x13\ x23 = (0::real))$

thm Collect_geom.AFFINE_HULL_FINITE:

$\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow hull\ affine\ s = GSPEC\ (\lambda GEN\%PVAR\%68::(real, ?'a::type)\ cart.\ \exists y::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%68\ (\exists u::(real, ?'a::type)\ cart \Rightarrow real.\ sum\ s\ u = (1::real) \wedge vsum\ s\ (\lambda v::(real, ?'a::type)\ cart.\ \%\ (u\ v)\ v) = y)\ y)$

thm Collect_geom.IN_AFFINE_HULL_IMP_COLLINEAR:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ c::(real, ?'a::type)\ cart.\ IN\ a\ (hull\ affine\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \longrightarrow collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))$

thm Collect_geom.FAFKVLR_ALT:

$\forall (v1::(real, ?'a::type)\ cart)\ (v2::(real, ?'a::type)\ cart)\ (v3::(real, ?'a::type)\ cart)\ v::(real, ?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) \wedge IN\ v\ (hull\ affine\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow Ex1\ (GABS\ (\lambda f::real \times real \times real \Rightarrow bool.\ \forall (t1::real)\ (t2::real)$

t3::real. GEQ (f (t1, t2, t3)) (v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real))))

thm Collect_geom.equivalent_lemma:

(∃ (t1::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real) (t2::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real) t3::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (t1 v1 v2 v3 v) v1) (vector_add (% (t2 v1 v2 v3 v) v2) (% (t3 v1 v2 v3 v) v3)) ∧ t1 v1 v2 v3 v + (t2 v1 v2 v3 v + t3 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 v1 v2 v3 v ∧ tb = t2 v1 v2 v3 v ∧ tc = t3 v1 v2 v3 v)) = (∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ (∃ (t1::real) (t2::real) t3::real. v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 ∧ tb = t2 ∧ tc = t3)))

thm Collect_geom.LAMBDA_TRIPLED_THM:

∀ t::?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type. GABS (λf::?'d::type × ?'c::type × ?'b::type ⇒ ?'a::type. ∀ (x::?'d::type) (y::?'c::type) z::?'b::type. GEQ (f (x, y, z)) (t x y z)) = (λp::?'d::type × ?'c::type × ?'b::type. t (fst p) (fst (snd p)) (snd (snd p)))

thm Collect_geom.FORALL_TRIPLED_THM:

∀ P::?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. All (GABS (λf::?'c::type × ?'b::type × ?'a::type ⇒ bool. ∀ (x::?'c::type) (y::?'b::type) z::?'a::type. GEQ (f (x, y, z)) (P x y z))) = (∀ (x::?'c::type) (y::?'b::type) z::?'a::type. P x y z)

thm Collect_geom.EXISTS_TRIPLED_THM:

∀ P::?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. Ex (GABS (λf::?'c::type × ?'b::type × ?'a::type ⇒ bool. ∀ (x::?'c::type) (y::?'b::type) z::?'a::type. GEQ (f (x, y, z)) (P x y z))) = (∃ (x::?'c::type) (y::?'b::type) z::?'a::type. P x y z)

thm Collect_geom.EXISTS_UNIQUE_TRIPLED_THM:

∀ P::?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ bool. Ex1 (GABS (λf::?'c::type × ?'b::type × ?'a::type ⇒ bool. ∀ (x::?'c::type) (y::?'b::type) z::?'a::type. GEQ (f (x, y, z)) (P x y z))) = (∃ (x::?'c::type) (y::?'b::type) z::?'a::type. P x y z ∧ (∀ (x'::?'c::type) (y'::?'b::type) z'::?'a::type. P x' y' z' ⟶ x' = x ∧ y' = y ∧ z' = z))

thm Collect_geom.theoremmm:

(∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ (∃ (t1::real) (t2::real) t3::real. v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 ∧ tb = t2 ∧ tc = t3))) = (∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ∧ IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ⟶ Ex1 (GABS (λf::real × real × real ⇒ bool. ∀ (t1::real) (t2::real) t3::real. GEQ (f (t1, t2, t3)) (v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real)))))

thm Collect_geom.FAFKVLR:

∃ (t1::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real) (t2::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real) t3::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (t1 v1 v2 v3 v) v1) (vector_add (% (t2 v1 v2 v3 v) v2) (% (t3 v1 v2 v3 v) v3)) ∧ t1 v1 v2 v3 v + (t2 v1 v2 v3 v + t3 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 v1 v2 v3 v ∧ tb = t2 v1 v2 v3 v ∧ tc = t3 v1 v2 v3 v)

thm Collect_geom.lemma11:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ (∃ (t1::real) (t2::real) t3::real. v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 ∧ tb = t2 ∧ tc = t3))

thm DEF_coef1:

coef1 = (SOME t1::nat ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ _2072798::nat. ∃ (t2::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real) t3::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull *affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (t1 _2072798 v1 v2 v3 v) v1) (vector_add (% (t2 v1 v2 v3 v) v2) (% (t3 v1 v2 v3 v) v3)) ∧ t1 _2072798 v1 v2 v3 v + (t2 v1 v2 v3 v + t3 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 _2072798 v1 v2 v3 v ∧ tb = t2 v1 v2 v3 v ∧ tc = t3 v1 v2 v3 v)) (70::nat)*

thm DEF_coef2:

coef2 = (SOME t2::nat ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ _2072799::nat. ∃ t3::(real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (coef1 v1 v2 v3 v) v1) (vector_add (% (t2 _2072799 v1 v2 v3 v) v2) (% (t3 v1 v2 v3 v) v3)) ∧ coef1 v1 v2 v3 v + (t2 _2072799 v1 v2 v3 v + t3 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = coef1 v1 v2 v3 v ∧ tb = t2 _2072799 v1 v2 v3 v ∧ tc = t3 v1 v2 v3 v)) (71::nat)

thm DEF_coef3:

coef3 = (SOME t3::nat ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ (real, ?'a::type) cart ⇒ real. ∀ (_2072800::nat) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (coef1 v1 v2 v3 v) v1) (vector_add (% (coef2 v1 v2 v3 v) v2) (% (t3 _2072800 v1 v2 v3 v) v3)) ∧ coef1 v1 v2 v3 v + (coef2 v1 v2 v3 v + t3 _2072800 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = coef1 v1 v2 v3 v ∧ tb = coef2 v1 v2 v3 v ∧ tc = t3 _2072800 v1 v2 v3 v)) (72::nat)

thm Collect_geom.COEFS:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. IN v (hull affine (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ v = vector_add (% (coef1 v1 v2 v3 v) v1) (vector_add (% (coef2 v1 v2 v3 v) v2) (% (coef3 v1 v2 v3 v) v3)) ∧ coef1 v1 v2 v3 v + (coef2 v1 v2 v3 v + coef3 v1 v2 v3 v) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = coef1 v1 v2 v3 v ∧ tb = coef2 v1 v2 v3 v ∧ tc = coef3 v1 v2 v3 v)

thm Collect_geom.simp_def2_conjunct3:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. DISJOINT (INSERT v2 (INSERT v3 EMPTY)) (INSERT v1 EMPTY) ⟶ aff_lt (INSERT v2 (INSERT v3 EMPTY)) (INSERT v1 EMPTY) = GSPEC (λGEN%PVAR%63::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%63 (∃ (t2::real) (t3::real) t1::real. t2 + (t3 + t1) = (1::real) ∧ t1 < (0::real) ∧ x = vector_add (% t2 v2) (vector_add (% t3 v3) (% t1 v1))) x)

thm Collect_geom.simp_def2_conjunct2:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. hull affine (INSERT x (INSERT y (INSERT z EMPTY))) = GSPEC (λGEN%PVAR%62::(real, ?'a::type) cart. ∃ t::(real, ?'a::type) cart. SETSPEC GEN%PVAR%62 (∃ (a::real) (b::real) c::real. a + (b + c) = (1::real) ∧ t = vector_add (% a x) (vector_add (% b y) (% c z))) t)

thm Collect_geom.simp_def2_conjunct1:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. conv0 (INSERT x (INSERT y (INSERT z EMPTY))) = GSPEC (λGEN%PVAR%61::(real, ?'a::type) cart. ∃ t::(real, ?'a::type) cart. SETSPEC GEN%PVAR%61 (∃ (a::real) (b::real) c::real. a + (b + c) = (1::real) ∧ (0::real) < a ∧ (0::real) < b ∧ (0::real) < c ∧ t = vector_add (% a x) (vector_add (% b y) (% c z))) t)

thm Collect_geom.lem11:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v::(real, ?'a::type) cart. (∃ (a::real) (b::real) c::real. a + (b + c) = (1::real) ∧ v = vector_add (% a v1) (vector_add (% b v2) (% c v3))) ∧ ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ (∃ (t1::real) (t2::real) t3::real. v = vector_add (% t1 v1) (vector_add (% t2 v2) (% t3 v3)) ∧ t1 + (t2 + t3) = (1::real) ∧ (∀ (ta::real) (tb::real) tc::real. v = vector_add (% ta v1) (vector_add (% tb v2) (% tc v3)) ∧ ta + (tb + tc) = (1::real) ⟶ ta = t1 ∧ tb = t2 ∧ tc = t3))

thm Collect_geom.REAL_PER3:

∀ (a::real) (b::real) c::real. a + (b + c) = b + (a + c) ∧ a + (b + c) = c + (b + a)

thm Collect_geom.NOT_COLLINEAR_IMP_2_UNEQUAL:

¬ collinear (INSERT (?v0.0::(real, ?'a::type) cart) (INSERT (?va::(real, ?'a::type) cart) (INSERT (?vb::(real, ?'a::type) cart) EMPTY))) ⟶ ?v0.0 ≠ ?va ∧ ?v0.0 ≠ ?vb

thm Collect_geom.COLLINEAR_DISJOINT3:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. ¬ collinear (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ DISJOINT (INSERT v2 (INSERT v3 EMPTY)) (INSERT v1 EMPTY)

thm Collect_geom.COLLINEAR_DISJOINT_PERM3:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v3*::(*real*, *?'a*::*type*) *cart*. (¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *DISJOINT* (*INSERT v1* (*INSERT v2 EMPTY*)) (*INSERT v3 EMPTY*)) ∧ (¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *DISJOINT* (*INSERT v2* (*INSERT v3 EMPTY*)) (*INSERT v1 EMPTY*)) ∧ (¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *DISJOINT* (*INSERT v3* (*INSERT v1 EMPTY*)) (*INSERT v2 EMPTY*))

thm Collect_geom.simp_def_ge:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) ⟶ *aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%69*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%69* (∃ (*ta*::*real*) (*tb*::*real*) *t*::*real*. *ta* + (*tb* + *t*) = (*1*::*real*) ∧ (*0*::*real*) ≤ *t* ∧ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (% *t v0*))) *x*)

thm Collect_geom.IN_CONV3_EQ:

∀ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *IN v* (*conv* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) = (*IN v* (*aff_ge* (*INSERT v1* (*INSERT v2 EMPTY*)) (*INSERT v3 EMPTY*)) ∧ *IN v* (*aff_ge* (*INSERT v2* (*INSERT v3 EMPTY*)) (*INSERT v1 EMPTY*)) ∧ *IN v* (*aff_ge* (*INSERT v3* (*INSERT v1 EMPTY*)) (*INSERT v2 EMPTY*)))

thm Collect_geom.IN_CONV03_EQ:

∀ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *IN v* (*conv0* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) = (*IN v* (*aff_gt* (*INSERT v1* (*INSERT v2 EMPTY*)) (*INSERT v3 EMPTY*)) ∧ *IN v* (*aff_gt* (*INSERT v2* (*INSERT v3 EMPTY*)) (*INSERT v1 EMPTY*)) ∧ *IN v* (*aff_gt* (*INSERT v3* (*INSERT v1 EMPTY*)) (*INSERT v2 EMPTY*)))

thm Collect_geom.AFFINE_SET_GEN_BY_TWO_POINTS:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *affine* (*GSPEC* (λ*GEN%PVAR%70*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%70* (∃ (*ta*::*real*) *tb*::*real*. *ta* + *tb* = (*1*::*real*) ∧ *x* = *vector_add* (% *ta a*) (% *tb b*)) *x*))

thm Collect_geom.SET2_SU_EX:

SUBSET (*INSERT* (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) *EMPTY*)) (*?s*::*?'a*::*type* ⇒ *bool*) = (*IN ?a ?s* ∧ *IN ?b ?s*)

thm Collect_geom.GENERATING_POINT_IN_SET_AFF:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*INSERT a* (*INSERT b EMPTY*)) (*GSPEC* (λ*GEN%PVAR%71*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%71* (∃ (*ta*::*real*) *tb*::*real*. *ta* + *tb* = (*1*::*real*) ∧ *x* = *vector_add* (% *ta a*) (% *tb b*)) *x*))

thm Collect_geom.AFF_2POINTS_INTERPRET:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ aff\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%72::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%72\ (\exists (ta::real)\ tb::real.\ ta + tb = (1::real) \wedge x = vector\_add\ (\%\ ta\ a)\ (\%\ tb\ b))\ x)$

thm Collect_geom.IN_AFF_GE_INTERPRET_TO_AFF_GT_AND_AFF:

$\forall (v::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))\ (INSERT\ v3\ EMPTY) \longrightarrow IN\ v\ (aff\_ge\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))\ (INSERT\ v3\ EMPTY)) = (IN\ v\ (aff\_gt\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))\ (INSERT\ v3\ EMPTY)) \vee IN\ v\ (aff\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))))$

thm Collect_geom.AFFINE_AFF_HULL:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ (aff\ s)$

thm Collect_geom.AFFINE_CONTAIN_LINE:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ affine\ s \wedge SUBSET\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ s \longrightarrow SUBSET\ (aff\ (INSERT\ a\ (INSERT\ b\ EMPTY)))\ s$

thm Collect_geom.VECTOR_SUB_DISTRIBUTE:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ vector\_sub\ (\%\ a\ x)\ (\%\ a\ y) = \%\ a\ (vector\_sub\ x\ y)$

thm Collect_geom.CHANGE_SIDE:

$(?a::real) \neq (0::real) \longrightarrow ((?x::(real,\ ?'a::type)\ cart) = \%\ ?a\ (?y::(real,\ ?'a::type)\ cart)) = (\%\ ((1::real)\ /\ ?a)\ ?x = ?y)$

thm Collect_geom.PRE_INVERSE_SUB:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ SUBSET\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ (aff\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge a \neq b \longrightarrow SUBSET\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ (aff\ (INSERT\ a\ (INSERT\ b\ EMPTY)))$

thm Collect_geom.LEMMA5:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ x::(real,\ ?'a::type)\ cart \Rightarrow bool.\ line\ x \wedge SUBSET\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ x \wedge a \neq b \longrightarrow x = aff\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm Trigonometry2.COL_EQ_UPS_0:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) = (ups\_x\ ((distance\ (v1,\ v2))^2)\ ((distance\ (v1,\ v3))^2)\ ((distance\ (v2,\ v3))^2) = (0::real))$

thm Collect_geom.EQ_POW2_COND:

$\forall (a::real)\ b::real.\ (0::real) \leq a \wedge (0::real) \leq b \longrightarrow (a = b) = (a^2 = b^2)$

thm Collect_geom.D3_POS_LE:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (0::real) \leq d3\ x\ y$

thm DEF_delta_x12:

$delta\_x12 = (\lambda(\_2076957::real)\ (\_2076958::real)\ (\_2076959::real)\ (\_2076960::real)\ (\_2076961::real)\ \_2076962::real.\ -\ \_2076958 * \_2076960 + (-\ \_2076959 * \_2076961 + (\_2076962 * (-\ \_2076957 + (\_2076958 + (\_2076959 + (\_2076960 + (\_2076961 + -\ \_2076962))))) + (-\ \_2076957 * \_2076962 + (\_2076958 * \_2076961 + \_2076959 * \_2076960)))))$

thm Collect_geom.delta_x12:

$\forall (x12::real)\ (x34::real)\ (x13::real)\ (x24::real)\ (x14::real)\ x23::real.\ delta\_x12\ x12\ x13\ x14\ x23\ x24\ x34 = -\ x13 * x23 + (-\ x14 * x24 + (x34 * (-\ x12 + (x13 + (x14 + (x23 + (x24 + -\ x34))))) + (-\ x12 * x34 + (x13 * x24 + x14 * x23))))$

thm DEF_delta_x13:

$delta\_x13 = (\lambda(\_2077017::real)\ (\_2077018::real)\ (\_2077019::real)\ (\_2077020::real)\ (\_2077021::real)\ \_2077022::real.\ -\ \_2077017 * \_2077020 + (-\ \_2077019 * \_2077022 + (\_2077017 * \_2077022 + (\_2077021 * (\_2077017 + (-\ \_2077018 + (\_2077019 + (\_2077020 + (-\ \_2077021 + \_2077022))))) + (-\ \_2077018 * \_2077021 + \_2077019 * \_2077020)))))$

thm Collect_geom.delta_x13:

$\forall (x12::real)\ (x34::real)\ (x13::real)\ (x24::real)\ (x14::real)\ x23::real.\ delta\_x13\ x12\ x13\ x14\ x23\ x24\ x34 = -\ x12 * x23 + (-\ x14 * x34 + (x12 * x34 + (x24 * (x12 + (-\ x13 + (x14 + (x23 + (-\ x24 + x34))))) + (-\ x13 * x24 + x14 * x23))))$

thm DEF_delta_x14:

$delta\_x14 = (\lambda(\_2077077::real)\ (\_2077078::real)\ (\_2077079::real)\ (\_2077080::real)\ (\_2077081::real)\ \_2077082::real.\ -\ \_2077077 * \_2077081 + (-\ \_2077078 * \_2077082 + (\_2077077 * \_2077082 + (\_2077078 * \_2077081 + (\_2077080 * (\_2077077 + (\_2077078 + (-\ \_2077079 + (-\ \_2077080 + (\_2077081 + \_2077082))))) + -\ \_2077079 * \_2077080)))))$

thm Collect_geom.delta_x14:

$\forall (x12::real)\ (x13::real)\ (x24::real)\ (x34::real)\ (x14::real)\ x23::real.\ delta\_x14\ x12\ x13\ x14\ x23\ x24\ x34 = -\ x12 * x24 + (-\ x13 * x34 + (x12 * x34 + (x13 * x24 + (x23 * (x12 + (x13 + (-\ x14 + (-\ x23 + (x24 + x34))))) + -\ x14 * x23))))$

thm Collect_geom2.DIST_POW2_DOT:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (distance\ (a,\ b))^2 = dot\ (vector\_sub\ a\ b)\ (vector\_sub\ a\ b)$

thm Collect_geom.TO_UYCH:

$(0::real) < ups\_x\ (?a12.0::real)\ (?a13.0::real)\ (?a23.0::real) \longrightarrow delta\_x12\ (?a01.0::real)\ (?a02.0::real)\ (?a03.0::real)\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 + (delta\_x13\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 + delta\_x14\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0) = (1::real)$

thm Collect_geom.NOT_UPS_X_ZERO_IMP_SMT:

$ups\_x\ (?a12.0::real)\ (?a13.0::real)\ (?a23.0::real) \neq (0::real) \longrightarrow (delta\_x13\ (?a01.0::real)\ (?a02.0::real)\ (?a03.0::real)\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a12.0 + (delta\_x13\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (delta\_x14\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (?a12.0 + (?a13.0 - ?a23.0))) + (delta\_x14\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a13.0) = ?a01.0 - delta\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 \wedge (delta\_x14\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a23.0 + (delta\_x14\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (delta\_x12\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (?a23.0 + (?a12.0 - ?a13.0))) + (delta\_x12\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a12.0) = ?a02.0 - delta\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 \wedge (delta\_x12\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a13.0 + (delta\_x12\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (delta\_x13\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0 * (?a13.0 + (?a23.0 - ?a12.0))) + (delta\_x13\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0)^2 * ?a23.0) = ?a03.0 - delta\ ?a01.0\ ?a02.0\ ?a03.0\ ?a12.0\ ?a13.0\ ?a23.0\ /\ ups\_x\ ?a12.0\ ?a13.0\ ?a23.0$

thm Collect_geom.TROI_OI_DAT_HOI:

$(0::real) \leq ups\_x\ ((distance\ (?v1.0::(real,\ 3)\ cart,\ ?v2.0::(real,\ 3)\ cart))^2)\ ((distance\ (?v2.0,\ ?v3.0::(real,\ 3)\ cart))^2)\ ((distance\ (?v1.0,\ ?v3.0))^2)$

thm Collect_geom.ZERO_LE_UPS_X:

$(0::real) \leq ups\_x\ ((d3\ (?x::(real,\ 3)\ cart)\ (?y::(real,\ 3)\ cart))^2)\ ((d3\ ?x\ (?z::(real,\ 3)\ cart))^2)\ ((d3\ ?y\ ?z)^2)$

thm Collect_geom.NORM_POW2_SUM2:

$(vector\_norm\ (vector\_add\ (\%\ (?a::real)\ (?x::(real,\ ?'a::type)\ cart))\ (\%\ (?b::real)\ (?y::(real,\ ?'a::type)\ cart))))^2 = ?a^2 * (vector\_norm\ ?x)^2 + (real\_of\_nat\ (2::nat) * (?a * ?b * dot\ ?x\ ?y) + ?b^2 * (vector\_norm\ ?y)^2)$

thm Collect_geom.X_DOT_X_EQ:

$dot\ (?x::(real,\ ?'a::type)\ cart)\ ?x = (vector\_norm\ ?x)^2$

thm Collect_geom.SUB_DIST_POW2_INTERPRETE:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ c::real.\ ((distance\ (x,\ v))^2 - (distance\ (y,\ v))^2 = c) = (dot\ (vector\_sub\ (\%\ (real\_of\_nat\ (2::nat))\ v)\ (vector\_add\ x\ y))\ (vector\_sub\ y\ x) = c)$

thm Collect_geom.REAL_DIV_LZERO:

$\forall x::real.\ (0::real)\ /\ x = (0::real)$

thm Collect_geom.HALF_OF_LE16:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (a01::real)\ (a02::real)\ a03::real.\ \neg\ collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) \wedge LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (delta\ a01\ a02\ a03\ x12\ x13\ x23 = (0::real)))\ ((d3\ v2\ v3)^2)))\ ((d3\ v1\ v3)^2)))\ ((d3\ v1\ v2)^2) \longrightarrow (\exists v0::(real,\ 3)\ cart.\ IN\ v0\ (aff\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \wedge a01 = (d3\ v0\ v1)^2 \wedge a02 = (d3\ v0\ v2)^2 \wedge a03 = (d3\ v0\ v3)^2 \wedge LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda vv::real.\ LET\_END\ (LET\ (\lambda t1::real.\ LET\_END\ (LET\ (\lambda t2::real.\ LET\_END\ (LET\ (\lambda t3::real.\ LET\_END\ (v0 = vector\_add\ (\%\ t1\ v1)\ (vector\_add\ (\%\ t2\ v2)\ (\%\ t3\ v3))))\ (delta\_x14\ a01\ a02\ a03\ x12\ x13\ x23\ /\ vv)))\ (delta\_x13\ a01\ a02\ a03\ x12\ x13\ x23\ /\ vv)))\ (delta\_x12\ a01\ a02\ a03\ x12\ x13\ x23\ /\ vv)))\ (ups\_x\ x12\ x13\ x23)))\ ((d3\ v2\ v3)^2)))\ ((d3\ v1\ v3)^2)))\ ((d3\ v1\ v2)^2))$

thm Collect_geom.EQ_SUB_DIST_POW2_IMP_IDENTIFIED:

$\forall (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ (v3::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ SUBSET\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ (hull\ affine\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \wedge (distance\ (u,\ v2))^2 - (distance\ (u,\ v1))^2 = (distance\ (w,\ v2))^2 - (distance\ (w,\ v1))^2 \wedge (distance\ (u,\ v3))^2 - (distance\ (u,\ v1))^2 = (distance\ (w,\ v3))^2 - (distance\ (w,\ v1))^2 \longrightarrow w = u$

thm Collect_geom.SDIHJZK:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (a01::real)\ (a02::real)\ a03::real.\ \neg\ collinear\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) \wedge LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (delta\ a01\ a02\ a03\ x12\ x13\ x23 = (0::real)))\ ((d3\ v2\ v3)^2)))\ ((d3\ v1\ v3)^2)))\ ((d3\ v1\ v2)^2) \longrightarrow (\exists v0::(real,\ 3)\ cart.\ IN\ v0\ (aff\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \wedge a01 = (d3\ v0\ v1)^2 \wedge a02 = (d3\ v0\ v2)^2 \wedge a03 = (d3\ v0\ v3)^2 \wedge (\forall vv0::(real,\ 3)\ cart.\ IN\ vv0\ (aff\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \wedge a01 = (d3\ vv0\ v1)^2 \wedge a02 = (d3\ vv0\ v2)^2 \wedge a03 = (d3\ vv0\ v3)^2 \longrightarrow vv0 = v0) \wedge LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda vv::real.\ LET\_END\ (LET\ (\lambda t1::real.\ LET\_END\ (LET\ (\lambda t2::real.\ LET\_END\ (LET\ (\lambda t3::real.\ LET\_END\ (v0 = vector\_add\ (\%\ t1\ v1)\ (vector\_add\ (\%\ t2\ v2)\ (\%\ t3\ v3))))\ (delta\_x14\ a01\ a02\ a03\ x12\ x13\ x23\ /\ vv)))\ (delta\_x13\ a01\ a02\ a03$

x12 x13 x23 / vv))) (*delta_x12 a01 a02 a03 x12 x13 x23 / vv*))) (*ups_x x12 x13 x23*))) ((*d3 v2 v3*)^2))) ((*d3 v1 v3*)^2))) ((*d3 v1 v2*)^2))

thm Collect_geom.SDIHJZK_INTERPRETE:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*a01*::*real*) (*a02*::*real*) *a03*::*real*. ¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *delta a01 a02 a03* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2) = (*0*::*real*) ⟶ (∃ *v0*::(*real*, *3*) *cart*. *IN v0* (*aff* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ∧ *a01* = (*d3 v0 v1*)^2 ∧ *a02* = (*d3 v0 v2*)^2 ∧ *a03* = (*d3 v0 v3*)^2 ∧ (∀ *vv0*::(*real*, *3*) *cart*. *IN vv0* (*aff* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ∧ *a01* = (*d3 vv0 v1*)^2 ∧ *a02* = (*d3 vv0 v2*)^2 ∧ *a03* = (*d3 vv0 v3*)^2 ⟶ *vv0* = *v0*) ∧ *v0* = *vector_add* (% (*delta_x12 a01 a02 a03* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2) / *ups_x* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2)) *v1*) (*vector_add* (% (*delta_x13 a01 a02 a03* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2) / *ups_x* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2)) *v2*) (% (*delta_x14 a01 a02 a03* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2) / *ups_x* ((*d3 v1 v2*)^2) ((*d3 v1 v3*)^2) ((*d3 v2 v3*)^2)) *v3*)))

thm Collect_geom.DELTA_RRR_INTERPRETE:

delta (*?r*::*real*) *?r ?r* (*?a*::*real*) (*?b*::*real*) (*?c*::*real*) = − *?a* ∗ (*?b* ∗ *?c*) + *?r* ∗ *ups_x ?a ?b ?c*

thm Collect_geom.NOT_UPS_X_EQ_0_IMP:

ups_x (*?a*::*real*) (*?b*::*real*) (*?c*::*real*) ≠ (*0*::*real*) ⟶ *delta* (*?a* ∗ (*?b* ∗ *?c*) / *ups_x ?a ?b ?c*) (*?a* ∗ (*?b* ∗ *?c*) / *ups_x ?a ?b ?c*) (*?a* ∗ (*?b* ∗ *?c*) / *ups_x ?a ?b ?c*) *?a ?b ?c* = (*0*::*real*)

thm Collect_geom.PROVE_EXISTS_RADV:

∀ (*va*::(*real*, *3*) *cart*) (*vb*::(*real*, *3*) *cart*) *vc*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ (∃ *p*::(*real*, *3*) *cart*. *IN p* (*hull affine* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*)))) ∧ (∃ *c*::*real*. (∀ *w*::(*real*, *3*) *cart*. *IN w* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *c* = *distance* (*p*, *w*)) ∧ (∀ *p′*::(*real*, *3*) *cart*. *IN p′* (*hull affine* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*)))) ∧ (∀ *w*::(*real*, *3*) *cart*. *IN w* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *c* = *distance* (*p′*, *w*)) ⟶ *p* = *p′*)))

thm Collect_geom.COND_FOR_CIRCUMCENTER_PROPERTIESS:

¬ *collinear* (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) (*INSERT* (*?v3.0*::(*real*, *3*) *cart*) *EMPTY*))) ⟶ *IN* (*circumcenter* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*)))) (*hull affine* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*)))) ∧ (∃ *c*::*real*. ∀ *v*::(*real*, *3*) *cart*. *IN v* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*))) ⟶ *c* = *distance* (*circumcenter* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*))), *v*))

thm Collect_geom.DELTA_X14_RRR:

$delta\_x14$ $(?r::real)$ $?r$ $?r$ $(?a::real)$ $(?b::real)$ $(?c::real)$ $=$ $?a * (?b + (?c - ?a))$

thm Collect_geom.DELTA_X1I_RRR:

$delta\_x12\ (?r::real)\ ?r\ ?r\ (?a::real)\ (?b::real)\ (?c::real) = ?c * (?b + (?a - ?c)) \wedge delta\_x13\ ?r\ ?r\ ?r\ ?a\ ?b\ ?c = ?b * (?c + (?a - ?b)) \wedge delta\_x14\ ?r\ ?r\ ?r\ ?a\ ?b\ ?c = ?a * (?c + (?b - ?a))$

thm Collect_geom.PRE_RADV_COND:

$\neg\ collinear\ (INSERT\ (?va::(real,\ 3)\ cart)\ (INSERT\ (?vb::(real,\ 3)\ cart)\ (INSERT\ (?vc::(real,\ 3)\ cart)\ EMPTY))) \longrightarrow (\exists\, c::real.\ \forall\, w::(real,\ 3)\ cart.\ INSERT\ ?va\ (INSERT\ ?vb\ (INSERT\ ?vc\ EMPTY))\ w \longrightarrow c = distance\ (circumcenter\ (INSERT\ ?va\ (INSERT\ ?vb\ (INSERT\ ?vc\ EMPTY))),\ w))$

thm Collect_geom.NOT_COL_IMP_RADV_PROPERTIY:

$\neg\ collinear\ (INSERT\ (?va::(real,\ 3)\ cart)\ (INSERT\ (?vb::(real,\ 3)\ cart)\ (INSERT\ (?vc::(real,\ 3)\ cart)\ EMPTY))) \longrightarrow (\forall\, w::(real,\ 3)\ cart.\ INSERT\ ?va\ (INSERT\ ?vb\ (INSERT\ ?vc\ EMPTY))\ w \longrightarrow radV\ (INSERT\ ?va\ (INSERT\ ?vb\ (INSERT\ ?vc\ EMPTY))) = distance\ (circumcenter\ (INSERT\ ?va\ (INSERT\ ?vb\ (INSERT\ ?vc\ EMPTY))),\ w))$

thm Collect_geom.DELTA_X1I_RRR_conjunct2:

$delta\_x14\ (?r::real)\ ?r\ ?r\ (?a::real)\ (?b::real)\ (?c::real) = ?a * (?c + (?b - ?a))$

thm Collect_geom.DELTA_X1I_RRR_conjunct1:

$delta\_x13\ (?r::real)\ ?r\ ?r\ (?a::real)\ (?b::real)\ (?c::real) = ?b * (?c + (?a - ?b))$

thm Collect_geom.DELTA_X1I_RRR_conjunct0:

$delta\_x12\ (?r::real)\ ?r\ ?r\ (?a::real)\ (?b::real)\ (?c::real) = ?c * (?b + (?a - ?c))$

thm Collect_geom.CIRCUMCENTER_FORMULAR2:

$\forall\,(va::(real,\ 3)\ cart)\ (vb::(real,\ 3)\ cart)\ (vc::(real,\ 3)\ cart)\ (a::real)\ (b::real)\ c::real.\ a = d3\ vb\ vc \wedge b = d3\ va\ vc \wedge c = d3\ va\ vb \wedge \neg\ collinear\ (INSERT\ va\ (INSERT\ vb\ (INSERT\ vc\ EMPTY))) \longrightarrow LET\ (\lambda al\_a::real.\ LET\_END\ (LET\ (\lambda al\_b::real.\ LET\_END\ (LET\ (\lambda al\_c::real.\ LET\_END\ (vector\_add\ (\%\ al\_a\ va)\ (vector\_add\ (\%\ al\_b\ vb)\ (\%\ al\_c\ vc)) = circumcenter\ (INSERT\ va\ (INSERT\ vb\ (INSERT\ vc\ EMPTY)))))\ (c^2 * (a^2 + (b^2 - c^2)) / ups\_x\ (a^2)\ (b^2)\ (c^2))))\ (b^2 * (a^2 + (c^2 - b^2)) / ups\_x\ (a^2)\ (b^2)\ (c^2))))\ (a^2 * (b^2 + (c^2 - a^2)) / ups\_x\ (a^2)\ (b^2)\ (c^2))$

thm Collect_geom.NOT_COLL_IMP_RADV_FORMULAR:

$\forall\,(va::(real,\ 3)\ cart)\ (vb::(real,\ 3)\ cart)\ (vc::(real,\ 3)\ cart)\ (a::real)\ (b::real)\ c::real.\ a = d3\ vb\ vc \wedge b = d3\ va\ vc \wedge c = d3\ va\ vb \wedge \neg\ collinear\ (INSERT$

va (INSERT vb (INSERT vc EMPTY))) ⟶ radV (INSERT va (INSERT vb (INSERT vc EMPTY))) = eta_y a b c

thm Collect_geom2.CDEUSDF_CHANGE:

∀ (va::(real, 3) cart) (vb::(real, 3) cart) (vc::(real, 3) cart) (a::real) (b::real) c::real. a = d3 vb vc ∧ b = d3 va vc ∧ c = d3 va vb ∧ ¬ collinear (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ (∃ p::(real, 3) cart. IN p (hull affine (INSERT va (INSERT vb (INSERT vc EMPTY)))) ∧ (∃ c::real. (∀ w::(real, 3) cart. IN w (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ c = distance (p, w)) ∧ (∀ p′::(real, 3) cart. IN p′ (hull affine (INSERT va (INSERT vb (INSERT vc EMPTY)))) ∧ (∀ w::(real, 3) cart. IN w (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ c = distance (p′, w)) ⟶ p = p′))) ∧ LET (λal_a::real. LET_END (LET (λal_b::real. LET_END (LET (λal_c::real. LET_END (vector_add (% al_a va) (vector_add (% al_b vb) (% al_c vc)) = circumcenter (INSERT va (INSERT vb (INSERT vc EMPTY)))))) (c^2 ∗ (a^2 + (b^2 − c^2)) / ups_x (a^2) (b^2) (c^2)))) (b^2 ∗ (a^2 + (c^2 − b^2)) / ups_x (a^2) (b^2) (c^2)))) (a^2 ∗ (b^2 + (c^2 − a^2)) / ups_x (a^2) (b^2) (c^2)) ∧ radV (INSERT va (INSERT vb (INSERT vc EMPTY))) = eta_y a b c

thm Collect_geom.DIST_EQ_IS_UNIQUE:

SUBSET (INSERT (?u::(real, ?′a::type) cart) (INSERT (?w::(real, ?′a::type) cart) EMPTY)) (hull affine (INSERT (?v1.0::(real, ?′a::type) cart) (INSERT (?v2.0::(real, ?′a::type) cart) (INSERT (?v3.0::(real, ?′a::type) cart) EMPTY)))) ∧ distance (?u, ?v2.0) = distance (?u, ?v1.0) ∧ distance (?u, ?v3.0) = distance (?u, ?v1.0) ∧ distance (?w, ?v2.0) = distance (?w, ?v1.0) ∧ distance (?w, ?v3.0) = distance (?w, ?v1.0) ⟶ ?u = ?w

thm Collect_geom.NEVER_USED_AGAIN:

IN (?p::(real, ?′a::type) cart) (hull affine (INSERT (?va::(real, ?′a::type) cart) (INSERT (?vb::(real, ?′a::type) cart) (INSERT (?vc::(real, ?′a::type) cart) EMPTY)))) ∧ (?c::real) = distance (?p, ?va) ∧ ?c = distance (?p, ?vb) ∧ ?c = distance (?p, ?vc) ⟶ (∀ p′::(real, ?′a::type) cart. (IN p′ (hull affine (INSERT ?va (INSERT ?vb (INSERT ?vc EMPTY)))) ∧ distance (p′, ?vb) = distance (p′, ?va) ∧ distance (p′, ?vc) = distance (p′, ?va)) = (IN p′ (hull affine (INSERT ?va (INSERT ?vb (INSERT ?vc EMPTY)))) ∧ ?c = distance (p′, ?va) ∧ ?c = distance (p′, ?vb) ∧ ?c = distance (p′, ?vc)))

thm Collect_geom.TRUONG_WELL:

∀ (va::(real, 3) cart) (vb::(real, 3) cart) vc::(real, 3) cart. ¬ collinear (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ (∃ p::(real, 3) cart. IN p (hull affine (INSERT va (INSERT vb (INSERT vc EMPTY)))) ∧ (∃ c::real. ∀ w::(real, 3) cart. IN w (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ c = distance (p, w)) ∧ (∀ p′::(real, 3) cart. IN p′ (hull affine (INSERT va (INSERT vb (INSERT vc EMPTY)))) ∧ (∃ c::real. ∀ w::(real, 3) cart. IN w (INSERT va (INSERT vb (INSERT vc EMPTY))) ⟶ c = distance (p′, w)) ⟶ p = p′))

thm Collect_geom.NGAY_MONG6:

∀ (*va*::(*real*, *3*) *cart*) (*vb*::(*real*, *3*) *cart*) *vc*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ (∃ *p*::(*real*, *3*) *cart*. *IN p* (*hull affine* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*)))) ∧ (∃ *c*::*real*. ∀ *w*::(*real*, *3*) *cart*. *IN w* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *c* = *distance* (*p*, *w*)))

thm Collect_geom.CIRCUMCENTER_PROPTIES:

∀ (*va*::(*real*, *3*) *cart*) (*vb*::(*real*, *3*) *cart*) *vc*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *IN* (*circumcenter* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*)))) (*hull affine* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*)))) ∧ (∃ *c*::*real*. ∀ *w*::(*real*, *3*) *cart*. *IN w* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *c* = *distance* (*circumcenter* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))), *w*))

thm Collect_geom.SIMP_DOT_ALEM:

((*0*::*real*) < *dot* (*vector_sub* (*?b*::(*real*, *?'a*::*type*) *cart*) (*?a*::(*real*, *?'a*::*type*) *cart*)) (*?x*::(*real*, *?'a*::*type*) *cart*)) = (*dot ?x* (*vector_sub ?a ?b*) < (*0*::*real*))

thm Collect_geom.MONG7_ROI:

∀ (*p*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. (*distance* (*p*, *a*) = *distance* (*p*, *b*)) = (*dot* (*vector_sub p* (% ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add a b*))) (*vector_sub a b*) = (*0*::*real*))

thm Collect_geom.POXDVXO:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ∧ *p* = *circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) ⟶ *dot* (*vector_sub p* (% ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add v1 v2*))) (*vector_sub v1 v2*) = (*0*::*real*) ∧ *dot* (*vector_sub p* (% ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add v2 v3*))) (*vector_sub v2 v3*) = (*0*::*real*) ∧ *dot* (*vector_sub p* (% ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add v3 v1*))) (*vector_sub v3 v1*) = (*0*::*real*)

thm Collect_geom.NOT_COLL_IMP_RADV_EQ_ETA_Y:

∀ (*va*::(*real*, *3*) *cart*) (*vb*::(*real*, *3*) *cart*) *vc*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) ⟶ *radV* (*INSERT va* (*INSERT vb* (*INSERT vc EMPTY*))) = *eta_y* (*d3 vb vc*) (*d3 va vc*) (*d3 va vb*)

thm Collect_geom.TWO_EQ_IMP_COL3:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *x* = *y* ⟶ *collinear* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))

thm Collect_geom.NOT_CO_IMP_DIST_POS:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))) ⟶ (*0*::*real*) < *distance* (*x*, *y*)

thm Collect_geom.NOT_COLL_IMP_POS_SUM:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \longrightarrow (0::real) < (d3\ x\ y + (d3\ y\ z + d3\ z\ x))\ /\ real\_of\_nat\ (2::nat)$

thm Collect_geom.PER_SET2:

$INSERT\ (?a::?'a::type)\ (INSERT\ (?b::?'a::type)\ EMPTY) = INSERT\ ?b\ (INSERT\ ?a\ EMPTY)$

thm Collect_geom.COLLINEAR_AS_IN_CONV2:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) = (IN\ x\ (conv\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \vee IN\ y\ (conv\ (INSERT\ z\ (INSERT\ x\ EMPTY))) \vee IN\ z\ (conv\ (INSERT\ x\ (INSERT\ y\ EMPTY))))$

thm Collect_geom.COLLINEAR_IMP_POS_UPS2:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \longrightarrow (0::real) < ups\_x\_pow2\ (d3\ x\ y)\ (d3\ y\ z)\ (d3\ z\ x)$

thm Collect_geom.MUL3_SYM:

$\forall (a::real)\ (b::real)\ c::real.\ a * (b * c) = b * (a * c) \wedge a * (b * c) = c * (b * a)$

thm Collect_geom.ETA_X_SYMM:

$\forall (a::real)\ (b::real)\ c::real.\ eta\_x\ a\ b\ c = eta\_x\ b\ a\ c \wedge eta\_x\ a\ b\ c = eta\_x\ c\ b\ a$

thm Collect_geom.ETA_Y_SYYM:

$\forall (x::real)\ (y::real)\ z::real.\ eta\_y\ x\ y\ z = eta\_y\ y\ x\ z \wedge eta\_y\ x\ y\ z = eta\_y\ z\ y\ x$

thm Collect_geom.NOT_COL3_IMP_DIFF:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ \neg\ collinear\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) \longrightarrow \neg\ (a = b \vee a = c \vee b = c)$

thm Collect_geom.POW2_COND_LT:

$\forall (a::real)\ b::real.\ (0::real) < a \wedge (0::real) < b \longrightarrow (a \leq b) = (a^2 \leq b^2)$

thm Trigonometry2.DIV_POW2:

$((?a::real)\ /\ (?b::real))^2 = ?a^2\ /\ ?b^2$

thm Collect_geom.ETA_Y_2:

$eta\_y\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat)) = real\_of\_nat\ (2::nat)\ /\ sqrt\ (real\_of\_nat\ (3::nat))$

thm DEF_cayleytr:

cayleytr = (λ(_2081959::real) (_2081960::real) (_2081961::real) (_2081962::real) (_2081963::real) (_2081964::real) (_2081965::real) (_2081966::real) (_2081967::real) _2081968::real. real_of_nat (2::nat) ∗ (_2081963 ∗ (_2081965 ∗ _2081966)) + (real_of_nat (2::nat) ∗ (_2081963 ∗ (_2081964 ∗ _2081967)) + (− (1::real) ∗ (_2081963^2 ∗ _2081968) + (− real_of_nat (2::nat) ∗ (_2081962 ∗ (_2081963 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081962 ∗ (_2081963 ∗ _2081964)) + (real_of_nat (2::nat) ∗ (_2081962 ∗ _2081963^2) + (− real_of_nat (2::nat) ∗ (_2081961 ∗ (_2081963 ∗ _2081967)) + (− real_of_nat (2::nat) ∗ (_2081961 ∗ (_2081963 ∗ _2081965)) + (real_of_nat (2::nat) ∗ (_2081961 ∗ _2081963^2) + (real_of_nat (4::nat) ∗ (_2081961 ∗ (_2081962 ∗ _2081963)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081965 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081964 ∗ _2081967)) + (real_of_nat (4::nat) ∗ (_2081960 ∗ (_2081964 ∗ _2081965)) + (real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081963 ∗ _2081968)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081963 ∗ _2081965)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081963 ∗ _2081964)) + (real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081962 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081962 ∗ _2081964)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081962 ∗ _2081963)) + (real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081961 ∗ _2081967)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081961 ∗ _2081965)) + (− real_of_nat (2::nat) ∗ (_2081960 ∗ (_2081961 ∗ _2081963)) + (− (1::real) ∗ (_2081960^2 ∗ _2081968) + (real_of_nat (2::nat) ∗ (_2081960^2 ∗ _2081965) + (real_of_nat (2::nat) ∗ (_2081960^2 ∗ _2081964) + (real_of_nat (4::nat) ∗ (_2081959 ∗ (_2081966 ∗ _2081967)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081965 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081964 ∗ _2081967)) + (real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081963 ∗ _2081968)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081963 ∗ _2081967)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081963 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081962 ∗ _2081966)) + (real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081962 ∗ _2081964)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081962 ∗ _2081963)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081961 ∗ _2081967)) + (real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081961 ∗ _2081965)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081961 ∗ _2081963)) + (real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081968)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081967)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081966)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081965)) + (− real_of_nat (2::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081964)) + (real_of_nat (4::nat) ∗ (_2081959 ∗ (_2081960 ∗ _2081963)) + (− (1::real) ∗ (_2081959^2 ∗ _2081968) + (real_of_nat (2::nat) ∗ (_2081959^2 ∗ _2081967) + real_of_nat (2::nat) ∗ (_2081959^2 ∗ _2081966))

thm Collect_geom.cayleytr:

∀ (x15::real) (x14::real) (x25::real) (x24::real) (x13::real) (x23::real) (x45::real) (x35::real) (x12::real) x34::real. cayleytr x12 x13 x14 x15 x23 x24 x25 x34 x35 x45 = real_of_nat (2::nat) ∗ (x23 ∗ (x25 ∗ x34)) + (real_of_nat (2::nat) ∗ (x23 ∗ (x24 ∗ x35)) + (− (1::real) ∗ (x23^2 ∗ x45) + (− real_of_nat (2::nat) ∗ (x15 ∗ (x23 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x15 ∗ (x23 ∗ x24)) + (real_of_nat

(2::nat) ∗ (x15 ∗ x23²) + (− real_of_nat (2::nat) ∗ (x14 ∗ (x23 ∗ x35)) + (− real_of_nat (2::nat) ∗ (x14 ∗ (x23 ∗ x25)) + (real_of_nat (2::nat) ∗ (x14 ∗ x23²) + (real_of_nat (4::nat) ∗ (x14 ∗ (x15 ∗ x23)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x25 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x24 ∗ x35)) + (real_of_nat (4::nat) ∗ (x13 ∗ (x24 ∗ x25)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x23 ∗ x45)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x23 ∗ x25)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x23 ∗ x24)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ x24)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x15 ∗ x23)) + (real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ x35)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ x25)) + (− real_of_nat (2::nat) ∗ (x13 ∗ (x14 ∗ x23)) + (− (1::real) ∗ (x13² ∗ x45) + (real_of_nat (2::nat) ∗ (x13² ∗ x25) + (real_of_nat (2::nat) ∗ (x13² ∗ x24) + (real_of_nat (4::nat) ∗ (x12 ∗ (x34 ∗ x35)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x25 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x24 ∗ x35)) + (real_of_nat (2::nat) ∗ (x12 ∗ (x23 ∗ x45)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x23 ∗ x35)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x23 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x15 ∗ x34)) + (real_of_nat (2::nat) ∗ (x12 ∗ (x15 ∗ x24)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x15 ∗ x23)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x14 ∗ x35)) + (real_of_nat (2::nat) ∗ (x12 ∗ (x14 ∗ x25)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x14 ∗ x23)) + (real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ x45)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ x35)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ x34)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ x25)) + (− real_of_nat (2::nat) ∗ (x12 ∗ (x13 ∗ x24)) + (real_of_nat (4::nat) ∗ (x12 ∗ (x13 ∗ x23)) + (− (1::real) ∗ (x12² ∗ x45) + (real_of_nat (2::nat) ∗ (x12² ∗ x35) + real_of_nat (2::nat) ∗ (x12² ∗ x34))

thm Collect_geom.LEMMA50:

cayleyR (?x12.0::real) (?x13.0::real) (?x14.0::real) (?x15.0::real) (?x23.0::real) (?x24.0::real) (?x25.0::real) (?x34.0::real) (?x35.0::real) (?x45.0::real) = ups_x ?x12.0 ?x13.0 ?x23.0 ∗ ?x45.0² + (cayleytr ?x12.0 ?x13.0 ?x14.0 ?x15.0 ?x23.0 ?x24.0 ?x25.0 ?x34.0 ?x35.0 (0::real) ∗ ?x45.0 + cayleyR ?x12.0 ?x13.0 ?x14.0 ?x15.0 ?x23.0 ?x24.0 ?x25.0 ?x34.0 ?x35.0 (0::real))

thm Collect_geom.plane:

∀ x::(real, ?'a::type) cart ⇒ bool. plane x = (∃ (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. ¬ collinear (INSERT u (INSERT v (INSERT w EMPTY))) ∧ x = hull affine (INSERT u (INSERT v (INSERT w EMPTY))))

thm DEF_coplanar_alt:

coplanar_alt = (λ_2082099::(real, ?'a::type) cart ⇒ bool. ∃ x::(real, ?'a::type) cart ⇒ bool. plane x ∧ SUBSET _2082099 x)

thm Collect_geom.coplanar_alt:

∀ S::(real, ?'a::type) cart ⇒ bool. coplanar_alt S = (∃ x::(real, ?'a::type) cart ⇒ bool. plane x ∧ SUBSET S x)

thm DEF_condA:

$condA = (\lambda(\_2082104::(real, 3)\ cart)\ (\_2082105::(real, 3)\ cart)\ (\_2082106::(real, 3)\ cart)\ (\_2082107::(real, 3)\ cart)\ (\_2082108::real)\ (\_2082109::real)\ (\_2082110::real)\ (\_2082111::real)\ (\_2082112::real)\ \_2082113::?'a::type.\ \_2082104 \neq \_2082105 \wedge coplanar\_alt\ (INSERT\ \_2082104\ (INSERT\ \_2082105\ (INSERT\ \_2082106\ (INSERT\ \_2082107\ EMPTY)))) \wedge (distance\ (\_2082104, \_2082105))^2 = \_2082108 \wedge (distance\ (\_2082104, \_2082106))^2 = \_2082109 \wedge (distance\ (\_2082104, \_2082107))^2 = \_2082110 \wedge (distance\ (\_2082105, \_2082106))^2 = \_2082111 \wedge (distance\ (\_2082105, \_2082107))^2 = \_2082112)$

thm Collect_geom.condA:

$\forall (x34::?'a::type)\ (x12::real)\ (x13::real)\ (v1::(real, 3)\ cart)\ (x14::real)\ (v3::(real, 3)\ cart)\ (x23::real)\ (v2::(real, 3)\ cart)\ (v4::(real, 3)\ cart)\ x24::real.\ condA\ v1\ v2\ v3\ v4\ x12\ x13\ x14\ x23\ x24\ x34 = (v1 \neq v2 \wedge coplanar\_alt\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY)))) \wedge (distance\ (v1, v2))^2 = x12 \wedge (distance\ (v1, v3))^2 = x13 \wedge (distance\ (v1, v4))^2 = x14 \wedge (distance\ (v2, v3))^2 = x23 \wedge (distance\ (v2, v4))^2 = x24)$

thm DEF_det_vec3:

$det\_vec3 = (\lambda(\_2082244::(real, 3)\ cart)\ (\_2082245::(real, 3)\ cart)\ \_2082246::(real, 3)\ cart.\ \$\ \_2082244\ (1::nat) * (\$\ \_2082245\ (2::nat) * \$\ \_2082246\ (3::nat)) + (\$\ \_2082245\ (1::nat) * (\$\ \_2082246\ (2::nat) * \$\ \_2082244\ (3::nat)) + (\$\ \_2082246\ (1::nat) * (\$\ \_2082244\ (2::nat) * \$\ \_2082245\ (3::nat)) - (\$\ \_2082244\ (1::nat) * (\$\ \_2082246\ (2::nat) * \$\ \_2082245\ (3::nat)) + (\$\ \_2082245\ (1::nat) * (\$\ \_2082244\ (2::nat) * \$\ \_2082246\ (3::nat)) + \$\ \_2082246\ (1::nat) * (\$\ \_2082245\ (2::nat) * \$\ \_2082244\ (3::nat)))))))$

thm Collect_geom.det_vec3:

$\forall (c::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ a::(real, 3)\ cart.\ det\_vec3\ a\ b\ c = \$\ a\ (1::nat) * (\$\ b\ (2::nat) * \$\ c\ (3::nat)) + (\$\ b\ (1::nat) * (\$\ c\ (2::nat) * \$\ a\ (3::nat)) + (\$\ c\ (1::nat) * (\$\ a\ (2::nat) * \$\ b\ (3::nat)) - (\$\ a\ (1::nat) * (\$\ c\ (2::nat) * \$\ b\ (3::nat)) + (\$\ b\ (1::nat) * (\$\ a\ (2::nat) * \$\ c\ (3::nat)) + \$\ c\ (1::nat) * (\$\ b\ (2::nat) * \$\ a\ (3::nat))))))$

thm Collect_geom.COPLANAR_DET_EQ_0:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ v3::(real, 3)\ cart.\ coplanar\_alt\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) = (det\ (vector\ [vector\_sub\ v1\ v0, vector\_sub\ v2\ v0, vector\_sub\ v3\ v0]) = (0::real))$

thm Collect_geom.COPLANAR:

$(2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (\forall s::(real, ?'a::type)\ cart \Rightarrow bool.\ coplanar\_alt\ s = (\exists (u::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ SUBSET\ s\ (hull\ affine\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))))))$

thm Trigonometry2.DET_VEC3_EXPAND:

$det\ (vector\ [?a::(real,\ 3)\ cart,\ ?b::(real,\ 3)\ cart,\ ?c::(real,\ 3)\ cart]) = det\_vec3\ ?a\ ?b\ ?c$

thm Collect_geom.COPLANAR_DET_VEC3_EQ_0:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ coplanar\_alt\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) = (det\_vec3\ (vector\_sub\ v1\ v0)\ (vector\_sub\ v2\ v0)\ (vector\_sub\ v3\ v0) = (0::real))$

thm Collect_geom.COPLANAR_3:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ c::(real,\ ?'a::type)\ cart.\ (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow coplanar\_alt\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))$

thm Collect_geom.NONCOPLANAR_4_DISTINCT:

$\forall (a::(real,\ ?'a::type)\ cart)\ (b::(real,\ ?'a::type)\ cart)\ (c::(real,\ ?'a::type)\ cart)\ d::(real,\ ?'a::type)\ cart.\ \neg\ coplanar\_alt\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \wedge (2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow a \neq b \wedge a \neq c \wedge a \neq d \wedge b \neq c \wedge b \neq d \wedge c \neq d$

thm Collect_geom.NONCOPLANAR_3_BASIS:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (v0::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ \neg\ coplanar\_alt\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow (\exists (t1::real)\ (t2::real)\ t3::real.\ v = vector\_add\ (\%\ t1\ (vector\_sub\ v1\ v0))\ (vector\_add\ (\%\ t2\ (vector\_sub\ v2\ v0))\ (\%\ t3\ (vector\_sub\ v3\ v0))) \wedge (\forall (ta::real)\ (tb::real)\ tc::real.\ v = vector\_add\ (\%\ ta\ (vector\_sub\ v1\ v0))\ (vector\_add\ (\%\ tb\ (vector\_sub\ v2\ v0))\ (\%\ tc\ (vector\_sub\ v3\ v0))) \longrightarrow ta = t1 \wedge tb = t2 \wedge tc = t3))$

thm Collect_geom.DET_VEC3_AND_DELTA:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ d::(real,\ 3)\ cart.\ real\_of\_nat\ (4::nat) * (det\_vec3\ (vector\_sub\ a\ d)\ (vector\_sub\ b\ d)\ (vector\_sub\ c\ d))^2 = delta\ ((d3\ a\ d)^2)\ ((d3\ b\ d)^2)\ ((d3\ c\ d)^2)\ ((d3\ a\ b)^2)\ ((d3\ a\ c)^2)\ ((d3\ b\ c)^2)$

thm Collect_geom.LEMMA15:

$\forall (x1::(real,\ 3)\ cart)\ (x2::(real,\ 3)\ cart)\ (x3::(real,\ 3)\ cart)\ x4::(real,\ 3)\ cart.\ LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x14::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda x24::real.\ LET\_END\ (LET\ (\lambda x34::real.\ LET\_END\ (coplanar\_alt\ (INSERT\ x1\ (INSERT\ x2\ (INSERT\ x3\ (INSERT\ x4\ EMPTY)))) = (delta\ x12\ x13\ x14\ x23\ x24\ x34 = (0::real))))\ ((distance\ (x3,\ x4))^2)))\ ((distance\ (x2,\ x4))^2)))\ ((distance\ (x2,\ x3))^2)))\ ((distance\ (x1,\ x4))^2)))\ ((distance\ (x1,\ x3))^2)))\ ((distance\ (x1,\ x2))^2)$

thm Collect_geom.muy_delta:

$muy\_delta = delta$

thm Collect_geom.VCRJIHC:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) (*x34*::*?'a*::*type*) (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) *x24*::*real*. *condA v1 v2 v3 v4 x12 x13 x14 x23 x24 x34* $\longrightarrow$ *muy_delta x12 x13 x14 x23 x24* ((*distance* (*v3*, *v4*))^2) = (*0*::*real*)

thm Collect_geom.ZERO_NEUTRAL:

$\forall$ *x*::*real*. (*0*::*real*) $*$ *x* = (*0*::*real*) $\wedge$ *x* $*$ (*0*::*real*) = (*0*::*real*) $\wedge$ (*0*::*real*) + *x* = *x* $\wedge$ *x* + (*0*::*real*) = *x* $\wedge$ *x* $-$ (*0*::*real*) = *x* $\wedge$ $-$ (*0*::*real*) = (*0*::*real*)

thm Collect_geom.EQUATE_CONEFS_POLINOMIAL_POW2:

$\forall$ (*a*::*real*) (*b*::*real*) (*c*::*real*) (*aa*::*real*) (*bb*::*real*) *cc*::*real*. ($\forall$ *x*::*real*. *a* $*$ *x*^{2} + (*b* $*$ *x* + *c*) = *aa* $*$ *x*^{2} + (*bb* $*$ *x* + *cc*)) = (*a* = *aa* $\wedge$ *b* = *bb* $\wedge$ *c* = *cc*)

thm Collect_geom.LEMMA51:

$\forall$ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x15*::*real*) (*x23*::*real*) (*x24*::*real*) (*x25*::*real*) (*x34*::*real*) (*x35*::*real*) (*a*::*real*) (*b*::*real*) *c*::*real*. ($\forall$ *x45*::*real*. *cayleyR x12 x13 x14 x15 x23 x24 x25 x34 x35 x45* = *a* $*$ *x45*^{2} + (*b* $*$ *x45* + *c*)) $\longrightarrow$ *b*^{2} $-$ *real_of_nat* (*4*::*nat*) $*$ (*a* $*$ *c*) = *real_of_nat* (*16*::*nat*) $*$ (*delta x12 x13 x14 x23 x24 x34* $*$ *delta x12 x13 x15 x23 x25 x35*)

thm Collect_geom.NOT_TWO_EQ_IMP_COL_EQUAVALENT:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *v1* $\neq$ *v2* $\longrightarrow$ *collinear* (*INSERT v* (*INSERT v1* (*INSERT v2 EMPTY*))) = *IN v* (*aff* (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Collect_geom.LEMMA30:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) (*x34*::*?'a*::*type*) (*a*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*) (*b*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*) *c*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real*. *condA v1 v2 v3 v4 x12 x13 x14 x23 x24 x34* $\wedge$ ($\forall$ (*x12*::*real*) (*x13*::*real*) (*x14*::*real*) (*x23*::*real*) (*x24*::*real*) *x34*::*real*. *muy_delta x12 x13 x14 x23 x24 x34* = *a x12 x13 x14 x23 x24* $*$ *x34*^{2} + (*b x12 x13 x14 x23 x24* $*$ *x34* + *c x12 x13 x14 x23 x24*)) $\longrightarrow$ (*IN v3* (*aff* (*INSERT v1* (*INSERT v2 EMPTY*))) $\vee$ *IN v4* (*aff* (*INSERT v1* (*INSERT v2 EMPTY*)))) = ((*b x12 x13 x14 x23 x24*)^2 $-$ *real_of_nat* (*4*::*nat*) $*$ (*a x12 x13 x14 x23 x24* $*$ *c x12 x13 x14 x23 x24*) = (*0*::*real*))

thm DEF_muy_v:

muy_v = (λ(*_2085073*::(*real*, *?'a*::*type*) *cart*) (*_2085074*::(*real*, *?'a*::*type*) *cart*) (*_2085075*::(*real*, *?'a*::*type*) *cart*) (*_2085076*::(*real*, *?'a*::*type*) *cart*) (*_2085077*::(*real*, *?'a*::*type*) *cart*) *_2085078*::*real*. *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x15*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x25*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*LET* (λ*x35*::*real*. *LET_END* (*cayleyR x12 x13 x14 x15 x23 x24 x25 x34 x35 _2085078*)) ((*distance* (*_2085075*, *_2085077*))^2))) ((*distance* (*_2085075*, *_2085076*))^2))) ((*distance* (*_2085074*, *_2085077*))^2)))

$((distance\ (\_2085074,\ \_2085076))^2)))\ ((distance\ (\_2085074,\ \_2085075))^2)))$
$((distance\ (\_2085073,\ \_2085077))^2)))\ ((distance\ (\_2085073,\ \_2085076))^2)))$
$((distance\ (\_2085073,\ \_2085075))^2)))\ ((distance\ (\_2085073,\ \_2085074))^2))$

thm Collect_geom.muy_v:

$\forall (x45::real)\ (x5::(real,\ ?'a::type)\ cart)\ (x4::(real,\ ?'a::type)\ cart)\ (x3::(real,\ ?'a::type)\ cart)\ (x1::(real,\ ?'a::type)\ cart)\ x2::(real,\ ?'a::type)\ cart.\ muy\_v\ x1\ x2\ x3\ x4\ x5\ x45 = LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x14::real.\ LET\_END\ (LET\ (\lambda x15::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda x24::real.\ LET\_END\ (LET\ (\lambda x25::real.\ LET\_END\ (LET\ (\lambda x34::real.\ LET\_END\ (LET\ (\lambda x35::real.\ LET\_END\ (cayleyR\ x12\ x13\ x14\ x15\ x23\ x24\ x25\ x34\ x35\ x45))\ ((distance\ (x3,\ x5))^2)))\ ((distance\ (x3,\ x4))^2)))\ ((distance\ (x2,\ x5))^2)))\ ((distance\ (x2,\ x4))^2)))\ ((distance\ (x2,\ x3))^2)))\ ((distance\ (x1,\ x5))^2)))\ ((distance\ (x1,\ x4))^2)))\ ((distance\ (x1,\ x3))^2)))\ ((distance\ (x1,\ x2))^2)$

thm Collect_geom.ALE:

$\forall (x12::real)\ (x13::real)\ (x14::real)\ (x15::real)\ (x23::real)\ (x24::real)\ (x25::real)\ (x34::real)\ x35::real.\ (\forall (a::real)\ (b::real)\ c::real.\ (\forall x::real.\ cayleyR\ x12\ x13\ x14\ x15\ x23\ x24\ x25\ x34\ x35\ x = a * x^2 + (b * x + c)) \longrightarrow b^2 - real\_of\_nat\ (4::nat) * (a * c) = (0::real)) \longrightarrow (cayleytr\ x12\ x13\ x14\ x15\ x23\ x24\ x25\ x34\ x35\ (0::real))^2 - real\_of\_nat\ (4::nat) * (ups\_x\ x12\ x13\ x23 * cayleyR\ x12\ x13\ x14\ x15\ x23\ x24\ x25\ x34\ x35\ (0::real)) = (0::real)$

thm Collect_geom.DISCRIMINANT_OF_CAY:

$(cayleytr\ (?x12.0::real)\ (?x13.0::real)\ (?x14.0::real)\ (?x15.0::real)\ (?x23.0::real)\ (?x24.0::real)\ (?x25.0::real)\ (?x34.0::real)\ (?x35.0::real)\ (0::real))^2 - real\_of\_nat\ (4::nat) * (ups\_x\ ?x12.0\ ?x13.0\ ?x23.0 * cayleyR\ ?x12.0\ ?x13.0\ ?x14.0\ ?x15.0\ ?x23.0\ ?x24.0\ ?x25.0\ ?x34.0\ ?x35.0\ (0::real)) = real\_of\_nat\ (16::nat) * (delta\ ?x12.0\ ?x13.0\ ?x14.0\ ?x23.0\ ?x24.0\ ?x34.0 * delta\ ?x12.0\ ?x13.0\ ?x15.0\ ?x23.0\ ?x25.0\ ?x35.0)$

thm Collect_geom.GDLRUZB:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (v4::(real,\ 3)\ cart)\ (v5::(real,\ 3)\ cart)\ (a::?'c::type)\ (b::?'b::type)\ c::?'a::type.\ (coplanar\_alt\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY)))) \vee coplanar\_alt\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v5\ EMPTY))))) = (\forall (a::real)\ (b::real)\ c::real.\ (\forall x::real.\ muy\_v\ v1\ v2\ v3\ v4\ v5\ x = a * x^2 + (b * x + c)) \longrightarrow b^2 - real\_of\_nat\ (4::nat) * (a * c) = (0::real))$

thm Collect_geom.DET_VECC3_AND_DELTA:

$\forall (d::(real,\ 3)\ cart)\ (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ c::(real,\ 3)\ cart.\ delta\ ((d3\ d\ a)^2)\ ((d3\ d\ b)^2)\ ((d3\ d\ c)^2)\ ((d3\ a\ b)^2)\ ((d3\ a\ c)^2)\ ((d3\ b\ c)^2) = real\_of\_nat\ (4::nat) * (det\_vec3\ (vector\_sub\ a\ d)\ (vector\_sub\ b\ d)\ (vector\_sub\ c\ d))^2$

thm Collect_geom2.DELTA_POS_4POINTS:

$\forall$ (*x1*::(*real*, *3*) *cart*) (*x2*::(*real*, *3*) *cart*) (*x3*::(*real*, *3*) *cart*) *x4*::(*real*, *3*) *cart*. (*0*::*real*) $\leq$ *delta* ((*distance* (*x1*, *x2*))^2) ((*distance* (*x1*, *x3*))^2) ((*distance* (*x1*, *x4*))^2) ((*distance* (*x2*, *x3*))^2) ((*distance* (*x2*, *x4*))^2) ((*distance* (*x3*, *x4*))^2)

thm Collect_geom.UPS_X_POS:

(*0*::*real*) $\leq$ *ups_x* ((*vector_norm* (*vector_sub* (*?x1.0*::(*real*, *3*) *cart*) (*?x2.0*::(*real*, *3*) *cart*)))^2) ((*vector_norm* (*vector_sub* *?x1.0* (*?x3.0*::(*real*, *3*) *cart*)))^2) ((*vector_norm* (*vector_sub* *?x2.0* *?x3.0*))^2)

thm Collect_geom.UPS_X_SYM:

$\forall$ (*x*::*real*) (*y*::*real*) *z*::*real*. *ups_x x y z* = *ups_x y x z* $\wedge$ *ups_x x y z* = *ups_x x z y* $\wedge$ *ups_x x y z* = *ups_x x z y*

thm DEF_mk_vec3:

mk_vec3 = (λ(*_2085711*::*?'b*::*type*) (*_2085712*::*?'b*::*type*) *_2085713*::*?'b*::*type*. *vector* [*_2085711*, *_2085712*, *_2085713*])

thm Collect_geom2.mk_vec3:

$\forall$ (*a*::*?'b*::*type*) (*b*::*?'b*::*type*) *c*::*?'b*::*type*. *mk_vec3 a b c* = *vector* [*a*, *b*, *c*]

thm DEF_real3_of_triple:

real3_of_triple = (λ*_2085732*::*real* $\times$ *real* $\times$ *real*. *mk_vec3* (*fst _2085732*) (*fst* (*snd _2085732*)) (*snd* (*snd _2085732*)))

thm Collect_geom2.real3_of_triple:

$\forall$ (*a*::*real*) (*b*::*real*) *c*::*real*. *real3_of_triple* (*a*, *b*, *c*) = *mk_vec3 a b c*

thm DEF_triple_of_real3:

triple_of_real3 = (λ*_2085745*::(*real*, *3*) *cart*. ($ *_2085745* (*1*::*nat*), $ *_2085745* (*2*::*nat*), $ *_2085745* (*3*::*nat*)))

thm Collect_geom2.triple_of_real3:

$\forall$ *v*::(*real*, *3*) *cart*. *triple_of_real3 v* = ($ *v* (*1*::*nat*), $ *v* (*2*::*nat*), $ *v* (*3*::*nat*))

thm Collect_geom2.CAYLEYR_5POINTS:

$\forall$ (*x1*::(*real*, *3*) *cart*) (*x2*::(*real*, *3*) *cart*) (*x3*::(*real*, *3*) *cart*) (*x4*::(*real*, *3*) *cart*) *x5*::(*real*, *3*) *cart*. *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x15*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x25*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*LET* (λ*x35*::*real*. *LET_END* (*LET* (λ*x45*::*real*. *LET_END* (*cayleyR x12 x13 x14 x15 x23 x24 x25 x34 x35 x45* = (*0*::*real*))) ((*distance* (*x4*, *x5*))^2))) ((*distance* (*x3*, *x5*))^2))) ((*distance* (*x3*, *x4*))^2))) ((*distance* (*x2*, *x5*))^2))) ((*distance* (*x2*, *x4*))^2))) ((*distance* (*x2*, *x3*))^2))) ((*distance* (*x1*, *x5*))^2))) ((*distance* (*x1*, *x4*))^2))) ((*distance* (*x1*, *x3*))^2))) ((*distance* (*x1*, *x2*))^2)

thm Collect_geom2.LEMMA3:

$\forall$ (*x1*::(*real*, *3*) *cart*) (*x2*::(*real*, *3*) *cart*) (*x3*::(*real*, *3*) *cart*) (*x4*::(*real*, *3*) *cart*) *x5*::(*real*, *3*) *cart*. *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x15*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x25*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*LET* (λ*x35*::*real*. *LET_END* (*LET* (λ*x45*::*real*. *LET_END* ((*0*::*real*) $\leq$ *ups_x x12 x13 x23* $\wedge$ (*0*::*real*) $\leq$ *delta x12 x13 x14 x23 x24 x34* $\wedge$ *cayleyR x12 x13 x14 x15 x23 x24 x25 x34 x35 x45* = (*0*::*real*))) ((*distance* (*x4*, *x5*))^2))) ((*distance* (*x3*, *x5*))^2))) ((*distance* (*x3*, *x4*))^2))) ((*distance* (*x2*, *x5*))^2))) ((*distance* (*x2*, *x4*))^2))) ((*distance* (*x2*, *x3*))^2))) ((*distance* (*x1*, *x5*))^2))) ((*distance* (*x1*, *x4*))^2))) ((*distance* (*x1*, *x3*))^2))) ((*distance* (*x1*, *x2*))^2)

thm Collect_geom2.LEMMA52:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v5*::(*real*, *3*) *cart*. *muy_v v1 v2 v3 v4 v5* ((*d3 v4 v5*)^2) = (*0*::*real*)

thm Collect_geom2.PRE_VIET:

$\forall$ (*x*::*real*) (*x1*::*real*) *x2*::*real*. (*x* − *x1*) ∗ (*x* − *x2*) = *x*^{2} − (*x1* + *x2*) ∗ *x* + *x1* ∗ *x2* $\wedge$ (*?a*::*real*) ∗ ((*x* − *x1*) ∗ (*x* − *x2*)) = *?a* ∗ *x*^{2} + (− *?a* ∗ (*x1* + *x2*) ∗ *x* + *?a* ∗ (*x1* ∗ *x2*))

thm Collect_geom2.VIET_THEOREM:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*a*::*real*) (*b*::*real*) *c*::*real*. ($\forall$ *x*::*real*. *a* ∗ *x*^{2} + (*b* ∗ *x* + *c*) = *a* ∗ ((*x* − *x1*) ∗ (*x* − *x2*))) $\longrightarrow$ − *b* = *a* ∗ (*x1* + *x2*) $\wedge$ *c* = *a* ∗ (*x1* ∗ *x2*)

thm Collect_geom2.ADD_SUB_POW2_EX_conjunct0:

((*?a*::*real*) + (*?b*::*real*))^2 = *?a*^{2} + (*real_of_nat* (*2*::*nat*) ∗ (*?a* ∗ *?b*) + *?b*^{2})

thm Collect_geom2.ADD_SUB_POW2_EX:

((*?a*::*real*) + (*?b*::*real*))^2 = *?a*^{2} + (*real_of_nat* (*2*::*nat*) ∗ (*?a* ∗ *?b*) + *?b*^{2}) $\wedge$ (*?a* − *?b*)^2 = *?a*^{2} − *real_of_nat* (*2*::*nat*) ∗ (*?a* ∗ *?b*) + *?b*^{2}

thm Collect_geom2.PRESENT_SUB_POW2:

$\forall$ (*a*::*real*) *b*::*real*. (*a* − *b*)^2 = (*a* + *b*)^2 − *real_of_nat* (*4*::*nat*) ∗ (*a* ∗ *b*)

thm Collect_geom2.DIST_ROOT_AND_DISCRIMINANT:

$\forall$ (*a*::*real*) (*b*::*real*) (*c*::*real*) (*x1*::*real*) *x2*::*real*. ($\forall$ *x*::*real*. *a* ∗ *x*^{2} + (*b* ∗ *x* + *c*) = *a* ∗ ((*x* − *x1*) ∗ (*x* − *x2*))) $\longrightarrow$ *a*^{2} ∗ (*x1* − *x2*)^2 = *b*^{2} − *real_of_nat* (*4*::*nat*) ∗ (*a* ∗ *c*)

thm Collect_geom2.REAL_EQ_TO_LE_LT:

((*?a*::*real*) = (*?b*::*real*)) = ($\neg$ (*?a* < *?b* $\vee$ *?b* < *?a*))

thm Collect_geom2.FEBRUARY_13_09:

((*0*::*real*) < *dot* (*vector_sub* (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*)) (*vector_sub* (% (*real_of_nat* (*2*::*nat*)) (*?x*::(*real*, *?'a*::*type*) *cart*)) (*vector_add*

$?u\ ?v))) = ((0::real) < dot\ (vector\_sub\ ?u\ ?v)\ (vector\_sub\ ?x\ (\%\ ((1::real)\ /\ real\_of\_nat\ (2::nat))\ (vector\_add\ ?u\ ?v))))$

thm Collect_geom2.SUB_DOT_NEG_TO_POS:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ (dot\ (vector\_sub\ a\ b)\ (?x::(real,\ ?'a::type)\ cart) < (0::real)) = ((0::real) < dot\ (vector\_sub\ b\ a)\ ?x)$

thm Collect_geom2.BXVMKNF:

$\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ u \neq v \longrightarrow plane\_norm\ (bis\ u\ v)$

thm Collect_geom2.DELTA_X34_B:

$\forall (x12::real)\ (x13::real)\ (x14::real)\ (x23::real)\ (x24::real)\ x::real.\ delta\_x34\ x12\ x13\ x14\ x23\ x24\ x = -\ real\_of\_nat\ (2::nat) * (x12 * x) + b\_coef\ x12\ x13\ x14\ x23\ x24$

thm Collect_geom2.EQ_SQRT_POW2_EQ:

$(0::real) \leq (?a::real) \wedge (0::real) \leq (?b::real) \longrightarrow (?a = sqrt\ ?b) = (?a^2 = ?b)$

thm Trigonometry2.MUL_POW2:

$((?a::real) * (?b::real))^2 = ?a^2 * ?b^2$

thm Collect_geom2.LEMMA33:

$\forall (x34::?'a::type)\ (x12::real)\ (x13::real)\ (v1::(real,\ 3)\ cart)\ (x14::real)\ (v3::(real,\ 3)\ cart)\ (x23::real)\ (v2::(real,\ 3)\ cart)\ (v4::(real,\ 3)\ cart)\ (x24::real)\ (x34'::real)\ (x34''::real)\ a::real.\ condA\ v1\ v2\ v3\ v4\ x12\ x13\ x14\ x23\ x24\ x34 \wedge (\forall x::real.\ muy\_delta\ x12\ x13\ x14\ x23\ x24\ x = a * ((x - x34') * (x - x34''))) \wedge x34' \leq x34'' \longrightarrow delta\_x34\ x12\ x13\ x14\ x23\ x24\ x34' = sqrt\ (ups\_x\ x12\ x13\ x23 * ups\_x\ x12\ x14\ x24) \wedge delta\_x34\ x12\ x13\ x14\ x23\ x24\ x34'' = -\ sqrt\ (ups\_x\ x12\ x13\ x23 * ups\_x\ x12\ x14\ x24)$

thm Collect_geom2.LEMMA_OF_LE20:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ real\_of\_nat\ (2::nat) \leq d3\ x\ y \wedge d3\ x\ y \leq DECIMAL\ (252::nat)\ (100::nat) \wedge real\_of\_nat\ (2::nat) \leq d3\ x\ z \wedge d3\ x\ z \leq DECIMAL\ (22::nat)\ (10::nat) \wedge real\_of\_nat\ (2::nat) \leq d3\ y\ z \wedge d3\ y\ z \leq DECIMAL\ (22::nat)\ (10::nat) \longrightarrow \neg\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY)))$

thm Collect_geom2.LT_POW2_EQ_LT:

$(0::real) < (?a::real) \wedge (0::real) < (?b::real) \longrightarrow (?a < ?b) = (?a^2 < ?b^2)$

thm Collect_geom2.ETA_Y_LT_SQRT2:

$eta\_y\ (DECIMAL\ (22::nat)\ (10::nat))\ (DECIMAL\ (22::nat)\ (10::nat))\ (DECIMAL\ (252::nat)\ (100::nat)) < sqrt\ (DECIMAL\ (2::nat)\ (1::nat))$

thm Collect_geom2.ETA_YY_LT_SQRT2:

$eta\_y\ (DECIMAL\ (22::nat)\ (10::nat))\ (DECIMAL\ (22::nat)\ (10::nat))\ (DECIMAL\ (252::nat)\ (100::nat)) < sqrt\ (real\_of\_nat\ (2::nat))$

thm Collect_geom2.THANG_DEU:

$real\_of\_nat\ (2::nat) \leq (?x::real) \longrightarrow (real\_of\_nat\ (2::nat))^2 \leq ?x^2$

thm Collect_geom2.NGAY23_THANG2_09:

$real\_of\_nat\ (2::nat) \leq (?y::real) \wedge ?y \leq sqrt\ (real\_of\_nat\ (8::nat)) \longrightarrow (real\_of\_nat\ (2::nat))^2 \leq ?y * ?y \wedge ?y * ?y \leq real\_of\_nat\ (8::nat)$

thm Collect_geom2.SQRT8_POW2:

$(sqrt\ (real\_of\_nat\ (8::nat)))^2 = real\_of\_nat\ (8::nat)$

thm Collect_geom2.ETA_Y_SQRT8_2_251:

$eta\_y\ (sqrt\ (real\_of\_nat\ (8::nat)))\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat)) < DECIMAL\ (1453::nat)\ (1000::nat)$

thm Collect_geom2.CIRCUMCENTER_FORMULAR:

$\forall (va::(real, 3)\ cart)\ (vb::(real, 3)\ cart)\ vc::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ va\ (INSERT\ vb\ (INSERT\ vc\ EMPTY))) \longrightarrow circumcenter\ (INSERT\ va\ (INSERT\ vb\ (INSERT\ vc\ EMPTY))) = vector\_add\ (\%\ ((d3\ vb\ vc)^2 * ((d3\ va\ vc)^2 + ((d3\ va\ vb)^2 - (d3\ vb\ vc)^2))\ /\ ups\_x\ ((d3\ vb\ vc)^2)\ ((d3\ va\ vc)^2)\ ((d3\ va\ vb)^2))\ va)\ (vector\_add\ (\%\ ((d3\ va\ vc)^2 * ((d3\ vb\ vc)^2 + ((d3\ va\ vb)^2 - (d3\ va\ vc)^2))\ /\ ups\_x\ ((d3\ vb\ vc)^2)\ ((d3\ va\ vc)^2)\ ((d3\ va\ vb)^2))\ vb)\ (\%\ ((d3\ va\ vb)^2 * ((d3\ vb\ vc)^2 + ((d3\ va\ vc)^2 - (d3\ va\ vb)^2))\ /\ ups\_x\ ((d3\ vb\ vc)^2)\ ((d3\ va\ vc)^2)\ ((d3\ va\ vb)^2))\ vc))$

thm Collect_geom2.SUM_UPS_X_1:

$\forall (a::real)\ (b::real)\ c::real.\ (0::real) < ups\_x\ a\ b\ c \longrightarrow c * (b + (a - c))\ /\ ups\_x\ a\ b\ c + (a * (c + (b - a))\ /\ ups\_x\ a\ b\ c + b * (c + (a - b))\ /\ ups\_x\ a\ b\ c) = (1::real)$

thm Collect_geom2.FACTOR_OF_QUADRARTIC:

$\forall (a::real)\ (b::real)\ (c::real)\ x::real.\ a \neq (0::real) \wedge (0::real) \leq b^2 - real\_of\_nat\ (4::nat) * (a * c) \longrightarrow a * x^2 + (b * x + c) = a * ((x - (-\ b + sqrt\ (b^2 - real\_of\_nat\ (4::nat) * (a * c)))\ /\ (real\_of\_nat\ (2::nat) * a)) * (x - (-\ b - sqrt\ (b^2 - real\_of\_nat\ (4::nat) * (a * c)))\ /\ (real\_of\_nat\ (2::nat) * a)))$

thm Collect_geom2.COMPUTE_TO_QUA_POLY:

$DECIMAL\ (2696::nat)\ (1000::nat) \leq (?x::real) \wedge ?x \leq sqrt8 \longrightarrow ?x^2 * ((1::real)\ /\ (eta\_y\ ?x\ (DECIMAL\ (245::nat)\ (100::nat))\ (DECIMAL\ (245::nat)\ (100::nat)))^2 - (1::real)\ /\ (eta\_y\ ?x\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat)))^2) = real\_of\_nat\ (4331842500::nat)\ /\ real\_of\_nat\ (363188227801::nat) * ?x^{4::nat} + (-\ real\_of\_nat\ (45702201::nat)\ /\ real\_of\_nat\ (302530802::nat) * ?x^2 + real\_of\_nat\ (529046001::nat)\ /\ real\_of\_nat\ (2520040000::nat))$

thm Collect_geom2.PHAN_TICH:

$\forall x::real.\ real\_of\_nat\ (4331842500::nat)\ /\ real\_of\_nat\ (363188227801::nat) * ((x^2 - real\_of\_nat\ (488365801::nat)\ /\ real\_of\_nat\ (44090000::nat)) * (x^2 -$

$real\_of\_nat\ (2081667::nat)\ /\ real\_of\_nat\ (1310000::nat))) = real\_of\_nat\ (4331842500::nat)\ /\ real\_of\_nat\ (363188227801::nat) * x^{4::nat} + (-\ real\_of\_nat\ (45702201::nat)\ /\ real\_of\_nat\ (302530802::nat) * x^2 + real\_of\_nat\ (529046001::nat)\ /\ real\_of\_nat\ (2520040000::nat))$

thm Collect_geom2.Q_TR:

$\forall x::real.\ DECIMAL\ (2696::nat)\ (1000::nat) \leq x \wedge x \leq sqrt8 \longrightarrow x^2 * ((1::real)\ /\ (eta\_y\ x\ (DECIMAL\ (245::nat)\ (100::nat))\ (DECIMAL\ (245::nat)\ (100::nat)))^2 - (1::real)\ /\ (eta\_y\ x\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat)))^2) \leq (0::real)$

thm Collect_geom2.SQRT8_LT:

$sqrt\ (real\_of\_nat\ (8::nat)) < real\_of\_nat\ (4::nat) * DECIMAL\ (245::nat)\ (100::nat)$

thm Collect_geom2.IM_UP_POS:

$\forall x::real.\ DECIMAL\ (2696::nat)\ (1000::nat) \leq x \wedge x \leq sqrt8 \longrightarrow (0::real) < ups\_x\ (x * x)\ (DECIMAL\ (245::nat)\ (100::nat) * DECIMAL\ (245::nat)\ (100::nat))\ (DECIMAL\ (245::nat)\ (100::nat) * DECIMAL\ (245::nat)\ (100::nat)) \wedge (0::real) < ups\_x\ (x * x)\ (real\_of\_nat\ (2::nat) * real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat) * DECIMAL\ (251::nat)\ (100::nat))$

thm Collect_geom2.IMP_ETAY_POS:

$\forall x::real.\ DECIMAL\ (2696::nat)\ (1000::nat) \leq x \wedge x \leq sqrt8 \longrightarrow (0::real) < eta\_y\ x\ (DECIMAL\ (245::nat)\ (100::nat))\ (DECIMAL\ (245::nat)\ (100::nat)) \wedge (0::real) < eta\_y\ x\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat))$

thm Collect_geom2.REAL_LE_RDIV_0:

$\forall (a::real)\ b::real.\ (0::real) < b \longrightarrow ((0::real) \leq a\ /\ b) = ((0::real) \leq a)$

thm Collect_geom2.NHSJMDH:

$\forall y::real.\ DECIMAL\ (2696::nat)\ (1000::nat) \leq y \wedge y \leq sqrt8 \longrightarrow eta\_y\ y\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (251::nat)\ (100::nat)) \leq eta\_y\ y\ (DECIMAL\ (245::nat)\ (100::nat))\ (DECIMAL\ (245::nat)\ (100::nat))$

thm Collect_geom2.SQRT8_LE:

$(0::real) \leq sqrt\ (real\_of\_nat\ (8::nat))$

thm Collect_geom2.RELATE_POW2_conjunct0:

$((?a::real) = (0::real)) = (?a^2 = (0::real))$

thm Collect_geom2.RELATE_POW2_conjunct1:

$((0::real) < (?a::real)^2) = ((0::real) < ?a \vee \neg\ (0::real) \leq ?a)$

thm Collect_geom2.RELATE_POW2:

$((?a::real) = (0::real)) = (?a^2 = (0::real)) \wedge ((0::real) < ?a^2) = ((0::real) < ?a \vee \neg\ (0::real) \leq ?a)$

thm Pack1.bp_bdt:

$\forall (a::real)\ b::real.\ (0::real) \leq a \wedge (0::real) \leq b \longrightarrow (a < b) = (a^2 < b^2)$

thm Collect_geom2.POS_IMP_POW2:

$(0::real) \leq (?a::real) \wedge ?a \leq (?b::real) \longrightarrow ?a^2 \leq ?b^2$

thm Collect_geom2.SQRT8_LE_EQ_8_LESS_POW2:

$sqrt\ (real\_of\_nat\ (8::nat)) \leq (?a::real) \longrightarrow real\_of\_nat\ (8::nat) \leq ?a^2$

thm Collect_geom2.MINIMAL_QUADRATIC_POLY:

$\forall (b::real)\ (c::real)\ x::real.\ (real\_of\_nat\ (4::nat) * c - b^2)\ /\ real\_of\_nat\ (4::nat) \leq x^2 + (b * x + c)$

thm Collect_geom2.GREATER_THAN_MID_QUADRATIC_PO:

$\forall (b::real)\ (c::real)\ (x::real)\ x0::real.\ - b\ /\ real\_of\_nat\ (2::nat) \leq x0 \wedge x0 \leq x \longrightarrow x0^2 + (b * x0 + c) \leq x^2 + (b * x + c)$

thm Collect_geom2.SQRT8_TWO_TWO:

$sqrt\ (real\_of\_nat\ (8::nat)) \leq real\_of\_nat\ (2::nat) + real\_of\_nat\ (2::nat)$

thm Collect_geom2.A_POS_DELTA:

$(0::real) < delta\ ((DECIMAL\ (32::nat)\ (10::nat))^2)\ (sqrt8^2)\ ((real\_of\_nat\ (2::nat))^2)\ (sqrt8^2)\ ((real\_of\_nat\ (2::nat))^2)\ ((real\_of\_nat\ (2::nat))^2)$

thm Collect_geom2.MET_LAM_ROI:

$DECIMAL\ (32::nat)\ (10::nat) < sqrt8 + real\_of\_nat\ (2::nat) \wedge DECIMAL\ (32::nat)\ (10::nat) < real\_of\_nat\ (2::nat) + real\_of\_nat\ (2::nat) \wedge sqrt8 < sqrt8 + real\_of\_nat\ (2::nat) \wedge sqrt8 < real\_of\_nat\ (2::nat) + real\_of\_nat\ (2::nat)$

thm Collect_geom2.PROVE_POS_THINGS:

$\forall x::real.\ IN\ x\ (INSERT\ (DECIMAL\ (32::nat)\ (10::nat))\ (INSERT\ sqrt8\ (INSERT\ (real\_of\_nat\ (2::nat))\ (INSERT\ sqrt8\ (INSERT\ (real\_of\_nat\ (2::nat))\ (INSERT\ (real\_of\_nat\ (2::nat))\ EMPTY)))))) \longrightarrow (0::real) \leq x$

thm Collect_geom2.IMP_GT_THAN_TWO:

$\forall (v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ w2::(real, 3)\ cart.\ CARD\ (INSERT\ v1\ (INSERT\ w1\ (INSERT\ v2\ (INSERT\ w2\ EMPTY)))) = (4::nat) \wedge packing\ (INSERT\ v1\ (INSERT\ w1\ (INSERT\ v2\ (INSERT\ w2\ EMPTY)))) \longrightarrow real\_of\_nat\ (2::nat) \leq d3\ w1\ v2 \wedge real\_of\_nat\ (2::nat) \leq d3\ v2\ w2 \wedge real\_of\_nat\ (2::nat) \leq d3\ v1\ w2$

thm Collect_geom2.MET_LAM_ROI_conjunct1:

$DECIMAL\ (32::nat)\ (10::nat) < real\_of\_nat\ (2::nat) + real\_of\_nat\ (2::nat)$

thm Collect_geom2.LEMMA_FOR_PAHFWSI:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = (*4*::*nat*) $\wedge$ *packing* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) $\wedge$ *distance* (*v1*, *v3*) $\leq$ *DECIMAL* (*32*::*nat*) (*10*::*nat*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\leq$ *distance* (*v1*, *v2*) $\wedge$ *distance* (*v2*, *v4*) $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\longrightarrow$ ($\forall$ *x*::*real*. *IN x* (*INSERT* (*DECIMAL* (*32*::*nat*) (*10*::*nat*)) (*INSERT* (*DECIMAL* (*251*::*nat*) (*100*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*DECIMAL* (*251*::*nat*) (*100*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) *EMPTY*)))))) $\longrightarrow$ (*0*::*real*) $\leq$ *x*) $\wedge$ *DECIMAL* (*32*::*nat*) (*10*::*nat*) < *DECIMAL* (*251*::*nat*) (*100*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*32*::*nat*) (*10*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *DECIMAL* (*251*::*nat*) (*100*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ (*0*::*real*) < *delta* ((*DECIMAL* (*32*::*nat*) (*10*::*nat*))^2) ((*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) $\wedge$ *CARD* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = (*4*::*nat*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\leq$ *d3 v1 v2* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 v2 v3* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 v3 v4* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 v1 v4* $\wedge$ *d3 v1 v3* $\leq$ *DECIMAL* (*32*::*nat*) (*10*::*nat*) $\wedge$ *d3 v2 v4* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*)

thm Collect_geom2.LEMMA_OF_39:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT w1* (*INSERT w2 EMPTY*)))) = (*4*::*nat*) $\wedge$ *packing* (*INSERT v1* (*INSERT v2* (*INSERT w1* (*INSERT w2 EMPTY*)))) $\wedge$ *distance* (*w1*, *w2*) $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *distance* (*v1*, *v2*) $\leq$ *DECIMAL* (*307*::*nat*) (*100*::*nat*) $\longrightarrow$ ($\forall$ *x*::*real*. *IN x* (*INSERT* (*DECIMAL* (*251*::*nat*) (*100*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*DECIMAL* (*307*::*nat*) (*100*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) *EMPTY*)))))) $\longrightarrow$ (*0*::*real*) $\leq$ *x*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*307*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ *DECIMAL* (*307*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) $\wedge$ (*0*::*real*) < *delta* ((*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*DECIMAL* (*307*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) $\wedge$ *CARD* (*INSERT w1* (*INSERT v1* (*INSERT w2* (*INSERT v2 EMPTY*)))) = (*4*::*nat*) $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 w1 v1* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 v1 w2* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 w2 v2* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 w1 v2* $\wedge$ *d3 w1 w2* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *d3 v1 v2* $\leq$ *DECIMAL* (*307*::*nat*) (*100*::*nat*)

thm Collect_geom2.LEMMA_OF_LEMMA40:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT w1* (*INSERT w2 EMPTY*)))) = (*4*::*nat*) $\wedge$ *packing* (*INSERT v1* (*INSERT v2* (*INSERT w1* (*INSERT w2*

EMPTY)))) ∧ *distance* (*v1*, *v2*) ≤ *DECIMAL* (*32*::*nat*) (*10*::*nat*) ∧ *distance* (*w1*, *w2*) ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ≤ *distance* (*v1*, *w1*) ⟶ (∀ *x*::*real*. *IN x* (*INSERT* (*DECIMAL* (*32*::*nat*) (*10*::*nat*)) (*INSERT* (*DECIMAL* (*22*::*nat*) (*10*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*DECIMAL* (*251*::*nat*) (*100*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) (*INSERT* (*real_of_nat* (*2*::*nat*)) *EMPTY*)))))) ⟶ (*0*::*real*) ≤ *x*) ∧ *DECIMAL* (*32*::*nat*) (*10*::*nat*) < *DECIMAL* (*22*::*nat*) (*10*::*nat*) + *real_of_nat* (*2*::*nat*) ∧ *DECIMAL* (*32*::*nat*) (*10*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) ∧ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *DECIMAL* (*22*::*nat*) (*10*::*nat*) + *real_of_nat* (*2*::*nat*) ∧ *DECIMAL* (*251*::*nat*) (*100*::*nat*) < *real_of_nat* (*2*::*nat*) + *real_of_nat* (*2*::*nat*) ∧ (*0*::*real*) < *delta* ((*DECIMAL* (*32*::*nat*) (*10*::*nat*))^2) ((*DECIMAL* (*22*::*nat*) (*10*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ∧ *CARD* (*INSERT v1* (*INSERT w1* (*INSERT v2* (*INSERT w2 EMPTY*)))) = (*4*::*nat*) ∧ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ≤ *d3 v1 w1* ∧ *real_of_nat* (*2*::*nat*) ≤ *d3 w1 v2* ∧ *real_of_nat* (*2*::*nat*) ≤ *d3 v2 w2* ∧ *real_of_nat* (*2*::*nat*) ≤ *d3 v1 w2* ∧ *d3 v1 v2* ≤ *DECIMAL* (*32*::*nat*) (*10*::*nat*) ∧ *d3 w1 w2* ≤ *DECIMAL* (*251*::*nat*) (*100*::*nat*)

thm Collect_geom2.LEOF41:

DECIMAL (*3114467*::*nat*) (*1000000*::*nat*) < (*?x*::*real*) ⟶ *delta* ((*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) ((*real_of_nat* (*2*::*nat*))^2) (*?x*^{2}) < (*0*::*real*)

thm Collect_geom2.LEMMA41:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = (*4*::*nat*) ∧ *d3 v1 v2* = *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *d3 v1 v3* = *real_of_nat* (*2*::*nat*) ∧ *d3 v1 v4* = *real_of_nat* (*2*::*nat*) ∧ *d3 v2 v3* = *real_of_nat* (*2*::*nat*) ∧ *d3 v2 v4* = *real_of_nat* (*2*::*nat*) ⟶ *d3 v3 v4* ≤ *DECIMAL* (*3114467*::*nat*) (*1000000*::*nat*)

thm Collect_geom2.LEMMA_OF_L42:

sqrt8 ≤ *d3* (*?v2.0*::(*real*, *3*) *cart*) (*?v4.0*::(*real*, *3*) *cart*) ∧ *DECIMAL* (*3488*::*nat*) (*1000*::*nat*) ≤ (*?x*::*real*) ⟶ − (*1*::*real*) ∗ (*?x*^{2} ∗ (*d3 ?v2.0 ?v4.0*)^2) + (− (*1*::*real*) ∗ *?x*$^{4::nat}$ + (*real_of_nat* (*103001*::*nat*) / *real_of_nat* (*5000*::*nat*) ∗ *?x*^{2} + − *real_of_nat* (*529046001*::*nat*) / *real_of_nat* (*100000000*::*nat*))) < (*0*::*real*)

thm Collect_geom2.LEMMA_IN_LEMMA42_P25:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *x*::*real*. *d3 v1 v2* = *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *d3 v1 v4* = *DECIMAL* (*251*::*nat*) (*100*::*nat*) ∧ *d3 v2 v3* = *real_of_nat* (*2*::*nat*) ∧ *d3 v3 v4* = *real_of_nat* (*2*::*nat*) ∧ *sqrt8* ≤ *d3 v2 v4* ∧ *DECIMAL* (*3488*::*nat*) (*1000*::*nat*) ≤ *x* ⟶ *delta* ((*d3 v1 v2*)^2) (*x*^{2}) ((*d3 v1 v4*)^2) ((*d3 v2 v3*)^2) ((*d3 v2 v4*)^2) ((*d3 v3 v4*)^2) < (*0*::*real*)

thm Collect_geom2.PAATDXJ:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. CARD (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) = (4::nat) ∧ d3 v1 v2 = DECIMAL (251::nat) (100::nat) ∧ d3 v1 v4 = DECIMAL (251::nat) (100::nat) ∧ d3 v2 v3 = real_of_nat (2::nat) ∧ d3 v3 v4 = real_of_nat (2::nat) ∧ sqrt8 ≤ d3 v2 v4 ⟶ d3 v1 v3 < DECIMAL (3488::nat) (1000::nat)

thm Collect_geom2.CONVEX_NORMBALL:

∀ (x::(real, ?'a::type) cart) e::real. convex (normball x e)

thm Collect_geom2.CONVEX_HULL4:

hull convex (INSERT (?v1.0::(real, ?'a::type) cart) (INSERT (?v2.0::(real, ?'a::type) cart) (INSERT (?v3.0::(real, ?'a::type) cart) (INSERT (?v4.0::(real, ?'a::type) cart) EMPTY)))) = GSPEC (λGEN%PVAR%890::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%890 (∃ u::(real, ?'a::type) cart ⇒ real. (∀ x::(real, ?'a::type) cart. IN x (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY)))) ⟶ (0::real) ≤ u x) ∧ sum (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY)))) u = (1::real) ∧ vsum (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY)))) (λx::(real, ?'a::type) cart. % (u x) x) = y) y)

thm Collect_geom2.CONVEX_HULL_4_EQUIV:

∀ (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) v4::(real, ?'a::type) cart. conv (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) = GSPEC (λGEN%PVAR%75::(real, ?'a::type) cart. ∃ x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%75 (∃ (a::real) (b::real) (c::real) d::real. (0::real) ≤ a ∧ (0::real) ≤ b ∧ (0::real) ≤ c ∧ (0::real) ≤ d ∧ a + (b + (c + d)) = (1::real) ∧ vector_add (% a v1) (vector_add (% b v2) (vector_add (% c v3) (% d v4))) = x) x)

thm Collect_geom2.TXDIACY:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) (d::(real, 3) cart) (v0::(real, 3) cart) r::real. SUBSET (INSERT a (INSERT b (INSERT c (INSERT d EMPTY)))) (normball v0 r) ⟶ SUBSET (hull convex (INSERT a (INSERT b (INSERT c (INSERT d EMPTY))))) (normball v0 r)

thm Collect_geom2.LEMMA76:

∃ (t1::(real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ real) (t2::(real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ real) (t3::(real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ real) t4::(real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ real. ∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ t1 v1 v2 v3 v4 v + (t2 v1 v2 v3 v4 v + (t3 v1 v2 v3 v4 v + t4 v1 v2 v3 v4 v)) = (1::real) ∧ v = vector_add (% (t1 v1 v2 v3 v4 v) v1) (vector_add (% (t2 *v1 v2 v3 v4 v) v2*) (*vector_add* (% (*t3 v1 v2 v3 v4 v*) *v3*) (% (*t4 v1 v2 v3 v4 v*) *v4*))) ∧ (∀(*ta*::*real*) (*tb*::*real*) (*tc*::*real*) *td*::*real*. *v* = *vector_add* (% *ta v1*) (*vector_add* (% *tb v2*) (*vector_add* (% *tc v3*) (% *td v4*))) ∧ *ta* + (*tb* + (*tc* + *td*)) = (*1*::*real*) ⟶ *ta* = *t1 v1 v2 v3 v4 v* ∧ *tb* = *t2 v1 v2 v3 v4 v* ∧ *tc* = *t3 v1 v2 v3 v4 v* ∧ *td* = *t4 v1 v2 v3 v4 v*)

thm DEF_COEF4_1:

COEF4_1 = (*SOME t1*::*nat* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀*_2088430*::*nat*. ∃(*t2*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*) (*t3*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*) *t4*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀(*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *t1 _2088430 v1 v2 v3 v4 v* + (*t2 v1 v2 v3 v4 v* + (*t3 v1 v2 v3 v4 v* + *t4 v1 v2 v3 v4 v*)) = (*1*::*real*) ∧ *v* = *vector_add* (% (*t1 _2088430 v1 v2 v3 v4 v*) *v1*) (*vector_add* (% (*t2 v1 v2 v3 v4 v*) *v2*) (*vector_add* (% (*t3 v1 v2 v3 v4 v*) *v3*) (% (*t4 v1 v2 v3 v4 v*) *v4*))) ∧ (∀(*ta*::*real*) (*tb*::*real*) (*tc*::*real*) *td*::*real*. *v* = *vector_add* (% *ta v1*) (*vector_add* (% *tb v2*) (*vector_add* (% *tc v3*) (% *td v4*))) ∧ *ta* + (*tb* + (*tc* + *td*)) = (*1*::*real*) ⟶ *ta* = *t1 _2088430 v1 v2 v3 v4 v* ∧ *tb* = *t2 v1 v2 v3 v4 v* ∧ *tc* = *t3 v1 v2 v3 v4 v* ∧ *td* = *t4 v1 v2 v3 v4 v*)) (*73*::*nat*)

thm DEF_COEF4_2:

COEF4_2 = (*SOME t2*::*nat* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀*_2088431*::*nat*. ∃(*t3*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*) *t4*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀(*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *COEF4_1 v1 v2 v3 v4 v* + (*t2 _2088431 v1 v2 v3 v4 v* + (*t3 v1 v2 v3 v4 v* + *t4 v1 v2 v3 v4 v*)) = (*1*::*real*) ∧ *v* = *vector_add* (% (*COEF4_1 v1 v2 v3 v4 v*) *v1*) (*vector_add* (% (*t2 _2088431 v1 v2 v3 v4 v*) *v2*) (*vector_add* (% (*t3 v1 v2 v3 v4 v*) *v3*) (% (*t4 v1 v2 v3 v4 v*) *v4*))) ∧ (∀(*ta*::*real*) (*tb*::*real*) (*tc*::*real*) *td*::*real*. *v* = *vector_add* (% *ta v1*) (*vector_add* (% *tb v2*) (*vector_add* (% *tc v3*) (% *td v4*))) ∧ *ta* + (*tb* + (*tc* + *td*)) = (*1*::*real*) ⟶ *ta* = *COEF4_1 v1 v2 v3 v4 v* ∧ *tb* = *t2 _2088431 v1 v2 v3 v4 v* ∧ *tc* = *t3 v1 v2 v3 v4 v* ∧ *td* = *t4 v1 v2 v3 v4 v*)) (*74*::*nat*)

thm DEF_COEF4_3:

COEF4_3 = (*SOME t3*::*nat* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀*_2088432*::*nat*. ∃*t4*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* ⇒ *real*. ∀(*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *COEF4_1 v1 v2 v3 v4 v* + (*COEF4_2 v1 v2 v3* v4 v + (t3 _2088432 v1 v2 v3 v4 v + t4 v1 v2 v3 v4 v)) = (1::real) ∧ v = vector_add (% (COEF4_1 v1 v2 v3 v4 v) v1) (vector_add (% (COEF4_2 v1 v2 v3 v4 v) v2) (vector_add (% (t3 _2088432 v1 v2 v3 v4 v) v3) (% (t4 v1 v2 v3 v4 v) v4))) ∧ (∀(ta::real) (tb::real) (tc::real) td::real. v = vector_add (% ta v1) (vector_add (% tb v2) (vector_add (% tc v3) (% td v4))) ∧ ta + (tb + (tc + td)) = (1::real) ⟶ ta = COEF4_1 v1 v2 v3 v4 v ∧ tb = COEF4_2 v1 v2 v3 v4 v ∧ tc = t3 _2088432 v1 v2 v3 v4 v ∧ td = t4 v1 v2 v3 v4 v)) (75::nat)

thm DEF_COEF4_4:

COEF4_4 = (SOME t4::nat ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ (real, 3) cart ⇒ real. ∀(_2088433::nat) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ COEF4_1 v1 v2 v3 v4 v + (COEF4_2 v1 v2 v3 v4 v + (COEF4_3 v1 v2 v3 v4 v + t4 _2088433 v1 v2 v3 v4 v)) = (1::real) ∧ v = vector_add (% (COEF4_1 v1 v2 v3 v4 v) v1) (vector_add (% (COEF4_2 v1 v2 v3 v4 v) v2) (vector_add (% (COEF4_3 v1 v2 v3 v4 v) v3) (% (t4 _2088433 v1 v2 v3 v4 v) v4))) ∧ (∀(ta::real) (tb::real) (tc::real) td::real. v = vector_add (% ta v1) (vector_add (% tb v2) (vector_add (% tc v3) (% td v4))) ∧ ta + (tb + (tc + td)) = (1::real) ⟶ ta = COEF4_1 v1 v2 v3 v4 v ∧ tb = COEF4_2 v1 v2 v3 v4 v ∧ tc = COEF4_3 v1 v2 v3 v4 v ∧ td = t4 _2088433 v1 v2 v3 v4 v)) (76::nat)

thm Collect_geom2.COEFS_4:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ COEF4_1 v1 v2 v3 v4 v + (COEF4_2 v1 v2 v3 v4 v + (COEF4_3 v1 v2 v3 v4 v + COEF4_4 v1 v2 v3 v4 v)) = (1::real) ∧ v = vector_add (% (COEF4_1 v1 v2 v3 v4 v) v1) (vector_add (% (COEF4_2 v1 v2 v3 v4 v) v2) (vector_add (% (COEF4_3 v1 v2 v3 v4 v) v3) (% (COEF4_4 v1 v2 v3 v4 v) v4))) ∧ (∀(ta::real) (tb::real) (tc::real) td::real. v = vector_add (% ta v1) (vector_add (% tb v2) (vector_add (% tc v3) (% td v4))) ∧ ta + (tb + (tc + td)) = (1::real) ⟶ ta = COEF4_1 v1 v2 v3 v4 v ∧ tb = COEF4_2 v1 v2 v3 v4 v ∧ tc = COEF4_3 v1 v2 v3 v4 v ∧ td = COEF4_4 v1 v2 v3 v4 v)

thm Collect_geom2.COEF_1_EQ_ZERO:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ (COEF4_1 v1 v2 v3 v4 v = (0::real)) = IN v (aff (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))))

thm Collect_geom2.EQ_IMP_COPLANAR:

∀(a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. a = b ∨ a = c ∨ a = d ⟶ coplanar_alt (INSERT a (INSERT b (INSERT c (INSERT d EMPTY))))

thm Collect_geom2.THEOREM_RE_AFF_LT31:

∀(v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) (vv::(real, ?'a::type) cart) x::(real, ?'a::type) cart. v1 ≠ vv ∧ v2 ≠ vv ∧ v3 ≠ vv ⟶ (∃f::(real, ?'a::type) cart ⇒ real. f vv < (0::real) ∧ sum (INSERT v1 (INSERT v2 (INSERT v3 (INSERT vv EMPTY)))) f = (1::real) ∧ x = vsum (INSERT v1 (INSERT v2 (INSERT v3 (INSERT vv EMPTY)))) (λv::(real, ?'a::type) cart. % (f v) v)) = GSPEC (λGEN%PVAR%79::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%79 (∃(a::real) (b::real) (c::real) t::real. a + (b + (c + t)) = (1::real) ∧ x = vector_add (% a v1) (vector_add (% b v2) (vector_add (% c v3) (% t vv))) ∧ t < (0::real)) x) x

thm Collect_geom2.AFF_LT31:

∀(v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) vv::(real, ?'a::type) cart. ¬ IN vv (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) ⟶ aff_lt (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT vv EMPTY) = GSPEC (λGEN%PVAR%80::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%80 (∃(a::real) (b::real) (c::real) t::real. t < (0::real) ∧ a + (b + (c + t)) = (1::real) ∧ x = vector_add (% a v1) (vector_add (% b v2) (vector_add (% c v3) (% t vv)))) x)

thm Collect_geom2.AFF_LT21:

∀(a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) v0::(real, ?'a::type) cart. a ≠ v0 ∧ b ≠ v0 ⟶ aff_lt (INSERT a (INSERT b EMPTY)) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%81::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%81 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ t < (0::real) ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x)

thm Collect_geom2.AFF_GT33:

∀(v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) (w1::(real, ?'a::type) cart) (w2::(real, ?'a::type) cart) w3::(real, ?'a::type) cart. HOL_Light_Import.INTER (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT w1 (INSERT w2 (INSERT w3 EMPTY))) = EMPTY ⟶ aff_gt (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT w1 (INSERT w2 (INSERT w3 EMPTY))) = GSPEC (λGEN%PVAR%82::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%82 (∃(a1::real) (a2::real) (a3::real) (b1::real) (b2::real) b3::real. (0::real) < b1 ∧ (0::real) < b2 ∧ (0::real) < b3 ∧ a1 + (a2 + (a3 + (b1 + (b2 + b3)))) = (1::real) ∧ x = vector_add (% a1 v1) (vector_add (% a2 v2) (vector_add (% a3 v3) (vector_add (% b1 w1) (vector_add (% b2 w2) (% b3 w3)))))) x)

thm Collect_geom2.AFF_GE33:

∀(v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) (v3::(real, ?'a::type) cart) (w1::(real, ?'a::type) cart) (w2::(real, ?'a::type) cart) w3::(real, ?'a::type) cart. HOL_Light_Import.INTER (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) (INSERT w1 (INSERT w2 (INSERT w3 EMPTY))) = EMPTY ⟶ aff_ge

(*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT w1* (*INSERT w2* (*INSERT w3 EMPTY*))) = *GSPEC* (λ*GEN%PVAR%83*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%83* (∃ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*b1*::*real*) (*b2*::*real*) *b3*::*real*. (*0*::*real*) ≤ *b1* ∧ (*0*::*real*) ≤ *b2* ∧ (*0*::*real*) ≤ *b3* ∧ *a1* + (*a2* + (*a3* + (*b1* + (*b2* + *b3*)))) = (*1*::*real*) ∧ *x* = *vector_add* (% *a1 v1*) (*vector_add* (% *a2 v2*) (*vector_add* (% *a3 v3*) (*vector_add* (% *b1 w1*) (*vector_add* (% *b2 w2*) (% *b3 w3*)))))) *x*)

thm Collect_geom2.AFF_GE_12:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. ¬ (*v0* = *a* ∨ *v0* = *b*) ⟶ *aff_ge* (*INSERT v0 EMPTY*) (*INSERT a* (*INSERT b EMPTY*)) = *GSPEC* (λ*GEN%PVAR%84*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%84* (∃ (*tv*::*real*) (*ta*::*real*) *tb*::*real*. (*0*::*real*) ≤ *ta* ∧ (*0*::*real*) ≤ *tb* ∧ *tv* + (*ta* + *tb*) = (*1*::*real*) ∧ *x* = *vector_add* (% *tv v0*) (*vector_add* (% *ta a*) (% *tb b*))) *x*)

thm Collect_geom2.INSET3:

IN (*?a*::*?'a*::*type*) (*INSERT ?a* (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*))) ∧ *IN ?b* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*))) ∧ *IN ?c* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*))) ∧ *INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*)) *?a* ∧ *INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*)) *?b* ∧ *INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*)) *?c*

thm Collect_geom2.INSET3_conjunct5:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) *?c*

thm Collect_geom2.INSET3_conjunct4:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) *?b*

thm Collect_geom2.INSET3_conjunct3:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) *?a*

thm Collect_geom2.INSET3_conjunct2:

IN (*?c*::*?'a*::*type*) (*INSERT* (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT ?c EMPTY*)))

thm Collect_geom2.INSET3_conjunct1:

IN (*?b*::*?'a*::*type*) (*INSERT* (*?a*::*?'a*::*type*) (*INSERT ?b* (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)))

thm Collect_geom2.INSET3_conjunct0:

IN (*?a*::*?'a*::*type*) (*INSERT ?a* (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)))

thm Collect_geom2.AFF_LE_LT33:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) (*w1*::(*real*, *?'a*::*type*) *cart*) (*w2*::(*real*, *?'a*::*type*) *cart*) *w3*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.INTER* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT w1* (*INSERT w2* (*INSERT w3 EMPTY*))) = *EMPTY* $\longrightarrow$ *aff_le* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT w1* (*INSERT w2* (*INSERT w3 EMPTY*))) = *GSPEC* (λ*GEN%PVAR%85*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%85* ($\exists$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*b1*::*real*) (*b2*::*real*) *b3*::*real*. *b1* $\leq$ (*0*::*real*) $\wedge$ *b2* $\leq$ (*0*::*real*) $\wedge$ *b3* $\leq$ (*0*::*real*) $\wedge$ *a1* + (*a2* + (*a3* + (*b1* + (*b2* + *b3*)))) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *a1 v1*) (*vector_add* (% *a2 v2*) (*vector_add* (% *a3 v3*) (*vector_add* (% *b1 w1*) (*vector_add* (% *b2 w2*) (% *b3 w3*)))))) *x*) $\wedge$ *aff_lt* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) (*INSERT w1* (*INSERT w2* (*INSERT w3 EMPTY*))) = *GSPEC* (λ*GEN%PVAR%86*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%86* ($\exists$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*b1*::*real*) (*b2*::*real*) *b3*::*real*. *b1* < (*0*::*real*) $\wedge$ *b2* < (*0*::*real*) $\wedge$ *b3* < (*0*::*real*) $\wedge$ *a1* + (*a2* + (*a3* + (*b1* + (*b2* + *b3*)))) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *a1 v1*) (*vector_add* (% *a2 v2*) (*vector_add* (% *a3 v3*) (*vector_add* (% *b1 w1*) (*vector_add* (% *b2 w2*) (% *b3 w3*)))))) *x*)

thm Collect_geom2.AFF_GES_LTS:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *v0* $\wedge$ *b* $\neq$ *v0* $\wedge$ *c* $\neq$ *v0* $\longrightarrow$ *aff_gt* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%87*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%87* ($\exists$ (*ta*::*real*) (*tb*::*real*) *t*::*real*. *ta* + (*tb* + *t*) = (*1*::*real*) $\wedge$ (*0*::*real*) < *t* $\wedge$ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (% *t v0*))) *x*) $\wedge$ *aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%88*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%88* ($\exists$ (*ta*::*real*) (*tb*::*real*) *t*::*real*. *ta* + (*tb* + *t*) = (*1*::*real*) $\wedge$ (*0*::*real*) $\leq$ *t* $\wedge$ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (% *t v0*))) *x*) $\wedge$ *aff_lt* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%89*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%89* ($\exists$ (*ta*::*real*) (*tb*::*real*) (*tc*::*real*) *t*::*real*. *t* < (*0*::*real*) $\wedge$ *ta* + (*tb* + (*tc* + *t*)) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (*vector_add* (% *tc c*) (% *t v0*)))) *x*) $\wedge$ *aff_gt* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%90*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%90* ($\exists$ (*ta*::*real*) (*tb*::*real*) (*tc*::*real*) *t*::*real*. (*0*::*real*) < *t* $\wedge$ *ta* + (*tb* + (*tc* + *t*)) = (*1*::*real*) $\wedge$ *x* = *vector_add* (% *ta a*) (*vector_add* (% *tb b*) (*vector_add* (% *tc c*) (% *t v0*)))) *x*)

thm Collect_geom2.AFF_GES_GTS:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. *a* $\neq$ *v0* $\wedge$ *b* $\neq$ *v0* $\wedge$ *c* $\neq$ *v0* $\longrightarrow$ *aff_gt* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT v0 EMPTY*) = *GSPEC* (λ*GEN%PVAR%91*::(*real*,

?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%91 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ (0::real) < t ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x) ∧ aff_ge (INSERT a (INSERT b EMPTY)) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%92::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%92 (∃(ta::real) (tb::real) t::real. ta + (tb + t) = (1::real) ∧ (0::real) ≤ t ∧ x = vector_add (% ta a) (vector_add (% tb b) (% t v0))) x) ∧ aff_lt (INSERT a (INSERT b (INSERT c EMPTY))) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%93::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%93 (∃(ta::real) (tb::real) (tc::real) t::real. t < (0::real) ∧ ta + (tb + (tc + t)) = (1::real) ∧ x = vector_add (% ta a) (vector_add (% tb b) (vector_add (% tc c) (% t v0)))) x) ∧ aff_gt (INSERT a (INSERT b (INSERT c EMPTY))) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%94::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%94 (∃(ta::real) (tb::real) (tc::real) t::real. (0::real) < t ∧ ta + (tb + (tc + t)) = (1::real) ∧ x = vector_add (% ta a) (vector_add (% tb b) (vector_add (% tc c) (% t v0)))) x) ∧ aff_ge (INSERT a (INSERT b (INSERT c EMPTY))) (INSERT v0 EMPTY) = GSPEC (λGEN%PVAR%95::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%95 (∃(ta::real) (tb::real) (tc::real) t::real. (0::real) ≤ t ∧ ta + (tb + (tc + t)) = (1::real) ∧ x = vector_add (% ta a) (vector_add (% tb b) (vector_add (% tc c) (% t v0)))) x)

thm Collect_geom2.COEF_1_POS_NEG:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ ((0::real) < COEF4_1 v1 v2 v3 v4 v) = IN v (aff_gt (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))) (INSERT v1 EMPTY)) ∧ (COEF4_1 v1 v2 v3 v4 v < (0::real)) = IN v (aff_lt (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))) (INSERT v1 EMPTY))

thm Collect_geom2.ALL_ABOUT_COEF_1:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ (COEF4_1 v1 v2 v3 v4 v < (0::real)) = IN v (aff_lt (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))) (INSERT v1 EMPTY)) ∧ (COEF4_1 v1 v2 v3 v4 v = (0::real)) = IN v (aff (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ∧ ((0::real) < COEF4_1 v1 v2 v3 v4 v) = IN v (aff_gt (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))) (INSERT v1 EMPTY))

thm Collect_geom2.PER_COEF1_WITH_COEF2:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (v4::(real, 3) cart) v::(real, 3) cart. ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ COEF4_2 v1 v2 v3 v4 v = COEF4_1 v2 v3 v4 v1 v

thm Collect_geom2.PER_COEF1_WITH_COEF3:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *COEF4_3 v1 v2 v3 v4 v* = *COEF4_1 v3 v4 v1 v2 v*

thm Collect_geom2.PER_COEF1_WITH_COEF4:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *COEF4_4 v1 v2 v3 v4 v* = *COEF4_1 v4 v1 v2 v3 v*

thm Collect_geom2.ALL_ABOUT_COEF_2:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ (*COEF4_2 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v2 EMPTY*)) ∧ (*COEF4_2 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_2 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v2 EMPTY*))

thm Collect_geom2.ALL_ABOUT_COEF_3:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ (*COEF4_3 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*))) (*INSERT v3 EMPTY*)) ∧ (*COEF4_3 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_3 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*))) (*INSERT v3 EMPTY*))

thm Collect_geom2.LEMMA77:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ (*COEF4_1 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v1 EMPTY*)) ∧ (*COEF4_1 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_1 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v1 EMPTY*)) ∧ (*COEF4_2 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v2 EMPTY*)) ∧ (*COEF4_2 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_2 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v1* (*INSERT v3* (*INSERT v4 EMPTY*))) (*INSERT v2 EMPTY*)) ∧ (*COEF4_3 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*))) (*INSERT v3 EMPTY*)) ∧ (*COEF4_3 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_3 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v2* (*INSERT v1* (*INSERT v4 EMPTY*))) (*INSERT v3 EMPTY*)) ∧ (*COEF4_4 v1 v2 v3 v4 v* < (*0*::*real*)) = *IN v* (*aff_lt* (*INSERT v2* (*INSERT v1* (*INSERT v3 EMPTY*))) (*INSERT* *v4 EMPTY*)) ∧ (*COEF4_4 v1 v2 v3 v4 v* = (*0*::*real*)) = *IN v* (*aff* (*INSERT v2* (*INSERT v1* (*INSERT v3 EMPTY*)))) ∧ ((*0*::*real*) < *COEF4_4 v1 v2 v3 v4 v*) = *IN v* (*aff_gt* (*INSERT v2* (*INSERT v1* (*INSERT v3 EMPTY*))) (*INSERT v4 EMPTY*))

thm Collect_geom2.ARIKWRQ:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *LET* (λ*s*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* (*CARD s* = (*4*::*nat*) ∧ ¬ *coplanar_alt s* ⟶ *conv s* = *HOL_Light_Import.INTER* (*aff_ge* (*DIFF s* (*INSERT v1 EMPTY*)) (*INSERT v1 EMPTY*)) (*HOL_Light_Import.INTER* (*aff_ge* (*DIFF s* (*INSERT v2 EMPTY*)) (*INSERT v2 EMPTY*)) (*HOL_Light_Import.INTER* (*aff_ge* (*DIFF s* (*INSERT v3 EMPTY*)) (*INSERT v3 EMPTY*)) (*aff_ge* (*DIFF s* (*INSERT v4 EMPTY*)) (*INSERT v4 EMPTY*)))))) (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))

thm Collect_geom2.MXHKOXR:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *LET* (λ*s*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* (*CARD s* = (*4*::*nat*) ∧ ¬ *coplanar_alt s* ⟶ *conv0 s* = *HOL_Light_Import.INTER* (*aff_gt* (*DIFF s* (*INSERT v1 EMPTY*)) (*INSERT v1 EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*DIFF s* (*INSERT v2 EMPTY*)) (*INSERT v2 EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*DIFF s* (*INSERT v3 EMPTY*)) (*INSERT v3 EMPTY*)) (*aff_gt* (*DIFF s* (*INSERT v4 EMPTY*)) (*INSERT v4 EMPTY*)))))) (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))

thm Collect_geom2.CONV0_4POINTS:

conv0 (*INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v3.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v4.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)))) = *GSPEC* (λ*GEN%PVAR%97*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%97* (∃ (*a*::*real*) (*b*::*real*) (*c*::*real*) *d*::*real*. (*0*::*real*) < *a* ∧ (*0*::*real*) < *b* ∧ (*0*::*real*) < *c* ∧ (*0*::*real*) < *d* ∧ *a* + (*b* + (*c* + *d*)) = (*1*::*real*) ∧ *vector_add* (% *a ?v1.0*) (*vector_add* (% *b ?v2.0*) (*vector_add* (% *c ?v3.0*) (% *d ?v4.0*))) = *x*) *x*)

thm Collect_geom2.ZRFMKPY:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v4*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *IN v* (*conv* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) = ((*0*::*real*) ≤ *COEF4_1 v1 v2 v3 v4 v* ∧ (*0*::*real*) ≤ *COEF4_2 v1 v2 v3 v4 v* ∧ (*0*::*real*) ≤ *COEF4_3 v1 v2 v3 v4 v* ∧ (*0*::*real*) ≤ *COEF4_4 v1 v2 v3 v4 v*) ∧ *IN v* (*conv0* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) = ((*0*::*real*) < *COEF4_1 v1 v2 v3 v4 v* ∧ (*0*::*real*) < *COEF4_2 v1 v2 v3 v4 v* ∧ (*0*::*real*) < *COEF4_3 v1 v2 v3 v4 v* ∧ (*0*::*real*) < *COEF4_4 v1 v2 v3 v4 v*)

thm Collect_geom2.QUAANG_TRUOONN:

$\forall$ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) *v4*::(*real*, *?'a*::*type*) *cart*. *CARD* (*INSERT v0* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) = (*5*::*nat*) $\wedge$ ($\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *x* $\neq$ *v0* $\wedge$ *IN x* (*HOL_Light_Import.INTER* (*cone v0* (*INSERT v1* (*INSERT v3 EMPTY*))) (*cone v0* (*INSERT v2* (*INSERT v4 EMPTY*))))) $\longrightarrow$ *HOL_Light_Import.INTER* (*conv* (*INSERT v1* (*INSERT v3 EMPTY*))) (*cone v0* (*INSERT v2* (*INSERT v4 EMPTY*))) $\neq$ *EMPTY*

thm Collect_geom2.JVDAFRS:

$\forall$ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) *v4*::(*real*, *?'a*::*type*) *cart*. *CARD* (*INSERT v0* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) = (*5*::*nat*) $\wedge$ ($\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *x* $\neq$ *v0* $\wedge$ *IN x* (*HOL_Light_Import.INTER* (*cone v0* (*INSERT v1* (*INSERT v3 EMPTY*))) (*cone v0* (*INSERT v2* (*INSERT v4 EMPTY*))))) $\longrightarrow$ $\neg$ (*HOL_Light_Import.INTER* (*conv* (*INSERT v1* (*INSERT v3 EMPTY*))) (*cone v0* (*INSERT v2* (*INSERT v4 EMPTY*))) = *EMPTY* $\wedge$ *HOL_Light_Import.INTER* (*conv* (*INSERT v2* (*INSERT v4 EMPTY*))) (*cone v0* (*INSERT v1* (*INSERT v3 EMPTY*))) = *EMPTY*)

thm Collect_geom2.SQRT8_POS:

(*0*::*real*) < *sqrt* (*real_of_nat* (*8*::*nat*))

thm Collect_geom2.SQRT8_LT_4_45:

sqrt8 < *DECIMAL* (*445*::*nat*) (*100*::*nat*)

thm Collect_geom2.PROVE_NOT_COLLINEAR:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. *real_of_nat* (*2*::*nat*) $\leq$ *d3 v0 v1* $\wedge$ *d3 v0 v1* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*245*::*nat*) (*100*::*nat*) $\leq$ *d3 v1 v2* $\wedge$ *d3 v1 v2* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*277*::*nat*) (*100*::*nat*) $\leq$ *d3 v0 v2* $\wedge$ *d3 v0 v2* $\leq$ *sqrt8* $\longrightarrow$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Collect_geom2.BPOW8APOW2CPOW2:

real_of_nat (*2*::*nat*) $\leq$ (*?a*::*real*) $\wedge$ *?a* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*245*::*nat*) (*100*::*nat*) $\leq$ (*?c*::*real*) $\wedge$ *?c* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*277*::*nat*) (*100*::*nat*) $\leq$ (*?b*::*real*) $\wedge$ *?b* $\leq$ *sqrt8* $\longrightarrow$ $?b^2$ $\leq$ *real_of_nat* (*8*::*nat*) $\wedge$ $?a^2$ $\leq$ (*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2 $\wedge$ $?c^2$ $\leq$ (*DECIMAL* (*251*::*nat*) (*100*::*nat*))^2

thm Collect_geom2.IMP_PRE_LE_19:

real_of_nat (*2*::*nat*) $\leq$ (*?a*::*real*) $\wedge$ *?a* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*245*::*nat*) (*100*::*nat*) $\leq$ (*?c*::*real*) $\wedge$ *?c* $\leq$ *DECIMAL* (*251*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*277*::*nat*) (*100*::*nat*) $\leq$ (*?b*::*real*) $\wedge$ *?b* $\leq$ *sqrt8* $\longrightarrow$ (*0*::*real*) < *real_of_nat* (*2*::*nat*) $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *?a* $\wedge$ (*0*::*real*) < *DECIMAL* (*277*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*277*::*nat*) (*100*::*nat*) $\leq$ *?b* $\wedge$ (*0*::*real*) < *DECIMAL* (*245*::*nat*) (*100*::*nat*) $\wedge$ *DECIMAL* (*245*::*nat*) (*100*::*nat*)

$\leq$ *?c* $\wedge$ *?a*^{2} $\leq$ (*DECIMAL* (*277*::*nat*) (*100*::*nat*))^2 + (*DECIMAL* (*245*::*nat*) (*100*::*nat*))^2 $\wedge$ *?b*^{2} $\leq$ (*real_of_nat* (*2*::*nat*))^2 + (*DECIMAL* (*245*::*nat*) (*100*::*nat*))^2 $\wedge$ *?c*^{2} $\leq$ (*real_of_nat* (*2*::*nat*))^2 + (*DECIMAL* (*277*::*nat*) (*100*::*nat*))^2

thm DEF_condC:

condC = (λ(*_2104467*::*real*) (*_2104468*::*real*) (*_2104469*::*real*) (*_2104470*::*real*) (*_2104471*::*real*) *_2104472*::*real*. ($\forall$ *x*::*real*. *IN x* (*INSERT _2104467* (*INSERT _2104468* (*INSERT _2104469* (*INSERT _2104470* (*INSERT _2104471* (*INSERT _2104472 EMPTY*)))))) $\longrightarrow$ (*0*::*real*) $\leq$ *x*) $\wedge$ *_2104467* $\leq$ *_2104468* + *_2104472* $\wedge$ *_2104467* $\leq$ *_2104469* + *_2104471* $\wedge$ *_2104470* < *_2104468* + *_2104469* $\wedge$ *_2104470* < *_2104472* + *_2104471* $\wedge$ (*0*::*real*) $\leq$ *delta* (*_2104467*^{2}) (*_2104468*^{2}) (*_2104469*^{2}) (*_2104470*^{2}) (*_2104471*^{2}) (*_2104472*^{2}))

thm Collect_geom2.condC:

$\forall$ (*M13*::*real*) (*m12*::*real*) (*m14*::*real*) (*M24*::*real*) (*m34*::*real*) *m23*::*real*. *condC M13 m12 m14 M24 m34 m23* = (($\forall$ *x*::*real*. *IN x* (*INSERT M13* (*INSERT m12* (*INSERT m14* (*INSERT M24* (*INSERT m34* (*INSERT m23 EMPTY*)))))) $\longrightarrow$ (*0*::*real*) $\leq$ *x*) $\wedge$ *M13* $\leq$ *m12* + *m23* $\wedge$ *M13* $\leq$ *m14* + *m34* $\wedge$ *M24* < *m12* + *m14* $\wedge$ *M24* < *m23* + *m34* $\wedge$ (*0*::*real*) $\leq$ *delta* (*M13*^{2}) (*m12*^{2}) (*m14*^{2}) (*M24*^{2}) (*m34*^{2}) (*m23*^{2}))

thm Collect_geom2.LT_SQ8_IMP_LT2:

(*?a*::*real*) < *sqrt* (*real_of_nat* (*8*::*nat*)) $\longrightarrow$ $\neg$ *real_of_nat* (*2*::*nat*) $\leq$ (*1*::*real*) / *real_of_nat* (*2*::*nat*) * *?a*

thm Collect_geom2.LE_FOR_LEMMA36:

(*CARD* (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*))) = (*3*::*nat*) $\wedge$ *packing* (*INSERT ?u* (*INSERT ?v* (*INSERT ?w EMPTY*))) $\wedge$ *distance* (*?u*, *?v*) < *sqrt8*) $\wedge$ $\neg$ *distance* (*?u*, *?v*) / *real_of_nat* (*2*::*nat*) < *distance* (*?w*, % ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add ?u ?v*)) $\longrightarrow$ *condC* (*sqrt* (*real_of_nat* (*8*::*nat*))) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*sqrt* (*real_of_nat* (*8*::*nat*))) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) $\wedge$ *CARD* (*INSERT ?u* (*INSERT ?w* (*INSERT ?v* (*INSERT* (*vector_add ?u* (*vector_sub ?v ?w*)) *EMPTY*)))) = (*4*::*nat*) $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 ?u ?w* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 ?w ?v* $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 ?v* (*vector_add ?u* (*vector_sub ?v ?w*)) $\wedge$ *real_of_nat* (*2*::*nat*) $\leq$ *d3 ?u* (*vector_add ?u* (*vector_sub ?v ?w*)) $\wedge$ *d3 ?u ?v* < *sqrt* (*real_of_nat* (*8*::*nat*)) $\wedge$ *d3 ?w* (*vector_add ?u* (*vector_sub ?v ?w*)) $\leq$ *sqrt* (*real_of_nat* (*8*::*nat*))

thm Collect_geom2.MIDDLE_POINT_IN_CONV2:

IN (% ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) (*vector_add* (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*))) (*conv* (*INSERT ?u* (*INSERT ?v EMPTY*)))

thm Collect_geom2.INTER_DISJONT_EX:

(*HOL_Light_Import.INTER* (*?a*::*?'a*::*type* $\Rightarrow$ *bool*) (*?b*::*?'a*::*type* $\Rightarrow$ *bool*) = *EMPTY*) = ($\forall$ *x*::*?'a*::*type*. $\neg$ (*IN x ?a* $\wedge$ *IN x ?b*))

thm Collect_geom2.PLANE_IMP_AFFINE:

plane (?p::(real, ?'a::type) cart ⇒ bool) ⟶ affine ?p

thm Collect_geom2.IMP_AFFINE_HULL_SUBSET:

FINITE (?a::(real, ?'a::type) cart ⇒ bool) ∧ SUBSET ?a (?s::(real, ?'a::type) cart ⇒ bool) ∧ ?a ≠ EMPTY ∧ affine ?s ⟶ SUBSET (hull affine ?a) ?s

thm Collect_geom2.SET_EQ_EX:

((?a::?'a::type ⇒ bool) = (?b::?'a::type ⇒ bool)) = (∀ x::?'a::type. IN x ?a = IN x ?b)

thm Collect_geom2.SET_EQ_TO_SUBSET:

((?a::?'a::type ⇒ bool) = (?b::?'a::type ⇒ bool)) = (SUBSET ?a ?b ∧ SUBSET ?b ?a)

thm Collect_geom2.OTHORGONAL_QUATER_FOR:

*delta (?x12.0::real) (?x12.0 + (?x23.0::real)) (?x12.0 + (?x24.0::real)) ?x23.0 ?x24.0 (?x34.0::real) = ?x12.0 * ups_x ?x23.0 ?x24.0 ?x34.0*

thm Collect_geom2.ORTHOGONAL_CROSS_PRODUCT:

dot (?u::(real, 3) cart) (cross ?u (?v::(real, 3) cart)) = (0::real) ∧ dot ?v (cross ?u ?v) = (0::real)

thm Collect_geom2.ORTHOGONAL_CROSS_PRODUCT_conjunct1:

dot (?v::(real, 3) cart) (cross (?u::(real, 3) cart) ?v) = (0::real)

thm Collect_geom2.ORTHOGONAL_CROSS_PRODUCT_conjunct0:

dot (?u::(real, 3) cart) (cross ?u (?v::(real, 3) cart)) = (0::real)

thm Collect_geom2.PITHAGOR_CROSS:

(distance (vector_add (?a::(real, 3) cart) (cross (vector_sub (?b::(real, 3) cart) ?a) (vector_sub (?c::(real, 3) cart) ?a)), ?b))^2 = *(distance (?b, ?a))*^{2} + *(vector_norm (cross (vector_sub ?b ?a) (vector_sub ?c ?a)))*^{2}

thm Collect_geom2.PITHAGOR_NORM:

dot (?a::(real, ?'a::type) cart) (?b::(real, ?'a::type) cart) = (0::real) ⟶ (distance (?a, ?b))^2 = *(vector_norm ?a)*^{2} + *(vector_norm ?b)*^{2}

thm Collect_geom2.VEC3_EQ_EX:

∀ (a::(real, 3) cart) b::(real, 3) cart. (a = b) = ($ a (1::nat) = $ b (1::nat) ∧ $ a (2::nat) = $ b (2::nat) ∧ $ a (3::nat) = $ b (3::nat))

thm Collect_geom2.CROSS_CONVERT:

cross (vector_sub (?b::(real, 3) cart) (?a::(real, 3) cart)) (vector_sub (?c::(real, 3) cart) (?d::(real, 3) cart)) = cross (vector_sub ?a ?b) (vector_sub ?d ?c)

thm Collect_geom2.NORM_CROSS_PRODUCT_UPS_X:

real_of_nat (*4*::*nat*) ∗ (*vector_norm* (*cross* (*vector_sub* (*?b*::(*real*, *3*) *cart*) (*?a*::(*real*, *3*) *cart*)) (*vector_sub* (*?c*::(*real*, *3*) *cart*) *?a*)))^2 = *ups_x* ((*distance* (*?a*, *?b*))^2) ((*distance* (*?a*, *?c*))^2) ((*distance* (*?b*, *?c*))^2)

thm Collect_geom2.NOT_COLLINEAR_IMP_CROSS_NOT_COPLANAR:

∀ (*u*::*?'b*::*type*) (*v*::*?'a*::*type*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* (*?a*::(*real*, *3*) *cart*) (*INSERT* (*?b*::(*real*, *3*) *cart*) (*INSERT* (*?c*::(*real*, *3*) *cart*) *EMPTY*))) ⟶ ¬ *coplanar_alt* (*INSERT* (*vector_add* *?a* (*cross* (*vector_sub* *?b* *?a*) (*vector_sub* *?c* *?a*))) (*INSERT* *?a* (*INSERT* *?b* (*INSERT* *?c* *EMPTY*))))

thm Collect_geom2.ORTHOGONAL_IMP_PITHAGOR:

dot (*?x*::(*real*, *?'a*::*type*) *cart*) (*vector_sub* (*?a*::(*real*, *?'a*::*type*) *cart*) (*?b*::(*real*, *?'a*::*type*) *cart*)) = (*0*::*real*) ⟶ (*distance* (*vector_add* *?a* *?x*, *?b*))^2 = (*vector_norm* *?x*)^2 + (*distance* (*?a*, *?b*))^2

thm Collect_geom2.NOT_COL_AND_ORTHO_IMP_NOT_COPL:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* *a* (*INSERT* *b* (*INSERT* *c* *EMPTY*))) ∧ *dot* *x* (*vector_sub* *a* *b*) = (*0*::*real*) ∧ *dot* *x* (*vector_sub* *a* *c*) = (*0*::*real*) ∧ *x* ≠ *vec* (*0*::*nat*) ⟶ ¬ *coplanar_alt* (*INSERT* (*vector_add* *a* *x*) (*INSERT* *a* (*INSERT* *b* (*INSERT* *c* *EMPTY*))))

thm Collect_geom2.PLANE_NORM_IMP_AFFINE:

∀ *p*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *plane_norm* *p* ⟶ *affine* *p*

thm Collect_geom2.IN_PLANE_IMP_OTHORGONAL:

dot (*?n*::(*real*, *?'a*::*type*) *cart*) (*vector_sub* (*?x*::(*real*, *?'a*::*type*) *cart*) (*?v0.0*::(*real*, *?'a*::*type*) *cart*)) = (*0*::*real*) ∧ *dot* *?n* (*vector_sub* (*?y*::(*real*, *?'a*::*type*) *cart*) *?v0.0*) = (*0*::*real*) ∧ *dot* *?n* (*vector_sub* (*?z*::(*real*, *?'a*::*type*) *cart*) *?v0.0*) = (*0*::*real*) ⟶ *dot* *?n* (*vector_sub* *?x* *?y*) = (*0*::*real*) ∧ *dot* *?n* (*vector_sub* *?x* *?z*) = (*0*::*real*)

thm Collect_geom2.IMP_A1_EQ_0:

dot (*?n*::(*real*, *?'a*::*type*) *cart*) (*vector_sub* (*?x*::(*real*, *?'a*::*type*) *cart*) (*?v0.0*::(*real*, *?'a*::*type*) *cart*)) = (*0*::*real*) ∧ *dot* *?n* (*vector_sub* (*?y*::(*real*, *?'a*::*type*) *cart*) *?v0.0*) = (*0*::*real*) ∧ *dot* *?n* (*vector_sub* (*?z*::(*real*, *?'a*::*type*) *cart*) *?v0.0*) = (*0*::*real*) ∧ *?n* ≠ *vec* (*0*::*nat*) ∧ (*?x'*::(*real*, *?'a*::*type*) *cart*) = *vector_add* (% (*?a1.0*::*real*) (*vector_add* *?x* *?n*)) (*vector_add* (% (*?a2.0*::*real*) *?x*) (*vector_add* (% (*?a3.0*::*real*) *?y*) (% (*?a4.0*::*real*) *?z*))) ∧ *?a1.0* + (*?a2.0* + (*?a3.0* + *?a4.0*)) = (*1*::*real*) ∧ *dot* *?n* (*vector_sub* *?x'* *?v0.0*) = (*0*::*real*) ⟶ *?a1.0* = (*0*::*real*)

thm Collect_geom2.SMWTDMU:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart* ⇒ *bool*. *plane_norm* *p* ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* *z* *EMPTY*)))

∧ *SUBSET* (*INSERT* *x* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) *p* ⟶ *p* = *aff* (*INSERT* *x* (*INSERT* *y* (*INSERT* *z* *EMPTY*)))

thm Collect_geom2.DET_VEC3_AS_CROSS_DOT:

det_vec3 (*?v1.0*::(*real*, *3*) *cart*) (*?v2.0*::(*real*, *3*) *cart*) (*?v3.0*::(*real*, *3*) *cart*) = *dot* (*cross* *?v1.0* *?v2.0*) *?v3.0*

thm Collect_geom2.COL_EQ_NORM_CROSS:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *collinear* (*INSERT* *v1* (*INSERT* *v2* (*INSERT* *v3* *EMPTY*))) = ((*vector_norm* (*cross* (*vector_sub* *v2* *v1*) (*vector_sub* *v3* *v1*)))^2 = (*0*::*real*))

thm Collect_geom2.COLLINEAR_IMP_COPLANAR:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::*?'a*::*type*) (*v3a*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *collinear* (*INSERT* *v1* (*INSERT* *v2* (*INSERT* *v3a* *EMPTY*))) ⟶ *coplanar_alt* (*INSERT* *v1* (*INSERT* *v2* (*INSERT* *v3a* (*INSERT* *v* *EMPTY*))))

thm Collect_geom2.POS_EQ_NOT_COPLANANR:

((*0*::*real*) < *delta* ((*distance* (*?x1.0*::(*real*, *3*) *cart*, *?x2.0*::(*real*, *3*) *cart*))^2) ((*distance* (*?x1.0*, *?x3.0*::(*real*, *3*) *cart*))^2) ((*distance* (*?x1.0*, *?x4.0*::(*real*, *3*) *cart*))^2) ((*distance* (*?x2.0*, *?x3.0*))^2) ((*distance* (*?x2.0*, *?x4.0*))^2) ((*distance* (*?x3.0*, *?x4.0*))^2)) = (¬ *coplanar_alt* (*INSERT* *?x1.0* (*INSERT* *?x2.0* (*INSERT* *?x3.0* (*INSERT* *?x4.0* *EMPTY*)))))

thm Collect_geom2.SUM_CHI_EQ_2DELTA:

LET (λ*chi11*::*real*. *LET_END* (*LET* (λ*chi22*::*real*. *LET_END* (*LET* (λ*chi33*::*real*. *LET_END* (*LET* (λ*chi44*::*real*. *LET_END* (*real_of_nat* (*2*::*nat*) ∗ *delta* (*?x12.0*::*real*) (*?x13.0*::*real*) (*?x14.0*::*real*) (*?x23.0*::*real*) (*?x24.0*::*real*) (*?x34.0*::*real*) = *chi11* + (*chi22* + (*chi33* + *chi44*)))) (*chi* *?x34.0* *?x24.0* *?x14.0* *?x23.0* *?x13.0* *?x12.0*))) (*chi* *?x34.0* *?x13.0* *?x23.0* *?x14.0* *?x24.0* *?x12.0*))) (*chi* *?x12.0* *?x24.0* *?x23.0* *?x14.0* *?x13.0* *?x34.0*))) (*chi* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0*)

thm Collect_geom2.NOT_0_IMP_SUM_CHI_1:

delta (*?x12.0*::*real*) (*?x13.0*::*real*) (*?x14.0*::*real*) (*?x23.0*::*real*) (*?x24.0*::*real*) (*?x34.0*::*real*) ≠ (*0*::*real*) ⟶ *chi* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0* / (*real_of_nat* (*2*::*nat*) ∗ *delta* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0*) + (*chi* *?x12.0* *?x24.0* *?x23.0* *?x14.0* *?x13.0* *?x34.0* / (*real_of_nat* (*2*::*nat*) ∗ *delta* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0*) + (*chi* *?x34.0* *?x13.0* *?x23.0* *?x14.0* *?x24.0* *?x12.0* / (*real_of_nat* (*2*::*nat*) ∗ *delta* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0*) + *chi* *?x34.0* *?x24.0* *?x14.0* *?x23.0* *?x13.0* *?x12.0* / (*real_of_nat* (*2*::*nat*) ∗ *delta* *?x12.0* *?x13.0* *?x14.0* *?x23.0* *?x24.0* *?x34.0*))) = (*1*::*real*)

thm Collect_geom2.PROVE_DIST_FROM_V1:

¬ *coplanar_alt* (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) (*INSERT* (*?v3.0*::(*real*, *3*) *cart*) (*INSERT* (*?v4.0*::(*real*, *3*) *cart*) *EMPTY*))))

⟶ LET (λx12::real. LET_END (LET (λx13::real. LET_END (LET (λx14::real. LET_END (LET (λx23::real. LET_END (LET (λx24::real. LET_END (LET (λx34::real. LET_END (LET (λchi11::real. LET_END (LET (λchi22::real. LET_END (LET (λchi33::real. LET_END (LET (λchi44::real. LET_END ((?p::(real, 3) cart) = % ((1::real) / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)) (vector_add (% chi11 ?v1.0) (vector_add (% chi22 ?v2.0) (vector_add (% chi33 ?v3.0) (% chi44 ?v4.0)))) ⟶ (d3 ?p ?v1.0)^2 = (1::real) / real_of_nat (2::nat) ∗ (rho_ij x12 x13 x14 x23 x24 x34 / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)))) (chi x34 x24 x14 x23 x13 x12))) (chi x34 x13 x23 x14 x24 x12))) (chi x12 x24 x23 x14 x13 x34))) (chi x12 x13 x14 x23 x24 x34))) ((distance (?v3.0, ?v4.0))^2))) ((distance (?v2.0, ?v4.0))^2))) ((distance (?v2.0, ?v3.0))^2))) ((distance (?v1.0, ?v4.0))^2))) ((distance (?v1.0, ?v3.0))^2))) ((distance (?v1.0, ?v2.0))^2)

thm Collect_geom2.PROVE_EQ_DIST_FROM4:

¬ coplanar_alt (INSERT (?v1.0::(real, 3) cart) (INSERT (?v2.0::(real, 3) cart) (INSERT (?v3.0::(real, 3) cart) (INSERT (?v4.0::(real, 3) cart) EMPTY)))) ⟶ LET (λx12::real. LET_END (LET (λx13::real. LET_END (LET (λx14::real. LET_END (LET (λx23::real. LET_END (LET (λx24::real. LET_END (LET (λx34::real. LET_END (LET (λchi11::real. LET_END (LET (λchi22::real. LET_END (LET (λchi33::real. LET_END (LET (λchi44::real. LET_END ((?p::(real, 3) cart) = % ((1::real) / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)) (vector_add (% chi11 ?v1.0) (vector_add (% chi22 ?v2.0) (vector_add (% chi33 ?v3.0) (% chi44 ?v4.0)))) ⟶ (d3 ?p ?v2.0)^2 = (1::real) / real_of_nat (2::nat) ∗ (rho_ij x12 x13 x14 x23 x24 x34 / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)) ∧ (d3 ?p ?v3.0)^2 = (1::real) / real_of_nat (2::nat) ∗ (rho_ij x12 x13 x14 x23 x24 x34 / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)) ∧ (d3 ?p ?v4.0)^2 = (1::real) / real_of_nat (2::nat) ∗ (rho_ij x12 x13 x14 x23 x24 x34 / (real_of_nat (2::nat) ∗ delta x12 x13 x14 x23 x24 x34)))) (chi x34 x24 x14 x23 x13 x12))) (chi x34 x13 x23 x14 x24 x12))) (chi x12 x24 x23 x14 x13 x34))) (chi x12 x13 x14 x23 x24 x34))) ((distance (?v3.0, ?v4.0))^2))) ((distance (?v2.0, ?v4.0))^2))) ((distance (?v2.0, ?v3.0))^2))) ((distance (?v1.0, ?v4.0))^2))) ((distance (?v1.0, ?v3.0))^2))) ((distance (?v1.0, ?v2.0))^2)

thm Collect_geom2.AFFINE_HULL_3:

hull affine (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) EMPTY))) = GSPEC (λGEN%PVAR%98::(real, ?'a::type) cart. ∃(u::real) (v::real) w::real. SETSPEC GEN%PVAR%98 (u + (v + w) = (1::real)) (vector_add (% u ?a) (vector_add (% v ?b) (% w ?c))))

thm Collect_geom2.AFFINE_HULL_4:

hull affine (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) (INSERT (?d::(real, ?'a::type) cart) EMPTY)))) = GSPEC (λGEN%PVAR%99::(real, ?'a::type) cart. ∃(u::real) (v::real) (w::real) z::real. SETSPEC GEN%PVAR%99 (u + (v + (w + z)) =

(1::real)) (vector_add (% u ?a) (vector_add (% v ?b) (vector_add (% w ?c) (% z ?d)))))

thm FORALL_IN_CLAUSES_conjunct1:

∀ (P::?'a::type ⇒ bool) (a::?'a::type) s::?'a::type ⇒ bool. (∀ x::?'a::type. IN x (INSERT a s) ⟶ P x) = (P a ∧ (∀ x::?'a::type. IN x s ⟶ P x))

thm FORALL_IN_CLAUSES_conjunct0:

∀ P::?'a::type ⇒ bool. (∀ x::?'a::type. IN x EMPTY ⟶ P x) = True

thm Collect_geom2.PROVE_EXISTS_CIR_OF_FOUR_POINTS:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. CARD (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) = (4::nat) ∧ ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ (∃ p::(real, 3) cart. IN p (hull affine (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))))) ∧ (∃ r::real. ∀ v::(real, 3) cart. IN v (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ r = distance (p, v)))

thm Collect_geom2.NOT_COPLANAR_IMP_EXISTS_CIR:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. CARD (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) = (4::nat) ∧ ¬ coplanar_alt (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ IN (circumcenter (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))))) (hull affine (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY))))) ∧ (∃ r::real. ∀ v::(real, 3) cart. IN v (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))) ⟶ r = distance (circumcenter (INSERT v1 (INSERT v2 (INSERT v3 (INSERT v4 EMPTY)))), v))

thm Collect_geom2.DIST_EQ_IMP_ORTHOGONAL:

distance (?pp::(real, ?'a::type) cart, ?v2.0::(real, ?'a::type) cart) = distance (?pp, ?v1.0::(real, ?'a::type) cart) ∧ distance (?p::(real, ?'a::type) cart, ?v2.0) = distance (?p, ?v1.0) ⟶ dot (vector_sub ?pp ?p) (vector_sub ?v2.0 ?v1.0) = (0::real)

thm Collect_geom2.IMP_OTHO4:

dot (?n::(real, ?'a::type) cart) (vector_sub (?v2.0::(real, ?'a::type) cart) (?v1.0::(real, ?'a::type) cart)) = (0::real) ∧ dot ?n (vector_sub (?v3.0::(real, ?'a::type) cart) ?v1.0) = (0::real) ∧ dot ?n (vector_sub (?v4.0::(real, ?'a::type) cart) ?v1.0) = (0::real) ∧ IN (?x::(real, ?'a::type) cart) (hull affine (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY))))) ∧ IN (?y::(real, ?'a::type) cart) (hull affine (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY))))) ⟶ dot ?n (vector_sub ?x ?y) = (0::real)

thm Collect_geom2.UNIQUE_EXISISTING_PROPERTY_C4:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = (*4*::*nat*) ∧ ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ (∀ *p*::(*real*, *3*) *cart*. *IN p* (*hull affine* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) ∧ (∃ *r*::*real*. ∀ *v*::(*real*, *3*) *cart*. *IN v* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *r* = *distance* (*p*, *v*)) ⟶ *p* = *circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))))

thm Collect_geom2.PROVE_IN_AFFINE_HULL_4:

delta (*?x12.0*::*real*) (*?x13.0*::*real*) (*?x14.0*::*real*) (*?x23.0*::*real*) (*?x24.0*::*real*) (*?x34.0*::*real*) ≠ (*0*::*real*) ⟶ *IN* (% ((*1*::*real*) / (*real_of_nat* (*2*::*nat*) ∗ *delta ?x12.0 ?x13.0 ?x14.0 ?x23.0 ?x24.0 ?x34.0*)) (*vector_add* (% (*chi ?x12.0 ?x13.0 ?x14.0 ?x23.0 ?x24.0 ?x34.0*) (*?v1.0*::(*real*, *3*) *cart*)) (*vector_add* (% (*chi ?x12.0 ?x24.0 ?x23.0 ?x14.0 ?x13.0 ?x34.0*) (*?v2.0*::(*real*, *3*) *cart*)) (*vector_add* (% (*chi ?x34.0 ?x13.0 ?x23.0 ?x14.0 ?x24.0 ?x12.0*) (*?v3.0*::(*real*, *3*) *cart*)) (% (*chi ?x34.0 ?x24.0 ?x14.0 ?x23.0 ?x13.0 ?x12.0*) (*?v4.0*::(*real*, *3*) *cart*)))))) (*hull affine* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0* (*INSERT ?v4.0 EMPTY*)))))

thm Collect_geom2.VBVYGGT:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *CARD* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = (*4*::*nat*) ∧ ¬ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *IN* (*circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) (*hull affine* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) ∧ (∃ *r*::*real*. ∀ *v*::(*real*, *3*) *cart*. *IN v* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *r* = *distance* (*circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))), *v*)) ∧ (∀ *p*::(*real*, *3*) *cart*. *IN p* (*hull affine* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) ∧ (∃ *r*::*real*. ∀ *v*::(*real*, *3*) *cart*. *IN v* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) ⟶ *r* = *distance* (*p*, *v*)) ⟶ *p* = *circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))) ∧ *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* (*LET* (λ*chi11*::*real*. *LET_END* (*LET* (λ*chi22*::*real*. *LET_END* (*LET* (λ*chi33*::*real*. *LET_END* (*LET* (λ*chi44*::*real*. *LET_END* (*circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) = % ((*1*::*real*) / (*real_of_nat* (*2*::*nat*) ∗ *delta x12 x13 x14 x23 x24 x34*)) (*vector_add* (% *chi11 v1*) (*vector_add* (% *chi22 v2*) (*vector_add* (% *chi33 v3*) (% *chi44 v4*)))))) (*chi x34 x24 x14 x23 x13 x12*))) (*chi x34 x13 x23 x14 x24 x12*))) (*chi x12 x24 x23 x14 x13 x34*))) (*chi x12 x13 x14 x23 x24 x34*))) ((*distance* (*v3*, *v4*))^2))) ((*distance* (*v2*, *v4*))^2))) ((*distance* (*v2*, *v3*))^2))) ((*distance* (*v1*, *v4*))^2))) ((*distance* (*v1*, *v3*))^2))) ((*distance* (*v1*, *v2*))^2)

thm Collect_geom2.THREE_POINTS_COP:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *coplanar_alt* (*INSERT* *v1* (*INSERT v2* (*INSERT v3 EMPTY*)))

thm Collect_geom2.PER_SET4:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) (*INSERT* (*?d*::*?'a*::*type*) *EMPTY*))) = *INSERT ?b* (*INSERT ?a* (*INSERT ?c* (*INSERT* *?d EMPTY*))) $\wedge$ *INSERT ?a* (*INSERT ?b* (*INSERT ?c* (*INSERT ?d EMPTY*))) = *INSERT ?c* (*INSERT ?b* (*INSERT ?a* (*INSERT ?d EMPTY*))) $\wedge$ *INSERT* *?a* (*INSERT ?b* (*INSERT ?c* (*INSERT ?d EMPTY*))) = *INSERT ?d* (*INSERT* *?b* (*INSERT ?c* (*INSERT ?a EMPTY*)))

thm Collect_geom2.NOT_COPLANAR_IMP_CARD4:

$\neg$ *coplanar_alt* (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) (*INSERT* (*?v3.0*::(*real*, *3*) *cart*) (*INSERT* (*?v4.0*::(*real*, *3*) *cart*) *EMPTY*)))) $\longrightarrow$ *CARD* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0* (*INSERT ?v4.0* *EMPTY*)))) = (*4*::*nat*)

thm Collect_geom2.NOT_COPLANAR_IMP_EXISTS_CIR2:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. $\neg$ *coplanar_alt* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) $\longrightarrow$ *IN* (*circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4* *EMPTY*))))) (*hull affine* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT* *v4 EMPTY*))))) $\wedge$ ($\exists$ *r*::*real*. $\forall$ *v*::(*real*, *3*) *cart*. *IN v* (*INSERT v1* (*INSERT* *v2* (*INSERT v3* (*INSERT v4 EMPTY*)))) $\longrightarrow$ *r* = *distance* (*circumcenter* (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*)))), *v*))

thm Collect_geom2.NOT_COPLANAR_IMP_RADV_PROPERTIES:

$\neg$ *coplanar_alt* (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) (*INSERT* (*?v3.0*::(*real*, *3*) *cart*) (*INSERT* (*?v4.0*::(*real*, *3*) *cart*) *EMPTY*)))) $\longrightarrow$ ($\forall$ *w*::(*real*, *3*) *cart*. *INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0* (*INSERT* *?v4.0 EMPTY*))) *w* $\longrightarrow$ *radV* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT* *?v3.0* (*INSERT ?v4.0 EMPTY*)))) = *distance* (*circumcenter* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0* (*INSERT ?v4.0 EMPTY*)))), *w*))

thm Collect_geom2.NOT_COL_EQ_UPS_X_POS:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. ($\neg$ *collinear* (*INSERT* *v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) = ((*0*::*real*) < *ups_x* ((*distance* (*v1*, *v2*))^2) ((*distance* (*v1*, *v3*))^2) ((*distance* (*v2*, *v3*))^2))

thm Collect_geom2.ETA_Y_POW2_EQ:

(*distance* (*?v1.0*::(*real*, *3*) *cart*, *?v2.0*::(*real*, *3*) *cart*))^2 $*$ ((*distance* (*?v1.0*, *?v3.0*::(*real*, *3*) *cart*))^2 $*$ (*distance* (*?v2.0*, *?v3.0*))^2) / *ups_x* ((*distance* (*?v1.0*, *?v2.0*))^2) ((*distance* (*?v1.0*, *?v3.0*))^2) ((*distance* (*?v2.0*, *?v3.0*))^2) = (*eta_y* (*d3 ?v2.0 ?v3.0*) (*d3 ?v1.0 ?v3.0*) (*d3 ?v1.0 ?v2.0*))^2

thm Collect_geom2.ETA_Y_POS_LE:

$(0::real) \leq eta\_y\ (d3\ (?v1.0::(real,\ 3)\ cart)\ (?v2.0::(real,\ 3)\ cart))\ (d3\ ?v1.0\ (?v3.0::(real,\ 3)\ cart))\ (d3\ ?v2.0\ ?v3.0)$

thm Collect_geom2.ZJEWPAP:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ v4::(real,\ 3)\ cart.\ LET\ (\lambda s::(real,\ 3)\ cart \Rightarrow bool.\ LET\_END\ (CARD\ s = (4::nat) \wedge \neg\ coplanar\_alt\ s \longrightarrow radV\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) \leq radV\ s))\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY))))$

thm Collect_geom2.NOT_EQ_BASIS_IMP_OTHORGANAL:

$\forall (i::nat)\ j::nat.\ i \neq j \longrightarrow dot\ (basis\ i)\ (basis\ j) = (0::real)$

thm Collect_geom2.BASIS_DIS_OTHORGONAL:

$dot\ (basis\ (1::nat))\ (basis\ (2::nat)) = (0::real) \wedge dot\ (basis\ (1::nat))\ (basis\ (3::nat)) = (0::real) \wedge dot\ (basis\ (2::nat))\ (basis\ (3::nat)) = (0::real)$

thm Collect_geom2.NORM_BASIS_VEC3:

$\forall i::nat.\ i = (1::nat) \vee i = (2::nat) \vee i = (3::nat) \longrightarrow vector\_norm\ (basis\ i) = (1::real)$

thm Collect_geom2.AAA_LEMMA:

$(0::real) < (?a::real) \wedge ?a \leq (?b::real) \wedge ?b \leq (?c::real) \wedge (?ll::bool) \longrightarrow (0::real) \leq ?b^2 - ?a^2 \wedge (0::real) \leq ?c^2 - ?b^2$

thm Collect_geom2.LLEEMAA:

$(0::real) < (?a::real) \wedge ?a \leq (?b::real) \wedge ?b \leq (?c::real) \wedge (0::real) < (?a'::real) \wedge ?a' \leq (?b'::real) \wedge ?b' \leq (?c'::real) \wedge ?a \leq ?a' \wedge ?b \leq ?b' \wedge ?c \leq ?c' \wedge (?l::bool) \longrightarrow (0::real) \leq ?a'^2 - ?a^2 \wedge (0::real) \leq ?b'^2 - ?b^2 \wedge (0::real) \leq ?c'^2 - ?c^2$

thm Collect_geom2.BASIS_DIS_OTHORGONAL_conjunct2:

$dot\ (basis\ (2::nat))\ (basis\ (3::nat)) = (0::real)$

thm Collect_geom2.BASIS_DIS_OTHORGONAL_conjunct1:

$dot\ (basis\ (1::nat))\ (basis\ (3::nat)) = (0::real)$

thm Collect_geom2.BASIS_DIS_OTHORGONAL_conjunct0:

$dot\ (basis\ (1::nat))\ (basis\ (2::nat)) = (0::real)$

thm Collect_geom2.LEMMA83:

$\forall (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ (e3::(real,\ 3)\ cart)\ (a::real)\ (b::real)\ (c::real)\ (a'::real)\ (b'::real)\ (c'::real)\ (t1::real)\ (t2::real)\ t3::real.\ e1 = basis\ (1::nat) \wedge e2 = basis\ (2::nat) \wedge e3 = basis\ (3::nat) \wedge (0::real) < a \wedge a \leq b \wedge b \leq c \wedge (0::real) < a' \wedge a' \leq b' \wedge b' \leq c' \wedge a \leq a' \wedge b \leq b' \wedge c \leq c' \wedge (\forall x::real.\ IN\ x\ (INSERT\ t1\ (INSERT\ t2\ (INSERT\ t3\ EMPTY))) \longrightarrow (0::real) < x) \wedge t1 + (t2 + t3) < (1::real) \wedge (?v::(real,\ 3)\ cart) = vector\_add$

(% ((t1 + (t2 + t3)) * a) e1) (vector_add (% ((t2 + t3) * sqrt (b^2 − a^2)) e2) (% (t3 * sqrt (c^2 − b^2)) e3)) ∧ (?v′::(real, 3) cart) = vector_add (% ((t1 + (t2 + t3)) * a′) e1) (vector_add (% ((t2 + t3) * sqrt (b'^2 − a'^2)) e2) (% (t3 * sqrt (c'^2 − b'^2)) e3)) ⟶ vector_norm ?v ≤ vector_norm ?v′

thm Collect_geom2.DELTA_TRIPLE_SUB_H_EXPAND:

delta ((?a01.0::real) − (?h::real)) ((?a02.0::real) − ?h) ((?a03.0::real) − ?h) (?x12.0::real) (?x13.0::real) (?x23.0::real) = delta ?a01.0 ?a02.0 ?a03.0 ?x12.0 ?x13.0 ?x23.0 − ?h * ups_x ?x12.0 ?x13.0 ?x23.0

thm Collect_geom2.PROVE_EXISTS_H_DELTA_0:

(0::real) < ups_x (?x12.0::real) (?x13.0::real) (?x23.0::real) ∧ (0::real) ≤ delta (?a01.0::real) (?a02.0::real) (?a03.0::real) ?x12.0 ?x13.0 ?x23.0 ⟶ (∃ h≥0::real. h = delta ?a01.0 ?a02.0 ?a03.0 ?x12.0 ?x13.0 ?x23.0 / ups_x ?x12.0 ?x13.0 ?x23.0 ∧ delta (?a01.0 − h) (?a02.0 − h) (?a03.0 − h) ?x12.0 ?x13.0 ?x23.0 = (0::real))

thm Collect_geom2.FIRST_POINT_IN_AFF3:

∀ (w::(real, ?′a::type) cart) (v1::(real, ?′a::type) cart) v2::(real, ?′a::type) cart. IN w (aff (INSERT w (INSERT v1 (INSERT v2 EMPTY))))

thm Collect_geom2.THREE_GEN_POINTS_IN_AFF3:

IN (?a::(real, ?′a::type) cart) (aff (INSERT ?a (INSERT (?b::(real, ?′a::type) cart) (INSERT (?c::(real, ?′a::type) cart) EMPTY)))) ∧ IN ?b (aff (INSERT ?a (INSERT ?b (INSERT ?c EMPTY)))) ∧ IN ?c (aff (INSERT ?a (INSERT ?b (INSERT ?c EMPTY))))

thm Collect_geom2.NOT_COPLANAR_IMP_NOT_COLLINEAR:

¬ coplanar_alt (INSERT (?v1.0::(real, 3) cart) (INSERT (?v2.0::(real, 3) cart) (INSERT (?v3.0::(real, 3) cart) (INSERT (?v::(real, 3) cart) EMPTY)))) ⟶ ¬ collinear (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 EMPTY)))

thm Collect_geom2.PROVE_THE_HYPOTHESI_FOR_74:

LET (λs::(real, 3) cart ⇒ bool. LET_END (CARD s = (4::nat) ∧ ¬ coplanar_alt s ∧ eta_y (d3 (?v1.0::(real, 3) cart) (?v2.0::(real, 3) cart)) (d3 ?v1.0 (?v3.0::(real, 3) cart)) (d3 ?v2.0 ?v3.0) ≤ (?r::real))) (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT (?v4.0::(real, 3) cart) EMPTY)))) ⟶ LET (λx12::real. LET_END (LET (λx13::real. LET_END (LET (λx23::real. LET_END (CARD (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY)))) = (4::nat) ∧ ¬ coplanar_alt (INSERT ?v1.0 (INSERT ?v2.0 (INSERT ?v3.0 (INSERT ?v4.0 EMPTY)))) ∧ (0::real) ≤ $?r^2$ ∧ (0::real) ≤ $?r^2$ ∧ (0::real) ≤ $?r^2$ ∧ (0::real) ≤ delta ($?r^2$) ($?r^2$) ($?r^2$) x12 x13 x23)) ((d3 ?v2.0 ?v3.0)^2))) ((d3 ?v1.0 ?v3.0)^2))) ((d3 ?v1.0 ?v2.0)^2)

thm Collect_geom2.INSERT_SUBSET:

SUBSET EMPTY (?s::?′a::type ⇒ bool) ∧ SUBSET (INSERT (?a::?′a::type) ?s) (?ss::?′a::type ⇒ bool) = (IN ?a ?ss ∧ SUBSET ?s ?ss)

thm Collect_geom2.INSERT_SUBSET_conjunct1:

SUBSET (*INSERT* (*?a*::*?'a*::*type*) (*?s*::*?'a*::*type* ⇒ *bool*)) (*?ss*::*?'a*::*type* ⇒ *bool*) = (*IN ?a ?ss* ∧ *SUBSET ?s ?ss*)

thm Collect_geom2.INSERT_SUBSET_conjunct0:

SUBSET EMPTY (*?s*::*?'a*::*type* ⇒ *bool*)

thm Collect_geom2.IMP_OTHORGONAL_AFF3:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. *dot u* (*vector_sub v1 v2*) = (*0*::*real*) ∧ *dot u* (*vector_sub v1 v3*) = (*0*::*real*) ⟶ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*INSERT x* (*INSERT y EMPTY*)) (*aff* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ⟶ *dot u* (*vector_sub x y*) = (*0*::*real*))

thm Collect_geom2.DIST_EQ_IMP_OTHORGONAL:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'a*::*type*) *cart*) *q*::(*real*, *?'a*::*type*) *cart*. *distance* (*p*, *a*) = *distance* (*p*, *b*) ∧ *distance* (*q*, *a*) = *distance* (*q*, *b*) ⟶ *dot* (*vector_sub p q*) (*vector_sub a b*) = (*0*::*real*)

thm Collect_geom2.PVLJZLA:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *v4*::(*real*, *3*) *cart*. *LET* (λ*s*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* (¬ *coplanar_alt s* ⟶ *IN* (*circumcenter s*) (*conv0 s*) = (*orientation s v1* (*op* < (*0*::*real*)) ∧ *orientation s v2* (*op* < (*0*::*real*)) ∧ *orientation s v3* (*op* < (*0*::*real*)) ∧ *orientation s v4* (*op* < (*0*::*real*))))) (*INSERT v1* (*INSERT v2* (*INSERT v3* (*INSERT v4 EMPTY*))))

thm Collect_geom2.IMP_IN_AFF_LT:

CARD (*INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v3.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v4.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)))) = (*4*::*nat*) ⟶ (∃ (*v*::*real*) (*v'*::*real*) (*v''*::*real*) *v'''*::*real*. *v* < (*0*::*real*) ∧ (*1*::*real*) = *v* + (*v'* + (*v''* + *v'''*)) ∧ (*?vv*::(*real*, *?'a*::*type*) *cart*) = *vector_add* (% *v ?v1.0*) (*vector_add* (% *v' ?v2.0*) (*vector_add* (% *v'' ?v3.0*) (% *v''' ?v4.0*)))) = *IN ?vv* (*aff_lt* (*INSERT ?v2.0* (*INSERT ?v3.0* (*INSERT ?v4.0 EMPTY*))) (*INSERT ?v1.0 EMPTY*))

thm Collect_geom2.SQRT4_EQ2:

sqrt (*real_of_nat* (*4*::*nat*)) = *real_of_nat* (*2*::*nat*)

thm Collect_geom2.SHOGYBS:

∀ (*x1*::(*real*, *3*) *cart*) (*x2*::(*real*, *3*) *cart*) (*x3*::(*real*, *3*) *cart*) *x4*::(*real*, *3*) *cart*. ¬ *coplanar_alt* (*INSERT x1* (*INSERT x2* (*INSERT x3* (*INSERT x4 EMPTY*)))) ⟶ *LET* (λ*x12*::*real*. *LET_END* (*LET* (λ*x13*::*real*. *LET_END* (*LET* (λ*x14*::*real*. *LET_END* (*LET* (λ*x23*::*real*. *LET_END* (*LET* (λ*x24*::*real*. *LET_END* (*LET* (λ*x34*::*real*. *LET_END* ((*0*::*real*) ≤ *rho_ij x12 x13 x14 x23 x24 x34*)) ((*distance* (*x3*, *x4*))^2))) ((*distance* (*x2*, *x4*))^2))) ((*distance* (*x2*, *x3*))^2))) ((*distance* (*x1*, *x4*))^2))) ((*distance* (*x1*, *x3*))^2))) ((*distance* (*x1*, *x2*))^2)

thm Collect_geom2.GDRQXLG:

$\forall (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ v4::(real,\ 3)\ cart.\ LET\ (\lambda s::(real,\ 3)\ cart \Rightarrow bool.\ LET\_END\ (LET\ (\lambda x12::real.\ LET\_END\ (LET\ (\lambda x13::real.\ LET\_END\ (LET\ (\lambda x14::real.\ LET\_END\ (LET\ (\lambda x23::real.\ LET\_END\ (LET\ (\lambda x24::real.\ LET\_END\ (LET\ (\lambda x34::real.\ LET\_END\ (CARD\ s = (4::nat) \wedge \neg\ coplanar\_alt\ s \longrightarrow radV\ s = sqrt\ (rho\_ij\ x12\ x13\ x14\ x23\ x24\ x34) / (real\_of\_nat\ (2::nat) * sqrt\ (delta\ x12\ x13\ x14\ x23\ x24\ x34))))\ ((distance\ (v3,\ v4))^2)))\ ((distance\ (v2,\ v4))^2)))\ ((distance\ (v2,\ v3))^2)))\ ((distance\ (v1,\ v4))^2)))\ ((distance\ (v1,\ v3))^2)))\ ((distance\ (v1,\ v2))^2)))\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ (INSERT\ v4\ EMPTY))))$

thm Real_ext.REAL_INV2:

$inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) * real\_of\_nat\ (2::nat) = (1::real) \wedge real\_of\_nat\ (2::nat) * inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) = (1::real)$

thm Real_ext.REAL_PROP_EQ_LMUL:

$\forall (x::real)\ (a::real)\ b::real.\ x * a = x * b \longrightarrow x \neq (0::real) \longrightarrow a = b$

thm Real_ext.REAL_PROP_EQ_RMUL:

$\forall (x::real)\ (a::real)\ b::real.\ a * x = b * x \longrightarrow x \neq (0::real) \longrightarrow a = b$

thm Real_ext.REAL_PROP_LE_LCANCEL:

$\forall (x::real)\ (a::real)\ b::real.\ x * a \leq x * b \longrightarrow (0::real) < x \longrightarrow a \leq b$

thm Real_ext.REAL_MUL_RTIMES_LE:

$\forall (x::real)\ (a::real)\ b::real.\ a * x \leq b * x \longrightarrow (0::real) < x \longrightarrow a \leq b$

thm Real_ext.REAL_LE_LMUL_LOCAL:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \longrightarrow (x * y \leq x * z) = (y \leq z)$

thm Real_ext.REAL_PROP_LE_LABS:

$\forall (x::real)\ (y::real)\ z::real.\ y \leq z \longrightarrow |x| * y \leq |x| * z$

thm Real_ext.REAL_LE_RMUL_IMP:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x \wedge y \leq z \longrightarrow y * x \leq z * x$

thm Real_ext.REAL_PROP_NN_RCANCEL:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) \leq y * x \longrightarrow (0::real) \leq y$

thm Real_ext.REAL_PROP_NN_LCANCEL:

$\forall (x::real)\ y::real.\ (0::real) < x \wedge (0::real) \leq x * y \longrightarrow (0::real) \leq y$

thm Real_ext.REAL_PROP_NZ_ABS:

$\forall x::real.\ x \neq (0::real) \longrightarrow |x| \neq (0::real)$

thm Tactics_jordan.pthm:

True

thm Tactics_jordan.SELECT_EXIST:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ (\exists\, y::?'a::type.\ P\ y) \wedge (\forall\, t::?'a::type.\ P\ t \longrightarrow Q\ t) \longrightarrow Q\ (Eps\ P)$

thm Tactics_jordan.SELECT_THM:

$\forall (P::?'a::type \Rightarrow bool)\ Q::?'a::type \Rightarrow bool.\ ((\exists\, y::?'a::type.\ P\ y) \longrightarrow (\forall\, t::?'a::type.\ P\ t \longrightarrow Q\ t)) \wedge (\neg\ (\exists\, y::?'a::type.\ P\ y) \longrightarrow (\forall\, t::?'a::type.\ Q\ t)) \longrightarrow Q\ (Eps\ P)$

thm Tactics_jordan.test_real_poly_tac:

$\forall (x::real)\ y::real.\ (x + real\_of\_nat\ (2::nat) * y) * (x - real\_of\_nat\ (2::nat) * y) = x * x - real\_of\_nat\ (4::nat) * (y * y)$

thm Tactics_jordan.strict_lemma:

$\forall (A::real)\ (B::real)\ C::real.\ A + B = C \longrightarrow (0::real) < B \longrightarrow A < C$

thm Tactics_jordan.weak_lemma:

$\forall (A::real)\ (B::real)\ C::real.\ A + B = C \longrightarrow (0::real) \leq B \longrightarrow A \leq C$

thm Tactics_jordan.strip_lt_lemma:

$\forall (B1::real)\ (B2::real)\ C::bool.\ ((0::real) < B1 + B2 \longrightarrow C) \longrightarrow (0::real) < B2 \longrightarrow (0::real) \leq B1 \longrightarrow C$

thm Tactics_jordan.strip_le_lemma:

$\forall (B1::real)\ (B2::real)\ C::bool.\ ((0::real) \leq B1 + B2 \longrightarrow C) \longrightarrow (0::real) \leq B2 \longrightarrow (0::real) \leq B1 \longrightarrow C$

thm Tactics_jordan.switch_lemma_le:

$\forall (A::real)\ (B::real)\ C::bool.\ ((0::real) \leq A \longrightarrow (0::real) \leq B \longrightarrow C) = ((0::real) \leq B \longrightarrow (0::real) \leq A \longrightarrow C)$

thm Tactics_jordan.switch_lemma_let:

$\forall (A::real)\ (B::real)\ C::bool.\ ((0::real) < A \longrightarrow (0::real) \leq B \longrightarrow C) = ((0::real) \leq B \longrightarrow (0::real) < A \longrightarrow C)$

thm Tactics_jordan.switch_lemma_lt:

$\forall (A::real)\ (B::real)\ C::bool.\ ((0::real) < A \longrightarrow (0::real) < B \longrightarrow C) = ((0::real) < B \longrightarrow (0::real) < A \longrightarrow C)$

thm Tactics_jordan.expand_prod_lt:

$\forall (B1::real)\ (B2::real)\ C::bool.\ ((0::real) < B1 * B2 \longrightarrow C) \longrightarrow (0::real) < B1 \longrightarrow (0::real) < B2 \longrightarrow C$

thm Tactics_jordan.expand_prod_le:

$\forall (B1::real)\ (B2::real)\ C::bool.\ ((0::real) \leq B1 * B2 \longrightarrow C) \longrightarrow (0::real) \leq B1 \longrightarrow (0::real) \leq B2 \longrightarrow C$

thm Tactics_jordan.test_ineq_tac:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) \leq x * y \wedge (0::real) < z \longrightarrow x * y < x * x + (real\_of\_nat\ (3::nat) * (x * y) + real\_of\_nat\ (4::nat))$

thm DEF_TAGB:

$TAGB = (\lambda\_2113904::bool.\ \_2113904)$

thm Tactics_jordan.tagb:

$\forall x::bool.\ TAGB\ x = x$

thm DEF_mark_term:

$mark\_term = (\lambda\_2114317::?'a::type.\ \_2114317)$

thm Tactics_jordan.mark_term:

$\forall u::?'a::type.\ mark\_term\ u = u$

thm Tactics_jordan.REAL_HALF_DOUBLE:

$\forall x::real.\ x\ /\ real\_of\_nat\ (2::nat) + x\ /\ real\_of\_nat\ (2::nat) = x$

thm Tactics_jordan.REAL_DOUBLE:

$\forall x::real.\ x + x = real\_of\_nat\ (2::nat) * x$

thm Tactics_jordan.REAL_POW_2_LE:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x^2 \leq y^2 \longrightarrow x \leq y$

thm Tactics_jordan.REAL_POW_2_LT:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x^2 < y^2 \longrightarrow x < y$

thm Tactics_jordan.unify_exists_example:

$\exists (x::nat)\ y::nat.\ x = y$

thm Tactics_jordan.drop_ant_tac_example:

$\forall (A::bool)\ (B::bool)\ (C::bool)\ (D::bool)\ E::bool.\ A \wedge (A \longrightarrow B) \wedge (C \longrightarrow D) \wedge C \longrightarrow E \vee C \vee B$

thm Num_ext_nabs.INT_NUM:

$\forall u::nat.\ integer\ (real\_of\_nat\ u)$

thm Num_ext_nabs.INT_NUM_REAL:

$\forall u::nat.\ real\_of\_int\ (int\ u) = real\_of\_nat\ u$

thm Num_ext_nabs.INT_IS_INT:

$\forall a::int.\ integer\ (real\_of\_int\ a)$

thm Num_ext_nabs.INT_OF_NUM_DEST:

$\forall (a::int)\ n::nat.\ (real\_of\_int\ a = real\_of\_nat\ n) = (a = int\ n)$

thm Num_ext_nabs.INT_REP:

$\forall a::int.\ \exists (n::nat)\ m::nat.\ a = int\ n - int\ m$

thm Num_ext_nabs.INT_REP2:

$\forall a::int.\ \exists n::nat.\ a = int\ n \vee a = -\ int\ n$

thm DEF_nabs:

$nabs = (\lambda\_2114717::int.\ SOME\ u::nat.\ \_2114717 = int\ u \vee \_2114717 = -\ int\ u)$

thm Num_ext_nabs.nabs:

$\forall n::int.\ nabs\ n = (SOME\ u::nat.\ n = int\ u \vee n = -\ int\ u)$

thm Num_ext_nabs.NABS_POS:

$\forall u::nat.\ nabs\ (int\ u) = u$

thm Num_ext_nabs.NABS_NEG:

$\forall n::nat.\ nabs\ (-\ int\ n) = n$

thm DEF_halfatn:

$halfatn = (\lambda\_2114794::real.\ \_2114794\ /\ (sqrt\ ((1::real) + \_2114794^2) + (1::real)))$

thm Taylor_atn.halfatn:

$\forall x::real.\ halfatn\ x = x\ /\ (sqrt\ ((1::real) + x^2) + (1::real))$

thm Taylor_atn.pos1:

$\forall x::real.\ (0::real) < (1::real) + x^2$

thm DEF_ssqrt:

$ssqrt = (\lambda\_2114801::real.\ if\ \_2114801 < (0::real)\ then\ 0::real\ else\ sqrt\ \_2114801)$

thm Float.ssqrt:

$\forall x::real.\ ssqrt\ x = (if\ x < (0::real)\ then\ 0::real\ else\ sqrt\ x)$

thm Taylor_atn.halfsqrt_ssqrt:

$\forall x::real.\ sqrt\ ((1::real) + x^2) = ssqrt\ ((1::real) + x^2)$

thm Taylor_atn.pos2:

$\forall x::real.\ (0::real) < sqrt\ ((1::real) + x^2) + (1::real)$

thm Taylor_atn.halfatn_bounds_abs:

$\forall x::real.\ |halfatn\ x| < (1::real)$

thm Taylor_atn.halfatn_bounds:

$\forall x::real. - (1::real) < halfatn\ x \wedge halfatn\ x < (1::real)$

thm Taylor_atn.halfatn_half:

$\forall (x::real)\ t::real. |x| < t \longrightarrow |halfatn\ x| < t\ /\ real\_of\_nat\ (2::nat)$

thm Taylor_atn.abs_pass_through:

$\forall (x::real)\ f::real \Rightarrow real.\ f\ (-\ x) = -\ f\ x \wedge (\forall y \geq 0::real.\ (0::real) \leq f\ y) \longrightarrow |f\ x| = f\ |x|$

thm Taylor_atn.atn_abs:

$\forall x::real. |atn\ x| = atn\ |x|$

thm Taylor_atn.atn_half_range:

$\forall x::real. |atn\ (halfatn\ x)| < pi\ /\ real\_of\_nat\ (4::nat)$

thm Taylor_atn.tan_one_one:

$\forall (x::real)\ y::real. |x| < pi\ /\ real\_of\_nat\ (2::nat) \wedge |y| < pi\ /\ real\_of\_nat\ (2::nat) \wedge tan\ x = tan\ y \longrightarrow x = y$

thm Taylor_atn.abs_lemma:

$\forall (f::?'a::type \Rightarrow real)\ x::real. ((\exists n::?'a::type.\ x = f\ n) \vee (\exists n::?'a::type.\ x = -\ f\ n)) = (\exists n::?'a::type. |x| = |f\ n|)$

thm Taylor_atn.cos_nz:

$\forall x::real. |x| < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow cos\ x \neq (0::real)$

thm Taylor_atn.cos_2nz:

$\forall x::real. |x| < pi\ /\ real\_of\_nat\ (4::nat) \longrightarrow cos\ (real\_of\_nat\ (2::nat) * x) \neq (0::real)$

thm Taylor_atn.halfatn_double:

$\forall x::real.\ cos\ (atn\ (halfatn\ x)) \neq (0::real) \wedge cos\ (real\_of\_nat\ (2::nat) * atn\ (halfatn\ x)) \neq (0::real)$

thm Taylor_atn.REAL_DIV_MUL2z:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \longrightarrow y\ /\ z = x^2 * y\ /\ (x^2 * z)$

thm Taylor_atn.atn_half:

$\forall x::real.\ atn\ x = real\_of\_nat\ (2::nat) * atn\ (halfatn\ x)$

thm Taylor_atn.id1:

$complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_pow\ (?z::(real, 2)\ cart)\ (2::nat))) = complex\_mul\ (complex\_inv\ (Cx\ (real\_of\_nat\ (2::nat))))\ (vector\_add\ (complex\_mul\ (complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_pow\ ?z\ (2::nat))))\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ ?z)))\ (complex\_mul\ (complex\_inv\ (vector\_add$

$(Cx\ (1::real))\ (complex\_pow\ ?z\ (2::nat))))\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ ?z))))$

thm Taylor_atn.id2:

$complex\_mul\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ (?z::(real,\ 2)\ cart)))\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ ?z)) = vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ (complex\_mul\ ii\ (complex\_mul\ ?z\ ?z)))$

thm Taylor_atn.id3:

$\forall (u::(real,\ 2)\ cart)\ a::(real,\ 2)\ cart.\ vector\_sub\ a\ (complex\_mul\ ii\ (complex\_mul\ ii\ u)) = vector\_add\ a\ u$

thm Taylor_atn.id4:

$\forall z::(real,\ 2)\ cart.\ vector\_add\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat)) = complex\_mul\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ z))\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ z))$

thm Taylor_atn.idz:

$\forall (z::(real,\ 2)\ cart)\ a::(real,\ 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \wedge (a = ii \vee a = vector\_neg\ ii) \longrightarrow vector\_add\ (Cx\ (1::real))\ (complex\_mul\ a\ z) \neq Cx\ (0::real)$

thm Taylor_atn.id4a:

$\forall z::(real,\ 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow complex\_mul\ (complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ z)))\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ z)) = Cx\ (1::real)$

thm Taylor_atn.id4b:

$\forall z::(real,\ 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow complex\_mul\ (complex\_inv\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ z)))\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ z)) = Cx\ (1::real)$

thm Taylor_atn.id5:

$\forall z::(real,\ 2)\ cart.\ (Re\ z = (0::real) \longrightarrow |Im\ z| < (1::real)) \longrightarrow complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_pow\ z\ (2::nat))) = complex\_mul\ (complex\_inv\ (Cx\ (real\_of\_nat\ (2::nat))))\ (vector\_add\ (complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ z)))\ (complex\_inv\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ ii\ z))))$

thm DEF_taylor_coeff_catn:

$taylor\_coeff\_catn = (\lambda(\_2115197::nat)\ \_2115198::(real,\ 2)\ cart.\ if\ \_2115197 = (0::nat)\ then\ catn\ \_2115198\ else\ complex\_mul\ (Cx\ (real\_of\_nat\ (fact\ (\_2115197 - (1::nat)))))\ (complex\_mul\ (complex\_inv\ (Cx\ (real\_of\_nat\ (2::nat))))\ (vector\_add\ (complex\_mul\ (complex\_pow\ (vector\_neg\ ii)\ (\_2115197 - (1::nat)))\ (complex\_pow\ (complex\_inv\ (vector\_add\ (Cx\ (1::real))\ (complex\_mul\ ii\ \_2115198)))\ \_2115197))$

(*complex_mul* (*complex_pow ii* (*_2115197* − (*1*::*nat*))) (*complex_pow* (*complex_inv* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_mul ii _2115198*))) *_2115197*)))))

thm Taylor_atn.taylor_coeff_catn:

∀ (*z*::(*real*, *2*) *cart*) *n*::*nat*. *taylor_coeff_catn n z* = (*if n* = (*0*::*nat*) *then catn z else complex_mul* (*Cx* (*real_of_nat* (*fact* (*n* − (*1*::*nat*))))) (*complex_mul* (*complex_inv* (*Cx* (*real_of_nat* (*2*::*nat*)))) (*vector_add* (*complex_mul* (*complex_pow* (*vector_neg ii*) (*n* − (*1*::*nat*))) (*complex_pow* (*complex_inv* (*vector_add* (*Cx* (*1*::*real*)) (*complex_mul ii z*))) *n*)) (*complex_mul* (*complex_pow ii* (*n* − (*1*::*nat*))) (*complex_pow* (*complex_inv* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_mul ii z*))) *n*)))))

thm Taylor_atn.taylor_coeff_catn0:

taylor_coeff_catn (*0*::*nat*) = *catn*

thm Taylor_atn.taylor_coeff_catn1:

∀ *z*::(*real*, *2*) *cart*. (*Re z* = (*0*::*real*) ⟶ |*Im z*| < (*1*::*real*)) ⟶ *has_complex_derivative catn* (*taylor_coeff_catn* (*1*::*nat*) *z*) (*at z*)

thm Taylor_atn.taylor_coeff_catn_pos:

∀ *n*>*0*::*nat*. *taylor_coeff_catn n* = (λ*z*::(*real*, *2*) *cart*. *complex_mul* (*Cx* (*real_of_nat* (*fact* (*n* − (*1*::*nat*))))) (*complex_mul* (*complex_inv* (*Cx* (*real_of_nat* (*2*::*nat*)))) (*vector_add* (*complex_mul* (*complex_pow* (*vector_neg ii*) (*n* − (*1*::*nat*))) (*complex_pow* (*complex_inv* (*vector_add* (*Cx* (*1*::*real*)) (*complex_mul ii z*))) *n*)) (*complex_mul* (*complex_pow ii* (*n* − (*1*::*nat*))) (*complex_pow* (*complex_inv* (*vector_sub* (*Cx* (*1*::*real*)) (*complex_mul ii z*))) *n*)))))

thm Taylor_atn.taylor_series_inv_pow:

∀ (*n*::*nat*) (*a*::(*real*, *2*) *cart*) *z*::(*real*, *2*) *cart*. *vector_add* (*Cx* (*1*::*real*)) (*complex_mul a z*) ≠ *Cx* (*0*::*real*) ⟶ *has_complex_derivative* (λ*z*::(*real*, *2*) *cart*. *complex_pow* (*complex_inv* (*vector_add* (*Cx* (*1*::*real*)) (*complex_mul a z*))) *n*) (*complex_mul* (*vector_neg* (*Cx* (*real_of_nat n*))) (*complex_mul a* (*complex_pow* (*complex_inv* (*vector_add* (*Cx* (*1*::*real*)) (*complex_mul a z*))) (*n* + (*1*::*nat*))))) (*at z*)

thm Taylor_atn.factorial_lemma:

∀ *n*>*0*::*nat*. *fact n* = *n* ∗ *fact* (*n* − (*1*::*nat*))

thm Taylor_atn.taylor_coeff_catn_deriv:

∀ (*z*::(*real*, *2*) *cart*) *n*::*nat*. (*Re z* = (*0*::*real*) ⟶ |*Im z*| < (*1*::*real*)) ⟶ *has_complex_derivative* (*taylor_coeff_catn n*) (*taylor_coeff_catn* (*n* + (*1*::*nat*)) *z*) (*at z*)

thm Taylor_atn.ipows2:

∀ *n*::*nat*. *complex_pow* (*vector_neg ii*) (*n* + (*2*::*nat*)) = *vector_neg* (*complex_pow* (*vector_neg ii*) *n*)

thm Taylor_atn.ipowsc2:

$\forall n::nat.\ complex\_pow\ ii\ (n + (2::nat)) = vector\_neg\ (complex\_pow\ ii\ n)$

thm Taylor_atn.taylor_coeff0:

$\forall n::nat.\ taylor\_coeff\_catn\ n\ (Cx\ (0::real)) = (if\ even\ n\ then\ Cx\ (0::real)\ else\ Cx\ (real\_of\_nat\ (fact\ (n - (1::nat))) * (- (1::real))^{(n - (1::nat))\ div\ (2::nat)}))$

thm Taylor_atn.term_bound:

$\forall (a::real)\ (n::nat)\ z::(real, 2)\ cart.\ Im\ z = (0::real) \longrightarrow vector\_norm\ (complex\_mul\ (complex\_pow\ (complex\_mul\ (Cx\ a)\ ii)\ n)\ (complex\_pow\ (complex\_inv\ (vector\_sub\ (Cx\ (1::real))\ (complex\_mul\ (Cx\ a)\ (complex\_mul\ ii\ z))))\ (n + (1::nat)))) \leq |a|^n$

thm Taylor_atn.taylor_error_bound:

$\forall (n::nat)\ z::(real, 2)\ cart.\ Im\ z = (0::real) \longrightarrow vector\_norm\ (taylor\_coeff\_catn\ (n + (1::nat))\ z) \leq real\_of\_nat\ (fact\ n)$

thm Taylor_atn.complex_taylor_catn:

$\forall (n::nat)\ s::(real, 2)\ cart \Rightarrow bool.\ s = GSPEC\ (\lambda GEN\%PVAR\%100::(real, 2)\ cart.\ \exists z::(real, 2)\ cart.\ SETSPEC\ GEN\%PVAR\%100\ (Im\ z = (0::real))\ z) \longrightarrow (\forall z::(real, 2)\ cart.\ IN\ (Cx\ (0::real))\ s \wedge IN\ z\ s \longrightarrow vector\_norm\ (vector\_sub\ (catn\ z)\ (vsum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ complex\_mul\ (taylor\_coeff\_catn\ i\ (Cx\ (0::real)))\ (complex\_div\ (complex\_pow\ z\ i)\ (Cx\ (real\_of\_nat\ (fact\ i))))))) \leq (vector\_norm\ z)^{n + (1::nat)})$

thm Taylor_atn.real_axis:

$\forall z::(real, 2)\ cart.\ (Im\ z = (0::real)) = (\exists x::real.\ z = Cx\ x)$

thm Taylor_atn.THREAD_IF:

$\forall (x::bool)\ (y::?'b::type)\ (z::?'b::type)\ f::?'b::type \Rightarrow ?'a::type.\ f\ (if\ x\ then\ y\ else\ z) = (if\ x\ then\ f\ y\ else\ f\ z)$

thm Taylor_atn.real_taylor_atn_ver1:

$\forall (n::nat)\ x::real.\ |atn\ x - sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ if\ even\ i\ then\ 0::real\ else\ (- (1::real))^{(i - (1::nat))\ div\ (2::nat)} * (x^i / real\_of\_nat\ i))| \leq |x|^{n + (1::nat)}$

thm Taylor_atn.sum_odd:

$\forall (g::nat \Rightarrow real)\ n::nat.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%102::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%102\ (ODD\ i \wedge IN\ i\ (dotdot\ (0::nat)\ ((2::nat) * n + (2::nat))))\ i))\ g = sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ g\ ((2::nat) * i + (1::nat)))$

thm Taylor_atn.sum_even:

$\forall (g::nat \Rightarrow real)\ n::nat.\ sum\ (GSPEC\ (\lambda GEN\%PVAR\%104::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%104\ (ODD\ i \wedge IN\ i\ (dotdot\ (0::nat)\ n))\ i))\ (\lambda i::nat.\ if\ even\ i\ then\ 0::real\ else\ g\ i) = sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ if\ even\ i\ then\ 0::real\ else\ g\ i)$

thm Taylor_atn.real_taylor_atn:

$\forall (n::nat)\ x::real.\ |atn\ x\ -\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda j::nat.\ (-\ (1::real))^{j}\ *\ (x^{(2::nat)\ *\ j\ +\ (1::nat)}\ /\ real\_of\_nat\ ((2::nat)\ *\ j\ +\ (1::nat))))| \leq |x|^{(2::nat)\ *\ n\ +\ (3::nat)}$

thm Taylor_atn.halfatn4:

$halfatn4\ =\ halfatn \circ (halfatn \circ (halfatn \circ halfatn))$

thm Taylor_atn.real_taylor_atn_halfatn4:

$\forall (n::nat)\ x::real.\ |atn\ (halfatn4\ x)\ -\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda j::nat.\ (-\ (1::real))^{j}\ *\ ((halfatn4\ x)^{(2::nat)\ *\ j\ +\ (1::nat)}\ /\ real\_of\_nat\ ((2::nat)\ *\ j\ +\ (1::nat))))| \leq inverse\_class.inverse\ (real\_of\_nat\ (8::nat))^{(2::nat)\ *\ n\ +\ (3::nat)}$

thm Taylor_atn.atn_halfatn4:

$\forall x::real.\ atn\ x\ =\ real\_of\_nat\ (16::nat)\ *\ atn\ (halfatn4\ x)$

thm Taylor_atn.real_taylor_atn_halfatn4_a:

$\forall (n::nat)\ x::real.\ |atn\ x\ -\ real\_of\_nat\ (16::nat)\ *\ sum\ (dotdot\ (0::nat)\ n)\ (\lambda j::nat.\ (-\ (1::real))^{j}\ *\ ((halfatn4\ x)^{(2::nat)\ *\ j\ +\ (1::nat)}\ /\ real\_of\_nat\ ((2::nat)\ *\ j\ +\ (1::nat))))| \leq inverse\_class.inverse\ (real\_of\_nat\ (2::nat))^{(6::nat)\ *\ n\ +\ (5::nat)}$

thm DEF_halfatn4_co:

$halfatn4\_co = (\lambda(\_2115802::real)\ \_2115803::nat.\ (-\ (1::real))^{\_2115803}\ *\ ((halfatn4\ \_2115802)^{(2::nat)\ *\ \_2115803\ +\ (1::nat)}\ /\ real\_of\_nat\ ((2::nat)\ *\ \_2115803\ +\ (1::nat))))$

thm Taylor_atn.halfatn4_co:

$\forall (x::real)\ j::nat.\ halfatn4\_co\ x\ j = (-\ (1::real))^{j}\ *\ ((halfatn4\ x)^{(2::nat)\ *\ j\ +\ (1::nat)}\ /\ real\_of\_nat\ ((2::nat)\ *\ j\ +\ (1::nat)))$

thm Taylor_atn.atn_bounds_anti:

$\forall (x::real)\ y::real.\ x \leq y \longrightarrow |atn\ x\ -\ atn\ y| \leq |x\ -\ y|$

thm Taylor_atn.atn_bounds:

$\forall (x::real)\ y::real.\ |atn\ x\ -\ atn\ y| \leq |x\ -\ y|$

thm Taylor_atn.real_taylor_atn_approx:

$\forall (n::nat)\ (x::real)\ (v::real)\ (eps1::real)\ (eps2::real)\ eps::real.\ inverse\_class.inverse\ (real\_of\_nat\ (2::nat))^{(6::nat)\ *\ n\ +\ (5::nat)} \leq eps1\ \wedge\ |real\_of\_nat\ (16::nat)\ *\ sum\ (dotdot\ (0::nat)\ n)\ (halfatn4\_co\ x)\ -\ v| \leq eps2\ \wedge\ eps1\ +\ eps2 \leq eps \longrightarrow |atn\ x\ -\ v| \leq eps$

thm DEF_twopow:

$twopow\ =\ (\lambda\_2115847::int.\ if\ \exists n::nat.\ \_2115847\ =\ int\ n\ then\ (real\_of\_nat\ (2::nat))^{nabs\ \_2115847}\ else\ inverse\_class.inverse\ (real\_of\_nat\ (2::nat))^{nabs\ \_2115847})$

thm Float.twopow:

$\forall x{::}int.\ twopow\ x = (if\ \exists n{::}nat.\ x = int\ n\ then\ (real\_of\_nat\ (2{::}nat))^{nabs\ x}$ $else\ inverse\_class.inverse\ (real\_of\_nat\ (2{::}nat))^{nabs\ x})$

thm DEF_float:

$float = (\lambda(\_2115852{::}int)\ \_2115853{::}int.\ real\_of\_int\ \_2115852 * twopow\ \_2115853)$

thm Float.float:

$\forall (x{::}int)\ n{::}int.\ float\ x\ n = real\_of\_int\ x * twopow\ n$

thm DEF_interval_eps:

$interval\_eps = (\lambda(\_2115864{::}real)\ \_2115865{::}real.\ op \leq |\_2115864 - \_2115865|)$

thm Float.interval_eps:

$\forall (x{::}real)\ (f{::}real)\ eps{::}real.\ interval\_eps\ x\ f\ eps = (|x - f| \leq eps)$

thm Float.TWOPOW_POS:

$\forall n{::}nat.\ twopow\ (int\ n) = (real\_of\_nat\ (2{::}nat))^n$

thm Float.TWOPOW_NEG:

$\forall n{::}nat.\ twopow\ (-\ int\ n) = inverse\_class.inverse\ (real\_of\_nat\ (2{::}nat))^n$

thm Float.TWOPOW_INV:

$\forall a{::}int.\ twopow\ (-\ a) = inverse\_class.inverse\ (twopow\ a)$

thm Float.INT_REP3:

$\forall a{::}int.\ \exists n{::}nat.\ a = int\ n \lor a = -\ int\ (n + (1{::}nat))$

thm Float.REAL_EQ_INV:

$\forall (x{::}real)\ y{::}real.\ (x = y) = (inverse\_class.inverse\ x = inverse\_class.inverse$ $y)$

thm Float.TWOPOW_ADD_1:

$\forall a{::}int.\ twopow\ (a + int\ (1{::}nat)) = twopow\ a * twopow\ (int\ (1{::}nat))$

thm Float.TWOPOW_0:

$twopow\ (int\ (0{::}nat)) = (1{::}real)$

thm Float.TWOPOW_SUB_NUM:

$\forall (m{::}nat)\ n{::}nat.\ twopow\ (int\ m - int\ n) = twopow\ (int\ m) * twopow\ (-\ int$ $n)$

thm Float.TWOPOW_ADD_NUM:

$\forall (m{::}nat)\ n{::}nat.\ twopow\ (int\ m + int\ n) = twopow\ (int\ m) * twopow\ (int\ n)$

thm Float.TWOPOW_ADD_INT:

$\forall (a{::}int)\ b{::}int.\ twopow\ (a + b) = twopow\ a * twopow\ b$

thm Float.TWOPOW_ABS:

$\forall\, a::int.\ |twopow\ a| = twopow\ a$

thm Float.TWOPOW_POW:

$\forall\, (a::int)\ n::nat.\ (twopow\ a)^n = twopow\ (a * int\ n)$

thm Float.FLOAT_NEG:

$\forall\, (a::int)\ m::int.\ -\ float\ a\ m = float\ (-\ a)\ m$

thm Float.FLOAT_MUL:

$\forall\, (a::int)\ (b::int)\ (m::int)\ n::int.\ float\ a\ m * float\ b\ n = float\ (a * b)\ (m + n)$

thm Float.FLOAT_ADD:

$\forall\, (a::int)\ (b::int)\ (c::nat)\ m::int.\ float\ a\ (m + int\ c) + float\ b\ m = float\ (int\ (2::nat)^c * a + b)\ m$

thm Float.FLOAT_ADD_EQ:

$\forall\, (a::int)\ (b::int)\ m::int.\ float\ a\ m + float\ b\ m = float\ (a + b)\ m$

thm Float.FLOAT_ADD_NP:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ float\ b\ (-\ int\ n) + float\ a\ (int\ m) = float\ a\ (int\ m) + float\ b\ (-\ int\ n)$

thm Float.FLOAT_ADD_PN:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ float\ a\ (int\ m) + float\ b\ (-\ int\ n) = float\ (int\ (2::nat)^{m\ +\ n} * a + b)\ (-\ int\ n)$

thm Float.FLOAT_ADD_PP:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ n \le m \longrightarrow float\ a\ (int\ m) + float\ b\ (int\ n) = float\ (int\ (2::nat)^{m\ -\ n} * a + b)\ (int\ n)$

thm Float.FLOAT_ADD_PPv2:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ m < n \longrightarrow float\ a\ (int\ m) + float\ b\ (int\ n) = float\ (int\ (2::nat)^{n\ -\ m} * b + a)\ (int\ m)$

thm Float.FLOAT_ADD_NN:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ n \le m \longrightarrow float\ a\ (-\ int\ m) + float\ b\ (-\ int\ n) = float\ (int\ (2::nat)^{m\ -\ n} * b + a)\ (-\ int\ m)$

thm Float.FLOAT_ADD_NNv2:

$\forall\, (a::int)\ (b::int)\ (m::nat)\ n::nat.\ m < n \longrightarrow float\ a\ (-\ int\ m) + float\ b\ (-\ int\ n) = float\ (int\ (2::nat)^{n\ -\ m} * a + b)\ (-\ int\ n)$

thm Float.FLOAT_SUB:

$\forall\, (a::int)\ (b::int)\ (n::int)\ m::int.\ float\ a\ n - float\ b\ m = float\ a\ n + float\ (-\ b)\ m$

thm Float.FLOAT_ABS:

$\forall (a::int)\ n::int.\ |float\ a\ n| = float\ |a|\ n$

thm Float.FLOAT_POW:

$\forall (a::int)\ (n::int)\ m::nat.\ (float\ a\ n)^m = float\ a^m\ (n * int\ m)$

thm Float.INT_SUB:

$\forall (a::int)\ b::int.\ a - b = a + - b$

thm Float.INT_ABS_NEG_NUM:

$\forall n::nat.\ |- int\ n| = int\ n$

thm INT_ADD_AC_conjunct2:

$(?m::int) + ((?n::int) + (?p::int)) = ?n + (?m + ?p)$

thm INT_ADD_AC_conjunct1:

$(?m::int) + (?n::int) + (?p::int) = ?m + (?n + ?p)$

thm Float.INT_ADD_NEG_NUM:

$\forall (x::nat)\ y::nat.\ - int\ x + int\ y = int\ y + - int\ x$

thm Float.INT_POW_NEG1:

$\forall (x::nat)\ n::nat.\ (- int\ x)^n = (- int\ (1::nat))^n * (int\ x)^n$

thm Float.INT_POW_EVEN_NEG1:

$\forall (x::nat)\ n::nat.\ (- int\ x)^{NUM\ (bit0\ n)} = (int\ x)^{NUM\ (bit0\ n)}$

thm Float.POW_MINUS1:

$\forall n::nat.\ (- (1::real))^{(2::nat) * n} = (1::real)$

thm Float.INT_POW_ODD_NEG1:

$\forall (x::nat)\ n::nat.\ (- int\ x)^{NUM\ (bit1\ n)} = - (int\ x)^{NUM\ (bit1\ n)}$

thm Float.INT_ADD_NEG:

$\forall (x::nat)\ y::nat.\ x < y \longrightarrow int\ x + - int\ y = - int\ (y - x)$

thm Float.INT_ADD_NEGv2:

$\forall (x::nat)\ y::nat.\ y \leq x \longrightarrow int\ x + - int\ y = int\ (x - y)$

thm Float.FLOAT_EQ:

$\forall (a::int)\ (b::int)\ (a'::int)\ b'::int.\ (float\ a\ b = float\ a'\ b') = (float\ a\ b - float\ a'\ b' = (0::real))$

thm Float.FLOAT_LT:

$\forall (a::int)\ (b::int)\ (a'::int)\ b'::int.\ (float\ a\ b < float\ a'\ b') = ((0::real) < float\ a'\ b' - float\ a\ b)$

thm Float.FLOAT_LE:

$\forall (a::int)\ (b::int)\ (a'::int)\ b'::int.\ (float\ a\ b \le float\ a'\ b') = ((0::real) \le float\ a'\ b' - float\ a\ b)$

thm Float.REAL_ADD_LE_SUBST_RHS:

$\forall (a::real)\ (b::real)\ (c::real)\ P::real \Rightarrow real.\ a \le P\ b \wedge (\forall x::real.\ P\ x = x + P\ (0::real)) \wedge b \le c \longrightarrow a \le P\ c$

thm Float.REAL_ADD_LE_SUBST_LHS:

$\forall (a::real)\ (b::real)\ (c::real)\ P::real \Rightarrow real.\ P\ a \le b \wedge (\forall x::real.\ P\ x = x + P\ (0::real)) \wedge c \le a \longrightarrow P\ c \le b$

thm Float.INTERVAL_ADD:

$\forall (x::real)\ (f::real)\ (ex::real)\ (y::real)\ (g::real)\ ey::real.\ interval\_eps\ x\ f\ ex \wedge interval\_eps\ y\ g\ ey \longrightarrow interval\_eps\ (x + y)\ (f + g)\ (ex + ey)$

thm Float.INTERVAL_NEG:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex = interval\_eps\ (-\ x)\ (-\ f)\ ex$

thm Float.INTERVAL_NEG2:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ (-\ x)\ f\ ex = interval\_eps\ x\ (-\ f)\ ex$

thm Float.INTERVAL_SUB:

$\forall (x::real)\ (f::real)\ (ex::real)\ (y::real)\ (g::real)\ ey::real.\ interval\_eps\ x\ f\ ex \wedge interval\_eps\ y\ g\ ey \longrightarrow interval\_eps\ (x - y)\ (f - g)\ (ex + ey)$

thm Float.REAL_PROP_LE_RABS:

$\forall (x::real)\ (y::real)\ z::real.\ y \le z \longrightarrow y * |x| \le z * |x|$

thm Float.REAL_LE_ABS_MUL:

$\forall (x::real)\ (y::real)\ (z::real)\ w::real.\ x \le y \wedge |z| \le w \longrightarrow x * w \le y * w$

thm Float.INTERVAL_ABS:

$\forall (x::real)\ (z::real)\ d::real.\ (x - d \le z \wedge z \le x + d) = (|z - x| \le d)$

thm Float.INTERVAL_MUL:

$\forall (x::real)\ (f::real)\ (ex::real)\ (y::real)\ (g::real)\ ey::real.\ interval\_eps\ x\ f\ ex \wedge interval\_eps\ y\ g\ ey \longrightarrow interval\_eps\ (x * y)\ (f * g)\ (|f| * ey + (ex * |g| + ex * ey))$

thm Float.INTERVAL_NUM:

$\forall n::nat.\ interval\_eps\ (real\_of\_nat\ n)\ (float\ (int\ n)\ (int\ (0::nat)))\ (float\ (int\ (0::nat))\ (int\ (0::nat)))$

thm Float.INTERVAL_CENTER:

$\forall (x::real)\ (f::real)\ (ex::real)\ g::real.\ interval\_eps\ x\ f\ ex \longrightarrow interval\_eps\ x\ g\ (|f - g| + ex)$

thm Float.INTERVAL_WIDTH:

$\forall (x::real)\ (f::real)\ (ex::real)\ ex'::real.\ ex \leq ex' \longrightarrow interval\_eps\ x\ f\ ex \longrightarrow interval\_eps\ x\ f\ ex'$

thm Float.INTERVAL_MAX:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow x \leq f + ex$

thm Float.INTERVAL_MIN:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow f - ex \leq x$

thm Float.INTERVAL_ABS_MIN:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow |f| - ex \leq |x|$

thm Float.INTERVAL_ABS_MAX:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow |x| \leq |f| + ex$

thm Float.INTERVAL_MK:

$LET\ (\lambda half::real.\ LET\_END\ (\forall (x::real)\ (xmin::real)\ xmax::real.\ xmin \leq x \wedge x \leq xmax \longrightarrow interval\_eps\ x\ ((xmin + xmax) * half)\ ((xmax - xmin) * half)))\ (float\ (int\ (1::nat))\ (-\ int\ (1::nat)))$

thm Float.INTERVAL_EPS_POS:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow (0::real) \leq ex$

thm Float.INTERVAL_EPS_0:

$\forall (x::real)\ (f::real)\ n::int.\ interval\_eps\ x\ f\ (float\ (int\ (0::nat))\ n) \longrightarrow x = f$

thm Float.REAL_EQ_RCANCEL_IMP′:

$\forall (x::real)\ (y::real)\ z::real.\ x * z = y * z \longrightarrow z \neq (0::real) \longrightarrow x = y$

thm Float.REAL_MK_POS_ABS_′:

$\forall x::real.\ x \neq (0::real) \longrightarrow (0::real) < |x|$

thm Float.INTERVAL_DIV:

$\forall (x::real)\ (f::real)\ (ex::real)\ (y::real)\ (g::real)\ (ey::real)\ (h::real)\ ez::real.\ interval\_eps\ x\ f\ ex \wedge interval\_eps\ y\ g\ ey \wedge ey < |g| \wedge ex + (|f - h * g| + |h| * ey) \leq ez * (|g| - ey) \longrightarrow interval\_eps\ (x\ /\ y)\ h\ ez$

thm Float.INTERVAL_ABSV:

$\forall (x::real)\ (f::real)\ ex::real.\ interval\_eps\ x\ f\ ex \longrightarrow interval\_eps\ |x|\ |f|\ ex$

thm Float.taylor_atn:

$\forall (n::nat)\ (x::real)\ (v::real)\ (eps2::real)\ eps::real.\ interval\_eps\ (real\_of\_nat\ (16::nat) * sum\ (dotdot\ (0::nat)\ n)\ (halfatn4\_co\ x))\ v\ eps2 \wedge float\ (int\ (1::nat))\ (-\ int\ ((6::nat) * n + (5::nat))) + eps2 \leq eps \longrightarrow interval\_eps\ (atn\ x)\ v\ eps$

thm Float.pi_atn:

$pi = real\_of\_nat\ (4::nat) * atn\ (1::real)$

thm Float.ssqrt_sqrt:

$\forall (x::real)\ (u::real)\ f::real.\ (0::real) < x \wedge interval\_eps\ (ssqrt\ x)\ u\ f \longrightarrow interval\_eps\ (sqrt\ x)\ u\ f$

thm Float.REAL_SSQRT_NEG:

$\forall x{<}0::real.\ ssqrt\ x = (0::real)$

thm Float.REAL_SSQRT_NN:

$\forall x{\geq}0::real.\ ssqrt\ x = sqrt\ x$

thm Float.REAL_MK_NN_SSQRT:

$\forall x::real.\ (0::real) \leq ssqrt\ x$

thm Float.REAL_SV_SSQRT_0:

$\forall x::?'a::type.\ ssqrt\ (0::real) = (0::real)$

thm Float.REAL_SV_SSQRT_n:

$\forall n::nat.\ ssqrt\ (real\_of\_nat\ n) = sqrt\ (real\_of\_nat\ n)$

thm Float.REAL_SSQRT_EQ_0:

$\forall x::real.\ ssqrt\ x = (0::real) \longrightarrow x \leq (0::real)$

thm Float.REAL_SSQRT_MONO:

$\forall x{\leq}?y::real.\ ssqrt\ x \leq ssqrt\ ?y$

thm Float.REAL_SSQRT_CHAR:

$\forall (x::real)\ t::real.\ (0::real) \leq t \wedge t * t = x \longrightarrow t = ssqrt\ x$

thm Float.REAL_SSQRT_SQUARE:

$\forall x{\geq}0::real.\ ssqrt\ x * ssqrt\ x = x$

thm Float.REAL_SSQRT_SQUARE′:

$\forall x{\geq}0::real.\ ssqrt\ (x * x) = x$

thm Float.INTERVAL_SSQRT:

$\forall (x::real)\ (f::real)\ (ex::real)\ (u::real)\ (ey::real)\ (ez::real)\ v::real.\ interval\_eps\ x\ f\ ex \wedge interval\_eps\ (u * u)\ f\ ey \wedge ex + ey \leq ez * (v + u) \wedge v * v \leq f - ex \wedge (0::real) < u \wedge (0::real) \leq v \longrightarrow interval\_eps\ (ssqrt\ x)\ u\ ez$

thm Float.TWOPOW_MK_POS:

$\forall a{::}int.\ (0{::}real) < twopow\ a$

thm Float.TWOPOW_NZ:

$\forall a{::}int.\ twopow\ a \neq (0{::}real)$

thm Float.FLOAT_ZERO:

$\forall (a{::}int)\ b{::}int.\ (float\ a\ b = (0{::}real)) = (a = int\ (0{::}nat))$

thm Float.INT_ZERO:

$\forall n{::}nat.\ (int\ n = int\ (0{::}nat)) = (n = (0{::}nat))$

thm Float.INT_ZERO_NEG:

$\forall n{::}nat.\ (-\ int\ n = int\ (0{::}nat)) = (n = (0{::}nat))$

thm Float.FLOAT_NN:

$\forall (a{::}int)\ b{::}int.\ ((0{::}real) \leq float\ a\ b) = (int\ (0{::}nat) \leq a)$

thm Float.INT_NN_NEG:

$\forall n{::}nat.\ (int\ (0{::}nat) \leq -\ int\ n) = (n = (0{::}nat))$

thm Float.FLOAT_POS:

$\forall (a{::}int)\ b{::}int.\ ((0{::}real) < float\ a\ b) = (int\ (0{::}nat) < a)$

thm Float.INT_POS′:

$\forall n{::}nat.\ (int\ (0{::}nat) < int\ n) = (n \neq (0{::}nat))$

thm Float.INT_POS_NEG:

$\forall n{::}nat.\ (int\ (0{::}nat) < -\ int\ n) = False$

thm Float.RAT_LEMMA1_SUB:

$(?y1.0{::}real) \neq (0{::}real) \wedge (?y2.0{::}real) \neq (0{::}real) \longrightarrow (?x1.0{::}real)\ /\ ?y1.0 - (?x2.0{::}real)\ /\ ?y2.0 = (?x1.0 * ?y2.0 - ?x2.0 * ?y1.0) * (inverse\_class.inverse\ ?y1.0 * inverse\_class.inverse\ ?y2.0)$

thm Float.INTERVAL_0:

$\forall (a{::}real)\ (f{::}real)\ ex{::}real.\ interval\_eps\ a\ f\ ex = ((0{::}real) \leq ex - |a - f|)$

thm Float.ABS_NUM_P:

$\forall (m{::}nat)\ n{::}nat.\ m \leq n \longrightarrow |real\_of\_nat\ n - real\_of\_nat\ m| = real\_of\_nat\ (m - n + (n - m))$

thm Float.ABS_NUM:

$\forall (m{::}nat)\ n{::}nat.\ |real\_of\_nat\ n - real\_of\_nat\ m| = real\_of\_nat\ (m - n + (n - m))$

thm Float.INTERVAL_TO_LESS:

$\forall (a::real)\ (f::real)\ (ex::real)\ (b::real)\ (g::real)\ ey::real.\ interval\_eps\ a\ f\ ex \wedge interval\_eps\ b\ g\ ey \wedge (0::real) < g - (ey + (ex + f)) \longrightarrow a < b$

thm Float.ABS_TO_INTERVAL:

$\forall (c::real)\ (u::real)\ k::real.\ |c - u| \leq k \longrightarrow (\forall (f::real)\ (g::real)\ (ex::real)\ ey::real.\ interval\_eps\ u\ f\ ex \wedge interval\_eps\ k\ g\ ey \longrightarrow interval\_eps\ c\ f\ (g + (ey + ex)))$

thm Float.lemma1:

$\forall (n::nat)\ (m::nat)\ p::nat.\ (real\_of\_nat\ p\ /\ real\_of\_nat\ m \leq real\_of\_nat\ n) = (real\_of\_nat\ p\ /\ real\_of\_nat\ m \leq real\_of\_nat\ n\ /\ (1::real))$

thm Float.lemma2:

$\forall (n::nat)\ (m::nat)\ p::nat.\ (real\_of\_nat\ p \leq real\_of\_nat\ n\ /\ real\_of\_nat\ m) = (real\_of\_nat\ p\ /\ (1::real) \leq real\_of\_nat\ n\ /\ real\_of\_nat\ m)$

thm Float.lemma3:

$\forall (a::nat)\ (b::nat)\ (c::nat)\ d::nat.\ (0::nat) < b \wedge (0::nat) < d \wedge a * d \leq c * b \longrightarrow real\_of\_nat\ a\ /\ real\_of\_nat\ b \leq real\_of\_nat\ c\ /\ real\_of\_nat\ d$

thm Float.pow_parity:

$\forall (x::real)\ u::nat.\ x^{NUM\ (bit0\ u)} = x^{NUM\ u} * x^{NUM\ u} \wedge x^{NUM\ (bit1\ u)} = x * (x^{NUM\ u} * x^{NUM\ u})$

thm Float.INTERVAL_MINMAX:

$\forall (x::real)\ (f::real)\ ex::real.\ f - ex \leq x \wedge x \leq f + ex \longrightarrow interval\_eps\ x\ f\ ex$

thm Float.INTERVAL_FLOAT:

$\forall (a::int)\ b::int.\ interval\_eps\ (float\ a\ b)\ (float\ a\ b)\ (float\ (int\ (0::nat))\ (int\ (0::nat)))$

thm Float.INTERVAL_OF_TWOPOW:

$\forall n::int.\ interval\_eps\ (twopow\ n)\ (float\ (int\ (1::nat))\ n)\ (float\ (int\ (0::nat))\ (int\ (0::nat)))$

thm Nonlinear_lemma.sqrt2_lb:

$DECIMAL\ (1414213::nat)\ (1000000::nat) < sqrt2$

thm Flyspeck_constants.bounds_conjunct1:

$sqrt2 < DECIMAL\ (1414214::nat)\ (1000000::nat)$

thm Flyspeck_constants.bounds_conjunct2:

$DECIMAL\ (2828427::nat)\ (1000000::nat) < sqrt8$

thm Flyspeck_constants.bounds_conjunct3:

$sqrt8 < DECIMAL\ (2828428::nat)\ (1000000::nat)$

thm Flyspeck_constants.bounds_conjunct4:
DECIMAL (*1732*::*nat*) (*1000*::*nat*) < *sqrt3*

thm Flyspeck_constants.bounds_conjunct5:
sqrt3 < *DECIMAL* (*17321*::*nat*) (*10000*::*nat*)

thm Flyspeck_constants.bounds_conjunct6:
DECIMAL (*314159*::*nat*) (*100000*::*nat*) < *pi*

thm Flyspeck_constants.bounds_conjunct7:
pi < *DECIMAL* (*31416*::*nat*) (*10000*::*nat*)

thm Flyspeck_constants.bounds_conjunct8:
DECIMAL (*551285*::*nat*) (*1000000*::*nat*) < *sol0*

thm Flyspeck_constants.bounds_conjunct9:
sol0 < *DECIMAL* (*551286*::*nat*) (*1000000*::*nat*)

thm Flyspeck_constants.bounds_conjunct10:
DECIMAL (*154065*::*nat*) (*100000*::*nat*) < *tau0*

thm Flyspeck_constants.bounds_conjunct11:
tau0 < *DECIMAL* (*154066*::*nat*) (*100000*::*nat*)

thm Flyspeck_constants.bounds_conjunct12:
DECIMAL (*740480*::*nat*) (*1000000*::*nat*) < *pi_rt18*

thm Flyspeck_constants.bounds_conjunct13:
pi_rt18 < *DECIMAL* (*740481*::*nat*) (*1000000*::*nat*)

thm Flyspeck_constants.bounds_conjunct14:
DECIMAL (*1012080*::*nat*) (*1000000*::*nat*) < *mm1*

thm Flyspeck_constants.bounds_conjunct15:
mm1 < *DECIMAL* (*1012081*::*nat*) (*1000000*::*nat*)

thm Flyspeck_constants.bounds_conjunct16:
DECIMAL (*2541*::*nat*) (*100000*::*nat*) < *mm2*

thm Flyspeck_constants.bounds_conjunct17:
mm2 < *DECIMAL* (*2542*::*nat*) (*100000*::*nat*)

thm Flyspeck_constants.bounds_conjunct18:
(*0*::*real*) < (*sqrt* (*real_of_nat* (*2*::*nat*)) − (*1*::*real*)) ∗ (*real_of_nat* (*5*::*nat*) ∗ (*hplus* − (*1*::*real*)))

thm Flyspeck_constants.bounds_conjunct19:

$(0::real) < sqrt\ (real\_of\_nat\ (2::nat)) - hplus$

thm Flyspeck_constants.bounds_conjunct20:

$(0::real) < hplus - (1::real)$

thm Flyspeck_constants.bounds:

$DECIMAL\ (1414213::nat)\ (1000000::nat) < sqrt2 \wedge sqrt2 < DECIMAL\ (1414214::nat)\ (1000000::nat) \wedge DECIMAL\ (2828427::nat)\ (1000000::nat) < sqrt8 \wedge sqrt8 < DECIMAL\ (2828428::nat)\ (1000000::nat) \wedge DECIMAL\ (1732::nat)\ (1000::nat) < sqrt3 \wedge sqrt3 < DECIMAL\ (17321::nat)\ (10000::nat) \wedge DECIMAL\ (314159::nat)\ (100000::nat) < pi \wedge pi < DECIMAL\ (31416::nat)\ (10000::nat) \wedge DECIMAL\ (551285::nat)\ (1000000::nat) < sol0 \wedge sol0 < DECIMAL\ (551286::nat)\ (1000000::nat) \wedge DECIMAL\ (154065::nat)\ (100000::nat) < tau0 \wedge tau0 < DECIMAL\ (154066::nat)\ (100000::nat) \wedge DECIMAL\ (740480::nat)\ (1000000::nat) < pi\_rt18 \wedge pi\_rt18 < DECIMAL\ (740481::nat)\ (1000000::nat) \wedge DECIMAL\ (1012080::nat)\ (1000000::nat) < mm1 \wedge mm1 < DECIMAL\ (1012081::nat)\ (1000000::nat) \wedge DECIMAL\ (2541::nat)\ (100000::nat) < mm2 \wedge mm2 < DECIMAL\ (2542::nat)\ (100000::nat) \wedge (0::real) < (sqrt\ (real\_of\_nat\ (2::nat)) - (1::real)) * (real\_of\_nat\ (5::nat) * (hplus - (1::real))) \wedge (0::real) < sqrt\ (real\_of\_nat\ (2::nat)) - hplus \wedge (0::real) < hplus - (1::real)$

thm DEF_dirac_delta:

$dirac\_delta = (\lambda(\_2116935::?'a::type)\ j::?'a::type.\ if\ \_2116935 = j\ then\ 1::real\ else\ (0::real))$

thm Misc_defs_and_lemmas.dirac_delta:

$\forall i::?'a::type.\ dirac\_delta\ i = (\lambda j::?'a::type.\ if\ i = j\ then\ 1::real\ else\ (0::real))$

thm DEF_min_num:

$min\_num = (\lambda\_2116940::nat \Rightarrow bool.\ SOME\ m::nat.\ IN\ m\ \_2116940 \wedge (\forall n::nat.\ IN\ n\ \_2116940 \longrightarrow m \leq n))$

thm Misc_defs_and_lemmas.min_num:

$\forall X::nat \Rightarrow bool.\ min\_num\ X = (SOME\ m::nat.\ IN\ m\ X \wedge (\forall n::nat.\ IN\ n\ X \longrightarrow m \leq n))$

thm Misc_defs_and_lemmas.min_least:

$\forall (X::nat \Rightarrow bool)\ c::nat.\ X\ c \longrightarrow X\ (min\_num\ X) \wedge min\_num\ X \leq c$

thm DEF_min_real:

$min\_real = (\lambda(\_2117129::real)\ \_2117130::real.\ if\ \_2117129 < \_2117130\ then\ \_2117129\ else\ \_2117130)$

thm Misc_defs_and_lemmas.min_real:

$\forall (x::real)\ y::real.\ min\_real\ x\ y = (if\ x < y\ then\ x\ else\ y)$

thm Misc_defs_and_lemmas.square_le:

$\forall (x::real)\ y::real.\ (0::real) \leq x \wedge (0::real) \leq y \wedge x * x \leq y * y \longrightarrow x \leq y$

thm Misc_defs_and_lemmas.max_num_sequence:

$\forall t::nat \Rightarrow nat.\ (\exists n::nat.\ \forall m \geq n.\ t\ m = (0::nat)) \longrightarrow (\exists M::nat.\ \forall i::nat.\ t\ i \leq M)$

thm Misc_defs_and_lemmas.REAL_INV_LT:

$\forall (x::real)\ (y::real)\ z::real.\ (0::real) < x \longrightarrow (inverse\_class.inverse\ x * y < z) = (y < x * z)$

thm Misc_defs_and_lemmas.REAL_MUL_NN:

$\forall (x::real)\ y::real.\ ((0::real) \leq x * y) = ((0::real) \leq x \wedge (0::real) \leq y \vee x \leq (0::real) \wedge y \leq (0::real))$

thm Misc_defs_and_lemmas.ABS_SQUARE:

$\forall (t::real)\ u::real.\ |t| < u \longrightarrow t * t < u * u$

thm Misc_defs_and_lemmas.ABS_SQUARE_LE:

$\forall (t::real)\ u::real.\ |t| \leq u \longrightarrow t * t \leq u * u$

thm Misc_defs_and_lemmas.POW_2_LT:

$\forall n::nat.\ real\_of\_nat\ n < (real\_of\_nat\ (2::nat))^n$

thm Misc_defs_and_lemmas.twopow_eps:

$\forall (R::real)\ e::real.\ \exists n::nat.\ (0::real) < R \wedge (0::real) < e \longrightarrow R * twopow\ (-\ int\ n) < e$

thm Misc_defs_and_lemmas.prod_EXISTS:

$\exists prod::nat \times nat \Rightarrow (nat \Rightarrow real) \Rightarrow real.\ (\forall (f::nat \Rightarrow real)\ n::nat.\ prod\ (n,\ 0::nat)\ f = (1::real)) \wedge (\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ prod\ (n,\ Suc\ m)\ f = prod\ (n,\ m)\ f * f\ (n + m))$

thm DEF_prod:

$prod = (SOME\ prod::nat \Rightarrow nat \times nat \Rightarrow (nat \Rightarrow real) \Rightarrow real.\ \forall\ \_2117739::nat.\ (\forall (f::nat \Rightarrow real)\ n::nat.\ prod\ \_2117739\ (n,\ 0::nat)\ f = (1::real)) \wedge (\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ prod\ \_2117739\ (n,\ Suc\ m)\ f = prod\ \_2117739\ (n,\ m)\ f * f\ (n + m)))\ (77::nat)$

thm Misc_defs_and_lemmas.prod_DEF:

$(\forall (f::nat \Rightarrow real)\ n::nat.\ prod\ (n,\ 0::nat)\ f = (1::real)) \wedge (\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ prod\ (n,\ Suc\ m)\ f = prod\ (n,\ m)\ f * f\ (n + m))$

thm Misc_defs_and_lemmas.prod_DEF_conjunct1:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ prod\ (n,\ Suc\ m)\ f = prod\ (n,\ m)\ f * f\ (n + m)$

thm Misc_defs_and_lemmas.prod_DEF_conjunct0:

$\forall (f::nat \Rightarrow real)\ n::nat.\ prod\ (n,\ 0::nat)\ f = (1::real)$

thm Misc_defs_and_lemmas.prod:

$\forall (n::nat)\ m::nat.\ prod\ (n,\ 0::nat)\ (?f::nat \Rightarrow real) = (1::real) \wedge prod\ (n,\ Suc\ m)\ ?f = prod\ (n,\ m)\ ?f * ?f\ (n + m)$

thm Misc_defs_and_lemmas.PROD_TWO:

$\forall (f::nat \Rightarrow real)\ (n::nat)\ p::nat.\ prod\ (0::nat,\ n)\ f * prod\ (n,\ p)\ f = prod\ (0::nat,\ n + p)\ f$

thm Misc_defs_and_lemmas.ABS_PROD:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ |prod\ (m,\ n)\ f| = prod\ (m,\ n)\ (\lambda n::nat.\ |f\ n|)$

thm Misc_defs_and_lemmas.PROD_EQ:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall r::nat.\ m \leq r \wedge r < n + m \longrightarrow f\ r = g\ r) \longrightarrow prod\ (m,\ n)\ f = prod\ (m,\ n)\ g$

thm Misc_defs_and_lemmas.PROD_POS:

$\forall f::nat \Rightarrow real.\ (\forall n::nat.\ (0::real) \leq f\ n) \longrightarrow (\forall (m::nat)\ n::nat.\ (0::real) \leq prod\ (m,\ n)\ f)$

thm Misc_defs_and_lemmas.PROD_POS_GEN:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\forall n \geq m.\ (0::real) \leq f\ n) \longrightarrow (0::real) \leq prod\ (m,\ n)\ f$

thm Misc_defs_and_lemmas.PROD_ABS:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ |prod\ (m,\ n)\ (\lambda m::nat.\ |f\ m|)| = prod\ (m,\ n)\ (\lambda m::nat.\ |f\ m|)$

thm Misc_defs_and_lemmas.PROD_ZERO:

$\forall (f::nat \Rightarrow real)\ (m::nat)\ n::nat.\ (\exists p \geq m.\ p < n + m \wedge f\ p = (0::real)) \longrightarrow prod\ (m,\ n)\ f = (0::real)$

thm Misc_defs_and_lemmas.PROD_MUL:

$\forall (f::nat \Rightarrow real)\ (g::nat \Rightarrow real)\ (m::nat)\ n::nat.\ prod\ (m,\ n)\ (\lambda n::nat.\ f\ n * g\ n) = prod\ (m,\ n)\ f * prod\ (m,\ n)\ g$

thm Misc_defs_and_lemmas.PROD_CMUL:

$\forall (f::nat \Rightarrow real)\ (c::real)\ (m::nat)\ n::nat.\ prod\ (m,\ n)\ (\lambda n::nat.\ c * f\ n) = c^n * prod\ (m,\ n)\ f$

thm Misc_defs_and_lemmas.SURJ_IMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ SURJ\ f\ a\ b \longrightarrow b = IMAGE\ f\ a$

thm Misc_defs_and_lemmas.SURJ_FINITE:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ a \wedge SURJ\ f\ a\ b \longrightarrow FINITE\ b$

thm Misc_defs_and_lemmas.BIJ_INVERSE:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ SURJ\ f\ a\ b \longrightarrow (\exists\ g::?'a::type \Rightarrow ?'b::type.\ INJ\ g\ b\ a)$

thm Misc_defs_and_lemmas.UNIONS_INTERS:

$\forall (X::?'a::type \Rightarrow bool)\ V::(?'a::type \Rightarrow bool) \Rightarrow bool.\ DIFF\ X\ (INTERS\ V) = UNIONS\ (IMAGE\ (DIFF\ X)\ V)$

thm Misc_defs_and_lemmas.INTERS_SUBSET:

$\forall (X::(?'a::type \Rightarrow bool) \Rightarrow bool)\ A::?'a::type \Rightarrow bool.\ IN\ A\ X \longrightarrow SUBSET\ (INTERS\ X)\ A$

thm Misc_defs_and_lemmas.sub_union:

$\forall (X::?'a::type \Rightarrow bool)\ U::(?'a::type \Rightarrow bool) \Rightarrow bool.\ U\ X \longrightarrow SUBSET\ X\ (UNIONS\ U)$

thm Misc_defs_and_lemmas.IMAGE_SURJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ a::?'b::type \Rightarrow bool.\ SURJ\ f\ a\ (IMAGE\ f\ a)$

thm Misc_defs_and_lemmas.SUBSET_PREIMAGE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (X::?'b::type \Rightarrow bool)\ Y::?'a::type \Rightarrow bool.\ SUBSET\ Y\ (IMAGE\ f\ X) \longrightarrow (\exists\ Z::?'b::type \Rightarrow bool.\ SUBSET\ Z\ X \wedge Y = IMAGE\ f\ Z)$

thm Misc_defs_and_lemmas.UNIONS_INTER:

$\forall (U::(?'a::type \Rightarrow bool) \Rightarrow bool)\ A::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (UNIONS\ U)\ A = UNIONS\ (IMAGE\ (HOL\_Light\_Import.INTER\ A)\ U)$

thm Misc_defs_and_lemmas.UNIONS_SUBSET:

$\forall (U::(?'a::type \Rightarrow bool) \Rightarrow bool)\ X::?'a::type \Rightarrow bool.\ (\forall\ A::?'a::type \Rightarrow bool.\ IN\ A\ U \longrightarrow SUBSET\ A\ X) \longrightarrow SUBSET\ (UNIONS\ U)\ X$

thm Misc_defs_and_lemmas.SUBSET_INTER:

$\forall (X::?'a::type \Rightarrow bool)\ (A::?'a::type \Rightarrow bool)\ B::?'a::type \Rightarrow bool.\ SUBSET\ X\ (HOL\_Light\_Import.INTER\ A\ B) = (SUBSET\ X\ A \wedge SUBSET\ X\ B)$

thm Misc_defs_and_lemmas.EMPTY_EXISTS:

$\forall\ X::?'a::type \Rightarrow bool.\ (X \neq EMPTY) = (\exists\ u::?'a::type.\ IN\ u\ X)$

thm Misc_defs_and_lemmas.UNIONS_UNIONS:

$\forall (A::(?'a::type \Rightarrow bool) \Rightarrow bool)\ B::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ A\ B \longrightarrow SUBSET\ (UNIONS\ A)\ (UNIONS\ B)$

thm Misc_defs_and_lemmas.UNIONS_IMAGE_UNIONS:

$\forall X::((?'a::type \Rightarrow bool) \Rightarrow bool) \Rightarrow bool.\ UNIONS\ (UNIONS\ X) = UNIONS\ (IMAGE\ UNIONS\ X)$

thm Misc_defs_and_lemmas.INTERS_SUBSET2:

$\forall (X::?'a::type \Rightarrow bool)\ A::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\exists x::?'a::type \Rightarrow bool.\ A\ x \wedge SUBSET\ x\ X) \longrightarrow SUBSET\ (INTERS\ A)\ X$

thm DEF_preimage:

$preimage = (\lambda(\_2118589::?'b::type \Rightarrow bool)\ (\_2118590::?'b::type \Rightarrow ?'a::type)\ \_2118591::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%106::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%106\ (IN\ x\ \_2118589 \wedge IN\ (\_2118590\ x)\ \_2118591)\ x))$

thm Misc_defs_and_lemmas.preimage:

$\forall (dom::?'b::type \Rightarrow bool)\ (f::?'b::type \Rightarrow ?'a::type)\ Z::?'a::type \Rightarrow bool.\ preimage\ dom\ f\ Z = GSPEC\ (\lambda GEN\%PVAR\%106::?'b::type.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%106\ (IN\ x\ dom \wedge IN\ (f\ x)\ Z)\ x)$

thm Misc_defs_and_lemmas.in_preimage:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (x::?'b::type)\ (Z::?'a::type \Rightarrow bool)\ dom::?'b::type \Rightarrow bool.\ IN\ x\ (preimage\ dom\ f\ Z) = (IN\ x\ dom \wedge IN\ (f\ x)\ Z)$

thm DEF_supp:

$supp = (\lambda(\_2118610::?'b::type \Rightarrow ?'a::type)\ x::?'b::type.\ \_2118610\ x \neq CHOICE\ HOL\_Light\_Import.UNIV)$

thm Misc_defs_and_lemmas.supp:

$\forall f::?'b::type \Rightarrow ?'a::type.\ supp\ f = (\lambda x::?'b::type.\ f\ x \neq CHOICE\ HOL\_Light\_Import.UNIV)$

thm DEF_func:

$func = (\lambda(\_2118615::?'b::type \Rightarrow bool)\ (\_2118616::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ (\forall x::?'b::type.\ IN\ x\ \_2118615 \longrightarrow IN\ (f\ x)\ \_2118616) \wedge SUBSET\ (supp\ f)\ \_2118615)$

thm Misc_defs_and_lemmas.func:

$\forall (b::?'b::type \Rightarrow bool)\ a::?'a::type \Rightarrow bool.\ func\ a\ b = (\lambda f::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'a::type.\ IN\ x\ a \longrightarrow IN\ (f\ x)\ b) \wedge SUBSET\ (supp\ f)\ a)$

thm DEF_reflexive:

$reflexive = (\lambda\_2118627::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ \forall x::?'a::type.\ \_2118627\ x\ x)$

thm Misc_defs_and_lemmas.reflexive:

$\forall f::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ reflexive\ f = (\forall x::?'a::type.\ f\ x\ x)$

thm DEF_symmetric:

symmetric = (λ*_2118632*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. ∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *_2118632 x y* ⟶ *_2118632 y x*)

thm Misc_defs_and_lemmas.symmetric:

∀ *f*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *symmetric f* = (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *f x y* ⟶ *f y x*)

thm DEF_transitive:

transitive = (λ*_2118637*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. ∀ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *z*::*?'a*::*type*. *_2118637 x y* ∧ *_2118637 y z* ⟶ *_2118637 x z*)

thm Misc_defs_and_lemmas.transitive:

∀ *f*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *transitive f* = (∀ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *z*::*?'a*::*type*. *f x y* ∧ *f y z* ⟶ *f x z*)

thm DEF_equivalence_relation:

equivalence_relation = (λ*_2118642*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *reflexive _2118642* ∧ *symmetric _2118642* ∧ *transitive _2118642*)

thm Misc_defs_and_lemmas.equivalence_relation:

∀ *f*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *equivalence_relation f* = (*reflexive f* ∧ *symmetric f* ∧ *transitive f*)

thm DEF_partition:

HOL_Light_Import.partition = (λ(*_2118647*::*?'a*::*type* ⇒ *bool*) *_2118648*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *UNIONS _2118648* = *_2118647* ∧ (∀ (*a*::*?'a*::*type* ⇒ *bool*) *b*::*?'a*::*type* ⇒ *bool*. *IN a _2118648* ∧ *IN b _2118648* ∧ *a* ≠ *b* ⟶ *EMPTY* = *HOL_Light_Import.INTER a b*))

thm Misc_defs_and_lemmas.partition_DEF:

∀ (*A*::*?'a*::*type* ⇒ *bool*) *SA*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *HOL_Light_Import.partition A SA* = (*UNIONS SA* = *A* ∧ (∀ (*a*::*?'a*::*type* ⇒ *bool*) *b*::*?'a*::*type* ⇒ *bool*. *IN a SA* ∧ *IN b SA* ∧ *a* ≠ *b* ⟶ *EMPTY* = *HOL_Light_Import.INTER a b*))

thm Misc_defs_and_lemmas.DIFF_DIFF2:

∀ (*X*::*?'a*::*type* ⇒ *bool*) *A*::*?'a*::*type* ⇒ *bool*. *SUBSET A X* ⟶ *DIFF X* (*DIFF X A*) = *A*

thm Misc_defs_and_lemmas.GSPEC_THM:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *x*::*?'a*::*type*. (∃ *y*::*?'a*::*type*. *P y* ∧ *x* = *y*) = *P x*

thm Misc_defs_and_lemmas.CARD_GE_REFL:

∀ *s*::*?'a*::*type* ⇒ *bool*. *>=_c s s*

thm Misc_defs_and_lemmas.FINITE_HAS_SIZE_LEMMA:

$\forall s$::?'a::type ⇒ bool. FINITE s ⟶ (∃ n::nat. >=_c (GSPEC (λGEN%PVAR%107::nat. ∃ x::nat. SETSPEC GEN%PVAR%107 (x < n) x)) s)

thm Misc_defs_and_lemmas.NUM2_COUNTABLE:

COUNTABLE (GSPEC (λGEN%PVAR%108::nat × nat. ∃ (x::nat) y::nat. SETSPEC GEN%PVAR%108 True (x, y)))

thm Misc_defs_and_lemmas.COUNTABLE_UNIONS:

∀ A::(?'a::type ⇒ bool) ⇒ bool. COUNTABLE A ∧ (∀ a::?'a::type ⇒ bool. IN a A ⟶ COUNTABLE a) ⟶ COUNTABLE (UNIONS A)

thm Misc_defs_and_lemmas.COUNTABLE_IMAGE:

∀ (A::?'b::type ⇒ bool) B::?'a::type ⇒ bool. COUNTABLE A ∧ (∃ f::?'b::type ⇒ ?'a::type. SUBSET B (IMAGE f A)) ⟶ COUNTABLE B

thm Misc_defs_and_lemmas.COUNTABLE_CARD:

∀ (A::?'b::type ⇒ bool) B::?'a::type ⇒ bool. COUNTABLE A ∧ >=_c A B ⟶ COUNTABLE B

thm Misc_defs_and_lemmas.COUNTABLE_NUMSEG:

∀ n::nat. COUNTABLE (GSPEC (λGEN%PVAR%109::nat. ∃ x::nat. SETSPEC GEN%PVAR%109 (x < n) x))

thm Misc_defs_and_lemmas.FINITE_COUNTABLE:

∀ A::?'a::type ⇒ bool. FINITE A ⟶ COUNTABLE A

thm Misc_defs_and_lemmas.num_SEG_UNION:

∀ i::nat. HOL_Light_Import.UNION (GSPEC (λGEN%PVAR%112::nat. ∃ u::nat. SETSPEC GEN%PVAR%112 (i < u) u)) (GSPEC (λGEN%PVAR%113::nat. ∃ m::nat. SETSPEC GEN%PVAR%113 (m ≤ i) m)) = HOL_Light_Import.UNIV

thm Misc_defs_and_lemmas.num_above_infinite:

∀ i::nat. ¬ FINITE (GSPEC (λGEN%PVAR%116::nat. ∃ u::nat. SETSPEC GEN%PVAR%116 (i < u) u))

thm Misc_defs_and_lemmas.INTER_FINITE:

∀ (s::?'a::type ⇒ bool) t::?'a::type ⇒ bool. (FINITE s ⟶ FINITE (HOL_Light_Import.INTER s t)) ∧ (FINITE t ⟶ FINITE (HOL_Light_Import.INTER s t))

thm Misc_defs_and_lemmas.num_above_finite:

∀ (i::nat) J::nat ⇒ bool. FINITE (HOL_Light_Import.INTER J (GSPEC (λGEN%PVAR%119::nat. ∃ u::nat. SETSPEC GEN%PVAR%119 (i < u) u))) ⟶ FINITE J

thm Misc_defs_and_lemmas.SUBSET_SUC:

∀ f::nat ⇒ ?'a::type ⇒ bool. (∀ i::nat. SUBSET (f i) (f (Suc i))) ⟶ (∀ (i::nat) j::nat. i ≤ j ⟶ SUBSET (f i) (f j))

thm Misc_defs_and_lemmas.SUBSET_SUC2:

$\forall f{::}nat \Rightarrow ?'a{::}type \Rightarrow bool.\ (\forall i{::}nat.\ SUBSET\ (f\ (Suc\ i))\ (f\ i)) \longrightarrow (\forall (i{::}nat)\ j{::}nat.\ i \leq j \longrightarrow SUBSET\ (f\ j)\ (f\ i))$

thm Misc_defs_and_lemmas.INFINITE_PIGEONHOLE:

$\forall (I{::}?'b{::}type \Rightarrow bool)\ (f{::}?'b{::}type \Rightarrow ?'a{::}type)\ (B{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool)\ C{::}?'a{::}type \Rightarrow bool.\ \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%125{::}?'b{::}type.\ \exists\ i{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%125\ (I\ i \wedge C\ (f\ i))\ i)) \wedge FINITE\ B \wedge SUBSET\ C\ (UNIONS\ B) \longrightarrow (\exists\ b{::}?'a{::}type \Rightarrow bool.\ B\ b \wedge \neg\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%126{::}?'b{::}type.\ \exists\ i{::}?'b{::}type.\ SETSPEC\ GEN\%PVAR\%126\ (I\ i \wedge HOL\_Light\_Import.INTER\ C\ b\ (f\ i))\ i)))$

thm Misc_defs_and_lemmas.real_FINITE:

$\forall s{::}real \Rightarrow bool.\ FINITE\ s \longrightarrow (\exists\ a{::}real.\ \forall x{::}real.\ IN\ x\ s \longrightarrow x \leq a)$

thm Misc_defs_and_lemmas.UNIONS_DELETE:

$\forall s{::}(?'a{::}type \Rightarrow bool) \Rightarrow bool.\ UNIONS\ s = UNIONS\ (DELETE\ s\ EMPTY)$

thm DEF_SUPP:

$SUPP = (\lambda(\_2119027{::}?'b{::}type \Rightarrow ?'a{::}type)\ x{::}?'b{::}type.\ \_2119027\ x \neq CHOICE\ HOL\_Light\_Import.UNIV)$

thm Misc_defs_and_lemmas.SUPP:

$\forall f{::}?'b{::}type \Rightarrow ?'a{::}type.\ SUPP\ f = (\lambda x{::}?'b{::}type.\ f\ x \neq CHOICE\ HOL\_Light\_Import.UNIV)$

thm DEF_FUN:

$FUN = (\lambda(\_2119032{::}?'b{::}type \Rightarrow bool)\ (\_2119033{::}?'a{::}type \Rightarrow bool)\ f{::}?'b{::}type \Rightarrow ?'a{::}type.\ (\forall x{::}?'b{::}type.\ IN\ x\ \_2119032 \longrightarrow IN\ (f\ x)\ \_2119033) \wedge SUBSET\ (SUPP\ f)\ \_2119032)$

thm Misc_defs_and_lemmas.FUN:

$\forall (b{::}?'b{::}type \Rightarrow bool)\ a{::}?'a{::}type \Rightarrow bool.\ FUN\ a\ b = (\lambda f{::}?'a{::}type \Rightarrow ?'b{::}type.\ (\forall x{::}?'a{::}type.\ IN\ x\ a \longrightarrow IN\ (f\ x)\ b) \wedge SUBSET\ (SUPP\ f)\ a)$

thm DEF_compose:

$compose = (\lambda(\_2119044{::}?'c{::}type \Rightarrow ?'b{::}type)\ (\_2119045{::}?'a{::}type \Rightarrow ?'c{::}type)\ x{::}?'a{::}type.\ \_2119044\ (\_2119045\ x))$

thm Misc_defs_and_lemmas.compose:

$\forall (f{::}?'c{::}type \Rightarrow ?'b{::}type)\ g{::}?'a{::}type \Rightarrow ?'c{::}type.\ compose\ f\ g = (\lambda x{::}?'a{::}type.\ f\ (g\ x))$

thm Misc_defs_and_lemmas.COMP_ASSOC:

$\forall (f{::}nat \Rightarrow nat)\ (g{::}nat \Rightarrow nat)\ h{::}nat \Rightarrow nat.\ compose\ f\ (compose\ g\ h) = compose\ (compose\ f\ g)\ h$

thm Misc_defs_and_lemmas.COMP_INJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ INJ\ f\ s\ t \wedge INJ\ g\ t\ u \longrightarrow INJ\ (compose\ g\ f)\ s\ u$

thm Misc_defs_and_lemmas.COMP_SURJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ u::?'a::type \Rightarrow bool.\ SURJ\ f\ s\ t \wedge SURJ\ g\ t\ u \longrightarrow SURJ\ (compose\ g\ f)\ s\ u$

thm Misc_defs_and_lemmas.COMP_BIJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ (g::?'b::type \Rightarrow ?'a::type)\ u::?'a::type \Rightarrow bool.\ BIJ\ f\ s\ t \wedge BIJ\ g\ t\ u \longrightarrow BIJ\ (compose\ g\ f)\ s\ u$

thm Misc_defs_and_lemmas.INVERSE_FN:

$\exists INVERSE::(?'b::type \Rightarrow ?'a::type) \Rightarrow (?'b::type \Rightarrow bool) \Rightarrow (?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'b::type.\ \forall (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ SURJ\ f\ a\ b \longrightarrow INJ\ (INVERSE\ f\ a\ b)\ b\ a \wedge (\forall x::?'a::type.\ IN\ x\ b \longrightarrow f\ (INVERSE\ f\ a\ b\ x) = x)$

thm DEF_INVERSE:

$INVERSE = (SOME\ INVERSE::nat \Rightarrow (?'b::type \Rightarrow ?'a::type) \Rightarrow (?'b::type \Rightarrow bool) \Rightarrow (?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'b::type.\ \forall (\_2119363::nat)\ (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ SURJ\ f\ a\ b \longrightarrow INJ\ (INVERSE\ \_2119363\ f\ a\ b)\ b\ a \wedge (\forall x::?'a::type.\ IN\ x\ b \longrightarrow f\ (INVERSE\ \_2119363\ f\ a\ b\ x) = x))\ (78::nat)$

thm Misc_defs_and_lemmas.INVERSE_DEF:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ SURJ\ f\ a\ b \longrightarrow INJ\ (INVERSE\ f\ a\ b)\ b\ a \wedge (\forall x::?'a::type.\ IN\ x\ b \longrightarrow f\ (INVERSE\ f\ a\ b\ x) = x)$

thm Misc_defs_and_lemmas.INVERSE_BIJ:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ BIJ\ f\ a\ b \longrightarrow BIJ\ (INVERSE\ f\ a\ b)\ b\ a$

thm Misc_defs_and_lemmas.INVERSE_XY:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ (x::?'b::type)\ y::?'a::type.\ BIJ\ f\ a\ b \wedge IN\ x\ a \wedge IN\ y\ b \longrightarrow (INVERSE\ f\ a\ b\ y = x) = (f\ x = y)$

thm Misc_defs_and_lemmas.FINITE_BIJ:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ a \wedge BIJ\ f\ a\ b \longrightarrow FINITE\ b$

thm Misc_defs_and_lemmas.FINITE_INJ:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ b \wedge INJ\ f\ a\ b \longrightarrow FINITE\ a$

thm Misc_defs_and_lemmas.FINITE_BIJ2:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ b \wedge BIJ\ f\ a\ b \longrightarrow FINITE\ a$

thm Misc_defs_and_lemmas.BIJ_CARD:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ a \wedge BIJ\ f\ a\ b \longrightarrow CARD\ a = CARD\ b$

thm Misc_defs_and_lemmas.PAIR_LEMMA:

$\forall (x::nat \times nat)\ (i::nat)\ j::nat.\ (fst\ x = i \wedge snd\ x = j) = (x = (i, j))$

thm Misc_defs_and_lemmas.CARD_SING:

$\forall u::?'a::type \Rightarrow bool.\ SING\ u \longrightarrow CARD\ u = (1::nat)$

thm Hypermap.FINITE_SINGLETON:

$\forall x::?'a::type.\ FINITE\ (INSERT\ x\ EMPTY)$

thm Misc_defs_and_lemmas.NUM_INTRO:

$\forall (f::?'a::type \Rightarrow nat)\ P::?'a::type \Rightarrow bool.\ (\forall (n::nat)\ g::?'a::type.\ f\ g = n \longrightarrow P\ g) \longrightarrow (\forall g::?'a::type.\ P\ g)$

thm Misc_defs_and_lemmas.DOMAIN_EMPTY:

$\forall b::?'b::type \Rightarrow bool.\ FUN\ EMPTY\ b = INSERT\ (\lambda u::?'a::type.\ CHOICE\ HOL\_Light\_Import.UNIV)\ EMPTY$

thm Misc_defs_and_lemmas.DOMAIN_INSERT:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ s::?'b::type.\ \neg\ IN\ s\ a \longrightarrow (\exists F::(?'b::type \Rightarrow ?'a::type) \Rightarrow (?'b::type \Rightarrow ?'a::type) \times ?'a::type.\ BIJ\ F\ (FUN\ (INSERT\ s\ a)\ b)\ (GSPEC\ (\lambda GEN\%PVAR\%129::(?'b::type \Rightarrow ?'a::type) \times ?'a::type.\ \exists (u::?'b::type \Rightarrow ?'a::type)\ v::?'a::type.\ SETSPEC\ GEN\%PVAR\%129\ (IN\ u\ (FUN\ a\ b) \wedge IN\ v\ b)\ (u, v))))$

thm Misc_defs_and_lemmas.CARD_DELETE_CHOICE:

$\forall a::?'a::type \Rightarrow bool.\ FINITE\ a \wedge a \neq EMPTY \longrightarrow Suc\ (CARD\ (DELETE\ a\ (CHOICE\ a))) = CARD\ a$

thm Misc_defs_and_lemmas.FUN_SIZE:

$\forall (b::?'b::type \Rightarrow bool)\ a::?'a::type \Rightarrow bool.\ FINITE\ a \wedge FINITE\ b \longrightarrow HAS\_SIZE\ (FUN\ a\ b)\ (CARD\ b)^{CARD\ a}$

thm DEF_drop0:

$drop0 = snd$

thm Misc_defs_and_lemmas.drop0:

$\forall$ *u*::*?'b*::*type* × *?'a*::*type*. *drop0 u* = *snd u*

thm DEF_drop1:

drop1 = (λ*_2128181*::*?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *snd* (*snd _2128181*))

thm Misc_defs_and_lemmas.drop1:

$\forall$ *u*::*?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *drop1 u* = *snd* (*snd u*)

thm DEF_drop2:

drop2 = (λ*_2128186*::*?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *snd* (*snd* (*snd _2128186*)))

thm Misc_defs_and_lemmas.drop2:

$\forall$ *u*::*?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *drop2 u* = *snd* (*snd* (*snd u*))

thm DEF_drop3:

drop3 = (λ*_2128191*::*?'e*::*type* × *?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *snd* (*snd* (*snd* (*snd _2128191*))))

thm Misc_defs_and_lemmas.drop3:

$\forall$ *u*::*?'e*::*type* × *?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *drop3 u* = *snd* (*snd* (*snd* (*snd u*)))

thm DEF_part0:

part0 = *fst*

thm Misc_defs_and_lemmas.part0:

$\forall$ *u*::*?'b*::*type* × *?'a*::*type*. *part0 u* = *fst u*

thm DEF_part1:

part1 = (λ*_2128201*::*?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *fst* (*drop0 _2128201*))

thm Misc_defs_and_lemmas.part1:

$\forall$ *u*::*?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *part1 u* = *fst* (*drop0 u*)

thm DEF_part2:

part2 = (λ*_2128206*::*?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *fst* (*drop1 _2128206*))

thm Misc_defs_and_lemmas.part2:

$\forall$ *u*::*?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *part2 u* = *fst* (*drop1 u*)

thm DEF_part3:

part3 = (λ*_2128211*::*?'e*::*type* × *?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. *fst* (*drop2 _2128211*))

thm Misc_defs_and_lemmas.part3:

$\forall u::?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ part3\ u = fst\ (drop2\ u)$

thm DEF_part4:

$part4 = (\lambda\_2128216::?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ fst\ (drop3\ \_2128216))$

thm Misc_defs_and_lemmas.part4:

$\forall u::?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ part4\ u = fst\ (drop3\ u)$

thm DEF_part5:

$part5 = (\lambda\_2128221::?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ fst\ (snd\ (snd\ (snd\ (snd\ (snd\ \_2128221))))))$

thm Misc_defs_and_lemmas.part5:

$\forall u::?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ part5\ u = fst\ (snd\ (snd\ (snd\ (snd\ (snd\ u)))))$

thm DEF_part6:

$part6 = (\lambda\_2128226::?'h::type \times ?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ \_2128226)))))))$

thm Misc_defs_and_lemmas.part6:

$\forall u::?'h::type \times ?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ part6\ u = fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ u))))))$

thm DEF_part7:

$part7 = (\lambda\_2128231::?'i::type \times ?'h::type \times ?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ \_2128231))))))))$

thm Misc_defs_and_lemmas.part7:

$\forall u::?'i::type \times ?'h::type \times ?'g::type \times ?'f::type \times ?'e::type \times ?'d::type \times ?'c::type \times ?'b::type \times ?'a::type.\ part7\ u = fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ u)))))))$

thm Tactics.WELLDEFINED_FUNCTION_1:

$(\exists f::?'c::type \Rightarrow ?'b::type.\ \forall x::?'a::type.\ f\ ((?s::?'a::type \Rightarrow ?'c::type)\ x) = (?t::?'a::type \Rightarrow ?'b::type)\ x) = (\forall (x::?'a::type)\ x'::?'a::type.\ ?s\ x = ?s\ x' \longrightarrow ?t\ x = ?t\ x')$

thm Tactics.WELLDEFINED_FUNCTION_2b:

$(\exists f::?'d::type \Rightarrow ?'c::type.\ \forall (x::?'b::type)\ y::?'a::type.\ (?P::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ y \longrightarrow f\ ((?s::?'b::type \Rightarrow ?'a::type \Rightarrow ?'d::type)\ x\ y) = (?t::?'b::type$

$\Rightarrow$?'a::type $\Rightarrow$?'c::type) x y) = ($\forall$ (x::?'b::type) (x'::?'b::type) (y::?'a::type) y'::?'a::type. ?P x y $\wedge$?P x' y' $\wedge$?s x y = ?s x' y' $\longrightarrow$?t x y = ?t x' y')

thm Hales_tactic.GSYM_EXISTS_PAIRED_THM:

$\forall$ P::?'b::type $\Rightarrow$?'a::type $\Rightarrow$ bool. ($\exists$ (x::?'b::type) y::?'a::type. P x y) = Ex (GABS (λf::?'b::type $\times$?'a::type $\Rightarrow$ bool. $\forall$ (x::?'b::type) y::?'a::type. GEQ (f (x, y)) (P x y)))

thm Hales_tactic.hold_def:

hold = id

thm Trigonometry1.REAL_DIV_MUL2:

$\forall$ (x::real) z::real. x $\neq$ (0::real) $\wedge$ z $\neq$ (0::real) $\longrightarrow$ ($\forall$ y::real. y / z = x * y / (x * z))

thm Trigonometry1.REAL_LT_MUL_3:

$\forall$ (x::real) (y::real) z::real. (0::real) < x $\wedge$ (0::real) < y $\wedge$ (0::real) < z $\longrightarrow$ (0::real) < x * (y * z)

thm Trigonometry1.SQRT_MUL_L:

$\forall$ (x::real) y::real. (0::real) $\leq$ x $\wedge$ (0::real) $\leq$ y $\longrightarrow$ x * sqrt y = sqrt (x^2 * y)

thm Trigonometry1.SQRT_MUL_R:

$\forall$ (x::real) y::real. (0::real) $\leq$ x $\wedge$ (0::real) $\leq$ y $\longrightarrow$ sqrt x * y = sqrt (x * y^2)

thm Trigonometry2.SIN_TOTAL_PERIODIC:

$\forall$ n::nat. sin ((?x::real) + real_of_nat n * (real_of_nat (2::nat) * pi)) = sin ?x

thm Trigonometry1.SIN_PERIODIC_N2PI:

$\forall$ (x::real) n::nat. sin (x + real_of_nat n * (real_of_nat (2::nat) * pi)) = sin x

thm Trigonometry1.COS_PERIODIC_N2PI:

$\forall$ (x::real) n::nat. cos (x + real_of_nat n * (real_of_nat (2::nat) * pi)) = cos x

thm Trigonometry1.CIRCLE_SINCOS_PI:

$\forall$ (x::real) y::real. x^2 + y^2 = (1::real) $\longrightarrow$ ($\exists$ t::real. (− pi < t $\wedge$ t $\leq$ pi) $\wedge$ x = cos t $\wedge$ y = sin t)

thm Trigonometry1.SIN_NEGPOS_PI:

$\forall$ x::real. − pi < x $\wedge$ x $\leq$ pi $\longrightarrow$ (sin x < (0::real)) = (− pi < x $\wedge$ x < (0::real)) $\wedge$ (sin x = (0::real)) = (x = (0::real) $\vee$ x = pi) $\wedge$ ((0::real) < sin x) = ((0::real) < x $\wedge$ x < pi)

thm Trigonometry.SCEZKRH1:

$\forall x::real.\ sin\ (pi\ /\ real\_of\_nat\ (2::nat) - x) = cos\ x$

thm Trigonometry1.COS_NEGPOS_PI:

$\forall x::real.\ - pi < x \wedge x \leq pi \longrightarrow (cos\ x < (0::real)) = (- pi < x \wedge x < - (pi\ /\ real\_of\_nat\ (2::nat)) \vee pi\ /\ real\_of\_nat\ (2::nat) < x \wedge x \leq pi) \wedge (cos\ x = (0::real)) = (x = - (pi\ /\ real\_of\_nat\ (2::nat)) \vee x = pi\ /\ real\_of\_nat\ (2::nat)) \wedge ((0::real) < cos\ x) = (- (pi\ /\ real\_of\_nat\ (2::nat)) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat))$

thm Trigonometry1.dist_lemma:

$\forall (x::real)\ y::real.\ x \neq (0::real) \vee y \neq (0::real) \longrightarrow (x\ /\ sqrt\ (x^2 + y^2))^2 + (y\ /\ sqrt\ (x^2 + y^2))^2 = (1::real) \wedge (0::real) < sqrt\ (x^2 + y^2)$

thm Trigonometry1.ATAN2_EXISTS:

$\forall (x::real)\ y::real.\ \exists t::real.\ (- pi < t \wedge t \leq pi) \wedge x = sqrt\ (x^2 + y^2) * cos\ t \wedge y = sqrt\ (x^2 + y^2) * sin\ t$

thm DEF_atan2_temp:

$atan2\_temp = (\lambda\_2128507::real \times real.\ if\ fst\ \_2128507 = (0::real) \wedge snd\ \_2128507 = (0::real)\ then\ pi\ else\ SOME\ t::real.\ (- pi < t \wedge t \leq pi) \wedge fst\ \_2128507 = sqrt\ ((fst\ \_2128507)^2 + (snd\ \_2128507)^2) * cos\ t \wedge snd\ \_2128507 = sqrt\ ((fst\ \_2128507)^2 + (snd\ \_2128507)^2) * sin\ t)$

thm Trigonometry1.ATAN2_TEMP_DEF:

$\forall (x::real)\ y::real.\ atan2\_temp\ (x,\ y) = (if\ x = (0::real) \wedge y = (0::real)\ then\ pi\ else\ SOME\ t::real.\ (- pi < t \wedge t \leq pi) \wedge x = sqrt\ (x^2 + y^2) * cos\ t \wedge y = sqrt\ (x^2 + y^2) * sin\ t)$

thm Trigonometry1.ATAN2_TEMP:

$\forall (x::real)\ y::real.\ (- pi < atan2\_temp\ (x,\ y) \wedge atan2\_temp\ (x,\ y) \leq pi) \wedge x = sqrt\ (x^2 + y^2) * cos\ (atan2\_temp\ (x,\ y)) \wedge y = sqrt\ (x^2 + y^2) * sin\ (atan2\_temp\ (x,\ y))$

thm Trigonometry1.ATAN2_TEMP_SPEC:

$\forall (x::real)\ y::real.\ \exists r::real.\ - pi < atan2\_temp\ (x,\ y) \wedge atan2\_temp\ (x,\ y) \leq pi \wedge x = r * cos\ (atan2\_temp\ (x,\ y)) \wedge y = r * sin\ (atan2\_temp\ (x,\ y)) \wedge (0::real) \leq r$

thm Trigonometry1.ATAN2_TEMP_BREAKDOWN:

$\forall (x::real)\ y::real.\ ((0::real) < x \longrightarrow atan2\_temp\ (x,\ y) = atn\ (y\ /\ x)) \wedge ((0::real) < y \longrightarrow atan2\_temp\ (x,\ y) = pi\ /\ real\_of\_nat\ (2::nat) - atn\ (x\ /\ y)) \wedge (y < (0::real) \longrightarrow atan2\_temp\ (x,\ y) = - (pi\ /\ real\_of\_nat\ (2::nat)) - atn\ (x\ /\ y)) \wedge (y = (0::real) \wedge x \leq (0::real) \longrightarrow atan2\_temp\ (x,\ y) = pi)$

thm Trigonometry1.ATAN2_TEMP_ALT:

$\forall (x::real)\ y::real.\ atan2\_temp\ (x,\ y) = (if\ |y| < x\ then\ atn\ (y\ /\ x)\ else\ if\ (0::real) < y\ then\ pi\ /\ real\_of\_nat\ (2::nat) - atn\ (x\ /\ y)\ else\ if\ y < (0::real)\ then\ - (pi\ /\ real\_of\_nat\ (2::nat)) - atn\ (x\ /\ y)\ else\ pi)$

thm Trigonometry1.ATAN_TEMP_ATN2:

$atn2 = atan2\_temp$

thm Trigonometry1.atn2_spec:

$\forall (x::real)\ y::real.\ \exists r::real.\ - pi < atn2\ (x,\ y) \wedge atn2\ (x,\ y) \leq pi \wedge x = r * cos\ (atn2\ (x,\ y)) \wedge y = r * sin\ (atn2\ (x,\ y)) \wedge (0::real) \leq r$

thm Trigonometry1.ATN2_BREAKDOWN:

$\forall (x::real)\ y::real.\ ((0::real) < x \longrightarrow atn2\ (x,\ y) = atn\ (y\ /\ x)) \wedge ((0::real) < y \longrightarrow atn2\ (x,\ y) = pi\ /\ real\_of\_nat\ (2::nat) - atn\ (x\ /\ y)) \wedge (y < (0::real) \longrightarrow atn2\ (x,\ y) = - (pi\ /\ real\_of\_nat\ (2::nat)) - atn\ (x\ /\ y)) \wedge (y = (0::real) \wedge x \leq (0::real) \longrightarrow atn2\ (x,\ y) = pi)$

thm Trigonometry1.ATN2_0_1:

$atn2\ (0::real,\ 1::real) = pi\ /\ real\_of\_nat\ (2::nat)$

thm Trigonometry1.ATN2_0_NEG_1:

$atn2\ (0::real,\ - (1::real)) = - (pi\ /\ real\_of\_nat\ (2::nat))$

thm Trigonometry1.ATN2_LMUL_EQ:

$\forall (a::real)\ (x::real)\ y::real.\ (0::real) < a \longrightarrow atn2\ (a * x,\ a * y) = atn2\ (x,\ y)$

thm Trigonometry1.ATN2_RNEG:

$\forall (x::real)\ y::real.\ y \neq (0::real) \vee (0::real) < x \longrightarrow atn2\ (x,\ - y) = - atn2\ (x,\ y)$

thm Trigonometry1.acs_atn2:

$\forall y::real.\ - (1::real) \leq y \wedge y \leq (1::real) \longrightarrow acs\ y = pi\ /\ real\_of\_nat\ (2::nat) - atn2\ (sqrt\ ((1::real) - y^2),\ y)$

thm Trigonometry.IHIQXLM:

$\forall (a::real)\ (b::real)\ c::real.\ ups\_x\ (a * a)\ (b * b)\ (c * c) = real\_of\_nat\ (16::nat) * ((a + (b + c))\ /\ real\_of\_nat\ (2::nat) * (((a + (b + c))\ /\ real\_of\_nat\ (2::nat) - a) * (((a + (b + c))\ /\ real\_of\_nat\ (2::nat) - b) * ((a + (b + c))\ /\ real\_of\_nat\ (2::nat) - c))))$

thm Trigonometry1.TRI_UPS_X_POS:

$\forall (a::real)\ (b::real)\ c::real.\ (0::real) < a \wedge (0::real) < b \wedge (0::real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow (0::real) \leq ups\_x\ (a * a)\ (b * b)\ (c * c)$

thm Trigonometry1.TRI_SQUARES_BOUNDS:

$\forall (a::real)\ (b::real)\ c::real.\ (0::real) < a \wedge (0::real) < b \wedge (0::real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow - (1::real) \leq (a * a + (b * b - c * c))$

$/ \ (real\_of\_nat\ (2{::}nat) * (a * b)) \wedge (a * a + (b * b - c * c)) \ / \ (real\_of\_nat\ (2{::}nat) * (a * b)) \leq (1{::}real)$

thm Trigonometry1.ATN2_ARCLENGTH:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow arclength\ a\ b\ c = pi \ / \ real\_of\_nat\ (2{::}nat) - atn2\ (sqrt\ (ups\_x\ (a * a)\ (b * b)\ (c * c)),\ a * a + (b * b - c * c))$

thm Trigonometry1.TRI_LEMMA:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow real\_of\_nat\ (2{::}nat) * (a * b) * ((a * a + (b * b - c * c)) \ / \ (real\_of\_nat\ (2{::}nat) * (a * b))) = a * a + (b * b - c * c)$

thm Trigonometry1.TRI_UPS_X_SQUARES:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow ups\_x\ (a * a)\ (b * b)\ (c * c) = (real\_of\_nat\ (2{::}nat) * (a * b))^2 * ((1{::}real) - ((a * a + (b * b - c * c)) \ / \ (real\_of\_nat\ (2{::}nat) * (a * b)))^2)$

thm Trigonometry1.TRI_UPS_X_SQRT:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow sqrt\ (ups\_x\ (a * a)\ (b * b)\ (c * c)) = real\_of\_nat\ (2{::}nat) * (a * b) * sqrt\ ((1{::}real) - ((a * a + (b * b - c * c)) \ / \ (real\_of\_nat\ (2{::}nat) * (a * b)))^2)$

thm Trigonometry1.ACS_ARCLENGTH:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow arclength\ a\ b\ c = acs\ ((a * a + (b * b - c * c)) \ / \ (real\_of\_nat\ (2{::}nat) * (a * b)))$

thm Trigonometry1.law_of_cosines:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow c^2 = a^2 + (b^2 - real\_of\_nat\ (2{::}nat) * (a * (b * cos\ (arclength\ a\ b\ c))))$

thm Trigonometry1.law_of_sines:

$\forall (a{::}real)\ (b{::}real)\ c{::}real.\ (0{::}real) < a \wedge (0{::}real) < b \wedge (0{::}real) \leq c \wedge c \leq a + b \wedge a \leq b + c \wedge b \leq c + a \longrightarrow real\_of\_nat\ (2{::}nat) * (a * (b * sin\ (arclength\ a\ b\ c))) = sqrt\ (ups\_x\ (a^2)\ (b^2)\ (c^2))$

thm Trigonometry1.DIST_TRIANGLE_DETAILS:

$((?u{::}(real,\ ?'a{::}type)\ cart) \neq (?v{::}(real,\ ?'a{::}type)\ cart) \wedge ?u \neq (?w{::}(real,\ ?'a{::}type)\ cart)) = (((0{::}real) < distance\ (?u,\ ?v) \wedge (0{::}real) < distance\ (?u,\ ?w) \wedge (0{::}real) \leq distance\ (?v,\ ?w) \wedge distance\ (?v,\ ?w) \leq distance\ (?u,\ ?v)$

+ *distance* (*?u*, *?w*) ∧ *distance* (*?u*, *?v*) ≤ *distance* (*?u*, *?w*) + *distance* (*?v*, *?w*) ∧ *distance* (*?u*, *?w*) ≤ *distance* (*?v*, *?w*) + *distance* (*?u*, *?v*))

thm Trigonometry1.arcVarc:

∀ (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *u* ≠ *v* ∧ *u* ≠ *w* ⟶ *arcV u v w* = *arclength* (*distance* (*u*, *v*)) (*distance* (*u*, *w*)) (*distance* (*v*, *w*))

thm Trigonometry.HQTBPCM2:

(*distance* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*))^2 = (*distance* (*?u*::(*real*, *3*) *cart*, *?v*))^2 + ((*distance* (*?u*, *?w*))^2 − *real_of_nat* (*2*::*nat*) ∗ (*distance* (*?u*, *?v*) ∗ (*distance* (*?u*, *?w*) ∗ *cos* (*arcV ?u ?v ?w*))))

thm Trigonometry1.DIST_L_ZERO:

∀ *v*::(*real*, *?'a*::*type*) *cart*. *distance* (*vec* (*0*::*nat*), *v*) = *vector_norm v*

thm Trigonometry1.DOT_COS:

dot (*?u*::(*real*, *3*) *cart*) (*?v*::(*real*, *3*) *cart*) = *vector_norm ?u* ∗ (*vector_norm ?v* ∗ *cos* (*arcV* (*vec* (*0*::*nat*)) *?u ?v*))

thm Trigonometry1.CART_EQ_3:

∀ (*x*::(*?'a*::*type*, *3*) *cart*) *y*::(*?'a*::*type*, *3*) *cart*. (*x* = *y*) = ($ *x* (*1*::*nat*) = $ *y* (*1*::*nat*) ∧ $ *x* (*2*::*nat*) = $ *y* (*2*::*nat*) ∧ $ *x* (*3*::*nat*) = $ *y* (*3*::*nat*))

thm Trigonometry1.LAMBDA_BETA_3:

$ (*lambda* (*?g*::*nat* ⇒ *?'a*::*type*)) (*1*::*nat*) = *?g* (*1*::*nat*) ∧ $ (*lambda ?g*) (*2*::*nat*) = *?g* (*2*::*nat*) ∧ $ (*lambda ?g*) (*3*::*nat*) = *?g* (*3*::*nat*)

thm Trigonometry1.LAMBDA_BETA_3_conjunct2:

$ (*lambda* (*?g*::*nat* ⇒ *?'a*::*type*)) (*3*::*nat*) = *?g* (*3*::*nat*)

thm Trigonometry1.LAMBDA_BETA_3_conjunct1:

$ (*lambda* (*?g*::*nat* ⇒ *?'a*::*type*)) (*2*::*nat*) = *?g* (*2*::*nat*)

thm Trigonometry1.LAMBDA_BETA_3_conjunct0:

$ (*lambda* (*?g*::*nat* ⇒ *?'a*::*type*)) (*1*::*nat*) = *?g* (*1*::*nat*)

thm Trigonometry1.VEC_COMPONENT_3:

∀ *k*::*nat*. $ (*vec k*) (*1*::*nat*) = *real_of_nat k* ∧ $ (*vec k*) (*2*::*nat*) = *real_of_nat k* ∧ $ (*vec k*) (*3*::*nat*) = *real_of_nat k*

thm Trigonometry1.VECTOR_ADD_COMPONENT_3:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. $ (*vector_add x y*) (*1*::*nat*) = $ *x* (*1*::*nat*) + $ *y* (*1*::*nat*) ∧ $ (*vector_add x y*) (*2*::*nat*) = $ *x* (*2*::*nat*) + $ *y* (*2*::*nat*) ∧ $ (*vector_add x y*) (*3*::*nat*) = $ *x* (*3*::*nat*) + $ *y* (*3*::*nat*)

thm Trigonometry1.VECTOR_NEG_COMPONENT_3:

$\forall x::(real, 3)\ cart.\ \$\ (vector\_neg\ x)\ (1::nat) = -\ \$\ x\ (1::nat) \wedge \$\ (vector\_neg\ x)\ (2::nat) = -\ \$\ x\ (2::nat) \wedge \$\ (vector\_neg\ x)\ (3::nat) = -\ \$\ x\ (3::nat)$

thm Trigonometry1.VECTOR_MUL_COMPONENT_3:

$\forall (c::real)\ x::(real, 3)\ cart.\ \$\ (\%\ c\ x)\ (1::nat) = c * \$\ x\ (1::nat) \wedge \$\ (\%\ c\ x)\ (2::nat) = c * \$\ x\ (2::nat) \wedge \$\ (\%\ c\ x)\ (3::nat) = c * \$\ x\ (3::nat)$

thm Trigonometry1.BASIS_3:

$\$\ (basis\ (1::nat))\ (1::nat) = (1::real) \wedge \$\ (basis\ (1::nat))\ (2::nat) = (0::real) \wedge \$\ (basis\ (1::nat))\ (3::nat) = (0::real) \wedge \$\ (basis\ (2::nat))\ (1::nat) = (0::real) \wedge \$\ (basis\ (2::nat))\ (2::nat) = (1::real) \wedge \$\ (basis\ (2::nat))\ (3::nat) = (0::real) \wedge \$\ (basis\ (3::nat))\ (1::nat) = (0::real) \wedge \$\ (basis\ (3::nat))\ (2::nat) = (0::real) \wedge \$\ (basis\ (3::nat))\ (3::nat) = (1::real)$

thm Trigonometry1.COMPONENTS_3:

$\forall v::(real, 3)\ cart.\ v = vector\_add\ (\%\ (\$\ v\ (1::nat))\ (basis\ (1::nat)))\ (vector\_add\ (\%\ (\$\ v\ (2::nat))\ (basis\ (2::nat)))\ (\%\ (\$\ v\ (3::nat))\ (basis\ (3::nat))))$

thm Trigonometry1.VECTOR_COMPONENTS_3:

$\forall (a::real)\ (b::real)\ c::real.\ vector\ [a, b, c] = vector\_add\ (\%\ a\ (basis\ (1::nat)))\ (vector\_add\ (\%\ b\ (basis\ (2::nat)))\ (\%\ c\ (basis\ (3::nat))))$

thm Trigonometry1.cross_skew:

$\forall (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ cross\ u\ v = vector\_neg\ (cross\ v\ u)$

thm Trigonometry1.cross_triple:

$\forall (u::(real, 3)\ cart)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ dot\ (cross\ u\ v)\ w = dot\ (cross\ v\ w)\ u$

thm Trigonometry1.NORM_CAUCHY_SCHWARZ_FRAC:

$\forall (u::(real, ?'a::type)\ cart)\ v::(real, ?'a::type)\ cart.\ u \neq vec\ (0::nat) \wedge v \neq vec\ (0::nat) \longrightarrow -\ (1::real) \leq dot\ u\ v\ /\ (vector\_norm\ u * vector\_norm\ v) \wedge dot\ u\ v\ /\ (vector\_norm\ u * vector\_norm\ v) \leq (1::real)$

thm Trigonometry1.CROSS_SQUARED:

$\forall (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ dot\ (cross\ u\ v)\ (cross\ u\ v) = ups\_x\ (dot\ u\ u)\ (dot\ v\ v)\ (dot\ (vector\_sub\ u\ v)\ (vector\_sub\ u\ v))\ /\ real\_of\_nat\ (4::nat)$

thm Trigonometry1.DIST_UPS_X_POS:

$(?u::(real, ?'a::type)\ cart) \neq (?v::(real, ?'a::type)\ cart) \wedge ?u \neq (?w::(real, ?'a::type)\ cart) \longrightarrow (0::real) \leq ups\_x\ ((distance\ (?u, ?v))^2)\ ((distance\ (?u, ?w))^2)\ ((distance\ (?v, ?w))^2)$

thm Trigonometry1.SQRT_DIV_R:

$(0::real) \leq (?x::real) \wedge (0::real) \leq (?y::real) \longrightarrow sqrt\ ?x\ /\ ?y = sqrt\ (?x\ /\ ?y^2) \wedge (0::real) \leq ?x\ /\ ?y^2$

thm Trigonometry1.NORM_CROSS:

$\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ vec\ (0::nat) \neq u \wedge vec\ (0::nat) \neq v \longrightarrow vector\_norm\ (cross\ u\ v) = sqrt\ (ups\_x\ ((vector\_norm\ u)^2)\ ((vector\_norm\ v)^2)\ ((distance\ (u,\ v))^2))\ /\ real\_of\_nat\ (2::nat)$

thm Trigonometry.UKBAHKV:

$vec\ (0::nat) \neq (?u::(real,\ 3)\ cart) \wedge vec\ (0::nat) \neq (?v::(real,\ 3)\ cart) \longrightarrow real\_of\_nat\ (2::nat) * (vector\_norm\ ?u * (vector\_norm\ ?v * sin\ (arcV\ (vec\ (0::nat))\ ?u\ ?v))) = sqrt\ (ups\_x\ ((vector\_norm\ ?u)^2)\ ((vector\_norm\ ?v)^2)\ ((distance\ (?u,\ ?v))^2))$

thm Trigonometry1.cross_mag:

$\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ vector\_norm\ (cross\ u\ v) = vector\_norm\ u * (vector\_norm\ v * sin\ (arcV\ (vec\ (0::nat))\ u\ v))$

thm DEF_cosV:

$cosV = (\lambda(\_2128608::(real,\ ?'a::type)\ cart)\ \_2128609::(real,\ ?'a::type)\ cart.\ dot\ \_2128608\ \_2128609\ /\ (vector\_norm\ \_2128608 * vector\_norm\ \_2128609))$

thm Trigonometry2.cosV:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ cosV\ u\ v = dot\ u\ v\ /\ (vector\_norm\ u * vector\_norm\ v)$

thm DEF_sinV:

$sinV = (\lambda(\_2128620::(real,\ ?'a::type)\ cart)\ \_2128621::(real,\ ?'a::type)\ cart.\ sqrt\ ((1::real) - (cosV\ \_2128620\ \_2128621)^2))$

thm Trigonometry2.sinV:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ sinV\ u\ v = sqrt\ ((1::real) - (cosV\ u\ v)^2)$

thm Trigonometry2.NOT_EQ_IMPCOS_ARC:

$(?v0.0::(real,\ ?'a::type)\ cart) \neq (?u::(real,\ ?'a::type)\ cart) \wedge ?v0.0 \neq (?w::(real,\ ?'a::type)\ cart) \longrightarrow cos\ (arcV\ ?v0.0\ ?u\ ?w) = dot\ (vector\_sub\ ?u\ ?v0.0)\ (vector\_sub\ ?w\ ?v0.0)\ /\ (vector\_norm\ (vector\_sub\ ?u\ ?v0.0) * vector\_norm\ (vector\_sub\ ?w\ ?v0.0))$

thm Trigonometry2.COLLINEAR_TRANSABLE:

$collinear\ (INSERT\ (?a::(real,\ ?'a::type)\ cart)\ (INSERT\ (?b::(real,\ ?'a::type)\ cart)\ (INSERT\ (?c::(real,\ ?'a::type)\ cart)\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ ?b\ ?a)\ (INSERT\ (vector\_sub\ ?c\ ?a)\ EMPTY)))$

thm Trigonometry2.ALLEMI_COLLINEAR:

$\%\ (dot\ (vector\_sub\ (?vc::(real,\ ?'a::type)\ cart)\ (?v0.0::(real,\ ?'a::type)\ cart))\ (vector\_sub\ ?vc\ ?v0.0))\ (vector\_sub\ (?va::(real,\ ?'a::type)\ cart)\ ?v0.0) = \%$

$(dot\ (vector\_sub\ ?va\ ?v0.0)\ (vector\_sub\ ?vc\ ?v0.0))\ (vector\_sub\ ?vc\ ?v0.0) \longrightarrow collinear\ (INSERT\ ?v0.0\ (INSERT\ ?vc\ (INSERT\ ?va\ EMPTY)))$

thm Trigonometry2.NOT_VEC0_IMP_LE1:

$(?x::(real,\ ?'a::type)\ cart) \neq vec\ (0::nat) \wedge (?y::(real,\ ?'a::type)\ cart) \neq vec\ (0::nat) \longrightarrow |dot\ ?x\ ?y\ /\ (vector\_norm\ ?x * vector\_norm\ ?y)| \leq (1::real)$

thm Trigonometry2.sin_acs_t:

$\forall y::real.\ - (1::real) \leq y \wedge y \leq (1::real) \longrightarrow sin\ (acs\ y) = sqrt\ ((1::real) - y^2)$

thm Trigonometry2.ABS_LE_1_IMP_SIN_ACS:

$\forall y::real.\ |y| \leq (1::real) \longrightarrow sin\ (acs\ y) = sqrt\ ((1::real) - y^2)$

thm Trigonometry2.NOT_2EQ_IMP_SIN_ARCV:

$(?v0.0::(real,\ ?'a::type)\ cart) \neq (?va::(real,\ ?'a::type)\ cart) \wedge ?v0.0 \neq (?vb::(real,\ ?'a::type)\ cart) \longrightarrow sin\ (arcV\ ?v0.0\ ?va\ ?vb) = sqrt\ ((1::real) - (dot\ (vector\_sub\ ?va\ ?v0.0)\ (vector\_sub\ ?vb\ ?v0.0)\ /\ (vector\_norm\ (vector\_sub\ ?va\ ?v0.0) * vector\_norm\ (vector\_sub\ ?vb\ ?v0.0)))^2)$

thm Trigonometry2.ABS_NOT_EQ_NORM_MUL:

$(|dot\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)| \neq vector\_norm\ ?x * vector\_norm\ ?y) = (|dot\ ?x\ ?y| < vector\_norm\ ?x * vector\_norm\ ?y)$

thm Trigonometry2.SQUARE_NORM_CAUCHY_SCHWARZ_POW2:

$(dot\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart))^2 \leq (vector\_norm\ ?x * vector\_norm\ ?y)^2$

thm Trigonometry2.SQRT_SEPARATED:

$sqrt\ (((vector\_norm\ (?x::(real,\ ?'a::type)\ cart) * vector\_norm\ (?y::(real,\ ?'a::type)\ cart))^2 - (dot\ ?x\ ?y)^2)\ /\ (vector\_norm\ ?x * vector\_norm\ ?y)^2) = sqrt\ ((vector\_norm\ ?x * vector\_norm\ ?y)^2 - (dot\ ?x\ ?y)^2)\ /\ sqrt\ ((vector\_norm\ ?x * vector\_norm\ ?y)^2)$

thm Trigonometry2.COMPUTE_NORM_OF_P:

$vector\_norm\ (vector\_sub\ (\%\ (dot\ (?vc::(real,\ ?'a::type)\ cart)\ ?vc)\ (?va::(real,\ ?'a::type)\ cart))\ (\%\ (dot\ ?va\ ?vc)\ ?vc)) = sqrt\ (dot\ ?vc\ ?vc * (dot\ ?va\ ?va * dot\ ?vc\ ?vc - (dot\ ?va\ ?vc)^2))$

thm Trigonometry2.MOVE_NORM_OUT_OF_SQRT:

$sqrt\ ((vector\_norm\ (?vc::(real,\ ?'a::type)\ cart))^2 * ((vector\_norm\ (?va::(real,\ ?'a::type)\ cart) * vector\_norm\ ?vc)^2 - (dot\ ?va\ ?vc)^2)) = vector\_norm\ ?vc * sqrt\ ((vector\_norm\ ?va * vector\_norm\ ?vc)^2 - (dot\ ?va\ ?vc)^2)$

thm Trigonometry2.PI2_EQ_ACS0:

$pi\ /\ real\_of\_nat\ (2::nat) = acs\ (0::real)$

thm Trigonometry2.ANGLE_EQ_ARCV:

∀ (vap::(real, ?'a::type) cart) vbp::(real, ?'a::type) cart. angle (vap, vec (0::nat), vbp) = arcV (vec (0::nat)) vap vbp

thm Trigonometry2.dihV:

∀ (w0::(real, ?'a::type) cart) (w1::(real, ?'a::type) cart) (w2::(real, ?'a::type) cart) w3::(real, ?'a::type) cart. dihV w0 w1 w2 w3 = LET (λva::(real, ?'a::type) cart. LET_END (LET (λvb::(real, ?'a::type) cart. LET_END (LET (λvc::(real, ?'a::type) cart. LET_END (LET (λvap::(real, ?'a::type) cart. LET_END (LET (λvbp::(real, ?'a::type) cart. LET_END (arcV (vec (0::nat)) vap vbp)) (vector_sub (% (dot vc vc) vb) (% (dot vb vc) vc)))) (vector_sub (% (dot vc vc) va) (% (dot va vc) vc)))) (vector_sub w1 w0))) (vector_sub w3 w0))) (vector_sub w2 w0)

thm Trigonometry.RLXWSTK:

∀ (v0::(real, ?'a::type) cart) (va::(real, ?'a::type) cart) (vb::(real, ?'a::type) cart) vc::(real, ?'a::type) cart. LET (λgam::real. LET_END (LET (λa::real. LET_END (LET (λb::real. LET_END (LET (λc::real. LET_END (¬ collinear (INSERT v0 (INSERT vc (INSERT va EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT vc (INSERT vb EMPTY))) ⟶ cos gam = (cos c − cos a ∗ cos b) / (sin a ∗ sin b))) (arcV v0 va vb))) (arcV v0 vc va))) (arcV v0 vc vb))) (dihV v0 vc va vb)

thm Trigonometry2.LE_AND_NOT_0_EQ_LT:

((0::real) ≤ (?a::real) ∧ ?a ≠ (0::real)) = ((0::real) < ?a)

thm Trigonometry2.NOT_COLLINEAR_IMP_NOT_SIN0:

¬ collinear (INSERT (?v0.0::(real, ?'a::type) cart) (INSERT (?va::(real, ?'a::type) cart) (INSERT (?vb::(real, ?'a::type) cart) EMPTY))) ⟶ sin (arcV ?v0.0 ?va ?vb) ≠ (0::real)

thm Trigonometry2.REAL_LE_LDIV:

∀ (x::real) z::real. (0::real) < z ∧ x ≤ z ⟶ x / z ≤ (1::real)

thm Trigonometry2.NOT_IDEN_IMP_ABS_LE:

(?v0.0::(real, ?'a::type) cart) ≠ (?va::(real, ?'a::type) cart) ∧ ?v0.0 ≠ (?vb::(real, ?'a::type) cart) ⟶ |dot (vector_sub ?va ?v0.0) (vector_sub ?vb ?v0.0) / (vector_norm (vector_sub ?va ?v0.0) ∗ vector_norm (vector_sub ?vb ?v0.0))| ≤ (1::real)

thm Trigonometry2.PROVE_SIN_LE:

(?v0.0::(real, ?'a::type) cart) ≠ (?va::(real, ?'a::type) cart) ∧ ?v0.0 ≠ (?vb::(real, ?'a::type) cart) ⟶ (0::real) ≤ sin (arcV ?v0.0 ?va ?vb)

thm Trigonometry2.COMPUTE_SIN_DIVH_POW2:

∀ (v0::(real, ?'a::type) cart) (va::(real, ?'a::type) cart) (vb::(real, ?'a::type) cart) vc::(real, ?'a::type) cart. LET (λbetaa::real. LET_END (LET (λa::real.

LET_END (*LET* (*λb*::*real*. *LET_END* (*LET* (*λc*::*real*. *LET_END* (*LET* (*λp*::*real*. *LET_END* (¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT va EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT vb EMPTY*))) ⟶ (*sin betaa*)^2 = *p* / (*sin a* ∗ *sin b*)^2)) ((*1*::*real*) − (*cos a*)^2 − (*cos b*)^2 − (*cos c*)^2 + *real_of_nat* (*2*::*nat*) ∗ (*cos a* ∗ (*cos b* ∗ *cos c*))))) (*arcV v0 va vb*))) (*arcV v0 vc va*))) (*arcV v0 vc vb*))) (*dihV v0 vc va vb*)

thm Trigonometry2.PROVE_P_LE:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*va*::(*real*, *?'a*::*type*) *cart*) (*vb*::(*real*, *?'a*::*type*) *cart*) *vc*::(*real*, *?'a*::*type*) *cart*. *LET* (*λa*::*real*. *LET_END* (*LET* (*λb*::*real*. *LET_END* (*LET* (*λc*::*real*. *LET_END* (*LET* (*λp*::*real*. *LET_END* (¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT va EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT vb EMPTY*))) ⟶ (*0*::*real*) ≤ *p*)) ((*1*::*real*) − (*cos a*)^2 − (*cos b*)^2 − (*cos c*)^2 + *real_of_nat* (*2*::*nat*) ∗ (*cos a* ∗ (*cos b* ∗ *cos c*))))) (*arcV v0 va vb*))) (*arcV v0 vc va*))) (*arcV v0 vc vb*)

thm Trigonometry2.NOT_COLLINEAR_IMP_2_UNEQUAL:

¬ *collinear* (*INSERT* (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?va*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?vb*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *?v0.0* ≠ *?va* ∧ *?v0.0* ≠ *?vb*

thm Trigonometry2.NOT_COLL_IM_SIN_LT:

¬ *collinear* (*INSERT* (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?va*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?vb*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ (*0*::*real*) < *sin* (*arcV ?v0.0 ?va ?vb*)

thm Trigonometry2.ARC_SYM:

arcV (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*?vc*::(*real*, *?'a*::*type*) *cart*) (*?vb*::(*real*, *?'a*::*type*) *cart*) = *arcV ?v0.0 ?vb ?vc*

thm Trigonometry2.SIN_MUL_EXPAND:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*va*::(*real*, *?'a*::*type*) *cart*) (*vb*::(*real*, *?'a*::*type*) *cart*) *vc*::(*real*, *?'a*::*type*) *cart*. *LET* (*λgam*::*real*. *LET_END* (*LET* (*λbet*::*real*. *LET_END* (*LET* (*λa*::*real*. *LET_END* (*LET* (*λb*::*real*. *LET_END* (*LET* (*λc*::*real*. *LET_END* (*LET* (*λp*::*real*. *LET_END* (¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT va EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT vb EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT va* (*INSERT vb EMPTY*))) ⟶ *sin gam* ∗ *sin bet* = *p* / (*sin b* ∗ (*sin c* ∗ (*sin a*)^2)))) ((*1*::*real*) − (*cos a*)^2 − (*cos b*)^2 − (*cos c*)^2 + *real_of_nat* (*2*::*nat*) ∗ (*cos a* ∗ (*cos b* ∗ *cos c*))))) (*arcV v0 va vb*))) (*arcV v0 vc va*))) (*arcV v0 vc vb*))) (*dihV v0 vb vc va*))) (*dihV v0 vc va vb*)

thm Trigonometry2.DIHV_SYM:

dihV (*?a*::(*real*, *?'a*::*type*) *cart*) (*?b*::(*real*, *?'a*::*type*) *cart*) (*?x*::(*real*, *?'a*::*type*) *cart*) (*?y*::(*real*, *?'a*::*type*) *cart*) = *dihV ?a ?b ?y ?x*

thm Trigonometry.NLVWBBW:

$\forall$ (*v0*::(*real*, *?'a*::*type*) *cart*) (*va*::(*real*, *?'a*::*type*) *cart*) (*vb*::(*real*, *?'a*::*type*) *cart*) *vc*::(*real*, *?'a*::*type*) *cart*. *LET* (λ*al*::*real*. *LET_END* (*LET* (λ*ga*::*real*. *LET_END* (*LET* (λ*be*::*real*. *LET_END* (*LET* (λ*a*::*real*. *LET_END* (*LET* (λ*b*::*real*. *LET_END* (*LET* (λ*c*::*real*. *LET_END* ($\neg$ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT va EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT vb EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT va* (*INSERT vb EMPTY*))) $\longrightarrow$ *cos c* $*$ (*sin al* $*$ *sin be*) = *cos ga* + *cos al* $*$ *cos be*)) (*arcV v0 va vb*))) (*arcV v0 vc va*))) (*arcV v0 vc vb*))) (*dihV v0 vb vc va*))) (*dihV v0 vc va vb*))) (*dihV v0 va vb vc*)

thm Trigonometry2.COMPUTE_NORM_POW2:

(*vector_norm* (*vector_sub* (% (*dot* (*?vc*::(*real*, *?'a*::*type*) *cart*) *?vc*) (*?vb*::(*real*, *?'a*::*type*) *cart*)) (% (*dot ?vb ?vc*) *?vc*)))^2 = ((*vector_norm ?vc*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vc*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vc*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vc*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vb*)^2 + (*vector_norm ?vb*)^2 − (*distance* (*?vb*, *?vb*))^2) / *real_of_nat* (*2*::*nat*)) − ((*vector_norm ?vb*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vb*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vb*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vb*, *?vc*))^2) / *real_of_nat* (*2*::*nat*))) − ((*vector_norm ?vb*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vb*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vc*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vc*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vc*)^2 + (*vector_norm ?vb*)^2 − (*distance* (*?vc*, *?vb*))^2) / *real_of_nat* (*2*::*nat*)) − ((*vector_norm ?vb*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vb*, *?vc*))^2) / *real_of_nat* (*2*::*nat*) $*$ (((*vector_norm ?vc*)^2 + (*vector_norm ?vc*)^2 − (*distance* (*?vc*, *?vc*))^2) / *real_of_nat* (*2*::*nat*)))

thm Trigonometry2.UPS_X_AND_HERON:

ups_x ((*?x1.0*::*real*)^2) ((*?x2.0*::*real*)^2) ((*?x6.0*::*real*)^2) = (*?x1.0* + (*?x2.0* + *?x6.0*)) $*$ ((*?x1.0* + (*?x2.0* − *?x6.0*)) $*$ ((*?x2.0* + (*?x6.0* − *?x1.0*)) $*$ (*?x6.0* + (*?x1.0* − *?x2.0*))))

thm Trigonometry2.UPS_X_POS:

(*distance* (*?v0.0*::(*real*, *?'a*::*type*) *cart*, *?v1.0*::(*real*, *?'a*::*type*) *cart*))^2 = (*?v01.0*::*real*) $\wedge$ (*distance* (*?v0.0*, *?v2.0*::(*real*, *?'a*::*type*) *cart*))^2 = (*?v02.0*::*real*) $\wedge$ (*distance* (*?v1.0*, *?v2.0*))^2 = (*?v12.0*::*real*) $\longrightarrow$ (*0*::*real*) $\leq$ *ups_x ?v01.0 ?v02.0 ?v12.0*

thm Trigonometry2.DIST_TRANSABLE:

distance (*vector_sub* (*?a*::(*real*, *?'a*::*type*) *cart*) (*?v0.0*::(*real*, *?'a*::*type*) *cart*), *?b*::(*real*, *?'a*::*type*) *cart*) = *distance* (*?a*, *vector_add ?b ?v0.0*)

thm Trigonometry2.PROVE_DELTA_OVER_SQRT_2UPS:

(*distance* (*?v0.0*::(*real*, *?'a*::*type*) *cart*, *?v1.0*::(*real*, *?'a*::*type*) *cart*))^2 = (*?v01.0*::*real*) $\wedge$ (*distance* (*?v0.0*, *?v2.0*::(*real*, *?'a*::*type*) *cart*))^2 = (*?v02.0*::*real*) $\wedge$ (*distance* (*?v0.0*, *?v3.0*::(*real*, *?'a*::*type*) *cart*))^2 = (*?v03.0*::*real*) $\wedge$ (*distance* (*?v1.0*, *?v2.0*))^2 = (*?v12.0*::*real*) $\wedge$ (*distance* (*?v1.0*, *?v3.0*))^2 = (*?v13.0*::*real*) $\wedge$ (*distance* (*?v2.0*, *?v3.0*))^2 = (*?v23.0*::*real*) $\wedge$ $\neg$ *collinear* (*INSERT ?v0.0* (*INSERT*

?v1.0 (INSERT ?v2.0 EMPTY))) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?v3.0 EMPTY))) ⟶ LET (λva::(real, ?'a::type) cart. LET_END (LET (λvb::(real, ?'a::type) cart. LET_END (LET (λvc::(real, ?'a::type) cart. LET_END (LET (λvap::(real, ?'a::type) cart. LET_END (LET (λvbp::(real, ?'a::type) cart. LET_END (dot (vector_sub vap (vec (0::nat))) (vector_sub vbp (vec (0::nat))) / (vector_norm (vector_sub vap (vec (0::nat))) ∗ vector_norm (vector_sub vbp (vec (0::nat)))))) (vector_sub (% (dot vc vc) vb) (% (dot vb vc) vc)))) (vector_sub (% (dot vc vc) va) (% (dot va vc) vc)))) (vector_sub ?v1.0 ?v0.0))) (vector_sub ?v3.0 ?v0.0))) (vector_sub ?v2.0 ?v0.0) = delta_x4 ?v01.0 ?v02.0 ?v03.0 ?v23.0 ?v13.0 ?v12.0 / sqrt (ups_x ?v01.0 ?v02.0 ?v12.0 ∗ ups_x ?v01.0 ?v03.0 ?v13.0)

thm Trigonometry2.FOR_LEMMA19:

∀ (v0::(real, ?'a::type) cart) (v1::(real, ?'a::type) cart) (v2::(real, ?'a::type) cart) v3::(real, ?'a::type) cart. LET (λga::real. LET_END (LET (λv01::real. LET_END (LET (λv02::real. LET_END (LET (λv03::real. LET_END (LET (λv12::real. LET_END (LET (λv13::real. LET_END (LET (λv23::real. LET_END (¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ⟶ ga = acs (delta_x4 v01 v02 v03 v23 v13 v12 / sqrt (ups_x v01 v02 v12 ∗ ups_x v01 v03 v13)))) ((distance (v2, v3))^2))) ((distance (v1, v3))^2))) ((distance (v1, v2))^2))) ((distance (v0, v3))^2))) ((distance (v0, v2))^2))) ((distance (v0, v1))^2))) (dihV v0 v1 v2 v3)

thm Trigonometry2.COMPUTE_DELTA_OVER:

(distance (?v0.0::(real, ?'a::type) cart, ?v1.0::(real, ?'a::type) cart))^2 = (?v01.0::real) ∧ (distance (?v0.0, ?v2.0::(real, ?'a::type) cart))^2 = (?v02.0::real) ∧ (distance (?v0.0, ?v3.0::(real, ?'a::type) cart))^2 = (?v03.0::real) ∧ (distance (?v1.0, ?v2.0))^2 = (?v12.0::real) ∧ (distance (?v1.0, ?v3.0))^2 = (?v13.0::real) ∧ (distance (?v2.0, ?v3.0))^2 = (?v23.0::real) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?v2.0 EMPTY))) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?v3.0 EMPTY))) ⟶ dot (vector_sub (% (dot (vector_sub ?v1.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v2.0 ?v0.0)) (% (dot (vector_sub ?v2.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v1.0 ?v0.0))) (vector_sub (% (dot (vector_sub ?v1.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v3.0 ?v0.0)) (% (dot (vector_sub ?v3.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v1.0 ?v0.0))) / (vector_norm (vector_sub (% (dot (vector_sub ?v1.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v2.0 ?v0.0)) (% (dot (vector_sub ?v2.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v1.0 ?v0.0))) ∗ vector_norm (vector_sub (% (dot (vector_sub ?v1.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v3.0 ?v0.0)) (% (dot (vector_sub ?v3.0 ?v0.0) (vector_sub ?v1.0 ?v0.0)) (vector_sub ?v1.0 ?v0.0)))) = delta_x4 ?v01.0 ?v02.0 ?v03.0 ?v23.0 ?v13.0 ?v12.0 / sqrt (ups_x ?v01.0 ?v02.0 ?v12.0 ∗ ups_x ?v01.0 ?v03.0 ?v13.0)

thm Trigonometry2.POS_COMPATIBLE_WITH_ATN2:

$(0::real) < (?a::real) \longrightarrow atn2\ (?x::real,\ ?y::real) = atn2\ (?a * ?x,\ ?a * ?y)$

thm Trigonometry2.NOT_COLLINEAR_IMP_UPS_LT:

$\neg$ *collinear* (*INSERT* (*?v0.0*::(*real*, *3*) *cart*) (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?v2.0*::(*real*, *3*) *cart*) *EMPTY*))) $\longrightarrow$ *LET* (λ*v01*::*real*. *LET_END* (*LET* (λ*v02*::*real*. *LET_END* (*LET* (λ*v12*::*real*. *LET_END* ((*0*::*real*) < *ups_x v01 v02 v12*)) ((*distance* (*?v1.0*, *?v2.0*))^2))) ((*distance* (*?v0.0*, *?v2.0*))^2))) ((*distance* (*?v0.0*, *?v1.0*))^2)

thm Trigonometry2.REAL_LT_DIV_0:

$\forall\,(a::real)\ b::real.\ (0::real) < b \longrightarrow ((0::real) < a\ /\ b) = ((0::real) < a)$

thm Trigonometry2.NOT_ZERO_EQ_POW2_LT:

$((?a::real) \neq (0::real)) = ((0::real) < ?a^2)$

thm Trigonometry.OJEKOJF:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *LET* (λ*ga*::*real*. *LET_END* (*LET* (λ*v01*::*real*. *LET_END* (*LET* (λ*v02*::*real*. *LET_END* (*LET* (λ*v03*::*real*. *LET_END* (*LET* (λ*v12*::*real*. *LET_END* (*LET* (λ*v13*::*real*. *LET_END* (*LET* (λ*v23*::*real*. *LET_END* ($\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) $\longrightarrow$ *ga* = *acs* (*delta_x4 v01 v02 v03 v23 v13 v12* / *sqrt* (*ups_x v01 v02 v12* $*$ *ups_x v01 v03 v13*)) $\wedge$ *ga* = *pi* / *real_of_nat* (*2*::*nat*) $-$ *atn2* (*sqrt* (*real_of_nat* (*4*::*nat*) $*$ (*v01* $*$ *delta_x v01 v02 v03 v23 v13 v12*)), *delta_x4 v01 v02 v03 v23 v13 v12*))) ((*distance* (*v2*, *v3*))^2))) ((*distance* (*v1*, *v3*))^2))) ((*distance* (*v1*, *v2*))^2))) ((*distance* (*v0*, *v3*))^2))) ((*distance* (*v0*, *v2*))^2))) ((*distance* (*v0*, *v1*))^2))) (*dihV v0 v1 v2 v3*)

thm Trigonometry2.LEMMA16_INTERPRETE:

$\forall$ (*v0*::(*real*, *?'a*::*type*) *cart*) (*va*::(*real*, *?'a*::*type*) *cart*) (*vb*::(*real*, *?'a*::*type*) *cart*) *vc*::(*real*, *?'a*::*type*) *cart*. $\neg$ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT va EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT vc* (*INSERT vb EMPTY*))) $\longrightarrow$ *cos* (*dihV v0 vc va vb*) = (*cos* (*arcV v0 va vb*) $-$ *cos* (*arcV v0 vc vb*) $*$ *cos* (*arcV v0 vc va*)) / (*sin* (*arcV v0 vc vb*) $*$ *sin* (*arcV v0 vc va*))

thm Trigonometry2.NOT_COLLINEAR_IMP_VEC_FOR_DIHV:

$\neg$ *collinear* (*INSERT* (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?vc*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?va*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) $\longrightarrow$ *vector_sub* (% (*dot* (*vector_sub ?vc ?v0.0*) (*vector_sub ?vc ?v0.0*)) (*vector_sub ?va ?v0.0*)) (% (*dot* (*vector_sub ?va ?v0.0*) (*vector_sub ?vc ?v0.0*)) (*vector_sub ?vc ?v0.0*)) $\neq$ *vec* (*0*::*nat*)

thm Trigonometry2.NOT_COLLINEAR_IMP_DIHV_BOUNDED:

$\neg$ *collinear* (*INSERT* (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT ?v0.0* (*INSERT ?v1.0* (*INSERT* (*?v3.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)))

$\longrightarrow$ $(0::real) \leq dihV\ ?v0.0\ ?v1.0\ ?v2.0\ ?v3.0 \wedge dihV\ ?v0.0\ ?v1.0\ ?v2.0\ ?v3.0 \leq pi$

thm Trigonometry2.DIHV_FORMULAR:

$dihV\ (?v0.0::(real,\ ?'a::type)\ cart)\ (?v1.0::(real,\ ?'a::type)\ cart)\ (?v2.0::(real,\ ?'a::type)\ cart)\ (?v3.0::(real,\ ?'a::type)\ cart) = arcV\ (vec\ (0::nat))\ (vector\_sub\ (\%\ (dot\ (vector\_sub\ ?v1.0\ ?v0.0)\ (vector\_sub\ ?v1.0\ ?v0.0))\ (vector\_sub\ ?v2.0\ ?v0.0))\ (\%\ (dot\ (vector\_sub\ ?v2.0\ ?v0.0)\ (vector\_sub\ ?v1.0\ ?v0.0))\ (vector\_sub\ ?v1.0\ ?v0.0)))\ (vector\_sub\ (\%\ (dot\ (vector\_sub\ ?v1.0\ ?v0.0)\ (vector\_sub\ ?v1.0\ ?v0.0))\ (vector\_sub\ ?v3.0\ ?v0.0))\ (\%\ (dot\ (vector\_sub\ ?v3.0\ ?v0.0)\ (vector\_sub\ ?v1.0\ ?v0.0))\ (vector\_sub\ ?v1.0\ ?v0.0)))$

thm Trigonometry2.COS_POW2_INTER:

$(cos\ (?x::real))^2 = (1::real) - (sin\ ?x)^2$

thm Trigonometry2.ISTYLPH:

$\forall (v0::(real,\ ?'a::type)\ cart)\ (v1::(real,\ ?'a::type)\ cart)\ (v2::(real,\ ?'a::type)\ cart)\ v3::(real,\ ?'a::type)\ cart.\ (0::real) \leq cos\ (arcV\ v0\ v2\ v3) \wedge dihV\ v0\ v3\ v1\ v2 = pi\ /\ real\_of\_nat\ (2::nat) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v2\ (INSERT\ v3\ EMPTY))) \wedge (?psi::real) = arcV\ v0\ v2\ v3 \wedge (?tte::real) = arcV\ v0\ v1\ v2 \longrightarrow dihV\ v0\ v1\ v2\ v3 = beta\ ?psi\ ?tte$

thm Trigonometry2.INTERS_SUBSET:

$(?P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (?a::?'a::type \Rightarrow bool) \longrightarrow SUBSET\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%132::?'a::type \Rightarrow bool.\ \exists x::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%132\ (?P\ x)\ x)))\ ?a$

thm Trigonometry2.AFFINE_SET_GENERARTED2:

$affine\ (GSPEC\ (\lambda GEN\%PVAR\%133::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%133\ (\exists t::real.\ x = vector\_add\ (\%\ t\ (?u::(real,\ ?'a::type)\ cart))\ (\%\ ((1::real) - t)\ (?v::(real,\ ?'a::type)\ cart)))\ x))$

thm Trigonometry2.BASED_POINT2:

$SUBSET\ (INSERT\ (?u::(real,\ ?'a::type)\ cart)\ (INSERT\ (?v::(real,\ ?'a::type)\ cart)\ EMPTY))\ (GSPEC\ (\lambda GEN\%PVAR\%134::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%134\ (\exists t::real.\ x = vector\_add\ (\%\ t\ ?u)\ (\%\ ((1::real) - t)\ ?v))\ x))$

thm Trigonometry2.AFFINE_AFF:

$affine\ (aff\ (?s::(real,\ ?'a::type)\ cart \Rightarrow bool))$

thm Trigonometry2.INSERT_EMPTY_SUBSET_conjunct0:

$SUBSET\ (INSERT\ (?x::?'a::type)\ (?s::?'a::type \Rightarrow bool))\ (?t::?'a::type \Rightarrow bool) = (IN\ ?x\ ?t \wedge SUBSET\ ?s\ ?t)$

thm Trigonometry2.INSERT_EMPTY_SUBSET:

SUBSET (*INSERT* (*?x*::*?'b*::*type*) (*?s*::*?'b*::*type* ⇒ *bool*)) (*?t*::*?'b*::*type* ⇒ *bool*) = (*IN ?x ?t* ∧ *SUBSET ?s ?t*) ∧ (∀ *s*::*?'a*::*type* ⇒ *bool*. *SUBSET EMPTY s*)

thm Trigonometry2.IN_P_HULL_INSERT:

IN (*?a*::*?'a*::*type*) (*hull* (*?P*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*INSERT ?a* (*?s*::*?'a*::*type* ⇒ *bool*)))

thm Trigonometry2.UV_IN_AFF2:

IN (*?u*::(*real*, *?'a*::*type*) *cart*) (*hull affine* (*INSERT ?u* (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ∧ *IN ?v* (*hull affine* (*INSERT ?u* (*INSERT ?v EMPTY*)))

thm Trigonometry2.AFF2:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *aff* (*INSERT u* (*INSERT v EMPTY*)) = *GSPEC* (λ*GEN%PVAR%135*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%135* (∃ *t*::*real*. *x* = *vector_add* (% *t u*) (% ((*1*::*real*) − *t*) *v*)) *x*)

thm Trigonometry2.EXISTS_PROJECTING_POINT:

∀ (*p*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. ∃ *pp*::(*real*, *?'a*::*type*) *cart*. *IN* (*vector_add u pp*) (*aff* (*INSERT u* (*INSERT v EMPTY*))) ∧ *dot* (*vector_sub p pp*) (*vector_sub u v*) = (*0*::*real*)

thm Trigonometry2.UV_IN_AFF2_IMP_TRANSABLE:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u* (*aff* (*INSERT v0* (*INSERT v1 EMPTY*))) ∧ *IN v* (*aff* (*INSERT v0* (*INSERT v1 EMPTY*))) ⟶ *dot* (*vector_sub u v0*) (*vector_sub v1 v0*) ∗ *dot* (*vector_sub v v0*) (*vector_sub v1 v0*) = *dot* (*vector_sub v1 v0*) (*vector_sub v1 v0*) ∗ *dot* (*vector_sub u v0*) (*vector_sub v v0*)

thm Trigonometry2.WHEN_K_POS_ARCV_STABLE:

(*0*::*real*) < (*?k*::*real*) ⟶ *arcV* (*vec* (*0*::*nat*)) (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*) = *arcV* (*vec* (*0*::*nat*)) *?u* (% *?k ?v*)

thm Trigonometry2.ARCV_VEC0_FORM:

arcV (*?v0.0*::(*real*, *?'a*::*type*) *cart*) (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*) = *arcV* (*vec* (*0*::*nat*)) (*vector_sub ?u ?v0.0*) (*vector_sub ?v ?v0.0*)

thm Trigonometry2.WHEN_K_POS_ARCV_STABLE2:

(*?k*::*real*) < (*0*::*real*) ⟶ *arcV* (*vec* (*0*::*nat*)) (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*) = *arcV* (*vec* (*0*::*nat*)) *?u* (% (− *?k*) *?v*)

thm Trigonometry2.WHEN_K_DIFF0_ARCV:

(*?k*::*real*) ≠ (*0*::*real*) ⟶ *arcV* (*vec* (*0*::*nat*)) (*?u*::(*real*, *?'a*::*type*) *cart*) (*?v*::(*real*, *?'a*::*type*) *cart*) = *arcV* (*vec* (*0*::*nat*)) *?u* (% |*?k*| *?v*)

thm Trigonometry2.PITHAGO_THEOREM:

dot (*?x*::(*real*, *?'a*::*type*) *cart*) (*?y*::(*real*, *?'a*::*type*) *cart*) = (*0*::*real*) ⟶ (*vector_norm* (*vector_add ?x ?y*))^2 = (*vector_norm ?x*)^2 + (*vector_norm ?y*)^2 ∧ (*vector_norm* (*vector_sub ?x ?y*))^2 = (*vector_norm ?x*)^2 + (*vector_norm ?y*)^2

thm Trigonometry2.NORM_SUB_INVERTABLE:

vector_norm (*vector_sub* (*?x*::(*real*, *?'a*::*type*) *cart*) (*?y*::(*real*, *?'a*::*type*) *cart*)) = *vector_norm* (*vector_sub ?y ?x*)

thm Trigonometry2.OTHORGONAL_WITH_COS:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart*. *v0* ≠ *w* ∧ *v0* ≠ *v1* ∧ *dot* (*vector_sub w p*) (*vector_sub v1 v0*) = (*0*::*real*) ⟶ *cos* (*arcV v0 v1 w*) = *dot* (*vector_sub p v0*) (*vector_sub v1 v0*) / (*vector_norm* (*vector_sub v1 v0*) ∗ *vector_norm* (*vector_sub w v0*))

thm Trigonometry2.SIMPLIZE_COS_IF_OTHOR:

∀ (*v0*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart*. *v0* ≠ *w* ∧ *v0* ≠ *v1* ∧ *vector_sub p v0* = % (*?k*::*real*) (*vector_sub v1 v0*) ∧ *dot* (*vector_sub w p*) (*vector_sub v1 v0*) = (*0*::*real*) ⟶ *cos* (*arcV v0 v1 w*) = *?k* ∗ (*vector_norm* (*vector_sub v1 v0*) / *vector_norm* (*vector_sub w v0*))

thm Trigonometry2.SIN_EQ_SQRT_ONE_SUB:

(*?v0.0*::(*real*, *?'a*::*type*) *cart*) ≠ (*?va*::(*real*, *?'a*::*type*) *cart*) ∧ *?v0.0* ≠ (*?vb*::(*real*, *?'a*::*type*) *cart*) ⟶ *sin* (*arcV ?v0.0 ?va ?vb*) = *sqrt* ((*1*::*real*) − (*cos* (*arcV ?v0.0 ?va ?vb*))^2)

thm Trigonometry2.SIN_DI_HOC:

(*?v0.0*::(*real*, *?'a*::*type*) *cart*) ≠ (*?w*::(*real*, *?'a*::*type*) *cart*) ∧ *?v0.0* ≠ (*?vb*::(*real*, *?'a*::*type*) *cart*) ∧ *IN* (*?p*::(*real*, *?'a*::*type*) *cart*) (*aff* (*INSERT ?v0.0* (*INSERT ?w EMPTY*))) ∧ *dot* (*vector_sub ?p ?vb*) (*vector_sub ?w ?v0.0*) = (*0*::*real*) ⟶ *sin* (*arcV ?v0.0 ?w ?vb*) = *vector_norm* (*vector_sub ?p ?vb*) / *vector_norm* (*vector_sub ?vb ?v0.0*)

thm Trigonometry2.CHANG_BIET_GI:

vector_sub (*?pu*::(*real*, *?'a*::*type*) *cart*) (*?p*::(*real*, *?'a*::*type*) *cart*) = % ((*1*::*real*) − (*?t*::*real*)) (*vector_sub* (*?w*::(*real*, *?'a*::*type*) *cart*) *?p*) ⟶ *IN ?pu* (*aff* (*INSERT ?p* (*INSERT ?w EMPTY*)))

thm Trigonometry2.SUB_DOT_EQ_0_INVERTALE:

(*dot* (*vector_sub* (*?a*::(*real*, *?'a*::*type*) *cart*) (*?b*::(*real*, *?'a*::*type*) *cart*)) (*?c*::(*real*, *?'a*::*type*) *cart*) = (*0*::*real*)) = (*dot* (*vector_sub ?b ?a*) *?c* = (*0*::*real*))

thm Trigonometry2.SIN_DI_HOC2:

(*?v0.0*::(*real*, *?'a*::*type*) *cart*) ≠ (*?w*::(*real*, *?'a*::*type*) *cart*) ∧ *?v0.0* ≠ (*?vb*::(*real*, *?'a*::*type*) *cart*) ∧ *IN* (*?p*::(*real*, *?'a*::*type*) *cart*) (*aff* (*INSERT ?v0.0* (*INSERT*

?w EMPTY))) ∧ *dot* (*vector_sub ?vb ?p*) (*vector_sub ?w ?v0.0*) = (*0::real*) ⟶ *sin* (*arcV ?v0.0 ?w ?vb*) = *vector_norm* (*vector_sub ?p ?vb*) / *vector_norm* (*vector_sub ?vb ?v0.0*)

thm Trigonometry2.KEY_LEMMA_FOR_ANGLES:

∀(*p*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) (*pu*::(*real*, *?'a*::*type*) *cart*) *pv*::(*real*, *?'a*::*type*) *cart*. *IN pu* (*aff* (*INSERT p* (*INSERT w EMPTY*))) ∧ *IN pv* (*aff* (*INSERT p* (*INSERT w EMPTY*))) ∧ *dot* (*vector_sub u pu*) (*vector_sub w p*) = (*0::real*) ∧ *dot* (*vector_sub v pv*) (*vector_sub w p*) = (*0::real*) ∧ ¬ (*p* = *u* ∨ *p* = *v* ∨ *p* = *w*) ⟶ *cos* (*arcV p w u* + *arcV p w v*) − *cos* (*arcV p u v*) = (*dot* (*vector_neg* (*vector_sub v pv*)) (*vector_sub u pu*) − *vector_norm* (*vector_sub v pv*) ∗ *vector_norm* (*vector_sub u pu*)) / (*vector_norm* (*vector_sub p u*) ∗ *vector_norm* (*vector_sub p v*))

thm Trigonometry2.ARCV_BOUNDED:

(*?v0.0*::(*real*, *?'a*::*type*) *cart*) ≠ (*?u*::(*real*, *?'a*::*type*) *cart*) ∧ *?v0.0* ≠ (*?v*::(*real*, *?'a*::*type*) *cart*) ⟶ (*0::real*) ≤ *arcV ?v0.0 ?u ?v* ∧ *arcV ?v0.0 ?u ?v* ≤ *pi*

thm Trigonometry2.COS_MONOPOLY:

∀(*a::real*) *b::real*. (*0::real*) ≤ *a* ∧ *a* ≤ *pi* ∧ (*0::real*) ≤ *b* ∧ *b* ≤ *pi* ⟶ (*a* < *b*) = (*cos b* < *cos a*)

thm Trigonometry2.COS_MONOPOLY_EQ:

∀(*a::real*) *b::real*. (*0::real*) ≤ *a* ∧ *a* ≤ *pi* ∧ (*0::real*) ≤ *b* ∧ *b* ≤ *pi* ⟶ (*a* ≤ *b*) = (*cos b* ≤ *cos a*)

thm Trigonometry2.END_POINT_ADD_IN_AFF2:

∀(*k::real*) (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN* (*vector_add u* (% *k* (*vector_sub u v*))) (*aff* (*INSERT u* (*INSERT v EMPTY*))) ∧ *IN* (*vector_add u* (% *k* (*vector_sub v u*))) (*aff* (*INSERT u* (*INSERT v EMPTY*)))

thm Trigonometry2.EXISTS_PROJECTING_POINT2:

∀(*p*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *v0*::(*real*, *?'a*::*type*) *cart*. ∃*pp*::(*real*, *?'a*::*type*) *cart*. *IN pp* (*aff* (*INSERT u* (*INSERT v0 EMPTY*))) ∧ *dot* (*vector_sub p pp*) (*vector_sub u v0*) = (*0::real*)

thm Trigonometry2.KEY_LEMMA_FOR_ANGLES1:

∀(*p*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) (*pu*::(*real*, *?'a*::*type*) *cart*) *pv*::(*real*, *?'a*::*type*) *cart*. *IN pu* (*aff* (*INSERT w* (*INSERT p EMPTY*))) ∧ *IN pv* (*aff* (*INSERT w* (*INSERT p EMPTY*))) ∧ *dot* (*vector_sub u pu*) (*vector_sub w p*) = (*0::real*) ∧ *dot* (*vector_sub v pv*) (*vector_sub w p*) = (*0::real*) ∧ ¬ (*p* = *u* ∨ *p* = *v* ∨ *p* = *w*) ⟶ *cos* (*arcV p w u* + *arcV p w v*) − *cos* (*arcV p u v*) = (*dot* (*vector_neg* (*vector_sub v pv*)) (*vector_sub u pu*) − *vector_norm* (*vector_sub* v pv) ∗ vector_norm (vector_sub u pu)) / (vector_norm (vector_sub p u) ∗ vector_norm (vector_sub p v))

thm Trigonometry.KEITDWB:

∀(p::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) x::(real, ?'a::type) cart. p ≠ x ∧ p ≠ u ∧ p ≠ v ⟶ arcV p u v ≤ arcV p u x + arcV p x v

thm Trigonometry2.FGNMPAV:

∀(p::(real, ?'a::type) cart) (n::nat) fv::nat ⇒ (real, ?'a::type) cart. (2::nat) ≤ n ∧ (∀i≤n. p ≠ fv i) ⟶ arcV p (fv (0::nat)) (fv n) ≤ sum (dotdot (0::nat) (n − (1::nat))) (λi::nat. arcV p (fv i) (fv (i + (1::nat))))

thm Trigonometry2.IN_A_PERIOD_IDE0:

(0::real) ≤ (?t12.0::real) ∧ ?t12.0 < real_of_nat (2::nat) ∗ pi ∧ ?t12.0 = real_of_nat (2::nat) ∗ (pi ∗ real_of_int (?k12.0::int)) ⟶ ?t12.0 = (0::real)

thm Trigonometry2.UNIQUE_PROPERTY_IN_A_PERIOD:

(0::real) ≤ (?t12.0::real) ∧ ?t12.0 < real_of_nat (2::nat) ∗ pi ∧ (0::real) ≤ (?a::real) ∧ ?a < real_of_nat (2::nat) ∗ pi ∧ ?t12.0 = ?a + real_of_nat (2::nat) ∗ (pi ∗ real_of_int (?k12.0::int)) ⟶ ?t12.0 = ?a

thm Trigonometry.PDPFQUK:

∀(t1::real) (t2::real) (k12::int) k21::int. (0::real) ≤ t1 ∧ t1 < real_of_nat (2::nat) ∗ pi ∧ (0::real) ≤ t2 ∧ t2 < real_of_nat (2::nat) ∗ pi ∧ (0::real) ≤ (?t12.0::real) ∧ ?t12.0 < real_of_nat (2::nat) ∗ pi ∧ (0::real) ≤ (?t21.0::real) ∧ ?t21.0 < real_of_nat (2::nat) ∗ pi ∧ ?t12.0 = t1 − t2 + real_of_nat (2::nat) ∗ (pi ∗ real_of_int k12) ∧ ?t21.0 = t2 − t1 + real_of_nat (2::nat) ∗ (pi ∗ real_of_int k21) ⟶ (t1 = t2 ⟶ ?t12.0 + ?t21.0 = (0::real)) ∧ (t1 ≠ t2 ⟶ ?t12.0 + ?t21.0 = real_of_nat (2::nat) ∗ pi)

thm Trigonometry2.QAFHJNM:

∀(v::(real, ?'a::type) cart) (w::(real, ?'a::type) cart) x::(real, ?'a::type) cart. LET (λe3::(real, ?'a::type) cart. LET_END (LET (λr::real. LET_END (LET (λphi::real. LET_END (v ≠ x ∧ v ≠ w ⟶ (∃u'::(real, ?'a::type) cart. dot u' e3 = (0::real) ∧ x = vector_add v (vector_add u' (% (r ∗ cos phi) e3))))) (arcV v w x))) (vector_norm (vector_sub x v)))) (% ((1::real) / vector_norm (vector_sub w v)) (vector_sub w v))

thm DOT_CROSS_SELF_conjunct3:

∀(x::(real, 3) cart) y::(real, 3) cart. dot (cross y x) y = (0::real)

thm DOT_CROSS_SELF_conjunct2:

∀(x::(real, 3) cart) y::(real, 3) cart. dot (cross x y) y = (0::real)

thm DOT_CROSS_SELF_conjunct1:

$\forall(x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ dot\ x\ (cross\ y\ x) = (0::real)$

thm DOT_CROSS_SELF_conjunct0:

$\forall(x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ dot\ x\ (cross\ x\ y) = (0::real)$

thm Trigonometry2.YBXRVTS:

$\forall(v::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ x::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ v\ (INSERT\ w\ (INSERT\ u\ EMPTY))) \wedge (?n::(real, 3)\ cart) = cross\ (vector\_sub\ w\ v)\ (vector\_sub\ u\ v) \wedge IN\ x\ (aff\ (INSERT\ v\ (INSERT\ w\ (INSERT\ u\ EMPTY)))) \longrightarrow angle\ (vector\_add\ v\ ?n,\ v,\ x) = pi\ /\ real\_of\_nat\ (2::nat)$

thm Trigonometry2.GIVEN_POINT_EXISTS_2_NOT_COLLINEAR:

$\forall x::(real, 3)\ cart.\ \exists(y::(real, 3)\ cart)\ z::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY)))$

thm Trigonometry2.NOT_BASISES_EQ_VEC0:

$\neg\ (basis\ (1::nat) = vec\ (0::nat) \vee basis\ (2::nat) = vec\ (0::nat) \vee basis\ (3::nat) = vec\ (0::nat))$

thm Trigonometry2.TOW_DISTINCT_POINTS_EXISTS_RD_NOT_COLLINEAR:

$\forall(x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ x \neq y \longrightarrow (\exists z::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY))))$

thm Trigonometry2.SUBSET_IMP_SUBSET_HULL:

$SUBSET\ (?a::?'a::type \Rightarrow bool)\ (?b::?'a::type \Rightarrow bool) \longrightarrow SUBSET\ ?a\ (hull\ (?P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ ?b)$

thm Trigonometry2.INSERT_EMPTY_SUBSET_conjunct1:

$\forall s::?'a::type \Rightarrow bool.\ SUBSET\ EMPTY\ s$

thm Trigonometry2.THREE_POINT_IMP_EXISTS_3:

$\forall(v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ v3::(real, 3)\ cart.\ \exists(w2::(real, 3)\ cart)\ w3::(real, 3)\ cart.\ \neg\ collinear\ (INSERT\ v1\ (INSERT\ w2\ (INSERT\ w3\ EMPTY))) \wedge SUBSET\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))\ (hull\ affine\ (INSERT\ v1\ (INSERT\ w2\ (INSERT\ w3\ EMPTY))))$

thm Trigonometry2.SUBSET_AFFINE_HULL3_EQ_SUB_PLANE:

$(\exists(u::(real, 3)\ cart)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ SUBSET\ (?S::(real, 3)\ cart \Rightarrow bool)\ (hull\ affine\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))))) = (\exists x::(real, 3)\ cart \Rightarrow bool.\ plane\ x \wedge SUBSET\ ?S\ x)$

thm Trigonometry2.NONCOPLANAR_3_BASIS:

$\forall(v1::(real, 3)\ cart)\ (v2::(real, 3)\ cart)\ (v3::(real, 3)\ cart)\ (v0::(real, 3)\ cart)\ v::(real, 3)\ cart.\ \neg\ coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow (\exists(t1::real)\ (t2::real)\ t3::real.\ v = vector\_add\ (\%\ t1\ (vector\_sub$

$v1\ v0))\ (vector\_add\ (\%\ t2\ (vector\_sub\ v2\ v0))\ (\%\ t3\ (vector\_sub\ v3\ v0)))$ $\wedge\ (\forall (ta::real)\ (tb::real)\ tc::real.\ v = vector\_add\ (\%\ ta\ (vector\_sub\ v1\ v0))$ $(vector\_add\ (\%\ tb\ (vector\_sub\ v2\ v0))\ (\%\ tc\ (vector\_sub\ v3\ v0))) \longrightarrow ta = t1$ $\wedge\ tb = t2 \wedge tc = t3))$

thm Trigonometry2.coplanar1:

$\forall S::(real,\ 3)\ cart \Rightarrow bool.\ coplanar\ S = (\exists x::(real,\ 3)\ cart \Rightarrow bool.\ plane\ x \wedge SUBSET\ S\ x)$

thm Trigonometry2.COPLANAR_DET_VEC3_EQ_0:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.$ $coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) =$ $(det\_vec3\ (vector\_sub\ v1\ v0)\ (vector\_sub\ v2\ v0)\ (vector\_sub\ v3\ v0) = (0::real))$

thm Trigonometry2.th:

$\forall (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow (\forall x::(real,\ 3)\ cart.\ \exists (t1::real)\ (t2::real)\ t3::real.\ x = vector\_add\ (\%\ t1\ e1)\ (vector\_add\ (\%\ t2\ e2)\ (\%\ t3\ e3)) \wedge (\forall (tt1::real)\ (tt2::real)\ tt3::real.\ x = vector\_add\ (\%\ tt1\ e1)\ (vector\_add\ (\%\ tt2\ e2)\ (\%\ tt3\ e3)) \longrightarrow tt1 = t1 \wedge tt2 = t2 \wedge tt3 = t3))$

thm Trigonometry2.NOT_EQ0_IMP_TRIGIZABLE:

$\neg\ ((?x::real) = (0::real) \wedge (?y::real) = (0::real)) \longrightarrow (?x\ /\ sqrt\ (?x^2 + ?y^2))^2 + (?y\ /\ sqrt\ (?x^2 + ?y^2))^2 = (1::real)$

thm Trigonometry2.POW2_1_BOUNDED:

$(?a::real)^2 + (?b::real)^2 = (1::real) \longrightarrow |?a| \leq (1::real) \wedge |?b| \leq (1::real)$

thm Trigonometry2.SIN_COMPLEMENTIVE:

$sin\ (?x::real) = sin\ (pi - ?x)$

thm Trigonometry2.CYLINDER_CORDINATE:

$\forall (w::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ (e3::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \wedge e3 = \%\ ((1::real)\ /\ vector\_norm\ (vector\_sub\ w\ u))\ (vector\_sub\ w\ u) \wedge \neg\ IN\ x\ (aff\ (INSERT\ w\ (INSERT\ u\ EMPTY))) \wedge w \neq u \longrightarrow (\exists (r::real)\ (phi::real)\ h::real.\ (0::real) < r \wedge x = vector\_add\ u\ (vector\_add\ (\%\ (r * cos\ phi)\ e1)\ (vector\_add\ (\%\ (r * sin\ phi)\ e2)\ (\%\ h\ (vector\_sub\ w\ u)))))$

thm Trigonometry2.COS_SUM_2PI:

$\forall x::real.\ cos\ x = cos\ (real\_of\_nat\ (2::nat) * pi - x)$

thm Trigonometry2.POW2_EQ_0:

$\forall a::real.\ (a^2 = (0::real)) = (a = (0::real))$

thm Trigonometry2.UNIT_BOUNDED_IN_TOW_FORMS:

$- (1::real) \leq (?a::real) \wedge ?a \leq (1::real) \longrightarrow (0::real) \leq (1::real) - ?a^2$

thm Trigonometry2.COS_TOTAL:

$- (1::real) \leq (?a::real) \wedge ?a \leq (1::real) \longrightarrow (\exists!x::real. (0::real) \leq x \wedge x \leq pi \wedge cos\ x = ?a)$

thm Trigonometry2.SUM_POW2_EQ1_UNIQUE_TRIG:

$\forall (a::real)\ b::real.\ a^2 + b^2 = (1::real) \longrightarrow (\exists!x::real. (0::real) \leq x \wedge x < real\_of\_nat\ (2::nat) * pi \wedge cos\ x = a \wedge sin\ x = b)$

thm Trigonometry2.PERIODIC_AT0_IMP_ANY:

$\forall (f::real \Rightarrow bool)\ (p::real)\ t::?'a::type. (0::real) < p \wedge (\forall x::real. f\ x = f\ (x + p)) \longrightarrow (\exists!x::real. (0::real) \leq x \wedge x < p \wedge f\ x) = (\forall t::real. (0::real) \leq t \wedge t < p \longrightarrow (\exists!x::real. t \leq x \wedge x < t + p \wedge f\ x))$

thm Trigonometry2.SUM_TWO_POW2S:

$(0::real) \leq (?a::real)^2 + (?b::real)^2$

thm Trigonometry2.IDENT_WHEN_IDENT_SIN_COS:

$(0::real) \leq (?x'::real) \wedge ?x' < real\_of\_nat\ (2::nat) * pi \wedge (0::real) \leq (?p::real) \wedge ?p < real\_of\_nat\ (2::nat) * pi \wedge cos\ ?x' = cos\ ?p \wedge sin\ ?x' = sin\ ?p \longrightarrow ?p = ?x'$

thm Trigonometry2.UNIQUE_EXISTSENCE_OF_RPHIH:

$\forall (w::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ (e3::(real, 3)\ cart)\ x::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \wedge e3 = \%\ ((1::real)\ /\ vector\_norm\ (vector\_sub\ w\ u))\ (vector\_sub\ w\ u) \wedge \neg\ IN\ x\ (aff\ (INSERT\ w\ (INSERT\ u\ EMPTY))) \wedge w \neq u \longrightarrow (\exists (r::real)\ (phii::real)\ h::real. ((0::real) \leq phii \wedge phii < real\_of\_nat\ (2::nat) * pi \wedge (0::real) < r \wedge x = vector\_add\ u\ (vector\_add\ (\%\ (r * cos\ phii)\ e1)\ (vector\_add\ (\%\ (r * sin\ phii)\ e2)\ (\%\ h\ (vector\_sub\ w\ u))))) \wedge (\forall (rr::real)\ (p::real)\ hh::real. (0::real) \leq p \wedge p < real\_of\_nat\ (2::nat) * pi \wedge (0::real) < rr \wedge x = vector\_add\ u\ (vector\_add\ (\%\ (rr * cos\ p)\ e1)\ (vector\_add\ (\%\ (rr * sin\ p)\ e2)\ (\%\ hh\ (vector\_sub\ w\ u)))) \longrightarrow rr = r \wedge p = phii \wedge hh = h))$

thm Trigonometry2.REAL_EXISTS_UNIQUE_TRANSABLE:

$\forall (f::real \Rightarrow bool)\ t::real. (\exists!x::real. f\ x) = (\exists!x::real. f\ (x - t))$

thm Trigonometry2.COND_FOR_EXISTS_ANY_PERI:

$(0::real) < (?p::real) \wedge (\forall x::real. (?f::real \Rightarrow bool)\ x = ?f\ (x + ?p)) \wedge (\exists!x::real. (0::real) \leq x \wedge x < ?p \wedge ?f\ x) \longrightarrow (\forall t::real. (0::real) \leq t \wedge t < ?p \longrightarrow (\exists!x::real. t \leq x \wedge x < t + ?p \wedge ?f\ x))$

thm Trigonometry2.IN_ORIGIN_PERIOD_IMP_UNIQUENESS:

$\forall (x::real)\ t::real. (0::real) \leq t \wedge t < real\_of\_nat\ (2::nat) * pi \longrightarrow (\exists!gg::real. (0::real) \leq gg \wedge gg < real\_of\_nat\ (2::nat) * pi \wedge cos\ x = cos\ (t + gg) \wedge sin\ x = sin\ (t + gg))$

thm Trigonometry2.GIVEN_VALUED_IMP_UNIQUE_EXISTENCE:

$\forall$ *x0::real.* $\exists$!*x::real.* (*0::real*) $\leq$ *x* $\wedge$ *x* < *real_of_nat* (*2::nat*) $*$ *pi* $\wedge$ *cos x* = *cos x0* $\wedge$ *sin x* = *sin x0*

thm Trigonometry2.EYFCXPP:

$\forall$ (*w::*(*real*, *3*) *cart*) (*u::*(*real*, *3*) *cart*) (*e1::*(*real*, *3*) *cart*) (*e2::*(*real*, *3*) *cart*) (*e3::*(*real*, *3*) *cart*) (*x1::*(*real*, *3*) *cart*) *x2::*(*real*, *3*) *cart. orthonormal e1 e2 e3* $\wedge$ *e3* = % ((*1::real*) / *vector_norm* (*vector_sub w u*)) (*vector_sub w u*) $\wedge$ $\neg$ *IN x1* (*aff* (*INSERT w* (*INSERT u EMPTY*))) $\wedge$ $\neg$ *IN x2* (*aff* (*INSERT w* (*INSERT u EMPTY*))) $\wedge$ *w* $\neq$ *u* $\longrightarrow$ ($\exists$ (*r1::real*) (*r2::real*) (*phii::real*) (*ssi::real*) (*h1::real*) *h2::real.* ((*0::real*) $\leq$ *phii* $\wedge$ *phii* < *real_of_nat* (*2::nat*) $*$ *pi* $\wedge$ (*0::real*) $\leq$ *ssi* $\wedge$ *ssi* < *real_of_nat* (*2::nat*) $*$ *pi* $\wedge$ (*0::real*) < *r1* $\wedge$ (*0::real*) < *r2* $\wedge$ *x1* = *vector_add u* (*vector_add* (% (*r1* $*$ *cos phii*) *e1*) (*vector_add* (% (*r1* $*$ *sin phii*) *e2*) (% *h1* (*vector_sub w u*)))) $\wedge$ *x2* = *vector_add u* (*vector_add* (% (*r2* $*$ *cos* (*phii* + *ssi*)) *e1*) (*vector_add* (% (*r2* $*$ *sin* (*phii* + *ssi*)) *e2*) (% *h2* (*vector_sub w u*))))) $\wedge$ ($\forall$ (*rr1::real*) (*rr2::real*) (*pphii::real*) (*ssii::real*) (*h11::real*) *h22::real.* (*0::real*) $\leq$ *pphii* $\wedge$ *pphii* < *real_of_nat* (*2::nat*) $*$ *pi* $\wedge$ (*0::real*) $\leq$ *ssii* $\wedge$ *ssii* < *real_of_nat* (*2::nat*) $*$ *pi* $\wedge$ (*0::real*) < *rr1* $\wedge$ (*0::real*) < *rr2* $\wedge$ *x1* = *vector_add u* (*vector_add* (% (*rr1* $*$ *cos pphii*) *e1*) (*vector_add* (% (*rr1* $*$ *sin pphii*) *e2*) (% *h11* (*vector_sub w u*)))) $\wedge$ *x2* = *vector_add u* (*vector_add* (% (*rr2* $*$ *cos* (*pphii* + *ssii*)) *e1*) (*vector_add* (% (*rr2* $*$ *sin* (*pphii* + *ssii*)) *e2*) (% *h22* (*vector_sub w u*)))) $\longrightarrow$ *rr1* = *r1* $\wedge$ *rr2* = *r2* $\wedge$ *pphii* = *phii* $\wedge$ *ssii* = *ssi* $\wedge$ *h11* = *h1* $\wedge$ *h22* = *h2*))

thm Trigonometry2.INTERGRAL_UNIONS_INTERVALS:

$\forall$ *N::nat. UNIONS* (*GSPEC* (λ*GEN%PVAR%140::real* $\Rightarrow$ *bool.* $\exists$ *n::nat. SETSPEC GEN%PVAR%140* ((*0::nat*) < *n* $\wedge$ *n* $\leq$ *N*) (*GSPEC* (λ*GEN%PVAR%139::real.* $\exists$ *x::real. SETSPEC GEN%PVAR%139* (*real_of_nat* (*n* − (*1::nat*)) $\leq$ *x* $\wedge$ *x* < *real_of_nat n*) *x*)))) = *GSPEC* (λ*GEN%PVAR%141::real.* $\exists$ *x::real. SETSPEC GEN%PVAR%141* ((*0::real*) $\leq$ *x* $\wedge$ *x* < *real_of_nat N*) *x*)

thm Trigonometry2.EXISTS_IN_UNIT_INTERVAL:

$\forall$ *x::real.* $\exists$ *n::int.* (*0::real*) $\leq$ *x* + *real_of_int n* $\wedge$ *x* + *real_of_int n* < (*1::real*)

thm Trigonometry2.MOVE_TO_UNIT_INTERVAL:

$\forall$ *x::real.* $\exists$ *n::int.* (*0::real*) $\leq$ *x* + *real_of_int n* $*$ (*real_of_nat* (*2::nat*) $*$ *pi*) $\wedge$ *x* + *real_of_int n* $*$ (*real_of_nat* (*2::nat*) $*$ *pi*) < *real_of_nat* (*2::nat*) $*$ *pi*

thm Trigonometry2.SIN_PERIODIC_IN_WHOLE:

$\forall$ *n::int. sin* ((*?x::real*) + *real_of_int n* $*$ (*real_of_nat* (*2::nat*) $*$ *pi*)) = *sin ?x*

thm Collect_geom.SUB_SUM_SUB_conjunct0:

(*?a::real*) − ((*?b::real*) + (*?c::real*)) = *?a* − *?b* − *?c*

thm Trigonometry2.COS_PERIODIC_IN_WHOLE:

cos ((*?x::real*) + *real_of_int* (*?n::int*) $*$ (*real_of_nat* (*2::nat*) $*$ *pi*)) = *cos ?x*

thm Trigonometry2.SIN_COS_PERIODIC_IN_WHOLE:

$\forall (n::int)\ x::real.\ sin\ (x + real\_of\_int\ n * (real\_of\_nat\ (2::nat) * pi)) = sin\ x \wedge cos\ (x + real\_of\_int\ n * (real\_of\_nat\ (2::nat) * pi)) = cos\ x$

thm Trigonometry2.SIN_COS_IDEN_IFF_DIFFER_PERS:

$\forall (x::real)\ y::real.\ (cos\ x = cos\ y \wedge sin\ x = sin\ y) = (\exists k::int.\ x = y + real\_of\_int\ k * (real\_of\_nat\ (2::nat) * pi))$

thm Trigonometry2.NOT_EQ_IMP_AFF_AND_COLL3:

$\forall (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ v \neq w \longrightarrow IN\ u\ (aff\ (INSERT\ v\ (INSERT\ w\ EMPTY))) = collinear\ (INSERT\ v\ (INSERT\ w\ (INSERT\ u\ EMPTY)))$

thm Trigonometry2.R_SIN_CIRCLE:

$\forall (r::real)\ x::real.\ (r * cos\ x)^2 + (r * sin\ x)^2 = r^2$

thm Trigonometry2.R_SIN_COS_IDENT:

$\forall (r::real)\ (rr::real)\ (x::real)\ y::real.\ (0::real) \leq r \wedge (0::real) \leq rr \wedge r * cos\ x = rr * cos\ y \wedge r * sin\ x = rr * sin\ y \longrightarrow r = rr \wedge (r = (0::real) \vee cos\ x = cos\ y \wedge sin\ x = sin\ y)$

thm Trigonometry2.R_POS_SIN_COS_IDENT:

$\forall (r::real)\ (rr::real)\ (x::real)\ y::real.\ (0::real) < r \wedge (0::real) < rr \wedge r * cos\ x = rr * cos\ y \wedge r * sin\ x = rr * sin\ y \longrightarrow r = rr \wedge cos\ x = cos\ y \wedge sin\ x = sin\ y$

thm Trigonometry2.BEGIN_POINT_PERIODIC:

$\forall (x::real)\ k::int.\ (0::real) \leq x \wedge x < real\_of\_nat\ (2::nat) * pi \wedge x = real\_of\_int\ k * (real\_of\_nat\ (2::nat) * pi) \longrightarrow x = (0::real)$

thm Trigonometry2.BODE_YEU_ANH_DI:

$\forall k::int.\ (0::real) \leq (?ppsssi::real) \wedge ?ppsssi < real\_of\_nat\ (2::nat) * pi \wedge (0::real) \leq (?ppsssi1.0::real) \wedge ?ppsssi1.0 < real\_of\_nat\ (2::nat) * pi \wedge (0::real) \leq (?aa::real) \wedge ?aa < real\_of\_nat\ (2::nat) * pi \wedge ?aa = ?ppsssi - ?ppsssi1.0 + real\_of\_int\ k * (real\_of\_nat\ (2::nat) * pi) \longrightarrow (?aa = (0::real)) = (?ppsssi = ?ppsssi1.0)$

thm Trigonometry2.ORTHONORMAL_BASIS:

$orthonormal\ (basis\ (1::nat))\ (basis\ (2::nat))\ (basis\ (3::nat))$

thm Trigonometry2.ORTHO_IMP_NORM_CROSS_PRODUCT:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ dot\ x\ y = (0::real) \longrightarrow (vector\_norm\ (cross\ x\ y))^2 = (vector\_norm\ x * vector\_norm\ y)^2$

thm Trigonometry2.TWO_UNIT_ORTH_VECTORS_IMP_ORTHONORMAL:

∀ (*e1*::(*real*, *3*) *cart*) *e3*::(*real*, *3*) *cart*. *vector_norm e1* = (*1*::*real*) ∧ *vector_norm e3* = (*1*::*real*) ∧ *dot e1 e3* = (*0*::*real*) ⟶ (∃ *e2*::(*real*, *3*) *cart*. *orthonormal e1 e2 e3*)

thm Trigonometry2.ORTHONORMAL_BASIS3:

dot (*basis* (*1*::*nat*)) (*basis* (*1*::*nat*)) = (*1*::*real*) ∧ *dot* (*basis* (*2*::*nat*)) (*basis* (*2*::*nat*)) = (*1*::*real*) ∧ *dot* (*basis* (*3*::*nat*)) (*basis* (*3*::*nat*)) = (*1*::*real*) ∧ *dot* (*basis* (*1*::*nat*)) (*basis* (*2*::*nat*)) = (*0*::*real*) ∧ *dot* (*basis* (*1*::*nat*)) (*basis* (*3*::*nat*)) = (*0*::*real*) ∧ *dot* (*basis* (*2*::*nat*)) (*basis* (*3*::*nat*)) = (*0*::*real*) ∧ (*0*::*real*) < *dot* (*cross* (*basis* (*1*::*nat*)) (*basis* (*2*::*nat*))) (*basis* (*3*::*nat*))

thm Trigonometry2.EXISTS_OTHOR_VECTOR_DIFFF_VEC0:

∀ *u*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *v* ≠ *vec* (*0*::*nat*) ∧ *dot u v* = (*0*::*real*)

thm Trigonometry2.INVERT_NORM_POS_LE:

∀ *x*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) ≤ (*1*::*real*) / *vector_norm x*

thm Trigonometry2.NOT_0_INVERTABLE:

((*?a*::*real*) ≠ (*0*::*real*)) = ((*1*::*real*) / *?a* ∗ *?a* = (*1*::*real*))

thm Trigonometry2.NOT_VEC0_UNITABLE:

∀ *u*::(*real*, *?'a*::*type*) *cart*. (*u* ≠ *vec* (*0*::*nat*)) = (*vector_norm* (% ((*1*::*real*) / *vector_norm u*) *u*) = (*1*::*real*))

thm Trigonometry2.EXISTS_UNIT_OTHOR_VECTOR:

∀ *u*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *vector_norm v* = (*1*::*real*) ∧ *dot u v* = (*0*::*real*)

thm Trigonometry2.AFF3_TRANSLATION_IMAGE:

aff (*IMAGE* (*vector_add* (*?v*::(*real*, *?'a*::*type*) *cart*)) (*INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v3.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)))) = *IMAGE* (*vector_add ?v*) (*aff* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*))))

thm Trigonometry2.IMAGE_INTER_AFF3:

HOL_Light_Import.INTER (*IMAGE* (*vector_add* (*?v*::(*real*, *?'a*::*type*) *cart*)) (*?s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*)) (*aff* (*IMAGE* (*vector_add ?v*) (*INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v3.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))))) = *IMAGE* (*vector_add ?v*) (*HOL_Light_Import.INTER ?s* (*aff* (*INSERT ?v1.0* (*INSERT ?v2.0* (*INSERT ?v3.0 EMPTY*)))))

thm Trigonometry2.DIHV_TRASABLE:

∀ *v*::(*real*, *?'a*::*type*) *cart*. *dihV* (*vector_add v* (*?u*::(*real*, *?'a*::*type*) *cart*)) (*vector_add v* (*?w*::(*real*, *?'a*::*type*) *cart*)) (*vector_add v* (*?v1.0*::(*real*, *?'a*::*type*) *cart*)) (*vector_add v* (*?v2.0*::(*real*, *?'a*::*type*) *cart*)) = *dihV ?u ?w ?v1.0 ?v2.0*

thm Trigonometry2.VECTOR_MUL_R_TO_L:

$\forall (a::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ a \neq (0::real) \wedge \%\ a\ x = y \longrightarrow x = \%\ ((1::real)\ /\ a)\ y$

thm Trigonometry2.AFF2_VEC0:

$aff\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (?w::(real,\ ?'a::type)\ cart)\ EMPTY)) = GSPEC\ (\lambda GEN\%PVAR\%144::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%144\ (\exists k::real.\ x = \%\ k\ ?w)\ x)$

thm Trigonometry2.PERPENCULAR_PART_IDENT0:

$(?w::(real,\ ?'a::type)\ cart) \neq vec\ (0::nat) \wedge vector\_sub\ (\%\ (dot\ ?w\ ?w)\ (?v1.0::(real,\ ?'a::type)\ cart))\ (\%\ (dot\ ?v1.0\ ?w)\ ?w) = vec\ (0::nat) \longrightarrow IN\ ?v1.0\ (aff\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?w\ EMPTY)))$

thm Trigonometry2.INSERT_INTER_EMPTY:

$HOL\_Light\_Import.INTER\ EMPTY\ (?s::?'a::type \Rightarrow bool) = EMPTY \wedge (HOL\_Light\_Import.INTER\ (INSERT\ (?a::?'a::type)\ ?s)\ (?ss::?'a::type \Rightarrow bool) = EMPTY) = (\neg\ IN\ ?a\ ?ss \wedge HOL\_Light\_Import.INTER\ ?s\ ?ss = EMPTY)$

thm Trigonometry2.ARCV_VEC0_ABS:

$(?ku::real) \neq (0::real) \wedge (?kv::real) \neq (0::real) \longrightarrow arcV\ (vec\ (0::nat))\ (?u::(real,\ ?'a::type)\ cart)\ (?v::(real,\ ?'a::type)\ cart) = arcV\ (vec\ (0::nat))\ (\%\ |?ku|\ ?u)\ (\%\ |?kv|\ ?v)$

thm Trigonometry2.WHEN_A_B_POS_ARCV_STABLE:

$\forall (a::real)\ (b::real)\ (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ (0::real) < a \wedge (0::real) < b \longrightarrow arcV\ (vec\ (0::nat))\ x\ y = arcV\ (vec\ (0::nat))\ (\%\ a\ x)\ (\%\ b\ y)$

thm Trigonometry2.THREE_POS_IMP_DIHV_STABLE:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ (0::real) < (?a::real) \wedge (0::real) < (?b::real) \wedge (0::real) < (?c::real) \longrightarrow dihV\ (vec\ (0::nat))\ x\ y\ z = dihV\ (vec\ (0::nat))\ (\%\ ?a\ x)\ (\%\ ?b\ y)\ (\%\ ?c\ z)$

thm Trigonometry2.VECTOR_OF_DIHV_ORTHONORMAL:

$dot\ (vector\_sub\ (\%\ (dot\ (?w::(real,\ ?'a::type)\ cart)\ ?w)\ (?v1.0::(real,\ ?'a::type)\ cart))\ (\%\ (dot\ ?v1.0\ ?w)\ ?w))\ ?w = (0::real)$

thm Trigonometry2.ORTHOGORNAL_UNITIZE:

$\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ dot\ x\ y = (0::real) \longrightarrow dot\ (\%\ ((1::real)\ /\ vector\_norm\ x)\ x)\ (\%\ ((1::real)\ /\ vector\_norm\ y)\ y) = (0::real)$

thm Trigonometry2.NOT_MUL_EQ0_EQ:

$\forall (x::real)\ y::real.\ (x * y \neq (0::real)) = (x \neq (0::real) \wedge y \neq (0::real))$

thm Trigonometry2.UNITS_NOT_EQ_0:

$\forall (x::real)\ y::real.\ x * y = (1::real) \longrightarrow x \neq (0::real) \wedge y \neq (0::real)$

thm Trigonometry2.REAL_MUL_LRINV:

$(?x::real) \neq (0::real) \longrightarrow inverse\_class.inverse\ ?x * ?x = (1::real) \wedge ?x * inverse\_class.inverse\ ?x = (1::real)$

thm Trigonometry2.NOT_EQ0_IMP_NEITHER_INVERT:

$(?a::real) \neq (0::real) \longrightarrow (1::real) / ?a \neq (0::real)$

thm Trigonometry2.PROJECTOR_NOT_EQ_VEC0:

$\forall (w::(real, ?'a::type)\ cart)\ v1::(real, ?'a::type)\ cart.\ (w \neq vec\ (0::nat) \wedge \neg\ IN\ v1\ (aff\ (INSERT\ (vec\ (0::nat))\ (INSERT\ w\ EMPTY)))) = (vector\_sub\ (\%\ (dot\ w\ w)\ v1)\ (\%\ (dot\ v1\ w)\ w) \neq vec\ (0::nat))$

thm Trigonometry2.NOT_EQ_VEC0_IMP_EQU_AFF_COLL:

$\forall (w::(real, ?'a::type)\ cart)\ u::(real, ?'a::type)\ cart.\ w \neq vec\ (0::nat) \longrightarrow IN\ u\ (aff\ (INSERT\ (vec\ (0::nat))\ (INSERT\ w\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ w\ (INSERT\ u\ EMPTY)))$

thm Trigonometry2.NOT_EQ_IMP_EXISTS_BASIC:

$\forall (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ v \neq w \longrightarrow (\exists (e1::(real, 3)\ cart)\ (e2::(real, 3)\ cart)\ e3::(real, 3)\ cart.\ orthonormal\ e1\ e2\ e3 \wedge \%\ (distance\ (w, v))\ e3 = vector\_sub\ w\ v)$

thm Trigonometry2.YVREJIS:

$\forall (v::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ w2::(real, 3)\ cart.\ cyclic\_set\ (INSERT\ w1\ (INSERT\ w2\ EMPTY))\ v\ w \longrightarrow (azim\ v\ w\ w1\ w2 = (0::real) \longrightarrow azim\ v\ w\ w1\ w2 + azim\ v\ w\ w2\ w1 = (0::real)) \wedge (azim\ v\ w\ w1\ w2 \neq (0::real) \longrightarrow azim\ v\ w\ w1\ w2 + azim\ v\ w\ w2\ w1 = real\_of\_nat\ (2::nat) * pi)$

thm Trigonometry2.INSERT_INTER_EMPTY_conjunct1:

$(HOL\_Light\_Import.INTER\ (INSERT\ (?a::?'a::type)\ (?s::?'a::type \Rightarrow bool))\ (?ss::?'a::type \Rightarrow bool) = EMPTY) = (\neg\ IN\ ?a\ ?ss \wedge HOL\_Light\_Import.INTER\ ?s\ ?ss = EMPTY)$

thm Trigonometry2.INSERT_INTER_EMPTY_conjunct0:

$HOL\_Light\_Import.INTER\ EMPTY\ (?s::?'a::type \Rightarrow bool) = EMPTY$

thm Trigonometry.QQZKTXU:

$\forall (v::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ v2::(real, 3)\ cart.\ LET\ (\lambda gammmma::real.\ LET\_END\ (HOL\_Light\_Import.INTER\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))\ (aff\ (INSERT\ v\ (INSERT\ w\ EMPTY))) = EMPTY \wedge v \neq w \longrightarrow cos\ (azim\ v\ w\ v1\ v2) = cos\ gammma))\ (dihV\ v\ w\ v1\ v2)$

thm DEF_real_itv:

$real\_itv = (\lambda(\_2149270::real)\ \_2149271::real.\ GSPEC\ (\lambda GEN\%PVAR\%145::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%145\ (\_2149270 \leq x \wedge x < \_2149271)\ x))$

thm Trigonometry2.real_itv:

$\forall (a::real)\ b::real.\ real\_itv\ a\ b = GSPEC\ (\lambda GEN\%PVAR\%145::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%145\ (a \leq x \wedge x < b)\ x)$

thm DEF_tri_itv:

$tri\_itv = (\lambda\_2149282::real.\ IN\ \_2149282\ (real\_itv\ (0::real)\ (real\_of\_nat\ (2::nat) * pi)))$

thm Trigonometry2.tri_itv_conjunct0:

$\forall x::real.\ tri\_itv\ x = IN\ x\ (real\_itv\ (0::real)\ (real\_of\_nat\ (2::nat) * pi))$

thm DEF_polar_lt:

$polar\_lt = (\lambda(\_2149287::(real,\ 2)\ cart)\ \_2149288::(real,\ 2)\ cart.\ \forall (ra::real)\ (aa::real)\ (rb::real)\ ab::real.\ (0::real) < ra \wedge (0::real) < rb \wedge \_2149287 = vector\ [ra * cos\ aa,\ ra * sin\ aa] \wedge \_2149288 = vector\ [rb * cos\ ab,\ rb * sin\ ab] \wedge tri\_itv\ aa \wedge tri\_itv\ ab \longrightarrow aa < ab \vee aa = ab \wedge ra < rb)$

thm Trigonometry2.polar_lt:

$\forall (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ polar\_lt\ a\ b = (\forall (ra::real)\ (aa::real)\ (rb::real)\ ab::real.\ (0::real) < ra \wedge (0::real) < rb \wedge a = vector\ [ra * cos\ aa,\ ra * sin\ aa] \wedge b = vector\ [rb * cos\ ab,\ rb * sin\ ab] \wedge tri\_itv\ aa \wedge tri\_itv\ ab \longrightarrow aa < ab \vee aa = ab \wedge ra < rb)$

thm DEF_polar_le:

$polar\_le = (\lambda(\_2149299::(real,\ 2)\ cart)\ \_2149300::(real,\ 2)\ cart.\ polar\_lt\ \_2149299\ \_2149300 \vee \_2149299 = \_2149300)$

thm Trigonometry2.polar_le:

$\forall (a::(real,\ 2)\ cart)\ b::(real,\ 2)\ cart.\ polar\_le\ a\ b = (polar\_lt\ a\ b \vee a = b)$

thm DEF_polar_cycle_on:

$polar\_cycle\_on = (\lambda(\_2149311::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ \_2149312::(real,\ 2)\ cart \Rightarrow bool.\ (\forall x::(real,\ 2)\ cart.\ IN\ x\ \_2149312 \longrightarrow IN\ (\_2149311\ x)\ \_2149312) \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ \_2149312 \longrightarrow polar\_lt\ x\ (\_2149311\ x) \wedge (\forall y::(real,\ 2)\ cart.\ IN\ y\ \_2149312 \longrightarrow \neg\ (polar\_lt\ x\ y \wedge polar\_lt\ y\ (\_2149311\ x))) \vee (\forall y::(real,\ 2)\ cart.\ IN\ y\ \_2149312 \longrightarrow polar\_le\ (\_2149311\ x)\ y \wedge polar\_le\ y\ x)))$

thm Trigonometry2.polar_cycle_on:

$\forall (W::(real,\ 2)\ cart \Rightarrow bool)\ f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ polar\_cycle\_on\ f\ W = ((\forall x::(real,\ 2)\ cart.\ IN\ x\ W \longrightarrow IN\ (f\ x)\ W) \wedge (\forall x::(real,\ 2)\ cart.\ IN\ x\ W \longrightarrow polar\_lt\ x\ (f\ x) \wedge (\forall y::(real,\ 2)\ cart.\ IN\ y\ W \longrightarrow \neg\ (polar\_lt\ x\ y \wedge polar\_lt\ y\ (f\ x))) \vee (\forall y::(real,\ 2)\ cart.\ IN\ y\ W \longrightarrow polar\_le\ (f\ x)\ y \wedge polar\_le\ y\ x)))$

thm DEF_pl_angle:

pl_angle = (λ*_2149323*::(*real*, *2*) *cart*. *SOME u*::*real*. *tri_itv u* ∧ (∃ *t*>*0*::*real*. *_2149323* = *vector* [*t* ∗ *cos u*, *t* ∗ *sin u*]))

thm Trigonometry2.pl_angle:

∀ *x*::(*real*, *2*) *cart*. *pl_angle x* = (*SOME u*::*real*. *tri_itv u* ∧ (∃ *t*>*0*::*real*. *x* = *vector* [*t* ∗ *cos u*, *t* ∗ *sin u*]))

thm DEF_arg_diff:

arg_diff = (λ(*_2149328*::(*real*, *2*) *cart*) *_2149329*::(*real*, *2*) *cart*. *LET* (λ*dd*::*real*. *LET_END* (*if polar_le _2149328 _2149329 then dd else dd* + *real_of_nat* (*2*::*nat*) ∗ *pi*)) (*pl_angle _2149329* − *pl_angle _2149328*))

thm Trigonometry2.arg_diff:

∀ (*b*::(*real*, *2*) *cart*) *a*::(*real*, *2*) *cart*. *arg_diff a b* = *LET* (λ*dd*::*real*. *LET_END* (*if polar_le a b then dd else dd* + *real_of_nat* (*2*::*nat*) ∗ *pi*)) (*pl_angle b* − *pl_angle a*)

thm Trigonometry2.VEC2_PRE_TRIG_FORM:

∀ *x*::(*real*, *2*) *cart*. *x* ≠ *vec* (*0*::*nat*) ⟶ ($ *x* (*1*::*nat*) / *sqrt* ($ *x* (*1*::*nat*) ∗ $ *x* (*1*::*nat*) + $ *x* (*2*::*nat*) ∗ $ *x* (*2*::*nat*)))^2 + ($ *x* (*2*::*nat*) / *sqrt* ($ *x* (*1*::*nat*) ∗ $ *x* (*1*::*nat*) + $ *x* (*2*::*nat*) ∗ $ *x* (*2*::*nat*)))^2 = (*1*::*real*)

thm Trigonometry2.PRE_TRIG_FORM_VEC2:

∀ *x*::(*real*, *2*) *cart*. *x* ≠ *vec* (*0*::*nat*) ⟶ (∃ *u*::*real*. *tri_itv u* ∧ *x* = *vector* [*vector_norm x* ∗ *cos u*, *vector_norm x* ∗ *sin u*])

thm Trigonometry2.PL_ANGLE_PROPERTY:

∀ *x*::(*real*, *2*) *cart*. *x* ≠ *vec* (*0*::*nat*) ⟶ *tri_itv* (*pl_angle x*) ∧ (∃ *t*>*0*::*real*. *x* = *vector* [*t* ∗ *cos* (*pl_angle x*), *t* ∗ *sin* (*pl_angle x*)])

thm Trigonometry2.POLAR_LT_IMP_NOT_EQ:

(*?x*::(*real*, *2*) *cart*) ≠ *vec* (*0*::*nat*) ∧ (*?y*::(*real*, *2*) *cart*) ≠ *vec* (*0*::*nat*) ⟶ *polar_lt ?x ?y* ⟶ *?x* ≠ *?y*

thm Trigonometry2.CART2_EQ:

(*vector* [*?a1.0*::*real*, *?a2.0*::*real*] = *vector* [*?b1.0*::*real*, *?b2.0*::*real*]) = (*?a1.0* = *?b1.0* ∧ *?a2.0* = *?b2.0*)

thm Trigonometry2.EXISTS_MAX_ELEMENT:

∀ (*S*::*?'a*::*type* ⇒ *bool*) *lt*::*?'a*::*type* ⇒ *?'a*::*type* ⇒ *bool*. *FINITE S* ∧ *S* ≠ *EMPTY* ∧ (∀ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *z*::*?'a*::*type*. *lt x y* ∧ *lt y z* ⟶ *lt x z*) ∧ (∀ *x*::*?'a*::*type*. ¬ *lt x x*) ∧ (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *S x* ∧ *S y* ∧ *x* ≠ *y* ⟶ *lt x y* ∨ *lt y x*) ⟶ (∃ *m*::*?'a*::*type*. *S m* ∧ (∀ *x*::*?'a*::*type*. *S x* ⟶ *lt x m* ∨ *x* = *m*))

thm Trigonometry2.NO_V0_IMP_NOT_SELF_POLLAR:

$(?x::(real,\ 2)\ cart) \neq vec\ (0::nat) \longrightarrow \neg\ polar\_lt\ ?x\ ?x$

thm Trigonometry2.EXISTS_MIN_IN_ORDERED_FINITE_SET:

$\forall (S::?'a::type \Rightarrow bool)\ lt::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ FINITE\ S \wedge S \neq EMPTY \wedge (\forall x::?'a::type.\ lt\ x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ lt\ x\ y \wedge lt\ y\ z \longrightarrow lt\ x\ z) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ lt\ x\ y \wedge lt\ y\ x \longrightarrow x = y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ lt\ x\ y \vee lt\ y\ x) \longrightarrow (\exists m::?'a::type.\ S\ m \wedge (\forall x::?'a::type.\ S\ x \longrightarrow lt\ m\ x))$

thm Trigonometry2.EXISTS_MA_OR_FI_SET:

$FINITE\ (?S::?'a::type \Rightarrow bool) \wedge ?S \neq EMPTY \wedge (\forall x::?'a::type.\ (?lt::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ ?lt\ y\ x \wedge ?lt\ z\ y \longrightarrow ?lt\ z\ x) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ ?lt\ y\ x \wedge ?lt\ x\ y \longrightarrow x = y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ ?lt\ y\ x \vee ?lt\ x\ y) \longrightarrow (\exists m::?'a::type.\ ?S\ m \wedge (\forall x::?'a::type.\ ?S\ x \longrightarrow ?lt\ x\ m))$

thm Trigonometry2.tri_itv:

$((\forall x::real.\ tri\_itv\ x = IN\ x\ (real\_itv\ (0::real)\ (real\_of\_nat\ (2::nat) * pi))) \wedge (\forall (a::real)\ b::real.\ real\_itv\ a\ b = GSPEC\ (\lambda GEN\%PVAR\%145::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%145\ (a \leq x \wedge x < b)\ x))) \wedge (\forall (P::(bool \Rightarrow ?'e::type \Rightarrow bool) \Rightarrow bool)\ x::?'e::type.\ IN\ x\ (GSPEC\ (\lambda v::?'e::type.\ P\ (SETSPEC\ v))) = P\ (\lambda(p::bool)\ t::?'e::type.\ p \wedge x = t)) \wedge (\forall (p::?'d::type \Rightarrow bool)\ x::?'d::type.\ IN\ x\ (GSPEC\ (\lambda v::?'d::type.\ \exists y::?'d::type.\ SETSPEC\ v\ (p\ y)\ y)) = p\ x) \wedge (\forall (P::(bool \Rightarrow ?'c::type \Rightarrow bool) \Rightarrow bool)\ x::?'c::type.\ GSPEC\ (\lambda v::?'c::type.\ P\ (SETSPEC\ v))\ x = P\ (\lambda(p::bool)\ t::?'c::type.\ p \wedge x = t)) \wedge (\forall (p::?'b::type \Rightarrow bool)\ x::?'b::type.\ GSPEC\ (\lambda v::?'b::type.\ \exists y::?'b::type.\ SETSPEC\ v\ (p\ y)\ y)\ x = p\ x) \wedge (\forall (p::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ p = p\ x)$

thm Trigonometry2.WHILE_POLAR_LT_IMP_ST:

$polar\_lt\ (?p0.0::(real,\ 2)\ cart)\ (?p::(real,\ 2)\ cart) \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%146::(real,\ 2)\ cart.\ \exists y::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%146\ (\exists N::nat.\ y = ITER\ N\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?p0.0 \wedge (\forall n::nat.\ (0::nat) \leq n \wedge n < N \longrightarrow polar\_lt\ (ITER\ n\ ?f\ ?p0.0)\ y) \wedge polar\_lt\ y\ ?p)\ y) \neq EMPTY$

thm Trigonometry2.DOT_ITSELF_2:

$(?x::(real,\ 2)\ cart) = vector\ [?a::real,\ ?b::real] \longrightarrow dot\ ?x\ ?x = ?a^2 + ?b^2$

thm Trigonometry2.NORM_VECTOR2_TRIG:

$(?x::(real,\ 2)\ cart) = vector\ [(?a::real) * cos\ (?t::real),\ ?a * sin\ ?t] \wedge (0::real) \leq ?a \longrightarrow vector\_norm\ ?x = ?a$

thm Trigonometry2.NOT_EQ_IMP_TOTAL_ORDER:

$\forall (x::(real,\ 2)\ cart)\ y::(real,\ 2)\ cart.\ x \neq y \longrightarrow polar\_lt\ x\ y \vee polar\_lt\ y\ x$

thm Trigonometry2.PROVE_XISTS_MAX_ELEMENT_LT_P:

$\forall W::(real,\ 2)\ cart \Rightarrow bool.\ (\forall x::(real,\ 2)\ cart.\ W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ W \wedge W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart$

⇒ (real, 2) cart) W ∧ polar_lt ?p0.0 (?p::(real, 2) cart) ∧ (?SS::(real, 2) cart ⇒ bool) = GSPEC (λGEN%PVAR%149::(real, 2) cart. ∃ y::(real, 2) cart. SETSPEC GEN%PVAR%149 (∃ N::nat. y = ITER N ?f ?p0.0 ∧ (∀ n::nat. (0::nat) ≤ n ∧ n < N ⟶ polar_lt (ITER n ?f ?p0.0) y) ∧ polar_lt y ?p) y) ⟶ (∃ mx::(real, 2) cart. IN mx ?SS ∧ (∀ x::(real, 2) cart. ?SS x ⟶ polar_lt x mx ∨ x = mx))

thm Trigonometry2.VEC0_BOTH_LT_GT:

(?y::(real, 2) cart) = vec (0::nat) ⟶ polar_lt (?x::(real, 2) cart) ?y ∧ polar_lt ?y (?z::(real, 2) cart)

thm Trigonometry2.POLAR_LT_TRANS:

(?y::(real, 2) cart) ≠ vec (0::nat) ⟶ polar_lt (?x::(real, 2) cart) ?y ∧ polar_lt ?y (?z::(real, 2) cart) ⟶ polar_lt ?x ?z

thm Trigonometry2.PROVE_EXISTING_MAX_IN_CYCLIC_FINITE_SET:

∀ W::(real, 2) cart ⇒ bool. FINITE W ∧ W ≠ EMPTY ∧ (∀ x::(real, 2) cart. W x ⟶ x ≠ vec (0::nat)) ⟶ (∃ m::(real, 2) cart. W m ∧ (∀ x::(real, 2) cart. W x ⟶ polar_lt x m ∨ x = m))

thm Trigonometry2.PROVE_MIN_ELEMENT_IN_FINITE_CYCLIC_SET:

∀ W::(real, 2) cart ⇒ bool. FINITE W ∧ W ≠ EMPTY ∧ (∀ x::(real, 2) cart. W x ⟶ x ≠ vec (0::nat)) ⟶ (∃ n::(real, 2) cart. W n ∧ (∀ x::(real, 2) cart. W x ⟶ polar_lt n x ∨ n = x))

thm Trigonometry2.TOW_NON_VEC0_IMP_NOT_REFL_POLAR_LT:

(?x::(real, 2) cart) ≠ vec (0::nat) ∧ (?y::(real, 2) cart) ≠ vec (0::nat) ⟶ ¬ (polar_lt ?x ?y ∧ polar_lt ?y ?x)

thm Hypermap.CARD_SINGLETON:

∀ x::?'a::type. CARD (INSERT x EMPTY) = (1::nat)

thm Trigonometry2.POLAR_LE_REFL_EQ:

(polar_le (?a::(real, 2) cart) (?b::(real, 2) cart) ∧ polar_le ?b ?a) = (?a = ?b ∨ ?a = vec (0::nat) ∨ ?b = vec (0::nat))

thm Trigonometry2.POLAR_MONOPOLY_IN_FIRST_ITERVAL:

(∀ x::(real, 2) cart. (?W::(real, 2) cart ⇒ bool) x ⟶ x ≠ vec (0::nat)) ∧ FINITE ?W ∧ ?W (?p0.0::(real, 2) cart) ∧ polar_cycle_on (?f::(real, 2) cart ⇒ (real, 2) cart) ?W ∧ (∀ x::(real, 2) cart. ?W x ⟶ polar_le ?p0.0 x) ∧ (?i::nat) < CARD ?W − (1::nat) ⟶ polar_lt (ITER ?i ?f ?p0.0) (?f (ITER ?i ?f ?p0.0))

thm Trigonometry2.TRANS_SUC_IMP_INCREASE:

∀ f::nat ⇒ nat ⇒ bool. (∀ (x::nat) (y::nat) z::nat. f x y ∧ f y z ⟶ f x z) ∧ (∀ i::nat. f i (i + (1::nat))) ⟶ (∀ (i::nat) j::nat. i < j ⟶ f i j)

thm Trigonometry2.MONOPOLY_IN_FIRST_PERIOD:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall (i::nat)\ j::nat.\ i < j \wedge j < CARD\ ?W \longrightarrow polar\_lt\ (ITER\ i\ ?f\ ?p0.0)\ (ITER\ j\ ?f\ ?p0.0))$

thm Trigonometry2.FINITE_SEUBSET_OF_NATURAL:

$\forall n::nat.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%162::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%162\ (i < n)\ ((?f::nat \Rightarrow ?'a::type)\ i)))$

thm Trigonometry2.STRICTLY_INCREASE_PRESERVING_CARD:

$\forall (lt::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ f::nat \Rightarrow ?'a::type.\ (\forall (x::?'a::type)\ y::?'a::type.\ lt\ x\ y \longrightarrow x \neq y) \wedge (\forall (i::nat)\ j::nat.\ i < j \longrightarrow lt\ (f\ i)\ (f\ j)) \longrightarrow (\forall n::nat.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%167::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%167\ (i < n)\ (f\ i))) = n)$

thm Trigonometry2.XXXXX:

$\forall (lt::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ f::nat \Rightarrow ?'a::type.\ (\forall (x::?'a::type)\ y::?'a::type.\ lt\ x\ y \longrightarrow x \neq y) \wedge (\forall (i::nat)\ j::nat.\ i < j \wedge j < (?N::nat) \longrightarrow lt\ (f\ i)\ (f\ j)) \longrightarrow (\forall n<?N.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%172::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%172\ (i < n)\ (f\ i))) = n)$

thm Trigonometry2.TDHUFHCYVHYBCC:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall n<CARD\ ?W.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%175::(real,\ 2)\ cart.\ \exists y::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%175\ (\exists i<n.\ y = ITER\ i\ ?f\ ?p0.0)\ y)) = n)$

thm Trigonometry2.POLAR_CYCLIC_FUN_IMP_ALL_BELONG:

$(?W::(real,\ 2)\ cart \Rightarrow bool)\ (?p::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \longrightarrow (\forall n::nat.\ ?W\ (ITER\ n\ ?f\ ?p))$

thm Trigonometry2.CARD_W_AS_ALL_LESS_THAN_PERIODIC:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall n::nat.\ n = CARD\ ?W \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%181::(real,\ 2)\ cart.\ \exists y::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%181\ (\exists i<n.\ y = ITER\ i\ ?f\ ?p0.0)\ y)) = n)$

thm Trigonometry2.AUTOMAP_IMP_ALL_ITER_IN:

$(?W::?'a::type \Rightarrow bool)\ (?p::?'a::type) \wedge (\forall x::?'a::type.\ ?W\ x \longrightarrow IN\ ((?f::?'a::type \Rightarrow ?'a::type)\ x)\ ?W) \longrightarrow (\forall N::nat.\ IN\ (ITER\ N\ ?f\ ?p)\ ?W)$

thm Trigonometry2.AUTOMAP_IMP_ITER_SET_IS_A_SUBSET:

$(?W::?'a::type \Rightarrow bool)\ (?p::?'a::type) \wedge (\forall x::?'a::type.\ ?W\ x \longrightarrow IN\ ((?f::?'a::type \Rightarrow ?'a::type)\ x)\ ?W) \longrightarrow SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%182::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%182\ (\exists n::nat.\ y = ITER\ n\ ?f\ ?p)\ y))\ ?W$

thm Trigonometry2.TOW_NON_VEC0_POLAR_LE_IMP_NOT_LT:

$(?x::(real,\ 2)\ cart) \neq vec\ (0::nat) \wedge (?y::(real,\ 2)\ cart) \neq vec\ (0::nat) \wedge polar\_le\ ?x\ ?y \longrightarrow \neg\ polar\_lt\ ?y\ ?x$

thm Trigonometry2.CARD_W_IS_THE_PERIODIC:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow ITER\ (CARD\ ?W)\ ?f\ ?p0.0 = ?p0.0$

thm Trigonometry2.ITER_CARD_W_IDENTIFICATION:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow ITER\ (CARD\ ?W)\ ?f\ x = x)$

thm Trigonometry2.EXISTS_STEPS_FOR_FOLLOWING_POINTS:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge polar\_le\ ?p0.0\ (?p::(real,\ 2)\ cart) \wedge ?W\ ?p \longrightarrow (\exists n<CARD\ ?W.\ ITER\ n\ ?f\ ?p0.0 = ?p \wedge (\forall nn<n.\ polar\_lt\ (ITER\ nn\ ?f\ ?p0.0)\ ?p))$

thm Trigonometry2.MONO_LE_IN_FIRST_PERIOD:

$(\forall x::(real,\ 2)\ cart.\ (?W::(real,\ 2)\ cart \Rightarrow bool)\ x \longrightarrow x \neq vec\ (0::nat)) \wedge FINITE\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge polar\_cycle\_on\ (?f::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall (i::nat)\ j::nat.\ i \leq j \wedge j < CARD\ ?W \longrightarrow polar\_le\ (ITER\ i\ ?f\ ?p0.0)\ (ITER\ j\ ?f\ ?p0.0))$

thm Trigonometry2.POLAR_LE_NOT_VEC0_IMP_PL_ANG_LE:

$polar\_le\ (?x::(real,\ 2)\ cart)\ (?y::(real,\ 2)\ cart) \wedge ?x \neq vec\ (0::nat) \wedge ?y \neq vec\ (0::nat) \longrightarrow pl\_angle\ ?x \leq pl\_angle\ ?y$

thm Trigonometry2.TWO_NOT_EQ_VECS_SUM_ARG_DIFF_TWO_PI:

$(?x::(real,\ 2)\ cart) \neq vec\ (0::nat) \wedge (?y::(real,\ 2)\ cart) \neq vec\ (0::nat) \wedge ?x \neq ?y \longrightarrow arg\_diff\ ?x\ ?y + arg\_diff\ ?y\ ?x = real\_of\_nat\ (2::nat) * pi$

thm Trigonometry2.ARG_DIFF_SUCCESSIBLE_IN_FIRST_PERIOD:

$\forall (W::(real,\ 2)\ cart \Rightarrow bool)\ xicm::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ FINITE\ W \wedge CARD\ W = (?n::nat) \wedge (\forall x::(real,\ 2)\ cart.\ W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge polar\_cycle\_on\ xicm\ W \longrightarrow (\forall (p::(real,\ 2)\ cart)\ (i::nat)\ j::nat.\ W\ p \wedge (0::nat) \leq i \wedge i \leq j \wedge j < ?n \longrightarrow arg\_diff\ p\ (ITER\ i\ xicm\ p) + arg\_diff\ (ITER\ i\ xicm\ p)\ (ITER\ j\ xicm\ p) = arg\_diff\ p\ (ITER\ j\ xicm\ p))$

thm Trigonometry2.TWO_NON_ZERO_VECS_NOT_EQ_EQ_PLT:

$(?x::(real,\ 2)\ cart) \neq vec\ (0::nat) \wedge (?y::(real,\ 2)\ cart) \neq vec\ (0::nat) \longrightarrow (?x \neq ?y) = (polar\_lt\ ?x\ ?y \vee polar\_lt\ ?y\ ?x)$

thm Trigonometry2.SUM_OVER_W_EQUAL_AT_ANY_POINT:

$FINITE\ (?W::(real,\ 2)\ cart \Rightarrow bool) \wedge CARD\ ?W = (?n::nat) \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge polar\_cycle\_on\ (?xicm::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge ?W\ (?p0.0::(real,\ 2)\ cart) \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow polar\_le\ ?p0.0\ x) \longrightarrow (\forall p::(real,\ 2)\ cart.\ ?W\ p \longrightarrow sum\ (dotdot\ (0::nat)\ (?n - (1::nat)))\ (\lambda i::nat.\ arg\_diff\ (ITER\ i\ ?xicm\ p)\ (ITER\ (i + (1::nat))\ ?xicm\ p)) = sum\ (dotdot\ (0::nat)\ (?n - (1::nat)))\ (\lambda i::nat.\ arg\_diff\ (ITER\ i\ ?xicm\ ?p0.0)\ (ITER\ (i + (1::nat))\ ?xicm\ ?p0.0)))$

thm Trigonometry2.SUM_INCREASE_ARG_DIFF:

$\forall (W::(real,\ 2)\ cart \Rightarrow bool)\ xicm::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ FINITE\ W \wedge CARD\ W = (?n::nat) \wedge (\forall x::(real,\ 2)\ cart.\ W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge polar\_cycle\_on\ xicm\ W \longrightarrow (\forall (p::(real,\ 2)\ cart)\ (i::nat)\ j::nat.\ W\ p \wedge (0::nat) \leq i \wedge i < j \wedge j < ?n \longrightarrow sum\ (dotdot\ i\ (j - (1::nat)))\ (\lambda i::nat.\ arg\_diff\ (ITER\ i\ xicm\ p)\ (ITER\ (i + (1::nat))\ xicm\ p)) = arg\_diff\ (ITER\ i\ xicm\ p)\ (ITER\ j\ xicm\ p))$

thm Trigonometry2.LEMMA_SUM_ALL_OVER_CYCLIC_SET:

$\forall (W::(real,\ 2)\ cart \Rightarrow bool)\ xicm::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart.\ FINITE\ W \wedge CARD\ W = (?n::nat) \wedge (\forall x::(real,\ 2)\ cart.\ W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge polar\_cycle\_on\ xicm\ W \wedge W\ (?p::(real,\ 2)\ cart) \longrightarrow (\exists (p::(real,\ 2)\ cart)\ q::(real,\ 2)\ cart.\ W\ p \wedge W\ q \wedge p \neq q) \longrightarrow sum\ (dotdot\ (0::nat)\ (?n - (1::nat)))\ (\lambda i::nat.\ arg\_diff\ (ITER\ i\ xicm\ ?p)\ (ITER\ (i + (1::nat))\ xicm\ ?p)) = real\_of\_nat\ (2::nat) * pi$

thm DEF_re_eqvl:

$re\_eqvl = (\lambda(\_2189404::real)\ \_2189405::real.\ \exists t{>}0::real.\ \_2189404 = t * \_2189405)$

thm Trigonometry2.re_eqvl:

$\forall (a::real)\ b::real.\ re\_eqvl\ a\ b = (\exists t{>}0::real.\ a = t * b)$

thm Trigonometry2.VEC_DIV_MOV:

$(?a::real) \neq (0::real) \longrightarrow (\%\ ((?b::real)\ /\ ?a)\ (?x::(real,\ ?'a::type)\ cart) = (?y::(real,\ ?'a::type)\ cart)) = (\%\ ?b\ ?x = \%\ ?a\ ?y)$

thm Trigonometry.JBDNJJB:

$\forall (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ re\_eqvl\ (sin\ (azim\ (vec\ (0::nat))\ u\ v\ w))\ (dot\ (cross\ u\ v)\ w)$

thm Trigonometry.ISRTTNZ:

$FINITE\ (?W::(real,\ 2)\ cart \Rightarrow bool) \wedge CARD\ ?W = (?n::nat) \wedge (\forall x::(real,\ 2)\ cart.\ ?W\ x \longrightarrow x \neq vec\ (0::nat)) \wedge polar\_cycle\_on\ (?xicm::(real,\ 2)\ cart \Rightarrow (real,\ 2)\ cart)\ ?W \wedge ?W\ (?p::(real,\ 2)\ cart) \wedge (\exists (p::(real,\ 2)\ cart)\ q::(real,\ 2)\ cart.\ ?W\ p \wedge ?W\ q \wedge p \neq q) \longrightarrow sum\ (dotdot\ (0::nat)\ (?n - (1::nat)))\ (\lambda i::nat.\ arg\_diff\ (ITER\ i\ ?xicm\ ?p)\ (ITER\ (i + (1::nat))\ ?xicm\ ?p)) = real\_of\_nat\ (2::nat) * pi \wedge (\forall (p::(real,\ 2)\ cart)\ (i::nat)\ j::nat.\ ?W\ p \wedge (0::nat) \leq i \wedge i \leq j \wedge j < ?n \longrightarrow arg\_diff\ p\ (ITER\ i\ ?xicm\ p) + arg\_diff\ (ITER\ i\ ?xicm\ p)\ (ITER\ j\ ?xicm\ p) = arg\_diff\ p\ (ITER\ j\ ?xicm\ p))$

thm Vukhacky_tactics.HAS_REAL_DERIVATIVE_CHAIN2:

$\forall (P::real \Rightarrow bool)\ (f::real \Rightarrow real)\ (g::real \Rightarrow real)\ (x::real)\ s::real \Rightarrow bool.\ (\forall x::real.\ P\ x \longrightarrow has\_real\_derivative\ g\ ((?g'::real \Rightarrow real)\ x)\ (atreal\ x)) \longrightarrow has\_real\_derivative\ f\ (?f'::real)\ (within\ (atreal\ x)\ s) \wedge P\ (f\ x) \longrightarrow has\_real\_derivative\ (\lambda x::real.\ g\ (f\ x))\ (?f' * ?g'\ (f\ x))\ (within\ (atreal\ x)\ s)$

thm Vukhacky_tactics.REDUCE_WITH_DIV_Euler_lemma:

$\forall (x::real)\ (y::real)\ z::real.\ y \neq (0::real) \wedge z \neq (0::real) \longrightarrow x * (y\ /\ (z * y)) = x\ /\ z$

thm Compute_2158872499.ATN_UPS_X_BREAKDOWN1:

$\forall (a::real)\ (b::real)\ c::real.\ (0::real) < (a + (b + c)) * ((a + (b - c)) * ((b + (c - a)) * (c + (a - b)))) \longrightarrow arclength\ a\ b\ c = pi\ /\ real\_of\_nat\ (2::nat) + atn\ ((c * c - a * a - b * b)\ /\ sqrt\ ((a + (b + c)) * ((a + (b - c)) * ((b + (c - a)) * (c + (a - b))))))$

thm Compute_2158872499.compute_one_first:

$\forall (y1::real)\ (y2::real)\ (s::real \Rightarrow bool)\ (hz::real)\ (g::real \Rightarrow real)\ x::real.\ (1::real) \leq hz \wedge hz < real\_of\_nat\ (2::nat) \wedge real\_of\_nat\ (2::nat) \leq y1 \wedge y1 \leq real\_of\_nat\ (2::nat) * hz \wedge real\_of\_nat\ (2::nat) \leq y2 \wedge y2 \leq real\_of\_nat\ (2::nat) * hz \wedge s = GSPEC\ (\lambda GEN\%PVAR\%190::real.\ \exists t::real.\ SETSPEC\ GEN\%PVAR\%190\ (y1 - real\_of\_nat\ (4::nat) < t \wedge t < real\_of\_nat\ (4::nat) - y2)\ t) \wedge g = (\lambda t::real.\ arclength\ y1\ (y2 + t)\ (real\_of\_nat\ (2::nat))) \wedge IN\ x\ s \wedge (?g'::real \Rightarrow real) = (\lambda x::real.\ - ((y2 + x)^2 - y1^2 + real\_of\_nat\ (4::nat))\ /\ ((y2 + x) * sqrt\ ((y1 + (y2 + x + real\_of\_nat\ (2::nat))) * ((y1 + (y2 + x - real\_of\_nat\ (2::nat))) * ((y2 + x + (real\_of\_nat\ (2::nat) - y1)) * (real\_of\_nat\ (2::nat) + (y1 - (y2 + x)))))))) \longrightarrow has\_real\_derivative\ g\ (?g'\ x)\ (within\ (atreal\ x)\ s)$

thm Compute_2158872499.compute_one_second:

$\forall (y1::real)\ (y2::real)\ (s::real \Rightarrow bool)\ (hz::real)\ (g::real \Rightarrow real)\ x::?'a::type.\ (1::real) \leq hz \wedge hz < real\_of\_nat\ (2::nat) \wedge real\_of\_nat\ (2::nat) \leq y1 \wedge y1 \leq real\_of\_nat\ (2::nat) * hz \wedge real\_of\_nat\ (2::nat) \leq y2 \wedge y2 \leq real\_of\_nat\ (2::nat) * hz \wedge s = GSPEC\ (\lambda GEN\%PVAR\%192::real.\ \exists t::real.\ SETSPEC$

GEN%PVAR%192 (*y1* − *real_of_nat* (*4*::*nat*) < *t* ∧ *t* < *real_of_nat* (*4*::*nat*) − *y2*) *t*) ∧ *g* = (λ*t*::*real*. − ((*y2* + *t*)^{2} − *y1*^{2} + *real_of_nat* (*4*::*nat*)) / ((*y2* + *t*) ∗ *sqrt* ((*y1* + (*y2* + *t* + *real_of_nat* (*2*::*nat*))) ∗ ((*y1* + (*y2* + *t* − *real_of_nat* (*2*::*nat*))) ∗ ((*y2* + *t* + (*real_of_nat* (*2*::*nat*) − *y1*)) ∗ (*real_of_nat* (*2*::*nat*) + (*y1* − (*y2* + *t*)))))))) ⟶ *has_real_derivative* *g* ((− *real_of_nat* (*64*::*nat*) + (*real_of_nat* (*48*::*nat*) ∗ *y1*^{2} − *real_of_nat* (*12*::*nat*) ∗ *y1*$^{4::nat}$ + (*y1*$^{6::nat}$ + (*real_of_nat* (*80*::*nat*) ∗ *y2*^{2} − *real_of_nat* (*8*::*nat*) ∗ (*y1*^{2} ∗ *y2*^{2}) − *real_of_nat* (*3*::*nat*) ∗ (*y1*$^{4::nat}$ ∗ *y2*^{2}) − *real_of_nat* (*12*::*nat*) ∗ *y2*$^{4::nat}$ + (*real_of_nat* (*3*::*nat*) ∗ (*y1*^{2} ∗ *y2*$^{4::nat}$) − *y2*$^{6::nat}$))))) / (*y2*^{2} ∗ (*sqrt* (*ups_x* (*y1*^{2}) (*y2*^{2}) (*real_of_nat* (*4*::*nat*))) ∗ *ups_x* (*y1*^{2}) (*y2*^{2}) (*real_of_nat* (*4*::*nat*))))) (*within* (*atreal* (*0*::*real*)) *s*)

thm Compute_2158872499.COMPUTE_DERIVATIVE_ONE:

∀ (*y1*::*real*) (*y2*::*real*) (*s*::*real* ⇒ *bool*) (*hz*::*real*) (*g*::*real* ⇒ *real*) (*x*::*?'b*::*type*) *f*::*?'a*::*type*. (*1*::*real*) ≤ *hz* ∧ *hz* < *real_of_nat* (*2*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *y1* ∧ *y1* ≤ *real_of_nat* (*2*::*nat*) ∗ *hz* ∧ *real_of_nat* (*2*::*nat*) ≤ *y2* ∧ *y2* ≤ *real_of_nat* (*2*::*nat*) ∗ *hz* ∧ *s* = *GSPEC* (λ*GEN%PVAR%193*::*real*. ∃ *t*::*real*. *SETSPEC GEN%PVAR%193* (*y1* − *real_of_nat* (*4*::*nat*) < *t* ∧ *t* < *real_of_nat* (*4*::*nat*) − *y2*) *t*) ∧ *g* = (λ*t*::*real*. *arclength* *y1* (*y2* + *t*) (*real_of_nat* (*2*::*nat*))) ∧ (*?g'*::*real* ⇒ *real*) = (λ*t*::*real*. − ((*y2* + *t*)^{2} − *y1*^{2} + *real_of_nat* (*4*::*nat*)) / ((*y2* + *t*) ∗ *sqrt* ((*y1* + (*y2* + *t* + *real_of_nat* (*2*::*nat*))) ∗ ((*y1* + (*y2* + *t* − *real_of_nat* (*2*::*nat*))) ∗ ((*y2* + *t* + (*real_of_nat* (*2*::*nat*) − *y1*)) ∗ (*real_of_nat* (*2*::*nat*) + (*y1* − (*y2* + *t*)))))))) ⟶ (∀ *x*::*real*. *IN* *x* *s* ⟶ *has_real_derivative* *g* (*?g'* *x*) (*within* (*atreal* *x*) *s*)) ∧ *has_real_derivative* *?g'* ((− *real_of_nat* (*64*::*nat*) + (*real_of_nat* (*48*::*nat*) ∗ *y1*^{2} − *real_of_nat* (*12*::*nat*) ∗ *y1*$^{4::nat}$ + (*y1*$^{6::nat}$ + (*real_of_nat* (*80*::*nat*) ∗ *y2*^{2} − *real_of_nat* (*8*::*nat*) ∗ (*y1*^{2} ∗ *y2*^{2}) − *real_of_nat* (*3*::*nat*) ∗ (*y1*$^{4::nat}$ ∗ *y2*^{2}) − *real_of_nat* (*12*::*nat*) ∗ *y2*$^{4::nat}$ + (*real_of_nat* (*3*::*nat*) ∗ (*y1*^{2} ∗ *y2*$^{4::nat}$) − *y2*$^{6::nat}$))))) / (*y2*^{2} ∗ (*sqrt* (*ups_x* (*y1*^{2}) (*y2*^{2}) (*real_of_nat* (*4*::*nat*))) ∗ *ups_x* (*y1*^{2}) (*y2*^{2}) (*real_of_nat* (*4*::*nat*))))) (*within* (*atreal* (*0*::*real*)) *s*)

thm Compute_2158872499.compute_two_first:

∀ (*y1*::*real*) (*y2*::*real*) (*s*::*real* ⇒ *bool*) (*hz*::*real*) (*g*::*real* ⇒ *real*) *x*::*real*. (*1*::*real*) ≤ *hz* ∧ *hz* < *real_of_nat* (*2*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *y1* ∧ *y1* ≤ *real_of_nat* (*2*::*nat*) ∗ *hz* ∧ *real_of_nat* (*2*::*nat*) ≤ *y2* ∧ *y2* ≤ *real_of_nat* (*2*::*nat*) ∗ *hz* ∧ *s* = *GSPEC* (λ*GEN%PVAR%197*::*real*. ∃ *t*::*real*. *SETSPEC GEN%PVAR%197* (*y2* − *real_of_nat* (*2*::*nat*) − *y1* < *real_of_nat* (*2*::*nat*) ∗ *t* ∧ *real_of_nat* (*2*::*nat*) ∗ *t* < *y2* + (*real_of_nat* (*2*::*nat*) − *y1*)) *t*) ∧ *g* = (λ*t*::*real*. *arclength* (*y1* + *t*) (*y2* − *t*) (*real_of_nat* (*2*::*nat*))) ∧ *IN* *x* *s* ∧ (*?g'*::*real* ⇒ *real*) = (λ*x*::*real*. (*y2* − *y1* − *real_of_nat* (*2*::*nat*) ∗ *x*) ∗ ((*y2* + *y1*)^{2} − *real_of_nat* (*4*::*nat*)) / ((*y1* + *x*) ∗ ((*y2* − *x*) ∗ *sqrt* ((*y1* + (*y2* + *real_of_nat* (*2*::*nat*))) ∗ ((*y1* + (*y2* − *real_of_nat* (*2*::*nat*))) ∗ ((*y2* − *real_of_nat* (*2*::*nat*) ∗ *x* + (*real_of_nat* (*2*::*nat*) − *y1*)) ∗ (*real_of_nat* (*2*::*nat*) + (*y1* + (*real_of_nat* (*2*::*nat*) ∗ *x* − *y2*))))))))) ⟶ *has_real_derivative* *g* (*?g'* *x*) (*within* (*atreal* *x*) *s*)

thm Compute_2158872499.compute_two_second:

∀ (y1::real) (y2::real) (s::real ⇒ bool) (hz::real) (g::real ⇒ real) x::?'a::type. (1::real) ≤ hz ∧ hz < real_of_nat (2::nat) ∧ real_of_nat (2::nat) ≤ y1 ∧ y1 ≤ real_of_nat (2::nat) ∗ hz ∧ real_of_nat (2::nat) ≤ y2 ∧ y2 ≤ real_of_nat (2::nat) ∗ hz ∧ s = GSPEC (λGEN%PVAR%199::real. ∃ t::real. SETSPEC GEN%PVAR%199 (y2 − real_of_nat (2::nat) − y1 < real_of_nat (2::nat) ∗ t ∧ real_of_nat (2::nat) ∗ t < y2 + (real_of_nat (2::nat) − y1)) t) ∧ g = (λt::real. (y2 − y1 − real_of_nat (2::nat) ∗ t) ∗ (($y2$ + $y1)^2$ − real_of_nat (4::nat)) / ((y1 + t) ∗ ((y2 − t) ∗ sqrt ((y1 + (y2 + real_of_nat (2::nat))) ∗ ((y1 + (y2 − real_of_nat (2::nat))) ∗ ((y2 − real_of_nat (2::nat) ∗ t + (real_of_nat (2::nat) − y1)) ∗ (real_of_nat (2::nat) + (y1 + (real_of_nat (2::nat) ∗ t − y2))))))))) ⟶ has_real_derivative g (sqrt (ups_x ($y1^2$) ($y2^2$) (real_of_nat (4::nat))) ∗ (− real_of_nat (4::nat) ∗ $y1^2$ + ($y1^{4::nat}$ − real_of_nat (4::nat) ∗ ($y1^{3::nat}$ ∗ y2) − real_of_nat (4::nat) ∗ $y2^2$ + (real_of_nat (6::nat) ∗ ($y1^2$ ∗ $y2^2$) − real_of_nat (4::nat) ∗ (y1 ∗ $y2^{3::nat}$) + $y2^{4::nat}$))) / ($y1^2$ ∗ ($y2^2$ ∗ (real_of_nat (4::nat) − $(y1 - y2)^2)^2$))) (within (atreal (0::real)) s)

thm Compute_2158872499.COMPUTE_DERIVATIVE_TWO:

∀ (y1::real) (y2::real) (s::real ⇒ bool) (hz::real) (g::real ⇒ real) (x::real) f::?'a::type. (1::real) ≤ hz ∧ hz < real_of_nat (2::nat) ∧ real_of_nat (2::nat) ≤ y1 ∧ y1 ≤ real_of_nat (2::nat) ∗ hz ∧ real_of_nat (2::nat) ≤ y2 ∧ y2 ≤ real_of_nat (2::nat) ∗ hz ∧ s = GSPEC (λGEN%PVAR%200::real. ∃ t::real. SETSPEC GEN%PVAR%200 (y2 − real_of_nat (2::nat) − y1 < real_of_nat (2::nat) ∗ t ∧ real_of_nat (2::nat) ∗ t < y2 + (real_of_nat (2::nat) − y1)) t) ∧ g = (λt::real. arclength (y1 + t) (y2 − t) (real_of_nat (2::nat))) ∧ IN x s ∧ (?g'::real ⇒ real) = (λt::real. (y2 − y1 − real_of_nat (2::nat) ∗ t) ∗ (($y2$ + $y1)^2$ − real_of_nat (4::nat)) / ((y1 + t) ∗ ((y2 − t) ∗ sqrt ((y1 + (y2 + real_of_nat (2::nat))) ∗ ((y1 + (y2 − real_of_nat (2::nat))) ∗ ((y2 − real_of_nat (2::nat) ∗ t + (real_of_nat (2::nat) − y1)) ∗ (real_of_nat (2::nat) + (y1 + (real_of_nat (2::nat) ∗ t − y2))))))))) ⟶ (∀ x::real. IN x s ⟶ has_real_derivative g (?g' x) (within (atreal x) s)) ∧ has_real_derivative ?g' (sqrt (ups_x ($y1^2$) ($y2^2$) (real_of_nat (4::nat))) ∗ (− real_of_nat (4::nat) ∗ $y1^2$ + ($y1^{4::nat}$ − real_of_nat (4::nat) ∗ ($y1^{3::nat}$ ∗ y2) − real_of_nat (4::nat) ∗ $y2^2$ + (real_of_nat (6::nat) ∗ ($y1^2$ ∗ $y2^2$) − real_of_nat (4::nat) ∗ (y1 ∗ $y2^{3::nat}$) + $y2^{4::nat}$))) / ($y1^2$ ∗ ($y2^2$ ∗ (real_of_nat (4::nat) − $(y1 - y2)^2)^2$))) (within (atreal (0::real)) s)

thm Delta_x.COMPUTE_DELTA_X:

∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. (x1, x2, x3, x4, x5, x6) = xlist v0 v1 v2 v3 ⟶ delta_x x1 x2 x3 x4 x5 x6 = LET (λa::(real, 3) cart. LET_END (LET (λb::(real, 3) cart. LET_END (LET (λc::(real, 3) cart. LET_END (real_of_nat (4::nat) ∗ ($ a (1::nat) ∗ ($ b (2::nat) ∗ $ c (3::nat)) − $ a (1::nat) ∗ ($ b (3::nat) ∗ $ c (2::nat)) − $ a (2::nat) ∗ ($ b (1::nat) ∗ $ c (3::nat)) + ($ a (2::nat) ∗ ($ b (3::nat) ∗ $ c (1::nat)) + ($ a (3::nat) ∗ ($ b (1::nat) ∗ $ c (2::nat)) − $ a (3::nat) ∗ ($ b (2::nat) ∗ $ c (1::nat)))))$)^2$)) (vector_sub v3 v0))) (vector_sub v2 v0))) (vector_sub v1 v0)

thm Delta_x.DELTA_X_DET_3:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) *A*::((*real*, *3*) *cart*, *3*) *cart*. (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*) = *xlist v0 v1 v2 v3* $\wedge$ \$ *A* (*1*::*nat*) = *vector_sub v1 v0* $\wedge$ \$ *A* (*2*::*nat*) = *vector_sub v2 v0* $\wedge$ \$ *A* (*3*::*nat*) = *vector_sub v3 v0* $\longrightarrow$ *delta_x x1 x2 x3 x4 x5 x6* = *real_of_nat* (*4*::*nat*) $*$ (*det A*)^2

thm Delta_x.DELTA_X_LT_0_COLLINEAR:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*) = *xlist v0 v1 v2 v3* $\longrightarrow$ (*0*::*real*) < *delta_x x1 x2 x3 x4 x5 x6* $\longrightarrow$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v0* (*INSERT v3* (*INSERT v2 EMPTY*)))

thm Euler_complement.LEMMA_FOR_EULER_AFTER_RESCALE:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *vector_norm* (*vector_sub v1 v0*) = (*1*::*real*) $\longrightarrow$ *vector_norm* (*vector_sub v2 v0*) = (*1*::*real*) $\longrightarrow$ *vector_norm* (*vector_sub v3 v0*) = (*1*::*real*) $\longrightarrow$ *real_of_nat* (*2*::*nat*) $*$ ((*1*::*real*) + (*dot* (*vector_sub v2 v0*) (*vector_sub v3 v0*) + (*dot* (*vector_sub v3 v0*) (*vector_sub v1 v0*) + *dot* (*vector_sub v1 v0*) (*vector_sub v2 v0*)))) = *real_of_nat* (*8*::*nat*) $-$ *dot* (*vector_sub v2 v3*) (*vector_sub v2 v3*) $-$ *dot* (*vector_sub v1 v3*) (*vector_sub v1 v3*) $-$ *dot* (*vector_sub v1 v2*) (*vector_sub v1 v2*)

thm Euler_complement.DIHV_RESCALE_UNCHANGED:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*w0*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) (*w3*::(*real*, *3*) *cart*) (*m*::*real*) (*n*::*real*) *p*::*real*. *vector_sub v1 v0* = % *m* (*vector_sub w1 w0*) $\wedge$ *vector_sub v2 v0* = % *n* (*vector_sub w2 w0*) $\wedge$ *vector_sub v3 v0* = % *p* (*vector_sub w3 w0*) $\wedge$ (*0*::*real*) < *m* $\wedge$ (*0*::*real*) < *n* $\wedge$ (*0*::*real*) < *p* $\longrightarrow$ *dihV v0 v1 v2 v3* = *dihV w0 w1 w2 w3*

thm Euler_complement.COMPUTE_EULER_P_AFTER_RESCALE:

$\forall$ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v0'*::(*real*, *3*) *cart*) (*v1'*::(*real*, *3*) *cart*) (*v2'*::(*real*, *3*) *cart*) (*v3'*::(*real*, *3*) *cart*) (*m*::*real*) (*n*::*real*) *p*::*real*. *vector_sub v1 v0* = % *m* (*vector_sub v1' v0'*) $\wedge$ *vector_sub v2 v0* = % *n* (*vector_sub v2' v0'*) $\wedge$ *vector_sub v3 v0* = % *p* (*vector_sub v3' v0'*) $\wedge$ (*0*::*real*) < *m* $\wedge$ (*0*::*real*) < *n* $\wedge$ (*0*::*real*) < *p* $\longrightarrow$ *euler_p v0 v1 v2 v3* = *m* $*$ (*n* $*$ *p*) $*$ *euler_p v0' v1' v2' v3'*

thm Euler_complement.VECTOR_EQ_COMPONENT:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *i*::*nat*. *x* = *y* $\longrightarrow$ \$ *x i* = \$ *y i*

thm Euler_complement.COMPUTE_DELTA_X_AFTER_RESCALE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v0'*::(*real*, *3*) *cart*) (*v1'*::(*real*, *3*) *cart*) (*v2'*::(*real*, *3*) *cart*) (*v3'*::(*real*, *3*) *cart*) (*m*::*real*) (*n*::*real*) (*p*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x1'*::*real*) (*x2'*::*real*) (*x3'*::*real*) (*x4'*::*real*) (*x5'*::*real*) *x6'*::*real*. *vector_sub v1 v0* = % *m* (*vector_sub v1' v0'*) ∧ *vector_sub v2 v0* = % *n* (*vector_sub v2' v0'*) ∧ *vector_sub v3 v0* = % *p* (*vector_sub v3' v0'*) ∧ (*0*::*real*) ≤ *m* ∧ (*0*::*real*) ≤ *n* ∧ (*0*::*real*) ≤ *p* ∧ (*x1'*, *x2'*, *x3'*, *x4'*, *x5'*, *x6'*) = *xlist v0' v1' v2' v3'* ∧ (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*) = *xlist v0 v1 v2 v3* ∧ (*0*::*real*) ≤ *delta_x x1' x2' x3' x4' x5' x6'* ⟶ *delta_x x1 x2 x3 x4 x5 x6* = *m* ∗ (*n* ∗ *p*) ∗ (*m* ∗ (*n* ∗ *p*) ∗ *delta_x x1' x2' x3' x4' x5' x6'*)

thm Euler_complement.SQRT_DELTA_X_AFTER_RESCALE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*v0'*::(*real*, *3*) *cart*) (*v1'*::(*real*, *3*) *cart*) (*v2'*::(*real*, *3*) *cart*) (*v3'*::(*real*, *3*) *cart*) (*m*::*real*) (*n*::*real*) (*p*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x1'*::*real*) (*x2'*::*real*) (*x3'*::*real*) (*x4'*::*real*) (*x5'*::*real*) *x6'*::*real*. *vector_sub v1 v0* = % *m* (*vector_sub v1' v0'*) ∧ *vector_sub v2 v0* = % *n* (*vector_sub v2' v0'*) ∧ *vector_sub v3 v0* = % *p* (*vector_sub v3' v0'*) ∧ (*0*::*real*) ≤ *m* ∧ (*0*::*real*) ≤ *n* ∧ (*0*::*real*) ≤ *p* ∧ (*x1'*, *x2'*, *x3'*, *x4'*, *x5'*, *x6'*) = *xlist v0' v1' v2' v3'* ∧ (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*) = *xlist v0 v1 v2 v3* ∧ (*0*::*real*) ≤ *delta_x x1' x2' x3' x4' x5' x6'* ⟶ *sqrt* (*delta_x x1 x2 x3 x4 x5 x6*) = *m* ∗ (*n* ∗ *p*) ∗ *sqrt* (*delta_x x1' x2' x3' x4' x5' x6'*)

thm Euler_complement.EULER_FORMULA_RESCALE:

(∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*p*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*alpha1*::*real*) (*alpha2*::*real*) (*alpha3*::*real*) (*d*::*real*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) *w3*::(*real*, *3*) *cart*. *p* = *euler_p v0 v1 v2 v3* ∧ (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*) = *xlist v0 v1 v2 v3* ∧ *alpha1* = *dihV v0 v1 v2 v3* ∧ *alpha2* = *dihV v0 v2 v3 v1* ∧ *alpha3* = *dihV v0 v3 v1 v2* ∧ *d* = *delta_x x1 x2 x3 x4 x5 x6* ∧ *w1* = *vector_sub v1 v0* ∧ *w2* = *vector_sub v2 v0* ∧ *w3* = *vector_sub v3 v0* ∧ (*0*::*real*) < *d* ∧ *vector_norm w1* = (*1*::*real*) ∧ *vector_norm w2* = (*1*::*real*) ∧ *vector_norm w3* = (*1*::*real*) ⟶ *alpha1* + (*alpha2* + (*alpha3* − *pi*)) = *pi* − *real_of_nat* (*2*::*nat*) ∗ *atn2* (*sqrt d*, *real_of_nat* (*2*::*nat*) ∗ *p*)) ⟶ (∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *v3*::(*real*, *3*) *cart*. *LET* (λ*p*::*real*. *LET_END* (*LET* (*GABS* (λ*f*::*real* × *real* × *real* × *real* × *real* × *real* ⇒ *bool*. ∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *GEQ* (*f* (*x1*, *x2*, *x3*, *x4*, *x5*, *x6*)) (*LET_END* (*LET* (λ*alpha1*::*real*. *LET_END* (*LET* (λ*alpha2*::*real*. *LET_END* (*LET* (λ*alpha3*::*real*. *LET_END* (*LET* (λ*d*::*real*. *LET_END* ((*0*::*real*) < *d* ⟶ *alpha1* + (*alpha2* + (*alpha3* − *pi*)) = *pi* − *real_of_nat* (*2*::*nat*) ∗ *atn2* (*sqrt d*, *real_of_nat* (*2*::*nat*) ∗ *p*))) (*delta_x x1 x2 x3 x4 x5 x6*))) (*dihV v0 v3 v1 v2*))) (*dihV v0 v2 v3 v1*))) (*dihV v0 v1 v2 v3*))))) (*xlist v0 v1 v2 v3*))) (*euler_p v0 v1 v2 v3*))

thm Euler_complement.COLLINEAR_NORM_LT_0:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) *c*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT a*

$(INSERT\ b\ (INSERT\ c\ EMPTY))) \longrightarrow (0::real) < vector\_norm\ (vector\_sub\ a\ b)$

thm Euler_complement.REAL_LT_RSQRT2:

$\forall (x::real)\ y::real.\ x^2 < y \longrightarrow -\ sqrt\ y < x$

thm Euler_complement.EULER_TRIANGLE_REAL_INTERVAL:

$\forall (s::real \Rightarrow bool)\ (a::real)\ (b::real)\ c::real.\ s = GSPEC\ (\lambda GEN\%PVAR\%201::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%201\ ((0::real) < ups\_x\ x\ b\ c - x * (b * c))\ x) \wedge (0::real) < ups\_x\ a\ b\ c - a * (b * c) \longrightarrow is\_realinterval\ s$

thm Euler_complement.OJEKOJF2:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ LET\ (\lambda ga::real.\ LET\_END\ (LET\ (\lambda v01::real.\ LET\_END\ (LET\ (\lambda v02::real.\ LET\_END\ (LET\ (\lambda v03::real.\ LET\_END\ (LET\ (\lambda v12::real.\ LET\_END\ (LET\ (\lambda v13::real.\ LET\_END\ (LET\ (\lambda v23::real.\ LET\_END\ (\neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \longrightarrow ga = pi\ /\ real\_of\_nat\ (2::nat) - atn2\ (sqrt\ (real\_of\_nat\ (4::nat) * (v01 * delta\_x\ v01\ v02\ v03\ v23\ v13\ v12)),\ delta\_x4\ v01\ v02\ v03\ v23\ v13\ v12)))\ ((distance\ (v2,\ v3))^2)))\ ((distance\ (v1,\ v3))^2)))\ ((distance\ (v1,\ v2))^2)))\ ((distance\ (v0,\ v3))^2)))\ ((distance\ (v0,\ v2))^2)))\ ((distance\ (v0,\ v1))^2)))\ (dihV\ v0\ v1\ v2\ v3)$

thm Euler_complement.COMPUTE_DIHV_ATN2:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (gamma::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma = dihV\ v0\ v1\ v2\ v3 \wedge x1 = (distance\ (v0,\ v1))^2 \wedge x2 = (distance\ (v0,\ v2))^2 \wedge x3 = (distance\ (v0,\ v3))^2 \wedge x6 = (distance\ (v1,\ v2))^2 \wedge x5 = (distance\ (v1,\ v3))^2 \wedge x4 = (distance\ (v2,\ v3))^2 \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \longrightarrow gamma = pi\ /\ real\_of\_nat\ (2::nat) - atn2\ (sqrt\ (real\_of\_nat\ (4::nat) * (x1 * delta\_x\ x1\ x2\ x3\ x4\ x5\ x6)),\ delta\_x4\ x1\ x2\ x3\ x4\ x5\ x6)$

thm Euler_multivariate.HAS_REAL_DERIVATIVE_ZERO_CONSTANT2:

$\forall (f::real \Rightarrow real)\ (a::real)\ (b::?'a::type)\ (c::real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge IN\ a\ s \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f\ (0::real)\ (within\ (atreal\ x)\ s)) \wedge f\ a = c \longrightarrow (\forall x::real.\ IN\ x\ s \longrightarrow f\ x = c)$

thm Euler_multivariate.INTERVAL_DIVIDE_Euler_lemma:

$\forall c::real.\ (0::real) < (real\_of\_nat\ (4::nat) - c) * c \longrightarrow (0::real) < c \wedge c < real\_of\_nat\ (4::nat)$

thm Euler_multivariate.SQRT_RULE_Euler_lemma:

$\forall (x::real)\ y::real.\ x^2 = y \wedge (0::real) \leq x \longrightarrow x = sqrt\ y$

thm Euler_multivariate.REAL_INTERVAL_Euler_lemma:

$\forall (a::real)\ b::real.\ LET\ (\lambda P1::real \Rightarrow bool.\ LET\_END\ (LET\ (\lambda P2::real \Rightarrow bool.\ LET\_END\ (LET\ (\lambda P3::real \Rightarrow bool.\ LET\_END\ (is\_realinterval\ P1 \wedge is\_realinterval\ P2 \wedge is\_realinterval\ P3))\ (GSPEC\ (\lambda GEN\%PVAR\%204::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%204\ (a < x \wedge x < b)\ x))))\ (GSPEC\ (\lambda GEN\%PVAR\%203::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%203\ (a < x)\ x))))\ (GSPEC\ (\lambda GEN\%PVAR\%202::real.\ \exists x::real.\ SETSPEC\ GEN\%PVAR\%202\ (x < a)\ x))$

thm Euler_multivariate.DERIVATIVE_WRT_C1_Euler_lemma:

$\forall P::real \Rightarrow bool.\ is\_realinterval\ P \wedge IN\ (real\_of\_nat\ (2::nat))\ P \wedge (\forall x::real.\ IN\ x\ P \longrightarrow x \neq (0::real) \wedge real\_of\_nat\ (4::nat) - x \neq (0::real)) \longrightarrow (\forall c::real.\ IN\ c\ P \longrightarrow -\ pi\ /\ real\_of\_nat\ (2::nat) - real\_of\_nat\ (2::nat) * atn\ ((1::real) - c\ /\ real\_of\_nat\ (2::nat)) + real\_of\_nat\ (2::nat) * atn\ ((real\_of\_nat\ (4::nat) - c)\ /\ c) = (0::real))$

thm Euler_multivariate.DERIVATIVE_WRT_A_Euler_lemma:

$\forall (a::real)\ (b::real)\ c::real.\ LET\ (\lambda d::real.\ LET\_END\ ((0::real) < d \wedge ((0::real) < a \wedge (0::real) < b \wedge (0::real) < c) \wedge a < real\_of\_nat\ (4::nat) \wedge b < real\_of\_nat\ (4::nat) \wedge c < real\_of\_nat\ (4::nat) \longrightarrow pi\ /\ real\_of\_nat\ (2::nat) - atn\ ((-\ real\_of\_nat\ (2::nat) * a + (real\_of\_nat\ (2::nat) * b + (real\_of\_nat\ (2::nat) * c - b * c)))\ /\ (real\_of\_nat\ (2::nat) * sqrt\ d)) + (pi\ /\ real\_of\_nat\ (2::nat) - atn\ ((-\ real\_of\_nat\ (2::nat) * b + (real\_of\_nat\ (2::nat) * c + (real\_of\_nat\ (2::nat) * a - c * a)))\ /\ (real\_of\_nat\ (2::nat) * sqrt\ d)) + (pi\ /\ real\_of\_nat\ (2::nat) - atn\ ((-\ real\_of\_nat\ (2::nat) * c + (real\_of\_nat\ (2::nat) * a + (real\_of\_nat\ (2::nat) * b - a * b)))\ /\ (real\_of\_nat\ (2::nat) * sqrt\ d)) - pi)) = pi - real\_of\_nat\ (2::nat) * atn\ ((real\_of\_nat\ (8::nat) - a - b - c)\ /\ sqrt\ d)))\ (ups\_x\ a\ b\ c - a * (b * c))$

thm Euler_main_theorem.EULER_ANGLE_SUM_rescal:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ (p::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ (x6::real)\ (alpha1::real)\ (alpha2::real)\ (alpha3::real)\ (d::real)\ (w1::(real,\ 3)\ cart)\ (w2::(real,\ 3)\ cart)\ w3::(real,\ 3)\ cart.\ p = euler\_p\ v0\ v1\ v2\ v3 \wedge (x1,\ x2,\ x3,\ x4,\ x5,\ x6) = xlist\ v0\ v1\ v2\ v3 \wedge alpha1 = dihV\ v0\ v1\ v2\ v3 \wedge alpha2 = dihV\ v0\ v2\ v3\ v1 \wedge alpha3 = dihV\ v0\ v3\ v1\ v2 \wedge d = delta\_x\ x1\ x2\ x3\ x4\ x5\ x6 \wedge w1 = vector\_sub\ v1\ v0 \wedge w2 = vector\_sub\ v2\ v0 \wedge w3 = vector\_sub\ v3\ v0 \wedge (0::real) < d \wedge vector\_norm\ w1 = (1::real) \wedge vector\_norm\ w2 = (1::real) \wedge vector\_norm\ w3 = (1::real) \longrightarrow alpha1 + (alpha2 + (alpha3 - pi)) = pi - real\_of\_nat\ (2::nat) * atn2\ (sqrt\ d,\ real\_of\_nat\ (2::nat) * p)$

thm Euler_main_theorem.EULER_TRIANGLE:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ LET\ (\lambda p::real.\ LET\_END\ (LET\ (GABS\ (\lambda f::real \times real \times real \times real \times real \times real \Rightarrow bool.\ \forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ GEQ\ (f\ (x1,\ x2,\ x3,\ x4,\ x5,\ x6))\ (LET\_END\ (LET\ (\lambda alpha1::real.\ LET\_END\ (LET\ (\lambda alpha2::real.\ LET\_END\ (LET\ (\lambda alpha3::real.\ LET\_END\ (LET\ (\lambda d::real.\ LET\_END\ ((0::real) < d \longrightarrow alpha1 + (alpha2 + (alpha3 - pi)) = pi -$

$real\_of\_nat\ (2::nat) * atn2\ (sqrt\ d,\ real\_of\_nat\ (2::nat) * p))) (delta\_x\ x1\ x2\ x3\ x4\ x5\ x6))) (dihV\ v0\ v3\ v1\ v2))) (dihV\ v0\ v2\ v3\ v1))) (dihV\ v0\ v1\ v2\ v3))))) (xlist\ v0\ v1\ v2\ v3))) (euler\_p\ v0\ v1\ v2\ v3)$

thm DEF_unknown:

$unknown = True$

thm Trigonometry.ULEKUUB:

$unknown$

thm Trigonometry.LLOYXRK1:

$\forall x::real.\ cos\ (pi\ /\ real\_of\_nat\ (2::nat) + x) = -\ sin\ x$

thm Trigonometry.LLOYXRK2:

$\forall x::real.\ sin\ (pi\ /\ real\_of\_nat\ (2::nat) + x) = cos\ x$

thm Trigonometry.UIVNNRR1:

$\forall V::(real,\ ?'a::type)\ cart \Rightarrow bool.\ aff\_ge\ V\ EMPTY = aff\_gt\ V\ EMPTY$

thm DEF_parallel:

$parallel = (SOME\ parallel::nat \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall (\_2354585::nat)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ parallel\ \_2354585\ v\ w = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY))))\ (79::nat)$

thm Trigonometry.SWKFLBJ3:

$parallel\ (?v::(real,\ ?'a::type)\ cart)\ (?w::(real,\ ?'a::type)\ cart) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?v\ (INSERT\ ?w\ EMPTY)))$

thm Hvihvec.VECTOR_ANGLE_DOUBLECROSS:

$\forall (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ w \neq vec\ (0::nat) \wedge dot\ u\ w = (0::real) \wedge dot\ v\ w = (0::real) \longrightarrow vector\_angle\ (cross\ u\ w)\ (cross\ v\ w) = vector\_angle\ u\ v$

thm Hvihvec.HVIHVEC:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ v0 \neq v1 \longrightarrow dihV\ v0\ v1\ v2\ v3 = LET\ (\lambda(w1::(real,\ 3)\ cart)\ (w2::(real,\ 3)\ cart)\ w3::(real,\ 3)\ cart.\ LET\_END\ (vector\_angle\ (cross\ w1\ w2)\ (cross\ w1\ w3)))\ (vector\_sub\ v1\ v0)\ (vector\_sub\ v2\ v0)\ (vector\_sub\ v3\ v0)$

thm DEF_ratform:

$ratform = (\lambda(\_2354605::bool)\ (\_2354606::real)\ (\_2354607::real)\ \_2354608::real.\ \_2354605 \longrightarrow \_2354608 \neq (0::real) \wedge \_2354606 = \_2354607\ /\ \_2354608)$

thm Calc_derivative.ratform:

$\forall (p::bool)\ (r::real)\ (a::real)\ b::real.\ ratform\ p\ r\ a\ b = (p \longrightarrow b \neq (0::real) \wedge r = a\ /\ b)$

thm Calc_derivative.REAL_POW_NEQ_0:

$\forall (x{::}real)\ n{::}nat.\ (x^n \neq (0{::}real)) = (x \neq (0{::}real) \lor n = (0{::}nat))$

thm Calc_derivative.ratform_pow:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \longrightarrow ratform\ ?p1.0\ ?r1.0^{?n{::}nat}\ ?a1.0^{?n}\ ?b1.0^{?n}$

thm Calc_derivative.ratform_add:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \land ratform\ (?p2.0{::}bool)\ (?r2.0{::}real)\ (?a2.0{::}real)\ (?b2.0{::}real) \longrightarrow ratform\ (?p1.0 \land ?p2.0)\ (?r1.0 + ?r2.0)\ (?a1.0 * ?b2.0 + ?b1.0 * ?a2.0)\ (?b1.0 * ?b2.0)$

thm Calc_derivative.ratform_sub:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \land ratform\ (?p2.0{::}bool)\ (?r2.0{::}real)\ (?a2.0{::}real)\ (?b2.0{::}real) \longrightarrow ratform\ (?p1.0 \land ?p2.0)\ (?r1.0 - ?r2.0)\ (?a1.0 * ?b2.0 - ?b1.0 * ?a2.0)\ (?b1.0 * ?b2.0)$

thm Calc_derivative.ratform_neg:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \longrightarrow ratform\ ?p1.0\ (-\ ?r1.0)\ (-\ ?a1.0)\ ?b1.0$

thm Calc_derivative.ratform_mul:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \land ratform\ (?p2.0{::}bool)\ (?r2.0{::}real)\ (?a2.0{::}real)\ (?b2.0{::}real) \longrightarrow ratform\ (?p1.0 \land ?p2.0)\ (?r1.0 * ?r2.0)\ (?a1.0 * ?a2.0)\ (?b1.0 * ?b2.0)$

thm Calc_derivative.ratform_div:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \land ratform\ (?p2.0{::}bool)\ (?r2.0{::}real)\ (?a2.0{::}real)\ (?b2.0{::}real) \longrightarrow ratform\ (?p1.0 \land ?p2.0 \land ?a2.0 \neq (0{::}real))\ (?r1.0\ /\ ?r2.0)\ (?a1.0 * ?b2.0)\ (?b1.0 * ?a2.0)$

thm Calc_derivative.ratform_inv:

$ratform\ (?p1.0{::}bool)\ (?r1.0{::}real)\ (?a1.0{::}real)\ (?b1.0{::}real) \longrightarrow ratform\ (?p1.0 \land ?a1.0 \neq (0{::}real))\ (inverse\_class.inverse\ ?r1.0)\ ?b1.0\ ?a1.0$

thm Calc_derivative.trivial_ratform:

$\forall t{::}real.\ ratform\ True\ t\ t\ (1{::}real)$

thm Calc_derivative.lite_imp:

$ratform\ (?p{::}bool)\ ((?u{::}real) - (?v{::}real))\ (?a{::}real)\ (?b{::}real) \land ?p = (?p'{::}bool) \land ?a = (0{::}real) \longrightarrow ?p' \longrightarrow ?u = ?v$

thm Calc_derivative.lite_imp2:

$ratform\ (?p{::}bool)\ ((?u{::}real) - (?v{::}real))\ (?a{::}real)\ (?b{::}real) \land ?p = (?p'{::}bool) \land ?a = (?a'{::}real) \longrightarrow ?p' \land ?a' = (0{::}real) \longrightarrow ?u = ?v$

thm Calc_derivative.invert_den_lt:

$\forall (a::real)\ b::real.\ ((0::real) < a\ /\ b) = ((0::real) < a * b)$

thm Calc_derivative.invert_den_le:

$\forall (a::real)\ b::real.\ ((0::real) \leq a\ /\ b) = ((0::real) \leq a * b)$

thm Calc_derivative.invert_den_eq:

$\forall (a::real)\ b::real.\ (a\ /\ b = (0::real)) = (a * b = (0::real))$

thm Calc_derivative.imp_lt:

$\forall (p::bool)\ (p'::bool)\ (x::real)\ (a::real)\ (a'::real)\ (b::real)\ b'::real.\ (0::real) < x \wedge ratform\ p\ x\ a\ b \wedge p = p' \wedge a = a' \wedge b = b' \longrightarrow p' \longrightarrow (0::real) < a' * b'$

thm Calc_derivative.imp_le:

$\forall (p::bool)\ (p'::bool)\ (x::real)\ (a::real)\ (a'::real)\ (b::real)\ b'::real.\ (0::real) \leq x \wedge ratform\ p\ x\ a\ b \wedge p = p' \wedge a = a' \wedge b = b' \longrightarrow p' \longrightarrow (0::real) \leq a' * b'$

thm Calc_derivative.imp_eq:

$\forall (p::bool)\ (p'::bool)\ (x::real)\ (a::real)\ (a'::real)\ (b::real)\ b'::real.\ x = (0::real) \wedge ratform\ p\ x\ a\ b \wedge p = p' \wedge a = a' \wedge b = b' \longrightarrow p' \wedge b' \neq (0::real) \longrightarrow a' = (0::real)$

thm Calc_derivative.imp_nz:

$\forall (p::bool)\ (p'::bool)\ (x::real)\ (a::real)\ (a'::real)\ (b::real)\ b'::real.\ x \neq (0::real) \wedge ratform\ p\ x\ a\ b \wedge p = p' \wedge a = a' \wedge b = b' \longrightarrow p' \longrightarrow a' \neq (0::real) \wedge b' \neq (0::real)$

thm DEF_derived_form:

$derived\_form = (\lambda(\_2355499::bool)\ (\_2355500::real \Rightarrow real)\ (\_2355501::real)\ (\_2355502::real)\ \_2355503::real \Rightarrow bool.\ \_2355499 \longrightarrow has\_real\_derivative\ \_2355500\ \_2355501\ (within\ (atreal\ \_2355502)\ \_2355503))$

thm Calc_derivative.derived_form:

$\forall (p::bool)\ (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ derived\_form\ p\ f\ f'\ x\ s = (p \longrightarrow has\_real\_derivative\ f\ f'\ (within\ (atreal\ x)\ s))$

thm Calc_derivative.derived_form_add:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p1.0::bool)\ (?f1.0::real \Rightarrow real)\ (?f1'::real)\ x\ s \wedge derived\_form\ (?p2.0::bool)\ (?f2.0::real \Rightarrow real)\ (?f2'::real)\ x\ s \longrightarrow derived\_form\ (?p1.0 \wedge ?p2.0)\ (\lambda x::real.\ ?f1.0\ x + ?f2.0\ x)\ (?f1' + ?f2')\ x\ s$

thm Calc_derivative.derived_form_sub:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p1.0::bool)\ (?f1.0::real \Rightarrow real)\ (?f1'::real)\ x\ s \wedge derived\_form\ (?p2.0::bool)\ (?f2.0::real \Rightarrow real)\ (?f2'::real)\ x\ s \longrightarrow derived\_form\ (?p1.0 \wedge ?p2.0)\ (\lambda x::real.\ ?f1.0\ x - ?f2.0\ x)\ (?f1' - ?f2')\ x\ s$

thm Calc_derivative.derived_form_mul:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p1.0::bool)\ (?f1.0::real \Rightarrow real)\ (?f1'::real)\ x\ s \wedge derived\_form\ (?p2.0::bool)\ (?f2.0::real \Rightarrow real)\ (?f2'::real)\ x\ s \longrightarrow derived\_form\ (?p1.0 \wedge ?p2.0)\ (\lambda x::real.\ ?f1.0\ x * ?f2.0\ x)\ (?f1.0\ x * ?f2' + ?f1' * ?f2.0\ x)\ x\ s$

thm Calc_derivative.derived_form_div:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p1.0::bool)\ (?f1.0::real \Rightarrow real)\ (?f1'::real)\ x\ s \wedge derived\_form\ (?p2.0::bool)\ (?f2.0::real \Rightarrow real)\ (?f2'::real)\ x\ s \longrightarrow derived\_form\ (?p1.0 \wedge ?p2.0 \wedge ?f2.0\ x \neq (0::real))\ (\lambda x::real.\ ?f1.0\ x\ /\ ?f2.0\ x)\ ((?f1' * ?f2.0\ x - ?f1.0\ x * ?f2')\ /\ (?f2.0\ x)^2)\ x\ s$

thm Calc_derivative.derived_form_pow:

$\forall (n::nat)\ (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p::bool)\ (?f::real \Rightarrow real)\ (?f'::real)\ x\ s \longrightarrow derived\_form\ ?p\ (\lambda x::real.\ (?f\ x)^n)\ (real\_of\_nat\ n * ((?f\ x)^{n - (1::nat)} * ?f'))\ x\ s$

thm Calc_derivative.derived_form_neg:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ uminus\ (-\ (1::real))\ x\ s$

thm Calc_derivative.derived_form_const:

$\forall (c::real)\ (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ (\lambda x::real.\ c)\ (0::real)\ x\ s$

thm Calc_derivative.derived_form_sin:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ sin\ (cos\ x)\ x\ s$

thm Calc_derivative.derived_form_cos:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ cos\ (-\ sin\ x)\ x\ s$

thm Calc_derivative.derived_form_sqrt:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ ((0::real) < x)\ sqrt\ (inverse\_class.inverse\ (real\_of\_nat\ (2::nat) * sqrt\ x))\ x\ s$

thm Calc_derivative.derived_form_atn:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ atn\ (inverse\_class.inverse\ ((1::real) + x^2))\ x\ s$

thm Calc_derivative.derived_form_acs:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (|x| < (1::real))\ acs\ (-\ inverse\_class.inverse\ (sqrt\ ((1::real) - x^2)))\ x\ s$

thm Calc_derivative.derived_form_asn:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (|x| < (1::real))\ asn\ (inverse\_class.inverse\ (sqrt\ ((1::real) - x^2)))\ x\ s$

thm Calc_derivative.derived_form_inv:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (x \neq (0::real))\ inverse\_class.inverse\ (-\ inverse\_class.inverse\ (x^2))\ x\ s$

thm Calc_derivative.derived_form_id:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ (\lambda x::real.\ x)\ (1::real)\ x\ s$

thm Calc_derivative.derived_form_chain:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (?p::bool)\ (?g::real \Rightarrow real)\ (?g'::real)\ ((?f1.0::real \Rightarrow real)\ x)\ HOL\_Light\_Import.UNIV \wedge derived\_form\ (?p'::bool)\ (?f2.0::real \Rightarrow real)\ (?f'::real)\ x\ s \wedge ?f1.0 = ?f2.0 \longrightarrow derived\_form\ (?p \wedge ?p')\ (\lambda x::real.\ ?g\ (?f1.0\ x))\ (?f' * ?g')\ x\ s$

thm Calc_derivative.derived_form_generic:

$\forall (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ derived\_form\ (derived\_form\ True\ f\ f'\ x\ s)\ f\ f'\ x\ s$

thm Calc_derivative.region_conv:

$\forall (x::real)\ y::real.\ \neg\ (y = (0::real) \wedge x \leq (0::real)) \longrightarrow (0::real) < x \vee y < (0::real) \vee (0::real) < y$

thm Calc_derivative.x2notless0:

$\forall x::real.\ \neg\ x^2 < (0::real)$

thm Calc_derivative.x2notlesseq0:

$\forall x::real.\ x \neq (0::real) \longrightarrow \neg\ x^2 \leq (0::real)$

thm Calc_derivative.sumsquaresnot0:

$\forall (x::real)\ y::real.\ x \neq (0::real) \longrightarrow \neg\ x^2 + y^2 \leq (0::real)$

thm Calc_derivative.notzerodenom:

$\forall (a::real)\ (b::real)\ (c::real)\ d::real.\ b \neq (0::real) \longrightarrow (a * b - c * d) / b^2 * inverse\_class.inverse\ ((1::real) + (c / b)^2) = (a * b - c * d) / (b^2 + c^2)$

thm Calc_derivative.notzerodenom2:

$\forall (a::real)\ (b::real)\ (c::real)\ d::real.\ c \neq (0::real) \longrightarrow (a * b - c * d) / c^2 * inverse\_class.inverse\ ((1::real) + (b / c)^2) = (a * b - c * d) / (b^2 + c^2)$

thm Calc_derivative.derived_imp_pos_open:

$\forall (p::bool)\ (f::real \Rightarrow real)\ (f'::real)\ (x::real)\ s::real \Rightarrow bool.\ p \wedge derived\_form\ p\ f\ f'\ x\ s \wedge (0::real) < f\ x \longrightarrow (\exists d>0::real.\ \forall x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow (0::real) < f\ x')$

thm Calc_derivative.derived_imp_pos_open_2:

$\forall (p::bool)\ (g::real \Rightarrow real)\ (g'::real)\ (x::real)\ s::real \Rightarrow bool.\ p \wedge derived\_form\ p\ g\ g'\ x\ s \wedge (0::real) < g\ x \longrightarrow (\exists d>0::real.\ \forall x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow (0::real) < g\ x')$

thm Calc_derivative.derived_imp_pos_open_3:

$\forall (p::bool)\ (g::real \Rightarrow real)\ (g'::real)\ (x::real)\ s::real \Rightarrow bool.\ p \wedge derived\_form\ p\ g\ g'\ x\ s \wedge g\ x < (0::real) \longrightarrow (\exists\, d>0::real.\ \forall\, x'::real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow g\ x' < (0::real))$

thm Calc_derivative.deriv_pi:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ (\lambda x::real.\ pi\ /\ real\_of\_nat\ (2::nat))\ (0::real)\ x\ s$

thm Calc_derivative.deriv_pi_minus:

$\forall (x::real)\ (pf::bool)\ (f::real \Rightarrow real)\ (f'::real)\ s::real \Rightarrow bool.\ derived\_form\ pf\ f\ f'\ x\ s \longrightarrow derived\_form\ (True \wedge pf)\ (\lambda x::real.\ pi\ /\ real\_of\_nat\ (2::nat) - f\ x)\ ((0::real) - f')\ x\ s$

thm Calc_derivative.deriv_minus_pi:

$\forall (x::real)\ s::real \Rightarrow bool.\ derived\_form\ True\ (\lambda x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)))\ (0::real)\ x\ s$

thm Calc_derivative.deriv_minus_pi_minus:

$\forall (x::real)\ (pf::bool)\ (f::real \Rightarrow real)\ (f'::real)\ s::real \Rightarrow bool.\ derived\_form\ pf\ f\ f'\ x\ s \longrightarrow derived\_form\ (True \wedge pf)\ (\lambda x::real.\ -\ (pi\ /\ real\_of\_nat\ (2::nat)) - f\ x)\ ((0::real) - f')\ x\ s$

thm Calc_derivative.derived_form_chain_simple:

$\forall (x::real)\ (s::real \Rightarrow bool)\ (g::real \Rightarrow real)\ (g'::real)\ (f::real \Rightarrow real)\ (f'::real)\ (p::bool)\ p'::bool.\ derived\_form\ p\ g\ g'\ (f\ x)\ HOL\_Light\_Import.UNIV \wedge derived\_form\ p'\ f\ f'\ x\ s \longrightarrow derived\_form\ (p \wedge p')\ (\lambda x::real.\ g\ (f\ x))\ (f' * g')\ x\ s$

thm Calc_derivative.atn_lemma:

$\forall (x::real)\ (pf::bool)\ (f::real \Rightarrow real)\ (f'::real)\ (pg::bool)\ (g::real \Rightarrow real)\ (g'::real)\ s::real \Rightarrow bool.\ (0::real) < f\ x \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (True \wedge pg \wedge pf)\ (\lambda x::real.\ atn\ (g\ x\ /\ f\ x))\ ((g' * f\ x - g\ x * f')\ /\ (f\ x)^2 * inverse\_class.inverse\ ((1::real) + (g\ x\ /\ f\ x)^2))\ x\ s$

thm Calc_derivative.atn_notpi_lemma:

$\forall (x::real)\ (pf::bool)\ (f::real \Rightarrow real)\ (f'::real)\ (pg::bool)\ (g::real \Rightarrow real)\ (g'::real)\ s::real \Rightarrow bool.\ (0::real) < g\ x \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (True \wedge pf \wedge pg)\ (\lambda x::real.\ atn\ (f\ x\ /\ g\ x))\ ((f' * g\ x - f\ x * g')\ /\ (g\ x)^2 * inverse\_class.inverse\ ((1::real) + (f\ x\ /\ g\ x)^2))\ x\ s$

thm Calc_derivative.atn_notnegpi_lemma:

$\forall (x::real)\ (pf::bool)\ (f::real \Rightarrow real)\ (f'::real)\ (pg::bool)\ (g::real \Rightarrow real)\ (g'::real)\ s::real \Rightarrow bool.\ g\ x < (0::real) \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (True \wedge pf \wedge pg)\ (\lambda x::real.\ atn\ (f\ x\ /\ g\ x))\ ((f' * g\ x - f\ x * g')\ /\ (g\ x)^2 * inverse\_class.inverse\ ((1::real) + (f\ x\ /\ g\ x)^2))\ x\ s$

thm Calc_derivative.atn_lemma_2:

$\forall\,(x{::}real)\ (pf{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (pg{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ s{::}real \Rightarrow bool.\ (0{::}real) < g\ x \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (True \wedge pf \wedge pg)\ (\lambda x{::}real.\ pi\ /\ real\_of\_nat\ (2{::}nat) - atn\ (f\ x\ /\ g\ x))\ ((g' * f\ x - g\ x * f')\ /\ (g\ x)^2 * inverse\_class.inverse\ ((1{::}real) + (f\ x\ /\ g\ x)^2))\ x\ s$

thm Calc_derivative.atn_lemma_3:

$\forall\,(x{::}real)\ (pf{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (pg{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ s{::}real \Rightarrow bool.\ g\ x < (0{::}real) \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (True \wedge pf \wedge pg)\ (\lambda x{::}real.\ -\ (pi\ /\ real\_of\_nat\ (2{::}nat)) - atn\ (f\ x\ /\ g\ x))\ ((g' * f\ x - g\ x * f')\ /\ (g\ x)^2 * inverse\_class.inverse\ ((1{::}real) + (f\ x\ /\ g\ x)^2))\ x\ s$

thm Calc_derivative.atn2_atn_open:

$\forall\,(p{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (g{::}real \Rightarrow real)\ (x{::}real)\ s{::}real \Rightarrow bool.\ p \wedge derived\_form\ p\ f\ f'\ x\ s \wedge (0{::}real) < f\ x \longrightarrow (\exists\, d{>}0{::}real.\ \forall\, x'{::}real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow atn2\ (f\ x',\ g\ x') = atn\ (g\ x'\ /\ f\ x'))$

thm Calc_derivative.atn2_atn_open_2:

$\forall\,(p{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ (f{::}real \Rightarrow real)\ (x{::}real)\ s{::}real \Rightarrow bool.\ p \wedge derived\_form\ p\ g\ g'\ x\ s \wedge (0{::}real) < g\ x \longrightarrow (\exists\, d{>}0{::}real.\ \forall\, x'{::}real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow atn2\ (f\ x',\ g\ x') = pi\ /\ real\_of\_nat\ (2{::}nat) - atn\ (f\ x'\ /\ g\ x'))$

thm Calc_derivative.atn2_atn_open_3:

$\forall\,(p{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ (f{::}real \Rightarrow real)\ (x{::}real)\ s{::}real \Rightarrow bool.\ p \wedge derived\_form\ p\ g\ g'\ x\ s \wedge g\ x < (0{::}real) \longrightarrow (\exists\, d{>}0{::}real.\ \forall\, x'{::}real.\ IN\ x'\ s \wedge |x' - x| < d \longrightarrow atn2\ (f\ x',\ g\ x') = -\ (pi\ /\ real\_of\_nat\ (2{::}nat)) - atn\ (f\ x'\ /\ g\ x'))$

thm Calc_derivative.atn2_final_1:

$\forall\,(x{::}real)\ (pf{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (pg{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ s{::}real \Rightarrow bool.\ (0{::}real) < f\ x \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (IN\ x\ s \wedge pg \wedge pf)\ (\lambda x{::}real.\ atn2\ (f\ x,\ g\ x))\ ((g' * f\ x - g\ x * f')\ /\ (f\ x)^2 * inverse\_class.inverse\ ((1{::}real) + (g\ x\ /\ f\ x)^2))\ x\ s$

thm Calc_derivative.atn2_final_2:

$\forall\,(x{::}real)\ (pf{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (pg{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ s{::}real \Rightarrow bool.\ (0{::}real) < g\ x \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g\ g'\ x\ s \longrightarrow derived\_form\ (IN\ x\ s \wedge pg \wedge pf)\ (\lambda x{::}real.\ atn2\ (f\ x,\ g\ x))\ ((g' * f\ x - g\ x * f')\ /\ (g\ x)^2 * inverse\_class.inverse\ ((1{::}real) + (f\ x\ /\ g\ x)^2))\ x\ s$

thm Calc_derivative.atn2_final_3:

$\forall\,(x{::}real)\ (pf{::}bool)\ (f{::}real \Rightarrow real)\ (f'{::}real)\ (pg{::}bool)\ (g{::}real \Rightarrow real)\ (g'{::}real)\ s{::}real \Rightarrow bool.\ g\ x < (0{::}real) \wedge derived\_form\ pf\ f\ f'\ x\ s \wedge derived\_form\ pg\ g$

g' x $s \longrightarrow$ *derived_form* (*IN* x $s \wedge pg \wedge pf$) (λx::*real*. *atn2* (f x, g x)) (($g' * f$ $x - g$ $x * f'$) / $(g\ x)^2$ * *inverse_class.inverse* ((*1*::*real*) + $(f\ x\ /\ g\ x)^2$)) x s

thm Calc_derivative.atn2_deriv_simple1:

$\forall$ (x::*real*) (pf::*bool*) (f::*real* $\Rightarrow$ *real*) (f'::*real*) (pg::*bool*) (g::*real* $\Rightarrow$ *real*) (g'::*real*) s::*real* $\Rightarrow$ *bool*. (*0*::*real*) $<$ f x $\wedge$ *derived_form* pf f f' x s $\wedge$ *derived_form* pg g g' x $s \longrightarrow$ *derived_form* (*IN* x $s \wedge pg \wedge pf$) (λx::*real*. *atn2* (f x, g x)) (($g' * f$ $x - g$ $x * f'$) / ($(f\ x)^2 + (g\ x)^2$)) x s

thm Calc_derivative.atn2_deriv_simple2:

$\forall$ (x::*real*) (pf::*bool*) (f::*real* $\Rightarrow$ *real*) (f'::*real*) (pg::*bool*) (g::*real* $\Rightarrow$ *real*) (g'::*real*) s::*real* $\Rightarrow$ *bool*. (*0*::*real*) $<$ g x $\wedge$ *derived_form* pf f f' x s $\wedge$ *derived_form* pg g g' x $s \longrightarrow$ *derived_form* (*IN* x $s \wedge pg \wedge pf$) (λx::*real*. *atn2* (f x, g x)) (($g' * f$ $x - g$ $x * f'$) / ($(f\ x)^2 + (g\ x)^2$)) x s

thm Calc_derivative.atn2_deriv_simple3:

$\forall$ (x::*real*) (pf::*bool*) (f::*real* $\Rightarrow$ *real*) (f'::*real*) (pg::*bool*) (g::*real* $\Rightarrow$ *real*) (g'::*real*) s::*real* $\Rightarrow$ *bool*. g x $<$ (*0*::*real*) $\wedge$ *derived_form* pf f f' x s $\wedge$ *derived_form* pg g g' x $s \longrightarrow$ *derived_form* (*IN* x $s \wedge pg \wedge pf$) (λx::*real*. *atn2* (f x, g x)) (($g' * f$ $x - g$ $x * f'$) / ($(f\ x)^2 + (g\ x)^2$)) x s

thm Calc_derivative.atn2_derivative:

$\forall$ (x::*real*) (pf::*bool*) (f::*real* $\Rightarrow$ *real*) (f'::*real*) (pg::*bool*) (g::*real* $\Rightarrow$ *real*) (g'::*real*) s::*real* $\Rightarrow$ *bool*. $\neg$ (g x = (*0*::*real*) $\wedge$ f x $\leq$ (*0*::*real*)) $\wedge$ *derived_form* pf f f' x s $\wedge$ *derived_form* pg g g' x $s \longrightarrow$ *derived_form* (*IN* x $s \wedge pg \wedge pf$) (λx::*real*. *atn2* (f x, g x)) (($g' * f$ $x - g$ $x * f'$) / ($(f\ x)^2 + (g\ x)^2$)) x s

thm DEF_atn2curry:

atn2curry = (λ(*_2362787*::*real*) *_2362788*::*real*. *atn2* (*_2362787*, *_2362788*))

thm Calc_derivative.atn2curry:

$\forall$ (x::*real*) y::*real*. *atn2curry* x y = *atn2* (x, y)

thm Calc_derivative.derived_form_atn2curry:

$\forall$ (x::*real*) s::*real* $\Rightarrow$ *bool*. *derived_form* (*?p1.0*::*bool*) (*?f1.0*::*real* $\Rightarrow$ *real*) (*?f1'*::*real*) x s $\wedge$ *derived_form* (*?p2.0*::*bool*) (*?f2.0*::*real* $\Rightarrow$ *real*) (*?f2'*::*real*) x $s \longrightarrow$ *derived_form* (*?p1.0* $\wedge$ *?p2.0* $\wedge$ $\neg$ (*?f2.0* x = (*0*::*real*) $\wedge$ *?f1.0* x $\leq$ (*0*::*real*)) $\wedge$ *IN* x s) (λx::*real*. *atn2curry* (*?f1.0* x) (*?f2.0* x)) ((*?f2'* $*$ *?f1.0* x $-$ *?f2.0* x $*$ *?f1'*) / ((*?f1.0* $x)^2$ + (*?f2.0* $x)^2$)) x s

thm Nonlinear_lemma.NONLIN:

NONLIN = *True*

thm DEF_mardih_x:

mardih_x = (λ(*_2363426*::*real*) (*_2363427*::*real*) (*_2363428*::*real*) (*_2363429*::*real*) (*_2363430*::*real*) *_2363431*::*real*. *marchal_quartic* (*sqrt* *_2363426* / *DECIMAL*

(*20*::*nat*) (*10*::*nat*)) ∗ *dih_x _2363426 _2363427 _2363428 _2363429 _2363430 _2363431*)

thm Nonlinear_lemma.mardih_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *mardih_x x1 x2 x3 x4 x5 x6* = *marchal_quartic* (*sqrt x1* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih_x x1 x2 x3 x4 x5 x6*

thm DEF_mardih2_x:

mardih2_x = (λ(*_2363486*::*real*) (*_2363487*::*real*) (*_2363488*::*real*) (*_2363489*::*real*) (*_2363490*::*real*) *_2363491*::*real*. *marchal_quartic* (*sqrt _2363487* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih2_x _2363486 _2363487 _2363488 _2363489 _2363490 _2363491*)

thm Nonlinear_lemma.mardih2_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *mardih2_x x1 x2 x3 x4 x5 x6* = *marchal_quartic* (*sqrt x2* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih2_x x1 x2 x3 x4 x5 x6*

thm DEF_mardih3_x:

mardih3_x = (λ(*_2363546*::*real*) (*_2363547*::*real*) (*_2363548*::*real*) (*_2363549*::*real*) (*_2363550*::*real*) *_2363551*::*real*. *marchal_quartic* (*sqrt _2363548* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih3_x _2363546 _2363547 _2363548 _2363549 _2363550 _2363551*)

thm Nonlinear_lemma.mardih3_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *mardih3_x x1 x2 x3 x4 x5 x6* = *marchal_quartic* (*sqrt x3* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih3_x x1 x2 x3 x4 x5 x6*

thm DEF_mardih4_x:

mardih4_x = (λ(*_2363606*::*real*) (*_2363607*::*real*) (*_2363608*::*real*) (*_2363609*::*real*) (*_2363610*::*real*) *_2363611*::*real*. *marchal_quartic* (*sqrt _2363609* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih4_x _2363606 _2363607 _2363608 _2363609 _2363610 _2363611*)

thm Nonlinear_lemma.mardih4_x:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *mardih4_x x1 x2 x3 x4 x5 x6* = *marchal_quartic* (*sqrt x4* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih4_x x1 x2 x3 x4 x5 x6*

thm DEF_mardih5_x:

mardih5_x = (λ(*_2363666*::*real*) (*_2363667*::*real*) (*_2363668*::*real*) (*_2363669*::*real*) (*_2363670*::*real*) *_2363671*::*real*. *marchal_quartic* (*sqrt _2363670* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih5_x _2363666 _2363667 _2363668 _2363669 _2363670 _2363671*)

thm Nonlinear_lemma.mardih5_x:

$\forall (x1::real)$ $(x2::real)$ $(x3::real)$ $(x4::real)$ $(x5::real)$ $x6::real.$ $mardih5\_x$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ $marchal\_quartic$ $(sqrt$ $x5$ $/$ $DECIMAL$ $(20::nat)$ $(10::nat))$ $*$ $dih5\_x$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$

thm DEF_mardih6_x:

$mardih6\_x = (\lambda(\_2363726::real)$ $(\_2363727::real)$ $(\_2363728::real)$ $(\_2363729::real)$ $(\_2363730::real)$ $\_2363731::real.$ $marchal\_quartic$ $(sqrt$ $\_2363731$ $/$ $DECIMAL$ $(20::nat)$ $(10::nat))$ $*$ $dih6\_x$ $\_2363726$ $\_2363727$ $\_2363728$ $\_2363729$ $\_2363730$ $\_2363731)$

thm Nonlinear_lemma.mardih6_x:

$\forall (x1::real)$ $(x2::real)$ $(x3::real)$ $(x4::real)$ $(x5::real)$ $x6::real.$ $mardih6\_x$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ $marchal\_quartic$ $(sqrt$ $x6$ $/$ $DECIMAL$ $(20::nat)$ $(10::nat))$ $*$ $dih6\_x$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$

thm DEF_sqrt_x1:

$sqrt\_x1$ $=$ $(SOME$ $sqrt\_x1::nat$ $\Rightarrow$ $real$ $\Rightarrow$ $?'e::type$ $\Rightarrow$ $?'d::type$ $\Rightarrow$ $?'c::type$ $\Rightarrow$ $?'b::type$ $\Rightarrow$ $?'a::type$ $\Rightarrow$ $real.$ $\forall (\_2364227::nat)$ $(x1::real)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $sqrt\_x1$ $\_2364227$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ $sqrt$ $x1)$ $(80::nat)$

thm Nonlin_def.sqrt_x1:

$sqrt\_x1$ $(?x1.0::real)$ $(?x2.0::?'e::type)$ $(?x3.0::?'d::type)$ $(?x4.0::?'c::type)$ $(?x5.0::?'b::type)$ $(?x6.0::?'a::type)$ $=$ $sqrt$ $?x1.0$

thm DEF_sqrt_x2:

$sqrt\_x2$ $=$ $(SOME$ $sqrt\_x2::nat$ $\Rightarrow$ $?'e::type$ $\Rightarrow$ $real$ $\Rightarrow$ $?'d::type$ $\Rightarrow$ $?'c::type$ $\Rightarrow$ $?'b::type$ $\Rightarrow$ $?'a::type$ $\Rightarrow$ $real.$ $\forall (\_2364669::nat)$ $(x1::?'e::type)$ $(x2::real)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $sqrt\_x2$ $\_2364669$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ $sqrt$ $x2)$ $(81::nat)$

thm Nonlin_def.sqrt_x2:

$sqrt\_x2$ $(?x1.0::?'e::type)$ $(?x2.0::real)$ $(?x3.0::?'d::type)$ $(?x4.0::?'c::type)$ $(?x5.0::?'b::type)$ $(?x6.0::?'a::type)$ $=$ $sqrt$ $?x2.0$

thm DEF_sqrt_x3:

$sqrt\_x3 = (SOME$ $sqrt\_x3::nat \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow real \Rightarrow ?'c::type \Rightarrow$ $?'b::type \Rightarrow ?'a::type \Rightarrow real.$ $\forall (\_2365111::nat)$ $(x1::?'e::type)$ $(x2::?'d::type)$ $(x3::real)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $sqrt\_x3$ $\_2365111$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ $sqrt$ $x3)$ $(82::nat)$

thm Nonlin_def.sqrt_x3:

$sqrt\_x3$ $(?x1.0::?'e::type)$ $(?x2.0::?'d::type)$ $(?x3.0::real)$ $(?x4.0::?'c::type)$ $(?x5.0::?'b::type)$ $(?x6.0::?'a::type)$ $=$ $sqrt$ $?x3.0$

thm DEF_sqrt_x4:

$sqrt\_x4 = (SOME\ sqrt\_x4::nat \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow real \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real.\ \forall (\_2365553::nat)\ (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::real)\ (x5::?'b::type)\ x6::?'a::type.\ sqrt\_x4\ \_2365553\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x4)\ (83::nat)$

thm Nonlin_def.sqrt_x4:

$sqrt\_x4\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::real)\ (?x5.0::?'b::type)\ (?x6.0::?'a::type) = sqrt\ ?x4.0$

thm DEF_sqrt_x5:

$sqrt\_x5 = (SOME\ sqrt\_x5::nat \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow real \Rightarrow ?'a::type \Rightarrow real.\ \forall (\_2365995::nat)\ (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::?'b::type)\ (x5::real)\ x6::?'a::type.\ sqrt\_x5\ \_2365995\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x5)\ (84::nat)$

thm Nonlin_def.sqrt_x5:

$sqrt\_x5\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::?'b::type)\ (?x5.0::real)\ (?x6.0::?'a::type) = sqrt\ ?x5.0$

thm DEF_sqrt_x6:

$sqrt\_x6 = (SOME\ sqrt\_x6::nat \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real \Rightarrow real.\ \forall (\_2366437::nat)\ (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::?'b::type)\ (x5::?'a::type)\ x6::real.\ sqrt\_x6\ \_2366437\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x6)\ (85::nat)$

thm Nonlin_def.sqrt_x6:

$sqrt\_x6\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::?'b::type)\ (?x5.0::?'a::type)\ (?x6.0::real) = sqrt\ ?x6.0$

thm DEF_halfbump_x:

$halfbump\_x = (\lambda\_2366438::real.\ bump\ (sqrt\ \_2366438\ /\ real\_of\_nat\ (2::nat)))$

thm Nonlinear_lemma.halfbump_x:

$\forall x::real.\ halfbump\_x\ x = bump\ (sqrt\ x\ /\ real\_of\_nat\ (2::nat))$

thm DEF_halfbump_x1:

$halfbump\_x1 = (\lambda(\_2366443::real)\ (\_2366444::?'e::type)\ (\_2366445::?'d::type)\ (\_2366446::?'c::type)\ (\_2366447::?'b::type)\ \_2366448::?'a::type.\ halfbump\_x\ \_2366443)$

thm Nonlin_def.halfbump_x1:

$\forall (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x1::real.\ halfbump\_x1\ x1\ x2\ x3\ x4\ x5\ x6 = halfbump\_x\ x1$

thm DEF_halfbump_x4:

halfbump_x4 = (λ(*_2366503*::*?'e*::*type*) (*_2366504*::*?'d*::*type*) (*_2366505*::*?'c*::*type*) (*_2366506*::*real*) (*_2366507*::*?'b*::*type*) *_2366508*::*?'a*::*type*. *halfbump_x _2366506*)

thm Nonlin_def.halfbump_x4:

∀ (*x1*::*?'e*::*type*) (*x2*::*?'d*::*type*) (*x3*::*?'c*::*type*) (*x5*::*?'b*::*type*) (*x6*::*?'a*::*type*) *x4*::*real*. *halfbump_x4 x1 x2 x3 x4 x5 x6* = *halfbump_x x4*

thm DEF_unit6:

unit6 = (*SOME unit6*::*nat* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *real*. ∀ (*_2367004*::*nat*) (*x1*::*?'f*::*type*) (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) *x6*::*?'a*::*type*. *unit6 _2367004 x1 x2 x3 x4 x5 x6* = (*1*::*real*)) (*86*::*nat*)

thm Nonlin_def.unit6:

unit6 (*?x1.0*::*?'f*::*type*) (*?x2.0*::*?'e*::*type*) (*?x3.0*::*?'d*::*type*) (*?x4.0*::*?'c*::*type*) (*?x5.0*::*?'b*::*type*) (*?x6.0*::*?'a*::*type*) = (*1*::*real*)

thm DEF_proj_x1:

proj_x1 = (*SOME proj_x1*::*nat* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'f*::*type*. ∀ (*_2367446*::*nat*) (*x1*::*?'f*::*type*) (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) *x6*::*?'a*::*type*. *proj_x1 _2367446 x1 x2 x3 x4 x5 x6* = *x1*) (*87*::*nat*)

thm Functional_equation.proj_x1:

proj_x1 (*?x1.0*::*?'f*::*type*) (*?x2.0*::*?'e*::*type*) (*?x3.0*::*?'d*::*type*) (*?x4.0*::*?'c*::*type*) (*?x5.0*::*?'b*::*type*) (*?x6.0*::*?'a*::*type*) = *?x1.0*

thm DEF_proj_x2:

proj_x2 = (*SOME proj_x2*::*nat* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'e*::*type*. ∀ (*_2367888*::*nat*) (*x1*::*?'f*::*type*) (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) *x6*::*?'a*::*type*. *proj_x2 _2367888 x1 x2 x3 x4 x5 x6* = *x2*) (*88*::*nat*)

thm Functional_equation.proj_x2:

proj_x2 (*?x1.0*::*?'e*::*type*) (*?x2.0*::*?'f*::*type*) (*?x3.0*::*?'d*::*type*) (*?x4.0*::*?'c*::*type*) (*?x5.0*::*?'b*::*type*) (*?x6.0*::*?'a*::*type*) = *?x2.0*

thm DEF_proj_x3:

proj_x3 = (*SOME proj_x3*::*nat* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'d*::*type*. ∀ (*_2368330*::*nat*) (*x1*::*?'f*::*type*) (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) *x6*::*?'a*::*type*. *proj_x3 _2368330 x1 x2 x3 x4 x5 x6* = *x3*) (*89*::*nat*)

thm Functional_equation.proj_x3:

proj_x3 (*?x1.0*::*?'e*::*type*) (*?x2.0*::*?'d*::*type*) (*?x3.0*::*?'f*::*type*) (*?x4.0*::*?'c*::*type*) (*?x5.0*::*?'b*::*type*) (*?x6.0*::*?'a*::*type*) = *?x3.0*

thm DEF_proj_x4:

$proj\_x4 = (SOME\ proj\_x4::nat \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'c::type. \forall (\_2368772::nat)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ proj\_x4\ \_2368772\ x1\ x2\ x3\ x4\ x5\ x6 = x4)\ (90::nat)$

thm Functional_equation.proj_x4:

$proj\_x4\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::?'f::type)\ (?x5.0::?'b::type)\ (?x6.0::?'a::type) = ?x4.0$

thm DEF_proj_x5:

$proj\_x5 = (SOME\ proj\_x5::nat \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'b::type. \forall (\_2369214::nat)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ proj\_x5\ \_2369214\ x1\ x2\ x3\ x4\ x5\ x6 = x5)\ (91::nat)$

thm Functional_equation.proj_x5:

$proj\_x5\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::?'b::type)\ (?x5.0::?'f::type)\ (?x6.0::?'a::type) = ?x5.0$

thm DEF_proj_x6:

$proj\_x6 = (SOME\ proj\_x6::nat \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'a::type. \forall (\_2369656::nat)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ proj\_x6\ \_2369656\ x1\ x2\ x3\ x4\ x5\ x6 = x6)\ (92::nat)$

thm Functional_equation.proj_x6:

$proj\_x6\ (?x1.0::?'e::type)\ (?x2.0::?'d::type)\ (?x3.0::?'c::type)\ (?x4.0::?'b::type)\ (?x5.0::?'a::type)\ (?x6.0::?'f::type) = ?x6.0$

thm DEF_promote:

$promote = (SOME\ promote::nat \Rightarrow (?'g::type \Rightarrow ?'f::type) \Rightarrow ?'g::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'f::type. \forall (\_2370215::nat)\ (f::?'g::type \Rightarrow ?'f::type)\ (x1::?'g::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ promote\ \_2370215\ f\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1)\ (93::nat)$

thm Nonlin_def.promote:

$promote\ (?f::?'f::type \Rightarrow ?'g::type)\ (?x1.0::?'f::type)\ (?x2.0::?'e::type)\ (?x3.0::?'d::type)\ (?x4.0::?'c::type)\ (?x5.0::?'b::type)\ (?x6.0::?'a::type) = ?f\ ?x1.0$

thm DEF_unit0:

$unit0 = (SOME\ unit0::nat \Rightarrow real. \forall \_2370216::nat.\ unit0\ \_2370216 = (1::real))\ (94::nat)$

thm Nonlin_def.unit0:

$unit0 = (1::real)$

thm DEF_pow1:

$pow1 = (\lambda\_2370217::real.\ \_2370217^{1::nat})$

thm Nonlinear_lemma.pow1:

$\forall y::real.\ pow1\ y = y^{1::nat}$

thm DEF_pow2:

$pow2 = power2$

thm Nonlinear_lemma.pow2:

$\forall y::real.\ pow2\ y = y^2$

thm DEF_pow3:

$pow3 = (\lambda\_2370227::real.\ \_2370227^{3::nat})$

thm Nonlinear_lemma.pow3:

$\forall y::real.\ pow3\ y = y^{3::nat}$

thm DEF_pow4:

$pow4 = (\lambda\_2370232::real.\ \_2370232^{4::nat})$

thm Nonlinear_lemma.pow4:

$\forall y::real.\ pow4\ y = y^{4::nat}$

thm DEF_promote_pow2:

$promote\_pow2 = (\lambda(\_2370237::real)\ (\_2370238::?'e::type)\ (\_2370239::?'d::type)\ (\_2370240::?'c::type)\ (\_2370241::?'b::type)\ \_2370242::?'a::type.\ \_2370237^2)$

thm Nonlinear_lemma.promote_pow2:

$\forall (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x1::real.\ promote\_pow2\ x1\ x2\ x3\ x4\ x5\ x6 = x1^2$

thm DEF_promote_pow3:

$promote\_pow3 = (\lambda(\_2370297::real)\ (\_2370298::?'e::type)\ (\_2370299::?'d::type)\ (\_2370300::?'c::type)\ (\_2370301::?'b::type)\ \_2370302::?'a::type.\ \_2370297^{3::nat})$

thm Nonlinear_lemma.promote_pow3:

$\forall (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x1::real.\ promote\_pow3\ x1\ x2\ x3\ x4\ x5\ x6 = x1^{3::nat}$

thm DEF_compose6:

$compose6 = (\lambda(\_2370357::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real)\ (\_2370358::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real)\ (\_2370359::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real)\ (\_2370360::real \Rightarrow real \Rightarrow real$

⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*_2370361::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*_2370362::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*_2370363::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*_2370364::real*) (*_2370365::real*) (*_2370366::real*) (*_2370367::real*) (*_2370368::real*) *_2370369::real*. *_2370357* (*_2370358 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*) (*_2370359 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*) (*_2370360 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*) (*_2370361 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*) (*_2370362 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*) (*_2370363 _2370364 _2370365 _2370366 _2370367 _2370368 _2370369*))

thm Functional_equation.compose6:

∀ (*f::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p1::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p2::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p3::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p4::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p5::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*p6::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real*. *compose6 f p1 p2 p3 p4 p5 p6 x1 x2 x3 x4 x5 x6* = *f* (*p1 x1 x2 x3 x4 x5 x6*) (*p2 x1 x2 x3 x4 x5 x6*) (*p3 x1 x2 x3 x4 x5 x6*) (*p4 x1 x2 x3 x4 x5 x6*) (*p5 x1 x2 x3 x4 x5 x6*) (*p6 x1 x2 x3 x4 x5 x6*)

thm DEF_scale6:

scale6 = (λ(*_2370578::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*_2370579::real*) (*_2370580::real*) (*_2370581::real*) (*_2370582::real*) (*_2370583::real*) (*_2370584::real*) *_2370585::real*. *_2370578 _2370580 _2370581 _2370582 _2370583 _2370584 _2370585* ∗ *_2370579*)

thm Nonlin_def.scale6:

∀ (*f::real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) (*x6::real*) *r::real*. *scale6 f r x1 x2 x3 x4 x5 x6* = *f x1 x2 x3 x4 x5 x6* ∗ *r*

thm DEF_quadratic_root_plus_curry:

quadratic_root_plus_curry = (λ(*_2370674::real*) (*_2370675::real*) *_2370676::real*. *quadratic_root_plus* (*_2370674*, *_2370675*, *_2370676*))

thm Nonlinear_lemma.quadratic_root_plus_curry:

∀ (*a::real*) (*b::real*) *c::real*. *quadratic_root_plus_curry a b c* = *quadratic_root_plus* (*a*, *b*, *c*)

thm DEF_gamma3f_135_s_n:

gamma3f_135_s_n = (λ(*_2370695::real*) (*_2370696::real*) (*_2370697::real*) (*_2370698::real*) (*_2370699::real*) *_2370700::real*. *sqn* (*delta_y _2370695 _2370696 _2370697 _2370698 _2370699 _2370700*) ∗ ((*1::real*) / *real_of_nat* (*12::nat*) − *real_of_nat* (*2::nat*) ∗ (*mm1* / *pi*) ∗ (*y_of_x sol_euler_x_div_sqrtdelta _2370695 _2370696*

$\_2370697\ \_2370698\ \_2370699\ \_2370700 + (y\_of\_x\ sol\_euler156\_x\_div\_sqrtdelta\ \_2370695\ \_2370696\ \_2370697\ \_2370698\ \_2370699\ \_2370700 + y\_of\_x\ sol\_euler345\_x\_div\_sqrtdelta\ \_2370695\ \_2370696\ \_2370697\ \_2370698\ \_2370699\ \_2370700))))$

thm Nonlin_def.gamma3f_135_s_n:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ gamma3f\_135\_s\_n\ y1\ y2\ y3\ y4\ y5\ y6 = sqn\ (delta\_y\ y1\ y2\ y3\ y4\ y5\ y6) * ((1::real)\ /\ real\_of\_nat\ (12::nat) - real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (y\_of\_x\ sol\_euler\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6 + (y\_of\_x\ sol\_euler156\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6 + y\_of\_x\ sol\_euler345\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6)))$

thm DEF_gamma3f_126_s_n:

$gamma3f\_126\_s\_n = (\lambda(\_2370755::real)\ (\_2370756::real)\ (\_2370757::real)\ (\_2370758::real)\ (\_2370759::real)\ \_2370760::real.\ sqn\ (delta\_y\ \_2370755\ \_2370756\ \_2370757\ \_2370758\ \_2370759\ \_2370760) * ((1::real)\ /\ real\_of\_nat\ (12::nat) - real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (y\_of\_x\ sol\_euler\_x\_div\_sqrtdelta\ \_2370755\ \_2370756\ \_2370757\ \_2370758\ \_2370759\ \_2370760 + (y\_of\_x\ sol\_euler246\_x\_div\_sqrtdelta\ \_2370755\ \_2370756\ \_2370757\ \_2370758\ \_2370759\ \_2370760 + y\_of\_x\ sol\_euler156\_x\_div\_sqrtdelta\ \_2370755\ \_2370756\ \_2370757\ \_2370758\ \_2370759\ \_2370760))))$

thm Nonlinear_lemma.gamma3f_126_s_n:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ gamma3f\_126\_s\_n\ y1\ y2\ y3\ y4\ y5\ y6 = sqn\ (delta\_y\ y1\ y2\ y3\ y4\ y5\ y6) * ((1::real)\ /\ real\_of\_nat\ (12::nat) - real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (y\_of\_x\ sol\_euler\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6 + (y\_of\_x\ sol\_euler246\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6 + y\_of\_x\ sol\_euler156\_x\_div\_sqrtdelta\ y1\ y2\ y3\ y4\ y5\ y6)))$

thm DEF_lmdih_x_n:

$lmdih\_x\_n = (\lambda(\_2370815::real)\ (\_2370816::real)\ (\_2370817::real)\ (\_2370818::real)\ (\_2370819::real)\ \_2370820::real.\ sqn\ (delta\_x\ \_2370815\ \_2370816\ \_2370817\ \_2370818\ \_2370819\ \_2370820) * lmdih\_x\_div\_sqrtdelta\_posbranch\ \_2370815\ \_2370816\ \_2370817\ \_2370818\ \_2370819\ \_2370820)$

thm Nonlinear_lemma.lmdih_x_n:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ lmdih\_x\_n\ x1\ x2\ x3\ x4\ x5\ x6 = sqn\ (delta\_x\ x1\ x2\ x3\ x4\ x5\ x6) * lmdih\_x\_div\_sqrtdelta\_posbranch\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_lmdih2_x_n:

$lmdih2\_x\_n = (\lambda(\_2370875::real)\ (\_2370876::real)\ (\_2370877::real)\ (\_2370878::real)\ (\_2370879::real)\ \_2370880::real.\ sqn\ (delta\_x\ \_2370875\ \_2370876\ \_2370877\ \_2370878\ \_2370879\ \_2370880) * lmdih2\_x\_div\_sqrtdelta\_posbranch\ \_2370875\ \_2370876\ \_2370877\ \_2370878\ \_2370879\ \_2370880)$

thm Nonlinear_lemma.lmdih2_x_n:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. lmdih2_x_n x1 x2 x3 x4 x5 x6 = sqn (delta_x x1 x2 x3 x4 x5 x6) * lmdih2_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6

thm DEF_lmdih3_x_n:

lmdih3_x_n = (λ(_2370935::real) (_2370936::real) (_2370937::real) (_2370938::real) (_2370939::real) _2370940::real. sqn (delta_x _2370935 _2370936 _2370937 _2370938 _2370939 _2370940) * lmdih3_x_div_sqrtdelta_posbranch _2370935 _2370936 _2370937 _2370938 _2370939 _2370940)

thm Nonlinear_lemma.lmdih3_x_n:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. lmdih3_x_n x1 x2 x3 x4 x5 x6 = sqn (delta_x x1 x2 x3 x4 x5 x6) * lmdih3_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6

thm DEF_lmdih5_x_n:

lmdih5_x_n = (λ(_2370995::real) (_2370996::real) (_2370997::real) (_2370998::real) (_2370999::real) _2371000::real. sqn (delta_x _2370995 _2370996 _2370997 _2370998 _2370999 _2371000) * lmdih5_x_div_sqrtdelta_posbranch _2370995 _2370996 _2370997 _2370998 _2370999 _2371000)

thm Nonlinear_lemma.lmdih5_x_n:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. lmdih5_x_n x1 x2 x3 x4 x5 x6 = sqn (delta_x x1 x2 x3 x4 x5 x6) * lmdih5_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6

thm DEF_lmdih6_x_n:

lmdih6_x_n = (λ(_2371055::real) (_2371056::real) (_2371057::real) (_2371058::real) (_2371059::real) _2371060::real. sqn (delta_x _2371055 _2371056 _2371057 _2371058 _2371059 _2371060) * lmdih6_x_div_sqrtdelta_posbranch _2371055 _2371056 _2371057 _2371058 _2371059 _2371060)

thm Nonlinear_lemma.lmdih6_x_n:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. lmdih6_x_n x1 x2 x3 x4 x5 x6 = sqn (delta_x x1 x2 x3 x4 x5 x6) * lmdih6_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6

thm DEF_gamma3f_vLR_n:

gamma3f_vLR_n = (λ(_2371115::real) (_2371116::real) (_2371117::real) (_2371118::real) (_2371119::real) (_2371120::real) _2371121::real ⇒ real. (dih_y _2371115 _2371116 _2371117 _2371118 _2371119 _2371120 − upper_dih_y _2371115 _2371116 sqrt2 sqrt2 sqrt2 _2371120 − upper_dih_y _2371115 _2371117 sqrt2 sqrt2 sqrt2 _2371119) * ((vol2r _2371115 sqrt2 − vol2f _2371115 sqrt2 _2371121) / (real_of_nat (2::nat) * pi)))

thm Nonlinear_lemma.gamma3f_vLR_n:

∀ (y4::real) (y2::real) (y6::real) (y3::real) (y5::real) (y1::real) f::real ⇒ real. gamma3f_vLR_n y1 y2 y3 y4 y5 y6 f = (dih_y y1 y2 y3 y4 y5 y6 − upper_dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − upper_dih_y y1 y3 sqrt2 sqrt2 sqrt2 y5) ∗ ((vol2r y1 sqrt2 − vol2f y1 sqrt2 f) / (real_of_nat (2::nat) ∗ pi))

thm DEF_gamma3f_vL_n:

gamma3f_vL_n = (λ(_2371192::real) (_2371193::real) (_2371194::real) (_2371195::real) (_2371196::real) (_2371197::real) _2371198::real ⇒ real. (dih_y _2371192 _2371193 _2371194 _2371195 _2371196 _2371197 − upper_dih_y _2371192 _2371193 sqrt2 sqrt2 sqrt2 _2371197 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r _2371192 sqrt2 − vol2f _2371192 sqrt2 _2371198) / (real_of_nat (2::nat) ∗ pi)))

thm Nonlin_def.gamma3f_vL_n:

∀ (y3::real) (y4::real) (y5::real) (y2::real) (y6::real) (y1::real) f::real ⇒ real. gamma3f_vL_n y1 y2 y3 y4 y5 y6 f = (dih_y y1 y2 y3 y4 y5 y6 − upper_dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r y1 sqrt2 − vol2f y1 sqrt2 f) / (real_of_nat (2::nat) ∗ pi))

thm DEF_gamma3f_vLR_n0:

gamma3f_vLR_n0 = (λ(_2371269::real) (_2371270::real) (_2371271::real) (_2371272::real) (_2371273::real) _2371274::real. gamma3f_vLR_n _2371269 _2371270 _2371271 _2371272 _2371273 _2371274 (λx::real. 0::real))

thm Nonlin_def.gamma3f_vLR_n0:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. gamma3f_vLR_n0 y1 y2 y3 y4 y5 y6 = gamma3f_vLR_n y1 y2 y3 y4 y5 y6 (λx::real. 0::real)

thm DEF_gamma3f_vLR_nlfun:

gamma3f_vLR_nlfun = (λ(_2371329::real) (_2371330::real) (_2371331::real) (_2371332::real) (_2371333::real) _2371334::real. gamma3f_vLR_n _2371329 _2371330 _2371331 _2371332 _2371333 _2371334 lfun)

thm Nonlin_def.gamma3f_vLR_nlfun:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. gamma3f_vLR_nlfun y1 y2 y3 y4 y5 y6 = gamma3f_vLR_n y1 y2 y3 y4 y5 y6 lfun

thm DEF_gamma3f_vL_n0:

gamma3f_vL_n0 = (λ(_2371389::real) (_2371390::real) (_2371391::real) (_2371392::real) (_2371393::real) _2371394::real. gamma3f_vL_n _2371389 _2371390 _2371391 _2371392 _2371393 _2371394 (λx::real. 0::real))

thm Nonlinear_lemma.gamma3f_vL_n0:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. gamma3f_vL_n0 y1 y2 y3 y4 y5 y6 = gamma3f_vL_n y1 y2 y3 y4 y5 y6 (λx::real. 0::real)

thm DEF_gamma3f_vL_nlfun:

gamma3f_vL_nlfun = (λ(*_2371449*::*real*) (*_2371450*::*real*) (*_2371451*::*real*) (*_2371452*::*real*) (*_2371453*::*real*) *_2371454*::*real*. *gamma3f_vL_n _2371449 _2371450 _2371451 _2371452 _2371453 _2371454 lfun*)

thm Nonlin_def.gamma3f_vL_nlfun:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *gamma3f_vL_nlfun y1 y2 y3 y4 y5 y6* = *gamma3f_vL_n y1 y2 y3 y4 y5 y6 lfun*

thm DEF_ldih_x_n:

ldih_x_n = (λ(*_2371509*::*real*) (*_2371510*::*real*) (*_2371511*::*real*) (*_2371512*::*real*) (*_2371513*::*real*) *_2371514*::*real*. *sqn* (*delta_x _2371509 _2371510 _2371511 _2371512 _2371513 _2371514*) ∗ *ldih_x_div_sqrtdelta_posbranch _2371509 _2371510 _2371511 _2371512 _2371513 _2371514*)

thm Nonlin_def.ldih_x_n:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih_x_n x1 x2 x3 x4 x5 x6* = *sqn* (*delta_x x1 x2 x3 x4 x5 x6*) ∗ *ldih_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_ldih2_x_n:

ldih2_x_n = (λ(*_2371569*::*real*) (*_2371570*::*real*) (*_2371571*::*real*) (*_2371572*::*real*) (*_2371573*::*real*) *_2371574*::*real*. *sqn* (*delta_x _2371569 _2371570 _2371571 _2371572 _2371573 _2371574*) ∗ *ldih2_x_div_sqrtdelta_posbranch _2371569 _2371570 _2371571 _2371572 _2371573 _2371574*)

thm Nonlinear_lemma.ldih2_x_n:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih2_x_n x1 x2 x3 x4 x5 x6* = *sqn* (*delta_x x1 x2 x3 x4 x5 x6*) ∗ *ldih2_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_ldih3_x_n:

ldih3_x_n = (λ(*_2371629*::*real*) (*_2371630*::*real*) (*_2371631*::*real*) (*_2371632*::*real*) (*_2371633*::*real*) *_2371634*::*real*. *sqn* (*delta_x _2371629 _2371630 _2371631 _2371632 _2371633 _2371634*) ∗ *ldih3_x_div_sqrtdelta_posbranch _2371629 _2371630 _2371631 _2371632 _2371633 _2371634*)

thm Nonlinear_lemma.ldih3_x_n:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih3_x_n x1 x2 x3 x4 x5 x6* = *sqn* (*delta_x x1 x2 x3 x4 x5 x6*) ∗ *ldih3_x_div_sqrtdelta_posbranch x1 x2 x3 x4 x5 x6*

thm DEF_ldih5_x_n:

ldih5_x_n = (λ(*_2371689*::*real*) (*_2371690*::*real*) (*_2371691*::*real*) (*_2371692*::*real*) (*_2371693*::*real*) *_2371694*::*real*. *sqn* (*delta_x _2371689 _2371690 _2371691 _2371692 _2371693 _2371694*) ∗ *ldih5_x_div_sqrtdelta_posbranch _2371689 _2371690 _2371691 _2371692 _2371693 _2371694*)

thm Nonlinear_lemma.ldih5_x_n:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih5\_x\_n\ x1\ x2\ x3\ x4\ x5\ x6 = sqn\ (delta\_x\ x1\ x2\ x3\ x4\ x5\ x6) * ldih5\_x\_div\_sqrtdelta\_posbranch\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_ldih6_x_n:

$ldih6\_x\_n = (\lambda(\_2371749::real)\ (\_2371750::real)\ (\_2371751::real)\ (\_2371752::real)\ (\_2371753::real)\ \_2371754::real.\ sqn\ (delta\_x\ \_2371749\ \_2371750\ \_2371751\ \_2371752\ \_2371753\ \_2371754) * ldih6\_x\_div\_sqrtdelta\_posbranch\ \_2371749\ \_2371750\ \_2371751\ \_2371752\ \_2371753\ \_2371754)$

thm Nonlinear_lemma.ldih6_x_n:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih6\_x\_n\ x1\ x2\ x3\ x4\ x5\ x6 = sqn\ (delta\_x\ x1\ x2\ x3\ x4\ x5\ x6) * ldih6\_x\_div\_sqrtdelta\_posbranch\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_gamma3f_vLR_x_nlfun:

$gamma3f\_vLR\_x\_nlfun = (\lambda(\_2371809::real)\ (\_2371810::real)\ (\_2371811::real)\ (\_2371812::real)\ (\_2371813::real)\ \_2371814::real.\ gamma3f\_vLR\_nlfun\ (sqrt\ \_2371809)\ (sqrt\ \_2371810)\ (sqrt\ \_2371811)\ (sqrt\ \_2371812)\ (sqrt\ \_2371813)\ (sqrt\ \_2371814))$

thm Nonlinear_lemma.gamma3f_vLR_x_nlfun:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma3f\_vLR\_x\_nlfun\ x1\ x2\ x3\ x4\ x5\ x6 = gamma3f\_vLR\_nlfun\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6)$

thm DEF_gamma3f_vL_x_nlfun:

$gamma3f\_vL\_x\_nlfun = (\lambda(\_2371869::real)\ (\_2371870::real)\ (\_2371871::real)\ (\_2371872::real)\ (\_2371873::real)\ \_2371874::real.\ gamma3f\_vL\_nlfun\ (sqrt\ \_2371869)\ (sqrt\ \_2371870)\ (sqrt\ \_2371871)\ (sqrt\ \_2371872)\ (sqrt\ \_2371873)\ (sqrt\ \_2371874))$

thm Nonlinear_lemma.gamma3f_vL_x_nlfun:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma3f\_vL\_x\_nlfun\ x1\ x2\ x3\ x4\ x5\ x6 = gamma3f\_vL\_nlfun\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6)$

thm DEF_gamma3f_vLR_x_n0:

$gamma3f\_vLR\_x\_n0 = (\lambda(\_2371929::real)\ (\_2371930::real)\ (\_2371931::real)\ (\_2371932::real)\ (\_2371933::real)\ \_2371934::real.\ gamma3f\_vLR\_n0\ (sqrt\ \_2371929)\ (sqrt\ \_2371930)\ (sqrt\ \_2371931)\ (sqrt\ \_2371932)\ (sqrt\ \_2371933)\ (sqrt\ \_2371934))$

thm Nonlin_def.gamma3f_vLR_x_n0:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma3f\_vLR\_x\_n0\ x1\ x2\ x3\ x4\ x5\ x6 = gamma3f\_vLR\_n0\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6)$

thm DEF_gamma3f_vL_x_n0:

gamma3f_vL_x_n0 = (λ(*_2371989*::*real*) (*_2371990*::*real*) (*_2371991*::*real*) (*_2371992*::*real*) (*_2371993*::*real*) *_2371994*::*real*. *gamma3f_vL_n0* (*sqrt _2371989*) (*sqrt _2371990*) (*sqrt _2371991*) (*sqrt _2371992*) (*sqrt _2371993*) (*sqrt _2371994*))

thm Nonlin_def.gamma3f_vL_x_n0:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_vL_x_n0 x1 x2 x3 x4 x5 x6* = *gamma3f_vL_n0* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_135_x_s_n:

gamma3f_135_x_s_n = (λ(*_2372049*::*real*) (*_2372050*::*real*) (*_2372051*::*real*) (*_2372052*::*real*) (*_2372053*::*real*) *_2372054*::*real*. *gamma3f_135_s_n* (*sqrt _2372049*) *sqrt2* (*sqrt _2372051*) *sqrt2* (*sqrt _2372053*) *sqrt2*)

thm Nonlinear_lemma.gamma3f_135_x_s_n:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *gamma3f_135_x_s_n x1 x2 x3 x4 x5 x6* = *gamma3f_135_s_n* (*sqrt x1*) *sqrt2* (*sqrt x3*) *sqrt2* (*sqrt x5*) *sqrt2*

thm DEF_gamma3f_126_x_s_n:

gamma3f_126_x_s_n = (λ(*_2372109*::*real*) (*_2372110*::*real*) (*_2372111*::*real*) (*_2372112*::*real*) (*_2372113*::*real*) *_2372114*::*real*. *gamma3f_126_s_n* (*sqrt _2372109*) (*sqrt _2372110*) *sqrt2 sqrt2 sqrt2* (*sqrt _2372114*))

thm Nonlinear_lemma.gamma3f_126_x_s_n:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *gamma3f_126_x_s_n x1 x2 x3 x4 x5 x6* = *gamma3f_126_s_n* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm Nonlinear_lemma.sq_pow2:

∀ (*a*::*real*) *x*::*real*. $a^2 \leq x \longrightarrow$ *sqrt x* ∗ *sqrt x* = *x*

thm Nonlinear_lemma.sqrt2_sqrt2:

sqrt2 ∗ *sqrt2* = *real_of_nat* (*2*::*nat*)

thm DEF_ldih_x_126_n:

ldih_x_126_n = (λ(*_2372173*::*real*) (*_2372174*::*real*) (*_2372175*::*real*) (*_2372176*::*real*) *_2372177*::*real*. *ldih_x_n _2372173 _2372174* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)))

thm Nonlin_def.ldih_x_126_n:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *ldih_x_126_n x1 x2 x3 x4 x5 x6* = *ldih_x_n x1 x2* (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) *x6*

thm DEF_ldih2_x_126_n:

ldih2_x_126_n = (λ(*_2372233*::*real*) (*_2372234*::*real*) (*_2372235*::*real*) (*_2372236*::*real*) *_2372237*::*real*. *ldih2_x_n _2372233 _2372234* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)))

thm Nonlin_def.ldih2_x_126_n:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *ldih2_x_126_n x1 x2 x3 x4 x5 x6* = *ldih2_x_n x1 x2* (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) *x6*

thm DEF_ldih6_x_126_n:

ldih6_x_126_n = (λ(*_2372293*::*real*) (*_2372294*::*real*) (*_2372295*::*real*) (*_2372296*::*real*) *_2372297*::*real*. *ldih6_x_n _2372293 _2372294* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)))

thm Nonlin_def.ldih6_x_126_n:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *ldih6_x_126_n x1 x2 x3 x4 x5 x6* = *ldih6_x_n x1 x2* (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) (*sqrt2* ∗ *sqrt2*) *x6*

thm DEF_ldih_x_135_n:

ldih_x_135_n = (λ(*_2372353*::*real*) (*_2372354*::*real*) (*_2372355*::*real*) (*_2372356*::*real*) (*_2372357*::*real*) *_2372358*::*real*. *ldih_x_n _2372353* (*real_of_nat* (*2*::*nat*)) *_2372355* (*real_of_nat* (*2*::*nat*)) *_2372357* (*real_of_nat* (*2*::*nat*)))

thm Nonlinear_lemma.ldih_x_135_n:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *ldih_x_135_n x1 x2 x3 x4 x5 x6* = *ldih_x_n x1* (*sqrt2* ∗ *sqrt2*) *x3* (*sqrt2* ∗ *sqrt2*) *x5* (*sqrt2* ∗ *sqrt2*)

thm DEF_ldih3_x_135_n:

ldih3_x_135_n = (λ(*_2372413*::*real*) (*_2372414*::*real*) (*_2372415*::*real*) (*_2372416*::*real*) (*_2372417*::*real*) *_2372418*::*real*. *ldih3_x_n _2372413* (*real_of_nat* (*2*::*nat*)) *_2372415* (*real_of_nat* (*2*::*nat*)) *_2372417* (*real_of_nat* (*2*::*nat*)))

thm Nonlinear_lemma.ldih3_x_135_n:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *ldih3_x_135_n x1 x2 x3 x4 x5 x6* = *ldih3_x_n x1* (*sqrt2* ∗ *sqrt2*) *x3* (*sqrt2* ∗ *sqrt2*) *x5* (*sqrt2* ∗ *sqrt2*)

thm DEF_ldih5_x_135_n:

ldih5_x_135_n = (λ(*_2372473*::*real*) (*_2372474*::*real*) (*_2372475*::*real*) (*_2372476*::*real*) (*_2372477*::*real*) *_2372478*::*real*. *ldih5_x_n _2372473* (*real_of_nat* (*2*::*nat*)) *_2372475* (*real_of_nat* (*2*::*nat*)) *_2372477* (*real_of_nat* (*2*::*nat*)))

thm Nonlin_def.ldih5_x_135_n:

∀ (x2::real) (x4::real) (x6::real) (x1::real) (x3::real) x5::real. ldih5_x_135_n x1 x2 x3 x4 x5 x6 = ldih5_x_n x1 (sqrt2 ∗ sqrt2) x3 (sqrt2 ∗ sqrt2) x5 (sqrt2 ∗ sqrt2)

thm DEF_vol3f_sqrt2_lmplus:

vol3f_sqrt2_lmplus = (λ(_2372533::real) (_2372534::real) (_2372535::real) (_2372536::real) (_2372537::real) _2372538::real. real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ dih_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + (real_of_nat (2::nat) ∗ dih2_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + (real_of_nat (2::nat) ∗ dih6_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + (dih3_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + (dih4_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + (dih5_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 − real_of_nat (3::nat) ∗ pi)))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (lfun (_2372534 / real_of_nat (2::nat)) ∗ dih2_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538 + lfun (_2372538 / real_of_nat (2::nat)) ∗ dih6_y _2372533 _2372534 sqrt2 sqrt2 sqrt2 _2372538))

thm Nonlinear_lemma.vol3f_sqrt2_lmplus:

∀ (y3::real) (y4::real) (y5::real) (y1::real) (y2::real) y6::real. vol3f_sqrt2_lmplus y1 y2 y3 y4 y5 y6 = real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + (real_of_nat (2::nat) ∗ dih2_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + (real_of_nat (2::nat) ∗ dih6_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + (dih3_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + (dih4_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + (dih5_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − real_of_nat (3::nat) ∗ pi)))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (lfun (y2 / real_of_nat (2::nat)) ∗ dih2_y y1 y2 sqrt2 sqrt2 sqrt2 y6 + lfun (y6 / real_of_nat (2::nat)) ∗ dih6_y y1 y2 sqrt2 sqrt2 sqrt2 y6)

thm DEF_vol3f_x_sqrt2_lmplus:

vol3f_x_sqrt2_lmplus = (λ(_2372593::real) (_2372594::real) (_2372595::real) (_2372596::real) (_2372597::real) _2372598::real. vol3f_sqrt2_lmplus (sqrt _2372593) (sqrt _2372594) (sqrt _2372595) (sqrt _2372596) (sqrt _2372597) (sqrt _2372598))

thm Nonlin_def.vol3f_x_sqrt2_lmplus:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. vol3f_x_sqrt2_lmplus x1 x2 x3 x4 x5 x6 = vol3f_sqrt2_lmplus (sqrt x1) (sqrt x2) (sqrt x3) (sqrt x4) (sqrt x5) (sqrt x6)

thm DEF_vol3f_x_lfun:

vol3f_x_lfun = (λ(_2372653::real) (_2372654::real) (_2372655::real) (_2372656::real) (_2372657::real) _2372658::real. vol3f (sqrt _2372653) (sqrt _2372654) (sqrt _2372658) sqrt2 lfun)

thm Nonlinear_lemma.vol3f_x_lfun:

∀ (x3::real) (x4::real) (x5::real) (x1::real) (x2::real) x6::real. vol3f_x_lfun x1 x2 x3 x4 x5 x6 = vol3f (sqrt x1) (sqrt x2) (sqrt x6) sqrt2 lfun

thm DEF_vol3_x_sqrt:

vol3_x_sqrt = (λ(*_2372713*::*real*) (*_2372714*::*real*) (*_2372715*::*real*) (*_2372716*::*real*) (*_2372717*::*real*) *_2372718*::*real*. *vol_y sqrt2 sqrt2 sqrt2* (*sqrt _2372713*) (*sqrt _2372714*) (*sqrt _2372718*))

thm Nonlinear_lemma.vol3_x_sqrt:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *vol3_x_sqrt x1 x2 x3 x4 x5 x6* = *vol_y sqrt2 sqrt2 sqrt2* (*sqrt x1*) (*sqrt x2*) (*sqrt x6*)

thm DEF_gamma3f_126:

gamma3f_126 = (λ(*_2372773*::*real*) (*_2372774*::*real*) (*_2372775*::*real*) (*_2372776*::*real*) (*_2372777*::*real*) *_2372778*::*real*. *gamma3f _2372773 _2372774 _2372778 sqrt2*)

thm Nonlinear_lemma.gamma3f_126:

∀ (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y1*::*real*) (*y2*::*real*) (*y6*::*real*) *f*::*real* ⇒ *real*. *gamma3f_126 y1 y2 y3 y4 y5 y6 f* = *gamma3f y1 y2 y6 sqrt2 f*

thm DEF_gamma3f_135:

gamma3f_135 = (λ(*_2372850*::*real*) (*_2372851*::*real*) (*_2372852*::*real*) (*_2372853*::*real*) (*_2372854*::*real*) *_2372855*::*real*. *gamma3f _2372850 _2372852 _2372854 sqrt2*)

thm Nonlinear_lemma.gamma3f_135:

∀ (*y2*::*real*) (*y4*::*real*) (*y6*::*real*) (*y1*::*real*) (*y3*::*real*) (*y5*::*real*) *f*::*real* ⇒ *real*. *gamma3f_135 y1 y2 y3 y4 y5 y6 f* = *gamma3f y1 y3 y5 sqrt2 f*

thm DEF_gamma3f_vLR:

gamma3f_vLR = (λ(*_2372927*::*real*) (*_2372928*::*real*) (*_2372929*::*real*) (*_2372930*::*real*) (*_2372931*::*real*) (*_2372932*::*real*) *_2372933*::*real* ⇒ *real*. (*dih_y _2372927 _2372928 _2372929 _2372930 _2372931 _2372932* − *dih_y _2372927 _2372928 sqrt2 sqrt2 sqrt2 _2372932* − *dih_y _2372927 _2372929 sqrt2 sqrt2 sqrt2 _2372931*) ∗ ((*vol2r _2372927 sqrt2* − *vol2f _2372927 sqrt2 _2372933*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Nonlin_def.gamma3f_vLR:

∀ (*y4*::*real*) (*y2*::*real*) (*y6*::*real*) (*y3*::*real*) (*y5*::*real*) (*y1*::*real*) *f*::*real* ⇒ *real*. *gamma3f_vLR y1 y2 y3 y4 y5 y6 f* = (*dih_y y1 y2 y3 y4 y5 y6* − *dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6* − *dih_y y1 y3 sqrt2 sqrt2 sqrt2 y5*) ∗ ((*vol2r y1 sqrt2* − *vol2f y1 sqrt2 f*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))

thm DEF_gamma3f_vL:

gamma3f_vL = (λ(*_2373004*::*real*) (*_2373005*::*real*) (*_2373006*::*real*) (*_2373007*::*real*) (*_2373008*::*real*) (*_2373009*::*real*) *_2373010*::*real* ⇒ *real*. (*dih_y _2373004 _2373005 _2373006 _2373007 _2373008 _2373009* − *dih_y _2373004 _2373005 sqrt2 sqrt2 sqrt2 _2373009* − *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r _2373004 sqrt2* − *vol2f _2373004 sqrt2 _2373010*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Nonlin_def.gamma3f_vL:

$\forall$ (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y2*::*real*) (*y6*::*real*) (*y1*::*real*) *f*::*real* ⇒ *real*. *gamma3f_vL y1 y2 y3 y4 y5 y6 f* = (*dih_y y1 y2 y3 y4 y5 y6* − *dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6* − *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r y1 sqrt2* − *vol2f y1 sqrt2 f*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))

thm DEF_gamma3f_v:

gamma3f_v = (λ(*_2373081*::*real*) (*_2373082*::*real*) (*_2373083*::*real*) (*_2373084*::*real*) (*_2373085*::*real*) (*_2373086*::*real*) *_2373087*::*real* ⇒ *real*. (*dih_y _2373081 _2373082 _2373083 _2373084 _2373085 _2373086* − *real_of_nat* (*2*::*nat*) ∗ *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r _2373081 sqrt2* − *vol2f _2373081 sqrt2 _2373087*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Nonlinear_lemma.gamma3f_v:

$\forall$ (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*y1*::*real*) *f*::*real* ⇒ *real*. *gamma3f_v y1 y2 y3 y4 y5 y6 f* = (*dih_y y1 y2 y3 y4 y5 y6* − *real_of_nat* (*2*::*nat*) ∗ *DECIMAL* (*3*::*nat*) (*100*::*nat*)) ∗ ((*vol2r y1 sqrt2* − *vol2f y1 sqrt2 f*) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))

thm DEF_gamma3f_vLR0:

gamma3f_vLR0 = (λ(*_2373158*::*real*) (*_2373159*::*real*) (*_2373160*::*real*) (*_2373161*::*real*) (*_2373162*::*real*) *_2373163*::*real*. (*dih_y _2373158 _2373159 _2373160 _2373161 _2373162 _2373163* − *dih_y _2373158 _2373159 sqrt2 sqrt2 sqrt2 _2373163* − *dih_y _2373158 _2373160 sqrt2 sqrt2 sqrt2 _2373162*) ∗ ((*vol2r _2373158 sqrt2* − *real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*pi* ∗ ((*1*::*real*) − *_2373158* / (*sqrt2* ∗ *real_of_nat* (*2*::*nat*)))))) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Nonlin_def.gamma3f_vLR0:

$\forall$ (*y4*::*real*) (*y2*::*real*) (*y6*::*real*) (*y3*::*real*) (*y5*::*real*) *y1*::*real*. *gamma3f_vLR0 y1 y2 y3 y4 y5 y6* = (*dih_y y1 y2 y3 y4 y5 y6* − *dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6* − *dih_y y1 y3 sqrt2 sqrt2 sqrt2 y5*) ∗ ((*vol2r y1 sqrt2* − *real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*pi* ∗ ((*1*::*real*) − *y1* / (*sqrt2* ∗ *real_of_nat* (*2*::*nat*)))))) / (*real_of_nat* (*2*::*nat*) ∗ *pi*))

thm DEF_gamma3f_vLR_lfun:

gamma3f_vLR_lfun = (λ(*_2373218*::*real*) (*_2373219*::*real*) (*_2373220*::*real*) (*_2373221*::*real*) (*_2373222*::*real*) *_2373223*::*real*. (*dih_y _2373218 _2373219 _2373220 _2373221 _2373222 _2373223* − *dih_y _2373218 _2373219 sqrt2 sqrt2 sqrt2 _2373223* − *dih_y _2373218 _2373220 sqrt2 sqrt2 sqrt2 _2373222*) ∗ ((*vol2r _2373218 sqrt2* − (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*pi* ∗ ((*1*::*real*) − *_2373218* / (*sqrt2* ∗ *real_of_nat* (*2*::*nat*))))) − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*pi* ∗ *lfun* (*_2373218* / *real_of_nat* (*2*::*nat*)))))) / (*real_of_nat* (*2*::*nat*) ∗ *pi*)))

thm Nonlin_def.gamma3f_vLR_lfun:

∀ (y4::real) (y2::real) (y6::real) (y3::real) (y5::real) y1::real. gamma3f_vLR_lfun y1 y2 y3 y4 y5 y6 = (dih_y y1 y2 y3 y4 y5 y6 − dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − dih_y y1 y3 sqrt2 sqrt2 sqrt2 y5) ∗ ((vol2r y1 sqrt2 − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ ((1::real) − y1 / (sqrt2 ∗ real_of_nat (2::nat))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ lfun (y1 / real_of_nat (2::nat)))))) / (real_of_nat (2::nat) ∗ pi))

thm DEF_gamma3f_vL0:

gamma3f_vL0 = (λ(_2373278::real) (_2373279::real) (_2373280::real) (_2373281::real) (_2373282::real) _2373283::real. (dih_y _2373278 _2373279 _2373280 _2373281 _2373282 _2373283 − dih_y _2373278 _2373279 sqrt2 sqrt2 sqrt2 _2373283 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r _2373278 sqrt2 − real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ ((1::real) − _2373278 / (sqrt2 ∗ real_of_nat (2::nat)))))) / (real_of_nat (2::nat) ∗ pi)))

thm Nonlin_def.gamma3f_vL0:

∀ (y3::real) (y4::real) (y5::real) (y2::real) (y6::real) y1::real. gamma3f_vL0 y1 y2 y3 y4 y5 y6 = (dih_y y1 y2 y3 y4 y5 y6 − dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r y1 sqrt2 − real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ ((1::real) − y1 / (sqrt2 ∗ real_of_nat (2::nat)))))) / (real_of_nat (2::nat) ∗ pi))

thm DEF_gamma3f_vL_lfun:

gamma3f_vL_lfun = (λ(_2373338::real) (_2373339::real) (_2373340::real) (_2373341::real) (_2373342::real) _2373343::real. (dih_y _2373338 _2373339 _2373340 _2373341 _2373342 _2373343 − dih_y _2373338 _2373339 sqrt2 sqrt2 sqrt2 _2373343 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r _2373338 sqrt2 − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ ((1::real) − _2373338 / (sqrt2 ∗ real_of_nat (2::nat))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ lfun (_2373338 / real_of_nat (2::nat)))))) / (real_of_nat (2::nat) ∗ pi)))

thm Nonlin_def.gamma3f_vL_lfun:

∀ (y3::real) (y4::real) (y5::real) (y2::real) (y6::real) y1::real. gamma3f_vL_lfun y1 y2 y3 y4 y5 y6 = (dih_y y1 y2 y3 y4 y5 y6 − dih_y y1 y2 sqrt2 sqrt2 sqrt2 y6 − DECIMAL (3::nat) (100::nat)) ∗ ((vol2r y1 sqrt2 − (real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ ((1::real) − y1 / (sqrt2 ∗ real_of_nat (2::nat))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (real_of_nat (2::nat) ∗ (pi ∗ lfun (y1 / real_of_nat (2::nat)))))) / (real_of_nat (2::nat) ∗ pi))

thm DEF_gamma3f_v0:

gamma3f_v0 = (λ(_2373398::real) (_2373399::real) (_2373400::real) (_2373401::real) (_2373402::real) _2373403::real. (dih_y _2373398 _2373399 _2373400 _2373401 _2373402 _2373403 − real_of_nat (2::nat) ∗ DECIMAL (3::nat) (100::nat)) ∗ ((vol2r _2373398 sqrt2 − real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat

(2::nat) * (pi * ((1::real) − _2373398 / (sqrt2 * real_of_nat (2::nat)))))) / (real_of_nat (2::nat) * pi)))

thm Nonlin_def.gamma3f_v0:

∀ (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) y1::real. gamma3f_v0 y1 y2 y3 y4 y5 y6 = (dih_y y1 y2 y3 y4 y5 y6 − real_of_nat (2::nat) * DECIMAL (3::nat) (100::nat)) * ((vol2r y1 sqrt2 − real_of_nat (2::nat) * (mm1 / pi) * (real_of_nat (2::nat) * (pi * ((1::real) − y1 / (sqrt2 * real_of_nat (2::nat)))))) / (real_of_nat (2::nat) * pi))

thm DEF_gamma3f_v_lfun:

gamma3f_v_lfun = (λ(_2373458::real) (_2373459::real) (_2373460::real) (_2373461::real) (_2373462::real) _2373463::real. (dih_y _2373458 _2373459 _2373460 _2373461 _2373462 _2373463 − real_of_nat (2::nat) * DECIMAL (3::nat) (100::nat)) * ((vol2r _2373458 sqrt2 − (real_of_nat (2::nat) * (mm1 / pi) * (real_of_nat (2::nat) * (pi * ((1::real) − _2373458 / (sqrt2 * real_of_nat (2::nat))))) − real_of_nat (8::nat) * (mm2 / pi) * (real_of_nat (2::nat) * (pi * lfun (_2373458 / real_of_nat (2::nat)))))) / (real_of_nat (2::nat) * pi)))

thm Nonlin_def.gamma3f_v_lfun:

∀ (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) y1::real. gamma3f_v_lfun y1 y2 y3 y4 y5 y6 = (dih_y y1 y2 y3 y4 y5 y6 − real_of_nat (2::nat) * DECIMAL (3::nat) (100::nat)) * ((vol2r y1 sqrt2 − (real_of_nat (2::nat) * (mm1 / pi) * (real_of_nat (2::nat) * (pi * ((1::real) − y1 / (sqrt2 * real_of_nat (2::nat))))) − real_of_nat (8::nat) * (mm2 / pi) * (real_of_nat (2::nat) * (pi * lfun (y1 / real_of_nat (2::nat)))))) / (real_of_nat (2::nat) * pi))

thm DEF_dih_x_126_s2:

dih_x_126_s2 = (λ(_2373518::real) (_2373519::real) (_2373520::real) (_2373521::real) (_2373522::real) _2373523::real. dih_y (sqrt _2373518) (sqrt _2373519) sqrt2 sqrt2 sqrt2 (sqrt _2373523))

thm Nonlinear_lemma.dih_x_126_s2:

∀ (x3::real) (x4::real) (x5::real) (x1::real) (x2::real) x6::real. dih_x_126_s2 x1 x2 x3 x4 x5 x6 = dih_y (sqrt x1) (sqrt x2) sqrt2 sqrt2 sqrt2 (sqrt x6)

thm DEF_dih2_x_126_s2:

dih2_x_126_s2 = (λ(_2373578::real) (_2373579::real) (_2373580::real) (_2373581::real) (_2373582::real) _2373583::real. dih2_y (sqrt _2373578) (sqrt _2373579) sqrt2 sqrt2 sqrt2 (sqrt _2373583))

thm Nonlinear_lemma.dih2_x_126_s2:

∀ (x3::real) (x4::real) (x5::real) (x1::real) (x2::real) x6::real. dih2_x_126_s2 x1 x2 x3 x4 x5 x6 = dih2_y (sqrt x1) (sqrt x2) sqrt2 sqrt2 sqrt2 (sqrt x6)

thm DEF_dih3_x_126_s2:

dih3_x_126_s2 = (λ(*_2373638*::*real*) (*_2373639*::*real*) (*_2373640*::*real*) (*_2373641*::*real*) (*_2373642*::*real*) *_2373643*::*real*. *dih3_y* (*sqrt _2373638*) (*sqrt _2373639*) *sqrt2* *sqrt2 sqrt2* (*sqrt _2373643*))

thm Nonlin_def.dih3_x_126_s2:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *dih3_x_126_s2 x1* *x2 x3 x4 x5 x6* = *dih3_y* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm DEF_dih4_x_126_s2:

dih4_x_126_s2 = (λ(*_2373698*::*real*) (*_2373699*::*real*) (*_2373700*::*real*) (*_2373701*::*real*) (*_2373702*::*real*) *_2373703*::*real*. *dih4_y* (*sqrt _2373698*) (*sqrt _2373699*) *sqrt2* *sqrt2 sqrt2* (*sqrt _2373703*))

thm Nonlin_def.dih4_x_126_s2:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *dih4_x_126_s2 x1* *x2 x3 x4 x5 x6* = *dih4_y* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm DEF_dih5_x_126_s2:

dih5_x_126_s2 = (λ(*_2373758*::*real*) (*_2373759*::*real*) (*_2373760*::*real*) (*_2373761*::*real*) (*_2373762*::*real*) *_2373763*::*real*. *dih5_y* (*sqrt _2373758*) (*sqrt _2373759*) *sqrt2* *sqrt2 sqrt2* (*sqrt _2373763*))

thm Nonlin_def.dih5_x_126_s2:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *dih5_x_126_s2 x1* *x2 x3 x4 x5 x6* = *dih5_y* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm DEF_dih6_x_126_s2:

dih6_x_126_s2 = (λ(*_2373818*::*real*) (*_2373819*::*real*) (*_2373820*::*real*) (*_2373821*::*real*) (*_2373822*::*real*) *_2373823*::*real*. *dih6_y* (*sqrt _2373818*) (*sqrt _2373819*) *sqrt2* *sqrt2 sqrt2* (*sqrt _2373823*))

thm Nonlin_def.dih6_x_126_s2:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *dih6_x_126_s2 x1* *x2 x3 x4 x5 x6* = *dih6_y* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm DEF_ldih_x_126_s2:

ldih_x_126_s2 = (λ(*_2373878*::*real*) (*_2373879*::*real*) (*_2373880*::*real*) (*_2373881*::*real*) (*_2373882*::*real*) *_2373883*::*real*. *lfun* (*sqrt _2373878* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih_x_126_s2 _2373878 _2373879 _2373880 _2373881 _2373882* *_2373883*)

thm Nonlin_def.ldih_x_126_s2:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih_x_126_s2 x1* *x2 x3 x4 x5 x6* = *lfun* (*sqrt x1* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih_x_126_s2* *x1 x2 x3 x4 x5 x6*

thm DEF_ldih2_x_126_s2:

ldih2_x_126_s2 = (λ(*_2373938*::*real*) (*_2373939*::*real*) (*_2373940*::*real*) (*_2373941*::*real*) (*_2373942*::*real*) *_2373943*::*real*. *lfun* (*sqrt _2373939* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih2_x_126_s2 _2373938 _2373939 _2373940 _2373941 _2373942 _2373943*)

thm Nonlin_def.ldih2_x_126_s2:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih2_x_126_s2 x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x2* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih2_x_126_s2 x1 x2 x3 x4 x5 x6*

thm DEF_ldih6_x_126_s2:

ldih6_x_126_s2 = (λ(*_2373998*::*real*) (*_2373999*::*real*) (*_2374000*::*real*) (*_2374001*::*real*) (*_2374002*::*real*) *_2374003*::*real*. *lfun* (*sqrt _2374003* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih6_x_126_s2 _2373998 _2373999 _2374000 _2374001 _2374002 _2374003*)

thm Nonlinear_lemma.ldih6_x_126_s2:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih6_x_126_s2 x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x6* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih6_x_126_s2 x1 x2 x3 x4 x5 x6*

thm DEF_dih_x_135_s2:

dih_x_135_s2 = (λ(*_2374058*::*real*) (*_2374059*::*real*) (*_2374060*::*real*) (*_2374061*::*real*) (*_2374062*::*real*) *_2374063*::*real*. *dih_y* (*sqrt _2374058*) *sqrt2* (*sqrt _2374060*) *sqrt2* (*sqrt _2374062*) *sqrt2*)

thm Nonlinear_lemma.dih_x_135_s2:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *dih_x_135_s2 x1 x2 x3 x4 x5 x6* = *dih_y* (*sqrt x1*) *sqrt2* (*sqrt x3*) *sqrt2* (*sqrt x5*) *sqrt2*

thm DEF_dih2_x_135_s2:

dih2_x_135_s2 = (λ(*_2374118*::*real*) (*_2374119*::*real*) (*_2374120*::*real*) (*_2374121*::*real*) (*_2374122*::*real*) *_2374123*::*real*. *dih2_y* (*sqrt _2374118*) *sqrt2* (*sqrt _2374120*) *sqrt2* (*sqrt _2374122*) *sqrt2*)

thm Nonlin_def.dih2_x_135_s2:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *dih2_x_135_s2 x1 x2 x3 x4 x5 x6* = *dih2_y* (*sqrt x1*) *sqrt2* (*sqrt x3*) *sqrt2* (*sqrt x5*) *sqrt2*

thm DEF_dih3_x_135_s2:

dih3_x_135_s2 = (λ(*_2374178*::*real*) (*_2374179*::*real*) (*_2374180*::*real*) (*_2374181*::*real*) (*_2374182*::*real*) *_2374183*::*real*. *dih3_y* (*sqrt _2374178*) *sqrt2* (*sqrt _2374180*) *sqrt2* (*sqrt _2374182*) *sqrt2*)

thm Nonlin_def.dih3_x_135_s2:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *dih3_x_135_s2 x1 x2 x3 x4 x5 x6* = *dih3_y* (*sqrt x1*) *sqrt2* (*sqrt x3*) *sqrt2* (*sqrt x5*) *sqrt2*

thm DEF_dih4_x_135_s2:

$dih4\_x\_135\_s2 = (\lambda(\_2374238::real)\ (\_2374239::real)\ (\_2374240::real)\ (\_2374241::real)\ (\_2374242::real)\ \_2374243::real.\ dih4\_y\ (sqrt\ \_2374238)\ sqrt2\ (sqrt\ \_2374240)\ sqrt2\ (sqrt\ \_2374242)\ sqrt2)$

thm Nonlinear_lemma.dih4_x_135_s2:

$\forall (x2::real)\ (x4::real)\ (x6::real)\ (x1::real)\ (x3::real)\ x5::real.\ dih4\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih4\_y\ (sqrt\ x1)\ sqrt2\ (sqrt\ x3)\ sqrt2\ (sqrt\ x5)\ sqrt2$

thm DEF_dih5_x_135_s2:

$dih5\_x\_135\_s2 = (\lambda(\_2374298::real)\ (\_2374299::real)\ (\_2374300::real)\ (\_2374301::real)\ (\_2374302::real)\ \_2374303::real.\ dih5\_y\ (sqrt\ \_2374298)\ sqrt2\ (sqrt\ \_2374300)\ sqrt2\ (sqrt\ \_2374302)\ sqrt2)$

thm Nonlin_def.dih5_x_135_s2:

$\forall (x2::real)\ (x4::real)\ (x6::real)\ (x1::real)\ (x3::real)\ x5::real.\ dih5\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih5\_y\ (sqrt\ x1)\ sqrt2\ (sqrt\ x3)\ sqrt2\ (sqrt\ x5)\ sqrt2$

thm DEF_dih6_x_135_s2:

$dih6\_x\_135\_s2 = (\lambda(\_2374358::real)\ (\_2374359::real)\ (\_2374360::real)\ (\_2374361::real)\ (\_2374362::real)\ \_2374363::real.\ dih6\_y\ (sqrt\ \_2374358)\ sqrt2\ (sqrt\ \_2374360)\ sqrt2\ (sqrt\ \_2374362)\ sqrt2)$

thm Nonlin_def.dih6_x_135_s2:

$\forall (x2::real)\ (x4::real)\ (x6::real)\ (x1::real)\ (x3::real)\ x5::real.\ dih6\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6\ =\ dih6\_y\ (sqrt\ x1)\ sqrt2\ (sqrt\ x3)\ sqrt2\ (sqrt\ x5)\ sqrt2$

thm DEF_ldih_x_135_s2:

$ldih\_x\_135\_s2 = (\lambda(\_2374418::real)\ (\_2374419::real)\ (\_2374420::real)\ (\_2374421::real)\ (\_2374422::real)\ \_2374423::real.\ lfun\ (sqrt\ \_2374418\ /\ DECIMAL\ (20::nat)\ (10::nat)) * dih\_y\ (sqrt\ \_2374418)\ sqrt2\ (sqrt\ \_2374420)\ sqrt2\ (sqrt\ \_2374422)\ sqrt2)$

thm Nonlin_def.ldih_x_135_s2′:

$\forall (x2::real)\ (x4::real)\ (x6::real)\ (x1::real)\ (x3::real)\ x5::real.\ ldih\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6\ =\ lfun\ (sqrt\ x1\ /\ DECIMAL\ (20::nat)\ (10::nat)) * dih\_y\ (sqrt\ x1)\ sqrt2\ (sqrt\ x3)\ sqrt2\ (sqrt\ x5)\ sqrt2$

thm DEF_ldih3_x_135_s2:

$ldih3\_x\_135\_s2 = (\lambda(\_2374478::real)\ (\_2374479::real)\ (\_2374480::real)\ (\_2374481::real)\ (\_2374482::real)\ \_2374483::real.\ lfun\ (sqrt\ \_2374480\ /\ DECIMAL\ (20::nat)\ (10::nat)) * dih3\_x\_135\_s2\ \_2374478\ \_2374479\ \_2374480\ \_2374481\ \_2374482\ \_2374483)$

thm Nonlinear_lemma.ldih3_x_135_s2:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih3_x_135_s2 x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x3* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih3_x_135_s2 x1 x2 x3 x4 x5 x6*

thm DEF_ldih5_x_135_s2:

ldih5_x_135_s2 = (λ(*_2374538*::*real*) (*_2374539*::*real*) (*_2374540*::*real*) (*_2374541*::*real*) (*_2374542*::*real*) *_2374543*::*real*. *lfun* (*sqrt _2374542* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih5_x_135_s2 _2374538 _2374539 _2374540 _2374541 _2374542 _2374543*)

thm Nonlin_def.ldih5_x_135_s2:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ldih5_x_135_s2 x1 x2 x3 x4 x5 x6* = *lfun* (*sqrt x5* / *DECIMAL* (*20*::*nat*) (*10*::*nat*)) ∗ *dih5_x_135_s2 x1 x2 x3 x4 x5 x6*

thm DEF_delta_x_126_s2:

delta_x_126_s2 = (λ(*_2374598*::*real*) (*_2374599*::*real*) (*_2374600*::*real*) (*_2374601*::*real*) (*_2374602*::*real*) *_2374603*::*real*. *delta_y* (*sqrt _2374598*) (*sqrt _2374599*) *sqrt2 sqrt2 sqrt2* (*sqrt _2374603*))

thm Nonlinear_lemma.delta_x_126_s2:

∀ (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x1*::*real*) (*x2*::*real*) *x6*::*real*. *delta_x_126_s2 x1 x2 x3 x4 x5 x6* = *delta_y* (*sqrt x1*) (*sqrt x2*) *sqrt2 sqrt2 sqrt2* (*sqrt x6*)

thm DEF_delta_x_135_s2:

delta_x_135_s2 = (λ(*_2374658*::*real*) (*_2374659*::*real*) (*_2374660*::*real*) (*_2374661*::*real*) (*_2374662*::*real*) *_2374663*::*real*. *delta_y* (*sqrt _2374658*) *sqrt2* (*sqrt _2374660*) *sqrt2* (*sqrt _2374662*) *sqrt2*)

thm Nonlin_def.delta_x_135_s2:

∀ (*x2*::*real*) (*x4*::*real*) (*x6*::*real*) (*x1*::*real*) (*x3*::*real*) *x5*::*real*. *delta_x_135_s2 x1 x2 x3 x4 x5 x6* = *delta_y* (*sqrt x1*) *sqrt2* (*sqrt x3*) *sqrt2* (*sqrt x5*) *sqrt2*

thm DEF_gamma3f_x_vLR_lfun:

gamma3f_x_vLR_lfun = (λ(*_2374718*::*real*) (*_2374719*::*real*) (*_2374720*::*real*) (*_2374721*::*real*) (*_2374722*::*real*) *_2374723*::*real*. *gamma3f_vLR_lfun* (*sqrt _2374718*) (*sqrt _2374719*) (*sqrt _2374720*) (*sqrt _2374721*) (*sqrt _2374722*) (*sqrt _2374723*))

thm Nonlinear_lemma.gamma3f_x_vLR_lfun:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_vLR_lfun x1 x2 x3 x4 x5 x6* = *gamma3f_vLR_lfun* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_x_vLR0:

gamma3f_x_vLR0 = (λ(*_2374778*::*real*) (*_2374779*::*real*) (*_2374780*::*real*) (*_2374781*::*real*) (*_2374782*::*real*) *_2374783*::*real*. *gamma3f_vLR0* (*sqrt _2374778*) (*sqrt _2374779*) (*sqrt _2374780*) (*sqrt _2374781*) (*sqrt _2374782*) (*sqrt _2374783*))

thm Nonlin_def.gamma3f_x_vLR0:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_vLR0 x1 x2 x3 x4 x5 x6* = *gamma3f_vLR0* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_x_vL_lfun:

gamma3f_x_vL_lfun = (λ(*_2374838*::*real*) (*_2374839*::*real*) (*_2374840*::*real*) (*_2374841*::*real*) (*_2374842*::*real*) *_2374843*::*real*. *gamma3f_vL_lfun* (*sqrt _2374838*) (*sqrt _2374839*) (*sqrt _2374840*) (*sqrt _2374841*) (*sqrt _2374842*) (*sqrt _2374843*))

thm Nonlin_def.gamma3f_x_vL_lfun:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_vL_lfun x1 x2 x3 x4 x5 x6* = *gamma3f_vL_lfun* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_x_vL0:

gamma3f_x_vL0 = (λ(*_2374898*::*real*) (*_2374899*::*real*) (*_2374900*::*real*) (*_2374901*::*real*) (*_2374902*::*real*) *_2374903*::*real*. *gamma3f_vL0* (*sqrt _2374898*) (*sqrt _2374899*) (*sqrt _2374900*) (*sqrt _2374901*) (*sqrt _2374902*) (*sqrt _2374903*))

thm Nonlinear_lemma.gamma3f_x_vL0:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_vL0 x1 x2 x3 x4 x5 x6* = *gamma3f_vL0* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_x_v_lfun:

gamma3f_x_v_lfun = (λ(*_2374958*::*real*) (*_2374959*::*real*) (*_2374960*::*real*) (*_2374961*::*real*) (*_2374962*::*real*) *_2374963*::*real*. *gamma3f_v_lfun* (*sqrt _2374958*) (*sqrt _2374959*) (*sqrt _2374960*) (*sqrt _2374961*) (*sqrt _2374962*) (*sqrt _2374963*))

thm Nonlinear_lemma.gamma3f_x_v_lfun:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_v_lfun x1 x2 x3 x4 x5 x6* = *gamma3f_v_lfun* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_gamma3f_x_v0:

gamma3f_x_v0 = (λ(*_2375018*::*real*) (*_2375019*::*real*) (*_2375020*::*real*) (*_2375021*::*real*) (*_2375022*::*real*) *_2375023*::*real*. *gamma3f_v0* (*sqrt _2375018*) (*sqrt _2375019*) (*sqrt _2375020*) (*sqrt _2375021*) (*sqrt _2375022*) (*sqrt _2375023*))

thm Nonlin_def.gamma3f_x_v0:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_v0 x1 x2 x3 x4 x5 x6* = *gamma3f_v0* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*)

thm DEF_vol3_x_135_s2:

$vol3\_x\_135\_s2 = (\lambda(\_2375078::real)\ (\_2375079::real)\ (\_2375080::real)\ (\_2375081::real)\ (\_2375082::real)\ \_2375083::real.\ vol3r\ (sqrt\ \_2375078)\ (sqrt\ \_2375080)\ (sqrt\ \_2375082)\ sqrt2)$

thm Nonlin_def.vol3_x_135_s2:

$\forall (x2::real)\ (x4::real)\ (x6::real)\ (x1::real)\ (x3::real)\ x5::real.\ vol3\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6\ =\ vol3r\ (sqrt\ x1)\ (sqrt\ x3)\ (sqrt\ x5)\ sqrt2$

thm DEF_monomial:

$monomial = (\lambda(\_2375138::nat)\ (\_2375139::nat)\ (\_2375140::nat)\ (\_2375141::nat)\ (\_2375142::nat)\ (\_2375143::nat)\ (\_2375144::real)\ (\_2375145::real)\ (\_2375146::real)\ (\_2375147::real)\ (\_2375148::real)\ \_2375149::real.\ \_2375144^{\_2375138} * (\_2375145^{\_2375139} * (\_2375146^{\_2375140} * (\_2375147^{\_2375141} * (\_2375148^{\_2375142} * \_2375149^{\_2375143})))))$

thm Nonlin_def.monomial:

$\forall (y1::real)\ (n1::nat)\ (y2::real)\ (n2::nat)\ (y3::real)\ (n3::nat)\ (y4::real)\ (n4::nat)\ (y5::real)\ (n5::nat)\ (y6::real)\ n6::nat.\ monomial\ n1\ n2\ n3\ n4\ n5\ n6\ y1\ y2\ y3\ y4\ y5\ y6\ =\ y1^{n1} * (y2^{n2} * (y3^{n3} * (y4^{n4} * (y5^{n5} * y6^{n6}))))$

thm DEF_arclength_x_234:

$arclength\_x\_234 = (\lambda(\_2375330::real)\ (\_2375331::real)\ (\_2375332::real)\ (\_2375333::real)\ (\_2375334::real)\ \_2375335::real.\ arclength\ (sqrt\ \_2375331)\ (sqrt\ \_2375332)\ (sqrt\ \_2375333))$

thm Nonlin_def.arclength_x_234:

$\forall (x1::real)\ (x5::real)\ (x6::real)\ (x2::real)\ (x3::real)\ x4::real.\ arclength\_x\_234\ x1\ x2\ x3\ x4\ x5\ x6\ =\ arclength\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)$

thm DEF_arclength_x_126:

$arclength\_x\_126 = (\lambda(\_2375390::real)\ (\_2375391::real)\ (\_2375392::real)\ (\_2375393::real)\ (\_2375394::real)\ \_2375395::real.\ arclength\ (sqrt\ \_2375390)\ (sqrt\ \_2375391)\ (sqrt\ \_2375395))$

thm Nonlinear_lemma.arclength_x_126:

$\forall (x3::real)\ (x4::real)\ (x5::real)\ (x1::real)\ (x2::real)\ x6::real.\ arclength\_x\_126\ x1\ x2\ x3\ x4\ x5\ x6\ =\ arclength\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x6)$

thm DEF_uni:

$uni = (\lambda(\_2375450::(?'h::type \Rightarrow ?'g::type) \times (?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'h::type))\ (\_2375451::?'f::type)\ (\_2375452::?'e::type)\ (\_2375453::?'d::type)\ (\_2375454::?'c::type)\ (\_2375455::?'b::type)\ \_2375456::?'a::type.\ fst\ \_2375450\ (snd\ \_2375450\ \_2375451\ \_2375452\ \_2375453\ \_2375454\ \_2375455\ \_2375456))$

thm Nonlin_def.uni:

$\forall (f::?'h::type \Rightarrow ?'g::type)\ (x::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow ?'h::type)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)$

$(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ uni $(f,\ x)$ $x1$ $x2$ $x3$ $x4$ $x5$ $x6$ $=$ f $(x\ x1\ x2\ x3\ x4\ x5\ x6)$

thm DEF_constant6:

$constant6 = (\lambda(\_2375537::?'g::type)\ (\_2375538::?'f::type)\ (\_2375539::?'e::type)\ (\_2375540::?'d::type)\ (\_2375541::?'c::type)\ (\_2375542::?'b::type)\ \_2375543::?'a::type.\ \_2375537)$

thm Nonlin_def.constant6:

$\forall\,(x1::?'g::type)\ (x2::?'f::type)\ (x3::?'e::type)\ (x4::?'d::type)\ (x5::?'c::type)\ (x6::?'b::type)\ c::?'a::type.\ constant6\ c\ x1\ x2\ x3\ x4\ x5\ x6\ =\ c$

thm DEF_promote3_to_6:

$promote3\_to\_6 = (\lambda(\_2375614::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type)\ (\_2375615::?'g::type)\ (\_2375616::?'f::type)\ (\_2375617::?'e::type)\ (\_2375618::?'c::type)\ (\_2375619::?'b::type)\ \_2375620::?'a::type.\ \_2375614\ \_2375615\ \_2375616\ \_2375617)$

thm Nonlin_def.promote3_to_6:

$\forall\,(x4::?'g::type)\ (x5::?'f::type)\ (x6::?'e::type)\ (f::?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (x1::?'d::type)\ (x2::?'c::type)\ x3::?'b::type.\ promote3\_to\_6\ f\ x1\ x2\ x3\ x4\ x5\ x6\ =\ f\ x1\ x2\ x3$

thm DEF_promote1_to_6:

$promote1\_to\_6\ =\ (\lambda(\_2375691::?'g::type\ \Rightarrow\ ?'f::type)\ (\_2375692::?'g::type)\ (\_2375693::?'e::type)\ (\_2375694::?'d::type)\ (\_2375695::?'c::type)\ (\_2375696::?'b::type)\ \_2375697::?'a::type.\ \_2375691\ \_2375692)$

thm Nonlin_def.promote1_to_6:

$\forall\,(x2::?'g::type)\ (x3::?'f::type)\ (x4::?'e::type)\ (x5::?'d::type)\ (x6::?'c::type)\ (f::?'b::type \Rightarrow ?'a::type)\ x1::?'b::type.\ promote1\_to\_6\ f\ x1\ x2\ x3\ x4\ x5\ x6\ =\ f\ x1$

thm Functional_equation.rh0:

$rh0\ =\ (1::real)\ /\ (h0\ -\ (1::real))$

thm Nonlin_def.two6:

$two6\ =\ constant6\ (real\_of\_nat\ (2::nat))$

thm Nonlin_def.zero6:

$zero6\ =\ constant6\ (0::real)$

thm Nonlin_def.dummy6:

$dummy6\ =\ constant6\ (0::real)$

thm Nonlin_def.four6:

$four6\ =\ constant6\ (real\_of\_nat\ (4::nat))$

thm DEF_mk_126:

$mk\_126 = (\lambda\_2375768::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ $compose6\ \_2375768\ proj\_x1\ proj\_x2\ two6\ two6\ two6\ proj\_x6)$

thm Functional_equation.mk_126:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ mk\_126\ f = compose6\ f$ $proj\_x1\ proj\_x2\ two6\ two6\ two6\ proj\_x6$

thm DEF_mk_456:

$mk\_456 = (\lambda\_2375773::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ $compose6\ \_2375773\ two6\ two6\ two6\ proj\_x4\ proj\_x5\ proj\_x6)$

thm Functional_equation.mk_456:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ mk\_456\ f = compose6\ f$ $two6\ two6\ two6\ proj\_x4\ proj\_x5\ proj\_x6$

thm DEF_mk_135:

$mk\_135 = (\lambda\_2375778::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ $compose6\ \_2375778\ proj\_x1\ two6\ proj\_x3\ two6\ proj\_x5\ two6)$

thm Functional_equation.mk_135:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ mk\_135\ f = compose6\ f$ $proj\_x1\ two6\ proj\_x3\ two6\ proj\_x5\ two6$

thm DEF_add6:

$add6 = (\lambda(\_2375783::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type$ $\Rightarrow ?'a::type \Rightarrow real)\ (\_2375784::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type$ $\Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (\_2375785::?'f::type)\ (\_2375786::?'e::type)$ $(\_2375787::?'d::type)\ (\_2375788::?'c::type)\ (\_2375789::?'b::type)\ \_2375790::?'a::type.$ $\_2375783\ \_2375785\ \_2375786\ \_2375787\ \_2375788\ \_2375789\ \_2375790 + \_2375784$ $\_2375785\ \_2375786\ \_2375787\ \_2375788\ \_2375789\ \_2375790)$

thm Nonlin_def.add6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type$ $\Rightarrow real)\ (g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow$ $?'a::type \Rightarrow real)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)$ $(x5::?'b::type)\ x6::?'a::type.\ add6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6$ $+\ g\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_scalar6:

$scalar6 = (\lambda(\_2375879::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type$ $\Rightarrow ?'a::type \Rightarrow real)\ (\_2375880::real)\ (\_2375881::?'f::type)\ (\_2375882::?'e::type)$ $(\_2375883::?'d::type)\ (\_2375884::?'c::type)\ (\_2375885::?'b::type)\ \_2375886::?'a::type.$ $\_2375879\ \_2375881\ \_2375882\ \_2375883\ \_2375884\ \_2375885\ \_2375886 * \_2375880)$

thm Nonlin_def.scalar6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $(x6::?'a::type)$ $r::real.$ $scalar6\ f\ r\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 * r$

thm DEF_mul6:

$mul6 = (\lambda(\_2375975::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2375976::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2375977::?'f::type)$ $(\_2375978::?'e::type)$ $(\_2375979::?'d::type)$ $(\_2375980::?'c::type)$ $(\_2375981::?'b::type)$ $\_2375982::?'a::type.$ $\_2375975\ \_2375977\ \_2375978\ \_2375979\ \_2375980\ \_2375981\ \_2375982 * \_2375976\ \_2375977\ \_2375978\ \_2375979\ \_2375980\ \_2375981\ \_2375982)$

thm Nonlin_def.mul6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $mul6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 * g\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_div6:

$div6 = (\lambda(\_2376071::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2376072::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2376073::?'f::type)$ $(\_2376074::?'e::type)$ $(\_2376075::?'d::type)$ $(\_2376076::?'c::type)$ $(\_2376077::?'b::type)$ $\_2376078::?'a::type.$ $\_2376071\ \_2376073\ \_2376074\ \_2376075\ \_2376076\ \_2376077\ \_2376078 / \_2376072\ \_2376073\ \_2376074\ \_2376075\ \_2376076\ \_2376077\ \_2376078)$

thm Nonlin_def.div6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $div6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 / g\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_sub6:

$sub6 = (\lambda(\_2376167::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2376168::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(\_2376169::?'f::type)$ $(\_2376170::?'e::type)$ $(\_2376171::?'d::type)$ $(\_2376172::?'c::type)$ $(\_2376173::?'b::type)$ $\_2376174::?'a::type.$ $\_2376167\ \_2376169\ \_2376170\ \_2376171\ \_2376172\ \_2376173\ \_2376174 - \_2376168\ \_2376169\ \_2376170\ \_2376171\ \_2376172\ \_2376173\ \_2376174)$

thm Nonlin_def.sub6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$

$(x5::?'b::type)$ $x6::?'a::type.$ $sub6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 - g\ x1\ x2\ x3\ x4\ x5\ x6$

thm DEF_proj_y1:

$proj\_y1 = (\lambda(\_2376263::real)\ (\_2376264::?'e::type)\ (\_2376265::?'d::type)\ (\_2376266::?'c::type)\ (\_2376267::?'b::type)\ \_2376268::?'a::type.\ sqrt\ \_2376263)$

thm Nonlin_def.proj_y1:

$\forall (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x1::real.\ proj\_y1\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x1$

thm DEF_proj_y2:

$proj\_y2 = (\lambda(\_2376323::?'e::type)\ (\_2376324::real)\ (\_2376325::?'d::type)\ (\_2376326::?'c::type)\ (\_2376327::?'b::type)\ \_2376328::?'a::type.\ sqrt\ \_2376324)$

thm Nonlin_def.proj_y2:

$\forall (x1::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x2::real.\ proj\_y2\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x2$

thm DEF_proj_y3:

$proj\_y3 = (\lambda(\_2376383::?'e::type)\ (\_2376384::?'d::type)\ (\_2376385::real)\ (\_2376386::?'c::type)\ (\_2376387::?'b::type)\ \_2376388::?'a::type.\ sqrt\ \_2376385)$

thm Nonlin_def.proj_y3:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x3::real.\ proj\_y3\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x3$

thm DEF_proj_y4:

$proj\_y4 = (\lambda(\_2376443::?'e::type)\ (\_2376444::?'d::type)\ (\_2376445::?'c::type)\ (\_2376446::real)\ (\_2376447::?'b::type)\ \_2376448::?'a::type.\ sqrt\ \_2376446)$

thm Nonlin_def.proj_y4:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x4::real.\ proj\_y4\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x4$

thm DEF_proj_y5:

$proj\_y5 = (\lambda(\_2376503::?'e::type)\ (\_2376504::?'d::type)\ (\_2376505::?'c::type)\ (\_2376506::?'b::type)\ (\_2376507::real)\ \_2376508::?'a::type.\ sqrt\ \_2376507)$

thm Nonlin_def.proj_y5:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::?'b::type)\ (x6::?'a::type)\ x5::real.\ proj\_y5\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x5$

thm DEF_proj_y6:

$proj\_y6 = (\lambda(\_2376563::?'e::type)\ (\_2376564::?'d::type)\ (\_2376565::?'c::type)\ (\_2376566::?'b::type)\ \_2376567::?'a::type.\ sqrt)$

thm Nonlin_def.proj_y6:

$\forall$ (*x1*::*?'e*::*type*) (*x2*::*?'d*::*type*) (*x3*::*?'c*::*type*) (*x4*::*?'b*::*type*) (*x5*::*?'a*::*type*) *x6*::*real*. *proj_y6 x1 x2 x3 x4 x5 x6* = *sqrt x6*

thm DEF_domain6:

domain6 = (λ(*_2376623*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *bool*) (*_2376624*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) *_2376625*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*. $\forall$ (*x1*::*?'g*::*type*) (*x2*::*?'f*::*type*) (*x3*::*?'e*::*type*) (*x4*::*?'d*::*type*) (*x5*::*?'c*::*type*) *x6*::*?'b*::*type*. *_2376623 x1 x2 x3 x4 x5 x6* ⟶ *_2376624 x1 x2 x3 x4 x5 x6* = *_2376625 x1 x2 x3 x4 x5 x6*)

thm Nonlin_def.domain6:

$\forall$ (*h*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *bool*) (*f*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) *g*::*?'g*::*type* ⇒ *?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*. *domain6 h f g* = ($\forall$ (*x1*::*?'g*::*type*) (*x2*::*?'f*::*type*) (*x3*::*?'e*::*type*) (*x4*::*?'d*::*type*) (*x5*::*?'c*::*type*) *x6*::*?'b*::*type*. *h x1 x2 x3 x4 x5 x6* ⟶ *f x1 x2 x3 x4 x5 x6* = *g x1 x2 x3 x4 x5 x6*)

thm DEF_rotate234:

rotate234 = (λ*_2376644*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *compose6 _2376644 proj_x2 proj_x3 proj_x4 unit6 unit6 unit6*)

thm Nonlin_def.rotate234:

$\forall$ *f*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *rotate234 f* = *compose6 f proj_x2 proj_x3 proj_x4 unit6 unit6 unit6*

thm DEF_rotate126:

rotate126 = (λ*_2376649*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *compose6 _2376649 proj_x1 proj_x2 proj_x6 unit6 unit6 unit6*)

thm Nonlin_def.rotate126:

$\forall$ *f*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *rotate126 f* = *compose6 f proj_x1 proj_x2 proj_x6 unit6 unit6 unit6*

thm DEF_rotate345:

rotate345 = (λ*_2376654*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *compose6 _2376654 proj_x3 proj_x4 proj_x5 unit6 unit6 unit6*)

thm Nonlin_def.rotate345:

$\forall$ *f*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *rotate345 f* = *compose6 f proj_x3 proj_x4 proj_x5 unit6 unit6 unit6*

thm Nonlin_def.x1cube:

x1cube = mul6 proj_x1 (mul6 proj_x1 proj_x1)

thm DEF_truncate_sqrt:

truncate_sqrt = (λ(_2376659::real) _2376660::real. if _2376660 ≤ _2376659 then sqrt _2376659 else sqrt _2376660)

thm Functional_equation.truncate_sqrt:

∀ (c::real) x::real. truncate_sqrt c x = (if x ≤ c then sqrt c else sqrt x)

thm DEF_truncate_dih_x:

truncate_dih_x = (λ(_2376671::real) (_2376672::real) (_2376673::real) (_2376674::real) (_2376675::real) (_2376676::real) _2376677::real. LET (λd_x4::real. LET_END (LET (λd::real. LET_END (pi / real_of_nat (2::nat) + atn2 (sqrt (real_of_nat (4::nat) ∗ _2376672) ∗ truncate_sqrt _2376671 d, − d_x4))) (delta_x _2376672 _2376673 _2376674 _2376675 _2376676 _2376677))) (delta_x4 _2376672 _2376673 _2376674 _2376675 _2376676 _2376677))

thm Nonlin_def.truncate_dih_x:

∀ (c::real) (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. truncate_dih_x c x1 x2 x3 x4 x5 x6 = LET (λd_x4::real. LET_END (LET (λd::real. LET_END (pi / real_of_nat (2::nat) + atn2 (sqrt (real_of_nat (4::nat) ∗ x1) ∗ truncate_sqrt c d, − d_x4))) (delta_x x1 x2 x3 x4 x5 x6))) (delta_x4 x1 x2 x3 x4 x5 x6)

thm DEF_truncate_sol_x:

truncate_sol_x = (λ_2376748::real. add6 (truncate_dih_x _2376748) (add6 (rotate2 (truncate_dih_x _2376748)) (sub6 (rotate3 (truncate_dih_x _2376748)) (constant6 pi))))

thm Functional_equation.truncate_sol_x:

∀ c::real. truncate_sol_x c = add6 (truncate_dih_x c) (add6 (rotate2 (truncate_dih_x c)) (sub6 (rotate3 (truncate_dih_x c)) (constant6 pi)))

thm DEF_truncate_vol_x:

truncate_vol_x = (λ_2376753::real. scalar6 (uni (truncate_sqrt _2376753, delta_x)) ((1::real) / real_of_nat (12::nat)))

thm Nonlin_def.truncate_vol_x:

∀ c::real. truncate_vol_x c = scalar6 (uni (truncate_sqrt c, delta_x)) ((1::real) / real_of_nat (12::nat))

thm DEF_truncate_vol3r_456:

truncate_vol3r_456 = (λ_2376758::real. mk_456 (truncate_vol_x _2376758))

thm Functional_equation.truncate_vol3r_456:

∀ c::real. truncate_vol3r_456 c = mk_456 (truncate_vol_x c)

thm DEF_truncate_vol3f:

truncate_vol3f = (λ(*_2376763*::*real*) (*_2376764*::*real*) (*_2376765*::*real*) *_2376766*::*real*. *sub6* (*scalar6* (*add6* (*mk_456* (*rotate5* (*truncate_sol_x _2376763*))) (*add6* (*mk_456* (*rotate6* (*truncate_sol_x _2376763*))) (*mk_456* (*rotate4* (*truncate_sol_x _2376763*))))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2376764*) (*mk_456* (*rotate4* (*truncate_dih_x _2376763*)))) (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2376765*) (*mk_456* (*rotate5* (*truncate_dih_x _2376763*)))) (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2376766*) (*mk_456* (*rotate6* (*truncate_dih_x _2376763*)))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*))))

thm Functional_equation.truncate_vol3f:

∀ (*m4*::*real*) (*m5*::*real*) (*m6*::*real*) *c*::*real*. *truncate_vol3f c m4 m5 m6* = *sub6* (*scalar6* (*add6* (*mk_456* (*rotate5* (*truncate_sol_x c*))) (*add6* (*mk_456* (*rotate6* (*truncate_sol_x c*))) (*mk_456* (*rotate4* (*truncate_sol_x c*))))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m4*) (*mk_456* (*rotate4* (*truncate_dih_x c*)))) (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m5*) (*mk_456* (*rotate5* (*truncate_dih_x c*)))) (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6 proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m6*) (*mk_456* (*rotate6* (*truncate_dih_x c*)))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*)))

thm DEF_truncate_gamma2_x:

truncate_gamma2_x = (λ(*_2376795*::*real*) *_2376796*::*real*. (*real_of_nat* (*8*::*nat*) − *_2376796*) ∗ (*sqrt _2376796* / *real_of_nat* (*24*::*nat*)) − (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ ((*1*::*real*) − *sqrt _2376796* / *sqrt8*) − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*_2376795* ∗ *lfun* (*sqrt _2376796* / *real_of_nat* (*2*::*nat*)))))

thm Functional_equation.truncate_gamma2_x:

∀ (*m*::*real*) *x*::*real*. *truncate_gamma2_x m x* = (*real_of_nat* (*8*::*nat*) − *x*) ∗ (*sqrt x* / *real_of_nat* (*24*::*nat*)) − (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ ((*1*::*real*) − *sqrt x* / *sqrt8*) − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*m* ∗ *lfun* (*sqrt x* / *real_of_nat* (*2*::*nat*))))

thm DEF_truncate_gamma3f_x:

truncate_gamma3f_x = (λ(*_2376807*::*real*) (*_2376808*::*real*) (*_2376809*::*real*) *_2376810*::*real*. *sub6* (*truncate_vol3r_456 _2376807*) (*truncate_vol3f _2376807 _2376808 _2376809 _2376810*))

thm Nonlin_def.truncate_gamma3f_x:

∀ (*d*::*real*) (*m4*::*real*) (*m5*::*real*) *m6*::*real*. *truncate_gamma3f_x d m4 m5 m6* = *sub6* (*truncate_vol3r_456 d*) (*truncate_vol3f d m4 m5 m6*)

thm DEF_truncate_gamma23_x:

truncate_gamma23_x = (λ(*_2376839*::*real*) (*_2376840*::*real*) (*_2376841*::*real*) (*_2376842*::*real*) (*_2376843*::*real*) (*_2376844*::*real*) *_2376845*::*real*. *add6* (*scalar6* (*compose6* (*truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *_2376841* *_2376842* *_2376845*) *dummy6* *dummy6* *dummy6* *proj_x1* *proj_x2* *proj_x6*) *_2376839*) (*add6* (*scalar6* (*compose6* (*truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *_2376841* *_2376843* *_2376844*) *dummy6* *dummy6* *dummy6* *proj_x1* *proj_x3* *proj_x5*) *_2376840*) (*mul6* (*sub6* *dih_x* (*add6* (*mk_126* (*truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)))) (*mk_135* (*truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)))))) (*uni* (*truncate_gamma2_x* *_2376841*, *proj_x1*)))))

thm Functional_equation.truncate_gamma23_x:

∀ (*m2*::*real*) (*m6*::*real*) (*iw1*::*real*) (*m3*::*real*) (*m5*::*real*) (*iw2*::*real*) *m1*::*real*. *truncate_gamma23_x* *iw1* *iw2* *m1* *m2* *m3* *m5* *m6* = *add6* (*scalar6* (*compose6* (*truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *m1* *m2* *m6*) *dummy6* *dummy6* *dummy6* *proj_x1* *proj_x2* *proj_x6*) *iw1*) (*add6* (*scalar6* (*compose6* (*truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *m1* *m3* *m5*) *dummy6* *dummy6* *dummy6* *proj_x1* *proj_x3* *proj_x5*) *iw2*) (*mul6* (*sub6* *dih_x* (*add6* (*mk_126* (*truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)))) (*mk_135* (*truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)))))) (*uni* (*truncate_gamma2_x* *m1*, *proj_x1*))))

thm DEF_truncate_gamma23a_x:

truncate_gamma23a_x = (λ(*_2376916*::*real*) (*_2376917*::*real*) (*_2376918*::*real*) (*_2376919*::*real*) (*_2376920*::*real*) (*_2376921*::*real*) (*_2376922*::*real*) (*_2376923*::*real*) (*_2376924*::*real*) (*_2376925*::*real*) (*_2376926*::*real*) (*_2376927*::*real*) *_2376928*::*real*. *truncate_gamma3f_x* (*0*::*real*) *_2376918* *_2376919* *_2376922* (*0*::*real*) (*0*::*real*) (*0*::*real*) *_2376923* *_2376924* *_2376928* ∗ *_2376916* + (*truncate_gamma3f_x* (*0*::*real*) *_2376918* *_2376920* *_2376921* (*0*::*real*) (*0*::*real*) (*0*::*real*) *_2376923* *_2376925* *_2376927* ∗ *_2376917* + (*_2376926* − (*dih_x* *_2376923* *_2376924* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *_2376928* + *dih_x* *_2376923* (*real_of_nat* (*2*::*nat*)) *_2376925* (*real_of_nat* (*2*::*nat*)) *_2376927* (*real_of_nat* (*2*::*nat*)))) ∗ *truncate_gamma2_x* *_2376918* *_2376923*))

thm Functional_equation.truncate_gamma23a_x:

∀ (*m2*::*real*) (*m6*::*real*) (*iw1*::*real*) (*m3*::*real*) (*m5*::*real*) (*iw2*::*real*) (*az4*::*real*) (*x2*::*real*) (*x6*::*real*) (*x3*::*real*) (*x5*::*real*) (*m1*::*real*) *x1*::*real*. *truncate_gamma23a_x* *iw1* *iw2* *m1* *m2* *m3* *m5* *m6* *x1* *x2* *x3* *az4* *x5* *x6* = *truncate_gamma3f_x* (*0*::*real*) *m1* *m2* *m6* (*0*::*real*) (*0*::*real*) (*0*::*real*) *x1* *x2* *x6* ∗ *iw1* + (*truncate_gamma3f_x* (*0*::*real*) *m1* *m3* *m5* (*0*::*real*) (*0*::*real*) (*0*::*real*) *x1* *x3* *x5* ∗ *iw2* + (*az4* − (*dih_x* *x1* *x2* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *x6* + *dih_x* *x1* (*real_of_nat* (*2*::*nat*)) *x3* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)))) ∗ *truncate_gamma2_x* *m1* *x1*)

thm DEF_truncate_gamma23_x_C:

truncate_gamma23_x_C = (λ(*_2377137*::*real*) (*_2377138*::*real*) (*_2377139*::*real*) (*_2377140*::*real*) *_2377141*::*real*. *add6* (*scalar6* (*compose6* (*truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *_2377139* *_2377140* *_2377141*) *dummy6* *dummy6* *dummy6* *proj_x1* *proj_x2* *proj_x6*) *_2377138*) (*mul6* (*sub6* *dih_x* (*add6* (*mk_126*

$(truncate\_dih\_x\ (DECIMAL\ (14::nat)\ (100::nat))))\ (constant6\ \_2377137)))$
$(uni\ (truncate\_gamma2\_x\ \_2377139,\ proj\_x1))))$

thm Functional_equation.truncate_gamma23_x_C:

$\forall (m2::real)\ (m6::real)\ (iw1::real)\ (d::real)\ m1::real.\ truncate\_gamma23\_x\_C\ d\ iw1\ m1\ m2\ m6 = add6\ (scalar6\ (compose6\ (truncate\_gamma3f\_x\ (DECIMAL\ (14::nat)\ (100::nat))\ m1\ m2\ m6)\ dummy6\ dummy6\ dummy6\ proj\_x1\ proj\_x2\ proj\_x6)\ iw1)\ (mul6\ (sub6\ dih\_x\ (add6\ (mk\_126\ (truncate\_dih\_x\ (DECIMAL\ (14::nat)\ (100::nat))))\ (constant6\ d)))\ (uni\ (truncate\_gamma2\_x\ m1,\ proj\_x1)))$

thm DEF_truncate_gamma23_x_B:

$truncate\_gamma23\_x\_B = (\lambda\_2377182::real.\ mul6\ (sub6\ dih\_x\ (constant6\ (real\_of\_nat\ (2::nat) * DECIMAL\ (8::nat)\ (100::nat))))\ (uni\ (truncate\_gamma2\_x\ \_2377182,\ proj\_x1)))$

thm Functional_equation.truncate_gamma23_x_B:

$\forall m1::real.\ truncate\_gamma23\_x\_B\ m1 = mul6\ (sub6\ dih\_x\ (constant6\ (real\_of\_nat\ (2::nat) * DECIMAL\ (8::nat)\ (100::nat))))\ (uni\ (truncate\_gamma2\_x\ m1,\ proj\_x1))$

thm Functional_equation.cos797:

$cos797 = cos\ (DECIMAL\ (797::nat)\ (1000::nat))$

thm DEF_gamma2_x_div_azim:

$gamma2\_x\_div\_azim = (\lambda(\_2377187::real)\ \_2377188::real.\ (real\_of\_nat\ (8::nat) - \_2377188) * (sqrt\ \_2377188\ /\ real\_of\_nat\ (24::nat)) - (real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * ((1::real) - sqrt\ \_2377188\ /\ sqrt8) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * (\_2377187 * lfun\ (sqrt\ \_2377188\ /\ real\_of\_nat\ (2::nat)))))$

thm Functional_equation.gamma2_x_div_azim:

$\forall (m::real)\ x::real.\ gamma2\_x\_div\_azim\ m\ x = (real\_of\_nat\ (8::nat) - x) * (sqrt\ x\ /\ real\_of\_nat\ (24::nat)) - (real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * ((1::real) - sqrt\ x\ /\ sqrt8) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * (m * lfun\ (sqrt\ x\ /\ real\_of\_nat\ (2::nat))))$

thm DEF_gamma2_x1_div_a:

$gamma2\_x1\_div\_a = (\lambda\_2377199::real.\ promote1\_to\_6\ (gamma2\_x\_div\_azim\ \_2377199))$

thm Nonlin_def.gamma2_x1_div_a:

$\forall m::real.\ gamma2\_x1\_div\_a\ m = promote1\_to\_6\ (gamma2\_x\_div\_azim\ m)$

thm DEF_gamma3f_x_div_sqrtdelta:

$gamma3f\_x\_div\_sqrtdelta = (\lambda(\_2377204::real)\ (\_2377205::real)\ \_2377206::real.\ sub6\ (constant6\ ((1::real)\ /\ real\_of\_nat\ (12::nat)))\ (sub6\ (scalar6\ (add6\ (mk\_456\ (rotate5\ sol\_euler\_x\_div\_sqrtdelta))\ (add6\ (mk\_456\ (rotate6\ sol\_euler\_x\_div\_sqrtdelta))$

(*mk_456* (*rotate4* *sol_euler_x_div_sqrtdelta*)))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2377204*) (*mk_456* (*rotate4* *dih_x_div_sqrtdelta_posbranch*))) (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2377205*) (*mk_456* (*rotate5* *dih_x_div_sqrtdelta_posbranch*))) (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2377206*) (*mk_456* (*rotate6* *dih_x_div_sqrtdelta_posbranch*))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*)))))

thm Nonlin_def.gamma3f_x_div_sqrtdelta:

∀ (*m4*::*real*) (*m5*::*real*) *m6*::*real*. *gamma3f_x_div_sqrtdelta* *m4* *m5* *m6* = *sub6* (*constant6* ((*1*::*real*) / *real_of_nat* (*12*::*nat*))) (*sub6* (*scalar6* (*add6* (*mk_456* (*rotate5* *sol_euler_x_div_sqrtdelta*)) (*add6* (*mk_456* (*rotate6* *sol_euler_x_div_sqrtdelta*)) (*mk_456* (*rotate4* *sol_euler_x_div_sqrtdelta*)))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m4*) (*mk_456* (*rotate4* *dih_x_div_sqrtdelta_posbranch*))) (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m5*) (*mk_456* (*rotate5* *dih_x_div_sqrtdelta_posbranch*))) (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m6*) (*mk_456* (*rotate6* *dih_x_div_sqrtdelta_posbranch*))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*))))

thm DEF_vol3f_456:

vol3f_456 = (λ*_2377225*::*real*. *sub6* (*scalar6* (*add6* (*mk_456* (*rotate5* *sol_x*)) (*add6* (*mk_456* (*rotate6* *sol_x*)) (*mk_456* (*rotate4* *sol_x*)))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *_2377225*) (*mk_456* (*rotate4* *dih_x*))) (*add6* (*mul6* (*uni* (*lfun*, *scalar6* *proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) (*mk_456* (*rotate5* *dih_x*))) (*mul6* (*uni* (*lfun*, *scalar6* *proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) (*mk_456* (*rotate6* *dih_x*))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*))))

thm Functional_equation.vol3f_456:

∀ *m4*::*real*. *vol3f_456* *m4* = *sub6* (*scalar6* (*add6* (*mk_456* (*rotate5* *sol_x*)) (*add6* (*mk_456* (*rotate6* *sol_x*)) (*mk_456* (*rotate4* *sol_x*)))) (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*))) (*scalar6* (*add6* (*mul6* (*scalar6* (*uni* (*lfun*, *scalar6* *proj_y4* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *m4*) (*mk_456* (*rotate4* *dih_x*))) (*add6* (*mul6* (*uni* (*lfun*, *scalar6* *proj_y5* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) (*mk_456* (*rotate5* *dih_x*))) (*mul6* (*uni* (*lfun*, *scalar6* *proj_y6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) (*mk_456* (*rotate6* *dih_x*))))) (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*)))

thm DEF_gamma3_x:

gamma3_x = (λ*_2377230*::*real*. *sub6* (*mk_456* *vol_x*) (*vol3f_456* *_2377230*))

thm Functional_equation.gamma3_x:

∀ *m4*::*real*. *gamma3_x* *m4* = *sub6* (*mk_456* *vol_x*) (*vol3f_456* *m4*)

thm DEF_gamma23_full8_x:

gamma23_full8_x = (λ*_2377235*::*real*. *add6* (*compose6* (*gamma3_x _2377235*) *dummy6 dummy6 dummy6 proj_x1 proj_x2 proj_x6*) (*add6* (*compose6* (*gamma3_x _2377235*) *dummy6 dummy6 dummy6 proj_x1 proj_x3 proj_x5*) (*scalar6* (*sub6 dih_x* (*add6* (*mk_126 dih_x*) (*mk_135 dih_x*))) (*DECIMAL* (*8*::*nat*) (*1000*::*nat*)))))

thm Nonlin_def.gamma23_full8_x:

∀ *m1*::*real*. *gamma23_full8_x m1* = *add6* (*compose6* (*gamma3_x m1*) *dummy6 dummy6 dummy6 proj_x1 proj_x2 proj_x6*) (*add6* (*compose6* (*gamma3_x m1*) *dummy6 dummy6 dummy6 proj_x1 proj_x3 proj_x5*) (*scalar6* (*sub6 dih_x* (*add6* (*mk_126 dih_x*) (*mk_135 dih_x*))) (*DECIMAL* (*8*::*nat*) (*1000*::*nat*))))

thm DEF_gamma23_keep135_x:

gamma23_keep135_x = (λ*_2377240*::*real*. *add6* (*compose6* (*gamma3_x _2377240*) *dummy6 dummy6 dummy6 proj_x1 proj_x3 proj_x5*) (*scalar6* (*sub6 dih_x* (*mk_135 dih_x*)) (*DECIMAL* (*8*::*nat*) (*1000*::*nat*))))

thm Nonlin_def.gamma23_keep135_x:

∀ *m1*::*real*. *gamma23_keep135_x m1* = *add6* (*compose6* (*gamma3_x m1*) *dummy6 dummy6 dummy6 proj_x1 proj_x3 proj_x5*) (*scalar6* (*sub6 dih_x* (*mk_135 dih_x*)) (*DECIMAL* (*8*::*nat*) (*1000*::*nat*)))

thm DEF_dart_std4:

dart_std4 = (λ(*_2377245*::*?'a*::*type*) (*_2377246*::*?'a*::*type*) (*_2377247*::*?'a*::*type*) (*_2377248*::*?'a*::*type*) (*_2377249*::*?'a*::*type*) (*_2377250*::*?'a*::*type*) (*_2377251*::*?'a*::*type*) (*_2377252*::*?'a*::*type*) *_2377253*::*?'a*::*type*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377245*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377246*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377247*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*real_of_nat* (*2*::*nat*) ∗ *h0*, *_2377248*, *DECIMAL* (*437*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377249*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377250*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377251*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377252*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377253*, *real_of_nat* (*2*::*nat*) ∗ *h0*)])

thm Ineq.dart_std4:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) (*y6*::*?'a*::*type*) (*y7*::*?'a*::*type*) (*y8*::*?'a*::*type*) *y9*::*?'a*::*type*. *dart_std4 y1 y2 y3 y4 y5 y6 y7 y8 y9* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*real_of_nat* (*2*::*nat*) ∗ *h0*, *y4*, *DECIMAL* (*437*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y7*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y8*, *real_of_nat* (*2*::*nat*) ∗ *h0*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y9*, *real_of_nat* (*2*::*nat*) ∗ *h0*)]

thm DEF_dart4_diag3_b:

dart4_diag3_b = (λ(*_2377362*::*?'a*::*type*) (*_2377363*::*?'a*::*type*) (*_2377364*::*?'a*::*type*) (*_2377365*::*?'a*::*type*) (*_2377366*::*?'a*::*type*) (*_2377367*::*?'a*::*type*) (*_2377368*::*?'a*::*type*) (*_2377369*::*?'a*::*type*) *_2377370*::*?'a*::*type*. [(*real_of_nat* (*2*::*nat*), *_2377362*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377363*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377364*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*3*::*nat*), *_2377365*, *real_of_nat* (*3*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377366*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377367*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377368*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377369*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377370*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.dart4_diag3_b:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) (*y6*::*?'a*::*type*) (*y7*::*?'a*::*type*) (*y8*::*?'a*::*type*) *y9*::*?'a*::*type*. *dart4_diag3_b y1 y2 y3 y4 y5 y6 y7 y8 y9* = [(*real_of_nat* (*2*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*3*::*nat*), *y4*, *real_of_nat* (*3*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y7*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y8*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y9*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_dart_std3:

dart_std3 = (λ(*_2377479*::*?'a*::*type*) (*_2377480*::*?'a*::*type*) (*_2377481*::*?'a*::*type*) (*_2377482*::*?'a*::*type*) (*_2377483*::*?'a*::*type*) *_2377484*::*?'a*::*type*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377479*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377480*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377481*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377482*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377483*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377484*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.dart_std3:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_std3 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_dartX:

dartX = (λ(*_2377539*::*?'a*::*type*) (*_2377540*::*?'a*::*type*) (*_2377541*::*?'a*::*type*) (*_2377542*::*?'a*::*type*) (*_2377543*::*?'a*::*type*) *_2377544*::*?'a*::*type*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377539*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL*

(20::nat) (10::nat), _2377540, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377541, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), _2377542, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377543, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377544, DECIMAL (252::nat) (100::nat))])

thm Ineq.dartX:

∀ (y1::?'a::type) (y2::?'a::type) (y3::?'a::type) (y4::?'a::type) (y5::?'a::type) y6::?'a::type. dartX y1 y2 y3 y4 y5 y6 = [(DECIMAL (20::nat) (10::nat), y1, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y2, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y3, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), y4, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y5, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y6, DECIMAL (252::nat) (100::nat))]

thm DEF_dartY:

dartY = (λ(_2377599::?'a::type) (_2377600::?'a::type) (_2377601::?'a::type) (_2377602::?'a::type) (_2377603::?'a::type) _2377604::?'a::type. [(DECIMAL (20::nat) (10::nat), _2377599, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377600, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377601, DECIMAL (252::nat) (100::nat)), (sqrt8, _2377602, sqrt8), (DECIMAL (20::nat) (10::nat), _2377603, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377604, DECIMAL (252::nat) (100::nat))])

thm Ineq.dartY:

∀ (y1::?'a::type) (y2::?'a::type) (y3::?'a::type) (y4::?'a::type) (y5::?'a::type) y6::?'a::type. dartY y1 y2 y3 y4 y5 y6 = [(DECIMAL (20::nat) (10::nat), y1, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y2, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y3, DECIMAL (252::nat) (100::nat)), (sqrt8, y4, sqrt8), (DECIMAL (20::nat) (10::nat), y5, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y6, DECIMAL (252::nat) (100::nat))]

thm DEF_dart4_diag3:

dart4_diag3 = (λ(_2377659::?'a::type) (_2377660::?'a::type) (_2377661::?'a::type) (_2377662::?'a::type) (_2377663::?'a::type) _2377664::?'a::type. [(DECIMAL (20::nat) (10::nat), _2377659, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377660, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377661, DECIMAL (252::nat) (100::nat)), (DECIMAL (30::nat) (10::nat), _2377662, DECIMAL (30::nat) (10::nat)), (DECIMAL (20::nat) (10::nat), _2377663, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377664, DECIMAL (252::nat) (100::nat))])

thm Ineq.dart4_diag3:

∀ (y1::?'a::type) (y2::?'a::type) (y3::?'a::type) (y4::?'a::type) (y5::?'a::type) y6::?'a::type. dart4_diag3 y1 y2 y3 y4 y5 y6 = [(DECIMAL (20::nat) (10::nat), y1, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y2, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y3, DECIMAL (252::nat) (100::nat)), (DECIMAL (30::nat) (10::nat), y4, DECIMAL (30::nat) (10::nat)), (DECIMAL (20::nat) (10::nat), y5, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y6, DECIMAL (252::nat) (100::nat))]

thm DEF_apex_flat:

apex_flat = (λ(_2377719::?'a::type) (_2377720::?'a::type) (_2377721::?'a::type) (_2377722::?'a::type) (_2377723::?'a::type) _2377724::?'a::type. [(DECIMAL (20::nat) (10::nat), _2377719, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377720, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377721, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), _2377722, sqrt8), (DECIMAL (20::nat) (10::nat), _2377723, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377724, DECIMAL (252::nat) (100::nat))])

thm Ineq.apex_flat:

∀ (y1::?'a::type) (y2::?'a::type) (y3::?'a::type) (y4::?'a::type) (y5::?'a::type) y6::?'a::type. apex_flat y1 y2 y3 y4 y5 y6 = [(DECIMAL (20::nat) (10::nat), y1, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y2, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y3, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), y4, sqrt8), (DECIMAL (20::nat) (10::nat), y5, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y6, DECIMAL (252::nat) (100::nat))]

thm DEF_apex_A:

apex_A = (λ(_2377779::?'a::type) (_2377780::?'a::type) (_2377781::?'a::type) (_2377782::?'a::type) (_2377783::?'a::type) _2377784::?'a::type. [(DECIMAL (20::nat) (10::nat), _2377779, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377780, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377781, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), _2377782, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), _2377783, sqrt8), (DECIMAL (252::nat) (100::nat), _2377784, sqrt8)])

thm Ineq.apex_A:

∀ (y1::?'a::type) (y2::?'a::type) (y3::?'a::type) (y4::?'a::type) (y5::?'a::type) y6::?'a::type. apex_A y1 y2 y3 y4 y5 y6 = [(DECIMAL (20::nat) (10::nat), y1, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y2, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y3, DECIMAL (252::nat) (100::nat)), (DECIMAL (20::nat) (10::nat), y4, DECIMAL (252::nat) (100::nat)), (DECIMAL (252::nat) (100::nat), y5, sqrt8), (DECIMAL (252::nat) (100::nat), y6, sqrt8)]

thm DEF_dart_std3_small:

dart_std3_small = (λ(_2377839::?'a::type) (_2377840::?'a::type) (_2377841::?'a::type) (_2377842::?'a::type) (_2377843::?'a::type) _2377844::?'a::type. [(DECIMAL

(*20*::*nat*) (*10*::*nat*), *_2377839*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377840*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377841*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377842*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377843*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377844*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))])

thm Ineq.dart_std3_small:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_std3_small y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y4*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))]

thm Ineq.dart_std3_big:

dart_std3_big = *dart_std3*

thm DEF_apex_sup_flat:

apex_sup_flat = (λ(*_2377899*::*?'a*::*type*) (*_2377900*::*?'a*::*type*) (*_2377901*::*?'a*::*type*) (*_2377902*::*?'a*::*type*) (*_2377903*::*?'a*::*type*) *_2377904*::*?'a*::*type*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377899*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377900*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377901*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *_2377902*, *DECIMAL* (*30*::*nat*) (*10*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377903*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2377904*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_sup_flat:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_sup_flat y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *y4*, *DECIMAL* (*30*::*nat*) (*10*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_dart_std3_mini:

dart_std3_mini = (λ(*_2377959*::*?'a*::*type*) (*_2377960*::*?'a*::*type*) (*_2377961*::*?'a*::*type*) (*_2377962*::*?'a*::*type*) (*_2377963*::*?'a*::*type*) *_2377964*::*?'a*::*type*. [(*DECIMAL* (*2*::*nat*) (*1*::*nat*), *_2377959*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377960*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377961*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377962*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377963*, *DECIMAL*

(*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2377964*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))])

thm Ineq.dart_std3_mini:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_std3_mini y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*2*::*nat*) (*1*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))]

thm DEF_apex_flat_hll:

apex_flat_hll = (λ(*_2378019*::*?'a*::*type*) (*_2378020*::*?'a*::*type*) (*_2378021*::*?'a*::*type*) (*_2378022*::*?'a*::*type*) (*_2378023*::*?'a*::*type*) *_2378024*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378019*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378020*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378021*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378022*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378023*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378024*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_flat_hll:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_flat_hll y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_dart_std3_big_200_218:

dart_std3_big_200_218 = (λ(*_2378079*::*?'a*::*type*) (*_2378080*::*?'a*::*type*) (*_2378081*::*?'a*::*type*) (*_2378082*::*?'a*::*type*) (*_2378083*::*?'a*::*type*) *_2378084*::*?'a*::*type*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378079*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378080*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378081*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378082*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378083*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378084*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.dart_std3_big_200_218:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_std3_big_200_218 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apexffA:

apexffA = (λ(*_2378139*::*real*) (*_2378140*::*real*) (*_2378141*::*real*) (*_2378142*::*real*) (*_2378143*::*real*) *_2378144*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378139*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378140*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378141*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378142*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378143*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378144*, *sqrt8*)])

thm Ineq.apexffA:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexffA y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y6*, *sqrt8*)]

thm DEF_apexfA:

apexfA = (λ(*_2378199*::*real*) (*_2378200*::*real*) (*_2378201*::*real*) (*_2378202*::*real*) (*_2378203*::*real*) *_2378204*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378199*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378200*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378201*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378202*, *sqrt8*), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378203*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378204*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apexfA:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexfA y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *sqrt8*), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y5*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apexf4:

apexf4 = (λ(*_2378259*::*real*) (*_2378260*::*real*) (*_2378261*::*real*) (*_2378262*::*real*) (*_2378263*::*real*) *_2378264*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378259*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378260*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378261*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *_2378262*, *sqrt8*), (*DECIMAL* (*20*::*nat*)

(*10*::*nat*), *_2378263*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378264*, *sqrt8*)])

thm Ineq.apexf4:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexf4 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *y4*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y6*, *sqrt8*)]

thm DEF_apexff4:

apexff4 = (λ(*_2378319*::*real*) (*_2378320*::*real*) (*_2378321*::*real*) (*_2378322*::*real*) (*_2378323*::*real*) *_2378324*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378319*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378320*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378321*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *_2378322*, *sqrt8*), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378323*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378324*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apexff4:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexff4 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*sqrt8*, *y4*, *sqrt8*), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y5*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apexf5:

apexf5 = (λ(*_2378379*::*real*) (*_2378380*::*real*) (*_2378381*::*real*) (*_2378382*::*real*) (*_2378383*::*real*) *_2378384*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378379*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378380*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378381*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378382*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378383*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378384*, *sqrt8*)])

thm Ineq.apexf5:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexf5 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y6*, *sqrt8*)]

thm DEF_apexff5:

apexff5 = (λ(*_2378439*::*real*) (*_2378440*::*real*) (*_2378441*::*real*) (*_2378442*::*real*) (*_2378443*::*real*) *_2378444*::*real*. [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378439*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378440*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378441*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378442*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378443*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *_2378444*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apexff5:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *apexff5 y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y5*, *sqrt8*), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_std3_hll:

apex_std3_hll = (λ(*_2378499*::*?'a*::*type*) (*_2378500*::*?'a*::*type*) (*_2378501*::*?'a*::*type*) (*_2378502*::*?'a*::*type*) (*_2378503*::*?'a*::*type*) *_2378504*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378499*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378500*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378501*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378502*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378503*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378504*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_std3_hll:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_std3_hll y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_std3_small_hll:

apex_std3_small_hll = (λ(*_2378559*::*?'a*::*type*) (*_2378560*::*?'a*::*type*) (*_2378561*::*?'a*::*type*) (*_2378562*::*?'a*::*type*) (*_2378563*::*?'a*::*type*) *_2378564*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378559*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378560*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378561*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378562*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378563*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378564*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))])

thm Ineq.apex_std3_small_hll:

$\forall$ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_std3_small_hll y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*225*::*nat*) (*100*::*nat*))]

thm DEF_dart_mll_w:

dart_mll_w = (λ(*_2378619*::*?'a*::*type*) (*_2378620*::*?'a*::*type*) (*_2378621*::*?'a*::*type*) (*_2378622*::*?'a*::*type*) (*_2378623*::*?'a*::*type*) *_2378624*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378619*, *DECIMAL* (*236*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378620*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378621*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2378622*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378623*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378624*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm DEF_dart_mll_n:

dart_mll_n = (λ(*_2378679*::*?'a*::*type*) (*_2378680*::*?'a*::*type*) (*_2378681*::*?'a*::*type*) (*_2378682*::*?'a*::*type*) (*_2378683*::*?'a*::*type*) *_2378684*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378679*, *DECIMAL* (*236*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378680*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378681*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378682*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378683*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378684*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm DEF_dart_Hll_n:

dart_Hll_n = (λ(*_2378739*::*?'a*::*type*) (*_2378740*::*?'a*::*type*) (*_2378741*::*?'a*::*type*) (*_2378742*::*?'a*::*type*) (*_2378743*::*?'a*::*type*) *_2378744*::*?'a*::*type*. [(*DECIMAL* (*236*::*nat*) (*100*::*nat*), *_2378739*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378740*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378741*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378742*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378743*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378744*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm DEF_dart_Hll_w:

dart_Hll_w = (λ(*_2378799*::*?'a*::*type*) (*_2378800*::*?'a*::*type*) (*_2378801*::*?'a*::*type*) (*_2378802*::*?'a*::*type*) (*_2378803*::*?'a*::*type*) *_2378804*::*?'a*::*type*. [(*DECIMAL* (*236*::*nat*) (*100*::*nat*), *_2378799*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378800*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378801*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2378802*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378803*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378804*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.dart_mll_w:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_mll_w y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*236*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm Ineq.dart_mll_n:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_mll_n y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*236*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm Ineq.dart_Hll_n:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_Hll_n y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*236*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*225*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm Ineq.dart_Hll_w:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_Hll_w y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*236*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_dart_std3_lw:

dart_std3_lw = (λ(*_2378859*::*?'a*::*type*) (*_2378860*::*?'a*::*type*) (*_2378861*::*?'a*::*type*) (*_2378862*::*?'a*::*type*) (*_2378863*::*?'a*::*type*) *_2378864*::*?'a*::*type*. [(*DECIMAL* (*200*::*nat*) (*100*::*nat*), *_2378859*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378860*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378861*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2378862*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378863*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378864*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.dart_std3_lw:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *dart_std3_lw y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*200*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_flat_l:

apex_flat_l = (λ(*_2378919*::*?'a*::*type*) (*_2378920*::*?'a*::*type*) (*_2378921*::*?'a*::*type*) (*_2378922*::*?'a*::*type*) (*_2378923*::*?'a*::*type*) *_2378924*::*?'a*::*type*. [(*DECIMAL* (*2*::*nat*) (*1*::*nat*), *_2378919*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378920*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378921*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378922*, *sqrt8*), (*real_of_nat* (*2*::*nat*), *_2378923*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378924*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_flat_l:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_flat_l y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*2*::*nat*) (*1*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *sqrt8*), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_flat_h:

apex_flat_h = (λ(*_2378979*::*?'a*::*type*) (*_2378980*::*?'a*::*type*) (*_2378981*::*?'a*::*type*) (*_2378982*::*?'a*::*type*) (*_2378983*::*?'a*::*type*) *_2378984*::*?'a*::*type*. [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *_2378979*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378980*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378981*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *_2378982*, *sqrt8*), (*real_of_nat* (*2*::*nat*), *_2378983*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2378984*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_flat_h:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_flat_h y1 y2 y3 y4 y5 y6* = [(*DECIMAL* (*218*::*nat*) (*100*::*nat*), *y1*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *y4*, *sqrt8*), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_std3_lll_xww:

apex_std3_lll_xww = (λ(*_2379039*::*?'a*::*type*) (*_2379040*::*?'a*::*type*) (*_2379041*::*?'a*::*type*) (*_2379042*::*?'a*::*type*) (*_2379043*::*?'a*::*type*) *_2379044*::*?'a*::*type*. [(*real_of_nat* (*2*::*nat*), *_2379039*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379040*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379041*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379042*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2379043*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2379044*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_std3_lll_xww:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_std3_lll_xww y1 y2 y3 y4 y5 y6* = [(*real_of_nat* (*2*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_std3_lll_wxx:

apex_std3_lll_wxx = (λ(*_2379099*::*?'a*::*type*) (*_2379100*::*?'a*::*type*) (*_2379101*::*?'a*::*type*) (*_2379102*::*?'a*::*type*) (*_2379103*::*?'a*::*type*) *_2379104*::*?'a*::*type*. [(*real_of_nat* (*2*::*nat*), *_2379099*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379100*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379101*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *_2379102*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379103*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *_2379104*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))])

thm Ineq.apex_std3_lll_wxx:

∀ (*y1*::*?'a*::*type*) (*y2*::*?'a*::*type*) (*y3*::*?'a*::*type*) (*y4*::*?'a*::*type*) (*y5*::*?'a*::*type*) *y6*::*?'a*::*type*. *apex_std3_lll_wxx y1 y2 y3 y4 y5 y6* = [(*real_of_nat* (*2*::*nat*), *y1*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y2*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y3*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*225*::*nat*) (*100*::*nat*), *y4*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y5*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*real_of_nat* (*2*::*nat*), *y6*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))]

thm DEF_apex_std3_lhh:

apex_std3_lhh = (λ(*_2379159*::*?'a*::*type*) (*_2379160*::*?'a*::*type*) (*_2379161*::*?'a*::*type*) (*_2379162*::*?'a*::*type*) (*_2379163*::*?'a*::*type*) *_2379164*::*?'a*::*type*. [(*real_of_nat* (*2*::*nat*), *_2379159*, *DECIMAL* (*218*::*nat*) (*100*::*nat*)), (*DECIMAL* (*218*::*nat*)

$(100::nat)$, $\_2379160$, $DECIMAL\ (252::nat)\ (100::nat))$, $(DECIMAL\ (218::nat)\ (100::nat)$, $\_2379161$, $DECIMAL\ (252::nat)\ (100::nat))$, $(real\_of\_nat\ (2::nat)$, $\_2379162$, $DECIMAL\ (252::nat)\ (100::nat))$, $(real\_of\_nat\ (2::nat)$, $\_2379163$, $DECIMAL\ (252::nat)\ (100::nat))$, $(real\_of\_nat\ (2::nat)$, $\_2379164$, $DECIMAL\ (252::nat)\ (100::nat))])$

thm Ineq.apex_std3_lhh:

$\forall\ (y1::?'a::type)\ (y2::?'a::type)\ (y3::?'a::type)\ (y4::?'a::type)\ (y5::?'a::type)\ y6::?'a::type.\ apex\_std3\_lhh\ y1\ y2\ y3\ y4\ y5\ y6 = [(real\_of\_nat\ (2::nat),\ y1,\ DECIMAL\ (218::nat)\ (100::nat)),\ (DECIMAL\ (218::nat)\ (100::nat),\ y2,\ DECIMAL\ (252::nat)\ (100::nat)),\ (DECIMAL\ (218::nat)\ (100::nat),\ y3,\ DECIMAL\ (252::nat)\ (100::nat)),\ (real\_of\_nat\ (2::nat),\ y4,\ DECIMAL\ (252::nat)\ (100::nat)),\ (real\_of\_nat\ (2::nat),\ y5,\ DECIMAL\ (252::nat)\ (100::nat)),\ (real\_of\_nat\ (2::nat),\ y6,\ DECIMAL\ (252::nat)\ (100::nat))]$

thm DEF_dua:

$dua = (\lambda(\_2379219::real)\ (\_2379220::real)\ \_2379221::real.\ real\_of\_nat\ (2::nat) * (\_2379220 + (\_2379221 - \_2379219)))$

thm Mdtau.dua:

$\forall\ (b::real)\ (c::real)\ a::real.\ dua\ a\ b\ c = real\_of\_nat\ (2::nat) * (b + (c - a))$

thm DEF_safesqrt:

$safesqrt = (\lambda\_2379240::real.\ if\ (0::real) \leq \_2379240\ then\ sqrt\ \_2379240\ else\ (0::real))$

thm Mdtau.safesqrt:

$\forall\ x::real.\ safesqrt\ x = (if\ (0::real) \leq x\ then\ sqrt\ x\ else\ (0::real))$

thm DEF_mdtau_y:

$mdtau\_y = (\lambda(\_2379245::real)\ (\_2379246::real)\ (\_2379247::real)\ (\_2379248::real)\ (\_2379249::real)\ \_2379250::real.\ LET\ (\lambda x1::real.\ LET\_END\ (LET\ (\lambda x2::real.\ LET\_END\ (LET\ (\lambda x3::real.\ LET\_END\ (LET\ (\lambda x4::real.\ LET\_END\ (LET\ (\lambda x5::real.\ LET\_END\ (LET\ (\lambda x6::real.\ LET\_END\ (LET\ (\lambda chain0::real.\ LET\_END\ (LET\ (\lambda Pchain::real.\ LET\_END\ (LET\ (\lambda chain2::real.\ LET\_END\ (LET\ (\lambda u135::real.\ LET\_END\ (LET\ (\lambda u126::real.\ LET\_END\ (LET\ (\lambda u234::real.\ LET\_END\ (LET\ (\lambda uf::real.\ LET\_END\ (LET\ (\lambda du135::real.\ LET\_END\ (LET\ (\lambda du126::real.\ LET\_END\ (LET\ (\lambda Luf::real.\ LET\_END\ (LET\ (\lambda n4::real.\ LET\_END\ (LET\ (\lambda del4::real.\ LET\_END\ (LET\ (\lambda n5::real.\ LET\_END\ (LET\ (\lambda n6::real.\ LET\_END\ (LET\ (\lambda Dn4::real.\ LET\_END\ (LET\ (\lambda del::real.\ LET\_END\ (LET\ (\lambda del1::real.\ LET\_END\ (LET\ (\lambda del2::real.\ LET\_END\ (LET\ (\lambda del3::real.\ LET\_END\ (LET\ (\lambda Pdel::real.\ LET\_END\ (LET\ (\lambda Ldel::real.\ LET\_END\ (LET\ (\lambda sd4::real.\ LET\_END\ (LET\ (\lambda sd5::real.\ LET\_END\ (LET\ (\lambda sd6::real.\ LET\_END\ (LET\ (\lambda Dsd4::real.\ LET\_END\ (LET\ (\lambda m4diff::real.\ LET\_END\ (LET\ (\lambda m4::real.\ LET\_END\ (LET\ (\lambda m5::real.\ LET\_END\ (LET\ (\lambda m6::real.\ LET\_END\ (LET\ (\lambda const1::real.\ LET\_END\ (LET\ (\lambda rhoy1::real.\ LET\_END\ (LET\ (\lambda rhoy2::real.\ LET\_END\ (LET\ (\lambda rhoy3::real.$

LET_END (LET (λPrhoy1::real. LET_END (LET (λrr::real. LET_END (LET (λterm1::real. LET_END (LET (λt::real. LET_END (LET (λt2::real. LET_END (LET (λterm2a::real. LET_END (LET (λterm2::real. LET_END (LET (λterm3::real. LET_END (term1 + (term2 + term3))) (rr / uf))) (term2a ∗ Prhoy1))) (del ∗ (t ∗ matan (t2 ∗ del))))) (t ∗ t))) (sqrt (real_of_nat (4::nat) ∗ x1) / del4))) (Prhoy1 ∗ (pi ∗ safesqrt del)))) (rhoy1 ∗ m4 + (rhoy2 ∗ m5 + rhoy3 ∗ m6)))) (const1 / DECIMAL (52::nat) (100::nat)))) (rho _2379247))) (rho _2379246))) (rho _2379245))) (sol_y (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) / pi))) (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (del2 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u126 ∗ _2379246))))))))) (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (del3 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u135 ∗ _2379247))))))))) (m4diff ∗ (chain0 ∗ (u234 ∗ (_2379246 ∗ _2379247)))))) (real_of_nat (2::nat) ∗ (Dn4 ∗ sd4) − n4 ∗ Dsd4))) (real_of_nat (4::nat) ∗ del + real_of_nat (4::nat) ∗ (x1 ∗ del1)))) (real_of_nat (4::nat) ∗ (x3 ∗ del)))) (real_of_nat (4::nat) ∗ (x2 ∗ del)))) (real_of_nat (4::nat) ∗ (x1 ∗ del)))) (Pdel / del))) (del1 ∗ chain0))) (x1 ∗ x4 − x2 ∗ x4 − x1 ∗ x5 + (x2 ∗ x5 − x3 ∗ x6 + (x1 + (x2 − x3 + (x4 + (x5 − x6)))) ∗ x6)))) (x1 ∗ x4 − x3 ∗ x4 − x2 ∗ x5 − x1 ∗ x6 + (x3 ∗ x6 + x5 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6))))))) (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) (delta_x x1 x2 x3 x4 x5 x6))) (real_of_nat (2::nat) ∗ x1 − x2 − x3 + (real_of_nat (2::nat) ∗ x4 − x5 − x6)))) (x1 ∗ x2 − x1 ∗ x4 − x2 ∗ x5 + (x4 ∗ x5 − x3 ∗ (x1 + (x2 − x3 + (x4 + (x5 − x6)))) + x3 ∗ x6)))) (x1 ∗ x3 − x1 ∗ x4 + (x2 ∗ x5 − x3 ∗ x6 + (x4 ∗ x6 − x2 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6)))))))) (− n4))) (x2 ∗ x3 + (x1 ∗ x4 − x2 ∗ x5 − x3 ∗ x6 + (x5 ∗ x6 − x1 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6)))))))) ((du135 / u135 + du126 / u126) ∗ chain0 + (1::real) / _2379245))) (dua x1 x2 x6))) (dua x1 x3 x5))) (real_of_nat (4::nat) ∗ (u135 ∗ (u126 ∗ (u234 ∗ (_2379245 ∗ (_2379246 ∗ _2379247)))))))) (ups_x x2 x3 x4))) (ups_x x1 x2 x6))) (ups_x x1 x3 x5))) (real_of_nat (4::nat) ∗ x1))) (real_of_nat (2::nat)))) (real_of_nat (2::nat) ∗ _2379245))) (_2379250 ∗ _2379250))) (_2379249 ∗ _2379249))) (_2379248 ∗ _2379248))) (_2379247 ∗ _2379247))) (_2379246 ∗ _2379246))) (_2379245 ∗ _2379245))

thm Mdtau.mdtau_y:

∀(y6::real) (y5::real) (y4::real) (y3::real) (y2::real) y1::real. mdtau_y y1 y2 y3 y4 y5 y6 = LET (λx1::real. LET_END (LET (λx2::real. LET_END (LET (λx3::real. LET_END (LET (λx4::real. LET_END (LET (λx5::real. LET_END (LET (λx6::real. LET_END (LET (λchain0::real. LET_END (LET (λPchain::real. LET_END (LET (λchain2::real. LET_END (LET (λu135::real. LET_END (LET (λu126::real. LET_END (LET (λu234::real. LET_END (LET (λuf::real. LET_END (LET (λdu135::real. LET_END (LET (λdu126::real. LET_END (LET (λLuf::real. LET_END (LET (λn4::real. LET_END (LET (λdel4::real. LET_END (LET (λn5::real. LET_END (LET (λn6::real. LET_END (LET (λDn4::real. LET_END (LET (λdel::real. LET_END (LET (λdel1::real. LET_END (LET (λdel2::real. LET_END (LET (λdel3::real. LET_END (LET (λPdel::real.

LET_END (LET (λLdel::real. LET_END (LET (λsd4::real. LET_END (LET (λsd5::real. LET_END (LET (λsd6::real. LET_END (LET (λDsd4::real. LET_END (LET (λm4diff::real. LET_END (LET (λm4::real. LET_END (LET (λm5::real. LET_END (LET (λm6::real. LET_END (LET (λconst1::real. LET_END (LET (λrhoy1::real. LET_END (LET (λrhoy2::real. LET_END (LET (λrhoy3::real. LET_END (LET (λPrhoy1::real. LET_END (LET (λrr::real. LET_END (LET (λterm1::real. LET_END (LET (λt::real. LET_END (LET (λt2::real. LET_END (LET (λterm2a::real. LET_END (LET (λterm2::real. LET_END (LET (λterm3::real. LET_END (term1 + (term2 + term3))) (rr / uf))) (term2a ∗ Prhoy1))) (del ∗ (t ∗ matan (t2 ∗ del))))) (t ∗ t))) (sqrt (real_of_nat (4::nat) ∗ x1) / del4))) (Prhoy1 ∗ (pi ∗ safesqrt del)))) (rhoy1 ∗ m4 + (rhoy2 ∗ m5 + rhoy3 ∗ m6)))) (const1 / DECIMAL (52::nat) (100::nat)))) (rho y3))) (rho y2))) (rho y1))) (sol_y (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) / pi))) (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (del2 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u126 ∗ y2))))))))) (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (del3 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u135 ∗ y3))))))))) (m4diff ∗ (chain0 ∗ (u234 ∗ (y2 ∗ y3)))))) (real_of_nat (2::nat) ∗ (Dn4 ∗ sd4) − n4 ∗ Dsd4))) (real_of_nat (4::nat) ∗ del + real_of_nat (4::nat) ∗ (x1 ∗ del1)))) (real_of_nat (4::nat) ∗ (x3 ∗ del)))) (real_of_nat (4::nat) ∗ (x2 ∗ del)))) (real_of_nat (4::nat) ∗ (x1 ∗ del)))) (Pdel / del))) (del1 ∗ chain0))) (x1 ∗ x4 − x2 ∗ x4 − x1 ∗ x5 + (x2 ∗ x5 − x3 ∗ x6 + (x1 + (x2 − x3 + (x4 + (x5 − x6)))) ∗ x6)))) (x1 ∗ x4 − x3 ∗ x4 − x2 ∗ x5 − x1 ∗ x6 + (x3 ∗ x6 + x5 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6))))))) (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) (delta_x x1 x2 x3 x4 x5 x6))) (real_of_nat (2::nat) ∗ x1 − x2 − x3 + (real_of_nat (2::nat) ∗ x4 − x5 − x6)))) (x1 ∗ x2 − x1 ∗ x4 − x2 ∗ x5 + (x4 ∗ x5 − x3 ∗ (x1 + (x2 − x3 + (x4 + (x5 − x6)))) + x3 ∗ x6)))) (x1 ∗ x3 − x1 ∗ x4 + (x2 ∗ x5 − x3 ∗ x6 + (x4 ∗ x6 − x2 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6)))))))) (− n4))) (x2 ∗ x3 + (x1 ∗ x4 − x2 ∗ x5 − x3 ∗ x6 + (x5 ∗ x6 − x1 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) ((du135 / u135 + du126 / u126) ∗ chain0 + (1::real) / y1))) (dua x1 x2 x6))) (dua x1 x3 x5))) (real_of_nat (4::nat) ∗ (u135 ∗ (u126 ∗ (u234 ∗ (y1 ∗ (y2 ∗ y3)))))))) (ups_x x2 x3 x4))) (ups_x x1 x2 x6))) (ups_x x1 x3 x5))) (real_of_nat (4::nat) ∗ x1))) (real_of_nat (2::nat)))) (real_of_nat (2::nat) ∗ y1))) (y6 ∗ y6))) (y5 ∗ y5))) (y4 ∗ y4))) (y3 ∗ y3))) (y2 ∗ y2))) (y1 ∗ y1)

thm DEF_mdtau2uf_y:

mdtau2uf_y = (λ(_2379305::real) (_2379306::real) (_2379307::real) (_2379308::real) (_2379309::real) _2379310::real. LET (λx1::real. LET_END (LET (λx2::real. LET_END (LET (λx3::real. LET_END (LET (λx4::real. LET_END (LET (λx5::real. LET_END (LET (λx6::real. LET_END (LET (λchain0::real. LET_END (LET (λPchain::real. LET_END (LET (λchain2::real. LET_END (LET (λu135::real. LET_END (LET (λu126::real. LET_END (LET (λu234::real. LET_END (LET (λuf::real. LET_END (LET (λdu135::real. LET_END (LET (λdu126::real. LET_END (LET (λLuf::real. LET_END (LET (λn4::real. LET_END (LET (λdel4::real. LET_END (LET (λn5::real. LET_END (LET (λn6::real. LET_END (LET (λDn4::real. LET_END (LET (λdel::real. LET_END (LET (λdel1::real. LET_END (LET (λdel2::real. LET_END (LET (λdel3::real. LET_END (LET (λPdel::real. LET_END (LET (λLdel::real. LET_END (LET (λsd4::real. LET_END (LET (λsd5::real. LET_END (LET (λsd6::real. LET_END (LET (λDsd4::real. LET_END (LET (λm4diff::real. LET_END (LET (λm4::real. LET_END (LET (λm5::real. LET_END (LET (λm6::real. LET_END (LET (λconst1::real. LET_END (LET (λrhoy1::real. LET_END (LET (λrhoy2::real. LET_END (LET (λrhoy3::real. LET_END (LET (λPrhoy1::real. LET_END (LET (λrr::real. LET_END (LET (λD2n4::real. LET_END (LET (λD2sd4::real. LET_END (LET (λDm4diff::real. LET_END (LET (λPm4::real. LET_END (LET (λDdel3::real. LET_END (LET (λDdel2::real. LET_END (LET (λPm5::real. LET_END (LET (λPm6::real. LET_END (LET (λPrrC::real. LET_END (LET (λP2tauNum::real. LET_END (LET LET_END (P2tauNum / safesqrt del))) (PrrC + (− Luf − DECIMAL (5::nat) (10::nat) ∗ Ldel) ∗ rr))) (real_of_nat (2::nat) ∗ (Prhoy1 ∗ m4) + (rhoy1 ∗ Pm4 + (rhoy2 ∗ Pm5 + rhoy3 ∗ Pm6))))) ((Ddel2 ∗ (x1 ∗ u126) + (del2 ∗ ((1::real) ∗ u126) + del2 ∗ (x1 ∗ du126))) ∗ (chain0 ∗ (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (real_of_nat (2::nat) ∗ _2379306)))))))) ((Ddel3 ∗ (x1 ∗ u135) + (del3 ∗ ((1::real) ∗ u135) + del3 ∗ (x1 ∗ du135))) ∗ (chain0 ∗ (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (real_of_nat (2::nat) ∗ _2379307)))))))) (x4 + (x5 − x6)))) (x4 − x5 + x6))) ((Dm4diff ∗ chain2 + m4diff ∗ Pchain) ∗ (u234 ∗ (_2379306 ∗ _2379307))))) (real_of_nat (2::nat) ∗ (D2n4 ∗ sd4) + (Dn4 ∗ Dsd4 − n4 ∗ D2sd4)))) (− real_of_nat (8::nat) ∗ (x1 ∗ x4) + real_of_nat (8::nat) ∗ (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) (real_of_nat (2::nat)))) (rhoy1 ∗ m4 + (rhoy2 ∗ m5 + rhoy3 ∗ m6)))) (const1 / DECIMAL (52::nat) (100::nat)))) (rho _2379307))) (rho _2379306))) (rho _2379305))) (sol_y (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) / pi))) (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (del2 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u126 ∗ _2379306))))))))) (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (del3 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u135 ∗ _2379307))))))))) (m4diff ∗ (chain0 ∗ (u234 ∗ (_2379306 ∗ _2379307)))))) (real_of_nat (2::nat) ∗ (Dn4 ∗ sd4) − n4 ∗ Dsd4))) (real_of_nat (4::nat) ∗ del + real_of_nat (4::nat) ∗ (x1 ∗ del1)))) (real_of_nat (4::nat) ∗ (x3 ∗ del)))) (real_of_nat (4::nat) ∗ (x2 ∗ del)))) (real_of_nat (4::nat) ∗ (x1 ∗ del)))) (Pdel / del))) (del1 ∗ chain0))) (x1 ∗ x4 − x2 ∗ x4 − x1 ∗ x5 + (x2 ∗ x5 − x3 ∗ x6 + (x1 + (x2 − x3 + (x4 + (x5 − x6)))) ∗ x6)))) (x1 ∗ x4 − x3 ∗ x4 − x2 ∗ x5 − x1 ∗ x6 + (x3 ∗ x6 + x5 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6))))))) (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) (delta_x x1 x2 x3 x4 x5 x6))) (real_of_nat (2::nat) ∗ x1 − x2 − x3 + (real_of_nat (2::nat) ∗ x4 − x5 − x6)))) (x1 ∗ x2 − x1 ∗ x4 − x2 ∗ x5 + (x4 ∗ x5 − x3 ∗ (x1 + (x2 − x3 + (x4 + (x5 − x6)))) + x3 ∗ x6)))) (x1 ∗ x3 − x1 ∗ x4 + (x2 ∗ x5 − x3 ∗ x6 + (x4 ∗ x6 − x2 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6)))))))) (− n4))) (x2 ∗ x3 + (x1 ∗ x4 − x2 ∗ x5 − x3 ∗ x6 + (x5 ∗ x6 − x1 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) ((du135 / u135 +

du126 / u126) ∗ chain0 + (1::real) / _2379305))) (dua x1 x2 x6))) (dua x1 x3 x5))) (real_of_nat (4::nat) ∗ (u135 ∗ (u126 ∗ (u234 ∗ (_2379305 ∗ (_2379306 ∗ _2379307)))))))) (ups_x x2 x3 x4))) (ups_x x1 x2 x6))) (ups_x x1 x3 x5))) (real_of_nat (4::nat) ∗ x1))) (real_of_nat (2::nat)))) (real_of_nat (2::nat) ∗ _2379305))) (_2379310 ∗ _2379310))) (_2379309 ∗ _2379309))) (_2379308 ∗ _2379308))) (_2379307 ∗ _2379307))) (_2379306 ∗ _2379306))) (_2379305 ∗ _2379305))

thm Mdtau.mdtau2uf_y:

∀(y6::real) (y5::real) (y4::real) (y3::real) (y2::real) y1::real. mdtau2uf_y y1 y2 y3 y4 y5 y6 = LET (λx1::real. LET_END (LET (λx2::real. LET_END (LET (λx3::real. LET_END (LET (λx4::real. LET_END (LET (λx5::real. LET_END (LET (λx6::real. LET_END (LET (λchain0::real. LET_END (LET (λPchain::real. LET_END (LET (λchain2::real. LET_END (LET (λu135::real. LET_END (LET (λu126::real. LET_END (LET (λu234::real. LET_END (LET (λuf::real. LET_END (LET (λdu135::real. LET_END (LET (λdu126::real. LET_END (LET (λLuf::real. LET_END (LET (λn4::real. LET_END (LET (λdel4::real. LET_END (LET (λn5::real. LET_END (LET (λn6::real. LET_END (LET (λDn4::real. LET_END (LET (λdel::real. LET_END (LET (λdel1::real. LET_END (LET (λdel2::real. LET_END (LET (λdel3::real. LET_END (LET (λPdel::real. LET_END (LET (λLdel::real. LET_END (LET (λsd4::real. LET_END (LET (λsd5::real. LET_END (LET (λsd6::real. LET_END (LET (λDsd4::real. LET_END (LET (λm4diff::real. LET_END (LET (λm4::real. LET_END (LET (λm5::real. LET_END (LET (λm6::real. LET_END (LET (λconst1::real. LET_END (LET (λrhoy1::real. LET_END (LET (λrhoy2::real. LET_END (LET (λrhoy3::real. LET_END (LET (λPrhoy1::real. LET_END (LET (λrr::real. LET_END (LET (λD2n4::real. LET_END (LET (λD2sd4::real. LET_END (LET (λDm4diff::real. LET_END (LET (λPm4::real. LET_END (LET (λDdel3::real. LET_END (LET (λDdel2::real. LET_END (LET (λPm5::real. LET_END (LET (λPm6::real. LET_END (LET (λPrrC::real. LET_END (LET (λP2tauNum::real. LET_END (LET LET_END (P2tauNum / safesqrt del))) (PrrC + (− Luf − DECIMAL (5::nat) (10::nat) ∗ Ldel) ∗ rr))) (real_of_nat (2::nat) ∗ (Prhoy1 ∗ m4) + (rhoy1 ∗ Pm4 + (rhoy2 ∗ Pm5 + rhoy3 ∗ Pm6))))) ((Ddel2 ∗ (x1 ∗ u126) + (del2 ∗ ((1::real) ∗ u126) + del2 ∗ (x1 ∗ du126))) ∗ (chain0 ∗ (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (real_of_nat (2::nat) ∗ y2)))))))) ((Ddel3 ∗ (x1 ∗ u135) + (del3 ∗ ((1::real) ∗ u135) + del3 ∗ (x1 ∗ du135))) ∗ (chain0 ∗ (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (real_of_nat (2::nat) ∗ y3)))))))) (x4 + (x5 − x6)))) (x4 − x5 + x6))) ((Dm4diff ∗ chain2 + m4diff ∗ Pchain) ∗ (u234 ∗ (y2 ∗ y3))))) (real_of_nat (2::nat) ∗ (D2n4 ∗ sd4) + (Dn4 ∗ Dsd4 − n4 ∗ D2sd4)))) (− real_of_nat (8::nat) ∗ (x1 ∗ x4) + real_of_nat (8::nat) ∗ (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6)))))))))) (real_of_nat (2::nat)))) (rhoy1 ∗ m4 + (rhoy2 ∗ m5 + rhoy3 ∗ m6)))) (const1 / DECIMAL (52::nat) (100::nat)))) (rho y3))) (rho y2))) (rho y1))) (sol_y (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) / pi))) (− real_of_nat (4::nat) ∗ (x3 ∗ (u234 ∗ (del2

∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u126 ∗ y2))))))))) (− real_of_nat (4::nat) ∗ (x2 ∗ (u234 ∗ (del3 ∗ (real_of_nat (2::nat) ∗ (x1 ∗ (u135 ∗ y3))))))))) (m4diff ∗ (chain0 ∗ (u234 ∗ (y2 ∗ y3)))))) (real_of_nat (2::nat) ∗ (Dn4 ∗ sd4) − n4 ∗ Dsd4))) (real_of_nat (4::nat) ∗ del + real_of_nat (4::nat) ∗ (x1 ∗ del1)))) (real_of_nat (4::nat) ∗ (x3 ∗ del)))) (real_of_nat (4::nat) ∗ (x2 ∗ del)))) (real_of_nat (4::nat) ∗ (x1 ∗ del)))) (Pdel / del))) (del1 ∗ chain0))) (x1 ∗ x4 − x2 ∗ x4 − x1 ∗ x5 + (x2 ∗ x5 − x3 ∗ x6 + (x1 + (x2 − x3 + (x4 + (x5 − x6)))) ∗ x6)))) (x1 ∗ x4 − x3 ∗ x4 − x2 ∗ x5 − x1 ∗ x6 + (x3 ∗ x6 + x5 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6))))))) (− (x1 ∗ x4) + (x2 ∗ x5 − x3 ∗ x5 − x2 ∗ x6 + (x3 ∗ x6 + x4 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) (delta_x x1 x2 x3 x4 x5 x6))) (real_of_nat (2::nat) ∗ x1 − x2 − x3 + (real_of_nat (2::nat) ∗ x4 − x5 − x6)))) (x1 ∗ x2 − x1 ∗ x4 − x2 ∗ x5 + (x4 ∗ x5 − x3 ∗ (x1 + (x2 − x3 + (x4 + (x5 − x6)))) + x3 ∗ x6)))) (x1 ∗ x3 − x1 ∗ x4 + (x2 ∗ x5 − x3 ∗ x6 + (x4 ∗ x6 − x2 ∗ (x1 − x2 + (x3 + (x4 − x5 + x6)))))))) (− n4))) (x2 ∗ x3 + (x1 ∗ x4 − x2 ∗ x5 − x3 ∗ x6 + (x5 ∗ x6 − x1 ∗ (− x1 + (x2 + (x3 − x4 + (x5 + x6))))))))) ((du135 / u135 + du126 / u126) ∗ chain0 + (1::real) / y1))) (dua x1 x2 x6))) (dua x1 x3 x5))) (real_of_nat (4::nat) ∗ (u135 ∗ (u126 ∗ (u234 ∗ (y1 ∗ (y2 ∗ y3))))))) (ups_x x2 x3 x4))) (ups_x x1 x2 x6))) (ups_x x1 x3 x5))) (real_of_nat (4::nat) ∗ x1))) (real_of_nat (2::nat)))) (real_of_nat (2::nat) ∗ y1))) (y6 ∗ y6))) (y5 ∗ y5))) (y4 ∗ y4))) (y3 ∗ y3))) (y2 ∗ y2))) (y1 ∗ y1)

thm Mdtau.mdtau_y_LC:

mdtau_y_LC = mdtau_y

thm Mdtau.mdtau2uf_y_LC:

mdtau2uf_y_LC = mdtau2uf_y

thm DEF_mdtau:

mdtau = (λ(_2379365::real) (_2379366::real) (_2379367::real) (_2379368::real) (_2379369::real) _2379370::real. 0::real)

thm Main_estimate_ineq.mdtau_fake:

∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. mdtau y1 y2 y3 y4 y5 y6 = (0::real)

thm DEF_mdtau2:

mdtau2 = (λ(_2379425::real) (_2379426::real) (_2379427::real) (_2379428::real) (_2379429::real) _2379430::real. 0::real)

thm Main_estimate_ineq.mdtau2_fake:

∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. mdtau2 y1 y2 y3 y4 y5 y6 = (0::real)

thm Nonlinear_lemma.sqrt_x1:

sqrt_x1 (?x1.0::real) (?x2.0::?'e::type) (?x3.0::?'d::type) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) = sqrt ?x1.0

thm Nonlinear_lemma.sqrt_x2:

sqrt_x2 (?x1.0::?'e::type) (?x2.0::real) (?x3.0::?'d::type) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) = sqrt ?x2.0

thm Nonlinear_lemma.sqrt_x3:

sqrt_x3 (?x1.0::?'e::type) (?x2.0::?'d::type) (?x3.0::real) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) = sqrt ?x3.0

thm Nonlinear_lemma.sqrt_x4:

sqrt_x4 (?x1.0::?'e::type) (?x2.0::?'d::type) (?x3.0::?'c::type) (?x4.0::real) (?x5.0::?'b::type) (?x6.0::?'a::type) = sqrt ?x4.0

thm Nonlinear_lemma.sqrt_x5:

sqrt_x5 (?x1.0::?'e::type) (?x2.0::?'d::type) (?x3.0::?'c::type) (?x4.0::?'b::type) (?x5.0::real) (?x6.0::?'a::type) = sqrt ?x5.0

thm Nonlinear_lemma.sqrt_x6:

sqrt_x6 (?x1.0::?'e::type) (?x2.0::?'d::type) (?x3.0::?'c::type) (?x4.0::?'b::type) (?x5.0::?'a::type) (?x6.0::real) = sqrt ?x6.0

thm Nonlinear_lemma.halfbump_x1:

∀ *(x2::?'e::type) (x3::?'d::type) (x4::?'c::type) (x5::?'b::type) (x6::?'a::type) x1::real. halfbump_x1 x1 x2 x3 x4 x5 x6 = halfbump_x x1*

thm Nonlinear_lemma.halfbump_x4:

∀ *(x1::?'e::type) (x2::?'d::type) (x3::?'c::type) (x5::?'b::type) (x6::?'a::type) x4::real. halfbump_x4 x1 x2 x3 x4 x5 x6 = halfbump_x x4*

thm Functional_equation.unit6:

unit6 (?x1.0::?'f::type) (?x2.0::?'e::type) (?x3.0::?'d::type) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) = (1::real)

thm Nonlinear_lemma.promote:

promote (?f::?'f::type ⇒ ?'g::type) (?x1.0::?'f::type) (?x2.0::?'e::type) (?x3.0::?'d::type) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) = ?f ?x1.0

thm Nonlinear_lemma.tame_table_d_values:

tame_table_d (2::nat) (1::nat) = DECIMAL (103::nat) (1000::nat) ∧ tame_table_d (1::nat) (2::nat) = DECIMAL (2759::nat) (10000::nat) ∧ tame_table_d (0::nat) (3::nat) = DECIMAL (4488::nat) (10000::nat) ∧ tame_table_d (4::nat) (1::nat) = DECIMAL (6548::nat) (10000::nat) ∧ tame_table_d (6::nat) (0::nat) = DECIMAL (7578::nat) (10000::nat) ∧ tame_table_d (3::nat) (1::nat) = DECIMAL (3789::nat) (10000::nat) ∧ tame_table_d (2::nat) (2::nat) = DECIMAL (5518::nat) (10000::nat) ∧ tame_table_d (1::nat) (3::nat) = DECIMAL (7247::nat) (10000::nat) ∧ tame_table_d (0::nat) (4::nat) = DECIMAL (8976::nat) (10000::nat) ∧ tame_table_d (5::nat) (0::nat) = DECIMAL (4819::nat) (10000::nat)

$\wedge$ *tame_table_d (4::nat) (1::nat) = DECIMAL (6548::nat) (10000::nat)* $\wedge$ *tame_table_d (3::nat) (2::nat) = DECIMAL (8277::nat) (10000::nat)* $\wedge$ *tame_table_d (2::nat) (3::nat) = DECIMAL (10006::nat) (10000::nat)*

thm Nonlinear_lemma.unit0f:

*(?f::?'f::type ⇒ ?'e::type ⇒ ?'d::type ⇒ ?'c::type ⇒ ?'b::type ⇒ ?'a::type ⇒ real) (?x1.0::?'f::type) (?x2.0::?'e::type) (?x3.0::?'d::type) (?x4.0::?'c::type) (?x5.0::?'b::type) (?x6.0::?'a::type) * unit0 = ?f ?x1.0 ?x2.0 ?x3.0 ?x4.0 ?x5.0 ?x6.0*

thm Nonlinear_lemma.sqrt8_sqrt2:

*sqrt8 = real_of_nat (2::nat) * sqrt2*

thm Nonlinear_lemma.sqrt2_sqrt8:

*sqrt2 = DECIMAL (5::nat) (10::nat) * sqrt8*

thm Nonlinear_lemma.SQRT_MUL_POW_2:

$\forall$ *(a::real) b::real. (0::real)* $\leq$ *a* $\wedge$ *(0::real)* $\leq$ *b* $\longrightarrow$ *sqrt (a * a * b) = a * sqrt b*

thm Tame_general.sol0_EQ_sol_y:

sol0 = sol_y (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat))

thm Tame_general.sol0_over_pi_EQ_const1:

sol0 / pi = const1

thm Nonlinear_lemma.sol0_const1:

*sol0 = pi * const1*

thm Nonlinear_lemma.ineq_lemma_b:

$\forall$ *(a::real) (y::real) b::real. (0::real)* $\leq$ *a* $\wedge$ *(0::real)* $\leq$ *b* $\wedge$ *a* $\leq$ *y* $\wedge$ *y* $\leq$ *b* $\longrightarrow$ $a^2 \leq y^2 \wedge y^2 \leq b^2$ $\wedge$ *sqrt* $(y^2) = y$

thm Sphere.ineq_conjunct1:

$\forall$ *(a::real) (x::real) (b::real) (xs::(real × real × real) list) c::bool. ineq ((a, x, b) # xs) c = (a* $\leq$ *x* $\wedge$ *x* $\leq$ *b* $\longrightarrow$ *ineq xs c)*

thm Sphere.ineq_conjunct0:

$\forall$ *c::bool. ineq [] c = c*

thm Nonlinear_lemma.ineq_square2:

((0::real) $\leq$ *(?a1.0::real)* $\wedge$ *(0::real)* $\leq$ *(?a2.0::real)* $\wedge$ *(0::real)* $\leq$ *(?a3.0::real)* $\wedge$ *(0::real)* $\leq$ *(?a4.0::real)* $\wedge$ *(0::real)* $\leq$ *(?a5.0::real)* $\wedge$ *(0::real)* $\leq$ *(?a6.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b1.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b2.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b3.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b4.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b5.0::real)* $\wedge$ *(0::real)* $\leq$ *(?b6.0::real))*

$\wedge$ ($\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ineq* [(*?a1.0*^{2}, *x1*, *?b1.0*^{2}), (*?a2.0*^{2}, *x2*, *?b2.0*^{2}), (*?a3.0*^{2}, *x3*, *?b3.0*^{2}), (*?a4.0*^{2}, *x4*, *?b4.0*^{2}), (*?a5.0*^{2}, *x5*, *?b5.0*^{2}), (*?a6.0*^{2}, *x6*, *?b6.0*^{2})] ((*?P*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *bool*) (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*))) $\longrightarrow$ ($\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *ineq* [(*?a1.0*, *y1*, *?b1.0*), (*?a2.0*, *y2*, *?b2.0*), (*?a3.0*, *y3*, *?b3.0*), (*?a4.0*, *y4*, *?b4.0*), (*?a5.0*, *y5*, *?b5.0*), (*?a6.0*, *y6*, *?b6.0*)] (*?P y1 y2 y3 y4 y5 y6*))

thm Nonlinear_lemma.ineq_square2_9:

((*0*::*real*) $\leq$ (*?a1.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a2.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a3.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a4.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a5.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a6.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a7.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a8.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?a9.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b1.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b2.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b3.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b4.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b5.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b6.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b7.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b8.0*::*real*) $\wedge$ (*0*::*real*) $\leq$ (*?b9.0*::*real*)) $\wedge$ ($\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x7*::*real*) (*x8*::*real*) *x9*::*real*. *ineq* [(*?a1.0*^{2}, *x1*, *?b1.0*^{2}), (*?a2.0*^{2}, *x2*, *?b2.0*^{2}), (*?a3.0*^{2}, *x3*, *?b3.0*^{2}), (*?a4.0*^{2}, *x4*, *?b4.0*^{2}), (*?a5.0*^{2}, *x5*, *?b5.0*^{2}), (*?a6.0*^{2}, *x6*, *?b6.0*^{2}), (*?a7.0*^{2}, *x7*, *?b7.0*^{2}), (*?a8.0*^{2}, *x8*, *?b8.0*^{2}), (*?a9.0*^{2}, *x9*, *?b9.0*^{2})] ((*?P*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *bool*) (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) (*sqrt x7*) (*sqrt x8*) (*sqrt x9*))) $\longrightarrow$ ($\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*y7*::*real*) (*y8*::*real*) *y9*::*real*. *ineq* [(*?a1.0*, *y1*, *?b1.0*), (*?a2.0*, *y2*, *?b2.0*), (*?a3.0*, *y3*, *?b3.0*), (*?a4.0*, *y4*, *?b4.0*), (*?a5.0*, *y5*, *?b5.0*), (*?a6.0*, *y6*, *?b6.0*), (*?a7.0*, *y7*, *?b7.0*), (*?a8.0*, *y8*, *?b8.0*), (*?a9.0*, *y9*, *?b9.0*)] (*?P y1 y2 y3 y4 y5 y6 y7 y8 y9*))

thm Nonlinear_lemma.sqrt8_nn:

(*0*::*real*) $\leq$ *sqrt8*

thm Nonlinear_lemma.sqrt2_nn:

(*0*::*real*) $\leq$ *sqrt2*

thm Nonlinear_lemma.sqrt3_nn:

(*0*::*real*) $\leq$ *sqrt* (*real_of_nat* (*3*::*nat*))

thm Nonlinear_lemma.abc_quadratic:

abc_of_quadratic (λ*x*::*real*. (*?a*::*real*) $*$ *x*^{2} + ((*?b*::*real*) $*$ *x* + (*?c*::*real*))) = (*?a*, *?b*, *?c*)

thm Nonlinear_lemma.delta_quadratic:

$-$ *delta_x* (*?x1.0*::*real*) (*?x2.0*::*real*) (*?x3.0*::*real*) (*?x4.0*::*real*) (*?x5.0*::*real*) (*?x6.0*::*real*) = *?x1.0* $*$ *?x4.0*^{2} + ((*?x1.0* $*$ *?x1.0* + ((*?x3.0* $-$ *?x5.0*) $*$ (*?x2.0* $-$ *?x6.0*) $-$ *?x1.0* $*$ (*?x2.0* + (*?x3.0* + (*?x5.0* + *?x6.0*)))))) $*$ *?x4.0* + (*?x1.0* $*$ (*?x3.0* $*$ *?x5.0*) + (*?x1.0* $*$ (*?x2.0* $*$ *?x6.0*) $-$ *?x3.0* $*$ ((*?x1.0* + (*?x2.0* $-$ *?x3.0* + (*?x5.0* $-$ *?x6.0*))) $*$ *?x6.0*) $-$ *?x2.0* $*$ (*?x5.0* $*$ (*?x1.0* $-$ *?x2.0* + (*?x3.0* $-$ *?x5.0* + *?x6.0*))))))

thm Nonlinear_lemma.edge_flat_rewrite:

∀ (y1::real) (y2::real) (y3::real) (y5::real) y6::real. edge_flat y1 y2 y3 y5 y6 =
sqrt (quadratic_root_plus (y1 * y1, y1 * y1 * (y1 * y1) + ((y3 * y3 − y5 *
y5) * (y2 * y2 − y6 * y6) − y1 * y1 * (y2 * y2 + (y3 * y3 + (y5 * y5 +
y6 * y6)))), y1 * y1 * (y3 * y3 * (y5 * y5)) + (y1 * y1 * (y2 * y2 * (y6 *
y6)) − y3 * y3 * ((y1 * y1 + (y2 * y2 − y3 * y3 + (y5 * y5 − y6 * y6)))
* (y6 * y6)) − y2 * y2 * (y5 * y5 * (y1 * y1 − y2 * y2 + (y3 * y3 − y5 *
y5 + y6 * y6))))))

thm Nonlinear_lemma.enclosed_rewrite:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) (y7::real) (y8::real)
y9::real. enclosed y1 y2 y3 y4 y5 y6 y7 y8 y9 = sqrt (quadratic_root_plus (−
(1::real) * (y6 * y6)^2 + (− (1::real) * (y5 * y5 + − (1::real) * (y4 * y4))^2 +
real_of_nat (2::nat) * (y6 * y6 * (y5 * y5 + y4 * y4))), real_of_nat (2::nat)
* ((y5 * y5)^2 * (y2 * y2 + y8 * y8) + ((y6 * y6)^2 * (y3 * y3 + y9 * y9) +
(y4 * y4 * (y7 * y7 * (y4 * y4 + (− (1::real) * (y2 * y2) + − (1::real) * (y3
* y3))) + (y8 * y8 * (y3 * y3) + (y1 * y1 * (real_of_nat (2::nat) * (y7 *
y7) + (y4 * y4 + (− (1::real) * (y8 * y8) + − (1::real) * (y9 * y9)))) + y2
* y2 * (y9 * y9)))) + (− (1::real) * (y5 * y5 * (y4 * y4 * (y2 * y2) + (y4 *
y4 * (y8 * y8) + (− real_of_nat (2::nat) * (y2 * y2 * (y8 * y8)) + (y7 * y7
* (y4 * y4 + (y2 * y2 + − (1::real) * (y3 * y3))) + (y8 * y8 * (y3 * y3) +
(y1 * y1 * (y4 * y4 + (y8 * y8 + − (1::real) * (y9 * y9))) + y2 * y2 * (y9
* y9)))))))) + y6 * y6 * (real_of_nat (2::nat) * (y5 * y5 * (y4 * y4)) + (−
(1::real) * (y1 * y1 * (y4 * y4)) + (− (1::real) * (y7 * y7 * (y4 * y4)) + (−
(1::real) * (y5 * y5 * (y2 * y2)) + (y7 * y7 * (y2 * y2) + (− (1::real) * (y5
* y5 * (y8 * y8)) + (y1 * y1 * (y8 * y8) + (− (1::real) * (y5 * y5 * (y3 *
y3)) + (− (1::real) * (y7 * y7 * (y3 * y3)) + (− (1::real) * (y4 * y4 * (y3 *
y3)) + (− (1::real) * (y8 * y8 * (y3 * y3)) + (− (1::real) * (y5 * y5 * (y9 *
y9)) + (− (1::real) * (y1 * y1 * (y9 * y9)) + (− (1::real) * (y4 * y4 * (y9 *
y9)) + (− (1::real) * (y2 * y2 * (y9 * y9)) + real_of_nat (2::nat) * (y3 * y3
* (y9 * y9)))))))))))))))))))))), − (1::real) * ((y7 * y7)^2 * ((y4 * y4)^2 + ((y2
* y2 + − (1::real) * (y3 * y3))^2 + − real_of_nat (2::nat) * (y4 * y4 * (y2
* y2 + y3 * y3))))) + (− (1::real) * (y5 * y5 * (y2 * y2) + (− (1::real) *
(y5 * y5 * (y8 * y8)) + (− (1::real) * (y6 * y6 * (y3 * y3)) + (y8 * y8 *
(y3 * y3) + (y6 * y6 * (y9 * y9) + − (1::real) * (y2 * y2 * (y9 * y9)))))))^2
+ (− (1::real) * ((y1 * y1)^2 * ((y4 * y4)^2 + ((y8 * y8 + − (1::real) * (y9 *
y9))^2 + − real_of_nat (2::nat) * (y4 * y4 * (y8 * y8 + y9 * y9))))) + (−
real_of_nat (2::nat) * (y7 * y7 * (− real_of_nat (2::nat) * (y4 * y4 * (y2 *
y2 * (y3 * y3))) + (y4 * y4 * (y8 * y8 * (y3 * y3)) + (y2 * y2 * (y8 * y8 *
(y3 * y3)) + (− (1::real) * (y8 * y8 * (y3 * y3)^2) + (y5 * y5 * (− (1::real)
* (y2 * y2)^2 + (y4 * y4 * (y2 * y2 + − (1::real) * (y8 * y8)) + (y8 * y8 *
(y3 * y3) + y2 * y2 * (y8 * y8 + (y3 * y3 + − real_of_nat (2::nat) * (y9
* y9)))))) + (y4 * y4 * (y2 * y2 * (y9 * y9)) + (− (1::real) * ((y2 * y2)^2 *
(y9 * y9)) + (y2 * y2 * (y3 * y3 * (y9 * y9)) + y6 * y6 * (y4 * y4 * (y3
* y3 + − (1::real) * (y9 * y9)) + (y3 * y3 * (− real_of_nat (2::nat) * (y8

$*\ y8) + (-\ (1::real) * (y3 * y3) + y9 * y9)) + y2 * y2 * (y3 * y3 + y9 * y9)))))))))))) + real\_of\_nat\ (2::nat) * (y1 * y1 * (y6 * y6 * (y4 * y4 * (y3 * y3)) + (-\ (1::real) * (y6 * y6 * (y8 * y8 * (y3 * y3))) + (-\ (1::real) * (y4 * y4 * (y8 * y8 * (y3 * y3))) + ((y8 * y8)^2 * (y3 * y3) + (-\ (1::real) * (y6 * y6 * (y4 * y4 * (y9 * y9))) + (real\_of\_nat\ (2::nat) * (y6 * y6 * (y2 * y2 * (y9 * y9))) + (-\ (1::real) * (y4 * y4 * (y2 * y2 * (y9 * y9))) + (-\ (1::real) * (y6 * y6 * (y8 * y8 * (y9 * y9))) + (real\_of\_nat\ (2::nat) * (y4 * y4 * (y8 * y8 * (y9 * y9))) + (-\ (1::real) * (y2 * y2 * (y8 * y8 * (y9 * y9))) + (-\ (1::real) * (y6 * y6 * (y3 * y3 * (y9 * y9))) + (-\ (1::real) * (y8 * y8 * (y3 * y3 * (y9 * y9))) + (y6 * y6 * (y9 * y9)^2 + (y2 * y2 * (y9 * y9)^2 + (y5 * y5 * (y4 * y4 * (y2 * y2 + -\ (1::real) * (y8 * y8)) + (y8 * y8 * (y8 * y8 + (real\_of\_nat\ (2::nat) * (y3 * y3) + -\ (1::real) * (y9 * y9))) + -\ (1::real) * (y2 * y2 * (y8 * y8 + y9 * y9)))) + y7 * y7 * ((y4 * y4)^2 + ((y2 * y2 + -\ (1::real) * (y3 * y3)) * (y8 * y8 + -\ (1::real) * (y9 * y9)) + -\ (1::real) * (y4 * y4 * (y2 * y2 + (y8 * y8 + (y3 * y3 + y9 * y9)))))))))))))))))))))))))))$

thm Nonlinear_lemma.quad_root_plus_curry:

$\forall (a::real)\ (b::real)\ c::real.\ quadratic\_root\_plus\_curry\ a\ b\ c = (-\ b + sqrt\ (b^2 - real\_of\_nat\ (4::nat) * (a * c)))\ /\ (real\_of\_nat\ (2::nat) * a)$

thm Nonlinear_lemma.y_of_x_e:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ y\_of\_x\ (?f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow ?'a::type)\ y1\ y2\ y3\ y4\ y5\ y6 = ?f\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.vol_y_e:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ vol\_y\ y1\ y2\ y3\ y4\ y5\ y6 = y\_of\_x\ vol\_x\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.rad2_y_e:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ rad2\_y\ y1\ y2\ y3\ y4\ y5\ y6 = y\_of\_x\ rad2\_x\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.rad2_x_y:

$\forall (a1::real)\ (a2::real)\ (a3::real)\ (a4::real)\ (a5::real)\ (a6::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow rad2\_y\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6) = rad2\_x\ x1\ x2\ x3\ x4\ x5\ x6$

thm Nonlinear_lemma.delta_x4_delta4_y:

$\forall (a1::real)\ (a2::real)\ (a3::real)\ (a4::real)\ (a5::real)\ (a6::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow delta4\_y\ (sqrt\ x1)\ (sqrt\ x2)\ (sqrt\ x3)\ (sqrt\ x4)\ (sqrt\ x5)\ (sqrt\ x6) = delta\_x4\ x1\ x2\ x3\ x4\ x5\ x6$

thm Nonlinear_lemma.dih_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *dih_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *dih_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.dih2_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *dih2_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *dih2_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.dih3_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *dih3_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *dih3_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.delta_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *delta_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *delta_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.upper_dih_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *upper_dih_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *upper_dih_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.vol_x_y:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*a6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. $a1^2 \leq x1 \wedge a2^2 \leq x2 \wedge a3^2 \leq x3 \wedge a4^2 \leq x4 \wedge a5^2 \leq x5 \wedge a6^2 \leq x6 \longrightarrow$ *vol_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *vol_x x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.sqrt8_2:

sqrt8 * *sqrt8* = *DECIMAL* (*80*::*nat*) (*10*::*nat*)

thm Nonlinear_lemma.dih_x_sym:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih_x x1 x2 x3 x4 x5 x6* = *dih_x x1 x3 x2 x4 x6 x5*

thm Nonlinear_lemma.dih_x_sym2:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih_x x1 x2 x3 x4 x5 x6* = *dih_x x1 x5 x6 x4 x2 x3*

thm Nonlinear_lemma.dih_y_sym:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *dih_y y1 y2 y3 y4 y5 y6* = *dih_y y1 y3 y2 y4 y6 y5*

thm Nonlinear_lemma.dih_y_sym2:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *dih_y y1 y2 y3 y4 y5 y6* = *dih_y y1 y5 y6 y4 y2 y3*

thm Nonlinear_lemma.sol_y_123:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *sol_y y1 y2 y3 y4 y5 y6* = *dih_y y1 y2 y3 y4 y5 y6* + (*dih2_y y1 y2 y3 y4 y5 y6* + (*dih3_y y1 y2 y3 y4 y5 y6* − *pi*))

thm Nonlinear_lemma.rhazim2_alt:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *rhazim2 y1 y2 y3 y4 y5 y6* = *rho y2* ∗ *dih2_y y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.rhazim3_alt:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *rhazim3 y1 y2 y3 y4 y5 y6* = *rho y3* ∗ *dih3_y y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.taum_123:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *taum y1 y2 y3 y4 y5 y6* = *rhazim y1 y2 y3 y4 y5 y6* + (*rhazim2 y1 y2 y3 y4 y5 y6* + (*rhazim3 y1 y2 y3 y4 y5 y6* − ((*1*::*real*) + *const1*) ∗ *pi*))

thm Nonlinear_lemma.tauq_x_y:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) (*x6*::*real*) (*x7*::*real*) (*x8*::*real*) *x9*::*real*. *tauq* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) (*sqrt x7*) (*sqrt x8*) (*sqrt x9*) = *taum_x x1 x2 x3 x4 x5 x6* + *taum_x x7 x2 x3 x4 x8 x9*

thm Nonlinear_lemma.rho_alt:

∀ *y*::*real*. *rho y* = (*1*::*real*) + *const1* ∗ ((*y* − *real_of_nat* (*2*::*nat*)) / *DECIMAL* (*52*::*nat*) (*100*::*nat*))

thm Nonlinear_lemma.rho_sqrtx:

∀ *x*::*real*. *rho* (*sqrt x*) = (*1*::*real*) + *const1* ∗ ((*sqrt x* − *real_of_nat* (*2*::*nat*)) / *DECIMAL* (*52*::*nat*) (*100*::*nat*))

thm Nonlinear_lemma.lfun_ly:

∀ *h*::*real*. *lfun h* = *ly* (*real_of_nat* (*2*::*nat*) ∗ *h*)

thm Nonlinear_lemma.lfun1:

lfun (*1*::*real*) = (*1*::*real*)

thm Nonlinear_lemma.beta_bump_force_x:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *beta_bump_force_y* (*sqrt x1*) (*sqrt x2*) (*sqrt x3*) (*sqrt x4*) (*sqrt x5*) (*sqrt x6*) = *halfbump_x1 x1 x2 x3 x4 x5 x6* − *halfbump_x4 x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.halfbump_x_expand:

$\forall$ *x*≥*0*::*real*. *halfbump_x x* = − (*real_of_nat* (*4398119*::*nat*) / *real_of_nat* (*2376200*::*nat*)) + (*real_of_nat* (*17500*::*nat*) / *real_of_nat* (*11881*::*nat*) ∗ *sqrt x* − *real_of_nat* (*31250*::*nat*) / *real_of_nat* (*106929*::*nat*) ∗ *x*)

thm Nonlinear_lemma.vol4f_palt:

$\forall$ (*f*::*real* ⇒ *real*) (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *vol4f y1 y2 y3 y4 y5 y6 f* = − *real_of_nat* (*8*::*nat*) ∗ *mm1* + (*real_of_nat* (*4*::*nat*) ∗ (*mm1* / *pi*) ∗ (*dih_y y1 y2 y3 y4 y5 y6* + (*dih2_y y1 y2 y3 y4 y5 y6* + (*dih3_y y1 y2 y3 y4 y5 y6* + (*dih4_y y1 y2 y3 y4 y5 y6* + (*dih5_y y1 y2 y3 y4 y5 y6* + *dih6_y y1 y2 y3 y4 y5 y6*))))) + − *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*f* (*y1* / *real_of_nat* (*2*::*nat*)) ∗ *dih_y y1 y2 y3 y4 y5 y6* + (*f* (*y2* / *real_of_nat* (*2*::*nat*)) ∗ *dih2_y y1 y2 y3 y4 y5 y6* + (*f* (*y3* / *real_of_nat* (*2*::*nat*)) ∗ *dih3_y y1 y2 y3 y4 y5 y6* + (*f* (*y4* / *real_of_nat* (*2*::*nat*)) ∗ *dih4_y y1 y2 y3 y4 y5 y6* + (*f* (*y5* / *real_of_nat* (*2*::*nat*)) ∗ *dih5_y y1 y2 y3 y4 y5 y6* + *f* (*y6* / *real_of_nat* (*2*::*nat*)) ∗ *dih6_y y1 y2 y3 y4 y5 y6*))))))

thm Nonlinear_lemma.lmfun0:

$\forall$ *y*≥*real_of_nat* (*2*::*nat*) ∗ *h0*. *lmfun* (*y* / *real_of_nat* (*2*::*nat*)) = (*0*::*real*)

thm Nonlinear_lemma.lmfun_lfun:

$\forall$ *y*≤*real_of_nat* (*2*::*nat*) ∗ *h0*. *lmfun* (*y* / *real_of_nat* (*2*::*nat*)) = *lfun* (*y* / *real_of_nat* (*2*::*nat*))

thm Nonlinear_lemma.lmfun_lfun2:

$\forall$ *y*≤*h0*. *lmfun y* = *lfun y*

thm Nonlinear_lemma.edge_flat2_x_rewrite:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*?'a*::*type*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x2* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5* ∧ (*0*::*real*) ≤ *x6* ⟶ *edge_flat2_x x1 x2 x3 x4 x5 x6* = (*sqrt* (*quadratic_root_plus* (*x1*, *x1* ∗ *x1* + ((*x3* − *x5*) ∗ (*x2* − *x6*) − *x1* ∗ (*x2* + (*x3* + (*x5* + *x6*)))), *x1* ∗ (*x3* ∗ *x5*) + (*x1* ∗ (*x2* ∗ *x6*) − *x3* ∗ ((*x1* + (*x2* − *x3* + (*x5* − *x6*))) ∗ *x6*) − *x2* ∗ (*x5* ∗ (*x1* − *x2* + (*x3* − *x5* + *x6*)))))))^2

thm Nonlinear_lemma.edge_quadratic:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x5*::*real*) *x6*::*real*. *quadratic_root_plus* (*x1*, *x1* ∗ *x1* + ((*x3* − *x5*) ∗ (*x2* − *x6*) − *x1* ∗ (*x2* + (*x3* + (*x5* + *x6*)))), *x1* ∗ (*x3* ∗ *x5*) + (*x1* ∗ (*x2* ∗ *x6*) − *x3* ∗ ((*x1* + (*x2* − *x3* + (*x5* − *x6*))) ∗ *x6*) − *x2* ∗ (*x5* ∗ (*x1* − *x2* + (*x3* − *x5* + *x6*)))))) = (− (*x1* ∗ *x1*) + (*x1* ∗ *x2* + (*x1* ∗ *x3* − *x2* ∗ *x3* + (*x1* ∗ *x5* + (*x2* ∗ *x5* + (*x1* ∗ *x6* + (*x3* ∗ *x6* − *x5* ∗ *x6* + *sqrt* (*ups_x x1 x3 x5* ∗ *ups_x x1 x2 x6*)))))))) / (*real_of_nat* (*2*::*nat*) ∗ *x1*)

thm Nonlinear_lemma.quartic_has_real_derivative:

$\forall (x::real)\ (c0::real)\ (c1::real)\ (c2::real)\ (c3::real)\ c4::real.\ has\_real\_derivative\ (\lambda x::real.\ c0 * (1::real) + (c1 * x^{1::nat} + (c2 * x^2 + (c3 * x^{3::nat} + c4 * x^{4::nat}))))\ (c0 * (0::real) + (c1 * ((1::real) * (x^{(1::nat) - (1::nat)} * (1::real))) + (c2 * (real\_of\_nat\ (2::nat) * x^{(2::nat) - (1::nat)} * (1::real)) + (c3 * (real\_of\_nat\ (3::nat) * x^{(3::nat) - (1::nat)} * (1::real)) + c4 * (real\_of\_nat\ (4::nat) * x^{(4::nat) - (1::nat)} * (1::real))))))\ (atreal\ x)$

thm Nonlinear_lemma.quartic_continuous_on:

$\forall (s::real \Rightarrow bool)\ (c0::real)\ (c1::real)\ (c2::real)\ (c3::real)\ c4::real.\ real\_continuous\_on\ (\lambda x::real.\ c0 * x^{0::nat} + (c1 * x^{1::nat} + (c2 * x^2 + (c3 * x^{3::nat} + c4 * x^{4::nat}))))\ s$

thm Nonlinear_lemma.marchal_minus_lfun:

$\forall h::real.\ marchal\_quartic\ h - lfun\ h = inverse\_class.inverse\ (real\_of\_nat\ (65::nat) * (real\_of\_nat\ (1627::nat) * (sqrt\ (real\_of\_nat\ (2::nat)) - (1::real)))) * ((h - (1::real)) * ((- real\_of\_nat\ (512505::nat) + real\_of\_nat\ (770958::nat) * sqrt\ (real\_of\_nat\ (2::nat))) * h^{0::nat} + ((- real\_of\_nat\ (364208::nat) - real\_of\_nat\ (1295359::nat) * sqrt\ (real\_of\_nat\ (2::nat))) * h^{1::nat} + ((real\_of\_nat\ (1295359::nat) + real\_of\_nat\ (585000::nat) * sqrt\ (real\_of\_nat\ (2::nat))) * h^2 + - real\_of\_nat\ (585000::nat) * h^{3::nat}))))$

thm Nonlinear_lemma.hminus_cont:

$\forall s::real \Rightarrow bool.\ real\_continuous\_on\ (\lambda h::real.\ marchal\_quartic\ h - lfun\ h)\ s$

thm Nonlinear_lemma.hminus_exists:

$\exists x \geq DECIMAL\ (12::nat)\ (10::nat).\ x < DECIMAL\ (13::nat)\ (10::nat) \land marchal\_quartic\ x = lmfun\ x$

thm Nonlinear_lemma.hminus_prop:

$DECIMAL\ (12::nat)\ (10::nat) \leq hminus \land hminus < DECIMAL\ (13::nat)\ (10::nat) \land marchal\_quartic\ hminus = lmfun\ hminus$

thm Nonlinear_lemma.hminus_high:

$\forall h \geq h0.\ lmfun\ h = (0::real)$

thm Nonlinear_lemma.hminus_lt_h0:

$hminus < h0$

thm Nonlinear_lemma.h0_lt_hplus:

$h0 < hplus$

thm Nonlinear_lemma.hminus_prop_conjunct2:

$marchal\_quartic\ hminus = lmfun\ hminus$

thm Nonlinear_lemma.hminus_prop_conjunct1:

hminus < *DECIMAL* (*13*::*nat*) (*10*::*nat*)

thm Nonlinear_lemma.hminus_gt:

DECIMAL (*12*::*nat*) (*10*::*nat*) ≤ *hminus*

thm Nonlinear_lemma.lminus_ge_h0:

∀ *h*::*real*. *hplus* ≤ *h* ∧ *h* ≤ *sqrt* (*real_of_nat* (*2*::*nat*)) ⟶ *marchal_quartic h* ≤ (*0*::*real*)

thm Nonlinear_lemma.gcy_high:

∀ *y*≥*real_of_nat* (*2*::*nat*) ∗ *h0*. *gcy y* = *real_of_nat* (*4*::*nat*) ∗ (*mm1* / *pi*)

thm Nonlinear_lemma.gcy_low:

∀ *y*≤*real_of_nat* (*2*::*nat*) ∗ *h0*. *gcy y* = *gchi y*

thm Nonlinear_lemma.h0_lt_gt_conjunct6:

(*?y*::*real*) ≤ *real_of_nat* (*2*::*nat*) ∗ *hminus* ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*

thm Nonlinear_lemma.gcy_low_hminus:

∀ *y*≤*real_of_nat* (*2*::*nat*) ∗ *hminus*. *gcy y* = *gchi y*

thm Nonlinear_lemma.c2001:

∀ *y*≤*DECIMAL* (*2001*::*nat*) (*1000*::*nat*). *y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*

thm Nonlinear_lemma.gcy_low_const:

∀ *y*≤*DECIMAL* (*2001*::*nat*) (*1000*::*nat*). *gcy y* = *gchi y*

thm Nonlinear_lemma.gcy_high_hplus:

∀ *y*≥*real_of_nat* (*2*::*nat*) ∗ *hplus*. *gcy y* = *real_of_nat* (*4*::*nat*) ∗ (*mm1* / *pi*)

thm Nonlinear_lemma.vol4f_lmfun:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *vol4f y1 y2 y3 y4 y5 y6 lmfun* = − *real_of_nat* (*8*::*nat*) ∗ *mm1* + (*gcy y1* ∗ *dih_y y1 y2 y3 y4 y5 y6* + (*gcy y2* ∗ *dih2_y y1 y2 y3 y4 y5 y6* + (*gcy y3* ∗ *dih3_y y1 y2 y3 y4 y5 y6* + (*gcy y4* ∗ *dih4_y y1 y2 y3 y4 y5 y6* + (*gcy y5* ∗ *dih5_y y1 y2 y3 y4 y5 y6* + *gcy y6* ∗ *dih6_y y1 y2 y3 y4 y5 y6*)))))

thm Nonlinear_lemma.gamma4fgcy_alt:

gamma4fgcy (*?y1.0*::*real*) (*?y2.0*::*real*) (*?y3.0*::*real*) (*?y4.0*::*real*) (*?y5.0*::*real*) (*?y6.0*::*real*) *lmfun* = *vol_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* − (− *real_of_nat* (*8*::*nat*) ∗ *mm1* + (*gcy ?y1.0* ∗ *dih_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* + (*gcy ?y2.0* ∗ *dih2_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* + (*gcy ?y3.0* ∗ *dih3_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* + (*gcy ?y4.0* ∗ *dih4_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* + (*gcy ?y5.0* ∗ *dih5_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* + *gcy ?y6.0* ∗ *dih6_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0*))))))

thm Nonlinear_lemma.vol3f_palt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*r*::*real*) *f*::*real* $\Rightarrow$ *real*. *y3* = *r* $\wedge$ *y4* = *r* $\wedge$ *y5* = *r* $\longrightarrow$ *vol3f y1 y2 y6 r f* = *real_of_nat* (*2*::*nat*) * (*mm1* / *pi*) * (*real_of_nat* (*2*::*nat*) * *dih_y y1 y2 y3 y4 y5 y6* + (*real_of_nat* (*2*::*nat*) * *dih2_y y1 y2 y3 y4 y5 y6* + (*real_of_nat* (*2*::*nat*) * *dih6_y y1 y2 y3 y4 y5 y6* + (*dih3_y y1 y2 y3 y4 y5 y6* + (*dih4_y y1 y2 y3 y4 y5 y6* + (*dih5_y y1 y2 y3 y4 y5 y6* − *real_of_nat* (*3*::*nat*) * *pi*)))))) − *real_of_nat* (*8*::*nat*) * (*mm2* / *pi*) * (*f* (*y1* / *real_of_nat* (*2*::*nat*)) * *dih_y y1 y2 y3 y4 y5 y6* + (*f* (*y2* / *real_of_nat* (*2*::*nat*)) * *dih2_y y1 y2 y3 y4 y5 y6* + *f* (*y6* / *real_of_nat* (*2*::*nat*)) * *dih6_y y1 y2 y3 y4 y5 y6*))

thm Nonlinear_lemma.vol3f_135_palt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*r*::*real*) *f*::*real* $\Rightarrow$ *real*. *y2* = *r* $\wedge$ *y4* = *r* $\wedge$ *y6* = *r* $\longrightarrow$ *vol3f y1 y3 y5 r f* = *real_of_nat* (*2*::*nat*) * (*mm1* / *pi*) * (*real_of_nat* (*2*::*nat*) * *dih_y y1 y2 y3 y4 y5 y6* + (*real_of_nat* (*2*::*nat*) * *dih3_y y1 y2 y3 y4 y5 y6* + (*real_of_nat* (*2*::*nat*) * *dih5_y y1 y2 y3 y4 y5 y6* + (*dih2_y y1 y2 y3 y4 y5 y6* + (*dih4_y y1 y2 y3 y4 y5 y6* + (*dih6_y y1 y2 y3 y4 y5 y6* − *real_of_nat* (*3*::*nat*) * *pi*)))))) − *real_of_nat* (*8*::*nat*) * (*mm2* / *pi*) * (*f* (*y1* / *real_of_nat* (*2*::*nat*)) * *dih_y y1 y2 y3 y4 y5 y6* + (*f* (*y3* / *real_of_nat* (*2*::*nat*)) * *dih3_y y1 y2 y3 y4 y5 y6* + *f* (*y5* / *real_of_nat* (*2*::*nat*)) * *dih5_y y1 y2 y3 y4 y5 y6*))

thm Nonlinear_lemma.vol3r_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) *r*::*real*. *y3* = *r* $\wedge$ *y4* = *r* $\wedge$ *y5* = *r* $\longrightarrow$ *vol3r y1 y2 y6 r* = *vol_y y1 y2 y3 y4 y5 y6*

thm Tame_general.COS_PI3:

cos (*pi* / *real_of_nat* (*3*::*nat*)) = (*1*::*real*) / *real_of_nat* (*2*::*nat*)

thm Nonlinear_lemma.ACS_2:

acs ((*1*::*real*) / *real_of_nat* (*2*::*nat*)) = *pi* / *real_of_nat* (*3*::*nat*)

thm Nonlinear_lemma.sol0_POS:

(*0*::*real*) < *sol0*

thm Nonlinear_lemma.vol4f_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *vol4f y1 y2 y3 y4 y5 y6 lfun* = − *real_of_nat* (*8*::*nat*) * *mm1* + ((*real_of_nat* (*4*::*nat*) * (*mm1* / *pi*) − *real_of_nat* (*8*::*nat*) * (*mm2* * (((*1*::*real*) + *const1*) / (*pi* * *const1*)))) * (*dih_y y1 y2 y3 y4 y5 y6* + (*dih2_y y1 y2 y3 y4 y5 y6* + (*dih3_y y1 y2 y3 y4 y5 y6* + (*dih4_y y1 y2 y3 y4 y5 y6* + (*dih5_y y1 y2 y3 y4 y5 y6* + *dih6_y y1 y2 y3 y4 y5 y6*))))) + *real_of_nat* (*8*::*nat*) * (*mm2* / (*pi* * *const1*)) * (*rhazim y1 y2 y3 y4 y5 y6* + (*rhazim2 y1 y2 y3 y4 y5 y6* + (*rhazim3 y1 y2 y3 y4 y5 y6* + (*rhazim4 y1 y2 y3 y4 y5 y6* + (*rhazim5 y1 y2 y3 y4 y5 y6* + *rhazim6 y1 y2 y3 y4 y5 y6*))))))

thm Nonlinear_lemma.vol2f_marchal_pow_y:

$\forall (r::real)\ y::real.\ vol2f\ y\ r\ marchal\_quartic = LET\ (\lambda fac::real.\ LET\_END\ (real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (real\_of\_nat\ (2::nat) * pi) - real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (real\_of\_nat\ (2::nat) * (pi * (inverse\_class.inverse\ (r * real\_of\_nat\ (2::nat)) * y^{1::nat}))) - fac * (real\_of\_nat\ (3::nat) * (sqrt2 * hplus)) + (fac * ((DECIMAL\ (15::nat)\ (10::nat) * sqrt2 + (DECIMAL\ (15::nat)\ (10::nat) * hplus + DECIMAL\ (85::nat)\ (10::nat) * (sqrt2 * hplus))) * y^{1::nat}) + (fac * ((-\ DECIMAL\ (75::nat)\ (100::nat) - DECIMAL\ (85::nat)\ (10::nat) * (sqrt2 * inverse\_class.inverse\ (real\_of\_nat\ (2::nat))) - DECIMAL\ (85::nat)\ (10::nat) * (hplus * inverse\_class.inverse\ (real\_of\_nat\ (2::nat))) - real\_of\_nat\ (9::nat) * (hplus * (sqrt2 * inverse\_class.inverse\ (real\_of\_nat\ (4::nat))))) * y^2) + (fac * ((DECIMAL\ (170::nat)\ (10::nat) * inverse\_class.inverse\ (real\_of\_nat\ (8::nat)) + (DECIMAL\ (90::nat)\ (10::nat) * (sqrt2 * inverse\_class.inverse\ (real\_of\_nat\ (8::nat))) + DECIMAL\ (90::nat)\ (10::nat) * (hplus * inverse\_class.inverse\ (real\_of\_nat\ (8::nat))))) * y^{3::nat}) - fac * (DECIMAL\ (90::nat)\ (10::nat) * (inverse\_class.inverse\ (real\_of\_nat\ (16::nat)) * y^{4::nat})))))))\ (-\ (real\_of\_nat\ (8::nat) * (mm2\ /\ pi)) * (real\_of\_nat\ (2::nat) * (pi * inverse\_class.inverse\ (DECIMAL\ (1627::nat)\ (1000::nat) * (sqrt2 - (1::real))))))$

thm Nonlinear_lemma.vol2r_y:

$\forall (y::real)\ r::real.\ vol2r\ y\ r = real\_of\_nat\ (2::nat) * (pi * (r * (r * inverse\_class.inverse\ (real\_of\_nat\ (3::nat))))) - DECIMAL\ (5::nat)\ (10::nat) * (pi * (inverse\_class.inverse\ (real\_of\_nat\ (3::nat)) * y^2))$

thm Nonlinear_lemma.ineq_expand6:

$\forall (a1::real)\ (a2::real)\ (a3::real)\ (a4::real)\ (a5::real)\ (a6::real)\ (b1::real)\ (b2::real)\ (b3::real)\ (b4::real)\ (b5::real)\ (b6::real)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ P::bool.\ ineq\ [(a1,\ ?y1.0::real,\ b1),\ (a2,\ ?y2.0::real,\ b2),\ (a3,\ ?y3.0::real,\ b3),\ (a4,\ ?y4.0::real,\ b4),\ (a5,\ ?y5.0::real,\ b5),\ (a6,\ ?y6.0::real,\ b6)]\ P = (a1 \le ?y1.0 \wedge ?y1.0 \le b1 \longrightarrow a2 \le ?y2.0 \wedge ?y2.0 \le b2 \longrightarrow a3 \le ?y3.0 \wedge ?y3.0 \le b3 \longrightarrow a4 \le ?y4.0 \wedge ?y4.0 \le b4 \longrightarrow a5 \le ?y5.0 \wedge ?y5.0 \le b5 \longrightarrow a6 \le ?y6.0 \wedge ?y6.0 \le b6 \longrightarrow P)$

thm Nonlinear_lemma.ineq_expand9:

$\forall (a1::real)\ (a2::real)\ (a3::real)\ (a4::real)\ (a5::real)\ (a6::real)\ (a7::real)\ (a8::real)\ (a9::real)\ (b1::real)\ (b2::real)\ (b3::real)\ (b4::real)\ (b5::real)\ (b6::real)\ (b7::real)\ (b8::real)\ (b9::real)\ (x1::?'i::type)\ (x2::?'h::type)\ (x3::?'g::type)\ (x4::?'f::type)\ (x5::?'e::type)\ (x6::?'d::type)\ (x7::?'c::type)\ (x8::?'b::type)\ (x9::?'a::type)\ P::bool.\ ineq\ [(a1,\ ?y1.0::real,\ b1),\ (a2,\ ?y2.0::real,\ b2),\ (a3,\ ?y3.0::real,\ b3),\ (a4,\ ?y4.0::real,\ b4),\ (a5,\ ?y5.0::real,\ b5),\ (a6,\ ?y6.0::real,\ b6),\ (a7,\ ?y7.0::real,\ b7),\ (a8,\ ?y8.0::real,\ b8),\ (a9,\ ?y9.0::real,\ b9)]\ P = (a1 \le ?y1.0 \wedge ?y1.0 \le b1 \longrightarrow a2 \le ?y2.0 \wedge ?y2.0 \le b2 \longrightarrow a3 \le ?y3.0 \wedge ?y3.0 \le b3 \longrightarrow a4 \le ?y4.0 \wedge ?y4.0 \le b4 \longrightarrow a5 \le ?y5.0 \wedge ?y5.0 \le b5 \longrightarrow a6 \le ?y6.0 \wedge ?y6.0 \le b6 \longrightarrow a7 \le ?y7.0 \wedge ?y7.0 \le b7 \longrightarrow a8 \le ?y8.0 \wedge ?y8.0 \le b8 \longrightarrow a9 \le ?y9.0 \wedge ?y9.0 \le b9 \longrightarrow P)$

thm Nonlinear_lemma.pathL_bound:

$\forall$ (*y*::*real*) *a*::*real* × *real*. *fst* (*pathL a*) ≤ *y* ∧ *y* ≤ *snd* (*pathL a*) ⟶ *fst a* ≤ *y* ∧ *y* ≤ *snd a*

thm Nonlinear_lemma.pathR_bound:

$\forall$ (*y*::*real*) *a*::*real* × *real*. *fst* (*pathR a*) ≤ *y* ∧ *y* ≤ *snd* (*pathR a*) ⟶ *fst a* ≤ *y* ∧ *y* ≤ *snd a*

thm Nonlinear_lemma.delta_x_eq0:

delta_x (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) = (*0*::*real*) ∧ *delta_x* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) = (*0*::*real*)

thm Nonlinear_lemma.delta_x4_eq64:

delta_x4 (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) = − *real_of_nat* (*64*::*nat*) ∧ *delta_x4* (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*8*::*nat*)) (*real_of_nat* (*4*::*nat*)) = *real_of_nat* (*64*::*nat*)

thm Nonlinear_lemma.atn2_0y:

atn2 (*0*::*real*, *real_of_nat* (*64*::*nat*)) = *pi* / *real_of_nat* (*2*::*nat*) ∧ *atn2* (*0*::*real*, − *real_of_nat* (*64*::*nat*)) = − *pi* / *real_of_nat* (*2*::*nat*)

thm Nonlinear_lemma.gamma3f_135_n_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *gamma3f_135_n y1 y2 y3 y4 y5 y6* = *gamma3f_135_s_n y1 y2 y3 y4 y5 y6* + *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*y_of_x lmdih_x_n y1 y2 y3 y4 y5 y6* + (*y_of_x lmdih3_x_n y1 y2 y3 y4 y5 y6* + *y_of_x lmdih5_x_n y1 y2 y3 y4 y5 y6*))

thm Nonlinear_lemma.gamma3f_126_n_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *gamma3f_126_n y1 y2 y3 y4 y5 y6* = *gamma3f_126_s_n y1 y2 y3 y4 y5 y6* + *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ (*y_of_x lmdih_x_n y1 y2 y3 y4 y5 y6* + (*y_of_x lmdih2_x_n y1 y2 y3 y4 y5 y6* + *y_of_x lmdih6_x_n y1 y2 y3 y4 y5 y6*))

thm Nonlinear_lemma.gamma23f_n_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) (*w1*::*nat*) (*w2*::*nat*) *f*::*real* ⇒ *real*. *gamma23f_n y1 y2 y3 y4 y5 y6 w1 w2 sqrt2 f* = *gamma3f_126_n y1 y2 sqrt2 sqrt2 sqrt2 y6* / *real_of_nat w1* + (*gamma3f_135_n y1 sqrt2 y3 sqrt2 y5 sqrt2* / *real_of_nat w2* + *gamma3f_vLR_n y1 y2 y3 y4 y5 y6 f*)

thm Nonlinear_lemma.gamma23f_126_03_n_alt:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*y6*::*real*) *f*::*real* ⇒ *real*. *gamma23f_126_03_n y1 y2 y3 y4 y5 y6* (*?w1.0*::*nat*) *sqrt2 f* = *gamma3f_126_n* *y1 y2 sqrt2 sqrt2 sqrt2 y6 / real_of_nat ?w1.0 + gamma3f_vL_n y1 y2 y3 y4 y5 y6 f*

thm Nonlinear_lemma.gamma3f_vLR_n0_case:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y1* ⟶ *gamma3f_vLR_n y1 y2 y3 y4 y5 y6 lmfun* = *gamma3f_vLR_n0 y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.gamma3f_vLR_nlfun_case:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *y1* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ⟶ *gamma3f_vLR_n y1 y2 y3 y4 y5 y6 lmfun* = *gamma3f_vLR_nlfun y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.gamma3f_vL_n0_case:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y1* ⟶ *gamma3f_vL_n y1 y2 y3 y4 y5 y6 lmfun* = *gamma3f_vL_n0 y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.gamma3f_vL_nlfun_case:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *y1* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ⟶ *gamma3f_vL_n y1 y2 y3 y4 y5 y6 lmfun* = *gamma3f_vL_nlfun y1 y2 y3 y4 y5 y6*

thm Nonlinear_lemma.sqrtxx:

∀ *x*≥*0*::*real*. *sqrt* (*x* ∗ *x*) = *x*

thm Nonlinear_lemma.hm0:

∀ *y*≤*real_of_nat* (*2*::*nat*) ∗ *hminus*. *y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*

thm Nonlinear_lemma.h0_lt_gt:

((*?y*::*real*) ≤ *DECIMAL* (*201*::*nat*) (*100*::*nat*) ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*) ∧ (*DECIMAL* (*28*::*nat*) (*10*::*nat*) ≤ *?y* ⟶ *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *?y*) ∧ (*?y* ≤ *real_of_nat* (*2*::*nat*) ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*) ∧ (*sqrt8* ≤ *?y* ⟶ *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *?y*) ∧ (*real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *?y* ⟶ (*0*::*real*) ≤ *?y*) ∧ (*real_of_nat* (*2*::*nat*) ≤ *?y* ⟶ (*0*::*real*) ≤ *?y*) ∧ (*?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *hminus* ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*) ∧ (*real_of_nat* (*2*::*nat*) ∗ *hminus* ≤ *?y* ⟶ (*0*::*real*) ≤ *?y*)

thm Nonlinear_lemma.lmdih0:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y1* ⟶ *y_of_x lmdih_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6* = (*0*::*real*)

thm Nonlinear_lemma.lmdih3_0:

∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y3* ⟶ *y_of_x lmdih3_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6* = (*0*::*real*)

thm Nonlinear_lemma.lmdih5_0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y5* ⟶ *y_of_x lmdih5_x_div_sqrtdelta_posbranch y1 y2 y3 y4 y5 y6* = (*0*::*real*)

thm Nonlinear_lemma.lmdih1_0′:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y1* ⟶ *lmdih_x_div_sqrtdelta_posbranch* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih3_0′:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y3* ⟶ *lmdih3_x_div_sqrtdelta_posbranch* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih5_0′:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y5* ⟶ *lmdih5_x_div_sqrtdelta_posbranch* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih_n0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y1* ⟶ *lmdih_x_n* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih2_n0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y2* ⟶ *lmdih2_x_n* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih3_n0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y3* ⟶ *lmdih3_x_n* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih5_n0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y5* ⟶ *lmdih5_x_n* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih6_n0:

$\forall$ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y6* ⟶ *lmdih6_x_n* (*y1* ∗ *y1*) (*y2* ∗ *y2*) (*y3* ∗ *y3*) (*y4* ∗ *y4*) (*y5* ∗ *y5*) (*y6* ∗ *y6*) = (*0*::*real*)

thm Nonlinear_lemma.lmdih_ldih:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y1 \wedge y1 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow y\_of\_x\ lmdih\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6 = y\_of\_x\ ldih\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.lmdih3_ldih3:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y3 \wedge y3 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow y\_of\_x\ lmdih3\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6 = y\_of\_x\ ldih3\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.lmdih5_ldih5:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y5 \wedge y5 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow y\_of\_x\ lmdih5\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6 = y\_of\_x\ ldih5\_x\_div\_sqrtdelta\_posbranch\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.lmdih_ldih′:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y1 \wedge y1 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6) = ldih\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.lmdih3_ldih3′:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y3 \wedge y3 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih3\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6) = ldih3\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.lmdih5_ldih5′:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y5 \wedge y5 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih5\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6) = ldih5\_x\_div\_sqrtdelta\_posbranch\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.lmdih_ldih_n:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y1 \wedge y1 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih\_x\_n\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6) = ldih\_x\_n\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.lmdih2_ldih2_n:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y2 \wedge y2 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih2\_x\_n\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6) = ldih2\_x\_n\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)\ (y4 * y4)\ (y5 * y5)\ (y6 * y6)$

thm Nonlinear_lemma.lmdih3_ldih3_n:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ (0::real) \leq y3 \wedge y3 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmdih3\_x\_n\ (y1 * y1)\ (y2 * y2)\ (y3 * y3)$

(y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6) = ldih3_x_n (y1 ∗ y1) (y2 ∗ y2) (y3 ∗ y3) (y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6)

thm Nonlinear_lemma.lmdih5_ldih5_n:

∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. (0::real) ≤ y5 ∧ y5 ≤ real_of_nat (2::nat) ∗ h0 ⟶ lmdih5_x_n (y1 ∗ y1) (y2 ∗ y2) (y3 ∗ y3) (y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6) = ldih5_x_n (y1 ∗ y1) (y2 ∗ y2) (y3 ∗ y3) (y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6)

thm Nonlinear_lemma.lmdih6_ldih6_n:

∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. (0::real) ≤ y6 ∧ y6 ≤ real_of_nat (2::nat) ∗ h0 ⟶ lmdih6_x_n (y1 ∗ y1) (y2 ∗ y2) (y3 ∗ y3) (y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6) = ldih6_x_n (y1 ∗ y1) (y2 ∗ y2) (y3 ∗ y3) (y4 ∗ y4) (y5 ∗ y5) (y6 ∗ y6)

thm Nonlinear_lemma.vol3f_lmln:

∀(y1::real) (y2::real) (y3::?'c::type) (y4::?'b::type) (y5::?'a::type) y6::real. y1 ≤ real_of_nat (2::nat) ∗ h0 ∧ y2 ≤ real_of_nat (2::nat) ∗ h0 ∧ y6 ≤ real_of_nat (2::nat) ∗ h0 ⟶ vol3f y1 y2 y6 sqrt2 lmfun = vol3f y1 y2 y6 sqrt2 lfun

thm Nonlinear_lemma.vol3_vol_x:

∀(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. (0::real) ≤ x1 ∧ (0::real) ≤ x2 ∧ (0::real) ≤ x6 ⟶ vol3_x_sqrt x1 x2 x3 x4 x5 x6 = vol_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6

thm Nonlinear_lemma.vol3f_x_lfun_alt:

∀(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. (0::real) ≤ x1 ∧ (0::real) ≤ x2 ∧ (0::real) ≤ x6 ⟶ vol3f_x_lfun x1 x2 x3 x4 x5 x6 = real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ dih_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (real_of_nat (2::nat) ∗ dih2_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (real_of_nat (2::nat) ∗ dih6_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih3_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih4_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih5_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 − real_of_nat (3::nat) ∗ pi)))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (ldih_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (ldih2_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + ldih6_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6))

thm Nonlinear_lemma.vol3f_x_sqrt2_lmplus_alt:

∀(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. (real_of_nat (2::nat) ∗ h0)^2 ≤ x1 ∧ (0::real) ≤ x2 ∧ (0::real) ≤ x6 ⟶ vol3f_x_sqrt2_lmplus x1 x2 x3 x4 x5 x6 = real_of_nat (2::nat) ∗ (mm1 / pi) ∗ (real_of_nat (2::nat) ∗ dih_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (real_of_nat (2::nat) ∗ dih2_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (real_of_nat (2::nat) ∗ dih6_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih3_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih4_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + (dih5_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 − real_of_nat (3::nat) ∗ pi)))))) − real_of_nat (8::nat) ∗ (mm2 / pi) ∗ (ldih2_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6 + ldih6_x x1 x2 (real_of_nat (2::nat)) (real_of_nat (2::nat)) (real_of_nat (2::nat)) x6)

thm Nonlinear_lemma.vol3f_lm0:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. real_of_nat (2::nat) ∗ h0 ≤ y1 ∧ y2 ≤ real_of_nat (2::nat) ∗ h0 ∧ y6 ≤ real_of_nat (2::nat) ∗ h0 ⟶ vol3f y1 y2 y6 sqrt2 lmfun = vol3f_sqrt2_lmplus y1 y2 y3 y4 y5 y6

thm Nonlinear_lemma.gamma23f′:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) (w1::nat) (w2::nat) f::real ⇒ real. gamma23f y1 y2 y3 y4 y5 y6 w1 w2 sqrt2 f = gamma3f_126 y1 y2 y3 y4 y5 y6 f / real_of_nat w1 + (gamma3f_135 y1 y2 y3 y4 y5 y6 f / real_of_nat w2 + gamma3f_vLR y1 y2 y3 y4 y5 y6 f)

thm Nonlinear_lemma.gamma23f_126_03′:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) (w1::nat) f::real ⇒ real. gamma23f_126_03 y1 y2 y3 y4 y5 y6 w1 sqrt2 f = gamma3f_126 y1 y2 y3 y4 y5 y6 f / real_of_nat w1 + gamma3f_vL y1 y2 y3 y4 y5 y6 f

thm Nonlinear_lemma.gamma23f_v′:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) (y6::real) (w1::?′a::type) f::real ⇒ real. gamma23f_red_03 y1 y2 y3 y4 y5 y6 sqrt2 f = gamma3f_v y1 y2 y3 y4 y5 y6 f

thm Nonlinear_lemma.gamma3f_vLR0_case:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. real_of_nat (2::nat) ∗ h0 ≤ y1 ⟶ gamma3f_vLR y1 y2 y3 y4 y5 y6 lmfun = gamma3f_vLR0 y1 y2 y3 y4 y5 y6

thm Nonlinear_lemma.gamma3f_vLR_lfun_case:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. y1 ≤ real_of_nat (2::nat) ∗ h0 ⟶ gamma3f_vLR y1 y2 y3 y4 y5 y6 lmfun = gamma3f_vLR_lfun y1 y2 y3 y4 y5 y6

thm Nonlinear_lemma.gamma3f_vL0_case:

∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. real_of_nat (2::nat) ∗ h0 ≤ y1 ⟶ gamma3f_vL y1 y2 y3 y4 y5 y6 lmfun = gamma3f_vL0 y1 y2 y3 y4 y5 y6

thm Nonlinear_lemma.gamma3f_vL_lfun_case:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ y1 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow gamma3f\_vL\ y1\ y2\ y3\ y4\ y5\ y6\ lmfun = gamma3f\_vL\_lfun\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.gamma3f_v0_case:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ real\_of\_nat\ (2::nat) * h0 \leq y1 \longrightarrow gamma3f\_v\ y1\ y2\ y3\ y4\ y5\ y6\ lmfun = gamma3f\_v0\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.gamma3f_v_lfun_case:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ y1 \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow gamma3f\_v\ y1\ y2\ y3\ y4\ y5\ y6\ lmfun = gamma3f\_v\_lfun\ y1\ y2\ y3\ y4\ y5\ y6$

thm Nonlinear_lemma.gamma3f_126_expand:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ gamma3f\_126\ y1\ y2\ y3\ y4\ y5\ y6\ (?f::real \Rightarrow real) = vol3r\ y1\ y2\ y6\ sqrt2 - (real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (real\_of\_nat\ (2::nat) * dih\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (real\_of\_nat\ (2::nat) * dih2\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (real\_of\_nat\ (2::nat) * dih6\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (dih3\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (dih4\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (dih5\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 - real\_of\_nat\ (3::nat) * pi)))))) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * (?f\ (y1\ /\ real\_of\_nat\ (2::nat)) * dih\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + (?f\ (y2\ /\ real\_of\_nat\ (2::nat)) * dih2\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6 + ?f\ (y6\ /\ real\_of\_nat\ (2::nat)) * dih6\_y\ y1\ y2\ sqrt2\ sqrt2\ sqrt2\ y6)))$

thm Nonlinear_lemma.gamma3f_135_expand:

$\forall (y1::real)\ (y2::real)\ (y3::real)\ (y4::real)\ (y5::real)\ y6::real.\ gamma3f\_135\ y1\ y2\ y3\ y4\ y5\ y6\ (?f::real \Rightarrow real) = vol3r\ y1\ y3\ y5\ sqrt2 - (real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * (real\_of\_nat\ (2::nat) * dih\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (real\_of\_nat\ (2::nat) * dih3\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (real\_of\_nat\ (2::nat) * dih5\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (dih2\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (dih4\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (dih6\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 - real\_of\_nat\ (3::nat) * pi)))))) - real\_of\_nat\ (8::nat) * (mm2\ /\ pi) * (?f\ (y1\ /\ real\_of\_nat\ (2::nat)) * dih\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + (?f\ (y3\ /\ real\_of\_nat\ (2::nat)) * dih3\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2 + ?f\ (y5\ /\ real\_of\_nat\ (2::nat)) * dih5\_y\ y1\ sqrt2\ y3\ sqrt2\ y5\ sqrt2)))$

thm Nonlinear_lemma.vol3r_126_x:

$vol3r\ (sqrt\ (?x1.0::real))\ (sqrt\ (?x2.0::real))\ (sqrt\ (?x6.0::real))\ sqrt2 = vol3\_x\_sqrt\ ?x1.0\ ?x2.0\ (?x3.0::real)\ (?x4.0::real)\ (?x5.0::real)\ ?x6.0$

thm Nonlinear_lemma.ldih_x_135_s2:

$\forall (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ldih\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6 = lfun\ (sqrt\ x1\ /\ DECIMAL\ (20::nat)\ (10::nat)) * dih\_x\_135\_s2\ x1\ x2\ x3\ x4\ x5\ x6$

thm Nonlinear_lemma.num1_poly:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *num1 x1 x2 x3 x4 x5 x6* = *real_of_nat* (*64*::*nat*) $*$ (*x1* $*$ *x4*) $-$ *real_of_nat* (*32*::*nat*) $*$ (*x2* $*$ *x4*) $-$ *real_of_nat* (*32*::*nat*) $*$ (*x3* $*$ *x4*) $-$ *real_of_nat* (*4*::*nat*) $*$ (*x1* $*$ *x4*^{2}) $-$ *real_of_nat* (*32*::*nat*) $*$ (*x2* $*$ *x5*) + (*real_of_nat* (*32*::*nat*) $*$ (*x3* $*$ *x5*) + (*real_of_nat* (*4*::*nat*) $*$ (*x2* $*$ (*x4* $*$ *x5*)) + (*real_of_nat* (*32*::*nat*) $*$ (*x2* $*$ *x6*) $-$ *real_of_nat* (*32*::*nat*) $*$ (*x3* $*$ *x6*) + *real_of_nat* (*4*::*nat*) $*$ (*x3* $*$ (*x4* $*$ *x6*)))))

thm Nonlinear_lemma.num_combo1_num1:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *num_combo1 x1 x2 x3 x4 x5 x6* = (*num1 x1 x2 x3 x4 x5 x6*)^2 $-$ *DECIMAL* (*1*::*nat*) (*100*::*nat*) $*$ *num2 x1 x2 x3 x4 x5 x6*

thm Nonlinear_lemma.ineq6_of_ineq5:

$\forall$ (*a1*::*real*) (*a2*::*real*) (*a3*::*real*) (*a4*::*real*) (*a5*::*real*) (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) (*b1*::*real*) (*b2*::*real*) (*b3*::*real*) (*b4*::*real*) (*b5*::*real*) *P*::*real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *real* $\Rightarrow$ *bool*. ($\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ineq* [(*a1*, *x1*, *b1*), (*a2*, *x2*, *b2*), (*a3*, *x3*, *b3*), (*a4*, *x4*, *b4*), (*a5*, *x5*, *b5*), (*1*::*real*, *x6*, *1*::*real*)] (*P x1 x2 x3 x4 x5*)) $\longrightarrow$ *ineq* [(*a1*, *y1*, *b1*), (*a2*, *y2*, *b2*), (*a3*, *y3*, *b3*), (*a4*, *y4*, *b4*), (*a5*, *y5*, *b5*)] (*P y1 y2 y3 y4 y5*)

thm Nonlinear_lemma.ineq6_of_ineq1:

$\forall$ (*a1*::*real*) (*y1*::*real*) (*b1*::*real*) *P*::*real* $\Rightarrow$ *bool*. ($\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ineq* [(*a1*, *x1*, *b1*), (*1*::*real*, *x2*, *1*::*real*), (*1*::*real*, *x3*, *1*::*real*), (*1*::*real*, *x4*, *1*::*real*), (*1*::*real*, *x5*, *1*::*real*), (*1*::*real*, *x6*, *1*::*real*)] (*P x1*)) $\longrightarrow$ *ineq* [(*a1*, *y1*, *b1*)] (*P y1*)

thm Nonlinear_lemma.taum_3flat_x_alt:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x23*::*real*) (*x13*::*real*) *x12*::*real*. *taum_3flat_x x1 x2 x3 x23 x13 x12* = *taum_x x1 x2 x3* (*edge_flat2_x x23 x2 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) + (*flat_term_x x12* + (*flat_term_x x23* + *flat_term_x x13*))

thm Nonlinear_lemma.taum_2flat_x_alt:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x13*::*real*) *x12*::*real*. *taum_2flat_x x1 x2 x3 x4 x13 x12* = *taum_x x1 x2 x3 x4* (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) + (*flat_term_x x12* + *flat_term_x x13*)

thm Nonlinear_lemma.taum_1flat_x_alt:

$\forall$ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x12*::*real*. *taum_1flat_x x1 x2 x3 x4 x5 x12* = *taum_x x1 x2 x3 x4 x5* (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) + *flat_term_x x12*

thm Nonlinear_lemma.euler_3flat_x_alt:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x23*::*real*) (*x13*::*real*) *x12*::*real*. *euler_3flat_x x1 x2 x3 x23 x13 x12* = *eulerA_x x1 x2 x3* (*edge_flat2_x x23 x2 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm Nonlinear_lemma.euler_2flat_x_alt:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x13*::*real*) *x12*::*real*. *euler_2flat_x x1 x2 x3 x4 x13 x12* = *eulerA_x x1 x2 x3 x4* (*edge_flat2_x x13 x1 x3* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*))) (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm Nonlinear_lemma.euler_1flat_x_alt:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x12*::*real*. *euler_1flat_x x1 x2 x3 x4 x5 x12* = *eulerA_x x1 x2 x3 x4 x5* (*edge_flat2_x x12 x1 x2* (*0*::*real*) (*real_of_nat* (*4*::*nat*)) (*real_of_nat* (*4*::*nat*)))

thm Nonlinear_lemma.dih_x_alt:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *dih_x x1 x2 x3 x4 x5 x6* = *pi* / *real_of_nat* (*2*::*nat*) + *atn2* (*sqrt* (*real_of_nat* (*4*::*nat*) ∗ (*x1* ∗ *delta_x x1 x2 x3 x4 x5 x6*)), − *delta_x4 x1 x2 x3 x4 x5 x6*)

thm Functional_equation.uni:

∀ (*f*::*?'h*::*type* ⇒ *?'g*::*type*) (*x*::*?'f*::*type* ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'h*::*type*) (*x1*::*?'f*::*type*) (*x2*::*?'e*::*type*) (*x3*::*?'d*::*type*) (*x4*::*?'c*::*type*) (*x5*::*?'b*::*type*) *x6*::*?'a*::*type*. *uni* (*f*, *x*) *x1 x2 x3 x4 x5 x6* = *f* (*x x1 x2 x3 x4 x5 x6*)

thm Functional_equation.constant6:

∀ (*x1*::*?'g*::*type*) (*x2*::*?'f*::*type*) (*x3*::*?'e*::*type*) (*x4*::*?'d*::*type*) (*x5*::*?'c*::*type*) (*x6*::*?'b*::*type*) *c*::*?'a*::*type*. *constant6 c x1 x2 x3 x4 x5 x6* = *c*

thm Functional_equation.promote3_to_6:

∀ (*x4*::*?'g*::*type*) (*x5*::*?'f*::*type*) (*x6*::*?'e*::*type*) (*f*::*?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) (*x1*::*?'d*::*type*) (*x2*::*?'c*::*type*) *x3*::*?'b*::*type*. *promote3_to_6 f x1 x2 x3 x4 x5 x6* = *f x1 x2 x3*

thm Functional_equation.promote1_to_6:

∀ (*x2*::*?'g*::*type*) (*x3*::*?'f*::*type*) (*x4*::*?'e*::*type*) (*x5*::*?'d*::*type*) (*x6*::*?'c*::*type*) (*f*::*?'b*::*type* ⇒ *?'a*::*type*) *x1*::*?'b*::*type*. *promote1_to_6 f x1 x2 x3 x4 x5 x6* = *f x1*

thm Functional_equation.functional_proj_x1:

proj_x1 = *promote1_to_6 id*

thm Functional_equation.functional_proj_x2:

$proj\_x2 = rotate2\ proj\_x1$

thm Functional_equation.functional_proj_x3:

$proj\_x3 = rotate3\ proj\_x1$

thm Functional_equation.functional_proj_x4:

$proj\_x4 = rotate4\ proj\_x1$

thm Functional_equation.functional_proj_x5:

$proj\_x5 = rotate5\ proj\_x1$

thm Functional_equation.functional_proj_x6:

$proj\_x6 = rotate6\ proj\_x1$

thm Functional_equation.two6:

$two6 = constant6\ (real\_of\_nat\ (2::nat))$

thm Functional_equation.zero6:

$zero6 = constant6\ (0::real)$

thm Functional_equation.dummy6:

$dummy6 = constant6\ (0::real)$

thm Functional_equation.four6:

$four6 = constant6\ (real\_of\_nat\ (4::nat))$

thm Functional_equation.add6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ add6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 + g\ x1\ x2\ x3\ x4\ x5\ x6$

thm Functional_equation.scalar6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ r::real.\ scalar6\ f\ r\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 * r$

thm Functional_equation.mul6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (x1::?'f::type)\ (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ x6::?'a::type.\ mul6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 * g\ x1\ x2\ x3\ x4\ x5\ x6$

thm Functional_equation.div6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $div6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6\ /\ g\ x1\ x2\ x3\ x4\ x5\ x6$

thm Functional_equation.sub6:

$\forall (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)$ $(x1::?'f::type)$ $(x2::?'e::type)$ $(x3::?'d::type)$ $(x4::?'c::type)$ $(x5::?'b::type)$ $x6::?'a::type.$ $sub6\ f\ g\ x1\ x2\ x3\ x4\ x5\ x6 = f\ x1\ x2\ x3\ x4\ x5\ x6 - g\ x1\ x2\ x3\ x4\ x5\ x6$

thm Functional_equation.proj_y1:

$\forall (x2::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x1::real.$ $proj\_y1\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x1$

thm Functional_equation.proj_y2:

$\forall (x1::?'e::type)\ (x3::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x2::real.$ $proj\_y2\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x2$

thm Functional_equation.proj_y3:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x4::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x3::real.$ $proj\_y3\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x3$

thm Functional_equation.proj_y4:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x5::?'b::type)\ (x6::?'a::type)\ x4::real.$ $proj\_y4\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x4$

thm Functional_equation.proj_y5:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::?'b::type)\ (x6::?'a::type)\ x5::real.$ $proj\_y5\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x5$

thm Functional_equation.proj_y6:

$\forall (x1::?'e::type)\ (x2::?'d::type)\ (x3::?'c::type)\ (x4::?'b::type)\ (x5::?'a::type)\ x6::real.$ $proj\_y6\ x1\ x2\ x3\ x4\ x5\ x6 = sqrt\ x6$

thm Functional_equation.domain6:

$\forall (h::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow bool)$ $(f::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type)$ $g::?'g::type \Rightarrow ?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type.$ $domain6\ h\ f\ g = (\forall (x1::?'g::type)\ (x2::?'f::type)$ $(x3::?'e::type)\ (x4::?'d::type)\ (x5::?'c::type)\ x6::?'b::type.\ h\ x1\ x2\ x3\ x4\ x5\ x6 \longrightarrow f\ x1\ x2\ x3\ x4\ x5\ x6 = g\ x1\ x2\ x3\ x4\ x5\ x6)$

thm Functional_equation.x1cube:

x1cube = mul6 proj_x1 (mul6 proj_x1 proj_x1)

thm Functional_equation.nonf_truncate_sol_x:

$\forall$ (*c::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_sol_x c x1 x2 x3 x4 x5 x6 = truncate_dih_x c x1 x2 x3 x4 x5 x6 +* (*truncate_dih_x c x2 x3 x1 x5 x6 x4 +* (*truncate_dih_x c x3 x1 x2 x6 x4 x5 − pi*))

thm Functional_equation.nonf_truncate_vol_x:

$\forall$ (*c::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_vol_x c x1 x2 x3 x4 x5 x6 = truncate_sqrt c* (*delta_x x1 x2 x3 x4 x5 x6*) / *real_of_nat* (*12::nat*)

thm Functional_equation.nonf_truncate_vol3r_456:

$\forall$ (*c::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_vol3r_456 c x1 x2 x3 x4 x5 x6 = truncate_vol_x c* (*real_of_nat* (*2::nat*)) (*real_of_nat* (*2::nat*)) (*real_of_nat* (*2::nat*)) *x4 x5 x6*

thm Functional_equation.nonf_truncate_vol3f:

$\forall$ (*c::real*) (*m4::real*) (*m5::real*) (*m6::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_vol3f c m4 m5 m6 x1 x2 x3 x4 x5 x6 =* (*truncate_sol_x c x5* (*real_of_nat* (*2::nat*)) *x4* (*real_of_nat* (*2::nat*)) *x6* (*real_of_nat* (*2::nat*)) + (*truncate_sol_x c x6* (*real_of_nat* (*2::nat*)) *x5* (*real_of_nat* (*2::nat*)) *x4* (*real_of_nat* (*2::nat*)) + *truncate_sol_x c x4* (*real_of_nat* (*2::nat*)) *x6* (*real_of_nat* (*2::nat*)) *x5* (*real_of_nat* (*2::nat*)))) ∗ (*real_of_nat* (*2::nat*) ∗ (*mm1* / *pi*)) − (*lfun* (*sqrt x4* ∗ *DECIMAL* (*5::nat*) (*10::nat*)) ∗ *m4* ∗ *truncate_dih_x c x4* (*real_of_nat* (*2::nat*)) *x6* (*real_of_nat* (*2::nat*)) *x5* (*real_of_nat* (*2::nat*)) + (*lfun* (*sqrt x5* ∗ *DECIMAL* (*5::nat*) (*10::nat*)) ∗ *m5* ∗ *truncate_dih_x c x5* (*real_of_nat* (*2::nat*)) *x4* (*real_of_nat* (*2::nat*)) *x6* (*real_of_nat* (*2::nat*)) + *lfun* (*sqrt x6* ∗ *DECIMAL* (*5::nat*) (*10::nat*)) ∗ *m6* ∗ *truncate_dih_x c x6* (*real_of_nat* (*2::nat*)) *x5* (*real_of_nat* (*2::nat*)) *x4* (*real_of_nat* (*2::nat*)))) ∗ (*real_of_nat* (*8::nat*) ∗ (*mm2* / *pi*))

thm Functional_equation.nonf_truncate_gamma3f_x:

$\forall$ (*d::real*) (*m4::real*) (*m5::real*) (*m6::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_gamma3f_x d m4 m5 m6 x1 x2 x3 x4 x5 x6 = truncate_vol3r_456 d x1 x2 x3 x4 x5 x6 − truncate_vol3f d m4 m5 m6 x1 x2 x3 x4 x5 x6*

thm Functional_equation.nonf_truncate_gamma23_x:

$\forall$ (*iw1::real*) (*iw2::real*) (*m1::real*) (*m2::real*) (*m3::real*) (*m5::real*) (*m6::real*) (*x1::real*) (*x2::real*) (*x3::real*) (*x4::real*) (*x5::real*) *x6::real. truncate_gamma23_x iw1 iw2 m1 m2 m3 m5 m6 x1 x2 x3 x4 x5 x6 = truncate_gamma3f_x* (*DECIMAL* (*14::nat*) (*100::nat*)) *m1 m2 m6* (*0::real*) (*0::real*) (*0::real*) *x1 x2 x6* ∗ *iw1* + (*truncate_gamma3f_x* (*DECIMAL* (*14::nat*) (*100::nat*)) *m1 m3 m5* (*0::real*) (*0::real*) (*0::real*) *x1 x3 x5* ∗ *iw2* + (*dih_x x1 x2 x3 x4 x5 x6* − (*truncate_dih_x* (*DECIMAL* (*14::nat*) (*100::nat*)) *x1 x2* (*real_of_nat* (*2::nat*)) (*real_of_nat*

(*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *x6* + *truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *x1* (*real_of_nat* (*2*::*nat*)) *x3* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)))) ∗ *truncate_gamma2_x m1 x1*)

thm Functional_equation.nonf_truncate_gamma23_x_B:

∀ (*m1*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *truncate_gamma23_x_B m1 x1 x2 x3 x4 x5 x6* = (*dih_x x1 x2 x3 x4 x5 x6* − *real_of_nat* (*2*::*nat*) ∗ *DECIMAL* (*8*::*nat*) (*100*::*nat*)) ∗ *truncate_gamma2_x m1 x1*

thm Functional_equation.nonf_truncate_gamma23_x_C:

∀ (*iw1*::*real*) (*m1*::*real*) (*m2*::*real*) (*m6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *truncate_gamma23_x_C* (*?d*::*real*) *iw1 m1 m2 m6 x1 x2 x3 x4 x5 x6* = *truncate_gamma3f_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *m1 m2 m6* (*0*::*real*) (*0*::*real*) (*0*::*real*) *x1 x2 x6* ∗ *iw1* + (*dih_x x1 x2 x3 x4 x5 x6* − (*truncate_dih_x* (*DECIMAL* (*14*::*nat*) (*100*::*nat*)) *x1 x2* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*)) *x6* + *?d*)) ∗ *truncate_gamma2_x m1 x1*

thm Functional_equation.functional_proj_y1:

proj_y1 = *promote1_to_6 sqrt*

thm Functional_equation.functional_proj_y2:

proj_y2 = *rotate2 proj_y1*

thm Functional_equation.functional_proj_y3:

proj_y3 = *rotate3 proj_y1*

thm Functional_equation.functional_proj_y4:

proj_y4 = *rotate4 proj_y1*

thm Functional_equation.functional_proj_y5:

proj_y5 = *rotate5 proj_y1*

thm Functional_equation.functional_proj_y6:

proj_y6 = *rotate6 proj_y1*

thm Functional_equation.functional_norm2hh_x:

norm2hh_x = *add6* (*uni* (*pow2*, *sub6 proj_y1* (*constant6* (*hminus* + *hplus*)))) (*add6* (*uni* (*pow2*, *sub6 proj_y2 two6*)) (*add6* (*uni* (*pow2*, *sub6 proj_y3 two6*)) (*add6* (*uni* (*pow2*, *sub6 proj_y4 two6*)) (*add6* (*uni* (*pow2*, *sub6 proj_y5 two6*)) (*uni* (*pow2*, *sub6 proj_y6 two6*))))))

thm Functional_equation.functional_eta2_126:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x2* ∧ (*0*::*real*) ≤ *x6*) *eta2_126* (*uni* (*pow2*, *rotate126* (*promote3_to_6 eta_x*)))

thm Functional_equation.functional_rotate2:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ *rotate2 f = compose6 f proj_x2 proj_x3 proj_x1 proj_x5 proj_x6 proj_x4*

thm Functional_equation.functional_rotate3:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ *rotate3 f = compose6 f proj_x3 proj_x1 proj_x2 proj_x6 proj_x4 proj_x5*

thm Functional_equation.functional_rotate4:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ *rotate4 f = compose6 f proj_x4 proj_x2 proj_x6 proj_x1 proj_x5 proj_x3*

thm Functional_equation.functional_rotate5:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ *rotate5 f = compose6 f proj_x5 proj_x3 proj_x4 proj_x2 proj_x6 proj_x1*

thm Functional_equation.functional_rotate6:

$\forall f::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.$ *rotate6 f = compose6 f proj_x6 proj_x1 proj_x5 proj_x3 proj_x4 proj_x2*

thm Functional_equation.functional_x1_delta_x:

x1_delta_x = mul6 proj_x1 delta_x

thm Functional_equation.functional_delta4_squared_x:

delta4_squared_x = uni (pow2, delta_x4)

thm Functional_equation.functional_vol_x:

vol_x = scalar6 (uni (sqrt, delta_x)) ((1::real) / real_of_nat (12::nat))

thm Functional_equation.functional_dih2_x:

dih2_x = rotate2 dih_x

thm Functional_equation.functional_dih3_x:

dih3_x = rotate3 dih_x

thm Functional_equation.functional_dih4_x:

domain6 $(\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real) \leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x3 \wedge (0::real) \leq x4 \wedge (0::real) \leq x5 \wedge (0::real) \leq x6)$ *dih4_x (rotate4 dih_x)*

thm Functional_equation.functional_dih5_x:

domain6 $(\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real) \leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x3 \wedge (0::real) \leq x4 \wedge (0::real) \leq x5 \wedge (0::real) \leq x6)$ *dih5_x (rotate5 dih_x)*

thm Functional_equation.functional_dih6_x:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real) \leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x3 \wedge (0::real) \leq x4 \wedge (0::real) \leq x5 \wedge (0::real) \leq x6)\ dih6\_x\ (rotate6\ dih\_x)$

thm Functional_equation.functional_lfun_y1:

$lfun\_y1 = scalar6\ (sub6\ (scalar6\ unit6\ h0)\ proj\_x1)\ rh0$

thm Functional_equation.functional_ldih_x:

$ldih\_x = mul6\ (scalar6\ (sub6\ (scalar6\ unit6\ h0)\ (scalar6\ proj\_y1\ (DECIMAL\ (5::nat)\ (10::nat))))\ rh0)\ dih\_x$

thm Functional_equation.functional_ldih2_x:

$ldih2\_x = rotate2\ ldih\_x$

thm Functional_equation.functional_ldih3_x:

$ldih3\_x = rotate3\ ldih\_x$

thm Functional_equation.functional_ldih6_x:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real) \leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x3 \wedge (0::real) \leq x4 \wedge (0::real) \leq x5 \wedge (0::real) \leq x6)\ ldih6\_x\ (rotate6\ ldih\_x)$

thm Functional_equation.functional_eulerA_x:

$eulerA\_x = add6\ (mul6\ proj\_y1\ (mul6\ proj\_y2\ proj\_y3))\ (add6\ (scalar6\ (mul6\ proj\_y1\ (add6\ proj\_x2\ (sub6\ proj\_x3\ proj\_x4)))\ (DECIMAL\ (5::nat)\ (10::nat)))\ (add6\ (scalar6\ (mul6\ proj\_y2\ (add6\ proj\_x1\ (sub6\ proj\_x3\ proj\_x5)))\ (DECIMAL\ (5::nat)\ (10::nat)))\ (scalar6\ (mul6\ proj\_y3\ (add6\ proj\_x1\ (sub6\ proj\_x2\ proj\_x6)))\ (DECIMAL\ (5::nat)\ (10::nat)))))$

thm Functional_equation.functional_gchi1_x:

$gchi1\_x = mul6\ (uni\ (gchi,\ proj\_y1))\ dih\_x$

thm Functional_equation.functional_gchi2_x:

$gchi2\_x = mul6\ (uni\ (gchi,\ proj\_y2))\ dih2\_x$

thm Functional_equation.functional_gchi3_x:

$gchi3\_x = mul6\ (uni\ (gchi,\ proj\_y3))\ dih3\_x$

thm Functional_equation.functional_gchi4_x:

$gchi4\_x = mul6\ (uni\ (gchi,\ proj\_y4))\ dih4\_x$

thm Functional_equation.functional_gchi5_x:

$gchi5\_x = mul6\ (uni\ (gchi,\ proj\_y5))\ dih5\_x$

thm Functional_equation.functional_gchi6_x:

$gchi6\_x = mul6\ (uni\ (gchi,\ proj\_y6))\ dih6\_x$

thm Functional_equation.functional_eta2_135:

eta2_135 = rotate3 eta2_126

thm Functional_equation.functional_eta2_456:

eta2_456 = rotate4 eta2_135

thm Functional_equation.functional_vol3_x_sqrt:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) $\leq$ *x1* $\wedge$ (*0*::*real*) $\leq$ *x2* $\wedge$ (*0*::*real*) $\leq$ *x6*) *vol3_x_sqrt* (*mk_126 vol_x*)

thm Functional_equation.functional_vol3f_x_lfun:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) $\leq$ *x1* $\wedge$ (*0*::*real*) $\leq$ *x2* $\wedge$ (*0*::*real*) $\leq$ *x6*) *vol3f_x_lfun* (*sub6* (*mul6* (*constant6* (*real_of_nat* (*2*::*nat*) * (*mm1* / *pi*))) (*add6* (*mul6 two6* (*mk_126 dih_x*)) (*add6* (*mul6 two6* (*mk_126 dih2_x*)) (*add6* (*mul6 two6* (*mk_126 dih6_x*)) (*add6* (*mk_126 dih3_x*) (*add6* (*mk_126 dih4_x*) (*sub6* (*mk_126 dih5_x*) (*constant6* (*real_of_nat* (*3*::*nat*) * *pi*))))))))) (*mul6* (*constant6* (*real_of_nat* (*8*::*nat*) * (*mm2* / *pi*))) (*add6* (*mk_126 ldih_x*) (*add6* (*mk_126 ldih2_x*) (*mk_126 ldih6_x*)))))

thm Functional_equation.functional_vol3f_x_sqrt2_lmplus:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*real_of_nat* (*2*::*nat*) * *h0*)^2 $\leq$ *x1* $\wedge$ (*0*::*real*) $\leq$ *x2* $\wedge$ (*0*::*real*) $\leq$ *x6*) *vol3f_x_sqrt2_lmplus* (*sub6* (*mul6* (*constant6* (*real_of_nat* (*2*::*nat*) * (*mm1* / *pi*))) (*add6* (*mul6 two6* (*mk_126 dih_x*)) (*add6* (*mul6 two6* (*mk_126 dih2_x*)) (*add6* (*mul6 two6* (*mk_126 dih6_x*)) (*add6* (*mk_126 dih3_x*) (*add6* (*mk_126 dih4_x*) (*sub6* (*mk_126 dih5_x*) (*constant6* (*real_of_nat* (*3*::*nat*) * *pi*))))))))) (*mul6* (*constant6* (*real_of_nat* (*8*::*nat*) * (*mm2* / *pi*))) (*add6* (*mk_126 ldih2_x*) (*mk_126 ldih6_x*))))

thm Functional_equation.functional_asn797k:

asn797k = *mul6 proj_x1* (*uni* (*asn*, *mul6* (*constant6 cos797*) (*uni* (*sin*, *div6* (*constant6 pi*) *proj_x1*))))

thm Functional_equation.functional_asnFnhk:

asnFnhk = *mul6 proj_x2* (*uni* (*asn*, *mul6* (*add6* (*mul6 proj_x1* (*constant6* (*sqrt3* / *DECIMAL* (*40*::*nat*) (*10*::*nat*)))) (*mul6* (*uni* (*sqrt*, *sub6 unit6* (*uni* (*pow2*, *mul6 proj_x1* (*constant6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*))))))) (*constant6* (*DECIMAL* (*5*::*nat*) (*10*::*nat*))))) (*uni* (*sin*, *div6* (*constant6 pi*) *proj_x2*))))

thm Functional_equation.functional_acs_sqrt_x1_d4:

acs_sqrt_x1_d4 = *uni* (*acs*, *scalar6 proj_y1* (*DECIMAL* (*25*::*nat*) (*100*::*nat*)))

thm Functional_equation.functional_acs_sqrt_x2_d4:

acs_sqrt_x2_d4 = *uni* (*acs*, *scalar6 proj_y2* (*DECIMAL* (*25*::*nat*) (*100*::*nat*)))

thm Functional_equation.functional_arclength_x_123:

$LET\ (\lambda al\_num::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ LET\_END$ $(LET\ (\lambda al\_den::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real.\ LET\_END$ $(LET\ (\lambda domain::real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow real \Rightarrow bool.\ LET\_END$ $(domain6\ domain\ arclength\_x\_123\ (uni\ (acs,\ div6\ al\_num\ al\_den))))\ (\lambda(x1::real)$ $(x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ ((0::real) < x1 \wedge (0::real) <$ $x2 \wedge (0::real) \leq x3) \wedge sqrt\ x3 \leq sqrt\ x1 + sqrt\ x2 \wedge sqrt\ x1 \leq sqrt\ x2 + sqrt$ $x3 \wedge sqrt\ x2 \leq sqrt\ x3 + sqrt\ x1)))\ (uni\ (sqrt,\ scalar6\ (mul6\ proj\_x1\ proj\_x2)$ $(real\_of\_nat\ (4::nat))))))\ (add6\ proj\_x1\ (sub6\ proj\_x2\ proj\_x3))$

thm Functional_equation.functional_arclength_234:

$arclength\_x\_234 = rotate234\ arclength\_x\_123$

thm Functional_equation.functional_arclength_126:

$arclength\_x\_126 = rotate126\ arclength\_x\_123$

thm Functional_equation.functional_sol_euler_x_divsqrtdelta:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real)$ $\leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x3)\ sol\_euler\_x\_div\_sqrtdelta\ (div6\ (uni$ $(matan,\ div6\ delta\_x\ (scalar6\ (mul6\ eulerA\_x\ eulerA\_x)\ (real\_of\_nat\ (4::nat)))))$ $eulerA\_x)$

thm Functional_equation.functional_dih_x_div_sqrtdelta_posbranch:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real)$ $\leq x1)\ dih\_x\_div\_sqrtdelta\_posbranch\ (mul6\ (div6\ (scalar6\ proj\_y1\ (real\_of\_nat$ $(2::nat)))\ delta\_x4)\ (uni\ (matan,\ div6\ (scalar6\ (mul6\ proj\_x1\ delta\_x)\ (real\_of\_nat$ $(4::nat)))\ (uni\ (pow2,\ delta\_x4)))))$

thm Functional_equation.functional_dih_x_126_s2:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real)$ $\leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x6)\ dih\_x\_126\_s2\ (mk\_126\ dih\_x)$

thm Functional_equation.functional_dih2_x_126_s2:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real)$ $\leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x6)\ dih2\_x\_126\_s2\ (mk\_126\ dih2\_x)$

thm Functional_equation.functional_dih3_x_126_s2:

$domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real)$ $\leq x1 \wedge (0::real) \leq x2 \wedge (0::real) \leq x6)\ dih3\_x\_126\_s2\ (mk\_126\ dih3\_x)$

thm Functional_equation.functional_dih4_x_126_s2:

$dih4\_x\_126\_s2 = mk\_126\ dih4\_x$

thm Functional_equation.functional_dih5_x_126_s2:

$dih5\_x\_126\_s2 = mk\_126\ dih5\_x$

thm Functional_equation.functional_dih6_x_126_s2:

dih6_x_126_s2 = mk_126 dih6_x

thm Functional_equation.functional_dih_x_135_s2:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5*) *dih_x_135_s2* (*mk_135 dih_x*)

thm Functional_equation.functional_dih2_x_135_s2:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5*) *dih2_x_135_s2* (*mk_135 dih2_x*)

thm Functional_equation.functional_dih3_x_135_s2:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5*) *dih3_x_135_s2* (*mk_135 dih3_x*)

thm Functional_equation.functional_dih4_x_135_s2:

dih4_x_135_s2 = mk_135 dih4_x

thm Functional_equation.functional_dih5_x_135_s2:

dih5_x_135_s2 = mk_135 dih5_x

thm Functional_equation.functional_dih6_x_135_s2:

dih6_x_135_s2 = mk_135 dih6_x

thm Functional_equation.functional_ldih_x_126_s2:

ldih_x_126_s2 = mul6 (*uni* (*lfun*, *div6 proj_y1 two6*)) *dih_x_126_s2*

thm Functional_equation.functional_ldih2_x_126_s2:

ldih2_x_126_s2 = mul6 (*uni* (*lfun*, *div6 proj_y2 two6*)) *dih2_x_126_s2*

thm Functional_equation.functional_ldih6_x_126_s2:

ldih6_x_126_s2 = mul6 (*uni* (*lfun*, *div6 proj_y6 two6*)) *dih6_x_126_s2*

thm Functional_equation.functional_ldih_x_135_s2:

ldih_x_135_s2 = mul6 (*uni* (*lfun*, *div6 proj_y1 two6*)) *dih_x_135_s2*

thm Functional_equation.functional_ldih3_x_135_s2:

ldih3_x_135_s2 = mul6 (*uni* (*lfun*, *div6 proj_y3 two6*)) *dih3_x_135_s2*

thm Functional_equation.functional_ldih5_x_135_s2:

ldih5_x_135_s2 = mul6 (*uni* (*lfun*, *div6 proj_y5 two6*)) *dih5_x_135_s2*

thm Functional_equation.functional_edge_flat2_x:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x2* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5* ∧ (*0*::*real*) ≤ *x6*) *edge_flat2_x* (*uni* (*pow2*, *uni* (*sqrt*, *compose6* (*promote3_to_6 quadratic_root_plus_curry*) *proj_x1* (*add6* (*mul6 proj_x1 proj_x1*) (*sub6* (*mul6* (*sub6 proj_x3 proj_x5*) (*sub6 proj_x2 proj_x6*)) (*mul6 proj_x1* (*add6 proj_x2* (*add6 proj_x3* (*add6 proj_x5 proj_x6*)))))) (*add6* (*mul6 proj_x1* (*mul6 proj_x3 proj_x5*)) (*sub6* (*sub6* (*mul6 proj_x1* (*mul6 proj_x2 proj_x6*)) (*mul6 proj_x3* (*mul6* (*add6 proj_x1* (*add6* (*sub6 proj_x2 proj_x3*) (*sub6 proj_x5 proj_x6*))) *proj_x6*))) (*mul6 proj_x2* (*mul6 proj_x5* (*add6* (*sub6 proj_x1 proj_x2*) (*add6* (*sub6 proj_x3 proj_x5*) *proj_x6*)))))) *zero6 zero6 zero6*)))

thm Functional_equation.functional_euler_3flat_x:

LET (λ*x4r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x5r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*euler_3flat_x* = *compose6 eulerA_x proj_x1 proj_x2 proj_x3 x4r x5r x6r*)) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x5 proj_x1 proj_x3 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x4 proj_x2 proj_x3 zero6 four6 four6*)

thm Functional_equation.functional_euler_2flat_x:

LET (λ*x5r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*euler_2flat_x* = *compose6 eulerA_x proj_x1 proj_x2 proj_x3 proj_x4 x5r x6r*)) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x5 proj_x1 proj_x3 zero6 four6 four6*)

thm Functional_equation.functional_euler_1flat_x:

LET (λ*x6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*euler_1flat_x* = *compose6 eulerA_x proj_x1 proj_x2 proj_x3 proj_x4 proj_x5 x6r*)) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*)

thm Functional_equation.functional_taum_3flat_x:

LET (λ*x4r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x5r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*taum_3flat_x* = *add6* (*compose6 taum_x proj_x1 proj_x2 proj_x3 x4r x5r x6r*) (*add6* (*uni* (*flat_term_x*, *proj_x4*)) (*add6* (*uni* (*flat_term_x*, *proj_x5*)) (*uni* (*flat_term_x*, *proj_x6*)))))) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x5 proj_x1 proj_x3 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x4 proj_x2 proj_x3 zero6 four6 four6*)

thm Functional_equation.functional_taum_2flat_x:

LET (λ*x5r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*LET* (λ*x6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*taum_2flat_x* = *add6* (*compose6 taum_x proj_x1 proj_x2 proj_x3 proj_x4 x5r x6r*) (*add6* (*uni* (*flat_term_x*, *proj_x5*)) (*uni* (*flat_term_x*, *proj_x6*))))) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*))) (*compose6 edge_flat2_x proj_x5 proj_x1 proj_x3 zero6 four6 four6*)

thm Functional_equation.functional_taum_1flat_x:

LET (*λx6r*::*real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real* ⇒ *real*. *LET_END* (*taum_1flat_x* = *add6* (*compose6 taum_x proj_x1 proj_x2 proj_x3 proj_x4 proj_x5 x6r*) (*uni* (*flat_term_x*, *proj_x6*)))) (*compose6 edge_flat2_x proj_x6 proj_x1 proj_x2 zero6 four6 four6*)

thm Functional_equation.functional_delta_x_126_s2:

domain6 (*λ*(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x2* ∧ (*0*::*real*) ≤ *x6*) *delta_x_126_s2* (*mk_126 delta_x*)

thm Functional_equation.functional_delta_x_135_s2:

domain6 (*λ*(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5*) *delta_x_135_s2* (*mk_135 delta_x*)

thm Functional_equation.functional_delta_pent_x:

delta_pent_x = *compose6 delta_x proj_x1 proj_x2 proj_x6 four6 four6* (*constant6* (*DECIMAL* (*104976*::*nat*) (*10000*::*nat*)))

thm Functional_equation.functional_vol3_x_135_s2:

domain6 (*λ*(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x5*) *vol3_x_135_s2* (*mk_135 vol_x*)

thm Functional_equation.functional_ldih_x_div_sqrtdelta_posbranch:

ldih_x_div_sqrtdelta_posbranch = *mul6* (*scalar6* (*sub6* (*constant6 h0*) (*scalar6 proj_y1* (*DECIMAL* (*5*::*nat*) (*10*::*nat*)))) *rh0*) *dih_x_div_sqrtdelta_posbranch*

thm Functional_equation.functional_ldih_x_n:

ldih_x_n = *mul6* (*uni* (*sqn*, *delta_x*)) *ldih_x_div_sqrtdelta_posbranch*

thm Functional_equation.functional_ldih_x_126_n:

ldih_x_126_n = *mk_126 ldih_x_n*

thm Functional_equation.functional_ldih2_x_126_n:

ldih2_x_126_n = *mk_126* (*rotate2 ldih_x_n*)

thm Functional_equation.functional_ldih6_x_126_n:

ldih6_x_126_n = *mk_126* (*rotate6 ldih_x_n*)

thm Functional_equation.functional_ldih_x_135_n:

ldih_x_135_n = *mk_135 ldih_x_n*

thm Functional_equation.functional_ldih3_x_135_n:

ldih3_x_135_n = *mk_135* (*rotate3 ldih_x_n*)

thm Functional_equation.functional_ldih5_x_135_n:

ldih5_x_135_n = *mk_135* (*rotate5 ldih_x_n*)

thm Functional_equation.functional1_rho:

$\forall y$::*real*. *rho y* = *y* ∗ (*const1* ∗ (*rh0* ∗ *DECIMAL* (*5*::*nat*) (*10*::*nat*))) + ((*1*::*real*) − *const1* ∗ *rh0*)

thm Functional_equation.functional1_flat_term_x:

$\forall y$::*real*. *flat_term_x y* = (*sqrt y* − *real_of_nat* (*2*::*nat*) ∗ *h0*) ∗ (*rh0* ∗ (*sol0* ∗ *DECIMAL* (*5*::*nat*) (*10*::*nat*)))

thm Functional_equation.functional1_lfun:

$\forall y$::*real*. *lfun y* = (*h0* − *y*) ∗ *rh0*

thm Functional_equation.functional_rhazim_x:

domain6 (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *x1* ∧ (*0*::*real*) ≤ *x2* ∧ (*0*::*real*) ≤ *x3* ∧ (*0*::*real*) ≤ *x4* ∧ (*0*::*real*) ≤ *x5* ∧ (*0*::*real*) ≤ *x6*) *rhazim_x* (*mul6* (*uni* (*rho*, *proj_y1*)) *dih_x*)

thm Functional_equation.functional_rhazim2_x:

rhazim2_x = *rotate2 rhazim_x*

thm Functional_equation.functional_rhazim3_x:

rhazim3_x = *rotate3 rhazim_x*

thm Functional_equation.functional_taum_x:

taum_x = *add6 rhazim_x* (*add6 rhazim2_x* (*sub6 rhazim3_x* (*constant6* (((*1*::*real*) + *const1*) ∗ *pi*))))

thm Functional_equation.functional_taum_x1:

$\forall$ (*a*::*real*) *b*::*real*. *domain6* (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *a* ∧ (*0*::*real*) ≤ *b*) (*taum_x1 a b*) (*compose6 taum_x four6 four6 four6* (*constant6* (*a* ∗ *a*)) (*constant6* (*b* ∗ *b*)) *proj_x1*)

thm Functional_equation.functional_taum_x2:

$\forall$ (*a*::*real*) *b*::*real*. *domain6* (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *a* ∧ (*0*::*real*) ≤ *b*) (*taum_x2 a b*) (*compose6 taum_x four6 four6 four6* (*constant6* (*a* ∗ *a*)) (*constant6* (*b* ∗ *b*)) *proj_x2*)

thm Functional_equation.functional_taum_x1_x2:

$\forall$ *a*::*real*. *domain6* (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *a*) (*taum_x1_x2 a*) (*compose6 taum_x four6 four6 four6* (*constant6* (*a* ∗ *a*)) *proj_x1 proj_x2*)

thm Functional_equation.functional_arclength_x1:

$\forall$ (*a*::*real*) *b*::*real*. *domain6* (λ(*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. (*0*::*real*) ≤ *a* ∧ (*0*::*real*) ≤ *b*) (*arclength_x1 a b*) (*compose6 arclength_x_123 proj_x1* (*constant6* (*a* ∗ *a*)) (*constant6* (*b* ∗ *b*)) *dummy6 dummy6 dummy6*)

thm Functional_equation.functional_arclength_x2:

$\forall (a::real)\ b::real.\ domain6\ (\lambda(x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ (0::real) \leq a \wedge (0::real) \leq b)\ (arclength\_x2\ a\ b)\ (compose6\ arclength\_x\_123\ proj\_x2\ (constant6\ (a * a))\ (constant6\ (b * b))\ dummy6\ dummy6\ dummy6)$

thm Functional_equation.functional_delta_126_x:

$\forall (a::real)\ (b::real)\ c::real.\ delta\_126\_x\ a\ b\ c = compose6\ delta\_x\ proj\_x1\ proj\_x2\ (constant6\ a)\ (constant6\ b)\ (constant6\ c)\ proj\_x6$

thm Functional_equation.functional_delta_234_x:

$\forall (a::real)\ (b::real)\ c::real.\ delta\_234\_x\ a\ b\ c = compose6\ delta\_x\ (constant6\ a)\ proj\_x2\ proj\_x3\ proj\_x4\ (constant6\ b)\ (constant6\ c)$

thm Functional_equation.functional_delta_135_x:

$\forall (a::real)\ (b::real)\ c::real.\ delta\_135\_x\ a\ b\ c = compose6\ delta\_x\ proj\_x1\ (constant6\ a)\ proj\_x3\ (constant6\ b)\ proj\_x5\ (constant6\ c)$

thm Functional_equation.functional_delta_sub1_x:

$\forall a::real.\ delta\_sub1\_x\ a = compose6\ delta\_x\ (constant6\ a)\ proj\_x2\ proj\_x3\ proj\_x4\ proj\_x5\ proj\_x6$

thm Functional_equation.functional_taum_sub1_x:

$\forall a::real.\ taum\_sub1\_x\ a = compose6\ taum\_x\ (constant6\ a)\ proj\_x2\ proj\_x3\ proj\_x4\ proj\_x5\ proj\_x6$

thm Functional_equation.functional_taum_sub246_x:

$\forall (a::real)\ (b::real)\ c::real.\ taum\_sub246\_x\ a\ b\ c = compose6\ taum\_x\ proj\_x1\ (constant6\ a)\ proj\_x3\ (constant6\ b)\ proj\_x5\ (constant6\ c)$

thm Functional_equation.functional_taum_sub345_x:

$\forall (a::real)\ (b::real)\ c::real.\ taum\_sub345\_x\ a\ b\ c = compose6\ taum\_x\ proj\_x1\ proj\_x2\ (constant6\ a)\ (constant6\ b)\ (constant6\ c)\ proj\_x6$

thm Functional_equation.functional_rhazim_x_div_sqrt_delta_posbranch:

$rhazim\_x\_div\_sqrtdelta\_posbranch = mul6\ (uni\ (rho,\ proj\_y1))\ dih\_x\_div\_sqrtdelta\_posbranch$

thm Functional_equation.functional_tau_residual:

$tau\_residual\_x = add6\ rhazim\_x\_div\_sqrtdelta\_posbranch\ (add6\ rhazim2\_x\_div\_sqrtdelta\_posbranch\ rhazim3\_x\_div\_sqrtdelta\_posbranch)$

thm Functional_equation.functional_halfbump_x1:

$halfbump\_x1 = promote1\_to\_6\ halfbump\_x$

thm Functional_equation.functional_halfbump_x4:

$halfbump\_x4 = rotate4\ halfbump\_x1$

thm Functional_equation.gamma2_x1_div_a:

$\forall\, m::real.\ gamma2\_x1\_div\_a\ m = promote1\_to\_6\ (gamma2\_x\_div\_azim\ m)$

thm Functional_equation.nonf_gamma2_x1_div_a:

$\forall\, (m::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma2\_x1\_div\_a\ m\ x1\ x2\ x3\ x4\ x5\ x6 = gamma2\_x\_div\_azim\ m\ x1$

thm Functional_equation.shift_scalar6:

$\forall\, (f::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ (g::?'f::type \Rightarrow ?'e::type \Rightarrow ?'d::type \Rightarrow ?'c::type \Rightarrow ?'b::type \Rightarrow ?'a::type \Rightarrow real)\ m::real.\ mul6\ (scalar6\ f\ m)\ g = mul6\ f\ (scalar6\ g\ m)$

thm Functional_equation.gamma3f_x_div_sqrtdelta_alt:

$\forall\, (m4::real)\ (m5::real)\ m6::real.\ gamma3f\_x\_div\_sqrtdelta\ m4\ m5\ m6 = sub6\ (constant6\ ((1::real)\ /\ real\_of\_nat\ (12::nat)))\ (sub6\ (scalar6\ (add6\ (mk\_456\ (rotate5\ sol\_euler\_x\_div\_sqrtdelta))\ (add6\ (mk\_456\ (rotate6\ sol\_euler\_x\_div\_sqrtdelta))\ (mk\_456\ (rotate4\ sol\_euler\_x\_div\_sqrtdelta))))\ (real\_of\_nat\ (2::nat) * (mm1\ /\ pi)))\ (scalar6\ (add6\ (mul6\ (uni\ (lfun,\ scalar6\ proj\_y4\ (DECIMAL\ (5::nat)\ (10::nat))))\ (scalar6\ (mk\_456\ (rotate4\ dih\_x\_div\_sqrtdelta\_posbranch))\ m4))\ (add6\ (mul6\ (uni\ (lfun,\ scalar6\ proj\_y5\ (DECIMAL\ (5::nat)\ (10::nat))))\ (scalar6\ (mk\_456\ (rotate5\ dih\_x\_div\_sqrtdelta\_posbranch))\ m5))\ (mul6\ (uni\ (lfun,\ scalar6\ proj\_y6\ (DECIMAL\ (5::nat)\ (10::nat))))\ (scalar6\ (mk\_456\ (rotate6\ dih\_x\_div\_sqrtdelta\_posbranch))\ m6))))\ (real\_of\_nat\ (8::nat) * (mm2\ /\ pi))))$

thm Functional_equation.nonf_gamma3_x:

$\forall\, (m1::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma3\_x\ m1\ x1\ x2\ x3\ x4\ x5\ x6 = vol\_x\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat))\ x4\ x5\ x6 - ((sol\_x\ x5\ (real\_of\_nat\ (2::nat))\ x4\ (real\_of\_nat\ (2::nat))\ x6\ (real\_of\_nat\ (2::nat)) + (sol\_x\ x6\ (real\_of\_nat\ (2::nat))\ x5\ (real\_of\_nat\ (2::nat))\ x4\ (real\_of\_nat\ (2::nat)) + sol\_x\ x4\ (real\_of\_nat\ (2::nat))\ x6\ (real\_of\_nat\ (2::nat))\ x5\ (real\_of\_nat\ (2::nat)))) * (real\_of\_nat\ (2::nat) * (mm1\ /\ pi)) - (lfun\ (sqrt\ x4 * DECIMAL\ (5::nat)\ (10::nat)) * m1 * dih\_x\ x4\ (real\_of\_nat\ (2::nat))\ x6\ (real\_of\_nat\ (2::nat))\ x5\ (real\_of\_nat\ (2::nat)) + (lfun\ (sqrt\ x5 * DECIMAL\ (5::nat)\ (10::nat)) * dih\_x\ x5\ (real\_of\_nat\ (2::nat))\ x4\ (real\_of\_nat\ (2::nat))\ x6\ (real\_of\_nat\ (2::nat)) + lfun\ (sqrt\ x6 * DECIMAL\ (5::nat)\ (10::nat)) * dih\_x\ x6\ (real\_of\_nat\ (2::nat))\ x5\ (real\_of\_nat\ (2::nat))\ x4\ (real\_of\_nat\ (2::nat)))) * (real\_of\_nat\ (8::nat) * (mm2\ /\ pi)))$

thm Functional_equation.nonf_gamma23_full8_x:

$\forall\, (m1::real)\ (x1::real)\ (x2::real)\ (x3::real)\ (x4::real)\ (x5::real)\ x6::real.\ gamma23\_full8\_x\ m1\ x1\ x2\ x3\ x4\ x5\ x6 = gamma3\_x\ m1\ (0::real)\ (0::real)\ (0::real)\ x1\ x2\ x6 + (gamma3\_x\ m1\ (0::real)\ (0::real)\ (0::real)\ x1\ x3\ x5 + (dih\_x\ x1\ x2\ x3\ x4\ x5\ x6 - (dih\_x\ x1\ x2\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (2::nat))\ x6 + dih\_x\ x1\ (real\_of\_nat\ (2::nat))\ x3\ (real\_of\_nat\ (2::nat))\ x5\ (real\_of\_nat\ (2::nat)))) * DECIMAL\ (8::nat)\ (1000::nat))$

thm Functional_equation.nonf_gamma23_keep135_x:

∀ (*m1*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma23_keep135_x m1 x1 x2 x3 x4 x5 x6* = *gamma3_x m1* (*0*::*real*) (*0*::*real*) (*0*::*real*) *x1 x3 x5* + (*dih_x x1 x2 x3 x4 x5 x6* − *dih_x x1* (*real_of_nat* (*2*::*nat*)) *x3* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*))) ∗ *DECIMAL* (*8*::*nat*) (*1000*::*nat*)

thm Functional_equation.nonf_gamma3f_x_div_sqrtdelta:

∀ (*m4*::*real*) (*m5*::*real*) (*m6*::*real*) (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *gamma3f_x_div_sqrtdelta m4 m5 m6 x1 x2 x3 x4 x5 x6* = (*1*::*real*) / *real_of_nat* (*12*::*nat*) − ((*sol_euler_x_div_sqrtdelta x5* (*real_of_nat* (*2*::*nat*)) *x4* (*real_of_nat* (*2*::*nat*)) *x6* (*real_of_nat* (*2*::*nat*)) + (*sol_euler_x_div_sqrtdelta x6* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)) *x4* (*real_of_nat* (*2*::*nat*)) + *sol_euler_x_div_sqrtdelta x4* (*real_of_nat* (*2*::*nat*)) *x6* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)))) ∗ (*real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*)) − (*lfun* (*sqrt x4* ∗ *DECIMAL* (*5*::*nat*) (*10*::*nat*)) ∗ *m4* ∗ *dih_x_div_sqrtdelta_posbranch x4* (*real_of_nat* (*2*::*nat*)) *x6* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)) + (*lfun* (*sqrt x5* ∗ *DECIMAL* (*5*::*nat*) (*10*::*nat*)) ∗ *m5* ∗ *dih_x_div_sqrtdelta_posbranch x5* (*real_of_nat* (*2*::*nat*)) *x4* (*real_of_nat* (*2*::*nat*)) *x6* (*real_of_nat* (*2*::*nat*)) + *lfun* (*sqrt x6* ∗ *DECIMAL* (*5*::*nat*) (*10*::*nat*)) ∗ *m6* ∗ *dih_x_div_sqrtdelta_posbranch x6* (*real_of_nat* (*2*::*nat*)) *x5* (*real_of_nat* (*2*::*nat*)) *x4* (*real_of_nat* (*2*::*nat*)))) ∗ (*real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*)))

thm Optimize.split_2h0:

∀ (*a*::*real*) (*b*::*real*) *y*::*real*. *a* ≤ *y* ∧ *y* ≤ *b* ⟶ *a* ≤ *y* ∧ *y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ∨ *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *y* ∧ *y* ≤ *b*

thm Optimize.split_2h0_strict:

∀ (*a*::*real*) (*b*::*real*) *y*::*real*. *a* ≤ *y* ∧ *y* ≤ *b* ⟶ *a* ≤ *y* ∧ *y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ∨ *real_of_nat* (*2*::*nat*) ∗ *h0* < *y* ∧ *y* ≤ *b*

thm Optimize.h0cutA:

∀ *y*≤*real_of_nat* (*2*::*nat*) ∗ *h0*. *h0cut y* = (*1*::*real*)

thm Optimize.h0cutB:

∀ *y*>*real_of_nat* (*2*::*nat*) ∗ *h0*. *h0cut y* = (*0*::*real*)

thm Optimize.h0cutC:

∀ *y*≤*real_of_nat* (*2*::*nat*) ∗ *hminus*. *h0cut y* = (*1*::*real*)

thm Nonlinear_lemma.h0_lt_gt_conjunct7:

real_of_nat (*2*::*nat*) ∗ *hminus* ≤ (*?y*::*real*) ⟶ (*0*::*real*) ≤ *?y*

thm Nonlinear_lemma.h0_lt_gt_conjunct5:

real_of_nat (*2*::*nat*) ≤ (*?y*::*real*) ⟶ (*0*::*real*) ≤ *?y*

thm Nonlinear_lemma.h0_lt_gt_conjunct4:

real_of_nat (*2*::*nat*) ∗ *h0* ≤ (*?y*::*real*) ⟶ (*0*::*real*) ≤ *?y*

thm Nonlinear_lemma.h0_lt_gt_conjunct3:

sqrt8 ≤ (*?y*::*real*) ⟶ *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *?y*

thm Nonlinear_lemma.h0_lt_gt_conjunct2:

(*?y*::*real*) ≤ *real_of_nat* (*2*::*nat*) ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*

thm Nonlinear_lemma.h0_lt_gt_conjunct1:

DECIMAL (*28*::*nat*) (*10*::*nat*) ≤ (*?y*::*real*) ⟶ *real_of_nat* (*2*::*nat*) ∗ *h0* ≤ *?y*

thm Nonlinear_lemma.h0_lt_gt_conjunct0:

(*?y*::*real*) ≤ *DECIMAL* (*201*::*nat*) (*100*::*nat*) ⟶ *?y* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0*

thm Nonlinear_lemma.tame_table_d_values_conjunct12:

tame_table_d (*2*::*nat*) (*3*::*nat*) = *DECIMAL* (*10006*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct11:

tame_table_d (*3*::*nat*) (*2*::*nat*) = *DECIMAL* (*8277*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct3:

tame_table_d (*4*::*nat*) (*1*::*nat*) = *DECIMAL* (*6548*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct9:

tame_table_d (*5*::*nat*) (*0*::*nat*) = *DECIMAL* (*4819*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct8:

tame_table_d (*0*::*nat*) (*4*::*nat*) = *DECIMAL* (*8976*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct7:

tame_table_d (*1*::*nat*) (*3*::*nat*) = *DECIMAL* (*7247*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct6:

tame_table_d (*2*::*nat*) (*2*::*nat*) = *DECIMAL* (*5518*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct5:

tame_table_d (*3*::*nat*) (*1*::*nat*) = *DECIMAL* (*3789*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct4:

tame_table_d (*6*::*nat*) (*0*::*nat*) = *DECIMAL* (*7578*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct2:

tame_table_d (*0*::*nat*) (*3*::*nat*) = *DECIMAL* (*4488*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct1:

tame_table_d (*1*::*nat*) (*2*::*nat*) = *DECIMAL* (*2759*::*nat*) (*10000*::*nat*)

thm Nonlinear_lemma.tame_table_d_values_conjunct0:

tame_table_d (2::nat) (1::nat) = DECIMAL (103::nat) (1000::nat)

thm Nonlinear_lemma.atn2_0y_conjunct1:

atn2 (0::real, − real_of_nat (64::nat)) = − pi / real_of_nat (2::nat)

thm Nonlinear_lemma.atn2_0y_conjunct0:

atn2 (0::real, real_of_nat (64::nat)) = pi / real_of_nat (2::nat)

thm Nonlinear_lemma.delta_x4_eq64_conjunct1:

delta_x4 (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) = real_of_nat (64::nat)

thm Nonlinear_lemma.delta_x4_eq64_conjunct0:

delta_x4 (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) = − real_of_nat (64::nat)

thm Nonlinear_lemma.delta_x_eq0_conjunct1:

delta_x (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) = (0::real)

thm Nonlinear_lemma.delta_x_eq0_conjunct0:

delta_x (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) (real_of_nat (8::nat)) (real_of_nat (4::nat)) (real_of_nat (4::nat)) = (0::real)

thm Optimize.gamma4f_delta0:

NONLIN ⟶ (∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. ineq [(sqrt8, y1, sqrt8), (real_of_nat (2::nat), y2, real_of_nat (2::nat)), (real_of_nat (2::nat), y3, real_of_nat (2::nat)), (sqrt8, y4, sqrt8), (real_of_nat (2::nat), y5, real_of_nat (2::nat)), (real_of_nat (2::nat), y6, real_of_nat (2::nat))] (gamma4f y1 y2 y3 y4 y5 y6 lmfun = (0::real)))

thm Vol1.sphere:

∀ x::(real, 3) cart ⇒ bool. sphere x = (∃ (v::(real, 3) cart) r::real. (0::real) < r ∧ x = GSPEC (λGEN%PVAR%210::(real, 3) cart. ∃ w::(real, 3) cart. SETSPEC GEN%PVAR%210 (vector_norm (vector_sub w v) = r) w))

thm DEF_radial_norm:

radial_norm = (λ(_2390449::real) (_2390450::(real, ?'a::type) cart) _2390451::(real, ?'a::type) cart ⇒ bool. SUBSET _2390451 (normball _2390450 _2390449) ∧ (∀ u::(real, ?'a::type) cart. IN (vector_add _2390450 u) _2390451 ⟶ (∀ t::real. (0::real) < t ∧ t ∗ vector_norm u < _2390449 ⟶ IN (vector_add _2390450 (% t u)) _2390451)))

thm Vol1.radial_norm:

$\forall (r::real)\ (x::(real,\ ?'a::type)\ cart)\ C::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ x\ C = (SUBSET\ C\ (normball\ x\ r) \wedge (\forall u::(real,\ ?'a::type)\ cart.\ IN\ (vector\_add\ x\ u)\ C \longrightarrow (\forall t::real.\ (0::real) < t \wedge t * vector\_norm\ u < r \longrightarrow IN\ (vector\_add\ x\ (\%\ t\ u))\ C)))$

thm DEF_eventually_radial_norm:

$eventually\_radial\_norm = (\lambda(\_2390470::(real,\ ?'a::type)\ cart)\ \_2390471::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists r{>}0::real.\ radial\_norm\ r\ \_2390470\ (HOL\_Light\_Import.INTER\ \_2390471\ (normball\ \_2390470\ r)))$

thm Vol1.eventually_radial_norm:

$\forall (C::(real,\ ?'a::type)\ cart \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ eventually\_radial\_norm\ x\ C = (\exists r{>}0::real.\ radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)))$

thm Vol1.c_cone:

$\forall (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ r::real.\ c\_cone\ (v,\ w,\ r) = GSPEC\ (\lambda GEN\%PVAR\%211::(real,\ 3)\ cart.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%211\ (dot\ (vector\_sub\ x\ v)\ w = vector\_norm\ (vector\_sub\ x\ v) * (vector\_norm\ w * r))\ x)$

thm Vol1.th1:

$\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ [a,\ b,\ c] = [a,\ b,\ c]$

thm Vol1.dodai:

$\forall (a::?'a::type)\ (b::?'a::type)\ c::?'a::type.\ length\ [a,\ b,\ c] = (3::nat)$

thm Vol1.th3:

$\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ (vec\ (1::nat))\ i = (1::real)$

thm Vol1.identity_scale:

$scale\ (vec\ (1::nat)) = id$

thm Vol1.th4:

$\forall S::?'a::type \Rightarrow bool.\ IMAGE\ id\ S = S$

thm Vol1.SET_EQ:

$((?A::(real,\ 3)\ cart \Rightarrow bool) = (?B::(real,\ 3)\ cart \Rightarrow bool)) = ((\forall a::(real,\ 3)\ cart.\ IN\ a\ ?A \longrightarrow IN\ a\ ?B) \wedge (\forall a::(real,\ 3)\ cart.\ IN\ a\ ?B \longrightarrow IN\ a\ ?A))$

thm Vol1.scale_mul:

$\forall (s::real)\ (t::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ scale\ (\%\ s\ t)\ x = \%\ s\ (scale\ t\ x)$

thm Vol1.normball_ellip0:

$\forall r::real.\ normball\ (vec\ (0::nat))\ r = ellipsoid\ (vec\ (1::nat))\ r$

thm Vol1.trans_normball:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ normball\ x\ r = IMAGE\ (vector\_add\ x)\ (normball\ (vec\ (0::nat))\ r)$

thm Vol1.measurable_normball0:

$\forall r::real.\ measurable\ (normball\ (vec\ (0::nat))\ r)$

thm Vol1.measurable_normball:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ measurable\ (normball\ x\ r)$

thm Vol1.rsduong:

$\forall (s::real)\ r::real.\ (0::real) < s \wedge s < r \longrightarrow (0::real) < r\ /\ s$

thm Vol1.rsnon_zero:

$\forall (s::real)\ r::real.\ (0::real) < s \wedge s < r \longrightarrow r\ /\ s \neq (0::real)$

thm Vol1.rduong:

$\forall (s::real)\ r::real.\ (0::real) < s \wedge s < r \longrightarrow (0::real) < r$

thm Vol1.rnon_zero:

$\forall (s::real)\ r::real.\ (0::real) < s \wedge s < r \longrightarrow r \neq (0::real)$

thm Vol1.rs_sr_unit:

$\forall (s::real)\ r::real.\ (0::real) < s \wedge s < r \longrightarrow s\ /\ r * (r\ /\ s) = (1::real)$

thm Vol1.trans_strech_trans_radial:

$\forall (r::real)\ (s::real)\ (x::(real,\ 3)\ cart)\ C::(real,\ 3)\ cart \Rightarrow bool.\ radial\_norm\ r\ x\ C \wedge (0::real) < s \wedge s < r \longrightarrow HOL\_Light\_Import.INTER\ C\ (normball\ x\ s) = IMAGE\ (vector\_add\ x)\ (IMAGE\ (scale\ (\%\ (s\ /\ r)\ (vec\ (1::nat))))\ (IMAGE\ (vector\_add\ (vector\_neg\ x))\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))))$

thm DEF_volume_prop_fix:

$volume\_prop\_fix = (\lambda\_2391204::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow real.\ (\forall C::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ C \longrightarrow (0::real) \leq \_2391204\ C) \wedge (\forall Z::(real,\ 3)\ cart \Rightarrow bool.\ negligible\ Z \longrightarrow \_2391204\ Z = (0::real)) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ Y::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ X \wedge measurable\ Y \wedge negligible\ (SDIFF\ X\ Y) \longrightarrow \_2391204\ X = \_2391204\ Y) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart.\ measurable\ X \wedge measurable\ (IMAGE\ (scale\ t)\ X) \longrightarrow \_2391204\ (IMAGE\ (scale\ t)\ X) = |\$\ t\ (1::nat) * (\$\ t\ (2::nat) * \$\ t\ (3::nat))| * \_2391204\ X) \wedge (\forall (X::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ measurable\ X \longrightarrow \_2391204\ (IMAGE\ (vector\_add\ v)\ X) = \_2391204\ X) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ (v3::(real,\ 3)\ cart)\ r::real.\ (0::real) < r \wedge \neg\ coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))) \longrightarrow \_2391204\ (solid\_triangle\ v0\ (INSERT\ v1\ (INSERT\ v2\ (INSERT\ v3\ EMPTY)))\ r) = vol\_solid\_triangle\ v0\ v1\ v2\ v3\ r) \wedge (\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,$

3) cart. _2391204 (conv0 (INSERT v0 (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))))) = vol_conv v0 v1 v2 v3) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (h::real) a::real. ¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ∧ (0::real) ≤ h ∧ (0::real) < a ∧ a ≤ (1::real) ⟶ _2391204 (HOL_Light_Import.INTER (frustt v0 v1 h a) (wedge v0 v1 v2 v3)) = vol_frustt_wedge v0 v1 v2 v3 h a) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (r::real) c::real. ¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ∧ (0::real) ≤ r ∧ − (1::real) ≤ c ∧ c ≤ (1::real) ⟶ _2391204 (HOL_Light_Import.INTER (conic_cap v0 v1 r c) (wedge v0 v1 v2 v3)) = vol_conic_cap_wedge v0 v1 v2 v3 r c) ∧ (∀ (a::(real, 3) cart) b::(real, 3) cart. _2391204 (rect a b) = vol_rect a b) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) r::real. ¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ∧ (0::real) ≤ r ⟶ _2391204 (HOL_Light_Import.INTER (normball v0 r) (wedge v0 v1 v2 v3)) = vol_ball_wedge v0 v1 v2 v3 r))

thm Vol1.volume_prop_fix:

volume_prop_fix HOL_Light_Import.measure = ((∀ C::(real, 3) cart ⇒ bool. measurable C ⟶ (0::real) ≤ HOL_Light_Import.measure C) ∧ (∀ Z::(real, 3) cart ⇒ bool. negligible Z ⟶ HOL_Light_Import.measure Z = (0::real)) ∧ (∀ (X::(real, 3) cart ⇒ bool) Y::(real, 3) cart ⇒ bool. measurable X ∧ measurable Y ∧ negligible (SDIFF X Y) ⟶ HOL_Light_Import.measure X = HOL_Light_Import.measure Y) ∧ (∀ (X::(real, 3) cart ⇒ bool) t::(real, 3) cart. measurable X ∧ measurable (IMAGE (scale t) X) ⟶ HOL_Light_Import.measure (IMAGE (scale t) X) = |$ t (1::nat) ∗ ($ t (2::nat) ∗ $ t (3::nat))| ∗ HOL_Light_Import.measure X) ∧ (∀ (X::(real, 3) cart ⇒ bool) v::(real, 3) cart. measurable X ⟶ HOL_Light_Import.measure (IMAGE (vector_add v) X) = HOL_Light_Import.measure X) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) r::real. (0::real) < r ∧ ¬ coplanar (INSERT v0 (INSERT v1 (INSERT v2 (INSERT v3 EMPTY)))) ⟶ HOL_Light_Import.measure (solid_triangle v0 (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))) r) = vol_solid_triangle v0 v1 v2 v3 r) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. HOL_Light_Import.measure (conv0 (INSERT v0 (INSERT v1 (INSERT v2 (INSERT v3 EMPTY))))) = vol_conv v0 v1 v2 v3) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (h::real) a::real. ¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ∧ (0::real) ≤ h ∧ (0::real) < a ∧ a ≤ (1::real) ⟶ HOL_Light_Import.measure (HOL_Light_Import.INTER (frustt v0 v1 h a) (wedge v0 v1 v2 v3)) = vol_frustt_wedge v0 v1 v2 v3 h a) ∧ (∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) (r::real) c::real. ¬ collinear (INSERT v0 (INSERT v1 (INSERT v2 EMPTY))) ∧ ¬ collinear (INSERT v0 (INSERT v1 (INSERT v3 EMPTY))) ∧ (0::real) ≤ r ∧ − (1::real) ≤ c ∧ c ≤ (1::real) ⟶ HOL_Light_Import.measure

(*HOL_Light_Import.INTER* (*conic_cap v0 v1 r c*) (*wedge v0 v1 v2 v3*)) = *vol_conic_cap_wedge v0 v1 v2 v3 r c*) ∧ (∀ (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *HOL_Light_Import.measure* (*rect a b*) = *vol_rect a b*) ∧ (∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *r*::*real*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) ∧ (*0*::*real*) ≤ *r* ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*normball v0 r*) (*wedge v0 v1 v2 v3*)) = *vol_ball_wedge v0 v1 v2 v3 r*))

thm volume_props_conjunct10:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *r*::*real*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) ∧ (*0*::*real*) ≤ *r* ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*normball v0 r*) (*wedge v0 v1 v2 v3*)) = *vol_ball_wedge v0 v1 v2 v3 r*

thm volume_props_conjunct8:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*r*::*real*) *c*::*real*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) ∧ (*0*::*real*) ≤ *r* ∧ − (*1*::*real*) ≤ *c* ∧ *c* ≤ (*1*::*real*) ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*conic_cap v0 v1 r c*) (*wedge v0 v1 v2 v3*)) = *vol_conic_cap_wedge v0 v1 v2 v3 r c*

thm volume_props_conjunct7:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) (*h*::*real*) *a*::*real*. ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *collinear* (*INSERT v0* (*INSERT v1* (*INSERT v3 EMPTY*))) ∧ (*0*::*real*) ≤ *h* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*frustt v0 v1 h a*) (*wedge v0 v1 v2 v3*)) = *vol_frustt_wedge v0 v1 v2 v3 h a*

thm volume_props_conjunct5:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*v3*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) < *r* ∧ ¬ *coplanar* (*INSERT v0* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*)))) ⟶ *HOL_Light_Import.measure* (*solid_triangle v0* (*INSERT v1* (*INSERT v2* (*INSERT v3 EMPTY*))) *r*) = *vol_solid_triangle v0 v1 v2 v3 r*

thm volume_props_conjunct0:

∀ *C*::(*real*, *3*) *cart* ⇒ *bool*. *measurable C* ⟶ (*0*::*real*) ≤ *HOL_Light_Import.measure C*

thm Vol1.VOLUME_FIX:

volume_prop_fix HOL_Light_Import.measure

thm Vol1.lemma_r_r′:

$\forall (C::(real, 3)\ cart \Rightarrow bool)\ (x::(real, 3)\ cart)\ (r::real)\ s::real.\ measurable\ C \wedge volume\_prop\_fix\ HOL\_Light\_Import.measure \wedge radial\_norm\ r\ x\ C \wedge (0::real) < s \wedge s < r \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) = HOL\_Light\_Import.measure\ C * (s\ /\ r)^{3::nat}$

thm Vol1.lemma_r_r′_fix:

$\forall (C::(real, 3)\ cart \Rightarrow bool)\ (x::(real, 3)\ cart)\ (r::real)\ s::real.\ measurable\ C \wedge radial\_norm\ r\ x\ C \wedge (0::real) < s \wedge s < r \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) = HOL\_Light\_Import.measure\ C * (s\ /\ r)^{3::nat}$

thm Vol1.normball_subset:

$\forall (x::(real, 3)\ cart)\ (r::real)\ r'::real.\ (0::real) < r' \wedge r' < r \longrightarrow SUBSET\ (normball\ x\ r')\ (normball\ x\ r)$

thm Vol1.subset_inter:

$\forall (A::(real, 3)\ cart \Rightarrow bool)\ B::(real, 3)\ cart \Rightarrow bool.\ SUBSET\ A\ B \longrightarrow HOL\_Light\_Import.INTER\ A\ B = A$

thm Vol1.normball_eq:

$\forall (C::(real, 3)\ cart \Rightarrow bool)\ (x::(real, 3)\ cart)\ (r::real)\ r'::real.\ (0::real) < r' \wedge r' < r \longrightarrow HOL\_Light\_Import.INTER\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))\ (normball\ x\ r') = HOL\_Light\_Import.INTER\ C\ (normball\ x\ r')$

thm Vol1.radial_normball:

$\forall (r::real)\ (x::(real, ?'a::type)\ cart)\ C::(real, ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ x\ C \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))$

thm Vol1.pow3_identity:

$(?b::real) = (?a::real) * ((?r'::real)\ /\ (?r::real))^{3::nat} \wedge (0::real) < ?r \wedge (0::real) < ?r' \longrightarrow ?a\ /\ ?r^{3::nat} = ?b\ /\ ?r'^{3::nat}$

thm Vol1.pre_def1_4_3b:

$\forall (x::(real, 3)\ cart)\ C::(real, 3)\ cart \Rightarrow bool.\ volume\_prop\_fix\ HOL\_Light\_Import.measure \longrightarrow (\exists\ s::real.\ \forall\ r::real.\ (0::real) < r \wedge measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \longrightarrow s = real\_of\_nat\ (3::nat) * (HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))\ /\ r^{3::nat}))$

thm Vol1.pre_def_4_3b:

$\exists\ s::(real, 3)\ cart \Rightarrow ((real, 3)\ cart \Rightarrow bool) \Rightarrow real.\ \forall (x::(real, 3)\ cart)\ (C::(real, 3)\ cart \Rightarrow bool)\ r::real.\ (0::real) < r \wedge measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball$

$x\ r)) \longrightarrow s\ x\ C = real\_of\_nat\ (3::nat) * (HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))\ /\ r^{3::nat})$

thm DEF_sol:

$sol = (SOME\ s::nat \Rightarrow (real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow real.\ \forall (\_2393453::nat)\ (x::(real,\ 3)\ cart)\ (C::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ (0::real) < r \wedge measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \longrightarrow s\ \_2393453\ x\ C = real\_of\_nat\ (3::nat) * (HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))\ /\ r^{3::nat}))\ (95::nat)$

thm Pack_defs.sol:

$\forall (x::(real,\ 3)\ cart)\ (C::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ (0::real) < r \wedge measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \longrightarrow sol\ x\ C = real\_of\_nat\ (3::nat) * (HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r))\ /\ r^{3::nat})$

thm Vol1.AFF_GT_1_3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%212::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%212\ (\exists (t1::real)\ (t2::real)\ (t3::real)\ t4::real.\ (0::real) < t2 \wedge (0::real) < t3 \wedge (0::real) < t4 \wedge t1 + (t2 + (t3 + t4)) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ u)\ (vector\_add\ (\%\ t3\ v)\ (\%\ t4\ w))))\ y)$

thm Vol1.aff_normball:

$\forall (t::real)\ (x::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ r::real.\ (0::real) < r \wedge (0::real) < t \wedge t * vector\_norm\ u < r \wedge IN\ (vector\_add\ x\ u)\ (normball\ x\ r) \longrightarrow IN\ (vector\_add\ x\ (\%\ t\ u))\ (normball\ x\ r)$

thm Vol1.aff_gt_radial:

$\forall (x::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ r::real.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge (0::real) < r \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY))))\ (normball\ x\ r))$

thm Vol1.tr5:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (C::(real,\ ?'a::type)\ cart \Rightarrow bool)\ C'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ v0\ C \wedge radial\_norm\ r\ v0\ C' \longrightarrow (\forall u::(real,\ ?'a::type)\ cart.\ IN\ (vector\_add\ v0\ u)\ (HOL\_Light\_Import.INTER\ C\ C') \longrightarrow (\forall t::real.\ (0::real) < t \wedge t * vector\_norm\ u < r \longrightarrow IN\ (vector\_add\ v0\ (\%\ t\ u))\ (HOL\_Light\_Import.INTER\ C\ C')))$

thm Vol1.tr6:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (C::(real,\ ?'a::type)\ cart \Rightarrow bool)\ C'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ v0\ C \wedge radial\_norm\ r\ v0\ C' \longrightarrow SUBSET\ (HOL\_Light\_Import.INTER\ C\ C')\ (normball\ v0\ r)$

thm Vol1.inter_radial:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (C::(real,\ ?'a::type)\ cart \Rightarrow bool)\ C'::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ v0\ C \wedge radial\_norm\ r\ v0\ C' \longrightarrow radial\_norm\ r\ v0\ (HOL\_Light\_Import.INTER\ C\ C')$

thm DEF_cube:

$cube = (\lambda\_2393731::(real,\ 3)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%213::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%213\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ \_2393731\ i < \$\ y\ i \wedge \$\ y\ i < \$\ \_2393731\ i + (1::real))\ y))$

thm Vol1.cube:

$\forall x::(real,\ 3)\ cart.\ cube\ x = GSPEC\ (\lambda GEN\%PVAR\%213::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%213\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ x\ i < \$\ y\ i \wedge \$\ y\ i < \$\ x\ i + (1::real))\ y)$

thm DEF_hinhcau:

$hinhcau = (\lambda(\_2393736::(real,\ 3)\ cart)\ \_2393737::real.\ GSPEC\ (\lambda GEN\%PVAR\%214::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%214\ (vector\_norm\ (vector\_sub\ \_2393736\ y) < \_2393737)\ y))$

thm Vol1.hinhcau:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ hinhcau\ x\ r = GSPEC\ (\lambda GEN\%PVAR\%214::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%214\ (vector\_norm\ (vector\_sub\ x\ y) < r)\ y)$

thm DEF_set_of_cube:

$set\_of\_cube = (\lambda\_2393748::(real,\ 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%215::(real,\ 3)\ cart \Rightarrow bool.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%215\ (IN\ x\ \_2393748)\ (cube\ x)))$

thm Vol1.set_of_cube:

$\forall S::(real,\ 3)\ cart \Rightarrow bool.\ set\_of\_cube\ S = GSPEC\ (\lambda GEN\%PVAR\%215::(real,\ 3)\ cart \Rightarrow bool.\ \exists x::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%215\ (IN\ x\ S)\ (cube\ x))$

thm DEF_union_of_cube:

$union\_of\_cube = (\lambda(\_2393753::(real,\ 3)\ cart)\ \_2393754::real.\ UNIONS\ (set\_of\_cube\ (hinhcau\ \_2393753\ \_2393754)))$

thm Vol1.union_of_cube:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ union\_of\_cube\ x\ r = UNIONS\ (set\_of\_cube\ (hinhcau\ x\ r))$

thm DEF_int_ball:

$int\_ball = (\lambda(\_2393765::(real,\ 3)\ cart)\ \_2393766::real.\ HOL\_Light\_Import.INTER\ (GSPEC\ (\lambda GEN\%PVAR\%216::(real,\ 3)\ cart.\ \exists\, y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%216\ (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow integer\ (\$\ y\ i))\ y))\ (hinhcau\ \_2393765\ \_2393766))$

thm Pack1.int_ball:

$\forall\, (x::(real,\ 3)\ cart)\ r::real.\ int\_ball\ x\ r = HOL\_Light\_Import.INTER\ (GSPEC\ (\lambda GEN\%PVAR\%216::(real,\ 3)\ cart.\ \exists\, y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%216\ (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow integer\ (\$\ y\ i))\ y))\ (hinhcau\ x\ r)$

thm DEF_union_of_int_cube:

$union\_of\_int\_cube = (\lambda(\_2393777::(real,\ 3)\ cart)\ \_2393778::real.\ UNIONS\ (set\_of\_cube\ (int\_ball\ \_2393777\ \_2393778)))$

thm Vol1.union_of_int_cube:

$\forall\, (x::(real,\ 3)\ cart)\ r::real.\ union\_of\_int\_cube\ x\ r = UNIONS\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.map0:

$map0 = cube$

thm Vol1.COMPONENT_LE_NORM_3:

$\forall\, (x::(real,\ 3)\ cart)\ i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow |\$\ x\ i| \leq vector\_norm\ x$

thm Vol1.component_hinhcau_bound:

$\forall\, (p::(real,\ 3)\ cart)\ (r::real)\ x::(real,\ 3)\ cart.\ IN\ x\ (hinhcau\ p\ r) \longrightarrow (\forall\, i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ p\ i - r \leq \$\ x\ i \wedge \$\ x\ i \leq \$\ p\ i + r)$

thm DEF_int_interval:

$int\_interval = (\lambda(\_2393825::real)\ \_2393826::real.\ GSPEC\ (\lambda GEN\%PVAR\%217::real.\ \exists\, x::real.\ SETSPEC\ GEN\%PVAR\%217\ (integer\ x \wedge \_2393825 \leq x \wedge x \leq \_2393826)\ x))$

thm Vol1.int_interval:

$\forall\, (a::real)\ b::real.\ int\_interval\ a\ b = GSPEC\ (\lambda GEN\%PVAR\%217::real.\ \exists\, x::real.\ SETSPEC\ GEN\%PVAR\%217\ (integer\ x \wedge a \leq x \wedge x \leq b)\ x)$

thm Vol1.finite_int_interval:

$\forall\, (a::real)\ b::real.\ FINITE\ (int\_interval\ a\ b)$

thm DEF_vector_to_pair:

$vector\_to\_pair = (\lambda\_2393837::(real,\ 3)\ cart.\ (\$\ \_2393837\ (1::nat),\ \$\ \_2393837\ (2::nat),\ \$\ \_2393837\ (3::nat)))$

thm Vol1.vector_to_pair:

$\forall x{::}(real,\ 3)\ cart.\ vector\_to\_pair\ x = (\$\ x\ (1{::}nat),\ \$\ x\ (2{::}nat),\ \$\ x\ (3{::}nat))$

thm Vol1.lm1:

$\forall (x{::}(real,\ 3)\ cart)\ (r{::}real)\ y{::}(real,\ 3)\ cart.\ IN\ y\ (int\_ball\ x\ r) \longrightarrow (\forall i{::}nat.\ (1{::}nat) \leq i \wedge i \leq (3{::}nat) \longrightarrow IN\ (\$\ y\ i)\ (int\_interval\ (\$\ x\ i - r)\ (\$\ x\ i + r)))$

thm Vol1.lm2:

$\forall (x{::}(real,\ 3)\ cart)\ (r{::}real)\ y{::}(real,\ 3)\ cart.\ IN\ y\ (int\_ball\ x\ r) \longrightarrow IN\ (vector\_to\_pair\ y)\ (CROSS\ (int\_interval\ (\$\ x\ (1{::}nat) - r)\ (\$\ x\ (1{::}nat) + r))\ (CROSS\ (int\_interval\ (\$\ x\ (2{::}nat) - r)\ (\$\ x\ (2{::}nat) + r))\ (int\_interval\ (\$\ x\ (3{::}nat) - r)\ (\$\ x\ (3{::}nat) + r))))$

thm Vol1.vector_eq1:

$\forall (x{::}(real,\ 3)\ cart)\ y{::}(real,\ 3)\ cart.\ (x = y) = (vector\_sub\ x\ y = vec\ (0{::}nat))$

thm Vol1.trip_eq:

$\forall (x{::}(real,\ 3)\ cart)\ y{::}(real,\ 3)\ cart.\ ((\$\ x\ (1{::}nat),\ \$\ x\ (2{::}nat),\ \$\ x\ (3{::}nat)) = (\$\ y\ (1{::}nat),\ \$\ y\ (2{::}nat),\ \$\ y\ (3{::}nat))) = (\forall i{::}nat.\ (1{::}nat) \leq i \wedge i \leq (3{::}nat) \longrightarrow \$\ x\ i = \$\ y\ i)$

thm Vol1.int_ball_subset_CROSS:

$\forall (x{::}(real,\ 3)\ cart)\ r{::}real.\ INJ\ vector\_to\_pair\ (int\_ball\ x\ r)\ (CROSS\ (int\_interval\ (\$\ x\ (1{::}nat) - r)\ (\$\ x\ (1{::}nat) + r))\ (CROSS\ (int\_interval\ (\$\ x\ (2{::}nat) - r)\ (\$\ x\ (2{::}nat) + r))\ (int\_interval\ (\$\ x\ (3{::}nat) - r)\ (\$\ x\ (3{::}nat) + r))))$

thm Pack1.FINITE_IMAGE_INJ:

$\forall (h{::}(real,\ 3)\ cart \Rightarrow ?'a{::}type)\ (s{::}(real,\ 3)\ cart \Rightarrow bool)\ t{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}(real,\ 3)\ cart.\ IN\ x\ s \longrightarrow IN\ (h\ x)\ t) \wedge INJ\ h\ s\ t \wedge FINITE\ t \longrightarrow FINITE\ s$

thm Vol1.finite_cross_int_interval:

$\forall (x{::}(real,\ 3)\ cart)\ r{::}real.\ FINITE\ (CROSS\ (int\_interval\ (\$\ x\ (1{::}nat) - r)\ (\$\ x\ (1{::}nat) + r))\ (CROSS\ (int\_interval\ (\$\ x\ (2{::}nat) - r)\ (\$\ x\ (2{::}nat) + r))\ (int\_interval\ (\$\ x\ (3{::}nat) - r)\ (\$\ x\ (3{::}nat) + r))))$

thm Pack1.finite_int_ball:

$\forall (x{::}(real,\ 3)\ cart)\ r{::}real.\ FINITE\ (int\_ball\ x\ r)$

thm Vol1.disjoint_line_interval:

$\forall (x{::}real)\ y{::}real.\ integer\ x \wedge integer\ y \wedge (\exists z{>}x.\ z < x + (1{::}real) \wedge y < z \wedge z < y + (1{::}real)) \longrightarrow x = y$

thm Vol1.vector_eq2:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (\forall\ i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ x\ i = \$\ y\ i) \longrightarrow x = y$

thm Vol1.nonempty_cube:

$\forall\ x::(real,\ 3)\ cart.\ cube\ x \neq EMPTY$

thm Vol1.disjoint_cube:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ (\forall\ i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow integer\ (\$\ x\ i) \wedge integer\ (\$\ y\ i)) \longrightarrow (\neg\ DISJOINT\ (cube\ x)\ (cube\ y)) = (x = y)$

thm Vol1.inj_map0:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ INJ\ map0\ (int\_ball\ x\ r)\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.surj_map0:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ SURJ\ map0\ (int\_ball\ x\ r)\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.bij_map0:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ BIJ\ map0\ (int\_ball\ x\ r)\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.surj2_map0:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ IMAGE\ map0\ (int\_ball\ x\ r) = set\_of\_cube\ (int\_ball\ x\ r)$

thm Vol1.int_ball_card_eq:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ CARD\ (int\_ball\ x\ r) = CARD\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.cube_eq_interval:

$\forall\ x::(real,\ 3)\ cart.\ cube\ x = open\_interval\ (x,\ vector\_add\ x\ (lambda\ (\lambda i::nat.\ 1::real)))$

thm Vol1.measurable_cube:

$\forall\ x::(real,\ 3)\ cart.\ measurable\ (cube\ x)$

thm Vol1.finite_set_of_cube:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ FINITE\ (set\_of\_cube\ (int\_ball\ x\ r))$

thm Vol1.measurable_unions_cube:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ measurable\ (UNIONS\ (set\_of\_cube\ (int\_ball\ x\ r)))$

thm Vol1.non_empty_cinterval:

$\forall\ x::(real,\ 3)\ cart.\ closed\_interval\ [(x,\ vector\_add\ x\ (lambda\ (\lambda i::nat.\ 1::real)))] \neq EMPTY$

thm Vol1.product_3:

$\forall f$::*nat* ⇒ *real*. *product* (*dotdot* (*1*::*nat*) (*3*::*nat*)) *f* = *f* (*1*::*nat*) ∗ (*f* (*2*::*nat*) ∗ *f* (*3*::*nat*))

thm Vol1.interval_upper_lowerbound:

∀ *x*::(*real*, *3*) *cart*. *interval_upperbound* (*closed_interval* [(*x*, *vector_add x* (*lambda* (λ*i*::*nat*. *1*::*real*)))]) = *vector_add x* (*lambda* (λ*i*::*nat*. *1*::*real*)) ∧ *interval_lowerbound* (*closed_interval* [(*x*, *vector_add x* (*lambda* (λ*i*::*nat*. *1*::*real*)))]) = *x*

thm Vol1.measure_cube:

∀ *x*::(*real*, *3*) *cart*. *HOL_Light_Import*.*measure* (*cube x*) = (*1*::*real*)

thm Vol1.has_measure_cube:

∀ (*x*::(*real*, *3*) *cart*) (*r*::*real*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *IN s* (*set_of_cube* (*int_ball x r*)) ⟶ *has_measure s* (*1*::*real*)

thm Vol1.negligible_cube:

∀ (*x*::(*real*, *3*) *cart*) (*r*::*real*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) *t*::(*real*, *3*) *cart* ⇒ *bool*. *IN s* (*set_of_cube* (*int_ball x r*)) ∧ *IN t* (*set_of_cube* (*int_ball x r*)) ∧ *s* ≠ *t* ⟶ *negligible* (*HOL_Light_Import*.*INTER s t*)

thm Vol1.SUM_CONST2:

∀ (*c*::*real*) *s*::*?'a*::*type* ⇒ *bool*. *FINITE s* ⟶ *sum s* (λ*n*::*?'a*::*type*. *c*) = *real_of_nat* (*CARD s*) ∗ *c*

thm Vol1.measure_unions_of_cube:

∀ (*x*::(*real*, *3*) *cart*) *r*::*real*. *HOL_Light_Import*.*measure* (*UNIONS* (*set_of_cube* (*int_ball x r*))) = *real_of_nat* (*CARD* (*set_of_cube* (*int_ball x r*)))

thm Vol1.pow_lesthan_1:

∀ *a*::*real*. (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ *a* ∗ *a* < (*1*::*real*)

thm Vol1.cube_to_ball:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *IN y* (*cube x*) ⟶ *vector_norm* (*vector_sub x y*) < *sqrt* (*real_of_nat* (*3*::*nat*))

thm Vol1.unions_cube_subset_ball:

∀ (*x*::(*real*, *3*) *cart*) *r*::*real*. *SUBSET* (*UNIONS* (*set_of_cube* (*int_ball x r*))) (*hinhcau x* (*r* + *sqrt* (*real_of_nat* (*3*::*nat*))))

thm Vol1.hinhcau_ball:

∀ (*x*::(*real*, *3*) *cart*) *r*::*real*. *hinhcau x r* = *ball* (*x*, *r*)

thm Vol1.measure_unions_cube_less_ball:

∀ (*x*::(*real*, *3*) *cart*) *r*::*real*. *HOL_Light_Import*.*measure* (*UNIONS* (*set_of_cube* (*int_ball x r*))) ≤ *HOL_Light_Import*.*measure* (*hinhcau x* (*r* + *sqrt* (*real_of_nat* (*3*::*nat*))))

thm Vol1.measure_hinhcau:

$\forall (x::(real, 3)\ cart)\ r::real.\ (0::real) \leq r \longrightarrow HOL\_Light\_Import.measure\ (hinhcau\ x\ r) = real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * r^{3::nat})$

thm Vol1.WQZISRI:

$\forall (p::(real, 3)\ cart)\ r::real.\ (0::real) \leq r \longrightarrow FINITE\ (int\_ball\ p\ r) \wedge real\_of\_nat\ (CARD\ (int\_ball\ p\ r)) \leq real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat})$

thm DEF_ccube:

$ccube = (\lambda\_2397137::(real, 3)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%223::(real, 3)\ cart.\ \exists y::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%223\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ \_2397137\ i \leq \$\ y\ i \wedge \$\ y\ i \leq \$\ \_2397137\ i + (1::real))\ y))$

thm Vol1.ccube:

$\forall x::(real, 3)\ cart.\ ccube\ x = GSPEC\ (\lambda GEN\%PVAR\%223::(real, 3)\ cart.\ \exists y::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%223\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow \$\ x\ i \leq \$\ y\ i \wedge \$\ y\ i \leq \$\ x\ i + (1::real))\ y)$

thm DEF_set_of_ccube:

$set\_of\_ccube = (\lambda\_2397142::(real, 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%224::(real, 3)\ cart \Rightarrow bool.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%224\ (IN\ x\ \_2397142)\ (ccube\ x)))$

thm Vol1.set_of_ccube:

$\forall S::(real, 3)\ cart \Rightarrow bool.\ set\_of\_ccube\ S = GSPEC\ (\lambda GEN\%PVAR\%224::(real, 3)\ cart \Rightarrow bool.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%224\ (IN\ x\ S)\ (ccube\ x))$

thm DEF_close_hinhcau:

$close\_hinhcau = (\lambda(\_2397147::(real, 3)\ cart)\ \_2397148::real.\ GSPEC\ (\lambda GEN\%PVAR\%225::(real, 3)\ cart.\ \exists y::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%225\ (vector\_norm\ (vector\_sub\ y\ \_2397147) \leq \_2397148)\ y))$

thm Vol1.close_hinhcau:

$\forall (x::(real, 3)\ cart)\ r::real.\ close\_hinhcau\ x\ r = GSPEC\ (\lambda GEN\%PVAR\%225::(real, 3)\ cart.\ \exists y::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%225\ (vector\_norm\ (vector\_sub\ y\ x) \leq r)\ y)$

thm DEF_map1:

$map1 = ccube$

thm Vol1.map1:

$\forall x::(real, 3)\ cart.\ map1\ x = ccube\ x$

thm Vol1.daumut_ccube:

$\forall x::(real, 3)\ cart.\ IN\ x\ (ccube\ x) \wedge IN\ (vector\_add\ x\ (lambda\ (\lambda i::nat.\ 1::real)))\ (ccube\ x)$

thm Vol1.different_ccube:

$\forall (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ ccube\ x = ccube\ y \longrightarrow x = y$

thm Vol1.inj_map1:

$\forall (x::(real, 3)\ cart)\ r::real.\ INJ\ map1\ (int\_ball\ x\ r)\ (set\_of\_ccube\ (int\_ball\ x\ r))$

thm Vol1.surj2_map1:

$\forall (x::(real, 3)\ cart)\ r::real.\ IMAGE\ map1\ (int\_ball\ x\ r) = set\_of\_ccube\ (int\_ball\ x\ r)$

thm Vol1.int_ball_card_eq_ccube:

$\forall (x::(real, 3)\ cart)\ r::real.\ CARD\ (int\_ball\ x\ r) = CARD\ (set\_of\_ccube\ (int\_ball\ x\ r))$

thm Vol1.ccube_eq_interval:

$\forall x::(real, 3)\ cart.\ ccube\ x = closed\_interval\ [(x, vector\_add\ x\ (lambda\ (\lambda i::nat.\ 1::real)))]$

thm Vol1.measurable_ccube:

$\forall x::(real, 3)\ cart.\ measurable\ (ccube\ x)$

thm Vol1.finite_set_of_ccube:

$\forall (x::(real, 3)\ cart)\ r::real.\ FINITE\ (set\_of\_ccube\ (int\_ball\ x\ r))$

thm Vol1.measurable_unions_ccube:

$\forall (x::(real, 3)\ cart)\ r::real.\ measurable\ (UNIONS\ (set\_of\_ccube\ (int\_ball\ x\ r)))$

thm Vol1.measure_ccube:

$\forall x::(real, 3)\ cart.\ HOL\_Light\_Import.measure\ (ccube\ x) = (1::real)$

thm Vol1.has_measure_ccube:

$\forall (x::(real, 3)\ cart)\ (r::real)\ s::(real, 3)\ cart \Rightarrow bool.\ IN\ s\ (set\_of\_ccube\ (int\_ball\ x\ r)) \longrightarrow has\_measure\ s\ (1::real)$

thm Vol1.measure_element_set_ccube:

$\forall (S::(real, 3)\ cart \Rightarrow bool)\ t::(real, 3)\ cart \Rightarrow bool.\ IN\ t\ (set\_of\_ccube\ S) \longrightarrow HOL\_Light\_Import.measure\ t = (1::real)$

thm Vol1.sum_measure_ccube:

$\forall (p::(real, 3)\ cart)\ r::real.\ sum\ (set\_of\_ccube\ (int\_ball\ p\ r))\ HOL\_Light\_Import.measure = real\_of\_nat\ (CARD\ (set\_of\_ccube\ (int\_ball\ p\ r)))$

thm Vol1.mea_unions_ccube_le_card:

$\forall (x::(real, 3)\ cart)\ r::real.\ HOL\_Light\_Import.measure\ (UNIONS\ (set\_of\_ccube\ (int\_ball\ x\ r))) \leq real\_of\_nat\ (CARD\ (set\_of\_ccube\ (int\_ball\ x\ r)))$

thm Vol1.in_ccube_floor:

$\forall x::(real, 3)\ cart.\ IN\ x\ (ccube\ (lambda\ (\lambda i::nat.\ HOL\_Light\_Import.floor\ (\$\ x\ i))))$

thm Vol1.pow_lesthan_eq_1:

$\forall a::real.\ (0::real) \leq a \wedge a \leq (1::real) \longrightarrow a * a \leq (1::real)$

thm Vol1.ccube_to_ball:

$\forall (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ IN\ y\ (ccube\ x) \longrightarrow vector\_norm\ (vector\_sub\ x\ y) \leq sqrt\ (real\_of\_nat\ (3::nat))$

thm Vol1.ball_subset_union_ccube:

$\forall (x::(real, 3)\ cart)\ r::real.\ SUBSET\ (ball\ (x, r - sqrt\ (real\_of\_nat\ (3::nat))))\ (UNIONS\ (set\_of\_ccube\ (int\_ball\ x\ r)))$

thm Vol1.volume_ball_gon:

$\forall (x::(real, 3)\ cart)\ r::real.\ sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow measurable\ (ball\ (x, r - sqrt\ (real\_of\_nat\ (3::nat)))) \wedge HOL\_Light\_Import.measure\ (ball\ (x, r - sqrt\ (real\_of\_nat\ (3::nat)))) = real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r - sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat})$

thm Vol1.lower_bound_measure_unions_ccube:

$\forall (x::(real, 3)\ cart)\ r::real.\ sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r - sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) \leq HOL\_Light\_Import.measure\ (UNIONS\ (set\_of\_ccube\ (int\_ball\ x\ r)))$

thm Vol1.PWVIIOL:

$\forall (x::(real, 3)\ cart)\ r::real.\ sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow FINITE\ (int\_ball\ x\ r) \wedge real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r - sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) \leq real\_of\_nat\ (CARD\ (int\_ball\ x\ r))$

thm DEF_integer_point:

$integer\_point = (\lambda\_2397937::(real, 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%228::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%228\ ((\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow integer\ (\$\ x\ i)) \wedge IN\ x\ \_2397937)\ x))$

thm Vol1.integer_point:

$\forall S::(real, 3)\ cart \Rightarrow bool.\ integer\_point\ S = GSPEC\ (\lambda GEN\%PVAR\%228::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%228\ ((\forall i::nat.\ (1::nat) \leq i \wedge i \leq (3::nat) \longrightarrow integer\ (\$\ x\ i)) \wedge IN\ x\ S)\ x)$

thm Vol1.int_ball_subset:

$\forall (x::(real,\ 3)\ cart)\ (r1::real)\ r2::real.\ r1 < r2 \longrightarrow SUBSET\ (int\_ball\ x\ r1)\ (int\_ball\ x\ r2)$

thm Vol1.card_int_ball_ineq:

$\forall (x::(real,\ 3)\ cart)\ (r1::real)\ r2::real.\ r1 < r2 \longrightarrow CARD\ (int\_ball\ x\ r1) \leq CARD\ (int\_ball\ x\ r2)$

thm Vol1.eq_def_intball:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \land (0::real) < k2 \land k2 \leq r \longrightarrow integer\_point\ (DIFF\ (ball\ (x,\ r + k1))\ (ball\ (x,\ r - k2))) = DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2))$

thm Vol1.card_int_ball_le:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \land (0::real) < k2 \land k2 + sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow real\_of\_nat\ (CARD\ (int\_ball\ x\ (r + k1))) \leq real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r + k1 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat})$

thm Vol1.card_int_ball_le2:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ r::real.\ (0::real) \leq k1 \land (0::real) \leq r \longrightarrow real\_of\_nat\ (CARD\ (int\_ball\ x\ (r + k1))) \leq real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r + k1 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat})$

thm Vol1.card_int_ball_pos:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \land (0::real) < k2 \land k2 + sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r - k2 - sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) \leq real\_of\_nat\ (CARD\ (int\_ball\ x\ (r - k2)))$

thm Vol1.bdt1_finiteness:

$\forall (r::real)\ k::real.\ (0::real) < r \longrightarrow real\_of\_nat\ (4::nat) * (pi * (r * k)) \leq real\_of\_nat\ (4::nat) * (pi * (r * |k|))$

thm Vol1.bdt2_finiteness:

$\forall (r::real)\ (k::real)\ i::nat.\ (0::real) < k \land k \leq r \longrightarrow (1::real) \leq r^{i}\ /\ k^{i}$

thm Vol1.bdt3_finiteness:

$\forall (r::real)\ (k1::real)\ (k2::real)\ k::real.\ (0::real) < r \land (0::real) < k1 \land (0::real) < k2 \longrightarrow (0::real) \leq real\_of\_nat\ (4::nat) * (pi * (r * |k|)) \land (0::real) \leq real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * ((k1 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat} + (k2 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}))$

thm Vol1.bdt4_finiteness:

$\forall (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \land (0::real) < k2 \land k2 + sqrt\ (real\_of\_nat\ (3::nat)) \leq r \longrightarrow real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * (r + k1 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) - real\_of\_nat\ (4::nat)\ /$

$real\_of\_nat\ (3::nat) * (pi * (r - k2 - sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) \leq real\_of\_nat\ (4::nat) / real\_of\_nat\ (3::nat) * (pi * (real\_of\_nat\ (3::nat) * (k1 + (k2 + real\_of\_nat\ (2::nat) * sqrt\ (real\_of\_nat\ (3::nat)))) + (((k1 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat} + (k2 + sqrt\ (real\_of\_nat\ (3::nat)))^{3::nat}) / (k2 + sqrt\ (real\_of\_nat\ (3::nat)))^2 + real\_of\_nat\ (3::nat) * (|(k1 + sqrt\ (real\_of\_nat\ (3::nat)))^2 - (k2 + sqrt\ (real\_of\_nat\ (3::nat)))^2| / (k2 + sqrt\ (real\_of\_nat\ (3::nat))))))) * r^2$

thm Vol1.card_diff_ineq:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2))) \leq CARD\ (int\_ball\ x\ (r + k1))$

thm Vol1.bdt6_finiteness:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \wedge (0::real) < r \longrightarrow real\_of\_nat\ (CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2)))) \leq real\_of\_nat\ (4::nat) / real\_of\_nat\ (3::nat) * (pi * (r + (k1 + sqrt\ (real\_of\_nat\ (3::nat))))^{3::nat})$

thm Vol1.bdt5_finiteness:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ k2::real.\ (0::real) < k1 \wedge (0::real) < k2 \longrightarrow (\exists C::real.\ \forall r \geq k2 + sqrt\ (real\_of\_nat\ (3::nat)).\ real\_of\_nat\ (CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2)))) \leq C * r^2)$

thm Vol1.bdt7_finiteness:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \wedge (0::real) < k2 \longrightarrow (\exists C::real.\ \forall r::real.\ r < k2 + sqrt\ (real\_of\_nat\ (3::nat)) \wedge k2 \leq r \longrightarrow real\_of\_nat\ (CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2)))) \leq C * r^2)$

thm Vol1.TXIWYHI:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ (k2::real)\ r::real.\ (0::real) < k1 \wedge (0::real) < k2 \longrightarrow (\exists C::real.\ \forall r \geq k2.\ real\_of\_nat\ (CARD\ (integer\_point\ (DIFF\ (ball\ (x,\ r + k1))\ (ball\ (x,\ r - k2)))))) \leq C * r^2)$

thm DEF_POWER:

$POWER = (SOME\ POWER::nat \Rightarrow (?'a::type \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall \_2398319::nat.\ (\forall f::?'a::type \Rightarrow ?'a::type.\ POWER\ \_2398319\ f\ (0::nat) = id) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ POWER\ \_2398319\ f\ (Suc\ n) = POWER\ \_2398319\ f\ n \circ f))\ (96::nat)$

thm Fan.POWER:

$(\forall f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ (0::nat) = id) \wedge (\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ POWER\ f\ (Suc\ n) = POWER\ f\ n \circ f)$

thm Fan.POWER_conjunct1:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ n::nat.\ POWER\ f\ (Suc\ n) = POWER\ f\ n \circ f$

thm Fan.POWER_0:

$\forall f{::}?'a{::}type \Rightarrow ?'a{::}type.\ POWER\ f\ (0{::}nat) = id$

thm Fan.POWER_1:

$\forall f{::}?'a{::}type \Rightarrow ?'a{::}type.\ POWER\ f\ (1{::}nat) = f$

thm Fan.POWER_2:

$\forall f{::}?'a{::}type \Rightarrow ?'a{::}type.\ POWER\ f\ (2{::}nat) = f \circ f$

thm DEF_orbit_map:

$orbit\_map = (\lambda(\_2398320{::}?'a{::}type \Rightarrow ?'a{::}type)\ \_2398321{::}?'a{::}type.\ GSPEC\ (\lambda GEN\%PVAR\%229{::}?'a{::}type.\ \exists n{::}nat.\ SETSPEC\ GEN\%PVAR\%229\ ((0{::}nat) \leq n)\ (POWER\ \_2398320\ n\ \_2398321)))$

thm Hypermap.orbit_map:

$\forall (f{::}?'a{::}type \Rightarrow ?'a{::}type)\ x{::}?'a{::}type.\ orbit\_map\ f\ x = GSPEC\ (\lambda GEN\%PVAR\%229{::}?'a{::}type.\ \exists n{::}nat.\ SETSPEC\ GEN\%PVAR\%229\ ((0{::}nat) \leq n)\ (POWER\ f\ n\ x))$

thm Hypermap.lemma_two_series_eq:

$\forall (p{::}nat \Rightarrow ?'a{::}type)\ (q{::}nat \Rightarrow ?'a{::}type)\ n{::}nat.\ (\forall i{\leq}n.\ p\ i = q\ i) \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%230{::}?'a{::}type.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%230\ (i \leq n)\ (p\ i)) = GSPEC\ (\lambda GEN\%PVAR\%231{::}?'a{::}type.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%231\ (i \leq n)\ (q\ i))$

thm Hypermap.lemma_add_one_assumption:

$\forall (P{::}nat \Rightarrow bool)\ n{::}nat.\ (\forall i{\leq}Suc\ n.\ P\ i) = ((\forall i{\leq}n.\ P\ i) \wedge P\ (Suc\ n))$

thm Hypermap.lemma_sub_part:

$\forall (P{::}nat \Rightarrow bool)\ (n{::}nat)\ m{::}nat.\ (\forall i{\leq}n.\ P\ i) \wedge m \leq n \longrightarrow (\forall i{\leq}m.\ P\ i)$

thm Hypermap.exist_hypermap:

$\exists H{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type).\ FINITE\ (fst\ H) \wedge permutes\ (fst\ (snd\ H))\ (fst\ H) \wedge permutes\ (fst\ (snd\ (snd\ H)))\ (fst\ H) \wedge permutes\ (snd\ (snd\ (snd\ H)))\ (fst\ H) \wedge fst\ (snd\ H) \circ (fst\ (snd\ (snd\ H)) \circ snd\ (snd\ (snd\ H))) = id$

thm TYDEF_hypermap:

$hypermap\ (tuple\_hypermap\ (?a{::}?'a{::}type\ hypermap)) = ?a \wedge (FINITE\ (fst\ (?r{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type))) \wedge permutes\ (fst\ (snd\ ?r))\ (fst\ ?r) \wedge permutes\ (fst\ (snd\ (snd\ ?r)))\ (fst\ ?r) \wedge permutes\ (snd\ (snd\ (snd\ ?r)))\ (fst\ ?r) \wedge fst\ (snd\ ?r) \circ (fst\ (snd\ (snd\ ?r)) \circ snd\ (snd\ (snd\ ?r))) = id) = (tuple\_hypermap\ (hypermap\ ?r) = ?r)$

thm Hypermap.hypermap_tybij_conjunct1:

$\forall r::(?'a::type \Rightarrow bool) \times (?'a::type \Rightarrow ?'a::type) \times (?'a::type \Rightarrow ?'a::type) \times (?'a::type \Rightarrow ?'a::type).\ (FINITE\ (fst\ r) \wedge permutes\ (fst\ (snd\ r))\ (fst\ r) \wedge permutes\ (fst\ (snd\ (snd\ r)))\ (fst\ r) \wedge permutes\ (snd\ (snd\ (snd\ r)))\ (fst\ r) \wedge fst\ (snd\ r) \circ (fst\ (snd\ (snd\ r)) \circ snd\ (snd\ (snd\ r))) = id) = (tuple\_hypermap\ (hypermap\ r) = r)$

thm Hypermap.hypermap_tybij_conjunct0:

$\forall a::?'a::type\ hypermap.\ hypermap\ (tuple\_hypermap\ a) = a$

thm Hypermap.hypermap_tybij:

$(\forall a::?'a::type\ hypermap.\ hypermap\ (tuple\_hypermap\ a) = a) \wedge (\forall r::(?'a::type \Rightarrow bool) \times (?'a::type \Rightarrow ?'a::type) \times (?'a::type \Rightarrow ?'a::type) \times (?'a::type \Rightarrow ?'a::type).\ (FINITE\ (fst\ r) \wedge permutes\ (fst\ (snd\ r))\ (fst\ r) \wedge permutes\ (fst\ (snd\ (snd\ r)))\ (fst\ r) \wedge permutes\ (snd\ (snd\ (snd\ r)))\ (fst\ r) \wedge fst\ (snd\ r) \circ (fst\ (snd\ (snd\ r)) \circ snd\ (snd\ (snd\ r))) = id) = (tuple\_hypermap\ (hypermap\ r) = r))$

thm DEF_dart:

$dart = (\lambda\_2398378::?'a::type\ hypermap.\ fst\ (tuple\_hypermap\ \_2398378))$

thm Hypermap.dart:

$\forall H::?'a::type\ hypermap.\ dart\ H = fst\ (tuple\_hypermap\ H)$

thm DEF_edge_map:

$edge\_map = (\lambda\_2398383::?'a::type\ hypermap.\ fst\ (snd\ (tuple\_hypermap\ \_2398383)))$

thm Hypermap.edge_map:

$\forall H::?'a::type\ hypermap.\ edge\_map\ H = fst\ (snd\ (tuple\_hypermap\ H))$

thm DEF_node_map:

$node\_map = (\lambda\_2398388::?'a::type\ hypermap.\ fst\ (snd\ (snd\ (tuple\_hypermap\ \_2398388))))$

thm Hypermap.node_map:

$\forall H::?'a::type\ hypermap.\ node\_map\ H = fst\ (snd\ (snd\ (tuple\_hypermap\ H)))$

thm DEF_face_map:

$face\_map = (\lambda\_2398393::?'a::type\ hypermap.\ snd\ (snd\ (snd\ (tuple\_hypermap\ \_2398393))))$

thm Hypermap.face_map:

$\forall H::?'a::type\ hypermap.\ face\_map\ H = snd\ (snd\ (snd\ (tuple\_hypermap\ H)))$

thm Hypermap.hypermap_lemma:

$\forall H::?'a::type\ hypermap.\ FINITE\ (dart\ H) \wedge permutes\ (edge\_map\ H)\ (dart\ H) \wedge permutes\ (node\_map\ H)\ (dart\ H) \wedge permutes\ (face\_map\ H)\ (dart\ H) \wedge edge\_map\ H \circ (node\_map\ H \circ face\_map\ H) = id$

thm Hypermap.edge_map_and_darts:

$\forall H::?'a::type\ hypermap.\ FINITE\ (dart\ H) \wedge permutes\ (edge\_map\ H)\ (dart\ H)$

thm Hypermap.node_map_and_darts:

$\forall H::?'a::type\ hypermap.\ FINITE\ (dart\ H) \wedge permutes\ (node\_map\ H)\ (dart\ H)$

thm Hypermap.face_map_and_darts:

$\forall H::?'a::type\ hypermap.\ FINITE\ (dart\ H) \wedge permutes\ (face\_map\ H)\ (dart\ H)$

thm DEF_edge:

$edge = (\lambda\_2398398::?'a::type\ hypermap.\ orbit\_map\ (edge\_map\ \_2398398))$

thm Hypermap.edge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ edge\ H\ x = orbit\_map\ (edge\_map\ H)\ x$

thm DEF_node:

$node = (\lambda\_2398410::?'a::type\ hypermap.\ orbit\_map\ (node\_map\ \_2398410))$

thm Hypermap.node:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ node\ H\ x = orbit\_map\ (node\_map\ H)\ x$

thm DEF_face:

$face = (\lambda\_2398422::?'a::type\ hypermap.\ orbit\_map\ (face\_map\ \_2398422))$

thm Hypermap.face:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ face\ H\ x = orbit\_map\ (face\_map\ H)\ x$

thm DEF_go_one_step:

$go\_one\_step = (\lambda(\_2398434::?'a::type\ hypermap)\ (\_2398435::?'a::type)\ \_2398436::?'a::type.\ \_2398436 = edge\_map\ \_2398434\ \_2398435 \vee \_2398436 = node\_map\ \_2398434\ \_2398435 \vee \_2398436 = face\_map\ \_2398434\ \_2398435)$

thm Hypermap.go_one_step:

$\forall (y::?'a::type)\ (H::?'a::type\ hypermap)\ x::?'a::type.\ go\_one\_step\ H\ x\ y = (y = edge\_map\ H\ x \vee y = node\_map\ H\ x \vee y = face\_map\ H\ x)$

thm DEF_is_path:

$is\_path = (SOME\ is\_path::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow bool.\ \forall \_2398462::nat.\ (\forall (H::?'a::type\ hypermap)\ p::nat \Rightarrow ?'a::type.\ is\_path\ \_2398462\ H\ p\ (0::nat) = True) \wedge (\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_path\ \_2398462\ H\ p\ (Suc\ n) = (is\_path\ \_2398462\ H\ p\ n \wedge go\_one\_step\ H\ (p\ n)\ (p\ (Suc\ n))))))\ (97::nat)$

thm Hypermap.is_path_conjunct0:

is_path (?H::?'a::type hypermap) (?p::nat ⇒ ?'a::type) (0::nat) = True

thm Hypermap.is_path_conjunct1:

is_path (?H::?'a::type hypermap) (?p::nat ⇒ ?'a::type) (Suc (?n::nat)) = (is_path ?H ?p ?n ∧ go_one_step ?H (?p ?n) (?p (Suc ?n)))

thm Hypermap.is_path:

is_path (?H::?'a::type hypermap) (?p::nat ⇒ ?'a::type) (0::nat) = True ∧ is_path ?H ?p (Suc (?n::nat)) = (is_path ?H ?p ?n ∧ go_one_step ?H (?p ?n) (?p (Suc ?n)))

thm DEF_is_in_component:

is_in_component = (λ(_2398463::?'a::type hypermap) (_2398464::?'a::type) _2398465::?'a::type. ∃(p::nat ⇒ ?'a::type) n::nat. p (0::nat) = _2398464 ∧ p n = _2398465 ∧ is_path _2398463 p n)

thm Hypermap.is_in_component:

∀(x::?'a::type) (y::?'a::type) H::?'a::type hypermap. is_in_component H x y = (∃(p::nat ⇒ ?'a::type) n::nat. p (0::nat) = x ∧ p n = y ∧ is_path H p n)

thm DEF_comb_component:

comb_component = (λ(_2398484::?'a::type hypermap) _2398485::?'a::type. GSPEC (λGEN%PVAR%232::?'a::type. ∃y::?'a::type. SETSPEC GEN%PVAR%232 (is_in_component _2398484 _2398485 y) y))

thm Hypermap.comb_component:

∀(H::?'a::type hypermap) x::?'a::type. comb_component H x = GSPEC (λGEN%PVAR%232::?'a::type. ∃y::?'a::type. SETSPEC GEN%PVAR%232 (is_in_component H x y) y)

thm DEF_set_of_orbits:

set_of_orbits = (λ(_2398496::?'a::type ⇒ bool) _2398497::?'a::type ⇒ ?'a::type. GSPEC (λGEN%PVAR%233::?'a::type ⇒ bool. ∃x::?'a::type. SETSPEC GEN%PVAR%233 (IN x _2398496) (orbit_map _2398497 x)))

thm Hypermap.set_of_orbits:

∀(D::?'a::type ⇒ bool) f::?'a::type ⇒ ?'a::type. set_of_orbits D f = GSPEC (λGEN%PVAR%233::?'a::type ⇒ bool. ∃x::?'a::type. SETSPEC GEN%PVAR%233 (IN x D) (orbit_map f x))

thm DEF_number_of_orbits:

number_of_orbits = (λ(_2398508::?'a::type ⇒ bool) _2398509::?'a::type ⇒ ?'a::type. CARD (set_of_orbits _2398508 _2398509))

thm Hypermap.number_of_orbits:

∀ (D::?'a::type ⇒ bool) f::?'a::type ⇒ ?'a::type. number_of_orbits D f = CARD (set_of_orbits D f)

thm DEF_edge_set:

edge_set = (λ_2398520::?'a::type hypermap. set_of_orbits (dart _2398520) (edge_map _2398520))

thm Hypermap.edge_set:

∀ H::?'a::type hypermap. edge_set H = set_of_orbits (dart H) (edge_map H)

thm DEF_node_set:

node_set = (λ_2398525::?'a::type hypermap. set_of_orbits (dart _2398525) (node_map _2398525))

thm Hypermap.node_set:

∀ H::?'a::type hypermap. node_set H = set_of_orbits (dart H) (node_map H)

thm DEF_face_set:

face_set = (λ_2398530::?'a::type hypermap. set_of_orbits (dart _2398530) (face_map _2398530))

thm Hypermap.face_set:

∀ H::?'a::type hypermap. face_set H = set_of_orbits (dart H) (face_map H)

thm DEF_set_components:

set_components = (λ(_2398535::?'a::type hypermap) _2398536::?'a::type ⇒ bool. GSPEC (λGEN%PVAR%234::?'a::type ⇒ bool. ∃ x::?'a::type. SETSPEC GEN%PVAR%234 (IN x _2398536) (comb_component _2398535 x)))

thm Hypermap.set_components:

∀ (D::?'a::type ⇒ bool) H::?'a::type hypermap. set_components H D = GSPEC (λGEN%PVAR%234::?'a::type ⇒ bool. ∃ x::?'a::type. SETSPEC GEN%PVAR%234 (IN x D) (comb_component H x))

thm DEF_set_part_components:

set_part_components = (λ(_2398547::?'a::type hypermap) _2398548::?'a::type ⇒ bool. GSPEC (λGEN%PVAR%235::?'a::type ⇒ bool. ∃ x::?'a::type. SETSPEC GEN%PVAR%235 (IN x _2398548) (comb_component _2398547 x)))

thm Hypermap.set_part_components:

∀ (D::?'a::type ⇒ bool) H::?'a::type hypermap. set_part_components H D = GSPEC (λGEN%PVAR%235::?'a::type ⇒ bool. ∃ x::?'a::type. SETSPEC GEN%PVAR%235 (IN x D) (comb_component H x))

thm DEF_set_of_components:

set_of_components = (λ_2398559::?'a::type hypermap. set_part_components _2398559 (dart _2398559))

thm Hypermap.set_of_components:

$\forall H::?'a::type\ hypermap.\ set\_of\_components\ H = set\_part\_components\ H\ (dart\ H)$

thm DEF_number_of_edges:

$number\_of\_edges = (\lambda\_2398564::?'a::type\ hypermap.\ CARD\ (edge\_set\ \_2398564))$

thm Hypermap.number_of_edges:

$\forall H::?'a::type\ hypermap.\ number\_of\_edges\ H = CARD\ (edge\_set\ H)$

thm DEF_number_of_nodes:

$number\_of\_nodes = (\lambda\_2398569::?'a::type\ hypermap.\ CARD\ (node\_set\ \_2398569))$

thm Hypermap.number_of_nodes:

$\forall H::?'a::type\ hypermap.\ number\_of\_nodes\ H = CARD\ (node\_set\ H)$

thm DEF_number_of_faces:

$number\_of\_faces = (\lambda\_2398574::?'a::type\ hypermap.\ CARD\ (face\_set\ \_2398574))$

thm Hypermap.number_of_faces:

$\forall H::?'a::type\ hypermap.\ number\_of\_faces\ H = CARD\ (face\_set\ H)$

thm DEF_number_of_components:

$number\_of\_components = (\lambda\_2398579::?'a::type\ hypermap.\ CARD\ (set\_of\_components\ \_2398579))$

thm Hypermap.number_of_components:

$\forall H::?'a::type\ hypermap.\ number\_of\_components\ H = CARD\ (set\_of\_components\ H)$

thm DEF_plain_hypermap:

$plain\_hypermap = (\lambda\_2398584::?'a::type\ hypermap.\ edge\_map\ \_2398584 \circ edge\_map\ \_2398584 = id)$

thm Hypermap.plain_hypermap:

$\forall H::?'a::type\ hypermap.\ plain\_hypermap\ H = (edge\_map\ H \circ edge\_map\ H = id)$

thm DEF_planar_hypermap:

$planar\_hypermap = (\lambda\_2398589::?'a::type\ hypermap.\ number\_of\_nodes\ \_2398589 + (number\_of\_edges\ \_2398589 + number\_of\_faces\ \_2398589) = CARD\ (dart\ \_2398589) + (2::nat) * number\_of\_components\ \_2398589)$

thm Hypermap.planar_hypermap:

$\forall H::?'a::type\ hypermap.\ planar\_hypermap\ H = (number\_of\_nodes\ H + (number\_of\_edges\ H + number\_of\_faces\ H) = CARD\ (dart\ H) + (2::nat) * number\_of\_components\ H)$

thm DEF_simple_hypermap:

$simple\_hypermap = (\lambda\_2398594::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_2398594) \longrightarrow HOL\_Light\_Import.INTER\ (node\ \_2398594\ x)\ (face\ \_2398594\ x) = INSERT\ x\ EMPTY)$

thm Hypermap.simple_hypermap:

$\forall H::?'a::type\ hypermap.\ simple\_hypermap\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow HOL\_Light\_Import.INTER\ (node\ H\ x)\ (face\ H\ x) = INSERT\ x\ EMPTY)$

thm DEF_dart_degenerate:

$dart\_degenerate = (\lambda(\_2398599::?'a::type\ hypermap)\ \_2398600::?'a::type.\ edge\_map\ \_2398599\ \_2398600 = \_2398600 \lor node\_map\ \_2398599\ \_2398600 = \_2398600 \lor face\_map\ \_2398599\ \_2398600 = \_2398600)$

thm Hypermap.dart_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\_degenerate\ H\ x = (edge\_map\ H\ x = x \lor node\_map\ H\ x = x \lor face\_map\ H\ x = x)$

thm DEF_dart_nondegenerate:

$dart\_nondegenerate = (\lambda(\_2398611::?'a::type\ hypermap)\ \_2398612::?'a::type.\ edge\_map\ \_2398611\ \_2398612 \neq \_2398612 \land node\_map\ \_2398611\ \_2398612 \neq \_2398612 \land face\_map\ \_2398611\ \_2398612 \neq \_2398612)$

thm Hypermap.dart_nondegenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\_nondegenerate\ H\ x = (edge\_map\ H\ x \neq x \land node\_map\ H\ x \neq x \land face\_map\ H\ x \neq x)$

thm DEF_is_edge_nondegenerate:

$is\_edge\_nondegenerate = (\lambda\_2398623::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_2398623) \longrightarrow edge\_map\ \_2398623\ x \neq x)$

thm Hypermap.is_edge_nondegenerate:

$\forall H::?'a::type\ hypermap.\ is\_edge\_nondegenerate\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow edge\_map\ H\ x \neq x)$

thm DEF_is_node_nondegenerate:

$is\_node\_nondegenerate = (\lambda\_2398628::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_2398628) \longrightarrow node\_map\ \_2398628\ x \neq x)$

thm Tame_defs.is_node_nondegenerate:

$\forall H::?'a::type\ hypermap.\ is\_node\_nondegenerate\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow node\_map\ H\ x \neq x)$

thm DEF_is_face_nondegenerate:

$is\_face\_nondegenerate = (\lambda\_2398633::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_2398633) \longrightarrow face\_map\ \_2398633\ x \neq x)$

thm Hypermap.is_face_nondegenerate:

$\forall H::?'a::type\ hypermap.\ is\_face\_nondegenerate\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow face\_map\ H\ x \neq x)$

thm Hypermap.LEFT_MULT_MAP:

$\forall (u::?'a::type \Rightarrow ?'a::type)\ (v::?'a::type \Rightarrow ?'a::type)\ w::?'a::type \Rightarrow ?'a::type.\ v = w \longrightarrow u \circ v = u \circ w$

thm Hypermap.RIGHT_MULT_MAP:

$\forall (u::?'a::type \Rightarrow ?'a::type)\ (v::?'a::type \Rightarrow ?'a::type)\ w::?'a::type \Rightarrow ?'a::type.\ u = v \longrightarrow u \circ w = v \circ w$

thm Hypermap.LEFT_INVERSE_EQUATION:

$\forall (s::?'a::type \Rightarrow bool)\ (u::?'a::type \Rightarrow ?'a::type)\ (v::?'a::type \Rightarrow ?'a::type)\ w::?'a::type \Rightarrow ?'a::type.\ permutes\ u\ s \wedge u \circ v = w \longrightarrow v = HOL\_Light\_Import.inverse\ u \circ w$

thm Hypermap.RIGHT_INVERSE_EQUATION:

$\forall (s::?'a::type \Rightarrow bool)\ (u::?'a::type \Rightarrow ?'a::type)\ (v::?'a::type \Rightarrow ?'a::type)\ w::?'a::type \Rightarrow ?'a::type.\ permutes\ v\ s \wedge u \circ v = w \longrightarrow u = w \circ HOL\_Light\_Import.inverse\ v$

thm Hypermap.iterate_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow ?'a::type.\ permutes\ u\ s \longrightarrow (\forall (n::nat)\ x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (POWER\ u\ n\ x)\ s)$

thm Hypermap.orbit_subset:

$\forall (s::?'a::type \Rightarrow bool)\ u::?'a::type \Rightarrow ?'a::type.\ permutes\ u\ s \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow SUBSET\ (orbit\_map\ u\ x)\ s)$

thm Hypermap.COM_POWER:

$\forall (n::nat)\ f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ (Suc\ n) = f \circ POWER\ f\ n$

thm Hypermap.COM_POWER_FUNCTION:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ f\ (POWER\ f\ n\ x) = POWER\ f\ (Suc\ n)\ x$

thm Hypermap.POWER_FUNCTION:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ POWER\ f\ n\ (f\ x) = POWER\ f\ (Suc\ n)\ x$

thm Hypermap.addition_exponents:

$\forall (m::nat)\ (n::nat)\ f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ (m + n) = POWER\ f\ m \circ POWER\ f\ n$

thm Hypermap.multiplication_exponents:

$\forall (m::nat)\ (n::nat)\ f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ (m * n) = POWER\ (POWER\ f\ n)\ m$

thm Hypermap.power_unit_map:

$\forall (n::nat)\ f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ n = id \longrightarrow (\forall m::nat.\ POWER\ f\ (m * n) = id)$

thm Hypermap.power_map_fix_point:

$\forall (n::nat)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ POWER\ f\ n\ x = x \longrightarrow (\forall m::nat.\ POWER\ f\ (m * n)\ x = x)$

thm Hypermap.lemma_add_exponent_function:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (m::nat)\ (n::nat)\ x::?'a::type.\ POWER\ p\ (m + n)\ x = POWER\ p\ m\ (POWER\ p\ n\ x)$

thm Hypermap.iterate_map_valuation:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ p\ (POWER\ p\ n\ x) = POWER\ p\ (Suc\ n)\ x$

thm Hypermap.iterate_map_valuation2:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ POWER\ p\ n\ (p\ x) = POWER\ p\ (Suc\ n)\ x$

thm Hypermap.in_orbit_lemma:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ (x::?'a::type)\ y::?'a::type.\ y = POWER\ f\ n\ x \longrightarrow IN\ y\ (orbit\_map\ f\ x)$

thm Hypermap.lemma_in_orbit:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ IN\ (POWER\ f\ n\ x)\ (orbit\_map\ f\ x)$

thm Hypermap.orbit_one_point:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ (f\ x = x) = (orbit\_map\ f\ x = INSERT\ x\ EMPTY)$

thm Hypermap.lemma_orbit_finite:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow FINITE\ (orbit\_map\ p\ x)$

thm Hypermap.orbit_cyclic:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (m::nat)\ x::?'a::type.\ m \neq (0::nat) \wedge POWER\ f\ m\ x = x \longrightarrow orbit\_map\ f\ x = GSPEC\ (\lambda GEN\%PVAR\%236::?'a::type.\ \exists k::nat.\ SETSPEC\ GEN\%PVAR\%236\ (k < m)\ (POWER\ f\ k\ x))$

thm Hypermap.power_permutation:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ s \longrightarrow (\forall n::nat.\ permutes\ (POWER\ p\ n)\ s)$

thm Hypermap.inverse_function:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.$ $permutes\ p\ s \wedge p\ x = y \longrightarrow x = HOL\_Light\_Import.inverse\ p\ y$

thm Hypermap.lemma_4functions:

$\forall (f::?'e::type \Rightarrow ?'d::type)\ (g::?'c::type \Rightarrow ?'e::type)\ (h::?'b::type \Rightarrow ?'c::type)$ $r::?'a::type \Rightarrow ?'b::type.\ f \circ (g \circ (h \circ r)) = f \circ (g \circ h \circ r)$

thm Hypermap.lemma_power_inverse_map:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ permutes\ p\ s \longrightarrow$ $POWER\ (HOL\_Light\_Import.inverse\ p)\ n \circ POWER\ p\ n = id \wedge POWER$ $p\ n \circ POWER\ (HOL\_Light\_Import.inverse\ p)\ n = id$

thm Hypermap.lemma_power_inverse:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ permutes\ p\ s \longrightarrow$ $POWER\ (HOL\_Light\_Import.inverse\ p)\ n = HOL\_Light\_Import.inverse\ (POWER$ $p\ n) \wedge HOL\_Light\_Import.inverse\ (POWER\ (HOL\_Light\_Import.inverse\ p)\ n)$ $= POWER\ p\ n$

thm Hypermap.inverse_power_function:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ (x::?'a::type)\ y::?'a::type.$ $permutes\ p\ s \longrightarrow (y = POWER\ p\ n\ x) = (x = POWER\ (HOL\_Light\_Import.inverse$ $p)\ n\ y)$

thm Hypermap.edge_map_inverse_representation:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ (y = edge\_map\ H\ x) = (x$ $= HOL\_Light\_Import.inverse\ (edge\_map\ H)\ y)$

thm Hypermap.node_map_inverse_representation:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ (y = node\_map\ H\ x) =$ $(x = HOL\_Light\_Import.inverse\ (node\_map\ H)\ y)$

thm Hypermap.face_map_inverse_representation:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ (y = face\_map\ H\ x) = (x$ $= HOL\_Light\_Import.inverse\ (face\_map\ H)\ y)$

thm Lvducxu.HYP_MAPS_INJ_conjunct0:

$\forall (x::?'a::type)\ y::?'a::type.\ (edge\_map\ (?H::?'a::type\ hypermap)\ x = edge\_map$ $?H\ y) = (x = y)$

thm Hypermap.edge_map_injective:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ (edge\_map\ H\ x = edge\_map$ $H\ y) = (x = y)$

thm Lvducxu.HYP_MAPS_INJ_conjunct1:

$\forall (x{::}?'a{::}type)\ y{::}?'a{::}type.\ (node\_map\ (?H{::}?'a{::}type\ hypermap)\ x = node\_map\ ?H\ y) = (x = y)$

thm Hypermap.node_map_injective:

$\forall (H{::}?'a{::}type\ hypermap)\ (x{::}?'a{::}type)\ y{::}?'a{::}type.\ (node\_map\ H\ x = node\_map\ H\ y) = (x = y)$

thm Lvducxu.HYP_MAPS_INJ_conjunct2:

$\forall (x{::}?'a{::}type)\ y{::}?'a{::}type.\ (face\_map\ (?H{::}?'a{::}type\ hypermap)\ x = face\_map\ ?H\ y) = (x = y)$

thm Hypermap.face_map_injective:

$\forall (H{::}?'a{::}type\ hypermap)\ (x{::}?'a{::}type)\ y{::}?'a{::}type.\ (face\_map\ H\ x = face\_map\ H\ y) = (x = y)$

thm Hypermap.lemma_dart_invariant:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ IN\ x\ (dart\ H) \longrightarrow IN\ (edge\_map\ H\ x)\ (dart\ H) \wedge IN\ (node\_map\ H\ x)\ (dart\ H) \wedge IN\ (face\_map\ H\ x)\ (dart\ H)$

thm Hypermap.lemma_dart_invariant_power_node:

$\forall (H{::}?'a{::}type\ hypermap)\ (x{::}?'a{::}type)\ n{::}nat.\ IN\ x\ (dart\ H) \longrightarrow IN\ (POWER\ (node\_map\ H)\ n\ x)\ (dart\ H)$

thm Hypermap.lemma_dart_invariant_power_face:

$\forall (H{::}?'a{::}type\ hypermap)\ (x{::}?'a{::}type)\ n{::}nat.\ IN\ x\ (dart\ H) \longrightarrow IN\ (POWER\ (face\_map\ H)\ n\ x)\ (dart\ H)$

thm Hypermap.lemma_dart_inveriant_under_inverse_maps:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ IN\ x\ (dart\ H) \longrightarrow IN\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)\ (dart\ H) \wedge IN\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)\ (dart\ H) \wedge IN\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x)\ (dart\ H)$

thm Hypermap.IMAGE_SEG:

$\forall (n{::}nat)\ f{::}nat \Rightarrow ?'a{::}type.\ IMAGE\ f\ (GSPEC\ (\lambda GEN\%PVAR\%237{::}nat.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%237\ (i < n)\ i)) = GSPEC\ (\lambda GEN\%PVAR\%238{::}?'a{::}type.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%238\ (i < n)\ (f\ i))$

thm Hypermap.FINITE_SERIES:

$\forall (n{::}nat)\ f{::}nat \Rightarrow ?'a{::}type.\ FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%239{::}?'a{::}type.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%239\ (i < n)\ (f\ i)))$

thm Hypermap.CARD_FINITE_SERIES_LE:

$\forall (n{::}nat)\ f{::}nat \Rightarrow ?'a{::}type.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%240{::}?'a{::}type.\ \exists i{::}nat.\ SETSPEC\ GEN\%PVAR\%240\ (i < n)\ (f\ i))) \le n$

thm Hypermap.LEMMA_INJ:

$\forall\,(n::nat)\ f::nat \Rightarrow\ ?'a::type.\ (\forall\,(i::nat)\ j::nat.\ i < n \wedge j < i \longrightarrow f\ i \neq f\ j) \longrightarrow (\forall\,(i::nat)\ j::nat.\ i < n \wedge j < n \wedge f\ i = f\ j \longrightarrow i = j)$

thm Hypermap.LEMMA_INJ2:

$\forall\,(n::nat)\ f::nat \Rightarrow\ ?'a::type.\ (\forall\,(i::nat)\ j::nat.\ i \leq n \wedge j < i \longrightarrow f\ j \neq f\ i) \longrightarrow (\forall\,(i::nat)\ j::nat.\ i \leq n \wedge j \leq n \wedge f\ i = f\ j \longrightarrow i = j)$

thm Hypermap.CARD_FINITE_SERIES_EQ:

$\forall\,(n::nat)\ f::nat \Rightarrow\ ?'a::type.\ (\forall\,(i::nat)\ j::nat.\ i < n \wedge j < i \longrightarrow f\ i \neq f\ j) \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%241::?'a::type.\ \exists\, i::nat.\ SETSPEC\ GEN\%PVAR\%241\ (i < n)\ (f\ i))) = n$

thm Hypermap.LM_AUX:

$\forall\,(m::nat)\ n::nat.\ m < n \longrightarrow (\exists\, k::nat.\ k \neq (0::nat) \wedge n = m + k)$

thm Hypermap.LM1:

$\forall\,(s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow\ ?'a::type)\ (n::nat)\ m::nat.\ permutes\ p\ s \wedge POWER\ p\ (m + n) = POWER\ p\ m \longrightarrow POWER\ p\ n = id$

thm Hypermap.lemma_sub_two_numbers:

$\forall\,(m::nat)\ (n::nat)\ p::nat.\ m - n - p = m - (n + p)$

thm Hypermap.LT1_NZ:

$\forall\, n::nat.\ ((1::nat) \leq n) = ((0::nat) < n)$

thm Hypermap.LT_SUC_PRE:

$\forall\, n>0::nat.\ n = Suc\ (pred\ n)$

thm Hypermap.LE_SUC_PRE:

$\forall\, n\geq 1::nat.\ Suc\ (pred\ n) = n$

thm Hypermap.LT_PRE:

$\forall\, n>0::nat.\ n = pred\ n + (1::nat)$

thm Hypermap.SUC_PRE_2:

$\forall\, n\geq 2::nat.\ Suc\ (Suc\ (pred\ (pred\ n))) = n$

thm Hypermap.LE_MOD_SUC:

$\forall\,(n::nat)\ m::nat.\ m\ mod\ Suc\ n \leq n$

thm Hypermap.LT_RIGHT_SUC:

$\forall\,(i::nat)\ n::nat.\ i < n \longrightarrow i < Suc\ n$

thm Hypermap.LE_RIGHT_SUC:

$\forall\,(i::nat)\ n::nat.\ i \leq n \longrightarrow i \leq Suc\ n$

thm Hypermap.LT_PRE_LE:

$\forall (i::nat)\ n::nat.\ i < n \longrightarrow i \leq pred\ n$

thm Hypermap.compare_left:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + n = p \longrightarrow m \leq p$

thm Hypermap.compare_right:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + n = p \longrightarrow n \leq p$

thm Hypermap.le_compare_left:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + n \leq p \longrightarrow m \leq p$

thm Hypermap.le_compare_right:

$\forall (m::nat)\ (n::nat)\ p::nat.\ m + n \leq p \longrightarrow n \leq p$

thm Hypermap.SEGMENT_TO_ONE:

$\forall n::nat.\ (n \leq (1::nat)) = (n = (0::nat) \vee n = (1::nat))$

thm Hypermap.SEGMENT_TO_TWO:

$\forall n::nat.\ (n \leq (2::nat)) = (n = (0::nat) \vee n = (1::nat) \vee n = (2::nat))$

thm Hypermap.EXPAND_SET_TWO_ELEMENTS:

$\forall p::nat \Rightarrow ?'a::type.\ GSPEC\ (\lambda GEN\%PVAR\%242::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%242\ (i \leq (1::nat))\ (p\ i)) = INSERT\ (p\ (0::nat))\ (INSERT\ (p\ (1::nat))\ EMPTY)$

thm Hypermap.EXPAND_SET_THREE_ELEMENTS:

$\forall p::nat \Rightarrow ?'a::type.\ GSPEC\ (\lambda GEN\%PVAR\%243::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%243\ (i \leq (2::nat))\ (p\ i)) = INSERT\ (p\ (0::nat))\ (INSERT\ (p\ (1::nat))\ (INSERT\ (p\ (2::nat))\ EMPTY))$

thm Hypermap.lemma_add_one_assumption_lt:

$\forall (P::nat \Rightarrow bool)\ n::nat.\ (\forall i<Suc\ n.\ P\ i) = ((\forall i<n.\ P\ i) \wedge P\ n)$

thm DEF_is_inj_list:

$is\_inj\_list = (SOME\ is\_inj\_list::nat \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow bool.\ \forall \_2399886::nat.\ (\forall p::nat \Rightarrow ?'a::type.\ is\_inj\_list\ \_2399886\ p\ (0::nat) = True) \wedge (\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ \_2399886\ p\ (Suc\ n) = (is\_inj\_list\ \_2399886\ p\ n \wedge (\forall i\leq n.\ p\ i \neq p\ (Suc\ n)))))\ (98::nat)$

thm Hypermap.is_inj_list_conjunct0:

$is\_inj\_list\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True$

thm Hypermap.is_inj_list_conjunct1:

$is\_inj\_list\ (?p::nat \Rightarrow ?'a::type)\ (Suc\ (?n::nat)) = (is\_inj\_list\ ?p\ ?n \wedge (\forall i\leq ?n.\ ?p\ i \neq ?p\ (Suc\ ?n)))$

thm Hypermap.is_inj_list:

$is\_inj\_list\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True \wedge is\_inj\_list\ ?p\ (Suc\ (?n::nat))$
$= (is\_inj\_list\ ?p\ ?n \wedge (\forall i \le ?n.\ ?p\ i \ne ?p\ (Suc\ ?n)))$

thm Hypermap.lemma_sub_list:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n \longrightarrow (\forall i \le n.\ is\_inj\_list\ p\ i)$

thm Hypermap.lemma_inj_list:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n = (\forall (i::nat)\ j::nat.\ i \le n \wedge j < i \longrightarrow p\ j \ne p\ i)$

thm Hypermap.lemma_inj_list2:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n = (\forall (i::nat)\ j::nat.\ i \le n \wedge j \le n \wedge p\ i = p\ j \longrightarrow i = j)$

thm DEF_support_list:

$support\_list = (\lambda(\_2400023::nat \Rightarrow ?'a::type)\ \_2400024::nat.\ GSPEC\ (\lambda GEN\%PVAR\%244::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%244\ (i \le \_2400024)\ (\_2400023\ i)))$

thm Hypermap.support_list:

$\forall (n::nat)\ p::nat \Rightarrow ?'a::type.\ support\_list\ p\ n = GSPEC\ (\lambda GEN\%PVAR\%244::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%244\ (i \le n)\ (p\ i))$

thm Hypermap.lemma_finite_list:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ FINITE\ (support\_list\ p\ n)$

thm Hypermap.lemma_size_list:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n \longrightarrow CARD\ (support\_list\ p\ n) = Suc\ n$

thm DEF_in_list:

$in\_list = (\lambda(\_2400035::nat \Rightarrow ?'a::type)\ (\_2400036::nat)\ \_2400037::?'a::type.\ IN\ \_2400037\ (support\_list\ \_2400035\ \_2400036))$

thm Hypermap.in_list:

$\forall (x::?'a::type)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ in\_list\ p\ n\ x = IN\ x\ (support\_list\ p\ n)$

thm Hypermap.lemma_in_list:

$\forall (p::nat \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ in\_list\ p\ n\ x = (\exists j \le n.\ x = p\ j)$

thm Hypermap.lemma_in_list2:

$\forall (p::nat \Rightarrow ?'a::type)\ (n::nat)\ (x::?'a::type)\ j::nat.\ j \le n \wedge x = p\ j \longrightarrow in\_list\ p\ n\ x$

thm Hypermap.lemma_element_in_list:

$\forall (p::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ i \le n \longrightarrow in\_list\ p\ n\ (p\ i)$

thm Hypermap.lemma_not_in_list:

$\forall (p::nat \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ (\neg\ in\_list\ p\ n\ x) = (\forall j \leq n.\ x \neq p\ j)$

thm DEF_is_disjoint:

$is\_disjoint = (\lambda(\_2400148::nat \Rightarrow ?'a::type)\ (\_2400149::nat \Rightarrow ?'a::type)\ (\_2400150::nat)\ \_2400151::nat.\ DISJOINT\ (support\_list\ \_2400148\ \_2400150)\ (support\_list\ \_2400149\ \_2400151))$

thm Hypermap.is_disjoint:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_disjoint\ p\ q\ n\ m = DISJOINT\ (support\_list\ p\ n)\ (support\_list\ q\ m)$

thm Hypermap.lemma_set_disjoint:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\neg\ DISJOINT\ s\ t) = (\exists x::?'a::type.\ IN\ x\ s \wedge IN\ x\ t)$

thm Hypermap.lemma_list_disjoint1:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_disjoint\ p\ q\ n\ m = (\forall i \leq n.\ \neg\ in\_list\ q\ m\ (p\ i))$

thm Hypermap.lemma_list_disjoint2:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_disjoint\ p\ q\ n\ m = (\forall i \leq m.\ \neg\ in\_list\ p\ n\ (q\ i))$

thm Hypermap.lemma_list_disjoint:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_disjoint\ p\ q\ n\ m = (\forall (i::nat)\ j::nat.\ i \leq n \wedge j \leq m \longrightarrow p\ i \neq q\ j)$

thm DEF_glue:

$glue = (\lambda(\_2400224::nat \Rightarrow ?'a::type)\ (\_2400225::nat \Rightarrow ?'a::type)\ (\_2400226::nat)\ i::nat.\ if\ i \leq \_2400226\ then\ \_2400224\ i\ else\ \_2400225\ (i - \_2400226))$

thm Hypermap.glue:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ n::nat.\ glue\ p\ q\ n = (\lambda i::nat.\ if\ i \leq n\ then\ p\ i\ else\ q\ (i - n))$

thm Hypermap.start_glue_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ n::nat.\ glue\ p\ q\ n\ (0::nat) = p\ (0::nat)$

thm Hypermap.first_glue_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ i \leq n \longrightarrow glue\ p\ q\ n\ i = p\ i$

thm Hypermap.second_glue_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ p\ n = q\ (0::nat) \longrightarrow glue\ p\ q\ n\ (n + i) = q\ i$

thm DEF_is_glueing:

$is\_glueing = (\lambda(\_2400253::nat \Rightarrow ?'a::type)\ (\_2400254::nat \Rightarrow ?'a::type)\ (\_2400255::nat)\ \_2400256::nat.\ \_2400253\ \_2400255 = \_2400254\ (0::nat) \wedge (\forall j::nat.\ (1::nat) \leq j \wedge j \leq \_2400256 \longrightarrow \neg\ in\_list\ \_2400253\ \_2400255\ (\_2400254\ j)))$

thm Hypermap.is_glueing:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_glueing\ p\ q\ n\ m = (p\ n = q\ (0::nat) \wedge (\forall j::nat.\ (1::nat) \leq j \wedge j \leq m \longrightarrow \neg\ in\_list\ p\ n\ (q\ j)))$

thm Hypermap.lemma_glueing_condition:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_list\ p\ n \wedge is\_inj\_list\ q\ m \longrightarrow is\_glueing\ p\ q\ n\ m = (p\ n = q\ (0::nat) \wedge (\forall i<n.\ \neg\ in\_list\ q\ m\ (p\ i)))$

thm Hypermap.lemma_glue_inj_lists:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_list\ p\ n \wedge is\_inj\_list\ q\ m \wedge is\_glueing\ p\ q\ n\ m \longrightarrow is\_inj\_list\ (glue\ p\ q\ n)\ (n + m)$

thm DEF_join:

$join = (\lambda(\_2400323::nat \Rightarrow ?'a::type)\ (\_2400324::nat \Rightarrow ?'a::type)\ (\_2400325::nat)\ i::nat.\ if\ i \leq \_2400325\ then\ \_2400323\ i\ else\ \_2400324\ (pred\ (i - \_2400325)))$

thm Hypermap.join:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ n::nat.\ join\ p\ q\ n = (\lambda i::nat.\ if\ i \leq n\ then\ p\ i\ else\ q\ (pred\ (i - n)))$

thm Hypermap.first_join_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ i \leq n \longrightarrow join\ p\ q\ n\ i = p\ i$

thm Hypermap.second_join_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ join\ p\ q\ n\ (n + Suc\ i) = q\ i$

thm Hypermap.lemma_join_inj_lists:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_list\ p\ n \wedge is\_inj\_list\ q\ m \wedge is\_disjoint\ p\ q\ n\ m \longrightarrow is\_inj\_list\ (join\ p\ q\ n)\ (n + (m + (1::nat)))$

thm Hypermap.inj_iterate_segment:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ n::nat.\ permutes\ p\ s \wedge n \neq (0::nat) \longrightarrow (\forall m::nat.\ m \neq (0::nat) \wedge m < n \longrightarrow POWER\ p\ m \neq id) \longrightarrow (\forall (i::nat)\ j::nat.\ i < n \wedge j < i \longrightarrow POWER\ p\ i \neq POWER\ p\ j)$

thm Hypermap.inj_iterate_lemma:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. permutes p s ∧ (∀ n::nat. n ≠ (0::nat) ⟶ POWER p n ≠ id) ⟶ (∀ m::nat. CARD (GSPEC (λGEN%PVAR%245::?'a::type ⇒ ?'a::type. ∃ k::nat. SETSPEC GEN%PVAR%245 (k < m) (POWER p k))) = m)

thm Hypermap.finite_order:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ (∃ n::nat. n ≠ (0::nat) ∧ POWER p n = id)

thm Hypermap.lemma_order_permutation_exists:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ (∃ n::nat. n ≠ (0::nat) ∧ POWER p n = id ∧ (∀ m::nat. m ≠ (0::nat) ∧ m < n ⟶ POWER p m ≠ id))

thm DEF_order_permutation:

order_permutation = (SOME n::nat ⇒ (?'a::type ⇒ bool) ⇒ (?'a::type ⇒ ?'a::type) ⇒ nat. ∀ (_2400435::nat) (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ n _2400435 s p ≠ (0::nat) ∧ POWER p (n _2400435 s p) = id ∧ (∀ m::nat. m ≠ (0::nat) ∧ m < n _2400435 s p ⟶ POWER p m ≠ id)) (99::nat)

thm Hypermap.lemma_order_permutation:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ order_permutation s p ≠ (0::nat) ∧ POWER p (order_permutation s p) = id ∧ (∀ m::nat. m ≠ (0::nat) ∧ m < order_permutation s p ⟶ POWER p m ≠ id)

thm Hypermap.inverse_element_lemma:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ (∃ j::nat. HOL_Light_Import.inverse p = POWER p j)

thm Hypermap.inverse_element_via_order:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ HOL_Light_Import.inverse p = POWER p (pred (order_permutation s p))

thm Hypermap.lemma_permutation_via_its_inverse:

∀ (s::?'a::type ⇒ bool) p::?'a::type ⇒ ?'a::type. FINITE s ∧ permutes p s ⟶ (∃ j::nat. p = POWER (HOL_Light_Import.inverse p) j)

thm Hypermap.power_inverse_element_lemma:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) n::nat. FINITE s ∧ permutes p s ⟶ (∃ j::nat. POWER (HOL_Light_Import.inverse p) n = POWER p j)

thm Hypermap.inverse_relation:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \wedge y = p\ x \longrightarrow (\exists k::nat.\ x = POWER\ p\ k\ y)$

thm Hypermap.power_power_relation:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (y::?'a::type)\ n::nat.\ FINITE\ s \wedge permutes\ p\ s \wedge POWER\ p\ n\ x = y \longrightarrow (\exists j::nat.\ x = POWER\ p\ j\ y)$

thm Hypermap.elim_power_function:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (n::nat)\ m::nat.\ permutes\ p\ s \wedge POWER\ p\ (m + n)\ x = POWER\ p\ m\ x \longrightarrow POWER\ p\ n\ x = x$

thm Wrgcvdr_cizmrrh.X_IN_ITS_ORBIT:

$IN\ (?x::?'a::type)\ (orbit\_map\ (?f::?'a::type \Rightarrow ?'a::type)\ ?x)$

thm Hypermap.orbit_reflect:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ IN\ x\ (orbit\_map\ f\ x)$

thm Hypermap.orbit_sym:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow IN\ x\ (orbit\_map\ p\ y) \longrightarrow IN\ y\ (orbit\_map\ p\ x)$

thm Hypermap.orbit_trans:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ IN\ x\ (orbit\_map\ f\ y) \wedge IN\ y\ (orbit\_map\ f\ z) \longrightarrow IN\ x\ (orbit\_map\ f\ z)$

thm Hypermap.partition_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ HOL\_Light\_Import.INTER\ (orbit\_map\ p\ x)\ (orbit\_map\ p\ y) = EMPTY \vee orbit\_map\ p\ x = orbit\_map\ p\ y)$

thm Hypermap.card_orbit_le:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (n::nat)\ x::?'a::type.\ n \neq (0::nat) \wedge POWER\ f\ n\ x = x \longrightarrow CARD\ (orbit\_map\ f\ x) \leq n$

thm Hypermap.cyclic_maps:

$\forall (D::?'a::type \Rightarrow bool)\ (e::?'a::type \Rightarrow ?'a::type)\ (n::?'a::type \Rightarrow ?'a::type)\ f::?'a::type \Rightarrow ?'a::type.\ FINITE\ D \wedge permutes\ e\ D \wedge permutes\ n\ D \wedge permutes\ f\ D \wedge e \circ (n \circ f) = id \longrightarrow n \circ (f \circ e) = id \wedge f \circ (e \circ n) = id$

thm Hypermap.cyclic_inverses_maps:

$\forall (D::?'a::type \Rightarrow bool)\ (e::?'a::type \Rightarrow ?'a::type)\ (n::?'a::type \Rightarrow ?'a::type)\ f::?'a::type \Rightarrow ?'a::type.\ FINITE\ D \wedge permutes\ e\ D \wedge permutes\ n\ D \wedge permutes\ f\ D \wedge e \circ (n \circ f) = id \longrightarrow HOL\_Light\_Import.inverse\ n \circ (HOL\_Light\_Import.inverse\ e \circ HOL\_Light\_Import.inverse\ f) = id$

thm Hypermap.edge_refl:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (edge\ H\ x)$

thm Hypermap.node_refl:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (node\ H\ x)$

thm Hypermap.face_refl:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (face\ H\ x)$

thm Hypermap.hypermap_cyclic:

$\forall H::?'a::type\ hypermap.\ node\_map\ H \circ (face\_map\ H \circ edge\_map\ H) = id \wedge face\_map\ H \circ (edge\_map\ H \circ node\_map\ H) = id$

thm Hypermap.inverse_hypermap_maps:

$\forall H::?'a::type\ hypermap.\ HOL\_Light\_Import.inverse\ (edge\_map\ H) = node\_map\ H \circ face\_map\ H \wedge HOL\_Light\_Import.inverse\ (node\_map\ H) = face\_map\ H \circ edge\_map\ H \wedge HOL\_Light\_Import.inverse\ (face\_map\ H) = edge\_map\ H \circ node\_map\ H$

thm Hypermap.inverse2_hypermap_maps:

$\forall H::?'a::type\ hypermap.\ edge\_map\ H = HOL\_Light\_Import.inverse\ (face\_map\ H) \circ HOL\_Light\_Import.inverse\ (node\_map\ H) \wedge node\_map\ H = HOL\_Light\_Import.inverse\ (edge\_map\ H) \circ HOL\_Light\_Import.inverse\ (face\_map\ H) \wedge face\_map\ H = HOL\_Light\_Import.inverse\ (node\_map\ H) \circ HOL\_Light\_Import.inverse\ (edge\_map\ H)$

thm Hypermap.lemmaZHQCZLX:

$\forall H::?'a::type\ hypermap.\ simple\_hypermap\ H \wedge plain\_hypermap\ H \wedge (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow (3::nat) \leq CARD\ (face\ H\ x)) \longrightarrow (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow node\_map\ H\ x \neq x)$

thm DEF_connected_hypermap:

$connected\_hypermap = (\lambda\_2400505::?'a::type\ hypermap.\ number\_of\_components\ \_2400505 = (1::nat))$

thm Hypermap.connected_hypermap:

$\forall H::?'a::type\ hypermap.\ connected\_hypermap\ H = (number\_of\_components\ H = (1::nat))$

thm Hypermap.CARD_TWO_ELEMENTS:

$\forall (x::?'a::type)\ y::?'a::type.\ x \neq y \longrightarrow CARD\ (INSERT\ x\ (INSERT\ y\ EMPTY)) = (2::nat)$

thm Hypermap.FINITE_TWO_ELEMENTS:

$\forall (x::?'a::type)\ y::?'a::type.\ FINITE\ (INSERT\ x\ (INSERT\ y\ EMPTY))$

thm Hypermap.CARD_ATLEAST_1:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ FINITE\ s \wedge IN\ x\ s \longrightarrow (1::nat) \leq CARD\ s$

thm Hypermap.CARD_ATLEAST_2:

$\forall (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ FINITE\ s \wedge IN\ x\ s \wedge IN\ y\ s \wedge x \neq y \longrightarrow (2::nat) \leq CARD\ s$

thm Hypermap.orbit_single_lemma:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ orbit\_map\ f\ y = INSERT\ x\ EMPTY \longrightarrow x = y$

thm Hypermap.finite_orbits_lemma:

$\forall (D::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ D \wedge permutes\ p\ D \longrightarrow FINITE\ (set\_of\_orbits\ D\ p)$

thm Hypermap.lemma_disjoints:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ (\forall v::?'a::type \Rightarrow bool.\ IN\ v\ s \longrightarrow DISJOINT\ t\ v) \longrightarrow DISJOINT\ t\ (UNIONS\ s)$

thm Hypermap.lemma_partition:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow s = UNIONS\ (set\_of\_orbits\ s\ p)$

thm Hypermap.lemma_card_of_disjoint_covering:

$\forall t::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ t \wedge (\forall u::?'a::type \Rightarrow bool.\ IN\ u\ t \longrightarrow FINITE\ u) \wedge (\forall (s1::?'a::type \Rightarrow bool)\ s2::?'a::type \Rightarrow bool.\ IN\ s1\ t \wedge IN\ s2\ t \wedge s1 \neq s2 \longrightarrow DISJOINT\ s1\ s2) \longrightarrow CARD\ (UNIONS\ t) = nsum\ t\ CARD$

thm Hypermap.card_partition_formula:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow CARD\ s = nsum\ (set\_of\_orbits\ s\ p)\ CARD$

thm Hypermap.lemma_card_lower_bound:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ m::nat.\ FINITE\ s \wedge permutes\ p\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow m \leq CARD\ (orbit\_map\ p\ x)) \longrightarrow m * number\_of\_orbits\ s\ p \leq CARD\ s$

thm Hypermap.lemma_card_eq:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ m::nat.\ FINITE\ s \wedge permutes\ p\ s \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow CARD\ (orbit\_map\ p\ x) = m) \longrightarrow CARD\ s = m * number\_of\_orbits\ s\ p$

thm Hypermap.lemma_orbit_convolution_map:

$\forall p::?'a::type \Rightarrow ?'a::type.\ p \circ p = id \longrightarrow (\forall x::?'a::type.\ orbit\_map\ p\ x = INSERT\ x\ (INSERT\ (p\ x)\ EMPTY))$

thm Hypermap.lemma_nondegenerate_convolution:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \wedge p \circ p = id \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow p\ x \neq x) \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow FINITE\ (orbit\_map\ p\ x) \wedge CARD\ (orbit\_map\ p\ x) = (2::nat))$

thm Hypermap.lemmaTGJISOK:

$\forall H::?'a::type\ hypermap.\ connected\_hypermap\ H \wedge plain\_hypermap\ H \wedge planar\_hypermap\ H \wedge (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow edge\_map\ H\ x \neq x \wedge (3::nat) \leq CARD\ (node\ H\ x)) \longrightarrow CARD\ (dart\ H) \leq (6::nat) * number\_of\_faces\ H - (12::nat)$

thm Hypermap.lemma_subpath:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_path\ H\ p\ n \longrightarrow (\forall i \leq n.\ is\_path\ H\ p\ i)$

thm Hypermap.lemma_path_subset:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ IN\ x\ (dart\ H) \wedge p\ (0::nat) = x \wedge is\_path\ H\ p\ n \longrightarrow IN\ (p\ n)\ (dart\ H)$

thm Hypermap.lemma_component_subset:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow SUBSET\ (comb\_component\ H\ x)\ (dart\ H)$

thm Hypermap.lemma_edge_subset:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow SUBSET\ (edge\ H\ x)\ (dart\ H)$

thm Hypermap.lemma_node_subset:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow SUBSET\ (node\ H\ x)\ (dart\ H)$

thm Hypermap.lemma_face_subset:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow SUBSET\ (face\ H\ x)\ (dart\ H)$

thm Hypermap.lemma_component_reflect:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (comb\_component\ H\ x)$

thm Hypermap.lemma_def_path:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_path\ H\ p\ n = (\forall i<n.\ go\_one\_step\ H\ (p\ i)\ (p\ (Suc\ i)))$

thm DEF_edge_path:

$edge\_path = (\lambda(\_2401150::?'a::type\ hypermap)\ (\_2401151::?'a::type)\ \_2401152::nat.\ POWER\ (edge\_map\ \_2401150)\ \_2401152\ \_2401151)$

thm Hypermap.edge_path:

∀(H::?'a::type hypermap) (x::?'a::type) i::nat. edge_path H x i = POWER (edge_map H) i x

thm DEF_node_path:

node_path = (λ(_2401171::?'a::type hypermap) (_2401172::?'a::type) _2401173::nat. POWER (node_map _2401171) _2401173 _2401172)

thm Hypermap.node_path:

∀(H::?'a::type hypermap) (x::?'a::type) i::nat. node_path H x i = POWER (node_map H) i x

thm DEF_face_path:

face_path = (λ(_2401192::?'a::type hypermap) (_2401193::?'a::type) _2401194::nat. POWER (face_map _2401192) _2401194 _2401193)

thm Hypermap.face_path:

∀(H::?'a::type hypermap) (x::?'a::type) i::nat. face_path H x i = POWER (face_map H) i x

thm Hypermap.lemma_edge_path:

∀(H::?'a::type hypermap) (x::?'a::type) k::nat. is_path H (edge_path H x) k

thm Hypermap.lemma_node_path:

∀(H::?'a::type hypermap) (x::?'a::type) k::nat. is_path H (node_path H x) k

thm Hypermap.lemma_face_path:

∀(H::?'a::type hypermap) (x::?'a::type) k::nat. is_path H (face_path H x) k

thm Hypermap.lemma_glue_paths:

∀(H::?'a::type hypermap) (p::nat ⇒ ?'a::type) (q::nat ⇒ ?'a::type) (n::nat) m::nat. is_path H p n ∧ is_path H q m ∧ p n = q (0::nat) ⟶ is_path H (glue p q n) (n + m)

thm Hypermap.concatenate_two_paths:

∀(H::?'a::type hypermap) (p::nat ⇒ ?'a::type) (q::nat ⇒ ?'a::type) (n::nat) m::nat. is_path H p n ∧ is_path H q m ∧ p n = q (0::nat) ⟶ (∃g::nat ⇒ ?'a::type. g (0::nat) = p (0::nat) ∧ g (n + m) = q m ∧ is_path H g (n + m) ∧ (∀i≤n. g i = p i) ∧ (∀i≤m. g (n + i) = q i))

thm Hypermap.concatenate_paths:

∀(H::?'a::type hypermap) (p::nat ⇒ ?'a::type) (q::nat ⇒ ?'a::type) (n::nat) m::nat. is_path H p n ∧ is_path H q m ∧ p n = q (0::nat) ⟶ (∃g::nat ⇒ ?'a::type. g (0::nat) = p (0::nat) ∧ g (n + m) = q m ∧ is_path H g (n + m))

thm Hypermap.lemma_component_trans:

∀(H::?'a::type hypermap) (x::?'a::type) (y::?'a::type) z::?'a::type. IN y (comb_component H x) ∧ IN z (comb_component H y) ⟶ IN z (comb_component H x)

thm Hypermap.lemma_reverse_path:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_path\ H\ p\ n \longrightarrow (\exists (q::nat \Rightarrow ?'a::type)\ m::nat.\ q\ (0::nat) = p\ n \wedge q\ m = p\ (0::nat) \wedge is\_path\ H\ q\ m)$

thm Hypermap.lemma_component_symmetry:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (comb\_component\ H\ x) \longrightarrow IN\ x\ (comb\_component\ H\ y)$

thm Hypermap.partition_components:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ comb\_component\ H\ x = comb\_component\ H\ y \vee HOL\_Light\_Import.INTER\ (comb\_component\ H\ x)\ (comb\_component\ H\ y) = EMPTY$

thm Hypermap.lemma_partition_by_components:

$\forall H::?'a::type\ hypermap.\ dart\ H = UNIONS\ (set\_of\_components\ H)$

thm DEF_one_step_contour:

$one\_step\_contour = (\lambda(\_2401315::?'a::type\ hypermap)\ (\_2401316::?'a::type)\ \_2401317::?'a::type.\ \_2401317 = face\_map\ \_2401315\ \_2401316 \vee \_2401317 = HOL\_Light\_Import.inverse\ (node\_map\ \_2401315)\ \_2401316)$

thm Hypermap.one_step_contour:

$\forall (y::?'a::type)\ (H::?'a::type\ hypermap)\ x::?'a::type.\ one\_step\_contour\ H\ x\ y = (y = face\_map\ H\ x \vee y = HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)$

thm DEF_is_contour:

$is\_contour = (SOME\ is\_contour::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow bool.\ \forall \_2401343::nat.\ (\forall (H::?'a::type\ hypermap)\ p::nat \Rightarrow ?'a::type.\ is\_contour\ \_2401343\ H\ p\ (0::nat) = True) \wedge (\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_contour\ \_2401343\ H\ p\ (Suc\ n) = (is\_contour\ \_2401343\ H\ p\ n \wedge one\_step\_contour\ H\ (p\ n)\ (p\ (Suc\ n))))))\ (100::nat)$

thm Hypermap.is_contour_conjunct0:

$is\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True$

thm Hypermap.is_contour_conjunct1:

$is\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (Suc\ (?n::nat)) = (is\_contour\ ?H\ ?p\ ?n \wedge one\_step\_contour\ ?H\ (?p\ ?n)\ (?p\ (Suc\ ?n)))$

thm Hypermap.is_contour:

$is\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True \wedge is\_contour\ ?H\ ?p\ (Suc\ (?n::nat)) = (is\_contour\ ?H\ ?p\ ?n \wedge one\_step\_contour\ ?H\ (?p\ ?n)\ (?p\ (Suc\ ?n)))$

thm Hypermap.lemma_subcontour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_contour\ H\ p\ n \longrightarrow (\forall i \le n.\ is\_contour\ H\ p\ i)$

thm Hypermap.lemma_def_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_contour\ H\ p\ n = (\forall i<n.\ one\_step\_contour\ H\ (p\ i)\ (p\ (Suc\ i)))$

thm Hypermap.lemma_glue_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_contour\ H\ p\ n \wedge is\_contour\ H\ q\ m \wedge p\ n = q\ (0::nat) \longrightarrow is\_contour\ H\ (glue\ p\ q\ n)\ (n + m)$

thm Hypermap.concatenate_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_contour\ H\ p\ n \wedge is\_contour\ H\ q\ m \wedge p\ n = q\ (0::nat) \longrightarrow (\exists g::nat \Rightarrow ?'a::type.\ g\ (0::nat) = p\ (0::nat) \wedge g\ (n + m) = q\ m \wedge is\_contour\ H\ g\ (n + m) \wedge (\forall i \le n.\ g\ i = p\ i) \wedge (\forall i \le m.\ g\ (n + i) = q\ i))$

thm DEF_node_contour:

$node\_contour = (\lambda(\_2401358::?'a::type\ hypermap)\ (\_2401359::?'a::type)\ \_2401360::nat.\ POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ \_2401358))\ \_2401360\ \_2401359)$

thm Hypermap.node_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ i::nat.\ node\_contour\ H\ x\ i = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ i\ x$

thm DEF_face_contour:

$face\_contour = (\lambda(\_2401379::?'a::type\ hypermap)\ (\_2401380::?'a::type)\ \_2401381::nat.\ POWER\ (face\_map\ \_2401379)\ \_2401381\ \_2401380)$

thm Hypermap.face_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ i::nat.\ face\_contour\ H\ x\ i = POWER\ (face\_map\ H)\ i\ x$

thm Hypermap.lemma_node_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ is\_contour\ H\ (node\_contour\ H\ x)\ k$

thm Hypermap.lemma_face_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ is\_contour\ H\ (face\_contour\ H\ x)\ k$

thm Hypermap.existence_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_path\ H\ p\ n \longrightarrow (\exists (q::nat \Rightarrow ?'a::type)\ m::nat.\ q\ (0::nat) = p\ (0::nat) \wedge q\ m = p\ n \wedge is\_contour\ H\ q\ m)$

thm Hypermap.lemmaKDAEDEX:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (comb\_component\ H\ x) \longrightarrow (\exists (p::nat \Rightarrow ?'a::type)\ n::nat.\ p\ (0::nat) = x \wedge p\ n = y \wedge is\_contour\ H\ p\ n)$

thm DEF_is_inj_contour:

$is\_inj\_contour = (SOME\ is\_inj\_contour::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow bool.\ \forall\ \_2401417::nat.\ (\forall (H::?'a::type\ hypermap)\ p::nat \Rightarrow ?'a::type.\ is\_inj\_contour\ \_2401417\ H\ p\ (0::nat) = True) \wedge (\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ \_2401417\ H\ p\ (Suc\ n) = (is\_inj\_contour\ \_2401417\ H\ p\ n \wedge one\_step\_contour\ H\ (p\ n)\ (p\ (Suc\ n)) \wedge (\forall\ i{\le}n.\ p\ i \neq p\ (Suc\ n)))))\ (101::nat)$

thm Hypermap.is_inj_contour_conjunct0:

$is\_inj\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True$

thm Hypermap.is_inj_contour_conjunct1:

$is\_inj\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (Suc\ (?n::nat)) = (is\_inj\_contour\ ?H\ ?p\ ?n \wedge one\_step\_contour\ ?H\ (?p\ ?n)\ (?p\ (Suc\ ?n)) \wedge (\forall\ i{\le}?n.\ ?p\ i \neq ?p\ (Suc\ ?n)))$

thm Hypermap.is_inj_contour:

$is\_inj\_contour\ (?H::?'a::type\ hypermap)\ (?p::nat \Rightarrow ?'a::type)\ (0::nat) = True \wedge is\_inj\_contour\ ?H\ ?p\ (Suc\ (?n::nat)) = (is\_inj\_contour\ ?H\ ?p\ ?n \wedge one\_step\_contour\ ?H\ (?p\ ?n)\ (?p\ (Suc\ ?n)) \wedge (\forall\ i{\le}?n.\ ?p\ i \neq ?p\ (Suc\ ?n)))$

thm Hypermap.lemma_sub_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n \longrightarrow (\forall\ i{\le}n.\ is\_inj\_contour\ H\ p\ i)$

thm Hypermap.identify_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n \wedge (\forall\ i{\le}n.\ p\ i = q\ i) \longrightarrow is\_inj\_contour\ H\ q\ n$

thm Hypermap.lemma_def_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n = (is\_contour\ H\ p\ n \wedge (\forall (i::nat)\ j::nat.\ i \le n \wedge j < i \longrightarrow p\ j \neq p\ i))$

thm DEF_isolated_dart:

$isolated\_dart = (\lambda(\_2401438::?'a::type\ hypermap)\ \_2401439::?'a::type.\ edge\_map\ \_2401438\ \_2401439 = \_2401439 \wedge node\_map\ \_2401438\ \_2401439 = \_2401439 \wedge face\_map\ \_2401438\ \_2401439 = \_2401439)$

thm Hypermap.isolated_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ isolated\_dart\ H\ x = (edge\_map\ H\ x = x \wedge node\_map\ H\ x = x \wedge face\_map\ H\ x = x)$

thm DEF_is_edge_degenerate:

$is\_edge\_degenerate = (\lambda(\_2401450{::}?'a{::}type\ hypermap)\ \_2401451{::}?'a{::}type.\ edge\_map\ \_2401450\ \_2401451 = \_2401451 \wedge node\_map\ \_2401450\ \_2401451 \neq \_2401451 \wedge face\_map\ \_2401450\ \_2401451 \neq \_2401451)$

thm Hypermap.is_edge_degenerate:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ is\_edge\_degenerate\ H\ x = (edge\_map\ H\ x = x \wedge node\_map\ H\ x \neq x \wedge face\_map\ H\ x \neq x)$

thm DEF_is_node_degenerate:

$is\_node\_degenerate = (\lambda(\_2401462{::}?'a{::}type\ hypermap)\ \_2401463{::}?'a{::}type.\ edge\_map\ \_2401462\ \_2401463 \neq \_2401463 \wedge node\_map\ \_2401462\ \_2401463 = \_2401463 \wedge face\_map\ \_2401462\ \_2401463 \neq \_2401463)$

thm Hypermap.is_node_degenerate:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ is\_node\_degenerate\ H\ x = (edge\_map\ H\ x \neq x \wedge node\_map\ H\ x = x \wedge face\_map\ H\ x \neq x)$

thm DEF_is_face_degenerate:

$is\_face\_degenerate = (\lambda(\_2401474{::}?'a{::}type\ hypermap)\ \_2401475{::}?'a{::}type.\ edge\_map\ \_2401474\ \_2401475 \neq \_2401475 \wedge node\_map\ \_2401474\ \_2401475 \neq \_2401475 \wedge face\_map\ \_2401474\ \_2401475 = \_2401475)$

thm Hypermap.is_face_degenerate:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ is\_face\_degenerate\ H\ x = (edge\_map\ H\ x \neq x \wedge node\_map\ H\ x \neq x \wedge face\_map\ H\ x = x)$

thm Hypermap.degenerate_lemma:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ dart\_degenerate\ H\ x = (isolated\_dart\ H\ x \vee is\_edge\_degenerate\ H\ x \vee is\_node\_degenerate\ H\ x \vee is\_face\_degenerate\ H\ x)$

thm Hypermap.lemma_category_darts:

$\forall (H{::}?'a{::}type\ hypermap)\ x{::}?'a{::}type.\ dart\_nondegenerate\ H\ x \vee dart\_degenerate\ H\ x$

thm Hypermap.lemma_pair_representation:

$\forall S{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type).\ S = (fst\ S,\ fst\ (snd\ S),\ fst\ (snd\ (snd\ S)),\ snd\ (snd\ (snd\ S)))$

thm Hypermap.lemma_pair_eq:

$\forall (S{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type))\ U{::}(?'a{::}type \Rightarrow bool) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type) \times (?'a{::}type \Rightarrow ?'a{::}type).\ fst\ S = fst\ U \wedge fst\ (snd\ S)$

$= fst\ (snd\ U) \wedge fst\ (snd\ (snd\ S)) = fst\ (snd\ (snd\ U)) \wedge snd\ (snd\ (snd\ S)) = snd\ (snd\ (snd\ U)) \longrightarrow S = U$

thm Hypermap.lemma_hypermap_eq:

$\forall (H::?'a::type\ hypermap)\ H'::?'a::type\ hypermap.\ (H = H') = (dart\ H = dart\ H' \wedge edge\_map\ H = edge\_map\ H' \wedge node\_map\ H = node\_map\ H' \wedge face\_map\ H = face\_map\ H')$

thm Hypermap.lemma_hypermap_rep:

$\forall (D::?'a::type \Rightarrow bool)\ (e::?'a::type \Rightarrow ?'a::type)\ (n::?'a::type \Rightarrow ?'a::type)\ f::?'a::type \Rightarrow ?'a::type.\ FINITE\ D \wedge permutes\ e\ D \wedge permutes\ n\ D \wedge permutes\ f\ D \wedge e \circ (n \circ f) = id \longrightarrow dart\ (hypermap\ (D,\ e,\ n,\ f)) = D \wedge edge\_map\ (hypermap\ (D,\ e,\ n,\ f)) = e \wedge node\_map\ (hypermap\ (D,\ e,\ n,\ f)) = n \wedge face\_map\ (hypermap\ (D,\ e,\ n,\ f)) = f$

thm DEF_shift:

$shift = (\lambda\_2401874::?'a::type\ hypermap.\ hypermap\ (dart\ \_2401874,\ node\_map\ \_2401874,\ face\_map\ \_2401874,\ edge\_map\ \_2401874))$

thm Hypermap.shift:

$\forall H::?'a::type\ hypermap.\ shift\ H = hypermap\ (dart\ H,\ node\_map\ H,\ face\_map\ H,\ edge\_map\ H)$

thm Hypermap.shift_lemma:

$\forall H::?'a::type\ hypermap.\ dart\ H = dart\ (shift\ H) \wedge edge\_map\ H = face\_map\ (shift\ H) \wedge node\_map\ H = edge\_map\ (shift\ H) \wedge face\_map\ H = node\_map\ (shift\ H)$

thm Hypermap.double_shift_lemma:

$\forall H::?'a::type\ hypermap.\ dart\ H = dart\ (shift\ (shift\ H)) \wedge edge\_map\ H = node\_map\ (shift\ (shift\ H)) \wedge node\_map\ H = face\_map\ (shift\ (shift\ H)) \wedge face\_map\ H = edge\_map\ (shift\ (shift\ H))$

thm DEF_edge_walkup:

$edge\_walkup = (\lambda(\_2401879::?'a::type\ hypermap)\ \_2401880::?'a::type.\ hypermap\ (DELETE\ (dart\ \_2401879)\ \_2401880,\ HOL\_Light\_Import.inverse\ (swap\ (\_2401880,\ face\_map\ \_2401879\ \_2401880) \circ face\_map\ \_2401879) \circ HOL\_Light\_Import.inverse\ (swap\ (\_2401880,\ node\_map\ \_2401879\ \_2401880) \circ node\_map\ \_2401879),\ swap\ (\_2401880,\ node\_map\ \_2401879\ \_2401880) \circ node\_map\ \_2401879,\ swap\ (\_2401880,\ face\_map\ \_2401879\ \_2401880) \circ face\_map\ \_2401879))$

thm Hypermap.edge_walkup:

$\forall (x::?'a::type)\ H::?'a::type\ hypermap.\ edge\_walkup\ H\ x = hypermap\ (DELETE\ (dart\ H)\ x,\ HOL\_Light\_Import.inverse\ (swap\ (x,\ face\_map\ H\ x) \circ face\_map\ H) \circ HOL\_Light\_Import.inverse\ (swap\ (x,\ node\_map\ H\ x) \circ node\_map\ H),$

swap (*x*, *node_map H x*) ∘ *node_map H*, *swap* (*x*, *face_map H x*) ∘ *face_map H*)

thm DEF_node_walkup:

node_walkup = (λ(*_2401891*::*?'a*::*type hypermap*) *_2401892*::*?'a*::*type*. *shift* (*shift* (*edge_walkup* (*shift _2401891*) *_2401892*)))

thm Hypermap.node_walkup:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *node_walkup H x* = *shift* (*shift* (*edge_walkup* (*shift H*) *x*))

thm DEF_face_walkup:

face_walkup = (λ(*_2401903*::*?'a*::*type hypermap*) *_2401904*::*?'a*::*type*. *shift* (*edge_walkup* (*shift* (*shift _2401903*)) *_2401904*))

thm Hypermap.face_walkup:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *face_walkup H x* = *shift* (*edge_walkup* (*shift* (*shift H*)) *x*)

thm DEF_double_edge_walkup:

double_edge_walkup = (λ(*_2401915*::*?'a*::*type hypermap*) *_2401916*::*?'a*::*type*. *edge_walkup* (*edge_walkup _2401915 _2401916*))

thm Hypermap.double_edge_walkup:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *double_edge_walkup H x y* = *edge_walkup* (*edge_walkup H x*) *y*

thm DEF_double_node_walkup:

double_node_walkup = (λ(*_2401936*::*?'a*::*type hypermap*) *_2401937*::*?'a*::*type*. *node_walkup* (*node_walkup _2401936 _2401937*))

thm Hypermap.double_node_walkup:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *double_node_walkup H x y* = *node_walkup* (*node_walkup H x*) *y*

thm DEF_double_face_walkup:

double_face_walkup = (λ(*_2401957*::*?'a*::*type hypermap*) *_2401958*::*?'a*::*type*. *face_walkup* (*face_walkup _2401957 _2401958*))

thm Hypermap.double_face_walkup:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *double_face_walkup H x y* = *face_walkup* (*face_walkup H x*) *y*

thm Hypermap.walkup_permutes:

∀ (*D*::*?'a*::*type* ⇒ *bool*) (*p*::*?'a*::*type* ⇒ *?'a*::*type*) *x*::*?'a*::*type*. *FINITE D* ∧ *permutes p D* ⟶ *permutes* (*swap* (*x*, *p x*) ∘ *p*) (*DELETE D x*)

thm Hypermap.PERMUTES_COMPOSITION:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (q::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ p\ s \wedge permutes\ q\ s \longrightarrow permutes\ (q \circ p)\ s$

thm Hypermap.lemma_edge_walkup:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\ (edge\_walkup\ H\ x) = DELETE\ (dart\ H)\ x \wedge edge\_map\ (edge\_walkup\ H\ x) = HOL\_Light\_Import.inverse\ (swap\ (x,\ face\_map\ H\ x) \circ face\_map\ H) \circ HOL\_Light\_Import.inverse\ (swap\ (x,\ node\_map\ H\ x) \circ node\_map\ H) \wedge node\_map\ (edge\_walkup\ H\ x) = swap\ (x,\ node\_map\ H\ x) \circ node\_map\ H \wedge face\_map\ (edge\_walkup\ H\ x) = swap\ (x,\ face\_map\ H\ x) \circ face\_map\ H$

thm Hypermap.node_map_walkup:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ node\_map\ (edge\_walkup\ H\ x)\ x = x \wedge node\_map\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x) = node\_map\ H\ x \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ x \longrightarrow node\_map\ (edge\_walkup\ H\ x)\ y = node\_map\ H\ y)$

thm Hypermap.face_map_walkup:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ face\_map\ (edge\_walkup\ H\ x)\ x = x \wedge face\_map\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x) = face\_map\ H\ x \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (face\_map\ H)\ x \longrightarrow face\_map\ (edge\_walkup\ H\ x)\ y = face\_map\ H\ y)$

thm Hypermap.lemma_edge_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (edge\_map\ H\ x = x) = (face\_map\ H\ x = HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)$

thm Hypermap.lemma_node_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (node\_map\ H\ x = x) = (edge\_map\ H\ x = HOL\_Light\_Import.inverse\ (face\_map\ H)\ x)$

thm Hypermap.lemma_face_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (face\_map\ H\ x = x) = (node\_map\ H\ x = HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)$

thm Hypermap.fixed_point_lemma:

$\forall (D::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ D \longrightarrow (\forall x::?'a::type.\ (p\ x = x) = (HOL\_Light\_Import.inverse\ p\ x = x))$

thm Hypermap.non_fixed_point_lemma:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ permutes\ p\ s \longrightarrow (\forall x::?'a::type.\ (p\ x \neq x) = (HOL\_Light\_Import.inverse\ p\ x \neq x))$

thm Hypermap.lemma_inverse_maps_at_nondegenerate_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\_nondegenerate\ H\ x \longrightarrow HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \neq x \wedge HOL\_Light\_Import.inverse\ (node\_map\ H)\ x \neq x \wedge HOL\_Light\_Import.inverse\ (face\_map\ H)\ x \neq x$

thm Hypermap.aux_permutes_conversion:

$\forall (D::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (q::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ permutes\ p\ D \wedge permutes\ q\ D \longrightarrow (HOL\_Light\_Import.inverse\ p\ (HOL\_Light\_Import.inverse\ q\ x) = y) = (q\ (p\ y) = x)$

thm Hypermap.edge_map_walkup:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ edge\_map\ (edge\_walkup\ H\ x)\ x = x \wedge (node\_map\ H\ x \neq x \wedge edge\_map\ H\ x \neq x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x) = edge\_map\ H\ x) \wedge (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x \neq x \wedge HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \neq x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) = HOL\_Light\_Import.inverse\ (face\_map\ H)\ x) \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \wedge y \neq node\_map\ H\ x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ y = edge\_map\ H\ y)$

thm DEF_power_list:

$power\_list = (\lambda(\_2402738::?'a::type \Rightarrow ?'a::type)\ (\_2402739::?'a::type)\ i::nat.\ POWER\ \_2402738\ i\ \_2402739)$

thm Hypermap.power_list:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ power\_list\ p\ x = (\lambda i::nat.\ POWER\ p\ i\ x)$

thm DEF_inj_orbit:

$inj\_orbit = (SOME\ inj\_orbit::nat \Rightarrow (?'a::type \Rightarrow ?'a::type) \Rightarrow ?'a::type \Rightarrow nat \Rightarrow bool.\ \forall \_2402757::nat.\ (\forall (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ inj\_orbit\ \_2402757\ p\ x\ (0::nat) = True) \wedge (\forall (n::nat)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ inj\_orbit\ \_2402757\ p\ x\ (Suc\ n) = (inj\_orbit\ \_2402757\ p\ x\ n \wedge (\forall j \leq n.\ POWER\ p\ (Suc\ n)\ x \neq POWER\ p\ j\ x))))\ (102::nat)$

thm Hypermap.inj_orbit_conjunct0:

$inj\_orbit\ (?p::?'a::type \Rightarrow ?'a::type)\ (?x::?'a::type)\ (0::nat) = True$

thm Hypermap.inj_orbit_conjunct1:

$inj\_orbit\ (?p::?'a::type \Rightarrow ?'a::type)\ (?x::?'a::type)\ (Suc\ (?n::nat)) = (inj\_orbit\ ?p\ ?x\ ?n \wedge (\forall j \leq ?n.\ POWER\ ?p\ (Suc\ ?n)\ ?x \neq POWER\ ?p\ j\ ?x))$

thm Hypermap.inj_orbit:

$inj\_orbit\ (?p::?'a::type \Rightarrow ?'a::type)\ (?x::?'a::type)\ (0::nat) = True \wedge inj\_orbit\ ?p\ ?x\ (Suc\ (?n::nat)) = (inj\_orbit\ ?p\ ?x\ ?n \wedge (\forall j \leq ?n.\ POWER\ ?p\ (Suc\ ?n)\ ?x \neq POWER\ ?p\ j\ ?x))$

thm Hypermap.lemma_inj_orbit_via_list:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ inj\_orbit\ p\ x\ n = is\_inj\_list\ (power\_list\ p\ x)\ n$

thm Hypermap.lemma_def_inj_orbit:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ inj\_orbit\ p\ x\ n = (\forall (i::nat)\ j::nat.\ i \leq n \wedge j < i \longrightarrow POWER\ p\ i\ x \neq POWER\ p\ j\ x)$

thm Hypermap.lemma_inj_orbit:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ inj\_orbit\ p\ x\ n = (\forall (i::nat)\ j::nat.\ i \leq n \wedge j \leq n \wedge POWER\ p\ i\ x = POWER\ p\ j\ x \longrightarrow i = j)$

thm Hypermap.lemma_sub_inj_orbit:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ inj\_orbit\ p\ x\ n \longrightarrow (\forall m \leq n.\ inj\_orbit\ p\ x\ m)$

thm Hypermap.inj_orbit_step:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ permutes\ p\ s \wedge inj\_orbit\ p\ x\ n \wedge POWER\ p\ (Suc\ n)\ x \neq x \longrightarrow inj\_orbit\ p\ x\ (Suc\ n)$

thm Hypermap.lemma_subset_orbit:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%251::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%251\ (i \leq n)\ (POWER\ p\ i\ x)))\ (orbit\_map\ p\ x)$

thm Hypermap.lemma_segment_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow (\forall m<CARD\ (orbit\_map\ p\ x).\ inj\_orbit\ p\ x\ m)$

thm Hypermap.lemma_cycle_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow POWER\ p\ (CARD\ (orbit\_map\ p\ x))\ x = x$

thm Hypermap.lemma_index_on_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \wedge IN\ y\ (orbit\_map\ p\ x) \longrightarrow (\exists n<CARD\ (orbit\_map\ p\ x).\ y = POWER\ p\ n\ x)$

thm Hypermap.lemma_congruence_on_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (n::nat)\ m::nat.\ FINITE\ s \wedge permutes\ p\ s \wedge n < CARD\ (orbit\_map\ p\ x) \wedge POWER\ p\ n\ x = POWER\ p\ m\ x \longrightarrow (\exists q::nat.\ m = q * CARD\ (orbit\_map\ p\ x) + n)$

thm DEF_is_edge_merge:

$is\_edge\_merge = (\lambda(\_2402936::?'a::type\ hypermap)\ \_2402937::?'a::type.\ dart\_nondegenerate\ \_2402936\ \_2402937 \wedge \neg\ IN\ (node\_map\ \_2402936\ \_2402937)\ (edge\ \_2402936\ \_2402937))$

thm Hypermap.is_edge_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_edge\_merge\ H\ x = (dart\_nondegenerate\ H\ x \wedge \neg\ IN\ (node\_map\ H\ x)\ (edge\ H\ x))$

thm DEF_is_node_merge:

$is\_node\_merge = (\lambda(\_2402948::?'a::type\ hypermap)\ \_2402949::?'a::type.\ dart\_nondegenerate\ \_2402948\ \_2402949 \wedge \neg\ IN\ (face\_map\ \_2402948\ \_2402949)\ (node\ \_2402948\ \_2402949))$

thm Hypermap.is_node_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_node\_merge\ H\ x = (dart\_nondegenerate\ H\ x \wedge \neg\ IN\ (face\_map\ H\ x)\ (node\ H\ x))$

thm DEF_is_face_merge:

$is\_face\_merge = (\lambda(\_2402960::?'a::type\ hypermap)\ \_2402961::?'a::type.\ dart\_nondegenerate\ \_2402960\ \_2402961 \wedge \neg\ IN\ (edge\_map\ \_2402960\ \_2402961)\ (face\ \_2402960\ \_2402961))$

thm Hypermap.is_face_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_face\_merge\ H\ x = (dart\_nondegenerate\ H\ x \wedge \neg\ IN\ (edge\_map\ H\ x)\ (face\ H\ x))$

thm DEF_is_edge_split:

$is\_edge\_split = (\lambda(\_2402972::?'a::type\ hypermap)\ \_2402973::?'a::type.\ dart\_nondegenerate\ \_2402972\ \_2402973 \wedge IN\ (node\_map\ \_2402972\ \_2402973)\ (edge\ \_2402972\ \_2402973))$

thm Hypermap.is_edge_split:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_edge\_split\ H\ x = (dart\_nondegenerate\ H\ x \wedge IN\ (node\_map\ H\ x)\ (edge\ H\ x))$

thm DEF_is_node_split:

$is\_node\_split = (\lambda(\_2402984::?'a::type\ hypermap)\ \_2402985::?'a::type.\ dart\_nondegenerate\ \_2402984\ \_2402985 \wedge IN\ (face\_map\ \_2402984\ \_2402985)\ (node\ \_2402984\ \_2402985))$

thm Hypermap.is_node_split:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_node\_split\ H\ x = (dart\_nondegenerate\ H\ x \wedge IN\ (face\_map\ H\ x)\ (node\ H\ x))$

thm DEF_is_face_split:

$is\_face\_split = (\lambda(\_2402996::?'a::type\ hypermap)\ \_2402997::?'a::type.\ dart\_nondegenerate\ \_2402996\ \_2402997 \wedge IN\ (edge\_map\ \_2402996\ \_2402997)\ (face\ \_2402996\ \_2402997))$

thm Hypermap.is_face_split:

∀ (H::?'a::type hypermap) x::?'a::type. is_face_split H x = (dart_nondegenerate H x ∧ IN (edge_map H x) (face H x))

thm Hypermap.INVERSE_EVALUATION:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) x::?'a::type. FINITE s ∧ permutes p s ⟶ (∃ j::nat. HOL_Light_Import.inverse p x = POWER p j x)

thm Hypermap.lemma_orbit_identity:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) (x::?'a::type) y::?'a::type. FINITE s ∧ permutes p s ∧ IN y (orbit_map p x) ⟶ orbit_map p x = orbit_map p y

thm Hypermap.lemma_edge_identity:

∀ (H::?'a::type hypermap) (x::?'a::type) y::?'a::type. IN y (edge H x) ⟶ edge H x = edge H y

thm Hypermap.lemma_node_identity:

∀ (H::?'a::type hypermap) (x::?'a::type) y::?'a::type. IN y (node H x) ⟶ node H x = node H y

thm Hypermap.lemma_face_identity:

∀ (H::?'a::type hypermap) (x::?'a::type) y::?'a::type. IN y (face H x) ⟶ face H x = face H y

thm Hypermap.lemma_orbit_disjoint:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) (x::?'a::type) y::?'a::type. FINITE s ∧ permutes p s ∧ ¬ IN y (orbit_map p x) ⟶ HOL_Light_Import.INTER (orbit_map p x) (orbit_map p y) = EMPTY

thm Hypermap.INVERSE_POWER_MAP:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) n::nat. FINITE s ∧ permutes p s ⟶ HOL_Light_Import.inverse p ∘ POWER p (Suc n) = POWER p n

thm Hypermap.INVERSE_POWER_EVALUATION:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) (x::?'a::type) n::nat. FINITE s ∧ permutes p s ⟶ HOL_Light_Import.inverse p (POWER p (Suc n) x) = POWER p n x

thm Hypermap.lemma_in_disjoint:

∀ (s::?'a::type ⇒ bool) (t::?'a::type ⇒ bool) x::?'a::type. HOL_Light_Import.INTER s t = EMPTY ∧ IN x s ⟶ ¬ IN x t

thm Hypermap.lemma_not_in_orbit:

∀ (s::?'a::type ⇒ bool) (p::?'a::type ⇒ ?'a::type) (x::?'a::type) (y::?'a::type) n::nat. FINITE s ∧ permutes p s ∧ ¬ IN y (orbit_map p x) ⟶ y ≠ POWER p n x

thm Hypermap.lemma_orbit_power:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ n::nat.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow orbit\_map\ p\ x = orbit\_map\ p\ (POWER\ p\ n\ x)$

thm Hypermap.lemma_inverse_in_orbit:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow IN\ (HOL\_Light\_Import.inverse\ p\ x)\ (orbit\_map\ p\ x)$

thm Hypermap.lemmaFKSNTKR:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ simple\_hypermap\ H \wedge IN\ x\ (dart\ H) \wedge edge\_map\ H\ x \neq x \wedge dart\_nondegenerate\ H\ x \wedge dart\_nondegenerate\ H\ (edge\_map\ H\ x) \longrightarrow (edge\_map\ H\ (edge\_map\ H\ x) = x \longrightarrow is\_face\_merge\ H\ x) \wedge is\_node\_merge\ H\ x$

thm DEF_planar_ind:

$planar\_ind = (\lambda\_2403169::?'a::type\ hypermap.\ real\_of\_nat\ (number\_of\_edges\ \_2403169) + (real\_of\_nat\ (number\_of\_nodes\ \_2403169) + (real\_of\_nat\ (number\_of\_faces\ \_2403169) - real\_of\_nat\ (CARD\ (dart\ \_2403169)) - real\_of\_nat\ (2::nat) * real\_of\_nat\ (number\_of\_components\ \_2403169))))$

thm Hypermap.planar_ind:

$\forall H::?'a::type\ hypermap.\ planar\_ind\ H = real\_of\_nat\ (number\_of\_edges\ H) + (real\_of\_nat\ (number\_of\_nodes\ H) + (real\_of\_nat\ (number\_of\_faces\ H) - real\_of\_nat\ (CARD\ (dart\ H)) - real\_of\_nat\ (2::nat) * real\_of\_nat\ (number\_of\_components\ H)))$

thm Hypermap.lemma_planar_hypermap:

$\forall H::?'a::type\ hypermap.\ planar\_hypermap\ H = (planar\_ind\ H = (0::real))$

thm Hypermap.lemma_null_hypermap_planar_index:

$\forall H::?'a::type\ hypermap.\ CARD\ (dart\ H) = (0::nat) \longrightarrow planar\_ind\ H = (0::real)$

thm Hypermap.lemma_shift_component_invariant:

$\forall H::?'a::type\ hypermap.\ set\_of\_components\ H = set\_of\_components\ (shift\ H)$

thm Hypermap.lemma_planar_invariant_shift:

$\forall H::?'a::type\ hypermap.\ planar\_ind\ H = planar\_ind\ (shift\ H)$

thm Hypermap.in_orbit_map1:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ IN\ (p\ x)\ (orbit\_map\ p\ x)$

thm Hypermap.lemma_orbit_eq:

$\forall (p::?'a::type \Rightarrow ?'a::type)\ (q::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ (\forall n::nat.\ POWER\ p\ n\ x = POWER\ q\ n\ x) \longrightarrow orbit\_map\ p\ x = orbit\_map\ q\ x$

thm Hypermap.lemma_not_in_orbit_powers:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) (*p*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) (*n*::*nat*) *m*::*nat*. *FINITE s* $\wedge$ *permutes p s* $\wedge$ $\neg$ *IN y* (*orbit_map p x*) $\longrightarrow$ *POWER p n y* $\neq$ *POWER p m x*

thm Hypermap.lemma_walkup_nodes:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) $\longrightarrow$ *DELETE* (*node_set H*) (*node H x*) = *DELETE* (*node_set* (*edge_walkup H x*)) (*node* (*edge_walkup H x*) (*HOL_Light_Import.inverse* (*node_map H*) *x*))

thm Hypermap.lemma_walkup_faces:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) $\longrightarrow$ *DELETE* (*face_set H*) (*face H x*) = *DELETE* (*face_set* (*edge_walkup H x*)) (*face* (*edge_walkup H x*) (*HOL_Light_Import.inverse* (*face_map H*) *x*))

thm Hypermap.lemma_walkup_first_edge_eq:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) $\wedge$ $\neg$ *IN x* (*edge H y*) $\wedge$ $\neg$ *IN* (*node_map H x*) (*edge H y*) $\longrightarrow$ *edge H y* = *edge* (*edge_walkup H x*) *y* $\wedge$ $\neg$ *IN* (*HOL_Light_Import.inverse* (*edge_map H*) *x*) (*edge H y*)

thm Hypermap.lemma_walkup_second_edge_eq:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) $\wedge$ *IN y* (*dart H*) $\wedge$ *y* $\neq$ *x* $\wedge$ $\neg$ *IN* (*node_map H x*) (*edge* (*edge_walkup H x*) *y*) $\wedge$ $\neg$ *IN* (*HOL_Light_Import.inverse* (*edge_map H*) *x*) (*edge* (*edge_walkup H x*) *y*) $\longrightarrow$ *edge H y* = *edge* (*edge_walkup H x*) *y* $\wedge$ $\neg$ *IN x* (*edge H y*) $\wedge$ $\neg$ *IN* (*node_map H x*) (*edge H y*)

thm Hypermap.lemma_walkup_edges:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) $\longrightarrow$ *DIFF* (*edge_set H*) (*INSERT* (*edge H x*) (*INSERT* (*edge H* (*node_map H x*)) *EMPTY*)) = *DIFF* (*edge_set* (*edge_walkup H x*)) (*INSERT* (*edge* (*edge_walkup H x*) (*node_map H x*)) (*INSERT* (*edge* (*edge_walkup H x*) (*HOL_Light_Import.inverse* (*edge_map H*) *x*)) *EMPTY*))

thm Hypermap.in_set_of_orbits:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) *p*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*. *permutes p s* $\longrightarrow$ ($\forall$ *x*::*?'a*::*type*. *IN x s* = *IN* (*orbit_map p x*) (*set_of_orbits s p*))

thm Hypermap.lemma_in_hypermap_orbits:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) = *IN* (*edge H x*) (*edge_set H*) $\wedge$ *IN x* (*dart H*) = *IN* (*node H x*) (*node_set H*) $\wedge$ *IN x* (*dart H*) = *IN* (*face H x*) (*face_set H*)

thm Hypermap.lemma_in_edge_set:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) = IN\ (edge\ H\ x)\ (edge\_set\ H)$

thm Hypermap.lemma_in_node_set:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) = IN\ (node\ H\ x)\ (node\_set\ H)$

thm Hypermap.lemma_in_face_set:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) = IN\ (face\ H\ x)\ (face\_set\ H)$

thm Hypermap.lemma_edge_representation:

$\forall (H::?'a::type\ hypermap)\ u::?'a::type \Rightarrow bool.\ IN\ u\ (edge\_set\ H) \longrightarrow (\exists x::?'a::type.\ IN\ x\ (dart\ H) \wedge u = edge\ H\ x)$

thm Hypermap.lemma_node_representation:

$\forall (H::?'a::type\ hypermap)\ u::?'a::type \Rightarrow bool.\ IN\ u\ (node\_set\ H) \longrightarrow (\exists x::?'a::type.\ IN\ x\ (dart\ H) \wedge u = node\ H\ x)$

thm Hypermap.lemma_face_representation:

$\forall (H::?'a::type\ hypermap)\ u::?'a::type \Rightarrow bool.\ IN\ u\ (face\_set\ H) \longrightarrow (\exists x::?'a::type.\ IN\ x\ (dart\ H) \wedge u = face\ H\ x)$

thm Hypermap.lemma_component_representation:

$\forall (H::?'a::type\ hypermap)\ u::?'a::type \Rightarrow bool.\ IN\ u\ (set\_of\_components\ H) \longrightarrow (\exists x::?'a::type.\ IN\ x\ (dart\ H) \wedge u = comb\_component\ H\ x)$

thm Hypermap.lemma_in_subset:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ SUBSET\ s\ t \wedge IN\ x\ s \longrightarrow IN\ x\ t$

thm Hypermap.lemma_complement_two_edges:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge IN\ y\ (dart\ H) \longrightarrow HOL\_Light\_Import.UNION\ (edge\ H\ x)\ (edge\ H\ y) = DIFF\ (dart\ H)\ (UNIONS\ (DIFF\ (edge\_set\ H)\ (INSERT\ (edge\ H\ x)\ (INSERT\ (edge\ H\ y)\ EMPTY))))$

thm Hypermap.lemma_edge_complement:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow edge\ H\ x = DIFF\ (dart\ H)\ (UNIONS\ (DELETE\ (edge\_set\ H)\ (edge\ H\ x)))$

thm Hypermap.lemma_in_walkup_dart:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge IN\ y\ (dart\ H) \wedge y \neq x \longrightarrow IN\ y\ (dart\ (edge\_walkup\ H\ x))$

thm Hypermap.lemma_edge_map_walkup_in_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge edge\_map\ H\ x \neq x \longrightarrow IN\ (edge\_map\ H\ x)\ (dart\ (edge\_walkup\ H\ x)) \wedge IN\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)\ (dart\ (edge\_walkup\ H\ x))$

thm Hypermap.lemma_node_map_walkup_in_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge node\_map\ H\ x \neq x \longrightarrow IN\ (node\_map\ H\ x)\ (dart\ (edge\_walkup\ H\ x)) \wedge IN\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)\ (dart\ (edge\_walkup\ H\ x))$

thm Hypermap.lemma_face_map_walkup_in_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge face\_map\ H\ x \neq x \longrightarrow IN\ (face\_map\ H\ x)\ (dart\ (edge\_walkup\ H\ x)) \wedge IN\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x)\ (dart\ (edge\_walkup\ H\ x))$

thm Hypermap.lemma_walkup_support_edges:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \longrightarrow HOL\_Light\_Import.UNION\ (edge\ H\ x)\ (edge\ H\ (node\_map\ H\ x)) = HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (HOL\_Light\_Import.UNION\ (edge\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x))\ (edge\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)))$

thm Hypermap.lemma_in_edge:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (edge\ H\ x) = (\exists j::nat.\ y = POWER\ (edge\_map\ H)\ j\ x)$

thm Hypermap.lemma_in_edge2:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ IN\ (POWER\ (edge\_map\ H)\ n\ x)\ (edge\ H\ x)$

thm Hypermap.lemma_edge_cycle:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ POWER\ (edge\_map\ H)\ (CARD\ (edge\ H\ x))\ x = x$

thm Hypermap.lemma_edge_split:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_edge\_split\ H\ x \longrightarrow \neg\ IN\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x)\ (edge\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x)) \wedge edge\ H\ x = HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (HOL\_Light\_Import.UNION\ (edge\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x))\ (edge\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x)))$

thm Hypermap.lemma_edge_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_edge\_merge\ H\ x \longrightarrow HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (edge\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x)) = HOL\_Light\_Import.UNION\ (edge\ H\ x)\ (edge\ H\ (node\_map\ H\ x))$

thm Hypermap.lemma_shift_non_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\_nondegenerate\ H\ x = dart\_nondegenerate\ (shift\ H)\ x$

thm Hypermap.lemma_change_node_walkup:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (is\_node\_merge\ H\ x \longrightarrow is\_edge\_merge\ (shift\ H)\ x) \wedge (is\_node\_split\ H\ x \longrightarrow is\_edge\_split\ (shift\ H)\ x)$

thm Hypermap.lemma_node_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_node\_merge\ H\ x \longrightarrow HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (node\ (node\_walkup\ H\ x)\ (face\_map\ H\ x)) = HOL\_Light\_Import.UNION\ (node\ H\ x)\ (node\ H\ (face\_map\ H\ x))$

thm Hypermap.lemma_node_split:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_node\_split\ H\ x \longrightarrow \neg\ IN\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)\ (node\ (node\_walkup\ H\ x)\ (face\_map\ H\ x)) \wedge node\ H\ x = HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (HOL\_Light\_Import.UNION\ (node\ (node\_walkup\ H\ x)\ (face\_map\ H\ x))\ (node\ (node\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x)))$

thm Hypermap.lemma_double_shift_non_degenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\_nondegenerate\ H\ x = dart\_nondegenerate\ (shift\ (shift\ H))\ x$

thm Hypermap.lemma_change_face_walkup:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (is\_face\_merge\ H\ x \longrightarrow is\_edge\_merge\ (shift\ (shift\ H))\ x) \wedge (is\_face\_split\ H\ x \longrightarrow is\_edge\_split\ (shift\ (shift\ H))\ x)$

thm Hypermap.lemma_face_merge:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_face\_merge\ H\ x \longrightarrow HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (face\ (face\_walkup\ H\ x)\ (edge\_map\ H\ x)) = HOL\_Light\_Import.UNION\ (face\ H\ x)\ (face\ H\ (edge\_map\ H\ x))$

thm Hypermap.lemma_face_split:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_face\_split\ H\ x \longrightarrow \neg\ IN\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)\ (face\ (face\_walkup\ H\ x)\ (edge\_map\ H\ x)) \wedge face\ H\ x = HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ (HOL\_Light\_Import.UNION\ (face\ (face\_walkup\ H\ x)\ (edge\_map\ H\ x))\ (face\ (face\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)))$

thm Hypermap.lemma_powers_in_component:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ j::nat.\ IN\ (POWER\ (edge\_map\ H)\ j\ x)\ (comb\_component\ H\ x) \wedge IN\ (POWER\ (node\_map\ H)\ j\ x)\ (comb\_component\ H\ x) \wedge IN\ (POWER\ (face\_map\ H)\ j\ x)\ (comb\_component\ H\ x)$

thm Hypermap.lemma_inverses_in_component:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *j*::*nat*. *IN* (*HOL_Light_Import*.*inverse* (*edge_map H*) *x*) (*comb_component H x*) ∧ *IN* (*HOL_Light_Import*.*inverse* (*node_map H*) *x*) (*comb_component H x*) ∧ *IN* (*HOL_Light_Import*.*inverse* (*face_map H*) *x*) (*comb_component H x*)

thm Hypermap.lemma_edge_subset_component:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *SUBSET* (*edge H x*) (*comb_component H x*)

thm Hypermap.lemma_node_subset_component:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *SUBSET* (*node H x*) (*comb_component H x*)

thm Hypermap.lemma_face_subset_component:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *SUBSET* (*face H x*) (*comb_component H x*)

thm Hypermap.lemma_component_identity:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN y* (*comb_component H x*) ⟶ *comb_component H x* = *comb_component H y*

thm Hypermap.lemma_walkup_first_component_eq:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) ∧ ¬ *IN x* (*comb_component H y*) ⟶ *comb_component H y* = *comb_component* (*edge_walkup H x*) *y* ∧ ¬ *IN* (*node_map H x*) (*comb_component H y*) ∧ ¬ *IN* (*HOL_Light_Import*.*inverse* (*edge_map H*) *x*) (*comb_component H y*)

thm Hypermap.lemma_walkup_second_component_eq:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) ∧ *IN y* (*dart H*) ∧ *y* ≠ *x* ∧ ¬ *IN* (*HOL_Light_Import*.*inverse* (*edge_map H*) *x*) (*comb_component* (*edge_walkup H x*) *y*) ∧ ¬ *IN* (*node_map H x*) (*comb_component* (*edge_walkup H x*) *y*) ⟶ *comb_component H y* = *comb_component* (*edge_walkup H x*) *y* ∧ ¬ *IN y* (*comb_component H x*)

thm Hypermap.lemma_walkup_components:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ⟶ *DELETE* (*set_of_components H*) (*comb_component H x*) = *DIFF* (*set_of_components* (*edge_walkup H x*)) (*INSERT* (*comb_component* (*edge_walkup H x*) (*node_map H x*)) (*INSERT* (*comb_component* (*edge_walkup H x*) (*HOL_Light_Import*.*inverse* (*edge_map H*) *x*)) *EMPTY*))

thm Hypermap.edge_degenerate_walkup_edge_map:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) ∧ *edge_map H x* = *x* ⟶ *edge_map* (*edge_walkup H x*) *y* = *edge_map H y*

thm Hypermap.node_degenerate_walkup_node_map:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge node\_map\ H\ x = x \longrightarrow node\_map\ (edge\_walkup\ H\ x)\ y = node\_map\ H\ y$

thm Hypermap.node_degenerate_walkup_edge_map:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge node\_map\ H\ x = x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ x = x \wedge edge\_map\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) = edge\_map\ H\ x \wedge (\forall y::?'a::type.\ y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ y = edge\_map\ H\ y)$

thm Hypermap.face_degenerate_walkup_face_map:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge face\_map\ H\ x = x \longrightarrow face\_map\ (edge\_walkup\ H\ x)\ y = face\_map\ H\ y$

thm Hypermap.face_degenerate_walkup_edge_map:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge face\_map\ H\ x = x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ x = x \wedge edge\_map\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) = edge\_map\ H\ x \wedge (\forall y::?'a::type.\ y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \longrightarrow edge\_map\ (edge\_walkup\ H\ x)\ y = edge\_map\ H\ y)$

thm Hypermap.edge_degenerate_walkup_first_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge edge\_map\ H\ x = x \longrightarrow node\_walkup\ H\ x = edge\_walkup\ H\ x$

thm Hypermap.edge_degenerate_walkup_second_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge edge\_map\ H\ x = x \longrightarrow face\_walkup\ H\ x = edge\_walkup\ H\ x$

thm Hypermap.edge_degenerate_walkup_third_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge edge\_map\ H\ x = x \longrightarrow node\_walkup\ H\ x = face\_walkup\ H\ x$

thm Hypermap.lemma_shift_cycle:

$\forall H::?'a::type\ hypermap.\ shift\ (shift\ (shift\ H)) = H$

thm Hypermap.lemma_eq_iff_shift_eq:

$\forall (H::?'a::type\ hypermap)\ H'::?'a::type\ hypermap.\ (H = H') = (shift\ H = shift\ H')$

thm Hypermap.lemma_degenerate_walkup_first_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_degenerate\ H\ x \longrightarrow node\_walkup\ H\ x = edge\_walkup\ H\ x$

thm Hypermap.lemma_degenerate_walkup_second_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_degenerate\ H\ x \longrightarrow face\_walkup\ H\ x = edge\_walkup\ H\ x$

thm Hypermap.lemma_degenerate_walkup_third_eq:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_degenerate\ H\ x \longrightarrow node\_walkup\ H\ x = face\_walkup\ H\ x$

thm Hypermap.component_at_isolated_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ isolated\_dart\ H\ x \longrightarrow comb\_component\ H\ x = INSERT\ x\ EMPTY$

thm Hypermap.LEMMA_CARD_DIFF:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ FINITE\ s \wedge SUBSET\ t\ s \longrightarrow CARD\ s = CARD\ (DIFF\ s\ t) + CARD\ t$

thm Hypermap.CARD_MINUS_ONE:

$\forall (s::?'a::type \Rightarrow bool)\ x::?'a::type.\ FINITE\ s \wedge IN\ x\ s \longrightarrow CARD\ s = CARD\ (DELETE\ s\ x) + (1::nat)$

thm Hypermap.CARD_MINUS_DIFF_TWO_SET:

$\forall (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ FINITE\ s \wedge IN\ x\ s \wedge IN\ y\ s \longrightarrow CARD\ s = CARD\ (DIFF\ s\ (INSERT\ x\ (INSERT\ y\ EMPTY))) + CARD\ (INSERT\ x\ (INSERT\ y\ EMPTY))$

thm Hypermap.EDGE_FINITE:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ FINITE\ (edge\ H\ x)$

thm Hypermap.EDGE_NOT_EMPTY:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (1::nat) \leq CARD\ (edge\ H\ x)$

thm Hypermap.NODE_FINITE:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ FINITE\ (node\ H\ x)$

thm Hypermap.NODE_NOT_EMPTY:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (1::nat) \leq CARD\ (node\ H\ x)$

thm Hypermap.FACE_FINITE:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ FINITE\ (face\ H\ x)$

thm Hypermap.FACE_NOT_EMPTY:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (1::nat) \leq CARD\ (face\ H\ x)$

thm Hypermap.FINITE_HYPERMAP_ORBITS:

$\forall H::?'a::type\ hypermap.\ FINITE\ (edge\_set\ H) \wedge FINITE\ (node\_set\ H) \wedge FINITE\ (face\_set\ H)$

thm Hypermap.FINITE_HYPERMAP_COMPONENTS:

$\forall H::?'a::type\ hypermap.\ FINITE\ (set\_of\_components\ H)$

thm Hypermap.WALKUP_EXCEPTION_COMPONENT:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow comb\_component\ (edge\_walkup\ H\ x)\ x = INSERT\ x\ EMPTY$

thm Hypermap.lemma_in_components:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) = IN\ (comb\_component\ H\ x)\ (set\_of\_components\ H)$

thm Hypermap.lemma_card_eq_reflect:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ s = t \longrightarrow CARD\ s = CARD\ t$

thm Hypermap.lemma_different_edges:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ \neg\ IN\ x\ (edge\ H\ y) \longrightarrow edge\ H\ x \neq edge\ H\ y$

thm Hypermap.lemma_different_nodes:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ \neg\ IN\ x\ (node\ H\ y) \longrightarrow node\ H\ x \neq node\ H\ y$

thm Hypermap.lemma_different_faces:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ \neg\ IN\ x\ (face\ H\ y) \longrightarrow face\ H\ x \neq face\ H\ y$

thm Hypermap.lemma_planar_index_on_walkup_at_isolated_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge isolated\_dart\ H\ x \longrightarrow planar\_ind\ H = planar\_ind\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_planar_index_on_walkup_at_edge_degenerate_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_edge\_degenerate\ H\ x \longrightarrow planar\_ind\ H = planar\_ind\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_planar_index_on_walkup_at_degenerate_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_degenerate\ H\ x \longrightarrow planar\_ind\ H = planar\_ind\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_card_walkup_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow CARD\ (dart\ H) = CARD\ (dart\ (edge\_walkup\ H\ x)) + (1::nat)$

thm Hypermap.lemma_splitting_case_count_edges:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_edge\_split\ H\ x \longrightarrow number\_of\_edges\ H + (1::nat) = number\_of\_edges\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_merge_case_count_edges:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge is\_edge\_merge\ H\ x \longrightarrow number\_of\_edges\ H = number\_of\_edges\ (edge\_walkup\ H\ x) + (1::nat)$

thm Hypermap.lemma_walkup_count_nodes:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \longrightarrow number\_of\_nodes\ H = number\_of\_nodes\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_walkup_count_faces:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \longrightarrow number\_of\_faces\ H = number\_of\_faces\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_walkup_count_splitting_components:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \wedge comb\_component\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x) \neq comb\_component\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) \longrightarrow number\_of\_components\ H + (1::nat) = number\_of\_components\ (edge\_walkup\ H\ x)$

thm Hypermap.lemma_walkup_count_not_splitting_components:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \wedge comb\_component\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x) = comb\_component\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) \longrightarrow number\_of\_components\ H = number\_of\_components\ (edge\_walkup\ H\ x)$

thm DEF_is_splitting_component:

$is\_splitting\_component = (\lambda(\_2410282::?'a::type\ hypermap)\ \_2410283::?'a::type.\ comb\_component\ (edge\_walkup\ \_2410282\ \_2410283)\ (node\_map\ \_2410282\ \_2410283) \neq comb\_component\ (edge\_walkup\ \_2410282\ \_2410283)\ (HOL\_Light\_Import.inverse\ (edge\_map\ \_2410282)\ \_2410283))$

thm Hypermap.is_splitting_component:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ is\_splitting\_component\ H\ x = (comb\_component\ (edge\_walkup\ H\ x)\ (node\_map\ H\ x) \neq comb\_component\ (edge\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x))$

thm Hypermap.lemma_planar_index_on_nondegenerate:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \longrightarrow (is\_edge\_split\ H\ x \wedge \neg\ is\_splitting\_component\ H\ x \longrightarrow planar\_ind\ H + real\_of\_nat\ (2::nat) = planar\_ind\ (edge\_walkup\ H\ x)) \wedge (\neg\ (is\_edge\_split\ H\ x \wedge \neg\ is\_splitting\_component\ H\ x) \longrightarrow planar\_ind\ H = planar\_ind\ (edge\_walkup\ H\ x))$

thm Hypermap.lemmaIUCLZYI:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (IN\ x\ (dart\ H) \wedge dart\_nondegenerate\ H\ x \longrightarrow (is\_edge\_split\ H\ x \longrightarrow number\_of\_edges\ H + (1::nat) = number\_of\_edges\ (edge\_walkup\ H\ x)) \wedge (is\_edge\_merge\ H\ x \longrightarrow number\_of\_edges\ H = number\_of\_edges\ (edge\_walkup\ H\ x) + (1::nat)) \wedge number\_of\_nodes\ H = number\_of\_nodes\ (edge\_walkup\ H\ x) \wedge number\_of\_faces\ H = number\_of\_faces\ (edge\_walkup\ H\ x) \wedge (is\_splitting\_component\ H\ x \longrightarrow number\_of\_components\ H + (1::nat) =$

number_of_components (*edge_walkup H x*)) ∧ (¬ *is_splitting_component H x* ⟶ *number_of_components H* = *number_of_components* (*edge_walkup H x*)) ∧ (*is_edge_split H x* ∧ ¬ *is_splitting_component H x* ⟶ *planar_ind H* + *real_of_nat* (*2*::*nat*) = *planar_ind* (*edge_walkup H x*)) ∧ (¬ (*is_edge_split H x* ∧ ¬ *is_splitting_component H x*) ⟶ *planar_ind H* = *planar_ind* (*edge_walkup H x*))) ∧ (*IN x* (*dart H*) ∧ *dart_degenerate H x* ⟶ *planar_ind H* = *planar_ind* (*edge_walkup H x*))

thm Hypermap.lemma_desc_planar_index:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ⟶ *planar_ind H* ≤ *planar_ind* (*edge_walkup H x*)

thm Hypermap.lemmaBISHKQW:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ⟶ *planar_ind H* ≤ *planar_ind* (*edge_walkup H x*) ∧ *planar_ind H* ≤ *planar_ind* (*node_walkup H x*) ∧ *planar_ind H* ≤ *planar_ind* (*face_walkup H x*)

thm Hypermap.lemmaFOAGLPA:

∀ *H*::*?'a*::*type hypermap*. *planar_ind H* ≤ (*0*::*real*)

thm Hypermap.lemmaSGCOSXK:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ∧ *planar_hypermap H* ⟶ *planar_hypermap* (*edge_walkup H x*) ∧ *planar_hypermap* (*node_walkup H x*) ∧ *planar_hypermap* (*face_walkup H x*)

thm Hypermap.convolution_rep:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *p*::*?'a*::*type* ⇒ *?'a*::*type*. *permutes p s* ⟶ (*p* ∘ *p* = *id*) = (*p* = *HOL_Light_Import.inverse p*)

thm Hypermap.convolution_inv:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *p*::*?'a*::*type* ⇒ *?'a*::*type*. *permutes p s* ⟶ (*p* ∘ *p* = *id*) = (*HOL_Light_Import.inverse p* ∘ *HOL_Light_Import.inverse p* = *id*)

thm Hypermap.convolution_belong:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *p*::*?'a*::*type* ⇒ *?'a*::*type*. *permutes p s* ⟶ (*p* ∘ *p* = *id*) = (∀ *x*::*?'a*::*type*. *IN x s* ⟶ *p* (*p x*) = *x*)

thm Hypermap.edge_convolution:

∀ *H*::*?'a*::*type hypermap*. *plain_hypermap H* = (∀ *x*::*?'a*::*type*. *IN x* (*dart H*) ⟶ *node_map H* (*face_map H* (*node_map H* (*face_map H x*))) = *x*)

thm Hypermap.edge_map_convolution:

∀ *H*::*?'a*::*type hypermap*. *plain_hypermap H* = (*edge_map H* = *node_map H* ∘ *face_map H*)

thm Hypermap.lemma_convolution_evaluation:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow (p\ (p\ x) = x) = (CARD\ (orbit\_map\ p\ x) \leq (2::nat))$

thm Hypermap.lemma_convolution_map:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow (p \circ p = id) = (\forall x::?'a::type.\ IN\ x\ s \longrightarrow CARD\ (orbit\_map\ p\ x) \leq (2::nat))$

thm Hypermap.lemma_orbit_of_size_2:

$\forall (s::?'a::type \Rightarrow bool)\ p::?'a::type \Rightarrow ?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow (CARD\ (orbit\_map\ p\ (?x::?'a::type)) = (2::nat)) = (p\ ?x \neq ?x \wedge p\ (p\ ?x) = ?x)$

thm Hypermap.EDGE_OF_SIZE_2:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (CARD\ (edge\ H\ x) = (2::nat)) = (edge\_map\ H\ x \neq x \wedge edge\_map\ H\ (edge\_map\ H\ x) = x)$

thm Hypermap.NODE_OF_SIZE_2:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (CARD\ (node\ H\ x) = (2::nat)) = (node\_map\ H\ x \neq x \wedge node\_map\ H\ (node\_map\ H\ x) = x)$

thm Hypermap.FACE_OF_SIZE_2:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ (CARD\ (face\ H\ x) = (2::nat)) = (face\_map\ H\ x \neq x \wedge face\_map\ H\ (face\_map\ H\ x) = x)$

thm Hypermap.lemma_sub_unions_diff:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ t\ s \longrightarrow UNIONS\ s = HOL\_Light\_Import.UNION\ (UNIONS\ (DIFF\ s\ t))\ (UNIONS\ t)$

thm Hypermap.lemma_card_unions_diff:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ t\ s \wedge FINITE\ (UNIONS\ s) \wedge (\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ IN\ a\ s \wedge IN\ b\ s \longrightarrow a = b \vee HOL\_Light\_Import.INTER\ a\ b = EMPTY) \longrightarrow CARD\ (UNIONS\ s) = CARD\ (UNIONS\ (DIFF\ s\ t)) + CARD\ (UNIONS\ t)$

thm Hypermap.lemma_card_partion2_unions:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge IN\ y\ (dart\ H) \longrightarrow CARD\ (dart\ H) = CARD\ (UNIONS\ (DIFF\ (edge\_set\ H)\ (INSERT\ (edge\ H\ x)\ (INSERT\ (edge\ H\ y)\ EMPTY)))) + CARD\ (UNIONS\ (INSERT\ (edge\ H\ x)\ (INSERT\ (edge\ H\ y)\ EMPTY)))$

thm Hypermap.CARD_UNION_EDGES_LE:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ CARD\ (HOL\_Light\_Import.UNION\ (edge\ H\ x)\ (edge\ H\ y)) \leq CARD\ (edge\ H\ x) + CARD\ (edge\ H\ y)$

thm Hypermap.lemma_card_partion2_unions_approx:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge IN\ y\ (dart\ H) \longrightarrow CARD\ (dart\ H) \leq CARD\ (UNIONS\ (DIFF\ (edge\_set\ H)\ (INSERT$

(*edge H x*) (*INSERT* (*edge H y*) *EMPTY*)))) + (*CARD* (*edge H x*) + *CARD* (*edge H y*))

thm Hypermap.lemma_card_partion2_unions_eq:

∀ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) ∧ *IN y* (*dart H*) ∧ *edge H x* ≠ *edge H y* ⟶ *CARD* (*dart H*) = *CARD* (*UNIONS* (*DIFF* (*edge_set H*) (*INSERT* (*edge H x*) (*INSERT* (*edge H y*) *EMPTY*)))) + (*CARD* (*edge H x*) + *CARD* (*edge H y*))

thm Hypermap.lemma_card_partion1_unions_eq:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ⟶ *CARD* (*dart H*) = *CARD* (*UNIONS* (*DELETE* (*edge_set H*) (*edge H x*))) + *CARD* (*edge H x*)

thm Hypermap.lemma_permutes_exception:

∀ (*s*::*?'a*::*type* ⇒ *bool*) (*p*::*?'a*::*type* ⇒ *?'a*::*type*) *x*::*?'a*::*type*. *permutes p s* ∧ ¬ *IN x s* ⟶ *p x* = *x*

thm Hypermap.map_permutes_outside_domain:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *p*::*?'a*::*type* ⇒ *?'a*::*type*. *permutes p s* ⟶ (∀ *x*::*?'a*::*type*. ¬ *IN x s* ⟶ *p x* = *x*)

thm Hypermap.power_permutation_outside_domain:

∀ (*s*::*?'a*::*type* ⇒ *bool*) (*p*::*?'a*::*type* ⇒ *?'a*::*type*) (*x*::*?'a*::*type*) *n*::*nat*. *permutes p s* ∧ ¬ *IN x s* ⟶ *POWER p n x* = *x*

thm Hypermap.lemma_edge_exception:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. ¬ *IN x* (*dart H*) ⟶ *edge H x* = *INSERT x EMPTY*

thm Hypermap.lemma_node_exception:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. ¬ *IN x* (*dart H*) ⟶ *node H x* = *INSERT x EMPTY*

thm Hypermap.lemma_face_exception:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. ¬ *IN x* (*dart H*) ⟶ *face H x* = *INSERT x EMPTY*

thm Hypermap.lemma_simple_hypermap:

simple_hypermap (*?H*::*?'a*::*type hypermap*) ⟶ (∀ *x*::*?'a*::*type*. *HOL_Light_Import.INTER* (*node ?H x*) (*face ?H x*) = *INSERT x EMPTY*)

thm Hypermap.double_edge_walkup_plain_hypermap:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) ∧ *plain_hypermap H* ∧ *CARD* (*node H x*) = (*2*::*nat*) ⟶ *plain_hypermap* (*double_edge_walkup H x* (*node_map H x*))

thm Hypermap.lemma_representaion_Wn:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ double\_node\_walkup\ H\ x\ y = shift\ (shift\ (double\_edge\_walkup\ (shift\ H)\ x\ y))$

thm Hypermap.lemma_representaion_Wf:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ double\_face\_walkup\ H\ x\ y = shift\ (double\_edge\_walkup\ (shift\ (shift\ H))\ x\ y)$

thm Hypermap.double_node_walkup_plain_hypermap:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \land plain\_hypermap\ H \land CARD\ (edge\ H\ x) = (2::nat) \longrightarrow plain\_hypermap\ (double\_node\_walkup\ H\ x\ (edge\_map\ H\ x))$

thm Hypermap.double_face_walkup_plain_hypermap:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \land plain\_hypermap\ H \land CARD\ (edge\ H\ x) = (2::nat) \longrightarrow plain\_hypermap\ (double\_face\_walkup\ H\ x\ (edge\_map\ H\ x))$

thm Hypermap.lemmaHOZKXVW:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \land plain\_hypermap\ H \longrightarrow (CARD\ (edge\ H\ x) = (2::nat) \longrightarrow plain\_hypermap\ (double\_face\_walkup\ H\ x\ (edge\_map\ H\ x)) \land plain\_hypermap\ (double\_node\_walkup\ H\ x\ (edge\_map\ H\ x))) \land (CARD\ (node\ H\ x) = (2::nat) \longrightarrow plain\_hypermap\ (double\_edge\_walkup\ H\ x\ (node\_map\ H\ x)))$

thm DEF_is_Moebius_contour:

$is\_Moebius\_contour = (\lambda(\_2411155::?'a::type\ hypermap)\ (\_2411156::nat \Rightarrow ?'a::type)\ \_2411157::nat.\ is\_inj\_contour\ \_2411155\ \_2411156\ \_2411157 \land (\exists (i::nat)\ j::nat.\ (0::nat) < i \land i \leq j \land j < \_2411157 \land \_2411156\ j = node\_map\ \_2411155\ (\_2411156\ (0::nat)) \land \_2411156\ \_2411157 = node\_map\ \_2411155\ (\_2411156\ i)))$

thm Hypermap.is_Moebius_contour:

$\forall (k::nat)\ (H::?'a::type\ hypermap)\ p::nat \Rightarrow ?'a::type.\ is\_Moebius\_contour\ H\ p\ k = (is\_inj\_contour\ H\ p\ k \land (\exists (i::nat)\ j::nat.\ (0::nat) < i \land i \leq j \land j < k \land p\ j = node\_map\ H\ (p\ (0::nat)) \land p\ k = node\_map\ H\ (p\ i)))$

thm Hypermap.lemma_contour_in_dart:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ IN\ (p\ (0::nat))\ (dart\ H) \land is\_contour\ H\ p\ n \longrightarrow IN\ (p\ n)\ (dart\ H)$

thm Hypermap.lemma_darts_in_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ IN\ (p\ (0::nat))\ (dart\ H) \land is\_contour\ H\ p\ n \longrightarrow SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%253::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%253\ (i \leq n)\ (p\ i)))\ (dart\ H)$

thm Hypermap.lemma_first_dart_on_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ (0::nat) < n \wedge is\_inj\_contour\ H\ p\ n \longrightarrow IN\ (p\ (0::nat))\ (dart\ H)$

thm Hypermap.lemma_darts_on_Moebius_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ k::nat.\ is\_Moebius\_contour\ H\ p\ k \longrightarrow (2::nat) \leq k \wedge IN\ (p\ (0::nat))\ (dart\ H) \wedge Suc\ k \leq CARD\ (dart\ H)$

thm Hypermap.lemma_Moebius_contour_points_subset_darts:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ k::nat.\ is\_Moebius\_contour\ H\ p\ k \longrightarrow SUBSET\ (GSPEC\ (\lambda GEN\%PVAR\%254::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%254\ (i \leq k)\ (p\ i)))\ (dart\ H) \wedge CARD\ (GSPEC\ (\lambda GEN\%PVAR\%255::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%255\ (i \leq k)\ (p\ i))) = Suc\ k$

thm Hypermap.lemma_darts_is_Moebius_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ k::nat.\ is\_Moebius\_contour\ H\ p\ k \wedge Suc\ k = CARD\ (dart\ H) \longrightarrow dart\ H = GSPEC\ (\lambda GEN\%PVAR\%256::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%256\ (i \leq k)\ (p\ i))$

thm Hypermap.lemma_point_in_list:

$\forall (p::nat \Rightarrow ?'a::type)\ (k::nat)\ x::?'a::type.\ IN\ x\ (GSPEC\ (\lambda GEN\%PVAR\%257::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%257\ (i \leq k)\ (p\ i))) = (\exists j \leq k.\ x = p\ j)$

thm Hypermap.lemma_point_not_in_list:

$\forall (p::nat \Rightarrow ?'a::type)\ (k::nat)\ x::?'a::type.\ (\neg\ IN\ x\ (GSPEC\ (\lambda GEN\%PVAR\%258::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%258\ (i \leq k)\ (p\ i)))) = (\forall j \leq k.\ x \neq p\ j)$

thm Hypermap.lemma_eliminate_dart_ouside_Moebius_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (k::nat)\ x::?'a::type.\ is\_Moebius\_contour\ H\ p\ k \wedge \neg\ IN\ x\ (GSPEC\ (\lambda GEN\%PVAR\%259::?'a::type.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%259\ (i \leq k)\ (p\ i))) \longrightarrow is\_Moebius\_contour\ (edge\_walkup\ H\ x)\ p\ k$

thm DEF_shift_path:

$shift\_path = (\lambda(\_2411280::nat \Rightarrow ?'a::type)\ (\_2411281::nat)\ i::nat.\ \_2411280\ (\_2411281 + i))$

thm Hypermap.shift_path:

$\forall (p::nat \Rightarrow ?'a::type)\ l::nat.\ shift\_path\ p\ l = (\lambda i::nat.\ p\ (l + i))$

thm Hypermap.lemma_shift_path_evaluation:

$\forall (p::nat \Rightarrow ?'a::type)\ (l::nat)\ i::nat.\ shift\_path\ p\ l\ i = p\ (l + i)$

thm Hypermap.lemma_shift_path:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (n::nat)\ l::nat.\ is\_path\ H\ p\ n \wedge l \leq n \longrightarrow is\_path\ H\ (shift\_path\ p\ l)\ (n - l)$

thm Hypermap.lemma_shift_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (n::nat)\ l::nat.\ is\_contour\ H\ p\ n \wedge l \leq n \longrightarrow is\_contour\ H\ (shift\_path\ p\ l)\ (n - l)$

thm Hypermap.lemma_shift_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (n::nat)\ l::nat.\ is\_inj\_contour\ H\ p\ n \wedge l \leq n \longrightarrow is\_inj\_contour\ H\ (shift\_path\ p\ l)\ (n - l)$

thm Hypermap.lemma_join_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_contour\ H\ p\ n \wedge is\_contour\ H\ q\ m \wedge one\_step\_contour\ H\ (p\ n)\ (q\ (0::nat)) \longrightarrow is\_contour\ H\ (join\ p\ q\ n)\ (n + (m + (1::nat)))$

thm Hypermap.lemma_inj_contour_via_list:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n = (is\_contour\ H\ p\ n \wedge is\_inj\_list\ p\ n)$

thm Hypermap.lemma_join_inj_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ n \wedge is\_inj\_contour\ H\ q\ m \wedge one\_step\_contour\ H\ (p\ n)\ (q\ (0::nat)) \wedge is\_disjoint\ p\ q\ n\ m \longrightarrow is\_inj\_contour\ H\ (join\ p\ q\ n)\ (n + (m + (1::nat)))$

thm Hypermap.lemma_glue_inj_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ n \wedge is\_inj\_contour\ H\ q\ m \wedge is\_glueing\ p\ q\ n\ m \longrightarrow is\_inj\_contour\ H\ (glue\ p\ q\ n)\ (n + m)$

thm Hypermap.concatenate_two_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ n \wedge is\_inj\_contour\ H\ q\ m \wedge p\ n = q\ (0::nat) \wedge (\forall j::nat.\ (0::nat) < j \wedge j \leq m \longrightarrow (\forall i \leq n.\ q\ j \neq p\ i)) \longrightarrow (\exists g::nat \Rightarrow ?'a::type.\ g\ (0::nat) = p\ (0::nat) \wedge g\ (n + m) = q\ m \wedge is\_inj\_contour\ H\ g\ (n + m) \wedge (\forall i \leq n.\ g\ i = p\ i) \wedge (\forall i \leq m.\ g\ (n + i) = q\ i))$

thm Hypermap.concatenate_two_disjoint_contours:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ (n::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ n \wedge is\_inj\_contour\ H\ q\ m \wedge one\_step\_contour\ H\ (p\ n)\ (q\ (0::nat)) \wedge (\forall (i::nat)\ j::nat.\ i \leq n \wedge j \leq m \longrightarrow q\ j \neq p\ i) \longrightarrow (\exists g::nat \Rightarrow ?'a::type.\ g\ (0::nat) = p\ (0::nat) \wedge g\ (n + (m + (1::nat))) = q\ m \wedge is\_inj\_contour\ H\ g\ (n + (m + (1::nat))) \wedge (\forall i \leq n.\ g\ i = p\ i) \wedge (\forall i \leq m.\ g\ (n + (i + (1::nat))) = q\ i))$

thm Hypermap.lemmaQZTPGJV:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_contour\ H\ p\ n \longrightarrow (\exists (q::nat \Rightarrow ?'a::type)\ m::nat.\ m \leq n \wedge q\ (0::nat) = p\ (0::nat) \wedge q\ m = p\ n$

$\wedge$ *is_inj_contour H q m* $\wedge$ ($\forall$ *i*<*m*. $\exists$ *j*$\geq$*i*. *j* < *n* $\wedge$ *q i* = *p j* $\wedge$ *q* (*Suc i*) = *p* (*Suc j*)))

thm Hypermap.lemma_one_step_contour:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *one_step_contour H x y* = (*y* = *face_map H x* $\vee$ *x* = *node_map H y*)

thm Hypermap.lemma_only_one_orbit:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) (*p*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) *x*::*?'a*::*type*. *FINITE s* $\wedge$ *permutes p s* $\wedge$ *orbit_map p x* = *s* $\longrightarrow$ *set_of_orbits s p* = *INSERT* (*orbit_map p x*) *EMPTY*

thm Hypermap.lemma_atmost_two_orbits:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) (*p*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *FINITE s* $\wedge$ *permutes p s* $\wedge$ *SUBSET s* (*HOL_Light_Import.UNION* (*orbit_map p x*) (*orbit_map p y*)) $\longrightarrow$ *number_of_orbits s p* $\leq$ (*2*::*nat*)

thm Hypermap.lemma_only_one_component:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *comb_component H x* = *dart H* $\longrightarrow$ *set_of_components H* = *INSERT* (*comb_component H x*) *EMPTY*

thm Hypermap.lemma_minimum_Moebius_hypermap:

$\forall$ *H*::*?'a*::*type hypermap*. *CARD* (*dart H*) = (*3*::*nat*) $\wedge$ ($\exists$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *k*::*nat*. *is_Moebius_contour H p k*) $\longrightarrow$ $\neg$ *planar_hypermap H*

thm Hypermap.dart_face_walkup:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *dart* (*face_walkup H x*) = *DELETE* (*dart H*) *x*

thm Hypermap.lemma_card_face_walkup_dart:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *IN x* (*dart H*) $\longrightarrow$ *CARD* (*dart H*) = *CARD* (*dart* (*face_walkup H x*)) + (*1*::*nat*)

thm Hypermap.face_map_face_walkup:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *face_map* (*face_walkup H x*) *x* = *x* $\wedge$ (*edge_map H x* $\neq$ *x* $\wedge$ *face_map H x* $\neq$ *x* $\longrightarrow$ *face_map* (*face_walkup H x*) (*edge_map H x*) = *face_map H x*) $\wedge$ (*HOL_Light_Import.inverse* (*node_map H*) *x* $\neq$ *x* $\wedge$ *HOL_Light_Import.inverse* (*face_map H*) *x* $\neq$ *x* $\longrightarrow$ *face_map* (*face_walkup H x*) (*HOL_Light_Import.inverse* (*face_map H*) *x*) = *HOL_Light_Import.inverse* (*node_map H*) *x*) $\wedge$ (*y* $\neq$ *x* $\wedge$ *y* $\neq$ *HOL_Light_Import.inverse* (*face_map H*) *x* $\wedge$ *y* $\neq$ *edge_map H x* $\longrightarrow$ *face_map* (*face_walkup H x*) *y* = *face_map H y*)

thm Hypermap.node_map_face_walkup:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *node_map* (*face_walkup H x*) *x* = *x* $\wedge$ *node_map* (*face_walkup H x*) (*HOL_Light_Import.inverse* (*node_map*

H) x) = $node\_map\ H\ x \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ x \longrightarrow node\_map\ (face\_walkup\ H\ x)\ y = node\_map\ H\ y)$

thm Hypermap.dart_node_walkup:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ dart\ (node\_walkup\ H\ x) = DELETE\ (dart\ H)\ x$

thm Hypermap.lemma_card_node_walkup_dart:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow CARD\ (dart\ H) = CARD\ (dart\ (node\_walkup\ H\ x)) + (1::nat)$

thm Hypermap.node_map_node_walkup:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ node\_map\ (node\_walkup\ H\ x)\ x = x \wedge (face\_map\ H\ x \neq x \wedge node\_map\ H\ x \neq x \longrightarrow node\_map\ (node\_walkup\ H\ x)\ (face\_map\ H\ x) = node\_map\ H\ x) \wedge (HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x \neq x \wedge HOL\_Light\_Import.inverse\ (node\_map\ H)\ x \neq x \longrightarrow node\_map\ (node\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ x) = HOL\_Light\_Import.inverse\ (edge\_map\ H)\ x) \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ x \wedge y \neq face\_map\ H\ x \longrightarrow node\_map\ (node\_walkup\ H\ x)\ y = node\_map\ H\ y)$

thm Hypermap.face_map_node_walkup:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ face\_map\ (node\_walkup\ H\ x)\ x = x \wedge face\_map\ (node\_walkup\ H\ x)\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ x) = face\_map\ H\ x \wedge (y \neq x \wedge y \neq HOL\_Light\_Import.inverse\ (face\_map\ H)\ x \longrightarrow face\_map\ (node\_walkup\ H\ x)\ y = face\_map\ H\ y)$

thm Hypermap.lemma_face_walkup_second_segment_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (k::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ k \wedge m < k \wedge node\_map\ H\ (p\ (m + (1::nat))) = p\ m \longrightarrow is\_inj\_contour\ (face\_walkup\ H\ (p\ m))\ (shift\_path\ p\ (m + (1::nat)))\ (k - (m + (1::nat)))$

thm Hypermap.lemma_face_walkup_eliminate_dart_on_Moebius_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (k::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ k \wedge (0::nat) < m \wedge m < k \wedge node\_map\ H\ (p\ (m + (1::nat))) = p\ m \longrightarrow is\_inj\_contour\ (face\_walkup\ H\ (p\ m))\ p\ (m - (1::nat)) \wedge is\_inj\_contour\ (face\_walkup\ H\ (p\ m))\ (shift\_path\ p\ (m + (1::nat)))\ (k - m - (1::nat)) \wedge one\_step\_contour\ (face\_walkup\ H\ (p\ m))\ (p\ (m - (1::nat)))\ (p\ (m + (1::nat)))$

thm Hypermap.lemma_node_walkup_second_segment_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (k::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ k \wedge m < k \wedge p\ (m + (1::nat)) = face\_map\ H\ (p\ m) \longrightarrow is\_inj\_contour\ (node\_walkup\ H\ (p\ m))\ (shift\_path\ p\ (m + (1::nat)))\ (k - (m + (1::nat)))$

thm Hypermap.lemma_node_walkup_eliminate_dart_on_Moebius_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ (k::nat)\ m::nat.\ is\_inj\_contour\ H\ p\ k \wedge (0::nat) < m \wedge m < k \wedge p\ (m + (1::nat)) = face\_map\ H\ (p\ m)$

$\longrightarrow$ *is_inj_contour* (*node_walkup H* (*p m*)) *p* (*m* − (*1*::*nat*)) ∧ *is_inj_contour* (*node_walkup H* (*p m*)) (*shift_path p* (*m* + (*1*::*nat*))) (*k* − *m* − (*1*::*nat*)) ∧ *one_step_contour* (*node_walkup H* (*p m*)) (*p* (*m* − (*1*::*nat*))) (*p* (*m* + (*1*::*nat*)))

thm Hypermap.lemmaLIPYTUI:

∀ *H*::*?'a*::*type hypermap*. *planar_hypermap H* ⟶ ¬ (∃ (*p*::*nat* ⇒ *?'a*::*type*) *k*::*nat*. *is_Moebius_contour H p k*)

thm Hypermap.exist_loop:

∃ *L*::(*?'a*::*type* ⇒ *bool*) × (*?'a*::*type* ⇒ *?'a*::*type*). *FINITE* (*fst L*) ∧ *permutes* (*snd L*) (*fst L*) ∧ (∃ *x*::*?'a*::*type*. *IN x* (*fst L*) ∧ *orbit_map* (*snd L*) *x* = *fst L*)

thm TYDEF_loop:

loop (*tuple_loop* (*?a*::*?'a*::*type loop*)) = *?a* ∧ (*FINITE* (*fst* (*?r*::(*?'a*::*type* ⇒ *bool*) × (*?'a*::*type* ⇒ *?'a*::*type*))) ∧ *permutes* (*snd ?r*) (*fst ?r*) ∧ (∃ *x*::*?'a*::*type*. *IN x* (*fst ?r*) ∧ *orbit_map* (*snd ?r*) *x* = *fst ?r*)) = (*tuple_loop* (*loop ?r*) = *?r*)

thm Hypermap.loop_tybij_conjunct1:

∀ *r*::(*?'a*::*type* ⇒ *bool*) × (*?'a*::*type* ⇒ *?'a*::*type*). (*FINITE* (*fst r*) ∧ *permutes* (*snd r*) (*fst r*) ∧ (∃ *x*::*?'a*::*type*. *IN x* (*fst r*) ∧ *orbit_map* (*snd r*) *x* = *fst r*)) = (*tuple_loop* (*loop r*) = *r*)

thm Hypermap.loop_tybij_conjunct0:

∀ *a*::*?'a*::*type loop*. *loop* (*tuple_loop a*) = *a*

thm Hypermap.loop_tybij:

(∀ *a*::*?'a*::*type loop*. *loop* (*tuple_loop a*) = *a*) ∧ (∀ *r*::(*?'a*::*type* ⇒ *bool*) × (*?'a*::*type* ⇒ *?'a*::*type*). (*FINITE* (*fst r*) ∧ *permutes* (*snd r*) (*fst r*) ∧ (∃ *x*::*?'a*::*type*. *IN x* (*fst r*) ∧ *orbit_map* (*snd r*) *x* = *fst r*)) = (*tuple_loop* (*loop r*) = *r*))

thm DEF_dart_of:

dart_of = (λ*_2413003*::*?'a*::*type loop*. *fst* (*tuple_loop _2413003*))

thm Hypermap.dart_of:

∀ *L*::*?'a*::*type loop*. *dart_of L* = *fst* (*tuple_loop L*)

thm DEF_next:

next = (λ*_2413008*::*?'a*::*type loop*. *snd* (*tuple_loop _2413008*))

thm Hypermap.next:

∀ *L*::*?'a*::*type loop*. *next L* = *snd* (*tuple_loop L*)

thm DEF_back:

back = (λ*_2413013*::*?'a*::*type loop*. *HOL_Light_Import.inverse* (*snd* (*tuple_loop _2413013*)))

thm Hypermap.back:

$\forall L::?'a::type\ loop.\ back\ L = HOL\_Light\_Import.inverse\ (snd\ (tuple\_loop\ L))$

thm DEF_belong:

$belong = (\lambda(\_2413018::?'a::type)\ \_2413019::?'a::type\ loop.\ IN\ \_2413018\ (dart\_of\ \_2413019))$

thm Hypermap.belong:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L = IN\ x\ (dart\_of\ L)$

thm DEF_size:

$HOL\_Light\_Import.size = (\lambda\_2413030::?'a::type\ loop.\ CARD\ (dart\_of\ \_2413030))$

thm Hypermap.size:

$\forall L::?'a::type\ loop.\ HOL\_Light\_Import.size\ L = CARD\ (dart\_of\ L)$

thm DEF_top:

$HOL\_Light\_Import.top = (\lambda\_2413035::?'a::type\ loop.\ pred\ (CARD\ (dart\_of\ \_2413035)))$

thm Hypermap.top:

$\forall L::?'a::type\ loop.\ HOL\_Light\_Import.top\ L = pred\ (CARD\ (dart\_of\ L))$

thm DEF_is_loop:

$is\_loop = (\lambda(\_2413040::?'a::type\ hypermap)\ \_2413041::?'a::type\ loop.\ \forall x::?'a::type.\ belong\ x\ \_2413041 \longrightarrow one\_step\_contour\ \_2413040\ x\ (next\ \_2413041\ x))$

thm Hypermap.is_loop:

$\forall (H::?'a::type\ hypermap)\ L::?'a::type\ loop.\ is\_loop\ H\ L = (\forall x::?'a::type.\ belong\ x\ L \longrightarrow one\_step\_contour\ H\ x\ (next\ L\ x))$

thm DEF_loop_path:

$loop\_path = (\lambda(\_2413052::?'a::type\ loop)\ (\_2413053::?'a::type)\ \_2413054::nat.\ POWER\ (next\ \_2413052)\ \_2413054\ \_2413053)$

thm Hypermap.loop_path:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ k::nat.\ loop\_path\ L\ x\ k = POWER\ (next\ L)\ k\ x$

thm Hypermap.lemma_loop_path_via_list:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ loop\_path\ L\ x = power\_list\ (next\ L)\ x$

thm Hypermap.loop_lemma:

$\forall L::?'a::type\ loop.\ FINITE\ (dart\_of\ L) \wedge permutes\ (next\ L)\ (dart\_of\ L) \wedge (\exists x::?'a::type.\ belong\ x\ L \wedge orbit\_map\ (next\ L)\ x = dart\_of\ L)$

thm Hypermap.lemma_loop_representation:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \wedge orbit\_map\ p\ x = s \longrightarrow dart\_of\ (loop\ (s,\ p)) = s \wedge next\ (loop\ (s,\ p)) = p$

thm Hypermap.lemma_loop_identity:

$\forall (L::?'a::type\ loop)\ L'::?'a::type\ loop.\ (L = L') = (dart\_of\ L = dart\_of\ L' \wedge next\ L = next\ L')$

thm Hypermap.lemma_permute_loop:

$\forall L::?'a::type\ loop.\ permutes\ (next\ L)\ (dart\_of\ L) \wedge permutes\ (back\ L)\ (dart\_of\ L)$

thm Hypermap.lemma_transitive_permutation:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow dart\_of\ L = orbit\_map\ (next\ L)\ x$

thm Hypermap.lemma_size:

$\forall L::?'a::type\ loop.\ dart\_of\ L \neq EMPTY \wedge (0::nat) < HOL\_Light\_Import.size\ L \wedge HOL\_Light\_Import.size\ L = Suc\ (HOL\_Light\_Import.top\ L)$

thm Hypermap.lemma_order_next:

$\forall L::?'a::type\ loop.\ POWER\ (next\ L)\ (HOL\_Light\_Import.size\ L) = id$

thm Hypermap.lemma_congruence_on_loop:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ (n::nat)\ m::nat.\ belong\ x\ L \wedge n \leq HOL\_Light\_Import.top\ L \wedge POWER\ (next\ L)\ n\ x = POWER\ (next\ L)\ m\ x \longrightarrow (\exists q::nat.\ m = q * HOL\_Light\_Import.size\ L + n)$

thm Hypermap.lemma_back_and_next_outside_loop:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ \neg\ belong\ x\ L \longrightarrow back\ L\ x = x \wedge next\ L\ x = x$

thm Hypermap.lemma_power_back_and_next_outside_loop:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ m::nat.\ \neg\ belong\ x\ L \longrightarrow POWER\ (back\ L)\ m\ x = x \wedge POWER\ (next\ L)\ m\ x = x$

thm Hypermap.lemma_inverse_on_loop:

$\forall L::?'a::type\ loop.\ next\ L = HOL\_Light\_Import.inverse\ (back\ L) \wedge back\ L = HOL\_Light\_Import.inverse\ (next\ L)$

thm Hypermap.lemma_inverse_evaluation:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ back\ L\ (next\ L\ x) = x \wedge next\ L\ (back\ L\ x) = x$

thm Hypermap.lemma_second_inverse_on_loop:

$\forall (L::?'a::type\ loop)\ m::nat.\ POWER\ (next\ L)\ m = HOL\_Light\_Import.inverse\ (POWER\ (back\ L)\ m) \wedge POWER\ (back\ L)\ m = HOL\_Light\_Import.inverse\ (POWER\ (next\ L)\ m)$

thm Hypermap.lemma_second_inverse_evaluation:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ m::nat.\ POWER\ (next\ L)\ m\ (POWER\ (back\ L)\ m\ x) = x \wedge POWER\ (back\ L)\ m\ (POWER\ (next\ L)\ m\ x) = x$

thm Hypermap.lemma_next_power_representation:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ belong\ x\ L \wedge belong\ y\ L \longrightarrow (\exists k \le HOL\_Light\_Import.top\ L.\ y = POWER\ (next\ L)\ k\ x)$

thm DEF_index:

$index = (SOME\ k::nat \Rightarrow ?'a::type\ loop \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow nat.\ \forall (\_2413168::nat)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ belong\ x\ L \wedge belong\ y\ L \longrightarrow k\ \_2413168\ L\ x\ y \le HOL\_Light\_Import.top\ L \wedge y = POWER\ (next\ L)\ (k\ \_2413168\ L\ x\ y)\ x)\ (103::nat)$

thm Hypermap.lemma_loop_index:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ belong\ x\ L \wedge belong\ y\ L \longrightarrow index\ L\ x\ y \le HOL\_Light\_Import.top\ L \wedge y = POWER\ (next\ L)\ (index\ L\ x\ y)\ x$

thm Hypermap.lemma_power_next_in_loop:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ k::nat.\ belong\ x\ L \longrightarrow belong\ (POWER\ (next\ L)\ k\ x)\ L$

thm Hypermap.lemma_belong_loop:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow (\forall y::?'a::type.\ belong\ y\ L = (\exists i \le HOL\_Light\_Import.top\ L.\ y = POWER\ (next\ L)\ i\ x))$

thm Hypermap.lemma_next_in_loop:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow belong\ (next\ L\ x)\ L$

thm Hypermap.lemma_power_back_in_loop:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ k::nat.\ belong\ x\ L \longrightarrow belong\ (POWER\ (back\ L)\ k\ x)\ L$

thm Hypermap.lemma_back_in_loop:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow belong\ (back\ L\ x)\ L$

thm Hypermap.determine_loop_index:

$\forall (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ k::nat.\ belong\ x\ L \wedge k \le HOL\_Light\_Import.top\ L \wedge y = POWER\ (next\ L)\ k\ x \longrightarrow index\ L\ x\ y = k$

thm Hypermap.support_loop_sub_dart:

$\forall$ (*H*::*?'a*::*type hypermap*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_loop H L* $\wedge$ *IN x* (*dart H*) $\wedge$ *belong x L* $\longrightarrow$ *SUBSET* (*dart_of L*) (*dart H*)

thm Hypermap.lemma_loop_contour:

$\forall$ (*H*::*?'a*::*type hypermap*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *n*::*nat*. *is_loop H L* $\wedge$ *belong x L* $\longrightarrow$ *is_contour H* (*loop_path L x*) *n*

thm Hypermap.lemma_inj_loop_path:

$\forall$ (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *belong x L* $\longrightarrow$ ($\forall$ *n*::*nat*. (*n* $\leq$ *HOL_Light_Import.top L*) = *is_inj_list* (*loop_path L x*) *n*)

thm Hypermap.let_order_for_loop:

$\forall$ (*H*::*?'a*::*type hypermap*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_loop H L* $\wedge$ *belong x L* $\longrightarrow$ *is_inj_contour H* (*loop_path L x*) (*HOL_Light_Import.top L*) $\wedge$ *one_step_contour H* (*loop_path L x* (*HOL_Light_Import.top L*)) (*loop_path L x* (*0*::*nat*))

thm Hypermap.lemma_list_next:

$\forall$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. $\exists$ *h*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*. $\forall$ *x*::*?'a*::*type*. ($\neg$ *in_list p n x* $\longrightarrow$ *h x* = *x*) $\wedge$ (*in_list p n x* $\longrightarrow$ ($\exists$ *j*$\leq$*n*. *x* = *p j* $\wedge$ *h x* = *p* (*Suc j mod Suc n*)))

thm DEF_samsara:

samsara = (*SOME h*::*nat* $\Rightarrow$ (*nat* $\Rightarrow$ *?'a*::*type*) $\Rightarrow$ *nat* $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *?'a*::*type*. $\forall$ (*_2413256*::*nat*) (*p*::*nat* $\Rightarrow$ *?'a*::*type*) (*n*::*nat*) *x*::*?'a*::*type*. ($\neg$ *in_list p n x* $\longrightarrow$ *h _2413256 p n x* = *x*) $\wedge$ (*in_list p n x* $\longrightarrow$ ($\exists$ *j*$\leq$*n*. *x* = *p j* $\wedge$ *h _2413256 p n x* = *p* (*Suc j mod Suc n*)))) (*104*::*nat*)

thm Hypermap.lemma_samsara:

$\forall$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) (*n*::*nat*) *x*::*?'a*::*type*. ($\neg$ *in_list p n x* $\longrightarrow$ *samsara p n x* = *x*) $\wedge$ (*in_list p n x* $\longrightarrow$ ($\exists$ *j*$\leq$*n*. *x* = *p j* $\wedge$ *samsara p n x* = *p* (*Suc j mod Suc n*)))

thm Hypermap.samsara_formula:

$\forall$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. *is_inj_list p n* $\longrightarrow$ ($\forall$ *j*$\leq$*n*. *samsara p n* (*p j*) = *p* (*Suc j mod Suc n*))

thm Hypermap.evaluation_samsara:

$\forall$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. *is_inj_list p n* $\longrightarrow$ *samsara p n* (*p n*) = *p* (*0*::*nat*) $\wedge$ ($\forall$ *j*<*n*. *samsara p n* (*p j*) = *p* (*Suc j*))

thm Hypermap.lemma_permutes_via_surjetive:

$\forall$ (*s*::*?'a*::*type* $\Rightarrow$ *bool*) *p*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*. *FINITE s* $\wedge$ ($\forall$ *x*::*?'a*::*type*. $\neg$ *IN x s* $\longrightarrow$ *p x* = *x*) $\wedge$ ($\forall$ *x*::*?'a*::*type*. *IN x s* $\longrightarrow$ *IN* (*p x*) *s*) $\wedge$ ($\forall$ *y*::*?'a*::*type*. *IN y s* $\longrightarrow$ ($\exists$ *x*::*?'a*::*type*. *p x* = *y*)) $\longrightarrow$ *permutes p s*

thm Hypermap.lemma_back_index:

$\forall (n::nat)\ i::nat.\ (0::nat) < i \wedge i \leq n \longrightarrow (i + n)\ mod\ Suc\ n = pred\ i$

thm Hypermap.lemma_suc_mod:

$\forall (m::nat)\ n::nat.\ n \neq (0::nat) \longrightarrow Suc\ (m\ mod\ n)\ mod\ n = Suc\ m\ mod\ n$

thm Hypermap.lemma_from_index:

$\forall (n::nat)\ j::nat.\ j \leq n \longrightarrow Suc\ ((j + n)\ mod\ Suc\ n)\ mod\ Suc\ n = j$

thm Hypermap.lemma_from_index2:

$\forall (n::nat)\ i::nat.\ i \leq n \longrightarrow (Suc\ i\ mod\ Suc\ n + n)\ mod\ Suc\ n = i$

thm Hypermap.lemma_samsara_permute:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n \longrightarrow permutes\ (samsara\ p\ n)\ (support\_list\ p\ n)$

thm Hypermap.lemma_samsara_power:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n \longrightarrow POWER\ (samsara\ p\ n)\ (Suc\ n)\ (p\ (0::nat)) = p\ (0::nat) \wedge (\forall j \leq n.\ POWER\ (samsara\ p\ n)\ j\ (p\ (0::nat)) = p\ j)$

thm Hypermap.lemma_generate_loop:

$\forall (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_list\ p\ n \longrightarrow dart\_of\ (loop\ (support\_list\ p\ n,\ samsara\ p\ n)) = support\_list\ p\ n \wedge next\ (loop\ (support\_list\ p\ n,\ samsara\ p\ n)) = samsara\ p\ n$

thm Hypermap.lemma_make_contour_loop:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n \wedge one\_step\_contour\ H\ (p\ n)\ (p\ (0::nat)) \longrightarrow is\_loop\ H\ (loop\ (support\_list\ p\ n,\ samsara\ p\ n))$

thm Hypermap.lemma_number_darts_of_inj_contour:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ is\_inj\_contour\ H\ p\ n \longrightarrow CARD\ (support\_list\ p\ n) = Suc\ n$

thm Hypermap.lemma_inj_contour_belong_darts:

$\forall (H::?'a::type\ hypermap)\ (p::nat \Rightarrow ?'a::type)\ n::nat.\ (0::nat) < n \wedge is\_inj\_contour\ H\ p\ n \longrightarrow SUBSET\ (support\_list\ p\ n)\ (dart\ H)$

thm Hypermap.lemma_dart_loop_via_path:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow dart\_of\ L = support\_list\ (loop\_path\ L\ x)\ (HOL\_Light\_Import.top\ L)$

thm Hypermap.lemma_belong:

$\forall (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow (\forall y::?'a::type.\ belong\ y\ L = in\_list\ (loop\_path\ L\ x)\ (HOL\_Light\_Import.top\ L)\ y)$

thm Hypermap.lemmaILTXRQD:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (p::nat \Rightarrow ?'a::type)\ k::nat.\ is\_loop\ H\ L \wedge is\_inj\_contour\ H\ p\ k \wedge (2::nat) \leq k \wedge belong\ (p\ (0::nat))\ L \wedge belong\ (p\ k)\ L \wedge (\forall i::nat.\ (0::nat) < i \wedge i < k \longrightarrow \neg\ belong\ (p\ i)\ L) \wedge (\forall (q::nat \Rightarrow ?'a::type)\ m::nat.\ \neg\ is\_Moebius\_contour\ H\ q\ m) \longrightarrow (p\ (1::nat) = HOL\_Light\_Import.inverse\ (node\_map\ H)\ (p\ (0::nat)) \longrightarrow p\ k \neq face\_map\ H\ (p\ (pred\ k))) \wedge (p\ (1::nat) = face\_map\ H\ (p\ (0::nat)) \longrightarrow p\ k \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (p\ (pred\ k)))$

thm Hypermap.inj_orbit_imp_inj_face_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ inj\_orbit\ (face\_map\ H)\ x\ k \longrightarrow is\_inj\_contour\ H\ (face\_contour\ H\ x)\ k$

thm Hypermap.lemma_inj_face_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ k < CARD\ (face\ H\ x) \longrightarrow is\_inj\_contour\ H\ (face\_contour\ H\ x)\ k$

thm Hypermap.lemma_face_cycle:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ POWER\ (face\_map\ H)\ (CARD\ (face\ H\ x))\ x = x$

thm Hypermap.lemma_card_inverse_map_eq:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ p\ s \longrightarrow orbit\_map\ (HOL\_Light\_Import.inverse\ p)\ x = orbit\_map\ p\ x$

thm Hypermap.inj_orbit_imp_inj_node_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ inj\_orbit\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ x\ k \longrightarrow is\_inj\_contour\ H\ (node\_contour\ H\ x)\ k$

thm Hypermap.lemma_inj_node_contour:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ k::nat.\ k < CARD\ (node\ H\ x) \longrightarrow is\_inj\_contour\ H\ (node\_contour\ H\ x)\ k$

thm Hypermap.lemma_node_cycle:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ POWER\ (node\_map\ H)\ (CARD\ (node\ H\ x))\ x = x$

thm Hypermap.lemma_node_inverse_cycle:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ (CARD\ (node\ H\ x))\ x = x$

thm Hypermap.lemma_node_contour_connection:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (node\ H\ x) \longrightarrow (\exists k<CARD\ (node\ H\ x).\ node\_contour\ H\ x\ (0::nat) = x \wedge node\_contour\ H\ x\ k = y \wedge is\_inj\_contour\ H\ (node\_contour\ H\ x)\ k)$

thm Hypermap.lemma_via_inverse_node_map:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (node\ H\ x) \longrightarrow (\exists j<CARD\ (node\ H\ x).\ y = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ j\ x)$

thm Hypermap.lemmaICJHAOQ:

$\forall (H::?'a::type\ hypermap)\ L::?'a::type\ loop.\ is\_loop\ H\ L \land (\forall (g::nat \Rightarrow ?'a::type)\ m::nat.\ \neg\ is\_Moebius\_contour\ H\ g\ m) \longrightarrow \neg\ (\exists (p::nat \Rightarrow ?'a::type)\ k::nat.\ (1::nat) \leq k \land is\_contour\ H\ p\ k \land belong\ (p\ (0::nat))\ L \land (\forall i::nat.\ (0::nat) < i \land i \leq k \longrightarrow \neg\ belong\ (p\ i)\ L) \land p\ (1::nat) = face\_map\ H\ (p\ (0::nat)) \land node\ H\ (p\ (0::nat)) \neq node\ H\ (p\ k) \land (\exists y::?'a::type.\ IN\ y\ (node\ H\ (p\ k)) \land belong\ y\ L))$

thm Hypermap.lemmaThreeDarts:

$\forall (H::?'a::type\ hypermap)\ L::?'a::type\ loop.\ is\_loop\ H\ L \land (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow (3::nat) \leq CARD\ (face\ H\ x)) \land (\exists (x::?'a::type)\ y::?'a::type.\ node\ H\ x \neq node\ H\ y \land belong\ x\ L \land belong\ y\ L) \longrightarrow (3::nat) \leq HOL\_Light\_Import.size\ L$

thm DEF_is_node_going:

$is\_node\_going = (\lambda(\_2414278::?'a::type\ hypermap)\ (\_2414279::?'a::type\ loop)\ (\_2414280::?'a::type)\ \_2414281::?'a::type.\ \exists k::nat.\ \_2414281 = POWER\ (next\ \_2414279)\ k\ \_2414280 \land (\forall i \leq k.\ POWER\ (next\ \_2414279)\ i\ \_2414280 = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ \_2414278))\ i\ \_2414280))$

thm Hypermap.is_node_going:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ is\_node\_going\ H\ L\ x\ y = (\exists k::nat.\ y = POWER\ (next\ L)\ k\ x \land (\forall i \leq k.\ POWER\ (next\ L)\ i\ x = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ i\ x))$

thm DEF_atom:

$atom = (\lambda(\_2414310::?'a::type\ hypermap)\ (\_2414311::?'a::type\ loop)\ \_2414312::?'a::type.\ GSPEC\ (\lambda GEN\%PVAR\%262::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%262\ (is\_node\_going\ \_2414310\ \_2414311\ \_2414312\ y \lor is\_node\_going\ \_2414310\ \_2414311\ y\ \_2414312)\ y))$

thm Hypermap.atom:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ atom\ H\ L\ x = GSPEC\ (\lambda GEN\%PVAR\%262::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%262\ (is\_node\_going\ H\ L\ x\ y \lor is\_node\_going\ H\ L\ y\ x)\ y)$

thm Hypermap.atom_reflect:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ IN\ x\ (atom\ H\ L\ x)$

thm Hypermap.atom_sym:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (atom\ H\ L\ x) \longrightarrow IN\ x\ (atom\ H\ L\ y)$

thm Hypermap.lemma_transitive_going:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$ $is\_node\_going\ H\ L\ x\ y \wedge is\_node\_going\ H\ L\ y\ z \longrightarrow is\_node\_going\ H\ L\ x\ z$

thm Hypermap.lemma_on_way_going:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$ $is\_node\_going\ H\ L\ x\ y \wedge is\_node\_going\ H\ L\ x\ z \longrightarrow is\_node\_going\ H\ L\ y\ z \vee$ $is\_node\_going\ H\ L\ z\ y$

thm Hypermap.lemma_second_on_way_going:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$ $is\_node\_going\ H\ L\ x\ z \wedge is\_node\_going\ H\ L\ y\ z \longrightarrow is\_node\_going\ H\ L\ x\ y \vee$ $is\_node\_going\ H\ L\ y\ x$

thm Hypermap.atom_trans:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$ $IN\ x\ (atom\ H\ L\ y) \wedge IN\ y\ (atom\ H\ L\ z) \longrightarrow IN\ x\ (atom\ H\ L\ z)$

thm Hypermap.lemma_atom_sub_loop:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow SUB$-$SET\ (atom\ H\ L\ x)\ (dart\_of\ L)$

thm Hypermap.lemma_atom_out_side_loop:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ \neg\ belong\ x\ L \longrightarrow$ $atom\ H\ L\ x = INSERT\ x\ EMPTY$

thm Hypermap.lemma_atom_sub_node:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ SUBSET\ (atom\ H\ L$ $x)\ (node\ H\ x)$

thm Hypermap.lemma_atom_sub_dart_set:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow$ $SUBSET\ (atom\ H\ L\ x)\ (dart\ H)$

thm Hypermap.lemma_atom_finite:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ FINITE\ (atom\ H\ L$ $x) \wedge (1::nat) \leq CARD\ (atom\ H\ L\ x)$

thm Hypermap.lemma_identity_atom:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ y$ $(atom\ H\ L\ x) \longrightarrow atom\ H\ L\ x = atom\ H\ L\ y$

thm Hypermap.lemma_atom_absorb_quark:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ y$ $(atom\ H\ L\ x) \wedge next\ L\ y = HOL\_Light\_Import.inverse\ (node\_map\ H)\ y \longrightarrow$ $IN\ (next\ L\ y)\ (atom\ H\ L\ x)$

thm Hypermap.lemma_second_absorb_quark:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (atom\ H\ L\ x) \wedge y = HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ y) \longrightarrow IN\ (back\ L\ y)\ (atom\ H\ L\ x)$

thm Hypermap.next_and_loop_darts:

$\forall L::?'a::type\ loop.\ FINITE\ (dart\_of\ L) \wedge permutes\ (next\ L)\ (dart\_of\ L)$

thm Hypermap.back_and_loop_darts:

$\forall L::?'a::type\ loop.\ FINITE\ (dart\_of\ L) \wedge permutes\ (back\ L)\ (dart\_of\ L)$

thm Hypermap.lemma_border_of_atom:

$\forall (H::?'a::type\ hypermap)\ L::?'a::type\ loop.\ \exists (h::?'a::type \Rightarrow ?'a::type)\ t::?'a::type \Rightarrow ?'a::type.\ \forall x::?'a::type.\ belong\ x\ L \wedge (\exists (y::?'a::type)\ z::?'a::type.\ belong\ y\ L \wedge belong\ z\ L \wedge node\ H\ y \neq node\ H\ z) \longrightarrow IN\ (h\ x)\ (atom\ H\ L\ x) \wedge IN\ (t\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (h\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (h\ x) \wedge t\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (t\ x))$

thm DEF_is_normal:

$is\_normal = (\lambda(\_2414613::?'a::type\ hypermap)\ \_2414614::?'a::type\ loop \Rightarrow bool.\ (\forall L::?'a::type\ loop.\ IN\ L\ \_2414614 \longrightarrow is\_loop\ \_2414613\ L \wedge (\exists x::?'a::type.\ IN\ x\ (dart\ \_2414613) \wedge belong\ x\ L)) \wedge (\forall L::?'a::type\ loop.\ IN\ L\ \_2414614 \longrightarrow (\exists (y::?'a::type)\ z::?'a::type.\ belong\ y\ L \wedge belong\ z\ L \wedge node\ \_2414613\ y \neq node\ \_2414613\ z)) \wedge (\forall (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ \_2414614 \wedge IN\ L'\ \_2414614 \wedge belong\ x\ L \wedge belong\ x\ L' \longrightarrow L = L') \wedge (\forall (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ L\ \_2414614 \wedge belong\ x\ L \wedge IN\ y\ (node\ \_2414613\ x) \longrightarrow (\exists L'::?'a::type\ loop.\ IN\ L'\ \_2414614 \wedge belong\ y\ L')))$

thm Hypermap.is_normal:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF = ((\forall L::?'a::type\ loop.\ IN\ L\ NF \longrightarrow is\_loop\ H\ L \wedge (\exists x::?'a::type.\ IN\ x\ (dart\ H) \wedge belong\ x\ L)) \wedge (\forall L::?'a::type\ loop.\ IN\ L\ NF \longrightarrow (\exists (y::?'a::type)\ z::?'a::type.\ belong\ y\ L \wedge belong\ z\ L \wedge node\ H\ y \neq node\ H\ z)) \wedge (\forall (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ x\ L' \longrightarrow L = L') \wedge (\forall (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \wedge IN\ y\ (node\ H\ x) \longrightarrow (\exists L'::?'a::type\ loop.\ IN\ L'\ NF \wedge belong\ y\ L')))$

thm Hypermap.lemm_nornal_loop_sub_dart:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \longrightarrow SUBSET\ (dart\_of\ L)\ (dart\ H)$

thm DEF_quotient_darts:

$quotient\_darts = (\lambda(\_2414625::?'a::type\ hypermap)\ \_2414626::?'a::type\ loop \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%263::?'a::type \Rightarrow bool.\ \exists (L::?'a::type\ loop)\ x::?'a::type.\ SETSPEC\ GEN\%PVAR\%263\ (IN\ L\ \_2414626 \wedge belong\ x\ L)\ (atom\ \_2414625\ L\ x)))$

thm Hypermap.quotient_darts:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ quotient\_darts\ H\ NF = GSPEC\ (\lambda GEN\%PVAR\%263::?'a::type \Rightarrow bool.\ \exists (L::?'a::type\ loop)\ x::?'a::type.\ SETSPEC\ GEN\%PVAR\%263\ (IN\ L\ NF \wedge belong\ x\ L)\ (atom\ H\ L\ x))$

thm DEF_support_darts:

$support\_darts = (\lambda\_2414637::?'a::type\ loop \Rightarrow bool.\ UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%264::?'a::type \Rightarrow bool.\ \exists L::?'a::type\ loop.\ SETSPEC\ GEN\%PVAR\%264\ (IN\ L\ \_2414637)\ (dart\_of\ L))))$

thm Hypermap.support_darts:

$\forall NF::?'a::type\ loop \Rightarrow bool.\ support\_darts\ NF = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%264::?'a::type \Rightarrow bool.\ \exists L::?'a::type\ loop.\ SETSPEC\ GEN\%PVAR\%264\ (IN\ L\ NF)\ (dart\_of\ L)))$

thm Hypermap.lemma_in_loop:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ belong\ x\ L \wedge IN\ y\ (atom\ H\ L\ x) \longrightarrow belong\ y\ L$

thm Hypermap.lemma_in_dart:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow IN\ x\ (dart\ H)$

thm Hypermap.lemma_support_and_atoms:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow support\_darts\ NF = UNIONS\ (quotient\_darts\ H\ NF)$

thm Hypermap.lemma_finite_support:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow SUBSET\ (support\_darts\ NF)\ (dart\ H) \wedge FINITE\ (support\_darts\ NF)$

thm Hypermap.lemma_in_support2:

$\forall (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \wedge IN\ L\ NF \longrightarrow IN\ x\ (support\_darts\ NF)$

thm Hypermap.lemma_in_support:

$\forall (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (support\_darts\ NF) = (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L)$

thm Hypermap.lemma_node_in_support2:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (x::?'a::type)\ n::nat.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow IN\ (POWER\ (node\_map\ H)\ n\ x)\ (support\_darts\ NF)$

thm Hypermap.lemma_loop_outside_node:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $is\_normal\ H\ NF \wedge IN\ L\ NF \longrightarrow \neg\ SUBSET\ (dart\_of\ L)\ (node\ H\ x)$

thm Hypermap.lemma_size_of_normal_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \longrightarrow (2::nat) \leq HOL\_Light\_Import.size\ L$

thm Hypermap.disjoint_loops:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ x\ L' \longrightarrow L = L'$

thm Hypermap.lemma_choice_function:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ \exists\ choice\_function::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ \forall x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow choice\_function\ x = INSERT\ x\ EMPTY) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge choice\_function\ x = atom\ H\ L\ x))$

thm DEF_choice:

$choice = (SOME\ choice\_function::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ \forall (\_2414743::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow choice\_function\ \_2414743\ H\ NF\ x = INSERT\ x\ EMPTY) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge choice\_function\ \_2414743\ H\ NF\ x = atom\ H\ L\ x)))\ (105::nat)$

thm Hypermap.lemma_choice:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow choice\ H\ NF\ x = INSERT\ x\ EMPTY) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge choice\ H\ NF\ x = atom\ H\ L\ x))$

thm Hypermap.first_unique_choice:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\forall x::?'a::type.\ \neg\ IN\ x\ (support\_darts\ NF) \longrightarrow choice\ H\ NF\ x = INSERT\ x\ EMPTY) \wedge (\forall (L::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \longrightarrow choice\ H\ NF\ x = atom\ H\ L\ x)$

thm Hypermap.unique_choice:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow choice\ H\ NF\ x = atom\ H\ L\ x$

thm Hypermap.lemma_in_quotient:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \longrightarrow IN\ (atom\ H\ L\ x)\ (quotient\_darts\ H\ NF)$

thm Hypermap.lemma_finite_quotient_darts:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow FINITE\ (quotient\_darts\ H\ NF)$

thm Hypermap.lemma_finite_normal_loops:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow FINITE\ NF \wedge CARD\ NF \leq CARD\ (dart\ H)$

thm Hypermap.lemma_border_of_atom2:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ \exists (h::?'a::type \Rightarrow ?'a::type)\ t::?'a::type \Rightarrow ?'a::type.\ \forall x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow h\ x = x \wedge t\ x = x) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge IN\ (h\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (h\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (h\ x) \wedge IN\ (t\ x)\ (atom\ H\ L\ x) \wedge t\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (t\ x))))$

thm DEF_head:

$head = (SOME\ h::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall \_2414790::nat.\ \exists t::?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow h\ \_2414790\ H\ NF\ x = x \wedge t\ H\ NF\ x = x) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge IN\ (h\ \_2414790\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (h\ \_2414790\ H\ NF\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (h\ \_2414790\ H\ NF\ x) \wedge IN\ (t\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge t\ H\ NF\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (t\ H\ NF\ x)))))\ (106::nat)$

thm DEF_tail:

$tail = (SOME\ t::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'a::type.\ \forall (\_2414791::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow head\ H\ NF\ x = x \wedge t\ \_2414791\ H\ NF\ x = x) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge IN\ (head\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (head\ H\ NF\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (head\ H\ NF\ x) \wedge IN\ (t\ \_2414791\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge t\ \_2414791\ H\ NF\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (t\ \_2414791\ H\ NF\ x)))))\ (107::nat)$

thm Hypermap.lemma_head_tail:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ x\ (support\_darts\ NF) \longrightarrow head\ H\ NF\ x = x \wedge tail\ H\ NF\ x = x) \wedge (IN\ x\ (support\_darts\ NF) \longrightarrow (\exists L::?'a::type\ loop.\ IN\ L\ NF \wedge belong\ x\ L \wedge IN\ (head\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (head\ H\ NF\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (head\ H\ NF\ x) \wedge IN\ (tail\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge tail\ H\ NF\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (tail\ H\ NF\ x))))$

thm Hypermap.lemma_unique_head:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge IN\ y\ (atom\ H\ L\ x) \wedge next\ L\ y \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ y \longrightarrow head\ H\ NF\ x = y$

thm Hypermap.lemma_unique_tail:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge IN\ y\ (atom\ H\ L\ x) \wedge y \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ y) \longrightarrow tail\ H\ NF\ x = y$

thm Hypermap.head_on_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow IN\ (head\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge next\ L\ (head\ H\ NF\ x) \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (head\ H\ NF\ x)$

thm Hypermap.tail_on_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow IN\ (tail\ H\ NF\ x)\ (atom\ H\ L\ x) \wedge tail\ H\ NF\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ (back\ L\ (tail\ H\ NF\ x))$

thm Hypermap.change_to_margin:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (x::?'a::type)\ L::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow atom\ H\ L\ x = atom\ H\ L\ (tail\ H\ NF\ x) \wedge atom\ H\ L\ x = atom\ H\ L\ (head\ H\ NF\ x)$

thm Hypermap.change_parameters:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge IN\ y\ (atom\ H\ L\ x) \longrightarrow head\ H\ NF\ y = head\ H\ NF\ x \wedge tail\ H\ NF\ y = tail\ H\ NF\ x$

thm Hypermap.margin_in_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow belong\ (head\ H\ NF\ x)\ L \wedge belong\ (tail\ H\ NF\ x)\ L$

thm Hypermap.lemma_map_next:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge IN\ (next\ L\ x)\ (atom\ H\ L\ x) \longrightarrow next\ L\ x = HOL\_Light\_Import.inverse\ (node\_map\ H)\ x$

thm Hypermap.next_head_outside_atom:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ $\neg$ *IN* (*next L* (*head H NF x*)) (*atom H L x*)

thm Hypermap.value_next_of_head:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ *next L* (*head H NF x*) = *face_map H* (*head H NF x*)

thm Hypermap.back_tail_outside_atom:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ $\neg$ *IN* (*back L* (*tail H NF x*)) (*atom H L x*)

thm Hypermap.face_map_on_margin:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ *belong* (*face_map H* (*head H NF x*)) *L* $\wedge$ *belong* (*HOL_Light_Import*.*inverse* (*face_map H*) (*tail H NF x*)) *L* $\wedge$ *face_map H* (*head H NF x*) = *tail H NF* (*face_map H* (*head H NF x*)) $\wedge$ *HOL_Light_Import*.*inverse* (*face_map H*) (*tail H NF x*) = *head H NF* (*HOL_Light_Import*.*inverse* (*face_map H*) (*tail H NF x*))

thm Hypermap.node_map_on_margin:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ ($\exists$ *L'*::*?'a*::*type loop*. *IN L' NF* $\wedge$ *belong* (*node_map H* (*tail H NF x*)) *L'* $\wedge$ *node_map H* (*tail H NF x*) = *head H NF* (*node_map H* (*tail H NF x*))) $\wedge$ ($\exists$ *P*::*?'a*::*type loop*. *IN P NF* $\wedge$ *belong* (*HOL_Light_Import*.*inverse* (*node_map H*) (*head H NF x*)) *P* $\wedge$ *HOL_Light_Import*.*inverse* (*node_map H*) (*head H NF x*) = *tail H NF* (*HOL_Light_Import*.*inverse* (*node_map H*) (*head H NF x*)))

thm Hypermap.node_map_free_loop:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ *node_map H* (*tail H NF x*) = *head H NF* (*node_map H* (*tail H NF x*)) $\wedge$ *HOL_Light_Import*.*inverse* (*node_map H*) (*head H NF x*) = *tail H NF* (*HOL_Light_Import*.*inverse* (*node_map H*) (*head H NF x*))

thm Hypermap.from_tail:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\wedge$ *IN y* (*atom H L x*) $\longrightarrow$ ($\forall$ *i*$\leq$*index L* (*tail H NF x*) *y*. *POWER* (*next L*) *i* (*tail H NF x*) = *POWER* (*HOL_Light_Import*.*inverse* (*node_map H*)) *i* (*tail H NF x*))

thm Hypermap.to_head:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\wedge$ *IN y* (*atom H*

L x) ⟶ (∀ i≤index L y (head H NF x). POWER (next L) i y = POWER (HOL_Light_Import.inverse (node_map H)) i y)

thm Hypermap.add_steps:

∀ (L::?'a::type loop) (x::?'a::type) (y::?'a::type) z::?'a::type. belong x L ∧ belong y L ∧ belong z L ∧ index L x y ≤ index L x z ⟶ index L x y + index L y z = index L x z

thm Hypermap.add_steps_in_atom:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) (x::?'a::type) y::?'a::type. is_normal H NF ∧ IN L NF ∧ belong x L ∧ IN y (atom H L x) ⟶ index L (tail H NF x) y + index L y (head H NF x) = index L (tail H NF x) (head H NF x)

thm Hypermap.lemma_in_atom:

∀ (H::?'a::type hypermap) (L::?'a::type loop) (x::?'a::type) m::nat. is_loop H L ∧ (∀ i≤m. POWER (next L) i x = POWER (HOL_Light_Import.inverse (node_map H)) i x) ⟶ IN (POWER (next L) m x) (atom H L x)

thm Hypermap.lemma_in_atom2:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_normal H NF ∧ IN L NF ∧ belong x L ⟶ (∀ i≤index L x (head H NF x). IN (POWER (next L) i x) (atom H L x))

thm Hypermap.atomic_particles:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_normal H NF ∧ IN L NF ∧ belong x L ⟶ atom H L x = GSPEC (λGEN%PVAR%266::?'a::type. ∃ i::nat. SETSPEC GEN%PVAR%266 (i ≤ index L (tail H NF x) (head H NF x)) (POWER (next L) i (tail H NF x))) ∧ (∀ i≤index L (tail H NF x) (head H NF x). POWER (next L) i (tail H NF x) = POWER (HOL_Light_Import.inverse (node_map H)) i (tail H NF x)) ∧ atom H L x = GSPEC (λGEN%PVAR%267::?'a::type. ∃ i::nat. SETSPEC GEN%PVAR%267 (i ≤ index L (tail H NF x) (head H NF x)) (POWER (HOL_Light_Import.inverse (node_map H)) i (tail H NF x)))

thm Hypermap.atom_one_point:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_normal H NF ∧ IN L NF ∧ belong x L ∧ head H NF x = tail H NF x ⟶ atom H L x = INSERT x EMPTY

thm Hypermap.lemma_atom_node_eq:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (x::?'a::type) L::?'a::type loop. is_normal H NF ∧ IN L NF ∧ belong x L ∧ IN (node_map H (tail H NF x)) (atom H L x) ⟶ atom H L x = node H x

thm Hypermap.lemma_fmap:

∀ (H::?'a::type hypermap) NF::?'a::type loop ⇒ bool. ∃ f::(?'a::type ⇒ bool) ⇒ ?'a::type ⇒ bool. ∀ s::?'a::type ⇒ bool. is_normal H NF ⟶ (¬ IN s

$(quotient\_darts\ H\ NF) \longrightarrow f\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \wedge s = atom\ H\ L\ x \wedge f\ s = atom\ H\ L\ (face\_map\ H\ (head\ H\ NF\ x))))$

thm Hypermap.lemma_nmap:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ \exists f::(?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow bool.\ \forall s::?'a::type \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow f\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ (node\_map\ H\ (tail\ H\ NF\ x))\ L' \wedge s = atom\ H\ L\ x \wedge f\ s = atom\ H\ L'\ (node\_map\ H\ (tail\ H\ NF\ x))))$

thm DEF_fmap:

$fmap = (SOME\ f::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow (?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow bool.\ \forall (\_2415527::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow f\ \_2415527\ H\ NF\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \wedge s = atom\ H\ L\ x \wedge f\ \_2415527\ H\ NF\ s = atom\ H\ L\ (face\_map\ H\ (head\ H\ NF\ x)))))\ (108::nat)$

thm Hypermap.lemma_face_map:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow fmap\ H\ NF\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge belong\ x\ L \wedge s = atom\ H\ L\ x \wedge fmap\ H\ NF\ s = atom\ H\ L\ (face\_map\ H\ (head\ H\ NF\ x))))$

thm DEF_nmap:

$nmap = (SOME\ f::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow (?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow bool.\ \forall (\_2415528::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow f\ \_2415528\ H\ NF\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ (node\_map\ H\ (tail\ H\ NF\ x))\ L' \wedge s = atom\ H\ L\ x \wedge f\ \_2415528\ H\ NF\ s = atom\ H\ L'\ (node\_map\ H\ (tail\ H\ NF\ x)))))\ (109::nat)$

thm Hypermap.lemma_node_map:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ s::?'a::type \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\neg\ IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow nmap\ H\ NF\ s = s) \wedge (IN\ s\ (quotient\_darts\ H\ NF) \longrightarrow (\exists (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ (node\_map\ H\ (tail\ H\ NF\ x))\ L' \wedge s = atom\ H\ L\ x \wedge nmap\ H\ NF\ s = atom\ H\ L'\ (node\_map\ H\ (tail\ H\ NF\ x))))$

thm Hypermap.unique_fmap:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow fmap\ H\ NF\ (atom\ H\ L\ x) = atom\ H\ L\ (face\_map\ H\ (head\ H\ NF\ x))$

thm Hypermap.unique_nmap:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (L'::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge IN\ L'\ NF \wedge belong\ x\ L \wedge belong\ (node\_map\ H\ (tail\ H\ NF\ x))\ L' \longrightarrow nmap\ H\ NF\ (atom\ H\ L\ x) = atom\ H\ L'\ (node\_map\ H\ (tail\ H\ NF\ x))$

thm Hypermap.fmap_permute:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow permutes\ (fmap\ H\ NF)\ (quotient\_darts\ H\ NF)$

thm Hypermap.nmap_permute:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow permutes\ (nmap\ H\ NF)\ (quotient\_darts\ H\ NF)$

thm DEF_emap:

$emap = (\lambda(\_2415819::?'a::type\ hypermap)\ \_2415820::?'a::type\ loop \Rightarrow bool.\ HOL\_Light\_Import.inverse\ (fmap\ \_2415819\ \_2415820) \circ HOL\_Light\_Import.inverse\ (nmap\ \_2415819\ \_2415820))$

thm Hypermap.emap:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ emap\ H\ NF = HOL\_Light\_Import.inverse\ (fmap\ H\ NF) \circ HOL\_Light\_Import.inverse\ (nmap\ H\ NF)$

thm Hypermap.emap_permute:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow permutes\ (emap\ H\ NF)\ (quotient\_darts\ H\ NF)$

thm DEF_quotient:

$HOL\_Light\_Import.quotient = (\lambda(\_2415831::?'a::type\ hypermap)\ \_2415832::?'a::type\ loop \Rightarrow bool.\ hypermap\ (quotient\_darts\ \_2415831\ \_2415832,\ emap\ \_2415831\ \_2415832,\ nmap\ \_2415831\ \_2415832,\ fmap\ \_2415831\ \_2415832))$

thm Hypermap.quotient:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ HOL\_Light\_Import.quotient\ H\ NF = hypermap\ (quotient\_darts\ H\ NF,\ emap\ H\ NF,\ nmap\ H\ NF,\ fmap\ H\ NF)$

thm Hypermap.lemma_quotient:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow dart\ (HOL\_Light\_Import.quotient\ H\ NF) = quotient\_darts\ H\ NF \wedge edge\_map\ (HOL\_Light\_Import.quotient\ H\ NF) = emap\ H\ NF \wedge node\_map\ (HOL\_Light\_Import.quotient$

$H\ NF) = nmap\ H\ NF \wedge face\_map\ (HOL\_Light\_Import.quotient\ H\ NF) = fmap\ H\ NF$

thm Hypermap.choice_reflect:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow IN\ x\ (choice\ H\ NF\ x)$

thm Hypermap.lemma_choice_in_quotient:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\forall x::?'a::type.\ IN\ (choice\ H\ NF\ x)\ (quotient\_darts\ H\ NF) = IN\ x\ (support\_darts\ NF))$

thm Hypermap.atom_via_choice:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow (\forall atm::?'a::type \Rightarrow bool.\ IN\ atm\ (quotient\_darts\ H\ NF) = (\exists x::?'a::type.\ IN\ x\ (support\_darts\ NF) \wedge atm = choice\ H\ NF\ x))$

thm Hypermap.choice_identity:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ y\ (choice\ H\ NF\ x) \longrightarrow choice\ H\ NF\ y = choice\ H\ NF\ x$

thm Hypermap.choice_at_margin:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow choice\ H\ NF\ x = choice\ H\ NF\ (tail\ H\ NF\ x) \wedge choice\ H\ NF\ x = choice\ H\ NF\ (head\ H\ NF\ x)$

thm Hypermap.choice_and_head_tail:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow IN\ (tail\ H\ NF\ x)\ (choice\ H\ NF\ x) \wedge IN\ (head\ H\ NF\ x)\ (choice\ H\ NF\ x)$

thm Hypermap.fmap_via_choice:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow IN\ (face\_map\ H\ (head\ H\ NF\ x))\ (support\_darts\ NF) \wedge face\_map\ H\ (head\ H\ NF\ x) = tail\ H\ NF\ (face\_map\ H\ (head\ H\ NF\ x)) \wedge fmap\ H\ NF\ (choice\ H\ NF\ x) = choice\ H\ NF\ (face\_map\ H\ (head\ H\ NF\ x))$

thm Hypermap.nmap_via_choice:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow IN\ (node\_map\ H\ (tail\ H\ NF\ x))\ (support\_darts\ NF) \wedge node\_map\ H\ (tail\ H\ NF\ x) = head\ H\ NF\ (node\_map\ H\ (tail\ H\ NF\ x)) \wedge nmap\ H\ NF\ (choice\ H\ NF\ x) = choice\ H\ NF\ (node\_map\ H\ (tail\ H\ NF\ x))$

thm Hypermap.emap_via_choice:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow IN\ (edge\_map\ H\ (head\ H\ NF\ x))\ (support\_darts\ NF) \wedge edge\_map\ H\ (head\ H\ NF\ x) = head\ H\ NF\ (edge\_map\ H\ (head\ H\ NF\ x)) \wedge emap\ H\ NF\ (choice\ H\ NF\ x) = choice\ H\ NF\ (edge\_map\ H\ (head\ H\ NF\ x))$

thm Hypermap.lemmaJMKRXLA:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \wedge plain\_hypermap\ H \longrightarrow plain\_hypermap\ (HOL\_Light\_Import.quotient\ H\ NF)$

thm Hypermap.COMPOSE_INJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ w::?'a::type \Rightarrow bool.\ INJ\ f\ s\ t \wedge INJ\ g\ t\ w \longrightarrow INJ\ (g \circ f)\ s\ w$

thm Hypermap.COMPOSE_SURJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ w::?'a::type \Rightarrow bool.\ SURJ\ f\ s\ t \wedge SURJ\ g\ t\ w \longrightarrow SURJ\ (g \circ f)\ s\ w$

thm Hypermap.COMPOSE_BIJ:

$\forall (f::?'c::type \Rightarrow ?'b::type)\ (g::?'b::type \Rightarrow ?'a::type)\ (s::?'c::type \Rightarrow bool)\ (t::?'b::type \Rightarrow bool)\ w::?'a::type \Rightarrow bool.\ BIJ\ f\ s\ t \wedge BIJ\ g\ t\ w \longrightarrow BIJ\ (g \circ f)\ s\ w$

thm Hypermap.BIJ_INVERSE:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ (s::?'b::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ BIJ\ f\ s\ t \longrightarrow (\exists g::?'a::type \Rightarrow ?'b::type.\ (\forall x::?'b::type.\ IN\ x\ s \longrightarrow g\ (f\ x) = x) \wedge (\forall x::?'a::type.\ IN\ x\ t \longrightarrow f\ (g\ x) = x) \wedge BIJ\ g\ t\ s)$

thm Hypermap.I_BIJ:

$\forall s::?'a::type \Rightarrow bool.\ BIJ\ id\ s\ s$

thm DEF_iso:

$iso = (\lambda(\_2416170::?'b::type\ hypermap)\ \_2416171::?'a::type\ hypermap.\ \exists f::?'b::type \Rightarrow ?'a::type.\ BIJ\ f\ (dart\ \_2416170)\ (dart\ \_2416171) \wedge (\forall x::?'b::type.\ IN\ x\ (dart\ \_2416170) \longrightarrow edge\_map\ \_2416171\ (f\ x) = f\ (edge\_map\ \_2416170\ x) \wedge node\_map\ \_2416171\ (f\ x) = f\ (node\_map\ \_2416170\ x) \wedge face\_map\ \_2416171\ (f\ x) = f\ (face\_map\ \_2416170\ x)))$

thm Hypermap.iso:

$\forall (H::?'b::type\ hypermap)\ H'::?'a::type\ hypermap.\ iso\ H\ H' = (\exists f::?'b::type \Rightarrow ?'a::type.\ BIJ\ f\ (dart\ H)\ (dart\ H') \wedge (\forall x::?'b::type.\ IN\ x\ (dart\ H) \longrightarrow edge\_map\ H'\ (f\ x) = f\ (edge\_map\ H\ x) \wedge node\_map\ H'\ (f\ x) = f\ (node\_map\ H\ x) \wedge face\_map\ H'\ (f\ x) = f\ (face\_map\ H\ x)))$

thm Hypermap.iso_reflect:

$\forall H::?'a::type\ hypermap.\ iso\ H\ H$

thm Hypermap.iso_sym:

$\forall (H::?'b::type\ hypermap)\ G::?'a::type\ hypermap.\ iso\ H\ G \longrightarrow iso\ G\ H$

thm Hypermap.iso_trans:

$\forall (H::?'c::type\ hypermap)\ (G::?'b::type\ hypermap)\ W::?'a::type\ hypermap.\ iso\ H\ G \wedge iso\ G\ W \longrightarrow iso\ H\ W$

thm DEF_cycle:

$cycle = (\lambda(\_2416194::?'a::type\ hypermap)\ \_2416195::?'a::type\ loop.\ GSPEC\ (\lambda GEN\%PVAR\%269::?'a::type \Rightarrow bool.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%269\ (belong\ x\ \_2416195)\ (atom\ \_2416194\ \_2416195\ x)))$

thm Hypermap.cycle:

$\forall (H::?'a::type\ hypermap)\ L::?'a::type\ loop.\ cycle\ H\ L = GSPEC\ (\lambda GEN\%PVAR\%269::?'a::type \Rightarrow bool.\ \exists x::?'a::type.\ SETSPEC\ GEN\%PVAR\%269\ (belong\ x\ L)\ (atom\ H\ L\ x))$

thm Hypermap.lemma_in_cycle2:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ x::?'a::type.\ belong\ x\ L \longrightarrow IN\ (atom\ H\ L\ x)\ (cycle\ H\ L)$

thm Hypermap.lemma_cycle_eq:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ L'::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge IN\ L'\ NF \wedge cycle\ H\ L = cycle\ H\ L' \longrightarrow L = L'$

thm Hypermap.lemma_cycle_is_face:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow cycle\ H\ L = orbit\_map\ (fmap\ H\ NF)\ (atom\ H\ L\ x)$

thm Hypermap.lemma_cycle_finite:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \longrightarrow FINITE\ (cycle\ H\ L)$

thm Hypermap.lemmaQuotientFace:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ is\_normal\ H\ NF \longrightarrow face\_set\ (HOL\_Light\_Import.quotient\ H\ NF) = GSPEC\ (\lambda GEN\%PVAR\%270::(?'a::type \Rightarrow bool) \Rightarrow bool.\ \exists L::?'a::type\ loop.\ SETSPEC\ GEN\%PVAR\%270\ (IN\ L\ NF)\ (cycle\ H\ L))$

thm Hypermap.lemma_support_cycle:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_normal\ H\ NF \wedge IN\ L\ NF \longrightarrow dart\_of\ L = UNIONS\ (cycle\ H\ L)$

thm Hypermap.lemmaQF:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow face\ (HOL\_Light\_Import.quotient\ H\ NF)\ (atom\ H\ L\ x) = cycle\ H\ L$

thm Hypermap.lemma_support_QF:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow UNIONS\ (face\ (HOL\_Light\_Import.quotient\ H\ NF)\ (atom\ H\ L\ x)) = dart\_of\ L$

thm Hypermap.lemma_in_unions:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ x::?'a::type.\ IN\ x\ t \wedge IN\ t\ s \longrightarrow IN\ x\ (UNIONS\ s)$

thm Hypermap.lemma_sub_support:

$\forall (s::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow SUBSET\ t\ (UNIONS\ s)$

thm Hypermap.lemma_in_QF:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \longrightarrow (\forall y::?'a::type.\ belong\ y\ L = IN\ (choice\ H\ NF\ y)\ (face\ (HOL\_Light\_Import.quotient\ H\ NF)\ (atom\ H\ L\ x)))$

thm Hypermap.lemma_in_QuotientFace:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge belong\ y\ L \longrightarrow IN\ (atom\ H\ L\ y)\ (face\ (HOL\_Light\_Import.quotient\ H\ NF)\ (atom\ H\ L\ x))$

thm DEF_support_node:

$support\_node = (\lambda(\_2416553::?'a::type\ hypermap)\ (\_2416554::?'a::type\ loop \Rightarrow bool)\ \_2416555::?'a::type \Rightarrow bool.\ UNIONS\ (node\ (HOL\_Light\_Import.quotient\ \_2416553\ \_2416554)\ \_2416555))$

thm Hypermap.support_node:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ atm::?'a::type \Rightarrow bool.\ support\_node\ H\ NF\ atm = UNIONS\ (node\ (HOL\_Light\_Import.quotient\ H\ NF)\ atm)$

thm Hypermap.lemma_node_sub_support_darts:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow SUBSET\ (node\ H\ x)\ (support\_darts\ NF)$

thm Hypermap.lemma_in_node:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (node\ H\ x) = (\exists n::nat.\ y = POWER\ (node\_map\ H)\ n\ x)$

thm Hypermap.lemma_in_node2:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ IN\ (POWER\ (node\_map\ H)\ n\ x)\ (node\ H\ x)$

thm Hypermap.lemma_choice_sub_node:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \longrightarrow SUBSET\ (choice\ H\ NF\ x)\ (node\ H\ x)$

thm Hypermap.lemma_support_QN:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow support\_node\ H\ NF\ (choice\ H\ NF\ x) = node\ H\ x$

thm Hypermap.lemma_QuotientNode:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow node\ (HOL\_Light\_Import.quotient\ H\ NF)\ (choice\ H\ NF\ x) = GSPEC\ (\lambda GEN\%PVAR\%271::?'a::type \Rightarrow bool.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%271\ (IN\ y\ (node\ H\ x))\ (choice\ H\ NF\ y))$

thm Hypermap.lemma_in_QN:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \longrightarrow (\forall y::?'a::type.\ IN\ (choice\ H\ NF\ y)\ (node\ (HOL\_Light\_Import.quotient\ H\ NF)\ (choice\ H\ NF\ x)) = IN\ y\ (node\ H\ x))$

thm Hypermap.lemma_in_QuotientNode:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \wedge IN\ y\ (node\ H\ x) \longrightarrow IN\ (choice\ H\ NF\ y)\ (node\ (HOL\_Light\_Import.quotient\ H\ NF)\ (choice\ H\ NF\ x))$

thm Hypermap.lemma_in_node3:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ is\_normal\ H\ NF \wedge IN\ x\ (support\_darts\ NF) \wedge IN\ (choice\ H\ NF\ y)\ (node\ (HOL\_Light\_Import.quotient\ H\ NF)\ (choice\ H\ NF\ x)) \longrightarrow IN\ y\ (node\ H\ x)$

thm Hypermap.lemma_in_face:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ IN\ (POWER\ (face\_map\ H)\ n\ x)\ (face\ H\ x)$

thm Hypermap.face_map_restrict:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ permutes\ (res\ (face\_map\ H)\ (face\ H\ x))\ (face\ H\ x)$

thm Hypermap.power_res_face_map:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ POWER\ (res\ (face\_map\ H)\ (face\ H\ x))\ n\ x = POWER\ (face\_map\ H)\ n\ x$

thm DEF_face_loop:

face_loop = (λ(_2416783::?'a::type hypermap) _2416784::?'a::type. loop (face _2416783 _2416784, res (face_map _2416783) (face _2416783 _2416784)))

thm Hypermap.face_loop:

∀ (H::?'a::type hypermap) x::?'a::type. face_loop H x = loop (face H x, res (face_map H) (face H x))

thm DEF_face_collection:

face_collection = (λ_2416795::?'a::type hypermap. GSPEC (λGEN%PVAR%272::?'a::type loop. ∃ x::?'a::type. SETSPEC GEN%PVAR%272 (IN x (dart _2416795)) (face_loop _2416795 x)))

thm Hypermap.face_collection:

∀ H::?'a::type hypermap. face_collection H = GSPEC (λGEN%PVAR%272::?'a::type loop. ∃ x::?'a::type. SETSPEC GEN%PVAR%272 (IN x (dart H)) (face_loop H x))

thm Hypermap.face_loop_rep:

∀ (H::?'a::type hypermap) x::?'a::type. dart_of (face_loop H x) = face H x ∧ next (face_loop H x) = res (face_map H) (face H x)

thm Hypermap.lemma_inverse_res:

∀ (H::?'a::type hypermap) (x::?'a::type) y::?'a::type. IN y (face H x) ⟶ HOL_Light_Import.inverse (res (face_map H) (face H x)) y = HOL_Light_Import.inverse (face_map H) y

thm Hypermap.face_loop_lemma:

∀ (H::?'a::type hypermap) x::?'a::type. is_loop H (face_loop H x)

thm Hypermap.lemma_edge_nondegenerate:

∀ H::?'a::type hypermap. is_edge_nondegenerate H = (∀ x::?'a::type. IN x (dart H) ⟶ face_map H x ≠ HOL_Light_Import.inverse (node_map H) x)

thm Hypermap.normal_face_collection:

∀ H::?'a::type hypermap. (∀ x::?'a::type. IN x (dart H) ⟶ (∃ y::?'a::type. IN y (dart H) ∧ IN y (face H x) ∧ node H x ≠ node H y)) ⟶ is_normal H (face_collection H)

thm Hypermap.lemma_support_face_collection:

∀ H::?'a::type hypermap. support_darts (face_collection H) = dart H

thm Hypermap.lemma_card_face_collection:

∀ H::?'a::type hypermap. FINITE (face_collection H) ∧ CARD (face_collection H) = number_of_faces H

thm Hypermap.lemma_inverse_in_face:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (face\ H\ x) \longrightarrow IN\ (HOL\_Light\_Import.inverse\ (face\_map\ H)\ y)\ (face\ H\ x)$

thm Hypermap.lemma_power_inverse_in_face:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (y::?'a::type)\ n::nat.\ IN\ y\ (face\ H\ x) \longrightarrow IN\ (POWER\ (HOL\_Light\_Import.inverse\ (face\_map\ H))\ n\ y)\ (face\ H\ x)$

thm Hypermap.lemma_power_inverse_in_face2:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ IN\ (POWER\ (HOL\_Light\_Import.inverse\ (face\_map\ H))\ n\ x)\ (face\ H\ x)$

thm Hypermap.lemma_inverse_in_node:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ y::?'a::type.\ IN\ y\ (node\ H\ x) \longrightarrow IN\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ y)\ (node\ H\ x)$

thm Hypermap.lemma_power_inverse_in_node:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (y::?'a::type)\ n::nat.\ IN\ y\ (node\ H\ x) \longrightarrow IN\ (POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ n\ y)\ (node\ H\ x)$

thm Hypermap.lemma_power_inverse_in_node2:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ n::nat.\ IN\ (POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ n\ x)\ (node\ H\ x)$

thm Hypermap.SING_EQ:

$\forall (x::?'a::type)\ y::?'a::type.\ (INSERT\ x\ EMPTY = INSERT\ y\ EMPTY) = (x = y)$

thm Hypermap.face_quotient_lemma:

$\forall H::?'a::type\ hypermap.\ is\_edge\_nondegenerate\ H \wedge (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow (\exists y::?'a::type.\ IN\ y\ (dart\ H) \wedge IN\ y\ (face\ H\ x) \wedge node\ H\ x \neq node\ H\ y)) \longrightarrow (\forall x::?'a::type.\ choice\ H\ (face\_collection\ H)\ x = INSERT\ x\ EMPTY) \wedge iso\ H\ (HOL\_Light\_Import.quotient\ H\ (face\_collection\ H))$

thm DEF_canon_loop:

$canon\_loop = (\lambda(\_2417086::?'a::type\ hypermap)\ \_2417087::?'a::type\ loop \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%273::(?'a::type \Rightarrow bool) \Rightarrow bool.\ \exists qf::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%273\ (IN\ qf\ (face\_set\ (HOL\_Light\_Import.quotient\ \_2417086\ \_2417087)) \wedge (\forall s::?'a::type \Rightarrow bool.\ IN\ s\ qf \longrightarrow CARD\ s = (1::nat)))\ qf))$

thm Hypermap.canon_loop:

$\forall (H::?'a::type\ hypermap)\ NF::?'a::type\ loop \Rightarrow bool.\ canon\_loop\ H\ NF = GSPEC\ (\lambda GEN\%PVAR\%273::(?'a::type \Rightarrow bool) \Rightarrow bool.\ \exists qf::(?'a::type \Rightarrow$

bool) ⇒ *bool*. *SETSPEC GEN%PVAR%273* (*IN qf* (*face_set* (*HOL_Light_Import.quotient H NF*)) ∧ (∀ *s*::*?'a*::*type* ⇒ *bool*. *IN s qf* ⟶ *CARD s* = (*1*::*nat*))) *qf*)

thm Hypermap.set_one_point:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *x*::*?'a*::*type*. *FINITE s* ∧ *CARD s* = (*1*::*nat*) ∧ *IN x s* ⟶ *s* = *INSERT x EMPTY*

thm Hypermap.lemma_canonical_function:

∀ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* ⇒ *bool*. *is_normal H NF* ⟶ (∀ *t*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *IN t* (*canon_loop H NF*) = (∃ *L*::*?'a*::*type loop*. *IN L NF* ∧ *t* = *cycle H L* ∧ (∀ *x*::*?'a*::*type*. *belong x L* ⟶ *L* = *face_loop H x* ∧ *atom H L x* = *INSERT x EMPTY*)))

thm Hypermap.lemmaSTKBEPH:

∀ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* ⇒ *bool*. *is_normal H NF* ∧ *number_of_faces H* ≤ *CARD* (*canon_loop H NF*) ⟶ *NF* = *face_collection H* ∧ *iso H* (*HOL_Light_Import.quotient H NF*)

thm Hypermap.edge_cyclic_map_lemma:

∀ (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) *k*::*nat*. ∃ *e*::*?'a*::*type* ⇒ *?'a*::*type*. ∀ *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *e x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *p j* ∧ *e x* = *q* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *q j* ∧ *e x* = *p* ((*j* + *k*) *mod Suc k*))))

thm Hypermap.node_cyclic_map_lemma:

∀ (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) *k*::*nat*. ∃ *n*::*?'a*::*type* ⇒ *?'a*::*type*. ∀ *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *n x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *p j* ∧ *n x* = *q j*)) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *q j* ∧ *n x* = *p j*)))

thm Hypermap.face_cyclic_map_lemma:

∀ (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) *k*::*nat*. ∃ *f*::*?'a*::*type* ⇒ *?'a*::*type*. ∀ *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *f x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *p j* ∧ *f x* = *p* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃ *j*≤*k*. *x* = *q j* ∧ *f x* = *q* ((*j* + *k*) *mod Suc k*))))

thm DEF_cyc_emap:

cyc_emap = (*SOME e*::*nat* ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ *nat* ⇒ *?'a*::*type* ⇒ *?'a*::*type*. ∀ (*_2418111*::*nat*) (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *e _2418111 p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *e _2418111 p q k x* = *q* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *e _2418111 p q k x* = *p* ((*j* + *k*) *mod Suc k*))))) (*110*::*nat*)

thm Hypermap.lemma_cyclic_edge_map:

∀(*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *cyc_emap p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *cyc_emap p q k x* = *q* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *cyc_emap p q k x* = *p* ((*j* + *k*) *mod Suc k*))))

thm DEF_cyc_nmap:

cyc_nmap = (*SOME n*::*nat* ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ *nat* ⇒ *?'a*::*type* ⇒ *?'a*::*type*. ∀(*_2418112*::*nat*) (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *n _2418112 p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *n _2418112 p q k x* = *q j*)) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *n _2418112 p q k x* = *p j*)))) (*111*::*nat*)

thm Hypermap.lemma_cyclic_node_map:

∀(*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *cyc_nmap p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *cyc_nmap p q k x* = *q j*)) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *cyc_nmap p q k x* = *p j*)))

thm DEF_cyc_fmap:

cyc_fmap = (*SOME f*::*nat* ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ (*nat* ⇒ *?'a*::*type*) ⇒ *nat* ⇒ *?'a*::*type* ⇒ *?'a*::*type*. ∀(*_2418113*::*nat*) (*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *f _2418113 p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *f _2418113 p q k x* = *p* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *f _2418113 p q k x* = *q* ((*j* + *k*) *mod Suc k*))))) (*112*::*nat*)

thm Hypermap.lemma_cyclic_face_map:

∀(*p*::*nat* ⇒ *?'a*::*type*) (*q*::*nat* ⇒ *?'a*::*type*) (*k*::*nat*) *x*::*?'a*::*type*. (¬ *IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ *cyc_fmap p q k x* = *x*) ∧ (*IN x* (*HOL_Light_Import.UNION* (*support_list p k*) (*support_list q k*)) ⟶ (*IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *p j* ∧ *cyc_fmap p q k x* = *p* (*Suc j mod Suc k*))) ∧ (¬ *IN x* (*support_list p k*) ⟶ (∃*j*≤*k*. *x* = *q j* ∧ *cyc_fmap p q k x* = *q* ((*j* + *k*) *mod Suc k*))))

thm Hypermap.lemma_cyclic_emap:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ k::nat.\ is\_inj\_list\ p\ k \wedge is\_inj\_list\ q\ k \wedge is\_disjoint\ p\ q\ k\ k \longrightarrow (\forall i \le k.\ cyc\_emap\ p\ q\ k\ (p\ i) = q\ (Suc\ i\ mod\ Suc\ k) \wedge cyc\_emap\ p\ q\ k\ (q\ i) = p\ ((i + k)\ mod\ Suc\ k))$

thm Hypermap.lemma_cyclic_nmap:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ k::nat.\ is\_inj\_list\ p\ k \wedge is\_inj\_list\ q\ k \wedge is\_disjoint\ p\ q\ k\ k \longrightarrow (\forall i \le k.\ cyc\_nmap\ p\ q\ k\ (p\ i) = q\ i \wedge cyc\_nmap\ p\ q\ k\ (q\ i) = p\ i)$

thm Hypermap.lemma_cyclic_fmap:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ k::nat.\ is\_inj\_list\ p\ k \wedge is\_inj\_list\ q\ k \wedge is\_disjoint\ p\ q\ k\ k \longrightarrow (\forall i \le k.\ cyc\_fmap\ p\ q\ k\ (p\ i) = p\ (Suc\ i\ mod\ Suc\ k) \wedge cyc\_fmap\ p\ q\ k\ (q\ i) = q\ ((i + k)\ mod\ Suc\ k))$

thm DEF_cyclic_hypermap:

$cyclic\_hypermap = (\lambda(\_2418138::nat \Rightarrow ?'a::type)\ (\_2418139::nat \Rightarrow ?'a::type)\ \_2418140::nat.\ hypermap\ (HOL\_Light\_Import.UNION\ (support\_list\ \_2418138\ \_2418140)\ (support\_list\ \_2418139\ \_2418140),\ cyc\_emap\ \_2418138\ \_2418139\ \_2418140,\ cyc\_nmap\ \_2418138\ \_2418139\ \_2418140,\ cyc\_fmap\ \_2418138\ \_2418139\ \_2418140))$

thm Hypermap.cyclic_hypermap:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ k::nat.\ cyclic\_hypermap\ p\ q\ k = hypermap\ (HOL\_Light\_Import.UNION\ (support\_list\ p\ k)\ (support\_list\ q\ k),\ cyc\_emap\ p\ q\ k,\ cyc\_nmap\ p\ q\ k,\ cyc\_fmap\ p\ q\ k)$

thm Hypermap.lemma_cyclic_hypermap:

$\forall (p::nat \Rightarrow ?'a::type)\ (q::nat \Rightarrow ?'a::type)\ k::nat.\ is\_inj\_list\ p\ k \wedge is\_inj\_list\ q\ k \wedge is\_disjoint\ p\ q\ k\ k \longrightarrow dart\ (cyclic\_hypermap\ p\ q\ k) = HOL\_Light\_Import.UNION\ (support\_list\ p\ k)\ (support\_list\ q\ k) \wedge edge\_map\ (cyclic\_hypermap\ p\ q\ k) = cyc\_emap\ p\ q\ k \wedge node\_map\ (cyclic\_hypermap\ p\ q\ k) = cyc\_nmap\ p\ q\ k \wedge face\_map\ (cyclic\_hypermap\ p\ q\ k) = cyc\_fmap\ p\ q\ k$

thm DEF_is_no_double_joins:

$is\_no\_double\_joins = (\lambda\_2418263::?'a::type\ hypermap.\ \forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ \_2418263) \wedge IN\ y\ (node\ \_2418263\ x) \wedge IN\ (edge\_map\ \_2418263\ y)\ (node\ \_2418263\ (edge\_map\ \_2418263\ x)) \longrightarrow x = y)$

thm Hypermap.is_no_double_joins:

$\forall H::?'a::type\ hypermap.\ is\_no\_double\_joins\ H = (\forall (x::?'a::type)\ y::?'a::type.\ IN\ x\ (dart\ H) \wedge IN\ y\ (node\ H\ x) \wedge IN\ (edge\_map\ H\ y)\ (node\ H\ (edge\_map\ H\ x)) \longrightarrow x = y)$

thm Hypermap.margin_in_support_darts:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) *x*::*?'a*::*type*. *is_normal H NF* $\wedge$ *IN x* (*support_darts NF*) $\longrightarrow$ *IN* (*head H NF x*) (*support_darts NF*) $\wedge$ *IN* (*tail H NF x*) (*support_darts NF*)

thm Hypermap.lemmaQuotientNoDoubleJoins:

$\forall$ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* $\Rightarrow$ *bool*. *is_normal H NF* $\wedge$ *is_no_double_joins H* $\wedge$ *plain_hypermap H* $\longrightarrow$ *is_no_double_joins* (*HOL_Light_Import.quotient H NF*)

thm Hypermap.lemmaSimpleQuotient:

$\forall$ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* $\Rightarrow$ *bool*. *is_normal H NF* $\longrightarrow$ *simple_hypermap* (*HOL_Light_Import.quotient H NF*) = ($\forall$ (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN L NF* $\wedge$ *belong x L* $\wedge$ *belong y L* $\wedge$ *IN y* (*node H x*) $\longrightarrow$ *atom H L x* = *atom H L y*)

thm Hypermap.lemmaNodalFixedPoint:

$\forall$ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* $\Rightarrow$ *bool*. *is_normal H NF* $\wedge$ *simple_hypermap* (*HOL_Light_Import.quotient H NF*) $\longrightarrow$ ($\neg$ *is_node_nondegenerate* (*HOL_Light_Import.quotient H NF*)) = ($\exists$ (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *IN L NF* $\wedge$ *belong x L* $\wedge$ *SUBSET* (*node H x*) (*dart_of L*))

thm DEF_ind:

ind = (*SOME ind*::*nat* $\Rightarrow$ *?'a*::*type hypermap* $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *nat* $\Rightarrow$ *nat*. $\forall$ *_2418403*::*nat*. ($\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *ind _2418403 H x* (*0*::*nat*) = (*0*::*nat*)) $\wedge$ ($\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *n*::*nat*. *ind _2418403 H x* (*Suc n*) = *ind _2418403 H x n* + *pred* (*CARD* (*node H* (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*))))) (*113*::*nat*)

thm Hypermap.ind:

($\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *ind H x* (*0*::*nat*) = (*0*::*nat*)) $\wedge$ ($\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *n*::*nat*. *ind H x* (*Suc n*) = *ind H x n* + *pred* (*CARD* (*node H* (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*))))

thm Hypermap.ind_conjunct1:

$\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *n*::*nat*. *ind H x* (*Suc n*) = *ind H x n* + *pred* (*CARD* (*node H* (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*)))

thm Hypermap.ind_conjunct0:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *ind H x* (*0*::*nat*) = (*0*::*nat*)

thm DEF_mirror:

mirror = (*SOME mirror*::*nat* $\Rightarrow$ *?'a*::*type hypermap* $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *nat* $\Rightarrow$ *nat* $\Rightarrow$ *?'a*::*type*. $\forall$ *_2418411*::*nat*. ($\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *mirror _2418411 H x* (*0*::*nat*) = *node_contour H* (*node_map H x*)) $\wedge$ ($\forall$ (*H*::*?'a*::*type hypermap*) (*x*::*?'a*::*type*) *n*::*nat*. *mirror _2418411 H x* (*Suc n*) = *join* (*mirror*

_2418411 H x n) (*node_contour H* (*HOL_Light_Import.inverse* (*node_map H*) (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*))) (*ind H x n*))) (*114::nat*)

thm Hypermap.mirror:

(∀ (*H::?'a::type hypermap*) *x::?'a::type*. *mirror H x* (*0::nat*) = *node_contour H* (*node_map H x*)) ∧ (∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *n::nat*. *mirror H x* (*Suc n*) = *join* (*mirror H x n*) (*node_contour H* (*HOL_Light_Import.inverse* (*node_map H*) (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*))) (*ind H x n*))

thm Hypermap.mirror_conjunct1:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *n::nat*. *mirror H x* (*Suc n*) = *join* (*mirror H x n*) (*node_contour H* (*HOL_Light_Import.inverse* (*node_map H*) (*POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*Suc n*) *x*))) (*ind H x n*)

thm Hypermap.mirror_conjunct0:

∀ (*H::?'a::type hypermap*) *x::?'a::type*. *mirror H x* (*0::nat*) = *node_contour H* (*node_map H x*)

thm DEF_complement:

complement = (λ(*_2418412::?'a::type hypermap*) (*_2418413::?'a::type*) *_2418414::nat*. *mirror _2418412 _2418413 _2418414 _2418414*)

thm Hypermap.complement:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *n::nat*. *complement H x n* = *mirror H x n n*

thm Hypermap.lemma_node_nondegenerate:

∀ *H::?'a::type hypermap*. *is_node_nondegenerate H* = (∀ *x::?'a::type*. *IN x* (*dart H*) ⟶ (*2::nat*) ≤ *CARD* (*node H x*))

thm Hypermap.lemma_in_node1:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *y::?'a::type*. *IN y* (*node H x*) ⟶ *IN* (*node_map H y*) (*node H x*)

thm Hypermap.lemma_increasing_index_one:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *n::nat*. *is_node_nondegenerate H* ∧ *IN x* (*dart H*) ⟶ *ind H x n* < *ind H x* (*Suc n*)

thm Hypermap.lemma_increasing_index:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) (*n::nat*) *m::nat*. *is_node_nondegenerate H* ∧ *IN x* (*dart H*) ∧ *n* < *m* ⟶ *ind H x n* < *ind H x m*

thm Hypermap.lemma_lower_bound_index:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) *n::nat*. *is_node_nondegenerate H* ∧ *IN x* (*dart H*) ⟶ *n* ≤ *ind H x n*

thm Hypermap.lemma_segment_complement:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (n::nat)\ i::nat.\ is\_node\_nondegenerate\ H \wedge IN\ x\ (dart\ H) \wedge i \leq n \longrightarrow (\forall j \leq ind\ H\ x\ i.\ mirror\ H\ x\ i\ j = mirror\ H\ x\ n\ j)$

thm Hypermap.lemma_indepentdent_complement:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (n::nat)\ m::nat.\ is\_node\_nondegenerate\ H \wedge IN\ x\ (dart\ H) \wedge n \leq m \longrightarrow complement\ H\ x\ n = mirror\ H\ x\ m\ n$

thm Hypermap.lemma_evaluation_complement:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (n::nat)\ i::nat.\ is\_node\_nondegenerate\ H \wedge IN\ x\ (dart\ H) \wedge n \leq ind\ H\ x\ i \longrightarrow complement\ H\ x\ n = mirror\ H\ x\ i\ n$

thm Hypermap.lemma_inc_monotone:

$\forall id::nat \Rightarrow nat.\ (\forall i::nat.\ id\ i < id\ (Suc\ i)) \longrightarrow (\forall (i::nat)\ j::nat.\ (i < j) = (id\ i < id\ j))$

thm Hypermap.lemma_inc_injective:

$\forall id::nat \Rightarrow nat.\ (\forall i::nat.\ id\ i < id\ (Suc\ i)) \longrightarrow (\forall (i::nat)\ j::nat.\ (i = j) = (id\ i = id\ j))$

thm Hypermap.lemma_inc_not_decreasing:

$\forall id::nat \Rightarrow nat.\ (\forall i::nat.\ id\ i < id\ (Suc\ i)) \longrightarrow (\forall (i::nat)\ j::nat.\ (i \leq j) = (id\ i \leq id\ j))$

thm Hypermap.lemma_num_partition:

$\forall id::nat \Rightarrow nat.\ id\ (0::nat) = (0::nat) \wedge (\forall i::nat.\ id\ i < id\ (Suc\ i)) \longrightarrow (\forall n::nat.\ (\exists i::nat.\ n = id\ i) \vee (\exists j::nat.\ id\ j < n \wedge n < id\ (Suc\ j)))$

thm Hypermap.index_representation:

$\forall (m::nat)\ (u::nat)\ n::nat.\ m < n \wedge n < u \longrightarrow (\exists j \geq 1::nat.\ j < u - m \wedge n = m + j)$

thm Hypermap.lemma_num_partition2:

$\forall id::nat \Rightarrow nat.\ id\ (0::nat) = (0::nat) \wedge (\forall i::nat.\ id\ i < id\ (Suc\ i)) \longrightarrow (\forall n::nat.\ n = (0::nat) \vee (\exists (i::nat)\ j::nat.\ (1::nat) \leq j \wedge j \leq id\ (Suc\ i) - id\ i \wedge n = id\ i + j))$

thm Hypermap.lemma_complement_path:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ plain\_hypermap\ H \wedge is\_node\_nondegenerate\ H \wedge IN\ x\ (dart\ H) \longrightarrow (\forall i::nat.\ complement\ H\ x\ (ind\ H\ x\ i) = node\_map\ H\ (POWER\ (HOL\_Light\_Import.inverse\ (face\_map\ H))\ i\ x)) \wedge (\forall i::nat.\ complement\ H\ x\ (Suc\ (ind\ H\ x\ i)) = HOL\_Light\_Import.inverse\ (node\_map\ H)\ (POWER\ (HOL\_Light\_Import.inverse\ (face\_map\ H))\ (Suc\ i)\ x)) \wedge (\forall i::nat.\ face\_map\ H\ (complement\ H\ x\ (ind\ H\ x\ i)) = complement\ H\ x\ (Suc\ (ind\ H\ x\ i))) \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq j \wedge j < CARD\ (node\ H\ (POWER$

(*HOL_Light_Import.inverse* (*face_map* *H*)) (*Suc* *i*) *x*)) ⟶ *complement* *H* *x* (*ind* *H* *x* *i* + *j*) = *POWER* (*HOL_Light_Import.inverse* (*node_map* *H*)) *j* (*POWER* (*HOL_Light_Import.inverse* (*face_map* *H*)) (*Suc* *i*) *x*)) ∧ (∀ *n*::*nat*. *is_contour* *H* (*complement* *H* *x*) *n*)

thm Hypermap.lemma_inj_complement:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *plain_hypermap* *H* ∧ *simple_hypermap* *H* ∧ *is_node_nondegenerate* *H* ∧ *IN* *x* (*dart* *H*) ⟶ *is_inj_contour* *H* (*complement* *H* *x*) (*pred* (*ind* *H* *x* (*CARD* (*face* *H* *x*))))

thm DEF_is_restricted:

is_restricted = (λ*_2419455*::*?'a*::*type hypermap*. *dart* *_2419455* ≠ *EMPTY* ∧ *planar_hypermap* *_2419455* ∧ *plain_hypermap* *_2419455* ∧ *connected_hypermap* *_2419455* ∧ *simple_hypermap* *_2419455* ∧ *is_no_double_joins* *_2419455* ∧ *is_edge_nondegenerate* *_2419455* ∧ *is_node_nondegenerate* *_2419455* ∧ (∀ *x*::*?'a*::*type*. *IN* *x* (*dart* *_2419455*) ⟶ (*3*::*nat*) ≤ *CARD* (*face* *_2419455* *x*)))

thm Hypermap.is_restricted:

∀ *H*::*?'a*::*type hypermap*. *is_restricted* *H* = (*dart* *H* ≠ *EMPTY* ∧ *planar_hypermap* *H* ∧ *plain_hypermap* *H* ∧ *connected_hypermap* *H* ∧ *simple_hypermap* *H* ∧ *is_no_double_joins* *H* ∧ *is_edge_nondegenerate* *H* ∧ *is_node_nondegenerate* *H* ∧ (∀ *x*::*?'a*::*type*. *IN* *x* (*dart* *H*) ⟶ (*3*::*nat*) ≤ *CARD* (*face* *H* *x*)))

thm DEF_canon:

canon = (λ(*_2419460*::*?'a*::*type hypermap*) *_2419461*::*?'a*::*type loop* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%275*::*?'a*::*type loop*. ∃ *L*::*?'a*::*type loop*. *SETSPEC* *GEN%PVAR%275* (*IN* *L* *_2419461* ∧ (∃ *x*::*?'a*::*type*. *belong* *x* *L* ∧ *L* = *face_loop* *_2419460* *x*)) *L*))

thm Hypermap.canon:

∀ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* ⇒ *bool*. *canon* *H* *NF* = *GSPEC* (λ*GEN%PVAR%275*::*?'a*::*type loop*. ∃ *L*::*?'a*::*type loop*. *SETSPEC* *GEN%PVAR%275* (*IN* *L* *NF* ∧ (∃ *x*::*?'a*::*type*. *belong* *x* *L* ∧ *L* = *face_loop* *H* *x*)) *L*)

thm DEF_canon_darts:

canon_darts = (λ(*_2419472*::*?'a*::*type hypermap*) *_2419473*::*?'a*::*type loop* ⇒ *bool*. *UNIONS* (*GSPEC* (λ*GEN%PVAR%276*::*?'a*::*type* ⇒ *bool*. ∃ *L*::*?'a*::*type loop*. *SETSPEC* *GEN%PVAR%276* (*IN* *L* (*canon* *_2419472* *_2419473*)) (*dart_of* *L*))))

thm Hypermap.canon_darts:

∀ (*H*::*?'a*::*type hypermap*) *NF*::*?'a*::*type loop* ⇒ *bool*. *canon_darts* *H* *NF* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%276*::*?'a*::*type* ⇒ *bool*. ∃ *L*::*?'a*::*type loop*. *SETSPEC* *GEN%PVAR%276* (*IN* *L* (*canon* *H* *NF*)) (*dart_of* *L*)))

thm Hypermap.is_in_canon_darts:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$
$belong\ x\ L \land IN\ L\ (canon\ H\ NF) \longrightarrow IN\ x\ (canon\_darts\ H\ NF)$

thm Hypermap.lemma_in_canon_darts:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ x::?'a::type.\ IN\ x\ (canon\_darts\ H\ NF) = (\exists L::?'a::type\ loop.\ IN\ L\ (canon\ H\ NF) \land belong\ x\ L)$

thm Hypermap.lemma_not_in_canon_darts:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$
$is\_normal\ H\ NF \land IN\ L\ NF \land \neg\ IN\ L\ (canon\ H\ NF) \land belong\ x\ L \longrightarrow \neg\ IN$
$x\ (canon\_darts\ H\ NF)$

thm Hypermap.lemma_power_canon_next:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$
$IN\ L\ (canon\ H\ NF) \land belong\ x\ L \longrightarrow (\forall n::nat.\ POWER\ (face\_map\ H)\ n\ x =$
$POWER\ (next\ L)\ n\ x)$

thm Hypermap.lemma_true_loop1:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_restricted$
$H \land is\_normal\ H\ NF \land IN\ L\ NF \longrightarrow IN\ L\ (canon\ H\ NF) = (\exists x::?'a::type.$
$belong\ x\ L \land (\forall n::nat.\ POWER\ (face\_map\ H)\ n\ x = POWER\ (next\ L)\ n\ x))$

thm Hypermap.lemma_true_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_restricted$
$H \land is\_normal\ H\ NF \land IN\ L\ NF \longrightarrow IN\ L\ (canon\ H\ NF) = (\exists x::?'a::type.$
$dart\_of\ L = face\ H\ x)$

thm Hypermap.lemma_true_loop_via_map:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_restricted$
$H \land is\_normal\ H\ NF \land IN\ L\ NF \longrightarrow IN\ L\ (canon\ H\ NF) = (\forall x::?'a::type.$
$belong\ x\ L \longrightarrow next\ L\ x = face\_map\ H\ x)$

thm Hypermap.lemma_false_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ L::?'a::type\ loop.\ is\_restricted$
$H \land is\_normal\ H\ NF \land IN\ L\ NF \longrightarrow (\neg\ IN\ L\ (canon\ H\ NF)) = (\exists x::?'a::type.$
$belong\ x\ L \land next\ L\ x = HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)$

thm Hypermap.lemma_next_on_normal_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$
$is\_normal\ H\ NF \land IN\ L\ NF \land belong\ x\ L \longrightarrow next\ L\ x = face\_map\ H\ x \lor$
$next\ L\ x = HOL\_Light\_Import.inverse\ (node\_map\ H)\ x$

thm Hypermap.lemma_next_exclusive:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$
$is\_restricted\ H \land is\_normal\ H\ NF \land IN\ L\ NF \land belong\ x\ L \longrightarrow (next\ L\ x =$
$face\_map\ H\ x) = (next\ L\ x \neq HOL\_Light\_Import.inverse\ (node\_map\ H)\ x)$

thm Hypermap.lemma_next_exclusive2:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ (*next L x* = *HOL_Light_Import.inverse* (*node_map H*) *x*) = (*next L x* $\neq$ *face_map H x*)

thm Hypermap.lemma_head_via_restricted:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ (*head H NF x* = *x*) = (*next L x* = *face_map H x*)

thm Hypermap.lemma_head:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\wedge$ *IN y* (*atom H L x*) $\longrightarrow$ (*head H NF x* = *y*) = (*next L y* = *face_map H y*)

thm Hypermap.lemma_tail_via_restricted:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ (*tail H NF x* = *x*) = (*x* = *face_map H* (*back L x*))

thm Hypermap.lemma_tail:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\wedge$ *IN y* (*atom H L x*) $\longrightarrow$ (*tail H NF x* = *y*) = (*y* = *face_map H* (*back L y*))

thm Hypermap.lemma_singleton_atom:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ *belong x L* $\longrightarrow$ (*atom H L x* = *INSERT x EMPTY*) = (*next L x* = *face_map H x* $\wedge$ *face_map H* (*back L x*) = *x*)

thm DEF_is_split_condition:

is_split_condition = (λ(*_2419752*::*?'a*::*type hypermap*) (*_2419753*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*_2419754*::*?'a*::*type loop*) *_2419755*::*?'a*::*type*. *is_restricted _2419752* $\wedge$ *is_normal _2419752 _2419753* $\wedge$ *IN _2419754 _2419753* $\wedge$ $\neg$ *IN _2419754* (*canon _2419752 _2419753*) $\wedge$ *belong _2419755 _2419754* $\wedge$ *head _2419752 _2419753 _2419755* = *_2419755*)

thm Hypermap.is_split_condition:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_split_condition H NF L x* = (*is_restricted H* $\wedge$ *is_normal H NF* $\wedge$ *IN L NF* $\wedge$ $\neg$ *IN L* (*canon H NF*) $\wedge$ *belong x L* $\wedge$ *head H NF x* = *x*)

thm Hypermap.lemma_mInside_Exists:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_split_condition H NF L x* $\longrightarrow$ ($\exists$ *m*::*nat*. ($\forall$ *i*$\leq$*Suc m*. *POWER* (*next L*) *i x*

= *POWER* (*face_map H*) *i x*) ∧ *POWER* (*next L*) (*Suc* (*Suc m*)) *x* ≠ *POWER* (*face_map H*) (*Suc* (*Suc m*)) *x*)

thm DEF_mInside:

mInside = (*SOME m::nat* ⇒ *?'a::type hypermap* ⇒ (*?'a::type loop* ⇒ *bool*) ⇒ *?'a::type loop* ⇒ *?'a::type* ⇒ *nat*. ∀ (*_2419824::nat*) (*H::?'a::type hypermap*) (*NF::?'a::type loop* ⇒ *bool*) (*L::?'a::type loop*) *x::?'a::type*. *is_split_condition H NF L x* ⟶ (∀ *i*≤*Suc* (*m _2419824 H NF L x*). *POWER* (*next L*) *i x* = *POWER* (*face_map H*) *i x*) ∧ *POWER* (*next L*) (*Suc* (*Suc* (*m _2419824 H NF L x*))) *x* ≠ *POWER* (*face_map H*) (*Suc* (*Suc* (*m _2419824 H NF L x*))) *x*) (*115::nat*)

thm Hypermap.lemma_mInside:

∀ (*H::?'a::type hypermap*) (*NF::?'a::type loop* ⇒ *bool*) (*L::?'a::type loop*) *x::?'a::type*. *is_split_condition H NF L x* ⟶ (∀ *i*≤*Suc* (*mInside H NF L x*). *POWER* (*next L*) *i x* = *POWER* (*face_map H*) *i x*) ∧ *POWER* (*next L*) (*Suc* (*Suc* (*mInside H NF L x*))) *x* ≠ *POWER* (*face_map H*) (*Suc* (*Suc* (*mInside H NF L x*))) *x*

thm Hypermap.lemma_bound_mInside:

∀ (*H::?'a::type hypermap*) (*NF::?'a::type loop* ⇒ *bool*) (*L::?'a::type loop*) *x::?'a::type*. *is_split_condition H NF L x* ⟶ *Suc* (*mInside H NF L x*) < *CARD* (*face H x*)

thm Hypermap.lemma_congruence_on_face:

∀ (*H::?'a::type hypermap*) (*x::?'a::type*) (*n::nat*) *m::nat*. *n* < *CARD* (*face H x*) ∧ *POWER* (*face_map H*) *n x* = *POWER* (*face_map H*) *m x* ⟶ (∃ *q::nat*. *m* = *q* ∗ *CARD* (*face H x*) + *n*)

thm DEF_dart_inside:

dart_inside = (λ(*_2419829::?'a::type hypermap*) (*_2419830::?'a::type loop* ⇒ *bool*) (*_2419831::?'a::type loop*) *_2419832::?'a::type*. *GSPEC* (λ*GEN%PVAR%277::?'a::type*. ∃ *i::nat*. *SETSPEC GEN%PVAR%277* ((*1::nat*) ≤ *i* ∧ *i* ≤ *mInside _2419829 _2419830 _2419831 _2419832*) (*POWER* (*face_map _2419829*) *i _2419832*)))

thm Hypermap.dart_inside:

∀ (*H::?'a::type hypermap*) (*NF::?'a::type loop* ⇒ *bool*) (*L::?'a::type loop*) *x::?'a::type*. *dart_inside H NF L x* = *GSPEC* (λ*GEN%PVAR%277::?'a::type*. ∃ *i::nat*. *SETSPEC GEN%PVAR%277* ((*1::nat*) ≤ *i* ∧ *i* ≤ *mInside H NF L x*) (*POWER* (*face_map H*) *i x*))

thm Hypermap.lemma_dart_inside_sub_loop:

∀ (*H::?'a::type hypermap*) (*NF::?'a::type loop* ⇒ *bool*) (*L::?'a::type loop*) *x::?'a::type*. *is_split_condition H NF L x* ⟶ *SUBSET* (*dart_inside H NF L x*) (*dart_of L*)

thm DEF_canon_flag:

canon_flag = (λ(*_2419867::?'a::type hypermap*) *_2419868::?'a::type loop* ⇒ *bool*. (∀ (*x::?'a::type*) *y::?'a::type*. *IN x* (*canon_darts _2419867 _2419868*) ∧

IN y (canon_darts _2419867 _2419868) ⟶ (∃ (p::nat ⇒ ?'a::type) k::nat. p (0::nat) = x ∧ p k = y ∧ is_contour _2419867 p k ∧ SUBSET (support_list p k) (canon_darts _2419867 _2419868))) ∧ (∀ (L::?'a::type loop) x::?'a::type. IN L _2419868 ∧ ¬ IN L (canon _2419867 _2419868) ∧ belong x L ⟶ IN (edge_map _2419867 (head _2419867 _2419868 x)) (canon_darts _2419867 _2419868)))

thm Hypermap.canon_flag:

∀ (H::?'a::type hypermap) NF::?'a::type loop ⇒ bool. canon_flag H NF = ((∀ (x::?'a::type) y::?'a::type. IN x (canon_darts H NF) ∧ IN y (canon_darts H NF) ⟶ (∃ (p::nat ⇒ ?'a::type) k::nat. p (0::nat) = x ∧ p k = y ∧ is_contour H p k ∧ SUBSET (support_list p k) (canon_darts H NF))) ∧ (∀ (L::?'a::type loop) x::?'a::type. IN L NF ∧ ¬ IN L (canon H NF) ∧ belong x L ⟶ IN (edge_map H (head H NF x)) (canon_darts H NF)))

thm DEF_flag:

flag = (λ(_2419879::?'a::type hypermap) (_2419880::?'a::type loop ⇒ bool) (_2419881::?'a::type loop) _2419882::?'a::type. (∀ (u::?'a::type) v::?'a::type. IN u (canon_darts _2419879 _2419880) ∧ IN v (canon_darts _2419879 _2419880) ⟶ (∃ (p::nat ⇒ ?'a::type) k::nat. p (0::nat) = u ∧ p k = v ∧ is_contour _2419879 p k ∧ SUBSET (support_list p k) (canon_darts _2419879 _2419880))) ∧ (∀ (L'::?'a::type loop) y::?'a::type. IN L' _2419880 ∧ ¬ IN L' (canon _2419879 _2419880) ∧ belong y L' ∧ ¬ IN (head _2419879 _2419880 y) (dart_inside _2419879 _2419880 _2419881 _2419882) ⟶ IN (edge_map _2419879 (head _2419879 _2419880 y)) (HOL_Light_Import.UNION (canon_darts _2419879 _2419880) (dart_inside _2419879 _2419880 _2419881 _2419882))))

thm Hypermap.flag:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. flag H NF L x = ((∀ (u::?'a::type) v::?'a::type. IN u (canon_darts H NF) ∧ IN v (canon_darts H NF) ⟶ (∃ (p::nat ⇒ ?'a::type) k::nat. p (0::nat) = u ∧ p k = v ∧ is_contour H p k ∧ SUBSET (support_list p k) (canon_darts H NF))) ∧ (∀ (L'::?'a::type loop) y::?'a::type. IN L' NF ∧ ¬ IN L' (canon H NF) ∧ belong y L' ∧ ¬ IN (head H NF y) (dart_inside H NF L x) ⟶ IN (edge_map H (head H NF y)) (HOL_Light_Import.UNION (canon_darts H NF) (dart_inside H NF L x))))

thm DEF_heading:

heading = (λ(_2419911::?'a::type hypermap) (_2419912::?'a::type loop ⇒ bool) (_2419913::?'a::type loop) _2419914::?'a::type. POWER (face_map _2419911) (Suc (mInside _2419911 _2419912 _2419913 _2419914)) _2419914)

thm Hypermap.heading:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. heading H NF L x = POWER (face_map H) (Suc (mInside H NF L x)) x

thm Hypermap.lemma_loop_eq_face:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (x::?'a::type)\ n::nat.\ (1::nat) \leq n \wedge belong\ x\ L \wedge (\forall i \leq n.\ POWER\ (next\ L)\ i\ x = POWER\ (face\_map\ H)\ i\ x) \wedge POWER\ (next\ L)\ n\ x = x \longrightarrow dart\_of\ L = face\ H\ x \wedge HOL\_Light\_Import.size\ L \leq n$

thm Hypermap.lemma_on_heading:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_split\_condition\ H\ NF\ L\ x \longrightarrow belong\ (heading\ H\ NF\ L\ x)\ L \wedge next\ L\ (heading\ H\ NF\ L\ x) = HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x) \wedge node\ H\ (heading\ H\ NF\ L\ x) \neq node\ H\ x$

thm Hypermap.lemma_face_contour_on_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ m::nat.\ is\_restricted\ H \wedge is\_normal\ H\ NF \wedge IN\ L\ NF \wedge belong\ x\ L \wedge head\ H\ NF\ x = x \wedge (\forall i \leq Suc\ m.\ POWER\ (next\ L)\ i\ x = POWER\ (face\_map\ H)\ i\ x) \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq m \longrightarrow atom\ H\ L\ (POWER\ (next\ L)\ i\ x) = INSERT\ (POWER\ (face\_map\ H)\ i\ x)\ EMPTY) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq m \longrightarrow POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ i\ (atom\ H\ L\ x) = INSERT\ (POWER\ (face\_map\ H)\ i\ x)\ EMPTY) \wedge POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ (Suc\ m)\ (atom\ H\ L\ x) = atom\ H\ L\ (POWER\ (face\_map\ H)\ (Suc\ m)\ x)$

thm Hypermap.lemma_atom_on_inside_dart:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_split\_condition\ H\ NF\ L\ x \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq mInside\ H\ NF\ L\ x \longrightarrow POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ i\ (atom\ H\ L\ x) = INSERT\ (POWER\ (face\_map\ H)\ i\ x)\ EMPTY) \wedge POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ (Suc\ (mInside\ H\ NF\ L\ x))\ (atom\ H\ L\ x) = atom\ H\ L\ (heading\ H\ NF\ L\ x)$

thm Hypermap.lemma_mInside_and_length_cycle:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_split\_condition\ H\ NF\ L\ x \longrightarrow Suc\ (mInside\ H\ NF\ L\ x) < CARD\ (cycle\ H\ L)$

thm Hypermap.lemma_mAdd_Exists:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_split\_condition\ H\ NF\ L\ x \longrightarrow (\exists p::nat.\ (\forall i::nat.\ (1::nat) \leq i \wedge i \leq p \longrightarrow \neg\ IN\ (POWER\ (face\_map\ H)\ i\ (heading\ H\ NF\ L\ x))\ (support\_darts\ NF)) \wedge IN\ (POWER\ (face\_map\ H)\ (Suc\ p)\ (heading\ H\ NF\ L\ x))\ (support\_darts\ NF))$

thm DEF_mAdd:

$mAdd = (SOME\ p::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type\ loop \Rightarrow ?'a::type \Rightarrow nat.\ \forall (\_2420039::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_split\_condition\ H\ NF\ L\ x \longrightarrow (\forall i::nat.\ (1::nat) \leq i \wedge i \leq p\ \_2420039\ H\ NF\ L\ x \longrightarrow \neg$

IN (POWER (face_map H) i (heading H NF L x)) (support_darts NF)) ∧ IN (POWER (face_map H) (Suc (p _2420039 H NF L x)) (heading H NF L x)) (support_darts NF)) (116::nat)

thm Hypermap.lemma_mAdd:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_split_condition H NF L x ⟶ (∀ i::nat. (1::nat) ≤ i ∧ i ≤ mAdd H NF L x ⟶ ¬ IN (POWER (face_map H) i (heading H NF L x)) (support_darts NF)) ∧ IN (POWER (face_map H) (Suc (mAdd H NF L x)) (heading H NF L x)) (support_darts NF)

thm DEF_is_marked:

is_marked = (λ(_2420040::?'a::type hypermap) (_2420041::?'a::type loop ⇒ bool) (_2420042::?'a::type loop) _2420043::?'a::type. is_restricted _2420040 ∧ is_normal _2420040 _2420041 ∧ IN _2420042 _2420041 ∧ belong _2420043 _2420042 ∧ next _2420042 _2420043 = face_map _2420040 _2420043 ∧ simple_hypermap (HOL_Light_Import.quotient _2420040 _2420041) ∧ is_node_nondegenerate (HOL_Light_Import.quotient _2420040 _2420041) ∧ IN (edge_map _2420040 _2420043) (canon_darts _2420040 _2420041) ∧ (IN _2420042 (canon _2420040 _2420041) ⟶ canon_flag _2420040 _2420041) ∧ (¬ IN _2420042 (canon _2420040 _2420041) ⟶ flag _2420040 _2420041 _2420042 _2420043))

thm Hypermap.is_marked:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_marked H NF L x = (is_restricted H ∧ is_normal H NF ∧ IN L NF ∧ belong x L ∧ next L x = face_map H x ∧ simple_hypermap (HOL_Light_Import.quotient H NF) ∧ is_node_nondegenerate (HOL_Light_Import.quotient H NF) ∧ IN (edge_map H x) (canon_darts H NF) ∧ (IN L (canon H NF) ⟶ canon_flag H NF) ∧ (¬ IN L (canon H NF) ⟶ flag H NF L x))

thm Hypermap.lemma_marked_dart:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_marked H NF L x ⟶ head H NF x = x ∧ IN (HOL_Light_Import.inverse (node_map H) x) (canon_darts H NF)

thm Hypermap.lemma_split_marked_loop:

∀ (H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_marked H NF L x ∧ ¬ IN L (canon H NF) ⟶ is_split_condition H NF L x

thm DEF_attach:

attach = (λ(_2420074::?'a::type hypermap) (_2420075::?'a::type loop ⇒ bool) (_2420076::?'a::type loop) _2420077::?'a::type. POWER (face_map _2420074) (Suc (mAdd _2420074 _2420075 _2420076 _2420077)) (heading _2420074 _2420075 _2420076 _2420077))

thm Hypermap.attach:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $attach\ H\ NF\ L\ x = POWER\ (face\_map\ H)\ (Suc\ (mAdd\ H\ NF\ L\ x))\ (heading\ H\ NF\ L\ x)$

thm Hypermap.lemma_new_darts_in_face:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $IN\ (heading\ H\ NF\ L\ x)\ (face\ H\ x) \wedge IN\ (attach\ H\ NF\ L\ x)\ (face\ H\ x)$

thm Hypermap.lemma_on_attach:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $is\_split\_condition\ H\ NF\ L\ x \longrightarrow node\ H\ (heading\ H\ NF\ L\ x) \neq node\ H\ (attach\ H\ NF\ L\ x) \wedge Suc\ (mAdd\ H\ NF\ L\ x) < CARD\ (face\ H\ x)$

thm Hypermap.lemmaLoopSeparation:

$\forall (H::?'a::type\ hypermap)\ (L::?'a::type\ loop)\ (p::nat \Rightarrow ?'a::type)\ k::nat.\ is\_loop\ H\ L \wedge (1::nat) \leq k \wedge is\_contour\ H\ p\ k \wedge belong\ (p\ (0::nat))\ L \wedge p\ (1::nat) = face\_map\ H\ (p\ (0::nat)) \wedge (\forall i::nat.\ (1::nat) \leq i \wedge i \leq k \longrightarrow \neg\ belong\ (p\ i)\ L) \wedge node\ H\ (p\ (0::nat)) \neq node\ H\ (p\ k) \wedge (\exists y::?'a::type.\ IN\ y\ (node\ H\ (p\ k)) \wedge belong\ y\ L) \longrightarrow \neg\ planar\_hypermap\ H$

thm Hypermap.lemmaHQYMRTX:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow belong\ (attach\ H\ NF\ L\ x)\ L \wedge (\forall k::nat.\ (1::nat) \leq k \wedge k \leq Suc\ (mInside\ H\ NF\ L\ x) \longrightarrow attach\ H\ NF\ L\ x \neq POWER\ (face\_map\ H)\ k\ x)$

thm Hypermap.lemma_route_exists:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow (\exists q>mInside\ H\ NF\ L\ x.\ q < CARD\ (cycle\ H\ L) \wedge POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ (Suc\ q)\ (atom\ H\ L\ x) = atom\ H\ L\ (attach\ H\ NF\ L\ x))$

thm DEF_mRoute:

$mRoute = (SOME\ q::nat \Rightarrow ?'a::type\ hypermap \Rightarrow (?'a::type\ loop \Rightarrow bool) \Rightarrow ?'a::type\ loop \Rightarrow ?'a::type \Rightarrow nat.\ \forall (\_2420323::nat)\ (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow mInside\ H\ NF\ L\ x < q\ \_2420323\ H\ NF\ L\ x \wedge q\ \_2420323\ H\ NF\ L\ x < CARD\ (cycle\ H\ L) \wedge POWER\ (face\_map\ (HOL\_Light\_Import.quotient\ H\ NF))\ (Suc\ (q\ \_2420323\ H\ NF\ L\ x))\ (atom\ H\ L\ x) = atom\ H\ L\ (attach\ H\ NF\ L\ x))\ (117::nat)$

thm Hypermap.lemma_route:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.$ $is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow mInside\ H\ NF\ L\ x < mRoute\ H\ NF\ L\ x \wedge mRoute\ H\ NF\ L\ x < CARD\ (cycle\ H\ L) \wedge POWER\ (face\_map$

(*HOL_Light_Import.quotient H NF*)) (*Suc* (*mRoute H NF L x*)) (*atom H L x*) = *atom H L* (*attach H NF L x*)

thm Hypermap.lemmaParameters:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_marked H NF L x* $\wedge$ $\neg$ *IN L* (*canon H NF*) $\longrightarrow$ *mInside H NF L x* < *mRoute H NF L x* $\wedge$ *mRoute H NF L x* < *CARD* (*cycle H L*) $\wedge$ *Suc* (*mInside H NF L x*) < *mAdd H NF L x* + *mRoute H NF L x* $\wedge$ *node H* (*heading H NF L x*) $\neq$ *node H x* $\wedge$ *node H* (*heading H NF L x*) $\neq$ *node H* (*attach H NF L x*)

thm DEF_genex:

genex = (λ(*_2420446*::*?'a*::*type hypermap*) (*_2420447*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*_2420448*::*?'a*::*type loop*) *_2420449*::*?'a*::*type*. *glue* (*loop_path _2420448* (*attach _2420446 _2420447 _2420448 _2420449*)) (*face_contour _2420446* (*heading _2420446 _2420447 _2420448 _2420449*)) (*index _2420448* (*attach _2420446 _2420447 _2420448 _2420449*) (*heading _2420446 _2420447 _2420448 _2420449*)))

thm Hypermap.genex:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *genex H NF L x* = *glue* (*loop_path L* (*attach H NF L x*)) (*face_contour H* (*heading H NF L x*)) (*index L* (*attach H NF L x*) (*heading H NF L x*))

thm DEF_tpx:

tpx = (λ(*_2420478*::*?'a*::*type hypermap*) (*_2420479*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*_2420480*::*?'a*::*type loop*) *_2420481*::*?'a*::*type*. *index _2420480* (*attach _2420478 _2420479 _2420480 _2420481*) (*heading _2420478 _2420479 _2420480 _2420481*) + *mAdd _2420478 _2420479 _2420480 _2420481*)

thm Hypermap.tpx:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *tpx H NF L x* = *index L* (*attach H NF L x*) (*heading H NF L x*) + *mAdd H NF L x*

thm DEF_geney:

geney = (λ(*_2420510*::*?'a*::*type hypermap*) (*_2420511*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*_2420512*::*?'a*::*type loop*) *_2420513*::*?'a*::*type*. *glue* (*loop_path _2420512* (*HOL_Light_Import.inverse* (*node_map _2420510*) (*heading _2420510 _2420511 _2420512 _2420513*))) (*complement _2420510* (*attach _2420510 _2420511 _2420512 _2420513*)) (*index _2420512* (*HOL_Light_Import.inverse* (*node_map _2420510*) (*heading _2420510 _2420511 _2420512 _2420513*)) (*node_map _2420510* (*attach _2420510 _2420511 _2420512 _2420513*))))

thm Hypermap.geney:

$\forall$ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* $\Rightarrow$ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *geney H NF L x* = *glue* (*loop_path L* (*HOL_Light_Import.inverse* (*node_map*

H) (*heading H NF L x*))) (*complement H* (*attach H NF L x*)) (*index L* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) (*node_map H* (*attach H NF L x*)))

thm DEF_tpy:

tpy = (λ(*_2420542*::*?'a*::*type hypermap*) (*_2420543*::*?'a*::*type loop* ⇒ *bool*) (*_2420544*::*?'a*::*type loop*) *_2420545*::*?'a*::*type*. *index _2420544* (*HOL_Light_Import.inverse* (*node_map _2420542*) (*heading _2420542 _2420543 _2420544 _2420545*)) (*node_map _2420542* (*attach _2420542 _2420543 _2420544 _2420545*)) + *ind _2420542* (*attach _2420542 _2420543 _2420544 _2420545*) (*mAdd _2420542 _2420543 _2420544 _2420545*))

thm Hypermap.tpy:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *tpy H NF L x* = *index L* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) (*node_map H* (*attach H NF L x*)) + *ind H* (*attach H NF L x*) (*mAdd H NF L x*)

thm DEF_dnax:

dnax = (λ(*_2420574*::*?'a*::*type hypermap*) (*_2420575*::*?'a*::*type loop* ⇒ *bool*) (*_2420576*::*?'a*::*type loop*) *_2420577*::*?'a*::*type*. *loop* (*support_list* (*genex _2420574 _2420575 _2420576 _2420577*) (*tpx _2420574 _2420575 _2420576 _2420577*), *samsara* (*genex _2420574 _2420575 _2420576 _2420577*) (*tpx _2420574 _2420575 _2420576 _2420577*)))

thm Hypermap.dnax:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *dnax H NF L x* = *loop* (*support_list* (*genex H NF L x*) (*tpx H NF L x*), *samsara* (*genex H NF L x*) (*tpx H NF L x*))

thm DEF_dnay:

dnay = (λ(*_2420606*::*?'a*::*type hypermap*) (*_2420607*::*?'a*::*type loop* ⇒ *bool*) (*_2420608*::*?'a*::*type loop*) *_2420609*::*?'a*::*type*. *loop* (*support_list* (*geney _2420606 _2420607 _2420608 _2420609*) (*tpy _2420606 _2420607 _2420608 _2420609*), *samsara* (*geney _2420606 _2420607 _2420608 _2420609*) (*tpy _2420606 _2420607 _2420608 _2420609*)))

thm Hypermap.dnay:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *dnay H NF L x* = *loop* (*support_list* (*geney H NF L x*) (*tpy H NF L x*), *samsara* (*geney H NF L x*) (*tpy H NF L x*))

thm Hypermap.lemma_genex_loop:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ⟶ *is_inj_contour H* (*genex H NF L x*) (*tpx H NF L x*) ∧ *face_map H* (*genex H NF L x* (*tpx H NF L x*)) = *genex H NF L x* (*0*::*nat*)

thm Hypermap.complement_index:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (m::nat)\ k::nat.\ is\_node\_nondegenerate\ H \wedge IN\ x\ (dart\ H) \wedge (1::nat) \leq k \wedge k \leq ind\ H\ x\ m \longrightarrow (\exists (i::nat)\ j::nat.\ i < m \wedge (1::nat) \leq j \wedge j < CARD\ (node\ H\ (POWER\ (HOL\_Light\_Import.inverse\ (face\_map\ H))\ (Suc\ i)\ x)) \wedge k = ind\ H\ x\ i + j)$

thm Hypermap.reduce_exponent:

$\forall (s::?'a::type \Rightarrow bool)\ (p::?'a::type \Rightarrow ?'a::type)\ (m::nat)\ (n::nat)\ x::?'a::type.\ permutes\ p\ s \wedge m \leq n \longrightarrow POWER\ (HOL\_Light\_Import.inverse\ p)\ m\ (POWER\ p\ n\ x) = POWER\ p\ (n - m)\ x$

thm Hypermap.lemma_on_adding_darts:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow next\ L\ (heading\ H\ NF\ L\ x) = HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x) \wedge back\ L\ (attach\ H\ NF\ L\ x) = node\_map\ H\ (attach\ H\ NF\ L\ x)$

thm Hypermap.lemma_geney_loop:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow is\_inj\_contour\ H\ (geney\ H\ NF\ L\ x)\ (tpy\ H\ NF\ L\ x) \wedge face\_map\ H\ (geney\ H\ NF\ L\ x\ (tpy\ H\ NF\ L\ x)) = geney\ H\ NF\ L\ x\ (0::nat)$

thm DEF_genesis:

$genesis = (\lambda(\_2420680::?'a::type\ hypermap)\ (\_2420681::?'a::type\ loop \Rightarrow bool)\ (\_2420682::?'a::type\ loop)\ \_2420683::?'a::type.\ HOL\_Light\_Import.UNION\ (DELETE\ \_2420681\ \_2420682)\ (INSERT\ (dnax\ \_2420680\ \_2420681\ \_2420682\ \_2420683)\ (INSERT\ (dnay\ \_2420680\ \_2420681\ \_2420682\ \_2420683)\ EMPTY)))$

thm Hypermap.genesis:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ genesis\ H\ NF\ L\ x = HOL\_Light\_Import.UNION\ (DELETE\ NF\ L)\ (INSERT\ (dnax\ H\ NF\ L\ x)\ (INSERT\ (dnay\ H\ NF\ L\ x)\ EMPTY))$

thm Hypermap.lemma_in_couple:

$\forall (x::?'a::type)\ (a::?'a::type)\ b::?'a::type.\ IN\ x\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = (x = a \vee x = b)$

thm Hypermap.lemma_on_dnax:

$\forall (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \wedge \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow genex\ H\ NF\ L\ x\ (0::nat) = attach\ H\ NF\ L\ x \wedge belong\ (attach\ H\ NF\ L\ x)\ (dnax\ H\ NF\ L\ x) \wedge HOL\_Light\_Import.top\ (dnax\ H\ NF\ L\ x) = tpx\ H\ NF\ L\ x \wedge (\forall i \leq index\ L\ (attach\ H\ NF\ L\ x)\ (heading\ H\ NF\ L\ x).\ POWER\ (next\ (dnax\ H\ NF\ L\ x))\ i\ (attach\ H\ NF\ L\ x) = POWER\ (next\ L)\ i\ (attach\ H\ NF\ L\ x)) \wedge (\forall i \leq mAdd\ H\ NF\ L\ x.\ POWER\ (next\ (dnax$

H NF L x)) (*index L* (*attach H NF L x*) (*heading H NF L x*) + *i*) (*attach H NF L x*) = *POWER* (*face_map H*) *i* (*heading H NF L x*))

thm Hypermap.lemma_on_dnay:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ⟶ *geney H NF L x* (*0*::*nat*) = *HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*) ∧ *belong* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) (*dnay H NF L x*) ∧ *HOL_Light_Import.top* (*dnay H NF L x*) = *tpy H NF L x* ∧ (∀ *i*≤*index L* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) (*node_map H* (*attach H NF L x*)). *POWER* (*next* (*dnay H NF L x*)) *i* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) = *POWER* (*next L*) *i* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*))) ∧ (∀ *i*≤*ind H* (*attach H NF L x*) (*mAdd H NF L x*). *POWER* (*next* (*dnay H NF L x*)) (*index L* (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) (*node_map H* (*attach H NF L x*)) + *i*) (*HOL_Light_Import.inverse* (*node_map H*) (*heading H NF L x*)) = *complement H* (*attach H NF L x*) *i*)

thm Hypermap.lemma_node_outside_support_darts:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) *i*::*nat*. *is_split_condition H NF L x* ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *mAdd H NF L x* ⟶ (∀ *y*::*?'a*::*type*. *IN y* (*node H* (*POWER* (*face_map H*) *i* (*heading H NF L x*))) ⟶ ¬ *IN y* (*support_darts NF*))

thm Hypermap.lemma_in_dnax:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ⟶ (∀ *y*::*?'a*::*type*. *belong y* (*dnax H NF L x*) = ((∃ *i*≤*index L* (*attach H NF L x*) (*heading H NF L x*). *y* = *POWER* (*next L*) *i* (*attach H NF L x*)) ∨ (∃ *i*≥*1*::*nat*. *i* ≤ *mAdd H NF L x* ∧ *y* = *POWER* (*face_map H*) *i* (*heading H NF L x*))))

thm Hypermap.lemma_in_dnax1:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *i*::*nat*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ∧ *i* ≤ *index L* (*attach H NF L x*) (*heading H NF L x*) ∧ *y* = *POWER* (*next L*) *i* (*attach H NF L x*) ⟶ *belong y* (*dnax H NF L x*)

thm Hypermap.lemma_in_dnax2:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *i*::*nat*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *mAdd H NF L x* ∧ *y* = *POWER* (*face_map H*) *i* (*heading H NF L x*) ⟶ *belong y* (*dnax H NF L x*)

thm Hypermap.lemma_in_dnay:

∀ (*H*::*?'a*::*type hypermap*) (*NF*::*?'a*::*type loop* ⇒ *bool*) (*L*::*?'a*::*type loop*) *x*::*?'a*::*type*. *is_marked H NF L x* ∧ ¬ *IN L* (*canon H NF*) ⟶ (∀ *y*::*?'a*::*type*. *belong y*

$(dnay\ H\ NF\ L\ x) = ((\exists\, i \le index\ L\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x))\ (node\_map\ H\ (attach\ H\ NF\ L\ x)).\ y = POWER\ (next\ L)\ i\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x))) \lor (\exists\, (i::nat)\ j::nat.\ (1::nat) \le i \land i \le mAdd\ H\ NF\ L\ x \land (1::nat) \le j \land j < CARD\ (node\ H\ (POWER\ (face\_map\ H)\ i\ (heading\ H\ NF\ L\ x))) \land y = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ j\ (POWER\ (face\_map\ H)\ i\ (heading\ H\ NF\ L\ x)))))$

thm Hypermap.lemma_in_dnay1:

$\forall\, (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ i::nat.\ is\_marked\ H\ NF\ L\ x \land \neg\ IN\ L\ (canon\ H\ NF) \land i \le index\ L\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x))\ (node\_map\ H\ (attach\ H\ NF\ L\ x)) \land y = POWER\ (next\ L)\ i\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x)) \longrightarrow belong\ y\ (dnay\ H\ NF\ L\ x)$

thm Hypermap.lemma_in_dnay2:

$\forall\, (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ (i::nat)\ j::nat.\ is\_marked\ H\ NF\ L\ x \land \neg\ IN\ L\ (canon\ H\ NF) \land (1::nat) \le i \land i \le mAdd\ H\ NF\ L\ x \land (1::nat) \le j \land j < CARD\ (node\ H\ (POWER\ (face\_map\ H)\ i\ (heading\ H\ NF\ L\ x))) \land y = POWER\ (HOL\_Light\_Import.inverse\ (node\_map\ H))\ j\ (POWER\ (face\_map\ H)\ i\ (heading\ H\ NF\ L\ x)) \longrightarrow belong\ y\ (dnay\ H\ NF\ L\ x)$

thm Hypermap.lemma_disjoint_new_loops:

$\forall\, (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \land \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow index\ L\ (HOL\_Light\_Import.inverse\ (node\_map\ H)\ (heading\ H\ NF\ L\ x))\ (node\_map\ H\ (attach\ H\ NF\ L\ x)) + Suc\ (index\ L\ (attach\ H\ NF\ L\ x)\ (heading\ H\ NF\ L\ x)) = HOL\_Light\_Import.top\ L \land (\forall\, u::?'a::type.\ belong\ u\ (dnax\ H\ NF\ L\ x) \longrightarrow \neg\ belong\ u\ (dnay\ H\ NF\ L\ x)) \land (\forall\, u::?'a::type.\ belong\ u\ (dnay\ H\ NF\ L\ x) \longrightarrow \neg\ belong\ u\ (dnax\ H\ NF\ L\ x)) \land \neg\ IN\ (dnax\ H\ NF\ L\ x)\ NF \land \neg\ IN\ (dnay\ H\ NF\ L\ x)\ NF \land (\forall\, u::?'a::type.\ belong\ u\ L \longrightarrow belong\ u\ (dnax\ H\ NF\ L\ x) \lor belong\ u\ (dnay\ H\ NF\ L\ x))$

thm Hypermap.lemma_normal_genesis:

$\forall\, (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ x::?'a::type.\ is\_marked\ H\ NF\ L\ x \land \neg\ IN\ L\ (canon\ H\ NF) \longrightarrow is\_normal\ H\ (genesis\ H\ NF\ L\ x)$

thm Hypermap.lemma_separation_on_loop:

$\forall\, (H::?'a::type\ hypermap)\ (NF::?'a::type\ loop \Rightarrow bool)\ (L::?'a::type\ loop)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ is\_restricted\ H \land is\_normal\ H\ NF \land IN\ L\ NF \land belong\ z\ L \land belong\ x\ L \land belong\ y\ L \land head\ H\ NF\ x = x \land index\ L\ z\ (head\ H\ NF\ z) < index\ L\ z\ y \land index\ L\ z\ y \le index\ L\ z\ x \longrightarrow index\ L\ z\ (head\ H\ NF\ z) < index\ L\ z\ (tail\ H\ NF\ y) \land index\ L\ z\ (tail\ H\ NF\ y) \le index\ L\ z\ y \land index\ L\ z\ y \le index\ L\ z\ (head\ H\ NF\ y) \land index\ L\ z\ (head\ H\ NF\ y) \le index\ L\ z\ x$

thm Hypermap.atom_eq:

∀(H::?'a::type hypermap) (N1::?'a::type loop ⇒ bool) (N2::?'a::type loop ⇒ bool) (L1::?'a::type loop) (L2::?'a::type loop) (x::?'a::type) (y::?'a::type) (m::nat) n::nat. is_restricted H ∧ is_normal H N1 ∧ is_normal H N2 ∧ IN L1 N1 ∧ IN L2 N2 ∧ belong x L1 ∧ belong x L2 ∧ belong y L1 ∧ n ≤ HOL_Light_Import.top L1 ∧ n ≤ HOL_Light_Import.top L2 ∧ m < index L1 x (tail H N1 y) ∧ index L1 x (tail H N1 y) ≤ index L1 x (head H N1 y) ∧ index L1 x (head H N1 y) < n ∧ (∀i::nat. m ≤ i ∧ i ≤ n ⟶ POWER (next L2) i x = POWER (next L1) i x) ⟶ tail H N2 y = tail H N1 y ∧ head H N2 y = head H N1 y ∧ atom H L2 y = atom H L1 y

thm Hypermap.lemma_dnax_atomic_structure:

∀(H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) x::?'a::type. is_marked H NF L x ∧ ¬ IN L (canon H NF) ⟶ index L (attach H NF L x) (head H NF (attach H NF L x)) ≤ index L (attach H NF L x) x ∧ index L (attach H NF L x) x + Suc (mInside H NF L x) = index L (attach H NF L x) (heading H NF L x) ∧ IN (dnax H NF L x) (genesis H NF L x) ∧ head H (genesis H NF L x) (attach H NF L x) = head H NF (attach H NF L x) ∧ tail H (genesis H NF L x) (attach H NF L x) = attach H NF L x ∧ (∀i≤index L (attach H NF L x) (head H NF (attach H NF L x)). atom H (dnax H NF L x) (POWER (next L) i (attach H NF L x)) = atom H (dnax H NF L x) (attach H NF L x)) ∧ (∀i::nat. index L (attach H NF L x) (head H NF (attach H NF L x)) < i ∧ i ≤ index L (attach H NF L x) x ⟶ tail H (genesis H NF L x) (POWER (next L) i (attach H NF L x)) = tail H NF (POWER (next L) i (attach H NF L x)) ∧ head H (genesis H NF L x) (POWER (next L) i (attach H NF L x)) = head H NF (POWER (next L) i (attach H NF L x)) ∧ atom H (dnax H NF L x) (POWER (next L) i (attach H NF L x)) = atom H L (POWER (next L) i (attach H NF L x))) ∧ (∀i::nat. index L (attach H NF L x) x < i ∧ i ≤ index L (attach H NF L x) (heading H NF L x) ⟶ atom H (dnax H NF L x) (POWER (next L) i (attach H NF L x)) = INSERT (POWER (next L) i (attach H NF L x)) EMPTY) ∧ (∀i::nat. (1::nat) ≤ i ∧ i ≤ mAdd H NF L x ⟶ atom H (dnax H NF L x) (POWER (face_map H) i (heading H NF L x)) = INSERT (POWER (face_map H) i (heading H NF L x)) EMPTY) ∧ (∀i≤Suc (mInside H NF L x) + Suc (mAdd H NF L x). POWER (next (dnax H NF L x)) i x = POWER (face_map H) i x)

thm Hypermap.go_into_atom:

∀(H::?'a::type hypermap) (NF::?'a::type loop ⇒ bool) (L::?'a::type loop) (x::?'a::type) y::?'a::type. is_normal H NF ∧ IN L NF ∧ belong x L ∧ belong y L ∧ ¬ IN y (atom H L x) ⟶ index L y (tail H NF x) ≤ index L y x

thm Hypermap.square_edge_convolution:

∀H::?'a::type hypermap. plain_hypermap H ⟶ (∀x::?'a::type. node_map H (face_map H (node_map H (face_map H x))) = x)

thm Hypermap.square_edge_convolution2:

∀H::?'a::type hypermap. plain_hypermap H ⟶ (∀x::?'a::type. face_map H (node_map H (face_map H (node_map H x))) = x)

thm SYM_ALT:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x) = (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y = R\ y\ x)$

thm TRANS_ALT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (S::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ U::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (x::?'a::type)\ z::?'a::type. (\exists y::?'a::type.\ R\ x\ y \wedge S\ y\ z) \longrightarrow U\ x\ z) = (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge S\ y\ z \longrightarrow U\ x\ z)$

thm DEF_RC:

$RC = (\lambda(R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type. \forall RC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (a0::?'a::type)\ a1::?'a::type.\ R\ a0\ a1 \vee a1 = a0 \longrightarrow RC'\ a0\ a1) \longrightarrow RC'\ a0\ a1)$

thm RC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type.\ RC\ R\ a0\ a1 = (R\ a0\ a1 \vee a1 = a0)$

thm RC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ RC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RC'\ x\ y) \wedge (\forall x::?'a::type.\ RC'\ x\ x) \longrightarrow (\forall (a0::?'a::type)\ a1::?'a::type.\ RC\ R\ a0\ a1 \longrightarrow RC'\ a0\ a1)$

thm RC_RULES:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RC\ R\ x\ y) \wedge (\forall x::?'a::type.\ RC\ R\ x\ x)$

thm RC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RC\ R\ x\ y$

thm RC_REFL:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ RC\ R\ x\ x$

thm RC_EXPLICIT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y = (R\ x\ y \vee x = y)$

thm RC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y \longrightarrow RC\ S\ x\ y)$

thm RC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (RC\ R = R) = (\forall x::?'a::type.\ R\ x\ x)$

thm RC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (RC\ R) = RC\ R$

thm RC_SYM:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y \longrightarrow RC\ R\ y\ x)$

thm RC_TRANS:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z) \longrightarrow (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ RC\ R\ x\ y \wedge RC\ R\ y\ z \longrightarrow RC\ R\ x\ z)$

thm DEF_SC:

$SC = (\lambda(R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type.\ \forall SC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (a0::?'a::type)\ a1::?'a::type.\ R\ a0\ a1 \vee SC'\ a1\ a0 \longrightarrow SC'\ a0\ a1) \longrightarrow SC'\ a0\ a1)$

thm SC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type.\ SC\ R\ a0\ a1 = (R\ a0\ a1 \vee SC\ R\ a1\ a0)$

thm SC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ SC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow SC'\ x\ y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ SC'\ x\ y \longrightarrow SC'\ y\ x) \longrightarrow (\forall (a0::?'a::type)\ a1::?'a::type.\ SC\ R\ a0\ a1 \longrightarrow SC'\ a0\ a1)$

thm SC_RULES:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow SC\ R\ x\ y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow SC\ R\ y\ x)$

thm SC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow SC\ R\ x\ y$

thm SC_SYM:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow SC\ R\ y\ x$

thm SC_EXPLICIT:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ SC\ R\ (?x::?'a::type)\ (?y::?'a::type) = (R\ ?x\ ?y \vee R\ ?y\ ?x)$

thm SC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow SC\ S\ x\ y)$

thm SC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (SC\ R = R) = (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x)$

thm SC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ SC\ (SC\ R) = SC\ R$

thm SC_REFL:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ R\ x\ x) \longrightarrow (\forall x::?'a::type.\ SC\ R\ x\ x)$

thm DEF_TC:

$TC = (\lambda(R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type.\ \forall TC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (a0::?'a::type)\ a1::?'a::type.\ R\ a0\ a1 \lor (\exists y::?'a::type.\ TC'\ a0\ y \land TC'\ y\ a1) \longrightarrow TC'\ a0\ a1) \longrightarrow TC'\ a0\ a1)$

thm TC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (a0::?'a::type)\ a1::?'a::type.\ TC\ R\ a0\ a1 = (R\ a0\ a1 \lor (\exists y::?'a::type.\ TC\ R\ a0\ y \land TC\ R\ y\ a1))$

thm TC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ TC'::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow TC'\ x\ y) \land (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ TC'\ x\ y \land TC'\ y\ z \longrightarrow TC'\ x\ z) \longrightarrow (\forall (a0::?'a::type)\ a1::?'a::type.\ TC\ R\ a0\ a1 \longrightarrow TC'\ a0\ a1)$

thm TC_RULES:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow TC\ R\ x\ y) \land (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ TC\ R\ x\ y \land TC\ R\ y\ z \longrightarrow TC\ R\ x\ z)$

thm TC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow TC\ R\ x\ y$

thm TC_TRANS:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ TC\ R\ x\ y \land TC\ R\ y\ z \longrightarrow TC\ R\ x\ z$

thm TC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow TC\ S\ x\ y)$

thm TC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool. (TC\ R = R) = (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z)$

thm TC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ TC\ (TC\ R) = TC\ R$

thm TC_REFL:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ R\ x\ x) \longrightarrow (\forall x::?'a::type.\ TC\ R\ x\ x)$

thm TC_SYM:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow TC\ R\ y\ x)$

thm RC_SC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (SC\ R) = SC\ (RC\ R)$

thm SC_RC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ SC\ (RC\ R) = RC\ (SC\ R)$

thm RC_TC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (TC\ R) = TC\ (RC\ R)$

thm TC_RC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ TC\ (RC\ R) = RC\ (TC\ R)$

thm TC_SC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ SC\ (TC\ R)\ x\ y \longrightarrow TC\ (SC\ R)\ x\ y$

thm TC_TRANS_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ TC\ R\ x\ y \wedge R\ y\ z \longrightarrow TC\ R\ x\ z$

thm TC_TRANS_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge TC\ R\ y\ z \longrightarrow TC\ R\ x\ z$

thm TC_CASES_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ TC\ R\ x\ z = (R\ x\ z \vee (\exists y::?'a::type.\ TC\ R\ x\ y \wedge R\ y\ z))$

thm TC_CASES_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ TC\ R\ x\ z = (R\ x\ z \vee (\exists y::?'a::type.\ R\ x\ y \wedge TC\ R\ y\ z))$

thm TC_INDUCT_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ P\ x\ y \wedge R\ y\ z \longrightarrow P\ x\ z) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow P\ x\ y)$

thm TC_INDUCT_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall (x::?'a::type)\ z::?'a::type.\ (\exists y::?'a::type.\ R\ x\ y \wedge P\ y\ z) \longrightarrow P\ x\ z) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow P\ x\ y)$

thm DEF_RSC:

$RSC = (\lambda\_2422541::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (SC\ \_2422541))$

thm RSC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RSC\ R = RC\ (SC\ R)$

thm RSC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RSC\ R\ x\ y$

thm RSC_REFL:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ RSC\ R\ x\ x$

thm RSC_SYM:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RSC\ R\ x\ y \longrightarrow RSC\ R\ y\ x$

thm RSC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RSC\ R\ x\ y = (x = y \vee R\ x\ y \vee R\ y\ x)$

thm RSC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall x::?'a::type.\ P\ x\ x) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ P\ x\ y \longrightarrow P\ y\ x) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RSC\ R\ x\ y \longrightarrow P\ x\ y)$

thm RSC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RSC\ R\ x\ y \longrightarrow RSC\ S\ x\ y)$

thm RSC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (RSC\ R = R) = ((\forall x::?'a::type.\ R\ x\ x) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x))$

thm RSC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RSC\ (RSC\ R) = RSC\ R$

thm DEF_RTC:

$RTC = (\lambda\_2422665::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (TC\ \_2422665))$

thm RTC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RTC\ R = RC\ (TC\ R)$

thm RTC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RTC\ R\ x\ y$

thm RTC_REFL:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ RTC\ R\ x\ x$

thm RTC_TRANS:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ y \wedge RTC\ R\ y\ z \longrightarrow RTC\ R\ x\ z$

thm RTC_RULES:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RTC\ R\ x\ y) \wedge (\forall x::?'a::type.\ RTC\ R\ x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ y \wedge RTC\ R\ y\ z \longrightarrow RTC\ R\ x\ z)$

thm RTC_TRANS_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ y \wedge R\ y\ z \longrightarrow RTC\ R\ x\ z$

thm RTC_TRANS_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge RTC\ R\ y\ z \longrightarrow RTC\ R\ x\ z$

thm RTC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ z = (x = z \vee (\exists y::?'a::type.\ RTC\ R\ x\ y \wedge RTC\ R\ y\ z))$

thm RTC_CASES_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ z = (x = z \vee (\exists y::?'a::type.\ RTC\ R\ x\ y \wedge R\ y\ z))$

thm RTC_CASES_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RTC\ R\ x\ z = (x = z \vee (\exists y::?'a::type.\ R\ x\ y \wedge RTC\ R\ y\ z))$

thm RTC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall x::?'a::type.\ P\ x\ x) \wedge (\forall (x::?'a::type)$

$(y::?'a::type)\ z::?'a::type.\ P\ x\ y \wedge P\ y\ z \longrightarrow P\ x\ z) \longrightarrow (\forall (x::?'a::type)$
$y::?'a::type.\ RTC\ R\ x\ y \longrightarrow P\ x\ y)$

thm RTC_INDUCT_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.$
$P\ x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ P\ x\ y \wedge R\ y\ z \longrightarrow P\ x\ z)$
$\longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \longrightarrow P\ x\ y)$

thm RTC_INDUCT_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.$
$P\ x\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge P\ y\ z \longrightarrow P\ x\ z)$
$\longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \longrightarrow P\ x\ y)$

thm RTC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)$
$y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \longrightarrow$
$RTC\ S\ x\ y)$

thm RTC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (RTC\ R = R) = ((\forall x::?'a::type.\ R\ x\ x) \wedge$
$(\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z))$

thm RTC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RTC\ (RTC\ R) = RTC\ R$

thm RTC_SYM:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R$
$y\ x) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \longrightarrow RTC\ R\ y\ x)$

thm RTC_STUTTER:

$RTC\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool) = RTC\ (\lambda(x::?'a::type)\ y::?'a::type.$
$?R\ x\ y \wedge x \neq y)$

thm TC_RTC_CASES_L:

$TC\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (?x::?'a::type)\ (?z::?'a::type) = (\exists y::?'a::type.$
$RTC\ ?R\ ?x\ y \wedge ?R\ y\ ?z)$

thm TC_RTC_CASES_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ TC\ R\ x\ z =$
$(\exists y::?'a::type.\ R\ x\ y \wedge RTC\ R\ y\ z)$

thm TC_TC_RTC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ TC\ R\ x\ z =$
$(\exists y::?'a::type.\ TC\ R\ x\ y \wedge RTC\ R\ y\ z)$

thm TC_RTC_TC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ TC\ R\ x\ z = (\exists y::?'a::type.\ RTC\ R\ x\ y \wedge TC\ R\ y\ z)$

thm RTC_NE_IMP_TC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \wedge x \neq y \longrightarrow TC\ R\ x\ y$

thm DEF_STC:

$STC = (\lambda\_2424176::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ TC\ (SC\ \_2424176))$

thm STC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ STC\ R = TC\ (SC\ R)$

thm STC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow STC\ R\ x\ y$

thm STC_SYM:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ STC\ R\ x\ y \longrightarrow STC\ R\ y\ x$

thm STC_TRANS:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ STC\ R\ x\ y \wedge STC\ R\ y\ z \longrightarrow STC\ R\ x\ z$

thm STC_TRANS_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ STC\ R\ x\ y \wedge R\ y\ z \longrightarrow STC\ R\ x\ z$

thm STC_TRANS_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge STC\ R\ y\ z \longrightarrow STC\ R\ x\ z$

thm STC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ STC\ R\ x\ z = (R\ x\ z \vee STC\ R\ z\ x \vee (\exists y::?'a::type.\ STC\ R\ x\ y \wedge STC\ R\ y\ z))$

thm STC_CASES_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ STC\ R\ x\ z = (R\ x\ z \vee STC\ R\ z\ x \vee (\exists y::?'a::type.\ STC\ R\ x\ y \wedge R\ y\ z))$

thm STC_CASES_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ STC\ R\ x\ z = (R\ x\ z \vee STC\ R\ z\ x \vee (\exists y::?'a::type.\ R\ x\ y \wedge STC\ R\ y\ z))$

thm STC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall (x::?'a::type)\ y::?'a::type.\ P\ x\ y \longrightarrow P\ y\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ P\ x\ y \wedge P\ y\ z \longrightarrow P\ x\ z) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ STC\ R\ x\ y \longrightarrow P\ x\ y)$

thm STC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ STC\ R\ x\ y \longrightarrow STC\ S\ x\ y)$

thm STC_CLOSED:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (STC\ R = R) = ((\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z))$

thm STC_IDEMP:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ STC\ (STC\ R) = STC\ R$

thm STC_REFL:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ R\ x\ x) \longrightarrow (\forall x::?'a::type.\ STC\ R\ x\ x)$

thm DEF_RSTC:

$RSTC = (\lambda\_2424468::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RC\ (TC\ (SC\ \_2424468)))$

thm RSTC:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RSTC\ R = RC\ (TC\ (SC\ R))$

thm RSTC_INC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_REFL:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ x::?'a::type.\ RSTC\ R\ x\ x$

thm RSTC_SYM:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RSTC\ R\ x\ y \longrightarrow RSTC\ R\ y\ x$

thm RSTC_TRANS:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ RSTC\ R\ x\ y \wedge RSTC\ R\ y\ z \longrightarrow RSTC\ R\ x\ z$

thm RSTC_RULES:

$\forall R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow RSTC\ R\ x\ y) \wedge (\forall x::?'a::type.\ RSTC\ R\ x\ x) \wedge (\forall (x::?'a::type)\ y::?'a::type.$

$RSTC\ R\ x\ y \longrightarrow RSTC\ R\ y\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$
$RSTC\ R\ x\ y \wedge RSTC\ R\ y\ z \longrightarrow RSTC\ R\ x\ z)$

thm RSTC_TRANS_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.$
$RSTC\ R\ x\ y \wedge R\ y\ z \longrightarrow RSTC\ R\ x\ z$

thm RSTC_TRANS_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ R$
$x\ y \wedge RSTC\ R\ y\ z \longrightarrow RSTC\ R\ x\ z$

thm RSTC_CASES:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RSTC\ R\ x\ z =$
$(x = z \vee R\ x\ z \vee RSTC\ R\ z\ x \vee (\exists\, y::?'a::type.\ RSTC\ R\ x\ y \wedge RSTC\ R\ y\ z))$

thm RSTC_CASES_L:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RSTC\ R\ x\ z =$
$(x = z \vee R\ x\ z \vee RSTC\ R\ z\ x \vee (\exists\, y::?'a::type.\ RSTC\ R\ x\ y \wedge R\ y\ z))$

thm RSTC_CASES_R:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ z::?'a::type.\ RSTC\ R\ x\ z =$
$(x = z \vee R\ x\ z \vee RSTC\ R\ z\ x \vee (\exists\, y::?'a::type.\ R\ x\ y \wedge RSTC\ R\ y\ z))$

thm RSTC_INDUCT:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)$
$y::?'a::type.\ R\ x\ y \longrightarrow P\ x\ y) \wedge (\forall x::?'a::type.\ P\ x\ x) \wedge (\forall (x::?'a::type)$
$y::?'a::type.\ P\ x\ y \longrightarrow P\ y\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)\ z::?'a::type.\ P$
$x\ y \wedge P\ y\ z \longrightarrow P\ x\ z) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RSTC\ R\ x\ y \longrightarrow P$
$x\ y)$

thm RSTC_MONO:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ S::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (\forall (x::?'a::type)$
$y::?'a::type.\ R\ x\ y \longrightarrow S\ x\ y) \longrightarrow (\forall (x::?'a::type)\ y::?'a::type.\ RSTC\ R\ x\ y$
$\longrightarrow RSTC\ S\ x\ y)$

thm RSTC_CLOSED:

$\forall\, R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ (RSTC\ R = R) = ((\forall\, x::?'a::type.\ R\ x\ x) \wedge$
$(\forall (x::?'a::type)\ y::?'a::type.\ R\ x\ y \longrightarrow R\ y\ x) \wedge (\forall (x::?'a::type)\ (y::?'a::type)$
$z::?'a::type.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z))$

thm RSTC_IDEMP:

$\forall\, R::?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ RSTC\ (RSTC\ R) = RSTC\ R$

thm RSC_INC_RC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y \longrightarrow$
$RSC\ R\ x\ y$

thm RSC_INC_SC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow RSC\ R\ x\ y$

thm RTC_INC_RC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y \longrightarrow RTC\ R\ x\ y$

thm RTC_INC_TC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow RTC\ R\ x\ y$

thm STC_INC_SC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow STC\ R\ x\ y$

thm STC_INC_TC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow STC\ R\ x\ y$

thm RSTC_INC_RC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_INC_SC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ SC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_INC_TC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ TC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_INC_RSC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RSC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_INC_RTC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RTC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm RSTC_INC_STC:

$\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ STC\ R\ x\ y \longrightarrow RSTC\ R\ x\ y$

thm DEF_INV:

$INV = (\lambda(\_2425413::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (\_2425414::?'a::type)\ \_2425415::?'b::type.\ \_2425413\ \_2425415\ \_2425414)$

thm INV:

$\forall (R::?'b::type \Rightarrow ?'a::type \Rightarrow bool)\ (y::?'b::type)\ x::?'a::type.\ INV\ R\ x\ y = R\ y\ x$

thm RC_INV:

$RC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (RC\ ?R)$

thm SC_INV:

$SC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (SC\ ?R)$

thm SC_INV_STRONG:

$SC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = SC\ ?R$

thm TC_INV:

$TC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (TC\ ?R)$

thm RSC_INV:

$RSC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (RSC\ ?R)$

thm RTC_INV:

$RTC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (RTC\ ?R)$

thm STC_INV:

$STC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (STC\ ?R)$

thm RSTC_INV:

$RSTC\ (INV\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)) = INV\ (RSTC\ ?R)$

thm DEF_RELPOW:

$RELPOW = (SOME\ RELPOW::nat \Rightarrow nat \Rightarrow (?'a::type \Rightarrow ?'a::type \Rightarrow bool) \Rightarrow ?'a::type \Rightarrow ?'a::type \Rightarrow bool.\ \forall \_2425489::nat.\ (\forall (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (x::?'a::type)\ y::?'a::type.\ RELPOW\ \_2425489\ (0::nat)\ R\ x\ y = (x = y)) \wedge (\forall (n::nat)\ (x::?'a::type)\ (R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ y::?'a::type.\ RELPOW\ \_2425489\ (Suc\ n)\ R\ x\ y = (\exists z::?'a::type.\ RELPOW\ \_2425489\ n\ R\ x\ z \wedge R\ z\ y)))\ (118::nat)$

thm RELPOW_conjunct0:

$RELPOW\ (0::nat)\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (?x::?'a::type)\ (?y::?'a::type) = (?x = ?y)$

thm RELPOW_conjunct1:

$RELPOW\ (Suc\ (?n::nat))\ (?R::?'a::type \Rightarrow ?'a::type \Rightarrow bool)\ (?x::?'a::type)\ (?y::?'a::type) = (\exists z::?'a::type.\ RELPOW\ ?n\ ?R\ ?x\ z \wedge ?R\ z\ ?y)$

thm RELPOW:

RELPOW (0::nat) (?R::?'a::type ⇒ ?'a::type ⇒ bool) (?x::?'a::type) (?y::?'a::type) = (?x = ?y) ∧ RELPOW (Suc (?n::nat)) ?R ?x ?y = (∃ z::?'a::type. RELPOW ?n ?R ?x z ∧ ?R z ?y)

thm RELPOW_R_conjunct1:

RELPOW (Suc (?n::nat)) (?R::?'a::type ⇒ ?'a::type ⇒ bool) (?x::?'a::type) (?y::?'a::type) = (∃ z::?'a::type. ?R ?x z ∧ RELPOW ?n ?R z ?y)

thm RELPOW_R:

RELPOW (0::nat) (?R::?'a::type ⇒ ?'a::type ⇒ bool) (?x::?'a::type) (?y::?'a::type) = (?x = ?y) ∧ RELPOW (Suc (?n::nat)) ?R ?x ?y = (∃ z::?'a::type. ?R ?x z ∧ RELPOW ?n ?R z ?y)

thm RELPOW_M:

∀ (m::nat) (n::nat) (x::?'a::type) y::?'a::type. RELPOW (m + n) (?R::?'a::type ⇒ ?'a::type ⇒ bool) x y = (∃ z::?'a::type. RELPOW m ?R x z ∧ RELPOW n ?R z y)

thm RTC_RELPOW:

∀ (R::?'a::type ⇒ ?'a::type ⇒ bool) (x::?'a::type) y::?'a::type. RTC R x y = (∃ n::nat. RELPOW n R x y)

thm TC_RELPOW:

∀ (R::?'a::type ⇒ ?'a::type ⇒ bool) (x::?'a::type) y::?'a::type. TC R x y = (∃ n::nat. RELPOW (Suc n) R x y)

thm RELPOW_SEQUENCE:

∀ (R::?'a::type ⇒ ?'a::type ⇒ bool) (n::nat) (x::?'a::type) y::?'a::type. RELPOW n R x y = (∃ f::nat ⇒ ?'a::type. f (0::nat) = x ∧ f n = y ∧ (∀ i<n. R (f i) (f (Suc i))))

thm DEF_bn_the:

bn_the = (λ_2426147::?'a::type HOL_Light_Import.option. SOME x::?'a::type. _2426147 = SOME x)

thm Tame_classification.bn_the:

∀ s::?'a::type HOL_Light_Import.option. bn_the s = (SOME x::?'a::type. s = SOME x)

thm DEF_bn_enum:

bn_enum = (λ_2426152::nat. GSPEC (λGEN%PVAR%278::nat. ∃ m::nat. SETSPEC GEN%PVAR%278 (m < _2426152) m))

thm Tame_classification.bn_enum:

∀ *n*::*nat*. *bn_enum n* = *GSPEC* (λ*GEN%PVAR%278*::*nat*. ∃ *m*::*nat*. *SET-SPEC GEN%PVAR%278* (*m* < *n*) *m*)

thm DEF_bn_filter:

bn_filter = (*SOME bn_filter*::*nat* ⇒ (*?'a*::*type* ⇒ *bool*) ⇒ *?'a*::*type list* ⇒ *?'a*::*type list*. ∀ *_2426163*::*nat*. (∀ *f*::*?'a*::*type* ⇒ *bool*. *bn_filter _2426163 f* [] = []) ∧ (∀ (*x*::*?'a*::*type*) (*f*::*?'a*::*type* ⇒ *bool*) *xs*::*?'a*::*type list*. *bn_filter _2426163 f* (*x* # *xs*) = (*if f x then x* # *bn_filter _2426163 f xs else bn_filter _2426163 f xs*))) (*119*::*nat*)

thm Tame_classification.bn_filter_conjunct0:

bn_filter (*?f*::*?'a*::*type* ⇒ *bool*) [] = []

thm Tame_classification.bn_filter_conjunct1:

bn_filter (*?f*::*?'a*::*type* ⇒ *bool*) ((*?x*::*?'a*::*type*) # (*?xs*::*?'a*::*type list*)) = (*if ?f ?x then ?x* # *bn_filter ?f ?xs else bn_filter ?f ?xs*)

thm Tame_classification.bn_filter:

bn_filter (*?f*::*?'a*::*type* ⇒ *bool*) [] = [] ∧ *bn_filter ?f* ((*?x*::*?'a*::*type*) # (*?xs*::*?'a*::*type list*)) = (*if ?f ?x then ?x* # *bn_filter ?f ?xs else bn_filter ?f ?xs*)

thm Tame_classification.bn_filter_FILTER:

bn_filter = *filter*

thm DEF_bn_concat:

bn_concat = (*SOME bn_concat*::*nat* ⇒ *?'a*::*type list list* ⇒ *?'a*::*type list*. ∀ *_2426167*::*nat*. *bn_concat _2426167* [] = [] ∧ (∀ (*x*::*?'a*::*type list*) *xs*::*?'a*::*type list list*. *bn_concat _2426167* (*x* # *xs*) = *x* @ *bn_concat _2426167 xs*)) (*120*::*nat*)

thm Tame_classification.bn_concat_conjunct0:

bn_concat [] = []

thm Tame_classification.bn_concat_conjunct1:

bn_concat ((*?x*::*?'a*::*type list*) # (*?xs*::*?'a*::*type list list*)) = *?x* @ *bn_concat ?xs*

thm Tame_classification.bn_concat:

bn_concat [] = [] ∧ *bn_concat* ((*?x*::*?'a*::*type list*) # (*?xs*::*?'a*::*type list list*)) = *?x* @ *bn_concat ?xs*

thm DEF_bn_listProd1:

bn_listProd1 = (λ*_2426168*::*?'b*::*type*. *map* (*Pair _2426168*))

thm Tame_classification.bn_listProd1:

∀ (*a*::*?'b*::*type*) *b*::*?'a*::*type list*. *bn_listProd1 a b* = *map* (*Pair a*) *b*

thm DEF_bn_listProd:

$bn\_listProd = (\lambda(\_2426180::?'b::type\ list)\ \_2426181::?'a::type\ list.\ bn\_concat\ (map\ (\lambda x::?'b::type.\ bn\_listProd1\ x\ \_2426181)\ \_2426180))$

thm Tame_classification.bn_listProd:

$\forall (b::?'b::type\ list)\ a::?'a::type\ list.\ bn\_listProd\ a\ b = bn\_concat\ (map\ (\lambda x::?'a::type.\ bn\_listProd1\ x\ b)\ a)$

thm DEF_bn_minimal:

$bn\_minimal = (SOME\ bn\_minimal::nat \Rightarrow (?'a::type \Rightarrow nat) \Rightarrow ?'a::type\ list \Rightarrow ?'a::type.\ \forall \_2426198::nat.\ (\forall f::?'a::type \Rightarrow nat.\ bn\_minimal\ \_2426198\ f\ [] = CHOICE\ HOL\_Light\_Import.UNIV) \wedge (\forall (x::?'a::type)\ (f::?'a::type \Rightarrow nat)\ xs::?'a::type\ list.\ bn\_minimal\ \_2426198\ f\ (x\ \#\ xs) = (if\ xs = []\ then\ x\ else\ LET\ (\lambda m::?'a::type.\ LET\_END\ (if\ f\ x \leq f\ m\ then\ x\ else\ m))\ (bn\_minimal\ \_2426198\ f\ xs))))\ (121::nat)$

thm Tame_classification.bn_minimal_conjunct0:

$bn\_minimal\ (?f::?'a::type \Rightarrow nat)\ [] = CHOICE\ HOL\_Light\_Import.UNIV$

thm Tame_classification.bn_minimal_conjunct1:

$bn\_minimal\ (?f::?'a::type \Rightarrow nat)\ ((?x::?'a::type)\ \#\ (?xs::?'a::type\ list)) = (if\ ?xs = []\ then\ ?x\ else\ LET\ (\lambda m::?'a::type.\ LET\_END\ (if\ ?f\ ?x \leq ?f\ m\ then\ ?x\ else\ m))\ (bn\_minimal\ ?f\ ?xs))$

thm Tame_classification.bn_minimal:

$bn\_minimal\ (?f::?'a::type \Rightarrow nat)\ [] = CHOICE\ HOL\_Light\_Import.UNIV \wedge bn\_minimal\ ?f\ ((?x::?'a::type)\ \#\ (?xs::?'a::type\ list)) = (if\ ?xs = []\ then\ ?x\ else\ LET\ (\lambda m::?'a::type.\ LET\_END\ (if\ ?f\ ?x \leq ?f\ m\ then\ ?x\ else\ m))\ (bn\_minimal\ ?f\ ?xs))$

thm DEF_bn_min_list:

$bn\_min\_list = (\lambda \_2426199::nat\ list.\ min\_num\ (set\_of\_list\ \_2426199))$

thm Tame_classification.bn_min_list:

$\forall xs::nat\ list.\ bn\_min\_list\ xs = min\_num\ (set\_of\_list\ xs)$

thm DEF_max_num:

$max\_num = (\lambda \_2426204::nat \Rightarrow bool.\ SOME\ m::nat.\ \_2426204\ m \wedge (\forall n::nat.\ \_2426204\ n \longrightarrow n \leq m))$

thm Tame_classification.max_num:

$\forall x::nat \Rightarrow bool.\ max\_num\ x = (SOME\ m::nat.\ x\ m \wedge (\forall n::nat.\ x\ n \longrightarrow n \leq m))$

thm DEF_bn_max_list:

$bn\_max\_list = (\lambda \_2426209::nat\ list.\ max\_num\ (set\_of\_list\ \_2426209))$

thm Tame_classification.bn_max_list:

∀ *xs*::*nat list*. *bn_max_list xs* = *max_num* (*set_of_list xs*)

thm DEF_bn_replace:

bn_replace = (*SOME bn_replace*::*nat* ⇒ *?'a*::*type* ⇒ *?'a*::*type list* ⇒ *?'a*::*type list* ⇒ *?'a*::*type list*. ∀ *_2426221*::*nat*. (∀ (*x*::*?'a*::*type*) *ys*::*?'a*::*type list*. *bn_replace _2426221 x ys* [] = []) ∧ (∀ (*z*::*?'a*::*type*) (*x*::*?'a*::*type*) (*ys*::*?'a*::*type list*) *zs*::*?'a*::*type list*. *bn_replace _2426221 x ys* (*z* # *zs*) = (*if z* = *x then ys* @ *zs else z* # *bn_replace _2426221 x ys zs*))) (*122*::*nat*)

thm Tame_classification.bn_replace_conjunct0:

bn_replace (*?x*::*?'a*::*type*) (*?ys*::*?'a*::*type list*) [] = []

thm Tame_classification.bn_replace_conjunct1:

bn_replace (*?x*::*?'a*::*type*) (*?ys*::*?'a*::*type list*) ((*?z*::*?'a*::*type*) # (*?zs*::*?'a*::*type list*)) = (*if ?z* = *?x then ?ys* @ *?zs else ?z* # *bn_replace ?x ?ys ?zs*)

thm Tame_classification.bn_replace:

bn_replace (*?x*::*?'a*::*type*) (*?ys*::*?'a*::*type list*) [] = [] ∧ *bn_replace ?x ?ys* ((*?z*::*?'a*::*type*) # (*?zs*::*?'a*::*type list*)) = (*if ?z* = *?x then ?ys* @ *?zs else ?z* # *bn_replace ?x ?ys ?zs*)

thm DEF_mapAt1:

mapAt1 = (*SOME mapAt1*::*nat* ⇒ (*?'a*::*type* ⇒ *?'a*::*type*) ⇒ *nat* ⇒ *?'a*::*type list* ⇒ *?'a*::*type list* ⇒ *?'a*::*type list*. ∀ *_2426230*::*nat*. (∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) (*n*::*nat*) *xs*::*?'a*::*type list*. *mapAt1 _2426230 f n xs* [] = *rev xs*) ∧ (∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) (*n*::*nat*) (*y*::*?'a*::*type*) (*xs*::*?'a*::*type list*) *ys*::*?'a*::*type list*. *mapAt1 _2426230 f n xs* (*y* # *ys*) = (*if n* = (*0*::*nat*) *then rev xs* @ *f y* # *ys else mapAt1 _2426230 f* (*n* − (*1*::*nat*)) (*y* # *xs*) *ys*))) (*123*::*nat*)

thm Tame_classification.mapAt1_conjunct0:

mapAt1 (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?n*::*nat*) (*?xs*::*?'a*::*type list*) [] = *rev ?xs*

thm Tame_classification.mapAt1_conjunct1:

mapAt1 (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?n*::*nat*) (*?xs*::*?'a*::*type list*) ((*?y*::*?'a*::*type*) # (*?ys*::*?'a*::*type list*)) = (*if ?n* = (*0*::*nat*) *then rev ?xs* @ *?f ?y* # *?ys else mapAt1 ?f* (*?n* − (*1*::*nat*)) (*?y* # *?xs*) *?ys*)

thm Tame_classification.mapAt1:

mapAt1 (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?n*::*nat*) (*?xs*::*?'a*::*type list*) [] = *rev ?xs* ∧ *mapAt1 ?f ?n ?xs* ((*?y*::*?'a*::*type*) # (*?ys*::*?'a*::*type list*)) = (*if ?n* = (*0*::*nat*) *then rev ?xs* @ *?f ?y* # *?ys else mapAt1 ?f* (*?n* − (*1*::*nat*)) (*?y* # *?xs*) *?ys*)

thm DEF_bn_mapAt:

bn_mapAt = (*SOME bn_mapAt*::*nat* ⇒ *nat list* ⇒ (*?'a*::*type* ⇒ *?'a*::*type*) ⇒ *?'a*::*type list* ⇒ *?'a*::*type list*. ∀ *_2426234*::*nat*. (∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) *xs*::*?'a*::*type list*. *bn_mapAt _2426234* [] *f xs* = *xs*) ∧ (∀ (*n*::*nat*) (*ns*::*nat list*)

$(f::?'a::type \Rightarrow ?'a::type)$ $xs::?'a::type$ *list*. *bn_mapAt _2426234* $(n \# ns)$ *f* *xs* = (*if* $n <$ *length xs then bn_mapAt _2426234 ns f* (*mapAt1 f n* [] *xs*) *else* *bn_mapAt _2426234 ns f xs*))) (*124*::*nat*)

thm Tame_classification.bn_mapAt_conjunct0:

bn_mapAt [] $(?f::?'a::type \Rightarrow ?'a::type)$ $(?xs::?'a::type\ list) = ?xs$

thm Tame_classification.bn_mapAt_conjunct1:

bn_mapAt $((?n::nat) \# (?ns::nat\ list))$ $(?f::?'a::type \Rightarrow ?'a::type)$ $(?xs::?'a::type$ *list*) = (*if* $?n <$ *length ?xs then bn_mapAt ?ns ?f* (*mapAt1 ?f ?n* [] *?xs*) *else* *bn_mapAt ?ns ?f ?xs*)

thm Tame_classification.bn_mapAt:

bn_mapAt [] $(?f::?'a::type \Rightarrow ?'a::type)$ $(?xs::?'a::type\ list) = ?xs \land$ *bn_mapAt* $((?n::nat) \# (?ns::nat\ list))$ *?f ?xs* = (*if* $?n <$ *length ?xs then bn_mapAt ?ns* *?f* (*mapAt1 ?f ?n* [] *?xs*) *else bn_mapAt ?ns ?f ?xs*)

thm DEF_bn_rotate1:

bn_rotate1 = (*SOME bn_rotate1*::$nat \Rightarrow ?'a::type\ list \Rightarrow ?'a::type\ list$. $\forall$ *_2426238*::*nat*. *bn_rotate1 _2426238* [] = [] $\land$ ($\forall$ $(xs::?'a::type\ list)$ $x::?'a::type$. *bn_rotate1* *_2426238* $(x \# xs) = xs$ @ [*x*])) (*125*::*nat*)

thm Tame_classification.bn_rotate1_conjunct0:

bn_rotate1 [] = []

thm Tame_classification.bn_rotate1_conjunct1:

bn_rotate1 $((?x::?'a::type) \# (?xs::?'a::type\ list)) = ?xs$ @ [*?x*]

thm Tame_classification.bn_rotate1:

bn_rotate1 [] = [] $\land$ *bn_rotate1* $((?x::?'a::type) \# (?xs::?'a::type\ list)) = ?xs$ @ [*?x*]

thm DEF_bn_rotate:

bn_rotate = *POWER bn_rotate1*

thm Tame_classification.bn_rotate:

$\forall$ (*n*::*nat*) $xs::?'a::type$ *list*. *bn_rotate n xs* = *POWER bn_rotate1 n xs*

thm DEF_bn_splitAtRec:

bn_splitAtRec = (*SOME bn_splitAtRec*::$nat \Rightarrow ?'a::type \Rightarrow ?'a::type\ list \Rightarrow ?'a::type\ list \Rightarrow ?'a::type\ list \times ?'a::type\ list$. $\forall$ *_2426258*::*nat*. ($\forall$ $(c::?'a::type)$ $bs::?'a::type$ *list*. *bn_splitAtRec _2426258 c bs* [] = (*bs*, [])) $\land$ ($\forall$ $(c::?'a::type)$ $(bs::?'a::type\ list)$ $(a::?'a::type)$ $xs::?'a::type$ *list*. *bn_splitAtRec _2426258 c bs* $(a \# xs)$ = (*if a* = *c then* (*bs*, *xs*) *else bn_splitAtRec _2426258 c* (*bs* @ [*a*]) *xs*))) (*126*::*nat*)

thm Tame_classification.bn_splitAtRec_conjunct0:

$bn\_splitAtRec\ (?c::?'a::type)\ (?bs::?'a::type\ list)\ []\ =\ (?bs,\ [])$

thm Tame_classification.bn_splitAtRec_conjunct1:

$bn\_splitAtRec\ (?c::?'a::type)\ (?bs::?'a::type\ list)\ ((?a::?'a::type)\ \#\ (?xs::?'a::type\ list))\ =\ (if\ ?a\ =\ ?c\ then\ (?bs,\ ?xs)\ else\ bn\_splitAtRec\ ?c\ (?bs\ @\ [?a])\ ?xs)$

thm Tame_classification.bn_splitAtRec:

$bn\_splitAtRec\ (?c::?'a::type)\ (?bs::?'a::type\ list)\ []\ =\ (?bs,\ [])\ \wedge\ bn\_splitAtRec\ ?c\ ?bs\ ((?a::?'a::type)\ \#\ (?xs::?'a::type\ list))\ =\ (if\ ?a\ =\ ?c\ then\ (?bs,\ ?xs)\ else\ bn\_splitAtRec\ ?c\ (?bs\ @\ [?a])\ ?xs)$

thm DEF_bn_splitAt:

$bn\_splitAt\ =\ (\lambda\_2426259::?'a::type.\ bn\_splitAtRec\ \_2426259\ [])$

thm Tame_classification.bn_splitAt:

$\forall (c::?'a::type)\ xs::?'a::type\ list.\ bn\_splitAt\ c\ xs\ =\ bn\_splitAtRec\ c\ []\ xs$

thm DEF_bn_between:

$bn\_between = (\lambda(\_2426271::?'a::type\ list)\ (\_2426272::?'a::type)\ \_2426273::?'a::type.\ LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre1::?'a::type\ list)\ post1::?'a::type\ list.\ GEQ\ (f\ (pre1,\ post1))\ (LET\_END\ (if\ set\_of\_list\ post1\ \_2426273\ then\ LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre2::?'a::type\ list)\ post2::?'a::type\ list.\ GEQ\ (f\ (pre2,\ post2))\ (LET\_END\ pre2)))\ (bn\_splitAt\ \_2426273\ post1)\ else\ LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre2::?'a::type\ list)\ post2::?'a::type\ list.\ GEQ\ (f\ (pre2,\ post2))\ (LET\_END\ (post1\ @\ pre2))))\ (bn\_splitAt\ \_2426273\ pre1)))))\ (bn\_splitAt\ \_2426272\ \_2426271))$

thm Tame_classification.bn_between:

$\forall (ram2::?'a::type)\ (ram1::?'a::type)\ vs::?'a::type\ list.\ bn\_between\ vs\ ram1\ ram2\ = LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre1::?'a::type\ list)\ post1::?'a::type\ list.\ GEQ\ (f\ (pre1,\ post1))\ (LET\_END\ (if\ set\_of\_list\ post1\ ram2\ then\ LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre2::?'a::type\ list)\ post2::?'a::type\ list.\ GEQ\ (f\ (pre2,\ post2))\ (LET\_END\ pre2)))\ (bn\_splitAt\ ram2\ post1)\ else\ LET\ (GABS\ (\lambda f::?'a::type\ list \times ?'a::type\ list \Rightarrow ?'a::type\ list.\ \forall (pre2::?'a::type\ list)\ post2::?'a::type\ list.\ GEQ\ (f\ (pre2,\ post2))\ (LET\_END\ (post1\ @\ pre2))))\ (bn\_splitAt\ ram2\ pre1)))))\ (bn\_splitAt\ ram1\ vs)$

thm DEF_bn_isTable:

$bn\_isTable\ =\ (\lambda(\_2426292::?'b::type\ \Rightarrow\ ?'a::type)\ (\_2426293::?'b::type\ list)\ \_2426294::(?'b::type\ \times\ ?'a::type)\ list.\ \forall p::?'b::type\ \times\ ?'a::type.\ set\_of\_list\ \_2426294\ p\ \longrightarrow\ snd\ p\ =\ \_2426292\ (fst\ p)\ \wedge\ set\_of\_list\ \_2426293\ (fst\ p))$

thm Tame_classification.bn_isTable:

$\forall$ (*t*::(*?'b*::*type* × *?'a*::*type*) *list*) (*f*::*?'b*::*type* ⇒ *?'a*::*type*) *vs*::*?'b*::*type list*. *bn_isTable f vs t* = ($\forall$ *p*::*?'b*::*type* × *?'a*::*type*. *set_of_list t p* ⟶ *snd p* = *f* (*fst p*) ∧ *set_of_list vs* (*fst p*))

thm DEF_bn_removeKey:

bn_removeKey = (λ*_2426313*::*?'b*::*type*. *filter* (λ*p*::*?'b*::*type* × *?'a*::*type*. *_2426313* ≠ *fst p*))

thm Tame_classification.bn_removeKey:

$\forall$ (*a*::*?'b*::*type*) *ps*::(*?'b*::*type* × *?'a*::*type*) *list*. *bn_removeKey a ps* = [*p*::*?'b*::*type* × *?'a*::*type*←*ps* . *a* ≠ *fst p*]

thm DEF_bn_removeKeyList:

bn_removeKeyList = (*SOME bn_removeKeyList*::*nat* ⇒ *?'b*::*type list* ⇒ (*?'b*::*type* × *?'a*::*type*) *list* ⇒ (*?'b*::*type* × *?'a*::*type*) *list*. $\forall$ *_2426328*::*nat*. ($\forall$ *ps*::(*?'b*::*type* × *?'a*::*type*) *list*. *bn_removeKeyList _2426328* [] *ps* = *ps*) ∧ ($\forall$ (*w*::*?'b*::*type*) (*ws*::*?'b*::*type list*) *ps*::(*?'b*::*type* × *?'a*::*type*) *list*. *bn_removeKeyList _2426328* (*w* # *ws*) *ps* = *bn_removeKey w* (*bn_removeKeyList _2426328 ws ps*))) (*127*::*nat*)

thm Tame_classification.bn_removeKeyList_conjunct0:

bn_removeKeyList [] (*?ps*::(*?'b*::*type* × *?'a*::*type*) *list*) = *?ps*

thm Tame_classification.bn_removeKeyList_conjunct1:

bn_removeKeyList ((*?w*::*?'b*::*type*) # (*?ws*::*?'b*::*type list*)) (*?ps*::(*?'b*::*type* × *?'a*::*type*) *list*) = *bn_removeKey ?w* (*bn_removeKeyList ?ws ?ps*)

thm Tame_classification.bn_removeKeyList:

bn_removeKeyList [] (*?ps*::(*?'b*::*type* × *?'a*::*type*) *list*) = *?ps* ∧ *bn_removeKeyList* ((*?w*::*?'b*::*type*) # (*?ws*::*?'b*::*type list*)) *?ps* = *bn_removeKey ?w* (*bn_removeKeyList ?ws ?ps*)

thm DEF_bn_congs:

bn_congs = (λ(*_2426329*::*?'a*::*type list*) *_2426330*::*?'a*::*type list*. ∃ *n*::*nat*. *_2426330* = *bn_rotate n _2426329*)

thm Tame_classification.bn_congs:

$\forall$ (*f2*::*?'a*::*type list*) *f1*::*?'a*::*type list*. *bn_congs f1 f2* = (∃ *n*::*nat*. *f2* = *bn_rotate n f1*)

thm DEF_bn_is_Hom:

bn_is_Hom = (λ(*_2426341*::*?'b*::*type* ⇒ *?'a*::*type*) (*_2426342*::*?'b*::*type list* ⇒ *bool*) *_2426343*::*?'a*::*type list* ⇒ *bool*. *IMAGE bn_congs* (*IMAGE* (*map _2426341*) *_2426342*) = *IMAGE bn_congs _2426343*)

thm Tame_classification.bn_is_Hom:

$\forall (phi::?'b::type \Rightarrow ?'a::type)\ (Fs1::?'b::type\ list \Rightarrow bool)\ Fs2::?'a::type\ list \Rightarrow bool.\ bn\_is\_Hom\ phi\ Fs1\ Fs2 = (IMAGE\ bn\_congs\ (IMAGE\ (map\ phi)\ Fs1) = IMAGE\ bn\_congs\ Fs2)$

thm DEF_bn_inj_on:

$bn\_inj\_on = (\lambda(\_2426362::?'b::type \Rightarrow ?'a::type)\ \_2426363::?'b::type \Rightarrow bool.\ \forall (x::?'b::type)\ y::?'b::type.\ \_2426363\ x \wedge \_2426363\ y \wedge \_2426362\ x = \_2426362\ y \longrightarrow x = y)$

thm Tame_classification.bn_inj_on:

$\forall (s::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ bn\_inj\_on\ f\ s = (\forall (x::?'b::type)\ y::?'b::type.\ s\ x \wedge s\ y \wedge f\ x = f\ y \longrightarrow x = y)$

thm DEF_bn_is_pr_Iso:

$bn\_is\_pr\_Iso = (\lambda(\_2426374::?'b::type \Rightarrow ?'a::type)\ (\_2426375::?'b::type\ list \Rightarrow bool)\ \_2426376::?'a::type\ list \Rightarrow bool.\ bn\_is\_Hom\ \_2426374\ \_2426375\ \_2426376 \wedge bn\_inj\_on\ \_2426374\ (UNIONS\ (IMAGE\ set\_of\_list\ \_2426375)))$

thm Tame_classification.bn_is_pr_Iso:

$\forall (Fs2::?'b::type\ list \Rightarrow bool)\ (phi::?'a::type \Rightarrow ?'b::type)\ Fs1::?'a::type\ list \Rightarrow bool.\ bn\_is\_pr\_Iso\ phi\ Fs1\ Fs2 = (bn\_is\_Hom\ phi\ Fs1\ Fs2 \wedge bn\_inj\_on\ phi\ (UNIONS\ (IMAGE\ set\_of\_list\ Fs1)))$

thm DEF_bn_is_hom:

$bn\_is\_hom = (\lambda(\_2426395::?'b::type \Rightarrow ?'a::type)\ (\_2426396::?'b::type\ list\ list)\ \_2426397::?'a::type\ list\ list.\ bn\_is\_Hom\ \_2426395\ (set\_of\_list\ \_2426396)\ (set\_of\_list\ \_2426397))$

thm Tame_classification.bn_is_hom:

$\forall (phi::?'b::type \Rightarrow ?'a::type)\ (fs1::?'b::type\ list\ list)\ fs2::?'a::type\ list\ list.\ bn\_is\_hom\ phi\ fs1\ fs2 = bn\_is\_Hom\ phi\ (set\_of\_list\ fs1)\ (set\_of\_list\ fs2)$

thm DEF_bn_is_pr_iso:

$bn\_is\_pr\_iso = (\lambda(\_2426416::?'b::type \Rightarrow ?'a::type)\ (\_2426417::?'b::type\ list\ list)\ \_2426418::?'a::type\ list\ list.\ bn\_is\_pr\_Iso\ \_2426416\ (set\_of\_list\ \_2426417)\ (set\_of\_list\ \_2426418))$

thm Tame_classification.bn_is_pr_iso:

$\forall (phi::?'b::type \Rightarrow ?'a::type)\ (fs1::?'b::type\ list\ list)\ fs2::?'a::type\ list\ list.\ bn\_is\_pr\_iso\ phi\ fs1\ fs2 = bn\_is\_pr\_Iso\ phi\ (set\_of\_list\ fs1)\ (set\_of\_list\ fs2)$

thm DEF_bn_is_Iso:

$bn\_is\_Iso = (\lambda(\_2426437::?'b::type \Rightarrow ?'a::type)\ (\_2426438::?'b::type\ list \Rightarrow bool)\ \_2426439::?'a::type\ list \Rightarrow bool.\ bn\_is\_pr\_Iso\ \_2426437\ \_2426438\ \_2426439 \vee bn\_is\_pr\_Iso\ \_2426437\ \_2426438\ (IMAGE\ rev\ \_2426439))$

thm Tame_classification.bn_is_Iso:

$\forall\,(phi::?'b::type \Rightarrow ?'a::type)\ (Fs1::?'b::type\ list \Rightarrow bool)\ Fs2::?'a::type\ list \Rightarrow bool.\ bn\_is\_Iso\ phi\ Fs1\ Fs2 = (bn\_is\_pr\_Iso\ phi\ Fs1\ Fs2 \lor bn\_is\_pr\_Iso\ phi\ Fs1\ (IMAGE\ rev\ Fs2))$

thm DEF_bn_is_iso:

$bn\_is\_iso = (\lambda(\_2426458::?'b::type \Rightarrow ?'a::type)\ (\_2426459::?'b::type\ list\ list)\ \_2426460::?'a::type\ list\ list.\ bn\_is\_Iso\ \_2426458\ (set\_of\_list\ \_2426459)\ (set\_of\_list\ \_2426460))$

thm Tame_classification.bn_is_iso:

$\forall\,(phi::?'b::type \Rightarrow ?'a::type)\ (fs1::?'b::type\ list\ list)\ fs2::?'a::type\ list\ list.\ bn\_is\_iso\ phi\ fs1\ fs2 = bn\_is\_Iso\ phi\ (set\_of\_list\ fs1)\ (set\_of\_list\ fs2)$

thm DEF_bn_cong_iso:

$bn\_cong\_iso = (\lambda(\_2426479::?'b::type\ list\ list)\ \_2426480::?'a::type\ list\ list.\ \exists\,phi::?'b::type \Rightarrow ?'a::type.\ bn\_is\_iso\ phi\ \_2426479\ \_2426480)$

thm Tame_classification.bn_cong_iso:

$\forall\,(fs1::?'b::type\ list\ list)\ fs2::?'a::type\ list\ list.\ bn\_cong\_iso\ fs1\ fs2 = (\exists\,phi::?'b::type \Rightarrow ?'a::type.\ bn\_is\_iso\ phi\ fs1\ fs2)$

thm DEF_bn_cong_pr_iso:

$bn\_cong\_pr\_iso = (\lambda(\_2426491::?'b::type\ list\ list)\ \_2426492::?'a::type\ list\ list.\ \exists\,phi::?'b::type \Rightarrow ?'a::type.\ bn\_is\_pr\_iso\ phi\ \_2426491\ \_2426492)$

thm Tame_classification.bn_cong_pr_iso:

$\forall\,(fs1::?'b::type\ list\ list)\ fs2::?'a::type\ list\ list.\ bn\_cong\_pr\_iso\ fs1\ fs2 = (\exists\,phi::?'b::type \Rightarrow ?'a::type.\ bn\_is\_pr\_iso\ phi\ fs1\ fs2)$

thm DEF_bn_pr_iso_in:

$bn\_pr\_iso\_in = (\lambda(\_2426503::?'b::type\ list\ list)\ \_2426504::?'a::type\ list\ list \Rightarrow bool.\ \exists\,y::?'a::type\ list\ list.\ bn\_cong\_pr\_iso\ \_2426503\ y \land \_2426504\ y)$

thm Tame_classification.bn_pr_iso_in:

$\forall\,(x::?'b::type\ list\ list)\ M::?'a::type\ list\ list \Rightarrow bool.\ bn\_pr\_iso\_in\ x\ M = (\exists\,y::?'a::type\ list\ list.\ bn\_cong\_pr\_iso\ x\ y \land M\ y)$

thm DEF_bn_pr_iso_subseteq:

$bn\_pr\_iso\_subseteq = (\lambda(\_2426515::?'b::type\ list\ list \Rightarrow bool)\ \_2426516::?'a::type\ list\ list \Rightarrow bool.\ \forall\,x::?'b::type\ list\ list.\ \_2426515\ x \longrightarrow bn\_pr\_iso\_in\ x\ \_2426516)$

thm Tame_classification.bn_pr_iso_subseteq:

$\forall\,(M::?'b::type\ list\ list \Rightarrow bool)\ N::?'a::type\ list\ list \Rightarrow bool.\ bn\_pr\_iso\_subseteq\ M\ N = (\forall\,x::?'b::type\ list\ list.\ M\ x \longrightarrow bn\_pr\_iso\_in\ x\ N)$

thm DEF_bn_iso_in:

$bn\_iso\_in = (\lambda(\_2426527::?'b::type\ list\ list)\ \_2426528::?'a::type\ list\ list \Rightarrow bool.\ \exists\, y::?'a::type\ list\ list.\ bn\_cong\_iso\ \_2426527\ y \wedge \_2426528\ y)$

thm Tame_classification.bn_iso_in:

$\forall\,(x::?'b::type\ list\ list)\ M::?'a::type\ list\ list \Rightarrow bool.\ bn\_iso\_in\ x\ M = (\exists\, y::?'a::type\ list\ list.\ bn\_cong\_iso\ x\ y \wedge M\ y)$

thm DEF_bn_iso_subseteq:

$bn\_iso\_subseteq = (\lambda(\_2426539::?'b::type\ list\ list \Rightarrow bool)\ \_2426540::?'a::type\ list\ list \Rightarrow bool.\ \forall\, x::?'b::type\ list\ list.\ \_2426539\ x \longrightarrow bn\_iso\_in\ x\ \_2426540)$

thm Tame_classification.bn_iso_subseteq:

$\forall\,(M::?'b::type\ list\ list \Rightarrow bool)\ N::?'a::type\ list\ list \Rightarrow bool.\ bn\_iso\_subseteq\ M\ N = (\forall\, x::?'b::type\ list\ list.\ M\ x \longrightarrow bn\_iso\_in\ x\ N)$

thm DEF_bn_rotate_to:

$bn\_rotate\_to = (\lambda(\_2426551::?'a::type\ list)\ \_2426552::?'a::type.\ \_2426552\ \#\ snd\ (bn\_splitAt\ \_2426552\ \_2426551)\ @\ fst\ (bn\_splitAt\ \_2426552\ \_2426551))$

thm Tame_classification.bn_rotate_to:

$\forall\,(v::?'a::type)\ vs::?'a::type\ list.\ bn\_rotate\_to\ vs\ v = v\ \#\ snd\ (bn\_splitAt\ v\ vs)\ @\ fst\ (bn\_splitAt\ v\ vs)$

thm DEF_bn_rotate_min:

$bn\_rotate\_min = (\lambda\_2426563::nat\ list.\ bn\_rotate\_to\ \_2426563\ (bn\_min\_list\ \_2426563))$

thm Tame_classification.bn_rotate_min:

$\forall\, vs::nat\ list.\ bn\_rotate\_min\ vs = bn\_rotate\_to\ vs\ (bn\_min\_list\ vs)$

thm DEF_bn_final_face:

$bn\_final\_face = snd$

thm Tame_classification.bn_final_face:

$\forall\,(vs::?'a::type)\ f::bool.\ bn\_final\_face\ (vs, f) = f$

thm DEF_bn_vertices_face:

$bn\_vertices\_face = fst$

thm Tame_classification.bn_vertices_face:

$\forall\,(f::?'b::type)\ vs::?'a::type.\ bn\_vertices\_face\ (vs, f) = vs$

thm DEF_bn_vertices_set:

$bn\_vertices\_set = (\lambda\_2426586::?'b::type\ list \times ?'a::type.\ set\_of\_list\ (bn\_vertices\_face\ \_2426586))$

thm Tame_classification.bn_vertices_set:

$\forall fs{::}?'b{::}type\ list \times ?'a{::}type.\ bn\_vertices\_set\ fs = set\_of\_list\ (bn\_vertices\_face\ fs)$

thm DEF_bn_setFinal:

$bn\_setFinal = (\lambda\_2426591{::}?'a{::}type \times bool.\ (fst\ \_2426591,\ True))$

thm Tame_classification.bn_setFinal:

$\forall (f{::}bool)\ vs{::}?'a{::}type.\ bn\_setFinal\ (vs,\ f) = (vs,\ True)$

thm DEF_bn_nextElem:

$bn\_nextElem = (SOME\ bn\_nextElem{::}nat \Rightarrow ?'a{::}type\ list \Rightarrow ?'a{::}type \Rightarrow ?'a{::}type \Rightarrow ?'a{::}type.\ \forall\ \_2426603{::}nat.\ (\forall (x{::}?'a{::}type)\ b{::}?'a{::}type.\ bn\_nextElem\ \_2426603\ []\ b\ x = b) \wedge (\forall (a{::}?'a{::}type)\ (aas{::}?'a{::}type\ list)\ (b{::}?'a{::}type)\ x{::}?'a{::}type.\ bn\_nextElem\ \_2426603\ (a\ \#\ aas)\ b\ x = (if\ x = a\ then\ if\ length\ aas = (0{::}nat)\ then\ b\ else\ hd\ aas\ else\ bn\_nextElem\ \_2426603\ aas\ b\ x)))\ (128{::}nat)$

thm Tame_classification.bn_nextElem_conjunct0:

$bn\_nextElem\ []\ (?b{::}?'a{::}type)\ (?x{::}?'a{::}type) = ?b$

thm Tame_classification.bn_nextElem_conjunct1:

$bn\_nextElem\ ((?a{::}?'a{::}type)\ \#\ (?aas{::}?'a{::}type\ list))\ (?b{::}?'a{::}type)\ (?x{::}?'a{::}type) = (if\ ?x = ?a\ then\ if\ length\ ?aas = (0{::}nat)\ then\ ?b\ else\ hd\ ?aas\ else\ bn\_nextElem\ ?aas\ ?b\ ?x)$

thm Tame_classification.bn_nextElem:

$bn\_nextElem\ []\ (?b{::}?'a{::}type)\ (?x{::}?'a{::}type) = ?b \wedge bn\_nextElem\ ((?a{::}?'a{::}type)\ \#\ (?aas{::}?'a{::}type\ list))\ ?b\ ?x = (if\ ?x = ?a\ then\ if\ length\ ?aas = (0{::}nat)\ then\ ?b\ else\ hd\ ?aas\ else\ bn\_nextElem\ ?aas\ ?b\ ?x)$

thm DEF_bn_nextVertex:

$bn\_nextVertex = (\lambda\_2426604{::}?'a{::}type\ list \times bool.\ bn\_nextElem\ (fst\ \_2426604)\ (hd\ (fst\ \_2426604)))$

thm Tame_classification.bn_nextVertex:

$\forall (f{::}bool)\ vs{::}?'a{::}type\ list.\ bn\_nextVertex\ (vs,\ f) = bn\_nextElem\ vs\ (hd\ vs)$

thm DEF_bn_edges:

$bn\_edges = (\lambda\_2426613{::}?'a{::}type\ list \times bool.\ IMAGE\ (\lambda a{::}?'a{::}type.\ (a,\ bn\_nextVertex\ \_2426613\ a))\ (bn\_vertices\_set\ \_2426613))$

thm Tame_classification.bn_edges:

$\forall fs{::}?'a{::}type\ list \times bool.\ bn\_edges\ fs = IMAGE\ (\lambda a{::}?'a{::}type.\ (a,\ bn\_nextVertex\ fs\ a))\ (bn\_vertices\_set\ fs)$

thm DEF_bn_nextVertices:

$bn\_nextVertices = (\lambda\_2426618{::}?'a{::}type\ list \times bool.\ POWER\ (bn\_nextVertex\ (fst\ \_2426618,\ snd\ \_2426618)))$

thm Tame_classification.bn_nextVertices:

$\forall$ (*vs*::*?'a*::*type list*) (*f*::*bool*) (*n*::*nat*) *v*::*?'a*::*type*. *bn_nextVertices* (*vs*, *f*) *n* *v* = *POWER* (*bn_nextVertex* (*vs*, *f*)) *n* *v*

thm DEF_bn_prevVertex:

bn_prevVertex = (λ*_2426645*::*?'a*::*type list* $\times$ *bool*. *bn_nextElem* (*rev* (*fst* *_2426645*)) (*last* (*fst* *_2426645*)))

thm Tame_classification.bn_prevVertex:

$\forall$ (*f*::*bool*) (*vs*::*?'a*::*type list*) *v*::*?'a*::*type*. *bn_prevVertex* (*vs*, *f*) *v* = *bn_nextElem* (*rev vs*) (*last vs*) *v*

thm DEF_bn_triangle:

bn_triangle = (λ*_2426662*::*?'a*::*type list* $\times$ *bool*. *length* (*fst* *_2426662*) = (*3*::*nat*))

thm Tame_classification.bn_triangle:

$\forall$ (*f*::*bool*) *vs*::*?'a*::*type list*. *bn_triangle* (*vs*, *f*) = (*length vs* = (*3*::*nat*))

thm Tame_classification.new_graph_th:

$\exists$ *x*::(*nat list* $\times$ *bool*) *list* $\times$ *nat* $\times$ (*nat list* $\times$ *bool*) *list list* $\times$ *nat list*. *True*

thm TYDEF_bn_graph:

mk_bn_graph (*dest_bn_graph* (*?a*::*bn_graph*)) = *?a* $\wedge$ *True* = (*dest_bn_graph* (*mk_bn_graph* (*?r*::(*nat list* $\times$ *bool*) *list* $\times$ *nat* $\times$ (*nat list* $\times$ *bool*) *list list* $\times$ *nat list*)) = *?r*)

thm Tame_classification.bn_graph_type_conjunct1:

$\forall$ *r*::(*nat list* $\times$ *bool*) *list* $\times$ *nat* $\times$ (*nat list* $\times$ *bool*) *list list* $\times$ *nat list*. *True* = (*dest_bn_graph* (*mk_bn_graph r*) = *r*)

thm Tame_classification.bn_graph_type_conjunct0:

$\forall$ *a*::*bn_graph*. *mk_bn_graph* (*dest_bn_graph a*) = *a*

thm Tame_classification.bn_graph_type:

($\forall$ *a*::*bn_graph*. *mk_bn_graph* (*dest_bn_graph a*) = *a*) $\wedge$ ($\forall$ *r*::(*nat list* $\times$ *bool*) *list* $\times$ *nat* $\times$ (*nat list* $\times$ *bool*) *list list* $\times$ *nat list*. *True* = (*dest_bn_graph* (*mk_bn_graph r*) = *r*))

thm DEF_bn_faces:

bn_faces = (λ*_2426671*::*bn_graph*. *fst* (*dest_bn_graph* *_2426671*))

thm Tame_classification.bn_faces:

$\forall$ *g*::*bn_graph*. *bn_faces g* = *fst* (*dest_bn_graph g*)

thm DEF_bn_Faces:

bn_Faces = (λ*_2426676*::*bn_graph*. *set_of_list* (*bn_faces* *_2426676*))

thm Tame_classification.bn_Faces:

$\forall g{::}bn\_graph.\ bn\_Faces\ g = set\_of\_list\ (bn\_faces\ g)$

thm DEF_bn_countVertices:

$bn\_countVertices = (\lambda\_2426681{::}bn\_graph.\ fst\ (snd\ (dest\_bn\_graph\ \_2426681)))$

thm Tame_classification.bn_countVertices:

$\forall g{::}bn\_graph.\ bn\_countVertices\ g = fst\ (snd\ (dest\_bn\_graph\ g))$

thm DEF_bn_vertices_graph:

$bn\_vertices\_graph = (\lambda\_2426686{::}bn\_graph.\ dotdot\ (0{::}nat)\ (bn\_countVertices\ \_2426686 - (1{::}nat)))$

thm Tame_classification.bn_vertices_graph:

$\forall g{::}bn\_graph.\ bn\_vertices\_graph\ g = dotdot\ (0{::}nat)\ (bn\_countVertices\ g - (1{::}nat))$

thm DEF_bn_faceListAt:

$bn\_faceListAt = (\lambda\_2426691{::}bn\_graph.\ fst\ (snd\ (snd\ (dest\_bn\_graph\ \_2426691))))$

thm Tame_classification.bn_faceListAt:

$\forall g{::}bn\_graph.\ bn\_faceListAt\ g = fst\ (snd\ (snd\ (dest\_bn\_graph\ g)))$

thm DEF_bn_facesAt:

$bn\_facesAt = (\lambda(\_2426696{::}bn\_graph)\ \_2426697{::}nat.\ EL\ \_2426697\ (bn\_faceListAt\ \_2426696))$

thm Tame_classification.bn_facesAt:

$\forall (v{::}nat)\ g{::}bn\_graph.\ bn\_facesAt\ g\ v = EL\ v\ (bn\_faceListAt\ g)$

thm DEF_bn_heights:

$bn\_heights = (\lambda\_2426708{::}bn\_graph.\ snd\ (snd\ (snd\ (dest\_bn\_graph\ \_2426708))))$

thm Tame_classification.bn_heights:

$\forall g{::}bn\_graph.\ bn\_heights\ g = snd\ (snd\ (snd\ (dest\_bn\_graph\ g)))$

thm DEF_bn_height:

$bn\_height = (\lambda(\_2426713{::}bn\_graph)\ \_2426714{::}nat.\ EL\ \_2426714\ (bn\_heights\ \_2426713))$

thm Tame_classification.bn_height:

$\forall (v{::}nat)\ g{::}bn\_graph.\ bn\_height\ g\ v = EL\ v\ (bn\_heights\ g)$

thm DEF_LIST_TO:

$LIST\_TO = (SOME\ LIST\_TO{::}nat \Rightarrow nat \Rightarrow nat\ list.\ \forall\ \_2426728{::}nat.\ LIST\_TO\ \_2426728\ (0{::}nat) = [] \wedge (\forall n{::}nat.\ LIST\_TO\ \_2426728\ (Suc\ n) = LIST\_TO\ \_2426728\ n\ @\ [n]))\ (129{::}nat)$

thm Tame_classification.LIST_TO_conjunct0:

$LIST\_TO\ (0::nat) = []$

thm Tame_classification.LIST_TO_conjunct1:

$LIST\_TO\ (Suc\ (?n::nat)) = LIST\_TO\ ?n\ @\ [?n]$

thm Tame_classification.LIST_TO:

$LIST\_TO\ (0::nat) = [] \wedge LIST\_TO\ (Suc\ (?n::nat)) = LIST\_TO\ ?n\ @\ [?n]$

thm DEF_UPT:

$UPT = (SOME\ UPT::nat \Rightarrow nat \Rightarrow nat \Rightarrow nat\ list.\ \forall\ \_2426735::nat.\ (\forall\ m::nat.\ UPT\ \_2426735\ m\ (0::nat) = []) \wedge (\forall\ (m::nat)\ n::nat.\ UPT\ \_2426735\ m\ (Suc\ n) = (if\ n < m\ then\ []\ else\ UPT\ \_2426735\ m\ n\ @\ [n]))) (130::nat)$

thm Tame_classification.UPT_conjunct0:

$UPT\ (?m::nat)\ (0::nat) = []$

thm Tame_classification.UPT_conjunct1:

$UPT\ (?m::nat)\ (Suc\ (?n::nat)) = (if\ ?n < ?m\ then\ []\ else\ UPT\ ?m\ ?n\ @\ [?n])$

thm Tame_classification.UPT:

$UPT\ (?m::nat)\ (0::nat) = [] \wedge UPT\ ?m\ (Suc\ (?n::nat)) = (if\ ?n < ?m\ then\ []\ else\ UPT\ ?m\ ?n\ @\ [?n])$

thm DEF_bn_graph:

$bn\_graph = (\lambda\_2426736::nat.\ LET\ (\lambda vs::nat\ list.\ LET\_END\ (LET\ (\lambda fs::(nat\ list \times bool)\ list.\ LET\_END\ (mk\_bn\_graph\ (fs,\ \_2426736,\ replicate\ \_2426736\ fs,\ replicate\ \_2426736\ (0::nat))))\ [(vs,\ True),\ (vs,\ False)]))\ (LIST\_TO\ \_2426736))$

thm Tame_classification.bn_graph:

$\forall\ n::nat.\ bn\_graph\ n = LET\ (\lambda vs::nat\ list.\ LET\_END\ (LET\ (\lambda fs::(nat\ list \times bool)\ list.\ LET\_END\ (mk\_bn\_graph\ (fs,\ n,\ replicate\ n\ fs,\ replicate\ n\ (0::nat))))\ [(vs,\ True),\ (vs,\ False)]))\ (LIST\_TO\ n)$

thm DEF_bn_finals:

$bn\_finals = (\lambda\_2426741::bn\_graph.\ filter\ bn\_final\_face\ (bn\_faces\ \_2426741))$

thm Tame_classification.bn_finals:

$\forall\ g::bn\_graph.\ bn\_finals\ g = filter\ bn\_final\_face\ (bn\_faces\ g)$

thm DEF_bn_nonFinals:

$bn\_nonFinals = (\lambda\_2426746::bn\_graph.\ [r::nat\ list \times bool \leftarrow bn\_faces\ \_2426746\ .\ \neg\ bn\_final\_face\ r])$

thm Tame_classification.bn_nonFinals:

$\forall g$::*bn_graph*. *bn_nonFinals g* = [*r*::*nat list* × *bool*←*bn_faces g* . ¬ *bn_final_face r*]

thm DEF_bn_countNonFinals:

bn_countNonFinals = (λ*_2426751*::*bn_graph*. *length* (*bn_nonFinals _2426751*))

thm Tame_classification.bn_countNonFinals:

∀ *g*::*bn_graph*. *bn_countNonFinals g* = *length* (*bn_nonFinals g*)

thm DEF_bn_finalGraph:

bn_finalGraph = (λ*_2426756*::*bn_graph*. *bn_countNonFinals _2426756* = (*0*::*nat*))

thm Tame_classification.bn_finalGraph:

∀ *g*::*bn_graph*. *bn_finalGraph g* = (*bn_countNonFinals g* = (*0*::*nat*))

thm DEF_bn_finalVertex:

bn_finalVertex = (λ(*_2426761*::*bn_graph*) *_2426762*::*nat*. ∀ *f*::*nat list* × *bool*. *set_of_list* (*bn_facesAt _2426761 _2426762*) *f* ⟶ *bn_final_face f*)

thm Tame_classification.bn_finalVertex:

∀ (*g*::*bn_graph*) *v*::*nat*. *bn_finalVertex g v* = (∀ *f*::*nat list* × *bool*. *set_of_list* (*bn_facesAt g v*) *f* ⟶ *bn_final_face f*)

thm DEF_bn_degree:

bn_degree = (λ(*_2426773*::*bn_graph*) *_2426774*::*nat*. *length* (*bn_facesAt _2426773 _2426774*))

thm Tame_classification.bn_degree:

∀ (*g*::*bn_graph*) *v*::*nat*. *bn_degree g v* = *length* (*bn_facesAt g v*)

thm DEF_bn_tri:

bn_tri = (λ(*_2426785*::*bn_graph*) *_2426786*::*nat*. *length* [*f*::*nat list* × *bool*←*bn_facesAt _2426785 _2426786* . *bn_final_face f* ∧ *length* (*bn_vertices_face f*) = (*3*::*nat*)])

thm Tame_classification.bn_tri:

∀ (*g*::*bn_graph*) *v*::*nat*. *bn_tri g v* = *length* [*f*::*nat list* × *bool*←*bn_facesAt g v* . *bn_final_face f* ∧ *length* (*bn_vertices_face f*) = (*3*::*nat*)]

thm DEF_bn_quad:

bn_quad = (λ(*_2426797*::*bn_graph*) *_2426798*::*nat*. *length* [*f*::*nat list* × *bool*←*bn_facesAt _2426797 _2426798* . *bn_final_face f* ∧ *length* (*bn_vertices_face f*) = (*4*::*nat*)])

thm Tame_classification.bn_quad:

∀ (*g*::*bn_graph*) *v*::*nat*. *bn_quad g v* = *length* [*f*::*nat list* × *bool*←*bn_facesAt g v* . *bn_final_face f* ∧ *length* (*bn_vertices_face f*) = (*4*::*nat*)]

thm DEF_bn_except:

$bn\_except = (\lambda(\_2426809::bn\_graph)\ \_2426810::nat.\ length\ [f::nat\ list \times bool \leftarrow bn\_facesAt\ \_2426809\ \_2426810\ .\ bn\_final\_face\ f \wedge (5::nat) \leq length\ (bn\_vertices\_face\ f)])$

thm Tame_classification.bn_except:

$\forall (g::bn\_graph)\ v::nat.\ bn\_except\ g\ v = length\ [f::nat\ list \times bool \leftarrow bn\_facesAt\ g\ v\ .\ bn\_final\_face\ f \wedge (5::nat) \leq length\ (bn\_vertices\_face\ f)]$

thm DEF_bn_vertextype:

$bn\_vertextype = (\lambda(\_2426821::bn\_graph)\ \_2426822::nat.\ (bn\_tri\ \_2426821\ \_2426822,\ bn\_quad\ \_2426821\ \_2426822,\ bn\_except\ \_2426821\ \_2426822))$

thm Tame_classification.bn_vertextype:

$\forall (g::bn\_graph)\ v::nat.\ bn\_vertextype\ g\ v = (bn\_tri\ g\ v,\ bn\_quad\ g\ v,\ bn\_except\ g\ v)$

thm DEF_bn_exceptionalVertex:

$bn\_exceptionalVertex = (\lambda(\_2426833::bn\_graph)\ \_2426834::nat.\ bn\_except\ \_2426833\ \_2426834 \neq (0::nat))$

thm Tame_classification.bn_exceptionalVertex:

$\forall (g::bn\_graph)\ v::nat.\ bn\_exceptionalVertex\ g\ v = (bn\_except\ g\ v \neq (0::nat))$

thm DEF_bn_noExceptionals:

$bn\_noExceptionals = (\lambda(\_2426845::bn\_graph)\ \_2426846::nat \Rightarrow bool.\ \forall v::nat.\ \_2426846\ v \longrightarrow \neg\ bn\_exceptionalVertex\ \_2426845\ v)$

thm Tame_classification.bn_noExceptionals:

$\forall (V::nat \Rightarrow bool)\ g::bn\_graph.\ bn\_noExceptionals\ g\ V = (\forall v::nat.\ V\ v \longrightarrow \neg\ bn\_exceptionalVertex\ g\ v)$

thm DEF_bn_edges_graph:

$bn\_edges\_graph = (\lambda\_2426857::bn\_graph.\ UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%279::nat \times nat \Rightarrow bool.\ \exists f::nat\ list \times bool.\ SETSPEC\ GEN\%PVAR\%279\ (bn\_Faces\ \_2426857\ f)\ (bn\_edges\ f))))$

thm Tame_classification.bn_edges_graph:

$\forall g::bn\_graph.\ bn\_edges\_graph\ g = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%279::nat \times nat \Rightarrow bool.\ \exists f::nat\ list \times bool.\ SETSPEC\ GEN\%PVAR\%279\ (bn\_Faces\ g\ f)\ (bn\_edges\ f)))$

thm DEF_bn_neighbors:

$bn\_neighbors = (\lambda(\_2426862::bn\_graph)\ \_2426863::nat.\ map\ (\lambda f::nat\ list \times bool.\ bn\_nextVertex\ f\ \_2426863)\ (bn\_facesAt\ \_2426862\ \_2426863))$

thm Tame_classification.bn_neighbors:

$\forall (g::bn\_graph)\ v::nat.\ bn\_neighbors\ g\ v = map\ (\lambda f::nat\ list \times bool.\ bn\_nextVertex\ f\ v)\ (bn\_facesAt\ g\ v)$

thm DEF_bn_directedLength:

$bn\_directedLength = (\lambda(\_2426874::?'b::type\ list \times ?'a::type)\ (\_2426875::?'b::type)\ \_2426876::?'b::type.\ if\ \_2426875 = \_2426876\ then\ 0::nat\ else\ length\ (bn\_between\ (bn\_vertices\_face\ \_2426874)\ \_2426875\ \_2426876) + (1::nat))$

thm Tame_classification.bn_directedLength:

$\forall (f::?'b::type\ list \times ?'a::type)\ (a::?'b::type)\ b::?'b::type.\ bn\_directedLength\ f\ a\ b = (if\ a = b\ then\ 0::nat\ else\ length\ (bn\_between\ (bn\_vertices\_face\ f)\ a\ b) + (1::nat))$

thm DEF_bn_tabulate0:

$bn\_tabulate0 = (\lambda\_2426895::nat \times (nat \Rightarrow ?'a::type).\ map\ (snd\ \_2426895)\ (LIST\_TO\ (fst\ \_2426895)))$

thm Tame_classification.bn_tabulate0:

$\forall p::nat \times (nat \Rightarrow ?'a::type).\ bn\_tabulate0\ p = map\ (snd\ p)\ (LIST\_TO\ (fst\ p))$

thm DEF_bn_tabulate:

$bn\_tabulate = (\lambda(\_2426900::nat)\ \_2426901::nat \Rightarrow ?'a::type.\ bn\_tabulate0\ (\_2426900, \_2426901))$

thm Tame_classification.bn_tabulate:

$\forall (n::nat)\ f::nat \Rightarrow ?'a::type.\ bn\_tabulate\ n\ f = bn\_tabulate0\ (n, f)$

thm DEF_bn_tabulate2:

$bn\_tabulate2 = (\lambda(\_2426912::nat)\ (\_2426913::nat)\ \_2426914::nat \Rightarrow nat \Rightarrow ?'a::type.\ bn\_tabulate\ \_2426912\ (\lambda i::nat.\ bn\_tabulate\ \_2426913\ (\_2426914\ i)))$

thm Tame_classification.bn_tabulate2:

$\forall (m::nat)\ (n::nat)\ f::nat \Rightarrow nat \Rightarrow ?'a::type.\ bn\_tabulate2\ m\ n\ f = bn\_tabulate\ m\ (\lambda i::nat.\ bn\_tabulate\ n\ (f\ i))$

thm DEF_bn_tabulate3:

$bn\_tabulate3 = (\lambda(\_2426933::nat)\ (\_2426934::nat)\ (\_2426935::nat)\ \_2426936::nat \Rightarrow nat \Rightarrow nat \Rightarrow ?'a::type.\ bn\_tabulate\ \_2426933\ (\lambda i::nat.\ bn\_tabulate\ \_2426934\ (\lambda j::nat.\ bn\_tabulate\ \_2426935\ (\_2426936\ i\ j))))$

thm Tame_classification.bn_tabulate3:

$\forall (l::nat)\ (m::nat)\ (n::nat)\ f::nat \Rightarrow nat \Rightarrow nat \Rightarrow ?'a::type.\ bn\_tabulate3\ l\ m\ n\ f = bn\_tabulate\ l\ (\lambda i::nat.\ bn\_tabulate\ m\ (\lambda j::nat.\ bn\_tabulate\ n\ (f\ i\ j)))$

thm DEF_bn_sub1:

$bn\_sub1 = (\lambda\_2426965::?'a::type\ list \times nat.\ EL\ (snd\ \_2426965)\ (fst\ \_2426965))$

thm Tame_classification.bn_sub1:

$\forall (n::nat)\ xs::?'a::type\ list.\ bn\_sub1\ (xs,\ n) = EL\ n\ xs$

thm DEF_bn_sub:

$bn\_sub = (\lambda(\_2426974::?'a::type\ list)\ \_2426975::nat.\ bn\_sub1\ (\_2426974,\ \_2426975))$

thm Tame_classification.bn_sub:

$\forall (a::?'a::type\ list)\ n::nat.\ bn\_sub\ a\ n = bn\_sub1\ (a,\ n)$

thm DEF_bn_enumBase:

$bn\_enumBase = (\lambda\_2426986::nat.\ map\ (\lambda i::nat.\ [i])\ (LIST\_TO\ (Suc\ \_2426986)))$

thm Tame_classification.bn_enumBase:

$\forall\, nmax::nat.\ bn\_enumBase\ nmax = map\ (\lambda i::nat.\ [i])\ (LIST\_TO\ (Suc\ nmax))$

thm DEF_bn_enumAppend:

$bn\_enumAppend = (\lambda(\_2426991::nat)\ \_2426992::nat\ list\ list.\ bn\_concat\ (map\ (\lambda is::nat\ list.\ map\ (\lambda n::nat.\ is\ @\ [n])\ (UPT\ (last\ is)\ (Suc\ \_2426991)))\ \_2426992))$

thm Tame_classification.bn_enumAppend:

$\forall (nmax::nat)\ iss::nat\ list\ list.\ bn\_enumAppend\ nmax\ iss = bn\_concat\ (map\ (\lambda is::nat\ list.\ map\ (\lambda n::nat.\ is\ @\ [n])\ (UPT\ (last\ is)\ (Suc\ nmax)))\ iss)$

thm DEF_bn_enumerator:

$bn\_enumerator = (\lambda(\_2427003::nat)\ \_2427004::nat.\ LET\ (\lambda nmax::nat.\ LET\_END\ (LET\ (\lambda k::nat.\ LET\_END\ (map\ (\lambda is::nat\ list.\ [0::nat]\ @\ is\ @\ [\_2427004 - (1::nat)])\ (POWER\ (bn\_enumAppend\ nmax)\ k\ (bn\_enumBase\ nmax))))\ (\_2427003 - (3::nat))))\ (\_2427004 - (2::nat)))$

thm Tame_classification.bn_enumerator:

$\forall (inner::nat)\ outer::nat.\ bn\_enumerator\ inner\ outer = LET\ (\lambda nmax::nat.\ LET\_END\ (LET\ (\lambda k::nat.\ LET\_END\ (map\ (\lambda is::nat\ list.\ [0::nat]\ @\ is\ @\ [outer - (1::nat)])\ (POWER\ (bn\_enumAppend\ nmax)\ k\ (bn\_enumBase\ nmax))))\ (inner - (3::nat))))\ (outer - (2::nat))$

thm Tame_classification.bn_enumTab:

$bn\_enumTab = bn\_tabulate2\ (9::nat)\ (9::nat)\ bn\_enumerator$

thm DEF_bn_enumt:

$bn\_enumt = (\lambda(\_2427015::nat)\ \_2427016::nat.\ if\ \_2427015 < (9::nat) \land \_2427016 < (9::nat)\ then\ bn\_sub\ (bn\_sub\ bn\_enumTab\ \_2427015)\ \_2427016\ else\ bn\_enumerator\ \_2427015\ \_2427016)$

thm Tame_classification.bn_enumt:

$\forall (inner::nat)\ outer::nat.\ bn\_enumt\ inner\ outer = (if\ inner < (9::nat) \land outer < (9::nat)\ then\ bn\_sub\ (bn\_sub\ bn\_enumTab\ inner)\ outer\ else\ bn\_enumerator\ inner\ outer)$

thm DEF_bn_hideDupsRec:

bn_hideDupsRec = (*SOME bn_hideDupsRec*::*nat* ⇒ *?'a*::*type* ⇒ *?'a*::*type list* ⇒ *?'a*::*type HOL_Light_Import.option list*. ∀ *_2427033*::*nat*. (∀ *a*::*?'a*::*type*. *bn_hideDupsRec _2427033 a* [] = []) ∧ (∀ (*a*::*?'a*::*type*) (*b*::*?'a*::*type*) *bs*::*?'a*::*type list*. *bn_hideDupsRec _2427033 a* (*b* # *bs*) = (*if a* = *b then NONE* # *bn_hideDupsRec _2427033 b bs else SOME b* # *bn_hideDupsRec _2427033 b bs*))) (*131*::*nat*)

thm Tame_classification.bn_hideDupsRec_conjunct0:

bn_hideDupsRec (*?a*::*?'a*::*type*) [] = []

thm Tame_classification.bn_hideDupsRec_conjunct1:

bn_hideDupsRec (*?a*::*?'a*::*type*) ((*?b*::*?'a*::*type*) # (*?bs*::*?'a*::*type list*)) = (*if ?a* = *?b then NONE* # *bn_hideDupsRec ?b ?bs else SOME ?b* # *bn_hideDupsRec ?b ?bs*)

thm Tame_classification.bn_hideDupsRec:

bn_hideDupsRec (*?a*::*?'a*::*type*) [] = [] ∧ *bn_hideDupsRec ?a* ((*?b*::*?'a*::*type*) # (*?bs*::*?'a*::*type list*)) = (*if ?a* = *?b then NONE* # *bn_hideDupsRec ?b ?bs else SOME ?b* # *bn_hideDupsRec ?b ?bs*)

thm DEF_bn_hideDups:

bn_hideDups = (*SOME bn_hideDups*::*nat* ⇒ *?'a*::*type list* ⇒ *?'a*::*type HOL_Light_Import.option list*. ∀ *_2427037*::*nat*. *bn_hideDups _2427037* [] = [] ∧ (∀ (*b*::*?'a*::*type*) *bs*::*?'a*::*type list*. *bn_hideDups _2427037* (*b* # *bs*) = *SOME b* # *bn_hideDupsRec b bs*)) (*132*::*nat*)

thm Tame_classification.bn_hideDups_conjunct0:

bn_hideDups [] = []

thm Tame_classification.bn_hideDups_conjunct1:

bn_hideDups ((*?b*::*?'a*::*type*) # (*?bs*::*?'a*::*type list*)) = *SOME ?b* # *bn_hideDupsRec ?b ?bs*

thm Tame_classification.bn_hideDups:

bn_hideDups [] = [] ∧ *bn_hideDups* ((*?b*::*?'a*::*type*) # (*?bs*::*?'a*::*type list*)) = *SOME ?b* # *bn_hideDupsRec ?b ?bs*

thm DEF_bn_indexToVertexList:

bn_indexToVertexList = (λ(*_2427038*::*?'a*::*type list* × *bool*) (*_2427039*::*?'a*::*type*) *_2427040*::*nat list*. *bn_hideDups* (*map* (λ*k*::*nat*. *bn_nextVertices _2427038 k _2427039*) *_2427040*))

thm Tame_classification.bn_indexToVertexList:

∀ (*f*::*?'a*::*type list* × *bool*) (*v*::*?'a*::*type*) *is*::*nat list*. *bn_indexToVertexList f v is* = *bn_hideDups* (*map* (λ*k*::*nat*. *bn_nextVertices f k v*) *is*)

thm DEF_bn_split_face:

bn_split_face = (λ(_2427059::?'b::type list × ?'a::type) (_2427060::?'b::type) (_2427061::?'b::type) _2427062::?'b::type list. LET (λvs::?'b::type list. LET_END (LET (λf1::?'b::type list. LET_END (LET (λf2::?'b::type list. LET_END ((rev _2427062 @ f1, False), f2 @ _2427062, False)) ([_2427061] @ bn_between vs _2427061 _2427060 @ [_2427060]))) ([_2427060] @ bn_between vs _2427060 _2427061 @ [_2427061]))) (bn_vertices_face _2427059))

thm Tame_classification.bn_split_face:

∀ (newVs::?'b::type list) (ram1::?'b::type) (ram2::?'b::type) f::?'b::type list × ?'a::type. bn_split_face f ram1 ram2 newVs = LET (λvs::?'b::type list. LET_END (LET (λf1::?'b::type list. LET_END (LET (λf2::?'b::type list. LET_END ((rev newVs @ f1, False), f2 @ newVs, False)) ([ram2] @ bn_between vs ram2 ram1 @ [ram1]))) ([ram1] @ bn_between vs ram1 ram2 @ [ram2]))) (bn_vertices_face f)

thm DEF_bn_replacefacesAt:

bn_replacefacesAt = (λ(_2427091::nat list) (_2427092::?'a::type) _2427093::?'a::type list. bn_mapAt _2427091 (bn_replace _2427092 _2427093))

thm Tame_classification.bn_replacefacesAt:

∀ (ns::nat list) (f::?'a::type) (fs::?'a::type list) Fs::?'a::type list list. bn_replacefacesAt ns f fs Fs = bn_mapAt ns (bn_replace f fs) Fs

thm DEF_bn_makeFaceFinalFaceList:

bn_makeFaceFinalFaceList = (λ_2427123::?'a::type × bool. bn_replace _2427123 [bn_setFinal _2427123])

thm Tame_classification.bn_makeFaceFinalFaceList:

∀ (f::?'a::type × bool) fs::(?'a::type × bool) list. bn_makeFaceFinalFaceList f fs = bn_replace f [bn_setFinal f] fs

thm DEF_bn_makeFaceFinal:

bn_makeFaceFinal = (λ(_2427135::nat list × bool) _2427136::bn_graph. mk_bn_graph (bn_makeFaceFinalFaceList _2427135 (bn_faces _2427136), bn_countVertices _2427136, map (bn_makeFaceFinalFaceList _2427135) (bn_faceListAt _2427136), bn_heights _2427136))

thm Tame_classification.bn_makeFaceFinal:

∀ (f::nat list × bool) g::bn_graph. bn_makeFaceFinal f g = mk_bn_graph (bn_makeFaceFinalFaceList f (bn_faces g), bn_countVertices g, map (bn_makeFaceFinalFaceList f) (bn_faceListAt g), bn_heights g)

thm DEF_bn_heightsNewVertices:

bn_heightsNewVertices = (λ(_2427147::nat) (_2427148::nat) _2427149::nat. map (λi::nat. min_num (INSERT (_2427147 + (i + (1::nat))) (INSERT (_2427148 + (_2427149 − i)) EMPTY))) (LIST_TO _2427149))

thm Tame_classification.bn_heightsNewVertices:

$\forall\,(h1::nat)\ (h2::nat)\ n::nat.\ bn\_heightsNewVertices\ h1\ h2\ n = map\ (\lambda i::nat.\ min\_num\ (INSERT\ (h1 + (i + (1::nat)))\ (INSERT\ (h2 + (n - i))\ EMPTY)))\ (LIST\_TO\ n)$

thm DEF_bn_splitFace:

$bn\_splitFace = (\lambda(\_2427168::bn\_graph)\ (\_2427169::nat)\ (\_2427170::nat)\ (\_2427171::nat\ list \times bool)\ \_2427172::nat\ list.\ LET\ (\lambda fs::(nat\ list \times bool)\ list.\ LET\_END\ (LET\ (\lambda n::nat.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda h::nat\ list.\ LET\_END\ (LET\ (\lambda lVs::nat.\ LET\_END\ (LET\ (\lambda vs1::nat\ list.\ LET\_END\ (LET\ (\lambda vs2::nat\ list.\ LET\_END\ (LET\ (GABS\ (\lambda f::(nat\ list \times bool) \times nat\ list \times bool \Rightarrow (nat\ list \times bool) \times (nat\ list \times bool) \times bn\_graph.\ \forall\,(f1::nat\ list \times bool)\ f2::nat\ list \times bool.\ GEQ\ (f\ (f1,\ f2))\ (LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (f1,\ f2,\ mk\_bn\_graph\ (bn\_replace\ \_2427171\ [f2]\ fs\ @\ [f1],\ n + lVs,\ Fs,\ h\ @\ bn\_heightsNewVertices\ (EL\ \_2427169\ h)\ (EL\ \_2427170\ h)\ lVs)))\ (Fs\ @\ replicate\ lVs\ [f1,\ f2])))\ (bn\_replacefacesAt\ [\_2427170]\ \_2427171\ [f1,\ f2]\ Fs)))\ (bn\_replacefacesAt\ [\_2427169]\ \_2427171\ [f2,\ f1]\ Fs)))\ (bn\_replacefacesAt\ vs2\ \_2427171\ [f2]\ Fs)))\ (bn\_replacefacesAt\ vs1\ \_2427171\ [f1]\ Fs)))))\ (bn\_split\_face\ \_2427171\ \_2427169\ \_2427170\ \_2427172)))\ (bn\_between\ (bn\_vertices\_face\ \_2427171)\ \_2427170\ \_2427169)))\ (bn\_between\ (bn\_vertices\_face\ \_2427171)\ \_2427169\ \_2427170)))\ (length\ \_2427172)))\ (bn\_heights\ \_2427168)))\ (bn\_faceListAt\ \_2427168)))\ (bn\_countVertices\ \_2427168)))\ (bn\_faces\ \_2427168))$

thm Tame_classification.bn_splitFace:

$\forall\,(oldF::nat\ list \times bool)\ (ram1::nat)\ (ram2::nat)\ (newVs::nat\ list)\ g::bn\_graph.\ bn\_splitFace\ g\ ram1\ ram2\ oldF\ newVs = LET\ (\lambda fs::(nat\ list \times bool)\ list.\ LET\_END\ (LET\ (\lambda n::nat.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda h::nat\ list.\ LET\_END\ (LET\ (\lambda lVs::nat.\ LET\_END\ (LET\ (\lambda vs1::nat\ list.\ LET\_END\ (LET\ (\lambda vs2::nat\ list.\ LET\_END\ (LET\ (GABS\ (\lambda f::(nat\ list \times bool) \times nat\ list \times bool \Rightarrow (nat\ list \times bool) \times (nat\ list \times bool) \times bn\_graph.\ \forall\,(f1::nat\ list \times bool)\ f2::nat\ list \times bool.\ GEQ\ (f\ (f1,\ f2))\ (LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (LET\ (\lambda Fs::(nat\ list \times bool)\ list\ list.\ LET\_END\ (f1,\ f2,\ mk\_bn\_graph\ (bn\_replace\ oldF\ [f2]\ fs\ @\ [f1],\ n + lVs,\ Fs,\ h\ @\ bn\_heightsNewVertices\ (EL\ ram1\ h)\ (EL\ ram2\ h)\ lVs)))\ (Fs\ @\ replicate\ lVs\ [f1,\ f2])))\ (bn\_replacefacesAt\ [ram2]\ oldF\ [f1,\ f2]\ Fs)))\ (bn\_replacefacesAt\ [ram1]\ oldF\ [f2,\ f1]\ Fs)))\ (bn\_replacefacesAt\ vs2\ oldF\ [f2]\ Fs)))\ (bn\_replacefacesAt\ vs1\ oldF\ [f1]\ Fs)))))\ (bn\_split\_face\ oldF\ ram1\ ram2\ newVs)))\ (bn\_between\ (bn\_vertices\_face\ oldF)\ ram2\ ram1)))\ (bn\_between\ (bn\_vertices\_face\ oldF)\ ram1\ ram2)))\ (length\ newVs)))\ (bn\_heights\ g)))\ (bn\_faceListAt\ g)))\ (bn\_countVertices\ g)))\ (bn\_faces\ g)$

thm DEF_bn_subdivFace0:

bn_subdivFace0 = (SOME bn_subdivFace0::nat ⇒ bn_graph ⇒ nat list × bool ⇒ nat ⇒ nat ⇒ nat HOL_Light_Import.option list ⇒ bn_graph. ∀ _2427222::nat. (∀ (u::nat) (n::nat) (f::nat list × bool) g::bn_graph. bn_subdivFace0 _2427222 g f u n [] = bn_makeFaceFinal f g) ∧ (∀ (vos::nat HOL_Light_Import.option list) (u::nat) (f::nat list × bool) (g::bn_graph) (n::nat) vo::nat HOL_Light_Import.option. bn_subdivFace0 _2427222 g f u n (vo # vos) = (if vo = NONE then bn_subdivFace0 _2427222 g f u (Suc n) vos else LET (λv::nat. LET_END (if bn_nextVertex f u = v ∧ n = (0::nat) then bn_subdivFace0 _2427222 g f v (0::nat) vos else LET (λws::nat list. LET_END (LET (GABS (λf::(nat list × bool) × (nat list × bool) × bn_graph ⇒ bn_graph. ∀ (f1::nat list × bool) (f2::nat list × bool) g′::bn_graph. GEQ (f (f1, f2, g′)) (LET_END (bn_subdivFace0 _2427222 g′ f2 v (0::nat) vos)))) (bn_splitFace g u v f ws))) (UPT (bn_countVertices g) (bn_countVertices g + n)))) (bn_the vo)))) (133::nat)

thm Tame_classification.bn_subdivFace0_conjunct0:

bn_subdivFace0 (?g::bn_graph) (?f::nat list × bool) (?u::nat) (?n::nat) [] = bn_makeFaceFinal ?f ?g

thm Tame_classification.bn_subdivFace0_conjunct1:

bn_subdivFace0 (?g::bn_graph) (?f::nat list × bool) (?u::nat) (?n::nat) ((?vo::nat HOL_Light_Import.option) # (?vos::nat HOL_Light_Import.option list)) = (if ?vo = NONE then bn_subdivFace0 ?g ?f ?u (Suc ?n) ?vos else LET (λv::nat. LET_END (if bn_nextVertex ?f ?u = v ∧ ?n = (0::nat) then bn_subdivFace0 ?g ?f v (0::nat) ?vos else LET (λws::nat list. LET_END (LET (GABS (λf::(nat list × bool) × (nat list × bool) × bn_graph ⇒ bn_graph. ∀ (f1::nat list × bool) (f2::nat list × bool) g′::bn_graph. GEQ (f (f1, f2, g′)) (LET_END (bn_subdivFace0 g′ f2 v (0::nat) ?vos)))) (bn_splitFace ?g ?u v ?f ws))) (UPT (bn_countVertices ?g) (bn_countVertices ?g + ?n)))) (bn_the ?vo))

thm Tame_classification.bn_subdivFace0:

bn_subdivFace0 (?g::bn_graph) (?f::nat list × bool) (?u::nat) (?n::nat) [] = bn_makeFaceFinal ?f ?g ∧ bn_subdivFace0 ?g ?f ?u ?n ((?vo::nat HOL_Light_Import.option) # (?vos::nat HOL_Light_Import.option list)) = (if ?vo = NONE then bn_subdivFace0 ?g ?f ?u (Suc ?n) ?vos else LET (λv::nat. LET_END (if bn_nextVertex ?f ?u = v ∧ ?n = (0::nat) then bn_subdivFace0 ?g ?f v (0::nat) ?vos else LET (λws::nat list. LET_END (LET (GABS (λf::(nat list × bool) × (nat list × bool) × bn_graph ⇒ bn_graph. ∀ (f1::nat list × bool) (f2::nat list × bool) g′::bn_graph. GEQ (f (f1, f2, g′)) (LET_END (bn_subdivFace0 g′ f2 v (0::nat) ?vos)))) (bn_splitFace ?g ?u v ?f ws))) (UPT (bn_countVertices ?g) (bn_countVertices ?g + ?n)))) (bn_the ?vo))

thm DEF_bn_subdivFace:

bn_subdivFace = (λ(_2427223::bn_graph) (_2427224::nat list × bool) _2427225::nat HOL_Light_Import.option list. bn_subdivFace0 _2427223 _2427224 (bn_the (hd _2427225)) (0::nat) (tl _2427225))

thm Tame_classification.bn_subdivFace:

$\forall\,(g::bn\_graph)\ (f::nat\ list \times bool)\ vos::nat\ HOL\_Light\_Import.option\ list.\ bn\_subdivFace\ g\ f\ vos = bn\_subdivFace0\ g\ f\ (bn\_the\ (hd\ vos))\ (0::nat)\ (tl\ vos)$

thm DEF_bn_RTranCl:

$bn\_RTranCl = (\lambda\_2427244::?'a::type \Rightarrow ?'a::type\ list.\ UNCURRY\ (RTC\ (\lambda(x::?'a::type)\ y::?'a::type.\ MEM\ y\ (\_2427244\ x))))$

thm Tame_classification.bn_RTranCl:

$\forall\,g::?'a::type \Rightarrow ?'a::type\ list.\ bn\_RTranCl\ g = UNCURRY\ (RTC\ (\lambda(x::?'a::type)\ y::?'a::type.\ MEM\ y\ (g\ x)))$

thm DEF_bn_invariant:

$bn\_invariant = (\lambda(\_2427249::?'a::type \Rightarrow bool)\ \_2427250::?'a::type \Rightarrow ?'a::type\ list.\ \forall\,(g::?'a::type)\ g'::?'a::type.\ MEM\ g\ (\_2427250\ g) \longrightarrow \_2427249\ g \longrightarrow \_2427249\ g')$

thm Tame_classification.bn_invariant:

$\forall\,(succs::?'a::type \Rightarrow ?'a::type\ list)\ P::?'a::type \Rightarrow bool.\ bn\_invariant\ P\ succs = (\forall\,(g::?'a::type)\ g'::?'a::type.\ MEM\ g\ (succs\ g) \longrightarrow P\ g \longrightarrow P\ g')$

thm DEF_bn_maxGon:

$bn\_maxGon = (\lambda\_2427261::nat.\ \_2427261 + (3::nat))$

thm Tame_classification.bn_maxGon:

$\forall\,p::nat.\ bn\_maxGon\ p = p + (3::nat)$

thm DEF_bn_duplicateEdge:

$bn\_duplicateEdge = (\lambda(\_2427266::bn\_graph)\ (\_2427267::nat\ list \times ?'a::type)\ (\_2427268::nat)\ \_2427269::nat.\ (2::nat) \le bn\_directedLength\ \_2427267\ \_2427268\ \_2427269 \wedge (2::nat) \le bn\_directedLength\ \_2427267\ \_2427269\ \_2427268 \wedge set\_of\_list\ (bn\_neighbors\ \_2427266\ \_2427268)\ \_2427269)$

thm Tame_classification.bn_duplicateEdge:

$\forall\,(f::nat\ list \times ?'a::type)\ (g::bn\_graph)\ (a::nat)\ b::nat.\ bn\_duplicateEdge\ g\ f\ a\ b = ((2::nat) \le bn\_directedLength\ f\ a\ b \wedge (2::nat) \le bn\_directedLength\ f\ b\ a \wedge set\_of\_list\ (bn\_neighbors\ g\ a)\ b)$

thm DEF_bn_containsUnacceptableEdgeSnd:

$bn\_containsUnacceptableEdgeSnd = (SOME\ bn\_containsUnacceptableEdgeSnd::nat \Rightarrow (nat \Rightarrow nat \Rightarrow bool) \Rightarrow nat \Rightarrow nat\ list \Rightarrow bool.\ \forall\,\_2427305::nat.\ (\forall\,(N::nat \Rightarrow nat \Rightarrow bool)\ v::nat.\ bn\_containsUnacceptableEdgeSnd\ \_2427305\ N\ v\ [] = False) \wedge (\forall\,(v::nat)\ (N::nat \Rightarrow nat \Rightarrow bool)\ (w::nat)\ ws::nat\ list.\ bn\_containsUnacceptableEdgeSnd\ \_2427305\ N\ v\ (w\ \#\ ws) = (if\ length\ ws = (0::nat)\ then\ False\ else\ LET\ (\lambda w'::nat.\ LET\_END\ (LET\ (\lambda ws'::nat\ list.\ LET\_END\ (if\ v < w \wedge w < w' \wedge$

N w w′ then True else bn_containsUnacceptableEdgeSnd _2427305 N w ws)) (*tl ws*))) (*hd ws*)))) (*134*::*nat*)

thm Tame_classification.bn_containsUnacceptableEdgeSnd_conjunct0:

bn_containsUnacceptableEdgeSnd (*?N*::*nat* ⇒ *nat* ⇒ *bool*) (*?v*::*nat*) [] = *False*

thm Tame_classification.bn_containsUnacceptableEdgeSnd_conjunct1:

bn_containsUnacceptableEdgeSnd (*?N*::*nat* ⇒ *nat* ⇒ *bool*) (*?v*::*nat*) ((*?w*::*nat*) # (*?ws*::*nat list*)) = (*if length ?ws* = (*0*::*nat*) *then False else LET* (λ*w′*::*nat*. *LET_END* (*LET* (λ*ws′*::*nat list*. *LET_END* (*if ?v* < *?w* ∧ *?w* < *w′* ∧ *?N ?w w′ then True else bn_containsUnacceptableEdgeSnd ?N ?w ?ws*)) (*tl ?ws*))) (*hd ?ws*))

thm Tame_classification.bn_containsUnacceptableEdgeSnd:

bn_containsUnacceptableEdgeSnd (*?N*::*nat* ⇒ *nat* ⇒ *bool*) (*?v*::*nat*) [] = *False* ∧ *bn_containsUnacceptableEdgeSnd ?N ?v* ((*?w*::*nat*) # (*?ws*::*nat list*)) = (*if length ?ws* = (*0*::*nat*) *then False else LET* (λ*w′*::*nat*. *LET_END* (*LET* (λ*ws′*::*nat list*. *LET_END* (*if ?v* < *?w* ∧ *?w* < *w′* ∧ *?N ?w w′ then True else bn_containsUnacceptableEdgeSnd ?N ?w ?ws*)) (*tl ?ws*))) (*hd ?ws*))

thm DEF_bn_containsUnacceptableEdge:

bn_containsUnacceptableEdge = (*SOME bn_containsUnacceptableEdge*::*nat* ⇒ (*nat* ⇒ *nat* ⇒ *bool*) ⇒ *nat list* ⇒ *bool*. ∀ *_2427312*::*nat*. (∀ *N*::*nat* ⇒ *nat* ⇒ *bool*. *bn_containsUnacceptableEdge _2427312 N* [] = *False*) ∧ (∀ (*N*::*nat* ⇒ *nat* ⇒ *bool*) (*v*::*nat*) *vs*::*nat list*. *bn_containsUnacceptableEdge _2427312 N* (*v* # *vs*) = (*if length vs* = (*0*::*nat*) *then False else LET* (λ*w*::*nat*. *LET_END* (*LET* (λ*ws*::*nat list*. *LET_END* (*if v* < *w* ∧ *N v w then True else bn_containsUnacceptableEdgeSnd N v vs*)) (*tl vs*))) (*hd vs*)))) (*135*::*nat*)

thm Tame_classification.bn_containsUnacceptableEdge_conjunct0:

bn_containsUnacceptableEdge (*?N*::*nat* ⇒ *nat* ⇒ *bool*) [] = *False*

thm Tame_classification.bn_containsUnacceptableEdge_conjunct1:

bn_containsUnacceptableEdge (*?N*::*nat* ⇒ *nat* ⇒ *bool*) ((*?v*::*nat*) # (*?vs*::*nat list*)) = (*if length ?vs* = (*0*::*nat*) *then False else LET* (λ*w*::*nat*. *LET_END* (*LET* (λ*ws*::*nat list*. *LET_END* (*if ?v* < *w* ∧ *?N ?v w then True else bn_containsUnacceptableEdgeSnd ?N ?v ?vs*)) (*tl ?vs*))) (*hd ?vs*))

thm Tame_classification.bn_containsUnacceptableEdge:

bn_containsUnacceptableEdge (*?N*::*nat* ⇒ *nat* ⇒ *bool*) [] = *False* ∧ *bn_containsUnacceptableEdge ?N* ((*?v*::*nat*) # (*?vs*::*nat list*)) = (*if length ?vs* = (*0*::*nat*) *then False else LET* (λ*w*::*nat*. *LET_END* (*LET* (λ*ws*::*nat list*. *LET_END* (*if ?v* < *w* ∧ *?N ?v w then True else bn_containsUnacceptableEdgeSnd ?N ?v ?vs*)) (*tl ?vs*))) (*hd ?vs*))

thm DEF_bn_containsDuplicateEdge:

bn_containsDuplicateEdge = (λ(*_2427313*::*bn_graph*) (*_2427314*::*nat list* × *bool*) *_2427315*::*nat*. *bn_containsUnacceptableEdge* (λ(*i*::*nat*) *j*::*nat*. *bn_duplicateEdge*

_2427313 _2427314 (bn_nextVertices _2427314 i _2427315) (bn_nextVertices _2427314 j _2427315)))

thm Tame_classification.bn_containsDuplicateEdge:

∀ (g::bn_graph) (f::nat list × bool) (v::nat) is::nat list. bn_containsDuplicateEdge g f v is = bn_containsUnacceptableEdge (λ(i::nat) j::nat. bn_duplicateEdge g f (bn_nextVertices f i v) (bn_nextVertices f j v)) is

thm DEF_bn_containsDuplicateEdge0:

bn_containsDuplicateEdge0 = (λ(_2427345::bn_graph) (_2427346::nat list × bool) (_2427347::nat) _2427348::nat list. (2::nat) ≤ length _2427348 ∧ ((∃ k<length _2427348 − (2::nat). LET (λi0::nat. LET_END (LET (λi1::nat. LET_END (LET (λi2::nat. LET_END (bn_duplicateEdge _2427345 _2427346 (bn_nextVertices _2427346 i1 _2427347) (bn_nextVertices _2427346 i2 _2427347) ∧ i0 < i1 ∧ i1 < i2)) (EL (k + (2::nat)) _2427348))) (EL (k + (1::nat)) _2427348))) (EL k _2427348)) ∨ LET (λi0::nat. LET_END (LET (λi1::nat. LET_END (bn_duplicateEdge _2427345 _2427346 (bn_nextVertices _2427346 i0 _2427347) (bn_nextVertices _2427346 i1 _2427347) ∧ i0 < i1)) (EL (1::nat) _2427348))) (EL (0::nat) _2427348)))

thm Tame_classification.bn_containsDuplicateEdge0:

∀ (g::bn_graph) (f::nat list × bool) (v::nat) is::nat list. bn_containsDuplicateEdge0 g f v is = ((2::nat) ≤ length is ∧ ((∃ k<length is − (2::nat). LET (λi0::nat. LET_END (LET (λi1::nat. LET_END (LET (λi2::nat. LET_END (bn_duplicateEdge g f (bn_nextVertices f i1 v) (bn_nextVertices f i2 v) ∧ i0 < i1 ∧ i1 < i2)) (EL (k + (2::nat)) is))) (EL (k + (1::nat)) is))) (EL k is)) ∨ LET (λi0::nat. LET_END (LET (λi1::nat. LET_END (bn_duplicateEdge g f (bn_nextVertices f i0 v) (bn_nextVertices f i1 v) ∧ i0 < i1)) (EL (1::nat) is))) (EL (0::nat) is)))

thm DEF_bn_generatePolygon:

bn_generatePolygon = (λ(_2427377::nat) (_2427378::nat) (_2427379::nat list × bool) _2427380::bn_graph. LET (λenumeration::nat list list. LET_END (LET (λenumeration::nat list list. LET_END (LET (λvertexLists::nat HOL_Light_Import.option list list. LET_END (map (bn_subdivFace _2427380 _2427379) vertexLists)) (map (bn_indexToVertexList _2427379 _2427378) enumeration))) [is::nat list←enumeration . ¬ bn_containsDuplicateEdge _2427380 _2427379 _2427378 is])) (bn_enumerator _2427377 (length (bn_vertices_face _2427379))))

thm Tame_classification.bn_generatePolygon:

∀ (g::bn_graph) (v::nat) (n::nat) f::nat list × bool. bn_generatePolygon n v f g = LET (λenumeration::nat list list. LET_END (LET (λenumeration::nat list list. LET_END (LET (λvertexLists::nat HOL_Light_Import.option list list. LET_END (map (bn_subdivFace g f) vertexLists)) (map (bn_indexToVertexList f v) enumeration))) [is::nat list←enumeration . ¬ bn_containsDuplicateEdge g f v is])) (bn_enumerator n (length (bn_vertices_face f)))

thm DEF_c_union:

c_union = (λ(_2427409::?'b::type list) _2427410::?'b::type ⇒ ?'a::type list. bn_concat (map _2427410 _2427409))

thm Tame_classification.c_union:

∀ (r::?'b::type ⇒ ?'a::type list) xs::?'b::type list. c_union xs r = bn_concat (map r xs)

thm DEF_bn_Seed:

bn_Seed = (λ_2427421::nat. bn_graph (bn_maxGon _2427421))

thm Tame_classification.bn_Seed:

∀ p::nat. bn_Seed p = bn_graph (bn_maxGon p)

thm Tame_classification.bn_minimalFace:

bn_minimalFace = bn_minimal (length ∘ bn_vertices_face)

thm DEF_bn_minimalVertex:

bn_minimalVertex = (λ(_2427426::bn_graph) _2427427::nat list × ?'a::type. bn_minimal (bn_height _2427426) (bn_vertices_face _2427427))

thm Tame_classification.bn_minimalVertex:

∀ (g::bn_graph) f::nat list × ?'a::type. bn_minimalVertex g f = bn_minimal (bn_height g) (bn_vertices_face f)

thm DEF_bn_next_plane:

bn_next_plane = (λ(_2427438::nat) _2427439::bn_graph. LET (λfs::(nat list × bool) list. LET_END (if fs = [] then [] else LET (λf::nat list × bool. LET_END (LET (λv::nat. LET_END (c_union (UPT (3::nat) (Suc (bn_maxGon _2427438))) (λi::nat. bn_generatePolygon i v f _2427439))) (bn_minimalVertex _2427439 f))) (bn_minimalFace fs))) (bn_nonFinals _2427439))

thm Tame_classification.bn_next_plane:

∀ (p::nat) g::bn_graph. bn_next_plane p g = LET (λfs::(nat list × bool) list. LET_END (if fs = [] then [] else LET (λf::nat list × bool. LET_END (LET (λv::nat. LET_END (c_union (UPT (3::nat) (Suc (bn_maxGon p))) (λi::nat. bn_generatePolygon i v f g))) (bn_minimalVertex g f))) (bn_minimalFace fs))) (bn_nonFinals g)

thm DEF_bn_planeGraphsP:

bn_planeGraphsP = (λ_2427450::nat. GSPEC (λGEN%PVAR%280::bn_graph. ∃ g::bn_graph. SETSPEC GEN%PVAR%280 (bn_RTranCl (bn_next_plane _2427450) (bn_Seed _2427450, g) ∧ bn_finalGraph g) g))

thm Tame_classification.bn_PlaneGraphsP:

∀ p::nat. bn_planeGraphsP p = GSPEC (λGEN%PVAR%280::bn_graph. ∃ g::bn_graph. SETSPEC GEN%PVAR%280 (bn_RTranCl (bn_next_plane p) (bn_Seed p, g) ∧ bn_finalGraph g) g)

thm Tame_classification.bn_PlaneGraphs:

bn_PlaneGraphs = UNIONS (IMAGE bn_planeGraphsP HOL_Light_Import.UNIV)

thm Tame_classification.bn_squanderTarget:

bn_squanderTarget = (15410::nat)

thm Tame_classification.bn_excessTCount:

bn_excessTCount = (6300::nat)

thm DEF_bn_squanderVertex:

bn_squanderVertex = (λ(_2427455::nat) _2427456::nat. if _2427455 = (0::nat) ∧ _2427456 = (3::nat) then 6180::nat else if _2427455 = (0::nat) ∧ _2427456 = (4::nat) then 9700::nat else if _2427455 = (1::nat) ∧ _2427456 = (2::nat) then 6560::nat else if _2427455 = (1::nat) ∧ _2427456 = (3::nat) then 6180::nat else if _2427455 = (2::nat) ∧ _2427456 = (1::nat) then 7970::nat else if _2427455 = (2::nat) ∧ _2427456 = (2::nat) then 4120::nat else if _2427455 = (2::nat) ∧ _2427456 = (3::nat) then 12851::nat else if _2427455 = (3::nat) ∧ _2427456 = (1::nat) then 3110::nat else if _2427455 = (3::nat) ∧ _2427456 = (2::nat) then 8170::nat else if _2427455 = (4::nat) ∧ _2427456 = (0::nat) then 3470::nat else if _2427455 = (4::nat) ∧ _2427456 = (1::nat) then 3660::nat else if _2427455 = (5::nat) ∧ _2427456 = (0::nat) then 400::nat else if _2427455 = (5::nat) ∧ _2427456 = (1::nat) then 11360::nat else if _2427455 = (6::nat) ∧ _2427456 = (0::nat) then 6860::nat else if _2427455 = (7::nat) ∧ _2427456 = (0::nat) then 14500::nat else bn_squanderTarget)

thm Dont_repeat_yourself.table_b_bn:

∀ (p::nat) q::nat. bn_squanderVertex p q = (if p = (0::nat) ∧ q = (3::nat) then 6180::nat else if p = (0::nat) ∧ q = (4::nat) then 9700::nat else if p = (1::nat) ∧ q = (2::nat) then 6560::nat else if p = (1::nat) ∧ q = (3::nat) then 6180::nat else if p = (2::nat) ∧ q = (1::nat) then 7970::nat else if p = (2::nat) ∧ q = (2::nat) then 4120::nat else if p = (2::nat) ∧ q = (3::nat) then 12851::nat else if p = (3::nat) ∧ q = (1::nat) then 3110::nat else if p = (3::nat) ∧ q = (2::nat) then 8170::nat else if p = (4::nat) ∧ q = (0::nat) then 3470::nat else if p = (4::nat) ∧ q = (1::nat) then 3660::nat else if p = (5::nat) ∧ q = (0::nat) then 400::nat else if p = (5::nat) ∧ q = (1::nat) then 11360::nat else if p = (6::nat) ∧ q = (0::nat) then 6860::nat else if p = (7::nat) ∧ q = (0::nat) then 14500::nat else bn_squanderTarget)

thm DEF_bn_squanderFace:

bn_squanderFace = (λ_2427467::nat. if _2427467 = (3::nat) then 0::nat else if _2427467 = (4::nat) then 2060::nat else if _2427467 = (5::nat) then 4819::nat else if _2427467 = (6::nat) then 7120::nat else bn_squanderTarget)

thm Tame_classification.bn_squanderFace:

∀ n::nat. bn_squanderFace n = (if n = (3::nat) then 0::nat else if n = (4::nat) then 2060::nat else if n = (5::nat) then 4819::nat else if n = (6::nat) then 7120::nat else bn_squanderTarget)

thm DEF_bn_separated2:

$bn\_separated2 = (\lambda(\_2427472::bn\_graph)\ \_2427473::nat \Rightarrow bool.\ \forall\, v::nat.\ \_2427473\ v \longrightarrow (\forall\, f::nat\ list \times bool.\ MEM\, f\ (bn\_facesAt\ \_2427472\ v) \longrightarrow \neg\ \_2427473\ (bn\_nextVertex\, f\, v)))$

thm Tame_classification.bn_separated2:

$\forall\,(g::bn\_graph)\ V::nat \Rightarrow bool.\ bn\_separated2\ g\ V = (\forall\, v::nat.\ V\, v \longrightarrow (\forall\, f::nat\ list \times bool.\ MEM\, f\ (bn\_facesAt\ g\ v) \longrightarrow \neg\ V\ (bn\_nextVertex\, f\, v)))$

thm DEF_bn_separated3:

$bn\_separated3 = (\lambda(\_2427484::bn\_graph)\ \_2427485::nat \Rightarrow bool.\ \forall\, v::nat.\ \_2427485\ v \longrightarrow (\forall\, f::nat\ list \times bool.\ MEM\, f\ (bn\_facesAt\ \_2427484\ v) \longrightarrow length\ (bn\_vertices\_face\ f) \leq (4::nat) \longrightarrow HOL\_Light\_Import.INTER\ (bn\_vertices\_set\, f)\ \_2427485 = INSERT\ v\ EMPTY))$

thm Tame_classification.bn_separated3:

$\forall\,(g::bn\_graph)\ V::nat \Rightarrow bool.\ bn\_separated3\ g\ V = (\forall\, v::nat.\ V\, v \longrightarrow (\forall\, f::nat\ list \times bool.\ MEM\, f\ (bn\_facesAt\ g\ v) \longrightarrow length\ (bn\_vertices\_face\, f) \leq (4::nat) \longrightarrow HOL\_Light\_Import.INTER\ (bn\_vertices\_set\, f)\ V = INSERT\ v\ EMPTY))$

thm DEF_bn_separated:

$bn\_separated = (\lambda(\_2427496::bn\_graph)\ \_2427497::nat \Rightarrow bool.\ bn\_separated2\ \_2427496\ \_2427497 \wedge bn\_separated3\ \_2427496\ \_2427497)$

thm Tame_classification.bn_separated:

$\forall\,(g::bn\_graph)\ V::nat \Rightarrow bool.\ bn\_separated\ g\ V = (bn\_separated2\ g\ V \wedge bn\_separated3\ g\ V)$

thm DEF_bn_admissible1:

$bn\_admissible1 = (\lambda(\_2427508::nat\ list \times bool \Rightarrow nat)\ \_2427509::bn\_graph.\ \forall\, f::nat\ list \times bool.\ bn\_Faces\ \_2427509\, f \longrightarrow bn\_squanderFace\ (length\ (bn\_vertices\_face\ f)) \leq \_2427508\ f)$

thm Tame_classification.bn_admissible1:

$\forall\,(g::bn\_graph)\ w::nat\ list \times bool \Rightarrow nat.\ bn\_admissible1\ w\ g = (\forall\, f::nat\ list \times bool.\ bn\_Faces\ g\, f \longrightarrow bn\_squanderFace\ (length\ (bn\_vertices\_face\, f)) \leq w\, f)$

thm DEF_LIST_SUM:

$LIST\_SUM = (\lambda(\_2427520::?'a::type\ list)\ \_2427521::?'a::type \Rightarrow nat.\ foldr\ (\lambda x::?'a::type.\ op + (\_2427521\ x))\ \_2427520\ (0::nat))$

thm Tame_classification.LIST_SUM:

$\forall\,(f::?'a::type \Rightarrow nat)\ xs::?'a::type\ list.\ LIST\_SUM\ xs\, f = foldr\ (\lambda x::?'a::type.\ op + (f\, x))\ xs\ (0::nat)$

thm DEF_bn_admissible2:

$bn\_admissible2 = (\lambda(\_2427532::nat\ list \times bool \Rightarrow nat)\ \_2427533::bn\_graph.\ \forall v::nat.\ bn\_vertices\_graph\ \_2427533\ v \longrightarrow bn\_except\ \_2427533\ v = (0::nat) \longrightarrow bn\_squanderVertex\ (bn\_tri\ \_2427533\ v)\ (bn\_quad\ \_2427533\ v) \leq LIST\_SUM\ (bn\_facesAt\ \_2427533\ v)\ \_2427532)$

thm Tame_classification.bn_admissible2:

$\forall (g::bn\_graph)\ w::nat\ list \times bool \Rightarrow nat.\ bn\_admissible2\ w\ g = (\forall v::nat.\ bn\_vertices\_graph\ g\ v \longrightarrow bn\_except\ g\ v = (0::nat) \longrightarrow bn\_squanderVertex\ (bn\_tri\ g\ v)\ (bn\_quad\ g\ v) \leq LIST\_SUM\ (bn\_facesAt\ g\ v)\ w)$

thm DEF_bn_admissible3:

$bn\_admissible3 = (\lambda(\_2427544::nat\ list \times bool \Rightarrow nat)\ \_2427545::bn\_graph.\ \forall v::nat.\ bn\_vertices\_graph\ \_2427545\ v \longrightarrow bn\_vertextype\ \_2427545\ v = (5::nat, 0::nat, 1::nat) \longrightarrow bn\_excessTCount \leq LIST\_SUM\ (filter\ bn\_triangle\ (bn\_facesAt\ \_2427545\ v))\ \_2427544)$

thm Tame_classification.bn_admissible3:

$\forall (g::bn\_graph)\ w::nat\ list \times bool \Rightarrow nat.\ bn\_admissible3\ w\ g = (\forall v::nat.\ bn\_vertices\_graph\ g\ v \longrightarrow bn\_vertextype\ g\ v = (5::nat, 0::nat, 1::nat) \longrightarrow bn\_excessTCount \leq LIST\_SUM\ (filter\ bn\_triangle\ (bn\_facesAt\ g\ v))\ w)$

thm DEF_bn_admissible:

$bn\_admissible = (\lambda(\_2427556::nat\ list \times bool \Rightarrow nat)\ \_2427557::bn\_graph.\ bn\_admissible1\ \_2427556\ \_2427557 \wedge bn\_admissible2\ \_2427556\ \_2427557 \wedge bn\_admissible3\ \_2427556\ \_2427557)$

thm Tame_classification.bn_admissible:

$\forall (w::nat\ list \times bool \Rightarrow nat)\ g::bn\_graph.\ bn\_admissible\ w\ g = (bn\_admissible1\ w\ g \wedge bn\_admissible2\ w\ g \wedge bn\_admissible3\ w\ g)$

thm DEF_bn_tame9a:

$bn\_tame9a = (\lambda\_2427568::bn\_graph.\ \forall f::nat\ list \times bool.\ bn\_Faces\ \_2427568\ f \longrightarrow (3::nat) \leq length\ (bn\_vertices\_face\ f) \wedge length\ (bn\_vertices\_face\ f) \leq (6::nat))$

thm Tame_classification.bn_tame9a:

$\forall g::bn\_graph.\ bn\_tame9a\ g = (\forall f::nat\ list \times bool.\ bn\_Faces\ g\ f \longrightarrow (3::nat) \leq length\ (bn\_vertices\_face\ f) \wedge length\ (bn\_vertices\_face\ f) \leq (6::nat))$

thm DEF_bn_tame10:

$bn\_tame10 = (\lambda\_2427573::bn\_graph.\ LET\ (\lambda n::nat.\ LET\_END\ ((13::nat) \leq n \wedge n \leq (15::nat)))\ (bn\_countVertices\ \_2427573))$

thm Tame_classification.bn_tame10:

$\forall g::bn\_graph.\ bn\_tame10\ g = LET\ (\lambda n::nat.\ LET\_END\ ((13::nat) \leq n \wedge n \leq (15::nat)))\ (bn\_countVertices\ g)$

thm DEF_bn_tame11a:

bn_tame11a = (λ_2427578::bn_graph. ∀ v::nat. bn_vertices_graph _2427578 v ⟶ (3::nat) ≤ bn_degree _2427578 v)

thm Tame_classification.bn_tame11a:

∀ g::bn_graph. bn_tame11a g = (∀ v::nat. bn_vertices_graph g v ⟶ (3::nat) ≤ bn_degree g v)

thm DEF_bn_tame11b:

bn_tame11b = (λ_2427583::bn_graph. ∀ v::nat. bn_vertices_graph _2427583 v ⟶ bn_degree _2427583 v ≤ (if bn_except _2427583 v = (0::nat) then 7::nat else (6::nat)))

thm Tame_classification.bn_tame11b:

∀ g::bn_graph. bn_tame11b g = (∀ v::nat. bn_vertices_graph g v ⟶ bn_degree g v ≤ (if bn_except g v = (0::nat) then 7::nat else (6::nat)))

thm DEF_bn_tame12o:

bn_tame12o = (λ_2427588::bn_graph. ∀ v::nat. bn_vertices_graph _2427588 v ⟶ bn_except _2427588 v ≠ (0::nat) ∧ bn_degree _2427588 v = (6::nat) ⟶ bn_vertextype _2427588 v = (5::nat, 0::nat, 1::nat))

thm Tame_classification.bn_tame12o:

∀ g::bn_graph. bn_tame12o g = (∀ v::nat. bn_vertices_graph g v ⟶ bn_except g v ≠ (0::nat) ∧ bn_degree g v = (6::nat) ⟶ bn_vertextype g v = (5::nat, 0::nat, 1::nat))

thm DEF_bn_tame13a:

bn_tame13a = (λ_2427593::bn_graph. ∃ w::nat list × bool ⇒ nat. bn_admissible w _2427593 ∧ LIST_SUM (bn_faces _2427593) w < bn_squanderTarget)

thm Tame_classification.bn_tame13a:

∀ g::bn_graph. bn_tame13a g = (∃ w::nat list × bool ⇒ nat. bn_admissible w g ∧ LIST_SUM (bn_faces g) w < bn_squanderTarget)

thm DEF_bn_tame:

bn_tame = (λ_2427598::bn_graph. bn_tame9a _2427598 ∧ bn_tame10 _2427598 ∧ bn_tame11a _2427598 ∧ bn_tame11b _2427598 ∧ bn_tame12o _2427598 ∧ bn_tame13a _2427598)

thm Tame_classification.bn_tame:

∀ g::bn_graph. bn_tame g = (bn_tame9a g ∧ bn_tame10 g ∧ bn_tame11a g ∧ bn_tame11b g ∧ bn_tame12o g ∧ bn_tame13a g)

thm DEF_bn_fgraph:

bn_fgraph = (λ_2427603::bn_graph. map bn_vertices_face (bn_faces _2427603))

thm Tame_classification.bn_fgraph:

$\forall g::bn\_graph.\ bn\_fgraph\ g = map\ bn\_vertices\_face\ (bn\_faces\ g)$

thm DEF_graph:

$graph = (\lambda\_2427608::(?'a::type \Rightarrow bool) \Rightarrow bool.\ \forall e::?'a::type \Rightarrow bool.\ \_2427608\ e \longrightarrow HAS\_SIZE\ e\ (2::nat))$

thm Fan_defs.graph:

$\forall E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ graph\ E = (\forall e::?'a::type \Rightarrow bool.\ E\ e \longrightarrow HAS\_SIZE\ e\ (2::nat))$

thm DEF_fan1:

$fan1 = (\lambda\_2427613::?'c::type \times (?'b::type \Rightarrow bool) \times ?'a::type.\ FINITE\ (fst\ (snd\ \_2427613)) \wedge \neg\ SUBSET\ (fst\ (snd\ \_2427613))\ EMPTY)$

thm Fan_defs.fan1:

$\forall (x::?'c::type)\ (E::?'b::type)\ V::?'a::type \Rightarrow bool.\ fan1\ (x,\ V,\ E) = (FINITE\ V \wedge \neg\ SUBSET\ V\ EMPTY)$

thm DEF_fan2:

$fan2 = (\lambda\_2427626::?'b::type \times (?'b::type \Rightarrow bool) \times ?'a::type.\ \neg\ IN\ (fst\ \_2427626)\ (fst\ (snd\ \_2427626)))$

thm Fan_defs.fan2:

$\forall (E::?'b::type)\ (x::?'a::type)\ V::?'a::type \Rightarrow bool.\ fan2\ (x,\ V,\ E) = (\neg\ IN\ x\ V)$

thm DEF_fan6:

$fan6 = (\lambda\_2427639::(real,\ ?'b::type)\ cart \times ?'a::type \times (((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall e::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ e\ (snd\ (snd\ \_2427639)) \longrightarrow \neg\ collinear\ (HOL\_Light\_Import.UNION\ (INSERT\ (fst\ \_2427639)\ EMPTY)\ e))$

thm Fan_defs.fan6:

$\forall (V::?'b::type)\ (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ ?'a::type)\ cart.\ fan6\ (x,\ V,\ E) = (\forall e::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ E \longrightarrow \neg\ collinear\ (HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ e))$

thm DEF_fan7:

$fan7 = (\lambda\_2427652::(real,\ ?'a::type)\ cart \times ((real,\ ?'a::type)\ cart \Rightarrow bool) \times (((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall (e1::(real,\ ?'a::type)\ cart \Rightarrow bool)\ e2::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ e1\ (HOL\_Light\_Import.UNION\ (snd\ (snd\ \_2427652))\ (GSPEC\ (\lambda GEN\%PVAR\%281::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%281\ (IN\ v\ (fst\ (snd\ \_2427652)))\ (INSERT\ v\ EMPTY)))) \wedge IN\ e2\ (HOL\_Light\_Import.UNION\ (snd\ (snd\ \_2427652))\ (GSPEC\ (\lambda GEN\%PVAR\%282::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists v::(real,\ ?'a::type)$

cart. SETSPEC GEN%PVAR%282 (IN v (fst (snd _2427652))) (INSERT v EMPTY)))) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT (fst _2427652) EMPTY) e1) (aff_ge (INSERT (fst _2427652) EMPTY) e2) = aff_ge (INSERT (fst _2427652) EMPTY) (HOL_Light_Import.INTER e1 e2))

thm Fan_defs.fan7:

∀(E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) (V::(real, ?'a::type) cart ⇒ bool) x::(real, ?'a::type) cart. fan7 (x, V, E) = (∀(e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%281::(real, ?'a::type) cart ⇒ bool. ∃v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%281 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%282::(real, ?'a::type) cart ⇒ bool. ∃v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%282 (IN v V) (INSERT v EMPTY)))) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER e1 e2))

thm DEF_FAN:

FAN = (λ_2427665::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) × (((real, ?'a::type) cart ⇒ bool) ⇒ bool). SUBSET (UNIONS (snd (snd _2427665))) (fst (snd _2427665)) ∧ graph (snd (snd _2427665)) ∧ fan1 (fst _2427665, fst (snd _2427665), snd (snd _2427665)) ∧ fan2 (fst _2427665, fst (snd _2427665), snd (snd _2427665)) ∧ fan6 (fst _2427665, fst (snd _2427665), snd (snd _2427665)) ∧ fan7 (fst _2427665, fst (snd _2427665), snd (snd _2427665)))

thm Fan_defs.FAN:

∀(x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FAN (x, V, E) = (SUBSET (UNIONS E) V ∧ graph E ∧ fan1 (x, V, E) ∧ fan2 (x, V, E) ∧ fan6 (x, V, E) ∧ fan7 (x, V, E))

thm DEF_set_of_edge:

set_of_edge = (λ(_2427678::?'a::type) (_2427679::?'a::type ⇒ bool) _2427680::(?'a::type ⇒ bool) ⇒ bool. GSPEC (λGEN%PVAR%283::?'a::type. ∃w::?'a::type. SETSPEC GEN%PVAR%283 (IN (INSERT _2427678 (INSERT w EMPTY)) _2427680 ∧ IN w _2427679) w))

thm Fan_defs.set_of_edge:

∀(v::?'a::type) (E::(?'a::type ⇒ bool) ⇒ bool) V::?'a::type ⇒ bool. set_of_edge v V E = GSPEC (λGEN%PVAR%283::?'a::type. ∃w::?'a::type. SETSPEC GEN%PVAR%283 (IN (INSERT v (INSERT w EMPTY)) E ∧ IN w V) w)

thm DEF_sigma_fan:

sigma_fan = (λ(_2427699::(real, 3) cart) (_2427700::(real, 3) cart ⇒ bool) (_2427701::((real, 3) cart ⇒ bool) ⇒ bool) (_2427702::(real, 3) cart) _2427703::(real, 3) cart. if set_of_edge _2427702 _2427700 _2427701 = INSERT _2427703

EMPTY then _2427703 else SOME w::(real, 3) cart. IN w (set_of_edge _2427702 _2427700 _2427701) ∧ w ≠ _2427703 ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge _2427702 _2427700 _2427701) ∧ w1 ≠ _2427703 ⟶ azim _2427699 _2427702 _2427703 w ≤ azim _2427699 _2427702 _2427703 w1))

thm Fan.sigma_fan:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. sigma_fan x V E v u = (if set_of_edge v V E = INSERT u EMPTY then u else SOME w::(real, 3) cart. IN w (set_of_edge v V E) ∧ w ≠ u ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge v V E) ∧ w1 ≠ u ⟶ azim x v u w ≤ azim x v u w1))

thm DEF_extension_sigma_fan:

extension_sigma_fan = (λ(_2427744::(real, 3) cart) (_2427745::(real, 3) cart ⇒ bool) (_2427746::((real, 3) cart ⇒ bool) ⇒ bool) (_2427747::(real, 3) cart) _2427748::(real, 3) cart. if ¬ IN _2427748 (set_of_edge _2427747 _2427745 _2427746) then _2427748 else sigma_fan _2427744 _2427745 _2427746 _2427747 _2427748)

thm Fan.extension_sigma_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. extension_sigma_fan x V E v u = (if ¬ IN u (set_of_edge v V E) then u else sigma_fan x V E v u)

thm DEF_inverse_sigma_fan:

inverse_sigma_fan = (λ(_2427789::(real, 3) cart) (_2427790::(real, 3) cart ⇒ bool) (_2427791::((real, 3) cart ⇒ bool) ⇒ bool) _2427792::(real, 3) cart. HOL_Light_Import.inverse (extension_sigma_fan _2427789 _2427790 _2427791 _2427792))

thm Fan_defs.inverse_sigma_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) v::(real, 3) cart. inverse_sigma_fan x V E v = HOL_Light_Import.inverse (extension_sigma_fan x V E v)

thm DEF_dart1_of_fan:

dart1_of_fan = (λ_2427821::(?'a::type ⇒ bool) × ((?'a::type ⇒ bool) ⇒ bool). GSPEC (λGEN%PVAR%284::?'a::type × ?'a::type. ∃ (v::?'a::type) w::?'a::type. SETSPEC GEN%PVAR%284 (IN (INSERT v (INSERT w EMPTY)) (snd _2427821)) (v, w)))

thm Fan_defs.dart1_of_fan:

∀ (V::?'a::type ⇒ bool) E::(?'a::type ⇒ bool) ⇒ bool. dart1_of_fan (V, E) = GSPEC (λGEN%PVAR%284::?'a::type × ?'a::type. ∃ (v::?'a::type) w::?'a::type. SETSPEC GEN%PVAR%284 (IN (INSERT v (INSERT w EMPTY)) E) (v, w))

thm DEF_dart_of_fan:

dart_of_fan = (λ*_2427830*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*). *HOL_Light_Import.UNION* (*GSPEC* (λ*GEN%PVAR%285*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%285* (*IN v* (*fst _2427830*) ∧ *set_of_edge v* (*fst _2427830*) (*snd _2427830*) = *EMPTY*) (*v*, *v*))) (*GSPEC* (λ*GEN%PVAR%286*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%286* (*IN* (*INSERT v* (*INSERT w EMPTY*)) (*snd _2427830*)) (*v*, *w*))))

thm Fan_defs.dart_of_fan:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *dart_of_fan* (*V*, *E*) = *HOL_Light_Import.UNION* (*GSPEC* (λ*GEN%PVAR%285*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%285* (*IN v V* ∧ *set_of_edge v V E* = *EMPTY*) (*v*, *v*))) (*GSPEC* (λ*GEN%PVAR%286*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%286* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*v*, *w*)))

thm DEF_i_fan:

i_fan = (λ(*_2427839*::(*real*, *3*) *cart*) (*_2427840*::(*real*, *3*) *cart* ⇒ *bool*) *_2427841*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *v*, *w*, *sigma_fan x _2427840 _2427841 v w*)))

thm Fan_defs.i_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *i_fan x V E* = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *v*, *w*, *sigma_fan x V E v w*))

thm DEF_extended_dart:

extended_dart = (λ(*_2427860*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) *_2427861*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *i_fan* (*vec* (*0*::*nat*)) (*fst _2427860*) (*snd _2427860*) (*vec* (*0*::*nat*), *fst _2427861*, *snd _2427861*, *snd _2427861*))

thm Fan_defs.extended_dart:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *extended_dart* (*V*, *E*) (*v*, *w*) = *i_fan* (*vec* (*0*::*nat*)) *V E* (*vec* (*0*::*nat*), *v*, *w*, *w*)

thm DEF_contracted_dart:

contracted_dart = (λ*_2427882*::*?'d*::*type* × *?'c*::*type* × *?'b*::*type* × *?'a*::*type*. (*fst* (*snd _2427882*), *fst* (*snd* (*snd _2427882*))))

thm Tame_defs.contracted_dart:

$\forall (x::?'d::type)\ (w1::?'c::type)\ (v::?'b::type)\ w::?'a::type.\ contracted\_dart\ (x,\ v,\ w,\ w1) = (v,\ w)$

thm DEF_e_fan_pair:

$e\_fan\_pair = (\lambda(\_2427899::?'d::type \times ?'c::type)\ \_2427900::?'b::type \times ?'a::type.\ (snd\ \_2427900,\ fst\ \_2427900))$

thm Fan_defs.e_fan_pair:

$\forall (V::?'d::type)\ (E::?'c::type)\ (w::?'b::type)\ v::?'a::type.\ e\_fan\_pair\ (V,\ E)\ (v,\ w) = (w,\ v)$

thm DEF_n_fan_pair:

$n\_fan\_pair = (\lambda(\_2427921::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool))\ \_2427922::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ (fst\ \_2427922,\ sigma\_fan\ (vec\ (0::nat))\ (fst\ \_2427921)\ (snd\ \_2427921)\ (fst\ \_2427922)\ (snd\ \_2427922)))$

thm Tame_defs.n_fan_pair:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ n\_fan\_pair\ (V,\ E)\ (v,\ w) = (v,\ sigma\_fan\ (vec\ (0::nat))\ V\ E\ v\ w)$

thm DEF_f_fan_pair:

$f\_fan\_pair = (\lambda(\_2427943::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool))\ \_2427944::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ (snd\ \_2427944,\ inverse\_sigma\_fan\ (vec\ (0::nat))\ (fst\ \_2427943)\ (snd\ \_2427943)\ (snd\ \_2427944)\ (fst\ \_2427944)))$

thm Fan_defs.f_fan_pair:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (w::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ f\_fan\_pair\ (V,\ E)\ (v,\ w) = (w,\ inverse\_sigma\_fan\ (vec\ (0::nat))\ V\ E\ w\ v)$

thm DEF_hypermap_of_fan:

$hypermap\_of\_fan = (\lambda\_2427965::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ LET\ (\lambda p::(((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart.\ LET\_END\ (hypermap\ (dart\_of\_fan\ (fst\ \_2427965,\ snd\ \_2427965),\ p\ e\_fan\_pair,\ p\ n\_fan\_pair,\ p\ f\_fan\_pair)))\ (\lambda t::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ (fst\ \_2427965,\ snd\ \_2427965))\ (dart1\_of\_fan\ (fst\ \_2427965,\ snd\ \_2427965))))$

thm Fan_defs.hypermap_of_fan:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ hypermap\_of\_fan\ (V,\ E) = LET\ (\lambda p::(((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)$

⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart*) ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart*. *LET_END* (*hypermap* (*dart_of_fan* (*V*, *E*), *p e_fan_pair*, *p n_fan_pair*, *p f_fan_pair*))) (λ*t*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart*. *res* (*t* (*V*, *E*)) (*dart1_of_fan* (*V*, *E*)))

thm DEF_e_fan_pair_ext:

e_fan_pair_ext = (λ(*_2427974*::(*?'a*::*type* ⇒ *bool*) × ((*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) *_2427975*::*?'a*::*type* × *?'a*::*type*. *if IN _2427975* (*dart1_of_fan* (*fst _2427974*, *snd _2427974*)) *then e_fan_pair* (*fst _2427974*, *snd _2427974*) *_2427975 else _2427975*)

thm Fan_defs.e_fan_pair_ext:

∀ (*V*::*?'a*::*type* ⇒ *bool*) (*E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) *x*::*?'a*::*type* × *?'a*::*type*. *e_fan_pair_ext* (*V*, *E*) *x* = (*if IN x* (*dart1_of_fan* (*V*, *E*)) *then e_fan_pair* (*V*, *E*) *x else x*)

thm DEF_n_fan_pair_ext:

n_fan_pair_ext = (λ(*_2427991*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) *_2427992*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *if IN _2427992* (*dart1_of_fan* (*fst _2427991*, *snd _2427991*)) *then n_fan_pair* (*fst _2427991*, *snd _2427991*) *_2427992 else _2427992*)

thm Fan_defs.n_fan_pair_ext:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *n_fan_pair_ext* (*V*, *E*) *x* = (*if IN x* (*dart1_of_fan* (*V*, *E*)) *then n_fan_pair* (*V*, *E*) *x else x*)

thm DEF_f_fan_pair_ext:

f_fan_pair_ext = (λ(*_2428008*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) *_2428009*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *if IN _2428009* (*dart1_of_fan* (*fst _2428008*, *snd _2428008*)) *then f_fan_pair* (*fst _2428008*, *snd _2428008*) *_2428009 else _2428009*)

thm Fan_defs.f_fan_pair_ext:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *f_fan_pair_ext* (*V*, *E*) *x* = (*if IN x* (*dart1_of_fan* (*V*, *E*)) *then f_fan_pair* (*V*, *E*) *x else x*)

thm Fan_defs.E_FAN_PAIR_EXT:

∀ (*V*::*?'a*::*type* ⇒ *bool*) *E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *e_fan_pair_ext* (*V*, *E*) = *res* (*e_fan_pair* (*V*, *E*)) (*dart1_of_fan* (*V*, *E*))

thm Fan_defs.F_FAN_PAIR_EXT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *f_fan_pair_ext* (*V*, *E*) = *res* (*f_fan_pair* (*V*, *E*)) (*dart1_of_fan* (*V*, *E*))

thm Fan_defs.N_FAN_PAIR_EXT:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *n_fan_pair_ext* (*V*, *E*) = *res* (*n_fan_pair* (*V*, *E*)) (*dart1_of_fan* (*V*, *E*))

thm Fan_defs.HYPERMAP_OF_FAN_ALT:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *hypermap_of_fan* (*V*, *E*) = *hypermap* (*dart_of_fan* (*V*, *E*), *e_fan_pair_ext* (*V*, *E*), *n_fan_pair_ext* (*V*, *E*), *f_fan_pair_ext* (*V*, *E*))

thm DEF_xfan:

xfan = (λ*_2428025*::(*real*, *?'b*::*type*) *cart* $\times$ *?'a*::*type* $\times$ (((*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*). *GSPEC* (λ*GEN%PVAR%287*::(*real*, *?'b*::*type*) *cart*. $\exists$ *v*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%287* ($\exists$ *e*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*. *snd* (*snd _2428025*) *e* $\wedge$ *IN v* (*aff_ge* (*INSERT* (*fst _2428025*) *EMPTY*) *e*)) *v*))

thm Fan_defs.xfan:

$\forall$ (*V*::*?'b*::*type*) (*E*::((*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *x*::(*real*, *?'a*::*type*) *cart*. *xfan* (*x*, *V*, *E*) = *GSPEC* (λ*GEN%PVAR%287*::(*real*, *?'a*::*type*) *cart*. $\exists$ *v*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%287* ($\exists$ *e*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *E e* $\wedge$ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)

thm DEF_yfan:

yfan = (λ*_2428038*::(*real*, *3*) *cart* $\times$ *?'a*::*type* $\times$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*). *DIFF HOL_Light_Import.UNIV* (*xfan* (*fst _2428038*, *fst* (*snd _2428038*), *snd* (*snd _2428038*))))

thm Fan_defs.yfan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::*?'a*::*type*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *yfan* (*x*, *V*, *E*) = *DIFF HOL_Light_Import.UNIV* (*xfan* (*x*, *V*, *E*))

thm DEF_w_dart_fan:

w_dart_fan = (λ(*_2428051*::(*real*, *3*) *cart*) (*_2428052*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*_2428053*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *_2428054*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *if* (*1*::*nat*) < *CARD* (*set_of_edge* (*fst* (*snd _2428054*)) *_2428052 _2428053*) *then wedge _2428051* (*fst* (*snd _2428054*)) (*fst* (*snd* (*snd _2428054*))) (*sigma_fan _2428051 _2428052 _2428053* (*fst* (*snd _2428054*)) (*fst* (*snd* (*snd _2428054*)))) *else if set_of_edge* (*fst* (*snd _2428054*)) *_2428052 _2428053* = *INSERT* (*fst* (*snd* (*snd _2428054*))) *EMPTY then DIFF HOL_Light_Import.UNIV* (*aff_ge* (*INSERT _2428051* (*INSERT* (*fst* (*snd _2428054*)) *EMPTY*)) (*INSERT* (*fst* (*snd* (*snd _2428054*))) *EMPTY*)) *else if set_of_edge* (*fst* (*snd _2428054*)) *_2428052 _2428053* = *EMPTY then DIFF HOL_Light_Import.UNIV* (*aff* (*INSERT _2428051* (*INSERT* (*fst* (*snd _2428054*)) *EMPTY*))) *else EMPTY*)

thm Fan_defs.w_dart_fan:

∀(y::(real, 3) cart) (w1::(real, 3) cart) (w::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (x::(real, 3) cart) v::(real, 3) cart. w_dart_fan x V E (y, v, w, w1) = (if (1::nat) < CARD (set_of_edge v V E) then wedge x v w (sigma_fan x V E v w) else if set_of_edge v V E = INSERT w EMPTY then DIFF HOL_Light_Import.UNIV (aff_ge (INSERT x (INSERT v EMPTY)) (INSERT w EMPTY)) else if set_of_edge v V E = EMPTY then DIFF HOL_Light_Import.UNIV (aff (INSERT x (INSERT v EMPTY))) else EMPTY)

thm DEF_azim_fan:

azim_fan = (λ(_2428104::(real, 3) cart) (_2428105::(real, 3) cart ⇒ bool) (_2428106::((real, 3) cart ⇒ bool) ⇒ bool) (_2428107::(real, 3) cart) _2428108::(real, 3) cart. if (1::nat) < CARD (set_of_edge _2428107 _2428105 _2428106) then azim _2428104 _2428107 _2428108 (sigma_fan _2428104 _2428105 _2428106 _2428107 _2428108) else real_of_nat (2::nat) ∗ pi)

thm Topology.azim_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. azim_fan x V E v w = (if (1::nat) < CARD (set_of_edge v V E) then azim x v w (sigma_fan x V E v w) else real_of_nat (2::nat) ∗ pi)

thm DEF_azim_dart:

azim_dart = (λ(_2428149::((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool)) _2428150::(real, 3) cart × (real, 3) cart. if fst _2428150 = snd _2428150 then real_of_nat (2::nat) ∗ pi else azim_fan (vec (0::nat)) (fst _2428149) (snd _2428149) (fst _2428150) (snd _2428150))

thm Tame_defs.azim_dart:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. azim_dart (V, E) (v, w) = (if v = w then real_of_nat (2::nat) ∗ pi else azim_fan (vec (0::nat)) V E v w)

thm DEF_rcone_fan:

rcone_fan = (λ(_2428171::(real, 3) cart) (_2428172::(real, 3) cart) _2428173::real. GSPEC (λGEN%PVAR%288::(real, 3) cart. ∃y::(real, 3) cart. SETSPEC GEN%PVAR%288 (distance (y, _2428171) ∗ (distance (_2428172, _2428171) ∗ _2428173) < dot (vector_sub y _2428171) (vector_sub _2428172 _2428171)) y))

thm Fan_defs.rcone_fan:

∀(v::(real, 3) cart) (x::(real, 3) cart) h::real. rcone_fan x v h = GSPEC (λGEN%PVAR%288::(real, 3) cart. ∃y::(real, 3) cart. SETSPEC GEN%PVAR%288 (distance (y, x) ∗ (distance (v, x) ∗ h) < dot (vector_sub y x) (vector_sub v x)) y)

thm DEF_rw_dart_fan:

rw_dart_fan = (λ(_2428192::(real, 3) cart) (_2428193::(real, 3) cart ⇒ bool) (_2428194::((real, 3) cart ⇒ bool) ⇒ bool) (_2428195::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) _2428196::real. HOL_Light_Import.INTER (w_dart_fan _2428192 _2428193 _2428194 (fst _2428195, fst (snd _2428195), fst (snd (snd _2428195)), snd (snd (snd _2428195)))) (rcone_fan _2428192 (fst (snd _2428195)) _2428196))

thm Fan_defs.rw_dart_fan:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart) (w::(real, 3) cart) (w1::(real, 3) cart) (x::(real, 3) cart) (v::(real, 3) cart) h::real. rw_dart_fan x V E (y, v, w, w1) h = HOL_Light_Import.INTER (w_dart_fan x V E (y, v, w, w1)) (rcone_fan x v h)

thm DEF_topological_component_yfan:

topological_component_yfan = (λ_2428261::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). GSPEC (λGEN%PVAR%289::(real, 3) cart ⇒ bool. ∃ y::(real, 3) cart. SETSPEC GEN%PVAR%289 (IN y (yfan (fst _2428261, fst (snd _2428261), snd (snd _2428261)))) (connected_component (yfan (fst _2428261, fst (snd _2428261), snd (snd _2428261))) y)))

thm Fan_defs.topological_component_yfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. topological_component_yfan (x, V, E) = GSPEC (λGEN%PVAR%289::(real, 3) cart ⇒ bool. ∃ y::(real, 3) cart. SETSPEC GEN%PVAR%289 (IN y (yfan (x, V, E))) (connected_component (yfan (x, V, E)) y))

thm DEF_dart_leads_into1:

dart_leads_into1 = (λ(_2428274::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool)) _2428275::(real, 3) cart × (real, 3) cart. SOME s::(real, 3) cart ⇒ bool. IN s (topological_component_yfan (fst _2428274, fst (snd _2428274), snd (snd _2428274))) ∧ (∃ eps<1::real. SUBSET (rw_dart_fan (fst _2428274) (fst (snd _2428274)) (snd (snd _2428274)) (fst _2428274, fst _2428275, snd _2428275, sigma_fan (fst _2428274) (fst (snd _2428274)) (snd (snd _2428274)) (fst _2428275) (snd _2428275)) eps) s))

thm Fan_defs.dart_leads_into1:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. dart_leads_into1 (x, V, E) (v, u) = (SOME s::(real, 3) cart ⇒ bool. IN s (topological_component_yfan (x, V, E)) ∧ (∃ eps<1::real. SUBSET (rw_dart_fan x V E (x, v, u, sigma_fan x V E v u) eps) s))

thm DEF_dartset_leads_into:

dartset_leads_into = (λ(_2428301::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool)) _2428302::(real, 3) cart × (real, 3) cart ⇒ bool. SOME s::(real, 3) cart ⇒ bool. ∀ y::(real, 3) cart × (real, 3) cart. IN *y _2428302* $\longrightarrow$ *s* = *dart_leads_into1* (*fst _2428301*, *fst* (*snd _2428301*), *snd* (*snd _2428301*)) *y*)

thm Tame_defs.dartset_leads_into:

$\forall$ (*ds*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *dartset_leads_into* (*x*, *V*, *E*) *ds* = (*SOME s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\forall$ *y*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN y ds* $\longrightarrow$ *s* = *dart_leads_into1* (*x*, *V*, *E*) *y*)

thm DEF_surrounded_node:

surrounded_node = (λ(*_2428323*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\times$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) *_2428324*::(*real*, *3*) *cart*. $\forall$ *x*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN x* (*dart_of_fan* (*fst _2428323*, *snd _2428323*)) $\wedge$ *fst x* = *_2428324* $\longrightarrow$ *azim_dart* (*fst _2428323*, *snd _2428323*) *x* < *pi*)

thm Tame_defs.surrounded_node:

$\forall$ (*v*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *surrounded_node* (*V*, *E*) *v* = ($\forall$ *x*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN x* (*dart_of_fan* (*V*, *E*)) $\wedge$ *fst x* = *v* $\longrightarrow$ *azim_dart* (*V*, *E*) *x* < *pi*)

thm DEF_fully_surrounded:

fully_surrounded = (λ*_2428340*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\times$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*). $\forall$ *x*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN x* (*dart_of_fan* (*fst _2428340*, *snd _2428340*)) $\longrightarrow$ *azim_dart* (*fst _2428340*, *snd _2428340*) *x* < *pi*)

thm Fan_defs.fully_surrounded:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *fully_surrounded* (*V*, *E*) = ($\forall$ *x*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN x* (*dart_of_fan* (*V*, *E*)) $\longrightarrow$ *azim_dart* (*V*, *E*) *x* < *pi*)

thm DEF_conforming_bijection:

conforming_bijection = (λ*_2428349*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\times$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*). $\forall$ *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN s* (*topological_component_yfan* (*vec* (*0*::*nat*), *fst _2428349*, *snd _2428349*)) $\longrightarrow$ ($\exists$!*f*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*fst _2428349*, *snd _2428349*))) $\wedge$ *s* = *dartset_leads_into* (*vec* (*0*::*nat*), *fst _2428349*, *snd _2428349*) *f*))

thm Fan_defs.conforming_bijection:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *conforming_bijection* (*V*, *E*) = ($\forall$ *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN s* (*topological_component_yfan* (*vec* (*0*::*nat*), *V*, *E*)) $\longrightarrow$ ($\exists$!*f*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*V*, *E*))) $\wedge$ *s* = *dartset_leads_into* (*vec* (*0*::*nat*), *V*, *E*) *f*))

thm DEF_conforming_half_space:

conforming_half_space = (λ*_2428358*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\times$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*). $\forall$ *f*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*fst _2428358*, *snd _2428358*))) ⟶ *dartset_leads_into* (*vec* (*0*::*nat*), *fst _2428358*, *snd _2428358*) *f* = *INTERS* (*GSPEC* (λ*GEN%PVAR%290*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%290* (*IN x f*) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*fst x*) (*INSERT* (*fst* (*f_fan_pair* (*fst _2428358*, *snd _2428358*) *x*)) *EMPTY*))) (*INSERT* (*fst* (*HOL_Light_Import.inverse* (*f_fan_pair* (*fst _2428358*, *snd _2428358*)) *x*)) *EMPTY*)))))

thm Fan_defs.conforming_half_space:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *conforming_half_space* (*V*, *E*) = (∀ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*V*, *E*))) ⟶ *dartset_leads_into* (*vec* (*0*::*nat*), *V*, *E*) *f* = *INTERS* (*GSPEC* (λ*GEN%PVAR%290*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%290* (*IN x f*) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*fst x*) (*INSERT* (*fst* (*f_fan_pair* (*V*, *E*) *x*)) *EMPTY*))) (*INSERT* (*fst* (*HOL_Light_Import.inverse* (*f_fan_pair* (*V*, *E*)) *x*)) *EMPTY*)))))

thm DEF_conforming_solid_angle:

conforming_solid_angle = (λ*_2428367*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*). ∀ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*fst _2428367*, *snd _2428367*))) ⟶ *LET* (λ*U*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* ((∀ *r*::*real*. *measurable* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) *U*)) ∧ *eventually_radial* (*vec* (*0*::*nat*)) *U* ∧ *sol* (*vec* (*0*::*nat*)) *U* = *real_of_nat* (*2*::*nat*) ∗ *pi* + *sum f* (λ*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_dart* (*fst _2428367*, *snd _2428367*) *x* − *pi*))) (*dartset_leads_into* (*vec* (*0*::*nat*), *fst _2428367*, *snd _2428367*) *f*))

thm Fan_defs.conforming_solid_angle:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *conforming_solid_angle* (*V*, *E*) = (∀ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*face_set* (*hypermap_of_fan* (*V*, *E*))) ⟶ *LET* (λ*U*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* ((∀ *r*::*real*. *measurable* (*HOL_Light_Import.INTER* (*ball* (*vec* (*0*::*nat*), *r*)) *U*)) ∧ *eventually_radial* (*vec* (*0*::*nat*)) *U* ∧ *sol* (*vec* (*0*::*nat*)) *U* = *real_of_nat* (*2*::*nat*) ∗ *pi* + *sum f* (λ*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_dart* (*V*, *E*) *x* − *pi*))) (*dartset_leads_into* (*vec* (*0*::*nat*), *V*, *E*) *f*))

thm DEF_conforming_diagonal:

conforming_diagonal = (λ*_2428376*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*). ∀ (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN f* (*face_set* (*hypermap_of_fan* (*fst _2428376*, *snd _2428376*))) ∧ *IN x f* ∧ *IN y f* ∧ *x* ≠ *y* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*fst x*) (*INSERT* (*fst y*) *EMPTY*))) ∧ (*y* = *f_fan_pair* (*fst _2428376*, *snd _2428376*) *x* ∨ *x* = *f_fan_pair* (*fst _2428376*, *snd _2428376*) *y* ∨ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*fst x*) (*INSERT* (*fst y*) *EMPTY*))) (*dartset_leads_into* (*vec* (*0*::*nat*), *fst _2428376*, *snd _2428376*) *f*)))

thm Fan_defs.conforming_diagonal:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. conforming_diagonal (V, E) = (∀(f::(real, 3) cart × (real, 3) cart ⇒ bool) (x::(real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart. IN f (face_set (hypermap_of_fan (V, E))) ∧ IN x f ∧ IN y f ∧ x ≠ y ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT (fst x) (INSERT (fst y) EMPTY))) ∧ (y = f_fan_pair (V, E) x ∨ x = f_fan_pair (V, E) y ∨ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT (fst x) (INSERT (fst y) EMPTY))) (dartset_leads_into (vec (0::nat), V, E) f)))

thm DEF_conforming:

conforming = (λ_2428385::((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). fully_surrounded (fst _2428385, snd _2428385) ∧ conforming_bijection (fst _2428385, snd _2428385) ∧ conforming_half_space (fst _2428385, snd _2428385) ∧ conforming_solid_angle (fst _2428385, snd _2428385) ∧ conforming_diagonal (fst _2428385, snd _2428385))

thm Fan_defs.conforming:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. conforming (V, E) = (fully_surrounded (V, E) ∧ conforming_bijection (V, E) ∧ conforming_half_space (V, E) ∧ conforming_solid_angle (V, E) ∧ conforming_diagonal (V, E))

thm Fan.graph:

∀E::(?'a::type ⇒ bool) ⇒ bool. graph E = (∀e::?'a::type ⇒ bool. E e ⟶ HAS_SIZE e (2::nat))

thm Fan.fan1:

∀(x::?'c::type) (E::?'b::type) V::?'a::type ⇒ bool. fan1 (x, V, E) = (FINITE V ∧ ¬ SUBSET V EMPTY)

thm Fan.fan2:

∀(E::?'b::type) (x::?'a::type) V::?'a::type ⇒ bool. fan2 (x, V, E) = (¬ IN x V)

thm DEF_fan3:

fan3 = (λ_2428394::(real, ?'a::type) cart × ((real, ?'a::type) cart ⇒ bool) × (((real, ?'a::type) cart ⇒ bool) ⇒ bool). ∀v::(real, ?'a::type) cart. IN v (fst (snd _2428394)) ⟶ cyclic_set (GSPEC (λGEN%PVAR%291::(real, ?'a::type) cart. ∃w::(real, ?'a::type) cart. SETSPEC GEN%PVAR%291 (IN (INSERT v (INSERT w EMPTY)) (snd (snd _2428394))) w)) (fst _2428394) v)

thm Fan.fan3:

∀(V::(real, ?'a::type) cart ⇒ bool) (E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) x::(real, ?'a::type) cart. fan3 (x, V, E) = (∀v::(real, ?'a::type) cart. IN v V ⟶ cyclic_set (GSPEC (λGEN%PVAR%291::(real, ?'a::type) cart. ∃w::(real, ?'a::type) cart. SETSPEC GEN%PVAR%291 (IN (INSERT v (INSERT w EMPTY)) E) w)) x v)

thm DEF_fan4:

$fan4 = (\lambda\_2428407::(real, ?'a::type)\ cart \times ((real, ?'a::type)\ cart \Rightarrow bool) \times (((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall e::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ (snd\ (snd\ \_2428407)) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ (fst\ \_2428407)\ EMPTY)\ e)\ (fst\ (snd\ \_2428407)) = EMPTY)$

thm Fan.fan4:

$\forall (E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (x::(real, ?'a::type)\ cart)\ V::(real, ?'a::type)\ cart \Rightarrow bool.\ fan4\ (x, V, E) = (\forall e::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ E \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ e)\ V = EMPTY)$

thm DEF_fan5:

$fan5 = (\lambda\_2428420::(real, ?'b::type)\ cart \times ?'a::type \times (((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall (e::(real, ?'b::type)\ cart \Rightarrow bool)\ f::(real, ?'b::type)\ cart \Rightarrow bool.\ IN\ e\ (snd\ (snd\ \_2428420)) \wedge IN\ f\ (snd\ (snd\ \_2428420)) \wedge e \neq f \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ (fst\ \_2428420)\ EMPTY)\ e)\ (aff\_gt\ (INSERT\ (fst\ \_2428420)\ EMPTY)\ f) = EMPTY)$

thm Fan.fan5:

$\forall (V::?'b::type)\ (E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ fan5\ (x, V, E) = (\forall (e::(real, ?'a::type)\ cart \Rightarrow bool)\ f::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ E \wedge IN\ f\ E \wedge e \neq f \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ e)\ (aff\_gt\ (INSERT\ x\ EMPTY)\ f) = EMPTY)$

thm DEF_base_point_fan:

$base\_point\_fan = fst$

thm Fan.base_point_fan:

$\forall (V::?'c::type)\ (E::?'b::type)\ x::?'a::type.\ base\_point\_fan\ (x, V, E) = x$

thm Fan.set_of_edge:

$\forall (v::?'a::type)\ (E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ V::?'a::type \Rightarrow bool.\ set\_of\_edge\ v\ V\ E = GSPEC\ (\lambda GEN\%PVAR\%292::?'a::type.\ \exists w::?'a::type.\ SETSPEC\ GEN\%PVAR\%292\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ w\ V)\ w)$

thm Fan.fan6:

$\forall (V::?'b::type)\ (E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ fan6\ (x, V, E) = (\forall e::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ E \longrightarrow \neg\ collinear\ (HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ e))$

thm Fan.fan7:

$\forall (E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ fan7\ (x, V, E) = (\forall (e1::(real, ?'a::type)\ cart \Rightarrow bool)\ e2::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e1\ (HOL\_Light\_Import.UNION\ E$

(*GSPEC* (*λGEN%PVAR%293*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%293* (*IN v V*) (*INSERT v EMPTY*)))) ∧ *IN e2* (*HOL_Light_Import.UNION E* (*GSPEC* (*λGEN%PVAR%294*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%294* (*IN v V*) (*INSERT v EMPTY*)))) ⟶ *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) *e1*) (*aff_ge* (*INSERT x EMPTY*) *e2*) = *aff_ge* (*INSERT x EMPTY*) (*HOL_Light_Import.INTER e1 e2*))

thm Fan.FAN:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) = (*SUBSET* (*UNIONS E*) *V* ∧ *graph E* ∧ *fan1* (*x*, *V*, *E*) ∧ *fan2* (*x*, *V*, *E*) ∧ *fan6* (*x*, *V*, *E*) ∧ *fan7* (*x*, *V*, *E*))

thm Fan.GRAPH:

∀ *E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *graph E* = (∀ *e*::*?'a*::*type* ⇒ *bool*. *IN e E* ⟶ *HAS_SIZE e* (*2*::*nat*))

thm Fan.CYCLIC_SET:

cyclic_set (*?W*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*?v*::(*real*, *?'a*::*type*) *cart*) (*?w*::(*real*, *?'a*::*type*) *cart*) = (*?v* ≠ *?w* ∧ *FINITE ?W* ∧ (∀ (*p*::(*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *?'a*::*type*) *cart*) *h*::*real*. *IN p ?W* ∧ *IN q ?W* ∧ *vector_sub p q* = *% h* (*vector_sub ?v ?w*) ⟶ *p* = *q*) ∧ *HOL_Light_Import.INTER ?W* (*hull affine* (*INSERT ?v* (*INSERT ?w EMPTY*))) = *EMPTY*)

thm Polyhedron.CYCLIC_SET_TRANSLATION_EQ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *cyclic_set* (*IMAGE* (*vector_add a*) *s*) (*vector_add a x*) (*vector_add a y*) = *cyclic_set s x y*

thm Polyhedron.CYCLIC_SET_LINEAR_IMAGE:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *cyclic_set* (*IMAGE f s*) (*f x*) (*f y*) = *cyclic_set s x y*

thm Polyhedron.GRAPH_TRANSLATION_EQ:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *graph* (*IMAGE* (*IMAGE* (*vector_add a*)) *E*) = *graph E*

thm Polyhedron.GRAPH_LINEAR_IMAGE_EQ:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *E*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *linear f* ∧ (∀ (*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *graph* (*IMAGE* (*IMAGE f*) *E*) = *graph E*

thm Fan.FAN1_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan1\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan1\ (x,\ V,\ E)$

thm Fan.FAN1_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ (V::(real, ?'b::type)\ cart \Rightarrow bool)\ E::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan1\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan1\ (x,\ V,\ E)$

thm Polyhedron.FAN2_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan2\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan2\ (x,\ V,\ E)$

thm Fan.FAN2_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ (V::(real, ?'b::type)\ cart \Rightarrow bool)\ E::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan2\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan2\ (x,\ V,\ E)$

thm Fan.FAN3_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan3\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan3\ (x,\ V,\ E)$

thm Fan.FAN3_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ (V::(real, ?'b::type)\ cart \Rightarrow bool)\ E::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan3\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan3\ (x,\ V,\ E)$

thm Fan.FAN4_TRANSLATION_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan4\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan4\ (x,\ V,\ E)$

thm Fan.FAN4_LINEAR_IMAGE_EQ:

$\forall (f::(real, ?'b::type)\ cart \Rightarrow (real, ?'a::type)\ cart)\ (x::(real, ?'b::type)\ cart)\ (V::(real, ?'b::type)\ cart \Rightarrow bool)\ E::((real, ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real, ?'b::type)\ cart)\ y::(real, ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan4\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan4\ (x,\ V,\ E)$

thm Polyhedron.FAN5_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan5\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan5\ (x,\ V,\ E)$

thm Fan.FAN5_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (V::(real,\ ?'b::type)\ cart \Rightarrow bool)\ E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan5\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan5\ (x,\ V,\ E)$

thm Polyhedron.FAN6_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan6\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan6\ (x,\ V,\ E)$

thm Fan.FAN6_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (V::(real,\ ?'b::type)\ cart \Rightarrow bool)\ E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan6\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan6\ (x,\ V,\ E)$

thm Polyhedron.FAN7_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ fan7\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = fan7\ (x,\ V,\ E)$

thm Fan.FAN7_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (V::(real,\ ?'b::type)\ cart \Rightarrow bool)\ E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fan7\ (f\ x,\ IMAGE\ f\ V,\ IMAGE\ (IMAGE\ f)\ E) = fan7\ (x,\ V,\ E)$

thm Polyhedron.FAN_TRANSLATION_EQ:

$\forall (a::(real,\ ?'a::type)\ cart)\ (x::(real,\ ?'a::type)\ cart)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vector\_add\ a\ x,\ IMAGE\ (vector\_add\ a)\ V,\ IMAGE\ (IMAGE\ (vector\_add\ a))\ E) = FAN\ (x,\ V,\ E)$

thm Polyhedron.FAN_LINEAR_IMAGE_EQ:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (x::(real,\ ?'b::type)\ cart)\ (V::(real,\ ?'b::type)\ cart \Rightarrow bool)\ E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.$

linear f ∧ (∀(*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *FAN* (*f x*, *IMAGE f V*, *IMAGE* (*IMAGE f*) *E*) = *FAN* (*x*, *V*, *E*)

thm Polyhedron.BASE_POINT_FAN_TRANSLATION_EQ:

∀(*a*::*real*) (*x*::*real*) (*V*::*real* ⇒ *bool*) *E*::(*real* ⇒ *bool*) ⇒ *bool*. *base_point_fan* (*a* + *x*, *IMAGE* (*op* + *a*) *V*, *IMAGE* (*IMAGE* (*op* + *a*)) *E*) = *a* + *base_point_fan* (*x*, *V*, *E*)

thm Fan.BASE_POINT_FAN_LINEAR_IMAGE_EQ:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) (*V*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *E*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *linear f* ⟶ *base_point_fan* (*f x*, *IMAGE f V*, *IMAGE* (*IMAGE f*) *E*) = *f* (*base_point_fan* (*x*, *V*, *E*))

thm Polyhedron.SET_OF_EDGE_TRANSLATION_EQ:

∀(*a*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *set_of_edge* (*vector_add a x*) (*IMAGE* (*vector_add a*) *V*) (*IMAGE* (*IMAGE* (*vector_add a*)) *E*) = *IMAGE* (*vector_add a*) (*set_of_edge x V E*)

thm Polyhedron.SET_OF_EDGE_LINEAR_IMAGE_EQ:

∀(*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'b*::*type*) *cart*) (*V*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *E*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *linear f* ∧ (∀(*x*::(*real*, *?'b*::*type*) *cart*) *y*::(*real*, *?'b*::*type*) *cart*. *f x* = *f y* ⟶ *x* = *y*) ⟶ *set_of_edge* (*f x*) (*IMAGE f V*) (*IMAGE* (*IMAGE f*) *E*) = *IMAGE f* (*set_of_edge x V E*)

thm Fan.set_edges_is_finite_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ *FINITE E*

thm Fan.remark_fan1:

∀(*v*::*?'a*::*type*) (*w*::*?'a*::*type*) (*V*::*?'a*::*type* ⇒ *bool*) *E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *IN v V* ∧ *IN w V* ⟶ *IN w* (*set_of_edge v V E*) = *IN v* (*set_of_edge w V E*)

thm Fan.remark_finite_fan1:

∀(*v*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE V* ⟶ *FINITE* (*set_of_edge v V E*)

thm Fan.properties_of_set_of_edge:

∀(*v*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *u*::(*real*, *3*) *cart*. *SUBSET* (*UNIONS E*) *V* ⟶ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* = *IN u* (*set_of_edge v V E*)

thm Fan.properties_of_set_of_edge_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ⟶ IN (INSERT v (INSERT u EMPTY)) E = IN u (set_of_edge v V E)

thm Fan.properties_of_graph:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (u::(real, 3) cart) v::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ⟶ IN v V

thm Fan.th3a12:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ DISJOINT (INSERT x (INSERT u EMPTY)) (INSERT v EMPTY)

thm Fan.th3a:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ DISJOINT (INSERT x (INSERT v EMPTY)) (INSERT u EMPTY)

thm Fan.th3b:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ x ≠ v

thm Fan.th3b1:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ x ≠ u

thm Fan.th3c:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ ¬ IN u (aff (INSERT x (INSERT v EMPTY)))

thm Fan.th3d:

∀ (x::?'a::type) (v::?'a::type) u::?'a::type. x ≠ v ∧ x ≠ u ⟶ DISJOINT (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))

thm Fan.th3:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ x ≠ v ∧ x ≠ u ∧ DISJOINT (INSERT x (INSERT v EMPTY)) (INSERT u EMPTY) ∧ DISJOINT (INSERT x (INSERT u EMPTY)) (INSERT v EMPTY) ∧ DISJOINT (INSERT x EMPTY) (INSERT v (INSERT u EMPTY)) ∧ ¬ IN u (aff (INSERT x (INSERT v EMPTY)))

thm Fan.collinear1_fan:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. ($\neg$ *collinear* (*INSERT x* (*INSERT u* (*INSERT v EMPTY*)))) = ($\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*))) $\wedge$ *x* $\neq$ *v*)

thm Fan.collinear_fan:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. ($\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*)))) = ($\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*))) $\wedge$ *x* $\neq$ *v*)

thm Fan.th4:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *x* $\neq$ *v*

thm Fan.remark4_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *x* $\neq$ *v* $\wedge$ *x* $\neq$ *u* $\wedge$ *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*) $\wedge$ *DISJOINT* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v EMPTY*) $\wedge$ *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) $\wedge$ $\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*)))

thm Fan.collinears_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ ($\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*)))) = ($\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*))))

thm Fan.remark1_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\longrightarrow$ *FINITE* (*set_of_edge v V E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* = *IN u* (*set_of_edge v V E*) $\wedge$ (*IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) $\wedge$ *x* $\neq$ *v* $\wedge$ *x* $\neq$ *u* $\wedge$ *v* $\neq$ *u* $\wedge$ *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*) $\wedge$ *DISJOINT* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v EMPTY*) $\wedge$ *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) $\wedge$ $\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*))) $\wedge$ *IN v V* $\wedge$ ($\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*)))) = ($\neg$ *IN u* (*aff* (*INSERT x* (*INSERT v EMPTY*)))))

thm Fan.exists_sigma_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *set_of_edge v V E* $\neq$ *INSERT u EMPTY* $\wedge$ *FAN* (*x*, *V*, *E*) $\wedge$ *IN u* (*set_of_edge v V E*) $\longrightarrow$ ($\exists$ *w*::(*real*, *3) cart. IN w (set_of_edge v V E) ∧ w ≠ u ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge v V E) ∧ w1 ≠ u ⟶ azim x v u w ≤ azim x v u w1))*

thm DEF_azim1:

*azim1 = (λ(_2431958::(real, 3) cart) (_2431959::(real, 3) cart) (_2431960::(real, 3) cart) _2431961::(real, 3) cart. real_of_nat (2::nat) * pi − azim _2431958 _2431959 _2431960 _2431961)*

thm Fan.azim1:

*∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. azim1 x v u w = real_of_nat (2::nat) * pi − azim x v u w*

thm Fan.exists_inverse_sigma_fan_alt:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. set_of_edge v V E ≠ INSERT u EMPTY ∧ FAN (x, V, E) ∧ IN u (set_of_edge v V E) ⟶ (∃ w::(real, 3) cart. IN w (set_of_edge v V E) ∧ w ≠ u ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge v V E) ∧ w1 ≠ u ⟶ azim1 x v u w ≤ azim1 x v u w1))

thm Fan.SIGMA_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. set_of_edge v V E ≠ INSERT u EMPTY ∧ FAN (x, V, E) ∧ IN u (set_of_edge v V E) ⟶ IN (sigma_fan x V E v u) (set_of_edge v V E) ∧ sigma_fan x V E v u ≠ u ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge v V E) ∧ w1 ≠ u ⟶ azim x v u (sigma_fan x V E v u) ≤ azim x v u w1)

thm Fan.sigma_fan_in_set_of_edge:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ IN u (set_of_edge v V E) ⟶ IN (sigma_fan x V E v u) (set_of_edge v V E)

thm Fan.AFF_GE_2_1:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. DISJOINT (INSERT x (INSERT v EMPTY)) (INSERT w EMPTY) ⟶ aff_ge (INSERT x (INSERT v EMPTY)) (INSERT w EMPTY) = GSPEC (λGEN%PVAR%306::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%306 (∃ (t1::real) (t2::real) t3::real. (0::real) ≤ t3 ∧ t1 + (t2 + t3) = (1::real) ∧ y = vector_add (% t1 x) (vector_add (% t2 v) (% t3 w))) y)

thm Fan.AFF_GE_1_2:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. DISJOINT (INSERT x EMPTY) (INSERT v (INSERT w EMPTY)) ⟶ aff_ge (INSERT x EMPTY) (INSERT v (INSERT w EMPTY)) = GSPEC

(λGEN%PVAR%307::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%307 (∃ (t1::real) (t2::real) t3::real. (0::real) ≤ t2 ∧ (0::real) ≤ t3 ∧ t1 + (t2 + t3) = (1::real) ∧ y = vector_add (% t1 x) (vector_add (% t2 v) (% t3 w))) y)

thm Fan.AFF_GE_1_1:

∀ (x::(real, ?'b::type) cart) (v::(real, ?'b::type) cart) w::?'a::type. DISJOINT (INSERT x EMPTY) (INSERT v EMPTY) ⟶ aff_ge (INSERT x EMPTY) (INSERT v EMPTY) = GSPEC (λGEN%PVAR%308::(real, ?'b::type) cart. ∃ y::(real, ?'b::type) cart. SETSPEC GEN%PVAR%308 (∃ (t1::real) t2::real. (0::real) ≤ t2 ∧ t1 + t2 = (1::real) ∧ y = vector_add (% t1 x) (% t2 v)) y)

thm Fan.UNIQUE_FOINT_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (aff_ge (INSERT x EMPTY) (INSERT v (INSERT w EMPTY))) ≠ aff_ge (INSERT x EMPTY) (INSERT v EMPTY) ⟶ u = w

thm Fan.UNIQUE1_POINT_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ IN w (aff_gt (INSERT x (INSERT v EMPTY)) (INSERT u EMPTY)) ⟶ u = w

thm Fan.UNIQUE_AZIM_POINT_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) w1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ IN (INSERT v (INSERT w1 EMPTY)) E ∧ azim x v u w = azim x v u w1 ⟶ w = w1

thm Fan.UNIQUE_AZIM_0_POINT_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ azim x v u w = (0::real) ⟶ u = w

thm Fan.UNIQUE_SIGMA_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. set_of_edge v V E ≠ INSERT u EMPTY ∧ FAN (x, V, E) ∧ IN u (set_of_edge v V E) ∧ IN w (set_of_edge v V E) ∧ w ≠ u ∧ (∀ w1::(real, 3) cart. IN w1 (set_of_edge v V

E) ∧ *w1* ≠ *u* ⟶ *azim x v u w* ≤ *azim x v u w1*) ⟶ *sigma_fan x V E v u* = *w*

thm Fan.CYCLIC_SET_EDGE_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN v V* ⟶ *cyclic_set* (*set_of_edge v V E*) *x v*

thm Fan.subset_cyclic_set_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *W*::(*real*, *3*) *cart* ⇒ *bool*. *SUBSET V W* ∧ *cyclic_set W x v* ⟶ *cyclic_set V x v*

thm Fan.property_of_cyclic_set:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ⟶ *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub u x*) *EMPTY*)))

thm Fan.property_of_cyclic_set1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w1 EMPTY*)))

thm Fan.property_of_cyclic_set2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w2 EMPTY*)))

thm Fan.property_of_cyclic_set3:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*)))

thm Fan.properties_of_cyclic_set:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ⟶ *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub u x*) *EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w2 EMPTY*)))

thm DEF_d1_fan:

d1_fan = (λ*_2451589*::(*real*, *3*) *cart* × ((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*). *GSPEC* (λ*GEN%PVAR%309*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃(*x′*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (w::(real, 3) cart) w1::(real, 3) cart. SETSPEC GEN%PVAR%309 (x′ = fst _2451589 ∧ IN (INSERT v (INSERT w EMPTY)) (snd (snd _2451589)) ∧ w1 = sigma_fan (fst _2451589) (fst (snd _2451589)) (snd (snd _2451589)) v w) (x′, v, w, w1)))

thm Fan.d1_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. d1_fan (x, V, E) = GSPEC (λGEN%PVAR%309::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∃(x′::(real, 3) cart) (v::(real, 3) cart) (w::(real, 3) cart) w1::(real, 3) cart. SETSPEC GEN%PVAR%309 (x′ = x ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ w1 = sigma_fan x V E v w) (x′, v, w, w1))

thm DEF_d2_fan:

d2_fan = (λ_2451602::?′b::type × (?′a::type ⇒ bool) × ((?′a::type ⇒ bool) ⇒ bool). GSPEC (λGEN%PVAR%310::?′b::type × ?′a::type. ∃(x′::?′b::type) v::?′a::type. SETSPEC GEN%PVAR%310 (x′ = fst _2451602 ∧ fst (snd _2451602) v ∧ set_of_edge v (fst (snd _2451602)) (snd (snd _2451602)) = EMPTY) (x′, v)))

thm Fan.d2_fan:

∀(x::?′b::type) (V::?′a::type ⇒ bool) E::(?′a::type ⇒ bool) ⇒ bool. d2_fan (x, V, E) = GSPEC (λGEN%PVAR%310::?′b::type × ?′a::type. ∃(x′::?′b::type) v::?′a::type. SETSPEC GEN%PVAR%310 (x′ = x ∧ V v ∧ set_of_edge v V E = EMPTY) (x′, v))

thm DEF_inverse_sigma_fan_alt:

inverse_sigma_fan_alt = (λ(_2451615::(real, 3) cart) (_2451616::(real, 3) cart ⇒ bool) (_2451617::((real, 3) cart ⇒ bool) ⇒ bool) (_2451618::(real, 3) cart) _2451619::(real, 3) cart. SOME a::(real, 3) cart. sigma_fan _2451615 _2451616 _2451617 _2451618 a = _2451619)

thm Fan.inverse_sigma_fan_alt:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. inverse_sigma_fan_alt x V E v w = (SOME a::(real, 3) cart. sigma_fan x V E v a = w)

thm DEF_e_fan:

e_fan = (λ(_2451660::(real, 3) cart) (_2451661::(real, 3) cart ⇒ bool) _2451662::((real, 3) cart ⇒ bool) ⇒ bool. GABS (λf::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∀(x::(real, 3) cart) (v::(real, 3) cart) (w::(real, 3) cart) w1::(real, 3) cart. GEQ (f (x, v, w, w1)) (x, w, v, sigma_fan x _2451661 _2451662 w v)))

thm Fan.e_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. e_fan x V E = GABS (λf::(real, 3) cart × (real, 3) cart × (real, 3) cart

× (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *w*, *v*, *sigma_fan x V E w v*))

thm DEF_f_fan:

f_fan = (λ(*_2451681*::(*real*, *3*) *cart*) (*_2451682*::(*real*, *3*) *cart* ⇒ *bool*) *_2451683*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *w*, *inverse_sigma_fan_alt x _2451682 _2451683 w v*, *v*)))

thm Fan.f_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *f_fan x V E* = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *w*, *inverse_sigma_fan_alt x V E w v*, *v*))

thm DEF_n_fan:

n_fan = (λ(*_2451702*::(*real*, *3*) *cart*) (*_2451703*::(*real*, *3*) *cart* ⇒ *bool*) *_2451704*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *v*, *sigma_fan x _2451703 _2451704 v w*, *sigma_fan x _2451703 _2451704 v* (*sigma_fan x _2451703 _2451704 v w*))))

thm Fan.n_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *n_fan x V E* = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *v*, *sigma_fan x V E v w*, *sigma_fan x V E v* (*sigma_fan x V E v w*)))

thm Fan.pr1:

pr1 = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) *x*)

thm Fan.pr2:

pr2 = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) *v*)

thm Fan.pr3:

pr3 = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) *w*)

thm Fan.pr4:

pr4 = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) *w1*)

thm DEF_power_map_points:

power_map_points = (*SOME power_map_points*::*nat* ⇒ (*?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'a*::*type*) ⇒ *?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *nat* ⇒ *?'a*::*type*. ∀ *_2451734*::*nat*. (∀ (*f*::*?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'a*::*type*) (*x*::*?'e*::*type*) (*V*::*?'d*::*type*) (*E*::*?'c*::*type*) (*v*::*?'b*::*type*) *w*::*?'a*::*type*. *power_map_points _2451734 f x V E v w* (*0*::*nat*) = *w*) ∧ (∀ (*f*::*?'e*::*type* ⇒ *?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'a*::*type*) (*x*::*?'e*::*type*) (*V*::*?'d*::*type*) (*E*::*?'c*::*type*) (*v*::*?'b*::*type*) (*w*::*?'a*::*type*) *n*::*nat*. *power_map_points _2451734 f x V E v w* (*Suc n*) = *f x V E v* (*power_map_points _2451734 f x V E v w n*))) (*136*::*nat*)

thm Fan.power_map_points_conjunct0:

power_map_points (*?f*::*?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'e*::*type* ⇒ *?'e*::*type*) (*?x*::*?'d*::*type*) (*?V*::*?'c*::*type*) (*?E*::*?'b*::*type*) (*?v*::*?'a*::*type*) (*?w*::*?'e*::*type*) (*0*::*nat*) = *?w*

thm Fan.power_map_points_conjunct1:

power_map_points (*?f*::*?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'e*::*type* ⇒ *?'e*::*type*) (*?x*::*?'d*::*type*) (*?V*::*?'c*::*type*) (*?E*::*?'b*::*type*) (*?v*::*?'a*::*type*) (*?w*::*?'e*::*type*) (*Suc* (*?n*::*nat*)) = *?f ?x ?V ?E ?v* (*power_map_points ?f ?x ?V ?E ?v ?w ?n*)

thm Fan.power_map_points:

power_map_points (*?f*::*?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'e*::*type* ⇒ *?'e*::*type*) (*?x*::*?'d*::*type*) (*?V*::*?'c*::*type*) (*?E*::*?'b*::*type*) (*?v*::*?'a*::*type*) (*?w*::*?'e*::*type*) (*0*::*nat*) = *?w* ∧ *power_map_points ?f ?x ?V ?E ?v ?w* (*Suc* (*?n*::*nat*)) = *?f ?x ?V ?E ?v* (*power_map_points ?f ?x ?V ?E ?v ?w ?n*)

thm DEF_o_funs:

o_funs = (λ(*_2451735*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *_2451736*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *GABS* (λ*f'*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *GEQ* (*f'* (*x*, *y*, *z*, *t*)) (*_2451735* (*pr1* (*_2451736* (*x*, *y*, *z*, *t*)), *pr2* (*_2451736* (*x*, *y*, *z*, *t*)), *pr3* (*_2451736* (*x*, *y*, *z*, *t*)), *pr4* (*_2451736* (*x*, *y*, *z*, *t*))))))

thm Fan.o_funs:

$\forall (f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ g::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ o\_funs\ f\ g = GABS\ (\lambda f'::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ \forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ (z::(real,\ 3)\ cart)\ t::(real,\ 3)\ cart.\ GEQ\ (f'\ (x,\ y,\ z,\ t))\ (f\ (pr1\ (g\ (x,\ y,\ z,\ t)),\ pr2\ (g\ (x,\ y,\ z,\ t)),\ pr3\ (g\ (x,\ y,\ z,\ t)),\ pr4\ (g\ (x,\ y,\ z,\ t)))))$

thm DEF_power_maps:

$power\_maps = (SOME\ power\_maps::nat \Rightarrow ((real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow nat \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ \forall\ \_2451756::nat.\ (\forall (f::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ power\_maps\ \_2451756\ f\ x\ V\ E\ (0::nat) = i\_fan\ x\ V\ E) \wedge (\forall (f::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ n::nat.\ power\_maps\ \_2451756\ f\ x\ V\ E\ (Suc\ n) = o\_funs\ (f\ x\ V\ E)\ (power\_maps\ \_2451756\ f\ x\ V\ E\ n)))\ (137::nat)$

thm Fan.power_maps_conjunct0:

$power\_maps\ (?f::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (?x::(real,\ 3)\ cart)\ (?V::(real,\ 3)\ cart \Rightarrow bool)\ (?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (0::nat) = i\_fan\ ?x\ ?V\ ?E$

thm Fan.power_maps_conjunct1:

$power\_maps\ (?f::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (?x::(real,\ 3)\ cart)\ (?V::(real,\ 3)\ cart \Rightarrow bool)\ (?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (Suc\ (?n::nat)) = o\_funs\ (?f\ ?x\ ?V\ ?E)\ (power\_maps\ ?f\ ?x\ ?V\ ?E\ ?n)$

thm Fan.power_maps:

$power\_maps\ (?f::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,$

3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (?x::(real, 3) cart) (?V::(real, 3) cart ⇒ bool) (?E::((real, 3) cart ⇒ bool) ⇒ bool) (0::nat) = i_fan ?x ?V ?E ∧ power_maps ?f ?x ?V ?E (Suc (?n::nat)) = o_funs (?f ?x ?V ?E) (power_maps ?f ?x ?V ?E ?n)

thm DEF_power_n_fan:

power_n_fan = (λ(_2451757::?'b::type) (_2451758::(real, 3) cart ⇒ bool) (_2451759::((real, 3) cart ⇒ bool) ⇒ bool) _2451760::nat. GABS (λf::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∀(x::(real, 3) cart) (v::(real, 3) cart) (w::(real, 3) cart) w1::?'a::type. GEQ (f (x, v, w, w1)) (x, v, power_map_points sigma_fan x _2451758 _2451759 v w _2451760, power_map_points sigma_fan x _2451758 _2451759 v w (Suc _2451760))))

thm DEF_a_node_fan:

a_node_fan = (λ(_2451789::(real, 3) cart) (_2451790::(real, 3) cart ⇒ bool) (_2451791::((real, 3) cart ⇒ bool) ⇒ bool) _2451792::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type. GSPEC (λGEN%PVAR%311::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∃a::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%311 (∃n::nat. a = power_maps n_fan _2451789 _2451790 _2451791 n (_2451789, fst (snd _2451792), fst (snd (snd _2451792)), sigma_fan _2451789 _2451790 _2451791 (fst (snd _2451792)) (fst (snd (snd _2451792))))) a))

thm Fan.a_node_fan:

∀(w1::?'a::type) (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. a_node_fan x V E (x, v, w, w1) = GSPEC (λGEN%PVAR%311::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∃a::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%311 (∃n::nat. a = power_maps n_fan x V E n (x, v, w, sigma_fan x V E v w)) a)

thm Fan.xfan:

∀(V::?'b::type) (E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) x::(real, ?'a::type) cart. xfan (x, V, E) = GSPEC (λGEN%PVAR%312::(real, ?'a::type) cart. ∃v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%312 (∃e::(real, ?'a::type) cart ⇒ bool. E e ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)

thm DEF_yfan_deprecated:

yfan_deprecated = (λ_2451842::(real, 3) cart × ?'a::type × (((real, 3) cart ⇒ bool) ⇒ bool). GSPEC (λGEN%PVAR%313::(real, 3) cart. ∃v::(real, 3) cart. SETSPEC GEN%PVAR%313 (∃e::(real, 3) cart ⇒ bool. snd (snd _2451842) e ∧ ¬ IN v (aff_ge (INSERT (fst _2451842) EMPTY) e)) v))

thm Fan.yfan_deprecated:

∀(V::?'a::type) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart. yfan_deprecated (x, V, E) = GSPEC (λGEN%PVAR%313::(real, 3) cart. ∃v::(real, 3) cart.

SETSPEC GEN%PVAR%313 ($\exists$ *e*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *E e* $\wedge$ $\neg$ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)

thm Fan.image_power_map_points:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN u* (*set_of_edge v V E*) $\longrightarrow$ *IN* (*power_map_points sigma_fan x V E v u i*) (*set_of_edge v V E*)

thm Fan.IN2_ORBITS_FAN:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *IN* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*)) *E*

thm Fan.IN1_ORBITS_FAN:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN u* (*set_of_edge v V E*) $\longrightarrow$ *IN* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*)) *E*

thm Fan.remark_power_map_points:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *IN* (*power_map_points sigma_fan x V E v u i*) (*set_of_edge v V E*) $\wedge$ *IN* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*)) *E* $\wedge$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*))) $\wedge$ *x* $\neq$ *power_map_points sigma_fan x V E v u i* $\wedge$ *v* $\neq$ *power_map_points sigma_fan x V E v u i* $\wedge$ *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*) $\wedge$ $\neg$ *IN* (*power_map_points sigma_fan x V E v u i*) (*aff* (*INSERT x* (*INSERT v EMPTY*))) $\wedge$ *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*))

thm Fan.imp_norm_not_zero_fan:

$\forall$ (*v*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *v* $\neq$ *x* $\longrightarrow$ *vector_norm* (*vector_sub v x*) $\neq$ (*0*::*real*)

thm Fan.imp_norm_gl_zero_fan:

$\forall$ (*v*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *v* $\neq$ *x* $\longrightarrow$ (*0*::*real*) $<$ *inverse_class.inverse* (*vector_norm* (*vector_sub v x*))

thm Fan.imp_inv_norm_not_zero_fan:

$\forall$ (*v*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *v* $\neq$ *x* $\longrightarrow$ *inverse_class.inverse* (*vector_norm* (*vector_sub v x*)) $\neq$ (*0*::*real*)

thm Fan.imp_norm_ge_zero_fan:

$\forall (v::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ v \neq x \longrightarrow (0::real) \leq inverse\_class.inverse\ (vector\_norm\ (vector\_sub\ v\ x))$

thm Fan.norm_of_normal_vector_is_unit_fan:

$\forall (v::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ v \neq x \longrightarrow vector\_norm\ (\%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_sub\ v\ x)))\ (vector\_sub\ v\ x)) = (1::real)$

thm DEF_e3_fan:

$e3\_fan = (\lambda(\_2452871::(real,\ 3)\ cart)\ (\_2452872::(real,\ 3)\ cart)\ \_2452873::(real,\ 3)\ cart.\ \%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_sub\ \_2452872\ \_2452871)))\ (vector\_sub\ \_2452872\ \_2452871))$

thm Fan.e3_fan:

$\forall (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ e3\_fan\ x\ v\ u = \%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_sub\ v\ x)))\ (vector\_sub\ v\ x)$

thm DEF_e2_fan:

$e2\_fan = (\lambda(\_2452892::(real,\ 3)\ cart)\ (\_2452893::(real,\ 3)\ cart)\ \_2452894::(real,\ 3)\ cart.\ \%\ (inverse\_class.inverse\ (vector\_norm\ (cross\ (e3\_fan\ \_2452892\ \_2452893\ \_2452894)\ (vector\_sub\ \_2452894\ \_2452892))))\ (cross\ (e3\_fan\ \_2452892\ \_2452893\ \_2452894)\ (vector\_sub\ \_2452894\ \_2452892)))$

thm Fan.e2_fan:

$\forall (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ e2\_fan\ x\ v\ u = \%\ (inverse\_class.inverse\ (vector\_norm\ (cross\ (e3\_fan\ x\ v\ u)\ (vector\_sub\ u\ x))))\ (cross\ (e3\_fan\ x\ v\ u)\ (vector\_sub\ u\ x))$

thm DEF_e1_fan:

$e1\_fan = (\lambda(\_2452913::(real,\ 3)\ cart)\ (\_2452914::(real,\ 3)\ cart)\ \_2452915::(real,\ 3)\ cart.\ cross\ (e2\_fan\ \_2452913\ \_2452914\ \_2452915)\ (e3\_fan\ \_2452913\ \_2452914\ \_2452915))$

thm Fan.e1_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ e1\_fan\ x\ v\ u = cross\ (e2\_fan\ x\ v\ u)\ (e3\_fan\ x\ v\ u)$

thm Fan.e3_mul_dist_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \longrightarrow \%\ (distance\ (v,\ x))\ (e3\_fan\ x\ v\ u) = vector\_sub\ v\ x$

thm Fan.norm_dot_fan:

$\forall x::(real,\ 3)\ cart.\ vector\_norm\ x = (1::real) \longrightarrow dot\ x\ x = (1::real)$

thm Fan.e3_is_normal_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ⟶ *dot* (*e3_fan* *x v u*) (*e3_fan x v u*) = (*1*::*real*)

thm Fan.e2_is_normal_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*e2_fan x v u*) (*e2_fan x v u*) = (*1*::*real*)

thm Fan.e2_orthogonal_e3_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*e2_fan x v u*) (*e3_fan x v u*) = (*0*::*real*)

thm Fan.e1_is_normal_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*e1_fan x v u*) (*e1_fan x v u*) = (*1*::*real*)

thm Fan.e1_orthogonal_e3_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*e1_fan x v u*) (*e3_fan x v u*) = (*0*::*real*)

thm Fan.e1_orthogonal_e2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*e1_fan x v u*) (*e2_fan x v u*) = (*0*::*real*)

thm Fan.e1_cross_e2_dot_e3_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ (*0*::*real*) < *dot* (*cross* (*e1_fan x v u*) (*e2_fan x v u*)) (*e3_fan x v u*)

thm Fan.orthonormal_e1_e2_e3_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *orthonormal* (*e1_fan x v u*) (*e2_fan x v u*) (*e3_fan x v u*)

thm Fan.dot_e2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *u* ≠ *x* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub* *u x*) *EMPTY*))) ⟶ *dot* (*vector_sub u x*) (*e2_fan x v u*) = (*0*::*real*)

thm Fan.vdot_e2_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg$ $collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub$ $u\ x)\ EMPTY))) \longrightarrow dot\ (vector\_sub\ v\ x)\ (e2\_fan\ x\ v\ u) = (0::real)$

thm Fan.vcross_e3_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg$ $collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub$ $u\ x)\ EMPTY))) \longrightarrow cross\ (vector\_sub\ v\ x)\ (e3\_fan\ x\ v\ u) = vec\ (0::nat)$

thm Fan.udot_e1_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg$ $collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub$ $u\ x)\ EMPTY))) \longrightarrow (0::real) < dot\ (vector\_sub\ u\ x)\ (e1\_fan\ x\ v\ u)$

thm Fan.udot_e1_fan1:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg$ $collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub$ $u\ x)\ EMPTY))) \longrightarrow (0::real) \leq dot\ (vector\_sub\ u\ x)\ (e1\_fan\ x\ v\ u)$

thm Fan.vdot_e1_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg$ $collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub$ $u\ x)\ EMPTY))) \longrightarrow dot\ (vector\_sub\ v\ x)\ (e1\_fan\ x\ v\ u) = (0::real)$

thm Fan.properties_coordinate:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x$ $(INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow orthonormal\ (e1\_fan\ x\ v\ u)\ (e2\_fan$ $x\ v\ u)\ (e3\_fan\ x\ v\ u) \wedge \%\ (distance\ (v,\ x))\ (e3\_fan\ x\ v\ u) = vector\_sub\ v\ x$ $\wedge\ cross\ (vector\_sub\ v\ x)\ (e3\_fan\ x\ v\ u) = vec\ (0::nat) \wedge dot\ (vector\_sub\ v\ x)$ $(e2\_fan\ x\ v\ u) = (0::real) \wedge dot\ (vector\_sub\ u\ x)\ (e2\_fan\ x\ v\ u) = (0::real) \wedge$ $(0::real) \leq dot\ (vector\_sub\ u\ x)\ (e1\_fan\ x\ v\ u) \wedge (0::real) < dot\ (vector\_sub$ $u\ x)\ (e1\_fan\ x\ v\ u) \wedge dot\ (vector\_sub\ v\ x)\ (e1\_fan\ x\ v\ u) = (0::real)$

thm Fan.module_of_vector:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)$ $(r::real)\ (psi::real)\ h::real.\ v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))$ $(INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub\ u\ x)\ EMPTY))) \wedge (0::real)$ $< r \wedge w = vector\_add\ (\%\ (r * cos\ psi)\ (e1\_fan\ x\ v\ u))\ (vector\_add\ (\%\ (r * sin$ $psi)\ (e2\_fan\ x\ v\ u))\ (\%\ h\ (vector\_sub\ v\ x))) \longrightarrow sqrt\ ((dot\ (cross\ w\ (e3\_fan\ x$ $v\ u))\ (e1\_fan\ x\ v\ u))^2 + (dot\ (cross\ w\ (e3\_fan\ x\ v\ u))\ (e2\_fan\ x\ v\ u))^2) = r$

thm Fan.collinear_imp_azim_is_rezo_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (y::real)\ (h1::real)$ $h2::real.\ v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u$ $EMPTY))) \wedge (0::real) \leq y \wedge y < real\_of\_nat\ (2::nat) * pi \wedge (\forall (e1::(real,\ 3)$ $cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \wedge \%\ (distance$

$(v,\ x))\ e3 = vector\_sub\ v\ x \longrightarrow (\exists (psi::real)\ (r1::real)\ r2::real.\ vector\_sub\ u\ x = vector\_add\ (\%\ (r1 * cos\ psi)\ e1)\ (vector\_add\ (\%\ (r1 * sin\ psi)\ e2)\ (\%\ h1\ (vector\_sub\ v\ x))) \wedge vector\_sub\ u\ x = vector\_add\ (\%\ (r2 * cos\ (psi + y))\ e1)\ (vector\_add\ (\%\ (r2 * sin\ (psi + y))\ e2)\ (\%\ h2\ (vector\_sub\ v\ x))) \wedge (0::real) < r1 \wedge (0::real) < r2)) \longrightarrow y = (0::real)$

thm Fan.azim_is_zero_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow azim\ x\ v\ u\ u = (0::real)$

thm Fan.SINCOS_PRINCIPAL_VALUE_FAN:

$\forall x::real.\ \exists y::real.\ ((0::real) \leq y \wedge y < real\_of\_nat\ (2::nat) * pi) \wedge sin\ y = sin\ x \wedge cos\ y = cos\ x$

thm Fan.sin_of_u_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (r1::real)\ (psi::real)\ h1::real.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ x\ (INSERT\ v\ EMPTY))) \wedge v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub\ u\ x)\ EMPTY))) \wedge (0::real) < r1 \wedge vector\_sub\ u\ x = vector\_add\ (\%\ (r1 * cos\ psi)\ (e1\_fan\ x\ v\ u))\ (vector\_add\ (\%\ (r1 * sin\ psi)\ (e2\_fan\ x\ v\ u))\ (\%\ h1\ (vector\_sub\ v\ x))) \longrightarrow sin\ psi = (0::real)$

thm Fan.cos_of_u_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (r1::real)\ (psi::real)\ h1::real.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ x\ (INSERT\ v\ EMPTY))) \wedge v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub\ u\ x)\ EMPTY))) \wedge (0::real) < r1 \wedge vector\_sub\ u\ x = vector\_add\ (\%\ (r1 * cos\ psi)\ (e1\_fan\ x\ v\ u))\ (vector\_add\ (\%\ (r1 * sin\ psi)\ (e2\_fan\ x\ v\ u))\ (\%\ h1\ (vector\_sub\ v\ x))) \longrightarrow cos\ psi = (1::real)$

thm Fan.sincos_of_u_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (r1::real)\ (psi::real)\ h1::real.\ \neg\ collinear\ (INSERT\ u\ (INSERT\ x\ (INSERT\ v\ EMPTY))) \wedge v \neq x \wedge u \neq x \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (vector\_sub\ v\ x)\ (INSERT\ (vector\_sub\ u\ x)\ EMPTY))) \wedge (0::real) < r1 \wedge vector\_sub\ u\ x = vector\_add\ (\%\ (r1 * cos\ psi)\ (e1\_fan\ x\ v\ u))\ (vector\_add\ (\%\ (r1 * sin\ psi)\ (e2\_fan\ x\ v\ u))\ (\%\ h1\ (vector\_sub\ v\ x))) \longrightarrow sin\ psi = (0::real) \wedge cos\ psi = (1::real)$

thm Fan.sincos1_of_u_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (r1::real)\ (psi::real)\ h1::real.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge (0::real) < r1 \wedge vector\_sub\ u\ x = vector\_add\ (\%\ (r1 * cos\ psi)\ (e1\_fan\ x\ v\ u))\ (vector\_add\ (\%\ (r1 * sin\ psi)\ (e2\_fan\ x\ v\ u))\ (\%\ h1\ (vector\_sub\ v\ x))) \longrightarrow sin\ psi = (0::real) \wedge cos\ psi = (1::real)$

thm Fan.sum1_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ∧ *azim x v u w1* + *azim x v w1 w2* < *real_of_nat* (*2*::*nat*) ∗ *pi* ⟶ *azim x v u w2* = *azim x v u w1* + *azim x v w1 w2*

thm Fan.sum3_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim x v u w1* + *azim x v w1 w2* < *real_of_nat* (*2*::*nat*) ∗ *pi* ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w2 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *azim x v u w2* = *azim x v u w1* + *azim x v w1 w2*

thm Fan.sum2_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *cyclic_set* (*INSERT u* (*INSERT w1* (*INSERT w2 EMPTY*))) *x v* ∧ *azim x v u w1* ≤ *azim x v u w2* ⟶ *azim x v u w2* = *azim x v u w1* + *azim x v w1 w2*

thm Fan.sum4_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim x v u w1* ≤ *azim x v u w2* ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w2 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *azim x v u w2* = *azim x v u w1* + *azim x v w1 w2*

thm Fan.sum5_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim x v w1 w2* ≤ *azim x v u w2* ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w2 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *azim x v u w2* = *azim x v u w1* + *azim x v w1 w2*

thm Fan.SUR_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ (∃ *w*::(*real*, *3*) *cart*. *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E v w* = *u*)

thm Fan.MONO_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E v u* = *sigma_fan x V E v w* ⟶ *u* = *w*

thm Fan.permutes_sigma_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \longrightarrow permutes\ (extension\_sigma\_fan\ x\ V\ E\ v)\ (set\_of\_edge\ v\ V\ E)$

thm Fan.exists_function_inverse_sigma_fan_alt:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \longrightarrow (\exists g::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart.\ (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow IN\ (INSERT\ v\ (INSERT\ (g\ w)\ EMPTY))\ E) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow sigma\_fan\ x\ V\ E\ v\ (g\ w) = w) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow g\ (sigma\_fan\ x\ V\ E\ v\ w) = w))$

thm DEF_inverse1_sigma_fan:

$inverse1\_sigma\_fan = (\lambda(\_2484524::(real,\ 3)\ cart)\ (\_2484525::(real,\ 3)\ cart \Rightarrow bool)\ (\_2484526::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ \_2484527::(real,\ 3)\ cart.\ SOME\ g::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart.\ (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ \_2484527\ (INSERT\ w\ EMPTY))\ \_2484526 \longrightarrow IN\ (INSERT\ \_2484527\ (INSERT\ (g\ w)\ EMPTY))\ \_2484526) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ \_2484527\ (INSERT\ w\ EMPTY))\ \_2484526 \longrightarrow sigma\_fan\ \_2484524\ \_2484525\ \_2484526\ \_2484527\ (g\ w) = w) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ \_2484527\ (INSERT\ w\ EMPTY))\ \_2484526 \longrightarrow g\ (sigma\_fan\ \_2484524\ \_2484525\ \_2484526\ \_2484527\ w) = w))$

thm Fan.inverse1_sigma_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real,\ 3)\ cart.\ inverse1\_sigma\_fan\ x\ V\ E\ v = (SOME\ g::(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart.\ (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow IN\ (INSERT\ v\ (INSERT\ (g\ w)\ EMPTY))\ E) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow sigma\_fan\ x\ V\ E\ v\ (g\ w) = w) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow g\ (sigma\_fan\ x\ V\ E\ v\ w) = w))$

thm Fan.INVERSE1_SIGMA_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \longrightarrow (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow IN\ (INSERT\ v\ (INSERT\ (inverse1\_sigma\_fan\ x\ V\ E\ v\ w)\ EMPTY))\ E) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow sigma\_fan\ x\ V\ E\ v\ (inverse1\_sigma\_fan\ x\ V\ E\ v\ w) = w) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow inverse1\_sigma\_fan\ x\ V\ E\ v\ (sigma\_fan\ x\ V\ E\ v\ w) = w)$

thm DEF_f1_fan:

f1_fan = (λ(*_2484556*::(*real*, *3*) *cart*) (*_2484557*::(*real*, *3*) *cart* ⇒ *bool*) *_2484558*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *w*, *inverse1_sigma_fan x _2484557 _2484558 w v*, *v*)))

thm Fan.f1_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *f1_fan x V E* = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *v*, *w*, *w1*)) (*x*, *w*, *inverse1_sigma_fan x V E w v*, *v*))

thm Fan.node_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *n*::*nat*. *power_maps n_fan x V E n* = *power_n_fan x V E n*

thm Fan.EQ_PAIR_4:

∀(*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) (*a1*::(*real*, *3*) *cart*) (*b1*::(*real*, *3*) *cart*) (*c1*::(*real*, *3*) *cart*) *da*::(*real*, *3*) *cart*. ((*a*, *b*, *c*, *da*) = (*a1*, *b1*, *c1*, *?d1.0*::(*real*, *3*) *cart*)) = (*a* = *a1* ∧ *b* = *b1* ∧ *c* = *c1* ∧ *da* = *?d1.0*)

thm Fan.MONO_N_FAN:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀(*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *n_fan x V E a* = *n_fan x V E b* ⟶ *a* = *b*)

thm Fan.SUR_N_FAN:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃*b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *n_fan x V E b* = *a*))

thm Fan.simp_inverse_sigma_fan_alt:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *inverse_sigma_fan_alt x V E v w* = *HOL_Light_Import.inverse* (*sigma_fan x V E v*) *w*

thm Fan.SUR_F1_FAN:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*

× (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃ *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *f1_fan x V E b* = *a*))

thm Fan.MONO_F1_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀ (*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *f1_fan x V E a* = *f1_fan x V E b* ⟶ *a* = *b*)

thm Fan.MONO_E_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀ (*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *e_fan x V E a* = *e_fan x V E b* ⟶ *a* = *b*)

thm Fan.SUR_E_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀ *a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃ *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *e_fan x V E b* = *a*))

thm Fan.permuters_of_enf_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀ (*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *n_fan x V E a* = *n_fan x V E b* ⟶ *a* = *b*) ∧ (∀ *a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃ *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *n_fan x V E b* = *a*)) ∧ (∀ (*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *f1_fan x V E a* = *f1_fan x V E b* ⟶ *a* = *b*) ∧ (∀ *a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃ *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *f1_fan x V E b* = *a*)) ∧ (∀ (*a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *e_fan x V E a* = *e_fan x V E b* ⟶ *a* = *b*) ∧ (∀ *a*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN a* (*d1_fan* (*x*, *V*, *E*)) ⟶ (∃ *b*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN b* (*d1_fan* (*x*, *V*, *E*)) ∧ *e_fan x V E b* = *a*))

thm Fan.condition_hypermap_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall a::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ a\ (d1\_fan\ (x, V, E)) \longrightarrow e\_fan\ x\ V\ E\ (n\_fan\ x\ V\ E\ (f1\_fan\ x\ V\ E\ a)) = a)$

thm Fan.plain_hypermap_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall a::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ a\ (d1\_fan\ (x, V, E)) \longrightarrow e\_fan\ x\ V\ E\ (e\_fan\ x\ V\ E\ a) = a)$

thm Fan.e_fan_no_fix_point:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall a::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ a\ (d1\_fan\ (x, V, E)) \longrightarrow e\_fan\ x\ V\ E\ a \neq a)$

thm Fan.f_fan_no_fix_point:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall a::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ a\ (d1\_fan\ (x, V, E)) \longrightarrow f1\_fan\ x\ V\ E\ a \neq a)$

thm Fan.orbit_map:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ orbit\_map\ f\ x = GSPEC\ (\lambda GEN\%PVAR\%314::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%314\ ((0::nat) \leq n)\ (POWER\ f\ n\ x))$

thm Fan.POWER_RIGHT:

$\forall (k::nat)\ f::?'a::type \Rightarrow ?'a::type.\ POWER\ f\ (Suc\ k) = f \circ POWER\ f\ k$

thm Fan.power_n_fan:

$\forall (l::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow POWER\ (n\_fan\ x\ V\ E)\ l\ (x, v, w, sigma\_fan\ x\ V\ E\ v\ w) = (x, v, power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ w\ l, power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ w\ (Suc\ l))$

thm Fan.distinct_nodes:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall (k::nat)\ (l::nat)\ a::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ a\ (d1\_fan\ (x, V, E)) \longrightarrow (POWER\ (n\_fan\ x\ V\ E)\ k \circ e\_fan\ x\ V\ E)\ a = (e\_fan\ x\ V\ E \circ POWER\ (n\_fan\ x\ V\ E)\ l)\ a \longrightarrow POWER\ (n\_fan\ x\ V\ E)\ l\ a = a)$

thm Fan.edge_lie_different_nodes:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \longrightarrow (\forall (n::nat)\ a::(real, 3)\ cart \times (real, 3)\ cart \times (real,$

3) cart × (real, 3) cart. IN a (d1_fan (x, V, E)) ⟶ e_fan x V E a ≠ POWER (n_fan x V E) n a)

thm DEF_d20_fan:

d20_fan = (λ_2492710::?'b::type × (?'a::type ⇒ bool) × ((?'a::type ⇒ bool) ⇒ bool). GSPEC (λGEN%PVAR%315::?'b::type × ?'a::type × ?'a::type × ?'a::type. ∃(x'::?'b::type) v::?'a::type. SETSPEC GEN%PVAR%315 (x' = fst _2492710 ∧ fst (snd _2492710) v ∧ set_of_edge v (fst (snd _2492710)) (snd (snd _2492710)) = EMPTY) (x', v, v, v)))

thm Fan.d20_fan:

∀(x::?'b::type) (V::?'a::type ⇒ bool) E::(?'a::type ⇒ bool) ⇒ bool. d20_fan (x, V, E) = GSPEC (λGEN%PVAR%315::?'b::type × ?'a::type × ?'a::type × ?'a::type. ∃(x'::?'b::type) v::?'a::type. SETSPEC GEN%PVAR%315 (x' = x ∧ V v ∧ set_of_edge v V E = EMPTY) (x', v, v, v))

thm DEF_d_fan:

d_fan = (λ_2492723::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). HOL_Light_Import.UNION (d1_fan (fst _2492723, fst (snd _2492723), snd (snd _2492723))) (d20_fan (fst _2492723, fst (snd _2492723), snd (snd _2492723))))

thm Fan.d_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. d_fan (x, V, E) = HOL_Light_Import.UNION (d1_fan (x, V, E)) (d20_fan (x, V, E))

thm DEF_hypermap_of_fanx:

hypermap_of_fanx = (λ_2492736::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). LET (λp::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. LET_END (hypermap (d_fan (fst _2492736, fst (snd _2492736), snd (snd _2492736)), p e_fan, p n_fan, p f_fan))) (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t (fst _2492736) (fst (snd _2492736)) (snd (snd _2492736))) (d1_fan (fst _2492736, fst (snd _2492736), snd (snd _2492736)))))

thm Fan.hypermap_of_fanx:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. hypermap_of_fanx (x, V, E) = LET (λp::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real,

3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. LET_END (hypermap (d_fan (x, V, E), p e_fan, p n_fan, p f_fan))) (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E)))

thm DEF_hypermap1_of_fanx:

hypermap1_of_fanx = (λ_2492749::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). LET (λp::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. LET_END (hypermap (d_fan (fst _2492749, fst (snd _2492749), snd (snd _2492749)), p e_fan, p n_fan, p f1_fan))) (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t (fst _2492749) (fst (snd _2492749)) (snd (snd _2492749))) (d1_fan (fst _2492749, fst (snd _2492749), snd (snd _2492749)))))

thm Fan.hypermap1_of_fanx:

∀ *(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. hypermap1_of_fanx (x, V, E) = LET (λp::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. LET_END (hypermap (d_fan (x, V, E), p e_fan, p n_fan, p f1_fan))) (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E)))*

thm Fan.finite_d1_fan:

∀ *(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ FINITE (d1_fan (x, V, E))*

thm Fan.finite_d20_fan:

∀ *(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ FINITE (d20_fan (x, V, E))*

thm Fan.finite_d_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ FINITE (d_fan (x, V, E))

thm Fan.subset_d_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. SUBSET (d1_fan (x, V, E)) (d_fan (x, V, E)) ∧ SUBSET (d20_fan (x, V, E)) (d_fan (x, V, E))

thm Fan.FACE_FAN_NOT_EMPTY:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) s::?'a::type. FAN (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ⟶ ds ≠ EMPTY

thm Fan.into_domain_e_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ (∀y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d_fan (x, V, E)) ⟶ IN (?p e_fan y) (d_fan (x, V, E)))

thm Fan.into_domain1_e_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ (∀y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d1_fan (x, V, E)) ⟶ IN (e_fan x V E y) (d1_fan (x, V, E)))

thm Fan.e_fan_permutes:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ permutes (?p e_fan) (d_fan (x, V, E))

thm Fan.into_domain_f1_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ (∀ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d_fan (x, V, E)) ⟶ IN (?p f1_fan y) (d_fan (x, V, E)))

thm Fan.into_domain1_f1_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ (∀ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d1_fan (x, V, E)) ⟶ IN (f1_fan x V E y) (d1_fan (x, V, E)))

thm Fan.f1_fan_permutes:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ permutes (?p f1_fan) (d_fan (x, V, E))

thm Fan.into_domain_n_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ (∀ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d_fan (x, V, E)) ⟶ IN (?p n_fan y) (d_fan (x, V, E)))

thm Fan.into_domain1_n_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ⟶ (∀ y::(real, 3) cart × (real, 3) cart × (real, 3) cart

× (real, 3) cart. IN y (d1_fan (x, V, E)) ⟶ IN (n_fan x V E y) (d1_fan (x, V, E)))

thm Fan.n_fan_permutes:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ permutes (?p n_fan) (d_fan (x, V, E))

thm Fan.id_enf_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ∧ ¬ IN y (d1_fan (x, V, E)) ⟶ ?p e_fan y = y ∧ ?p n_fan y = y ∧ ?p f1_fan y = y

thm Fan.id_power_enf_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (n::nat) p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ p = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ∧ ¬ IN y (d1_fan (x, V, E)) ⟶ POWER (p e_fan) n y = y ∧ POWER (p n_fan) n y = y ∧ POWER (p f1_fan) n y = y

thm Fan.into_domain_efn_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart ×

$(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) = (\lambda t{::}(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ x\ V\ E)\ (d1\_fan\ (x,\ V,\ E))) \longrightarrow (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow ?p\ e\_fan\ y = e\_fan\ x\ V\ E\ y) \wedge (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow ?p\ n\_fan\ y = n\_fan\ x\ V\ E\ y) \wedge (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow ?p\ f1\_fan\ y = f1\_fan\ x\ V\ E\ y)$

thm Fan.power_map_fix_set:

$\forall (n{::}nat)\ (f{::}?'a{::}type \Rightarrow ?'a{::}type)\ (g{::}?'a{::}type \Rightarrow ?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ (\forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow f\ x = g\ x) \wedge (\forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow IN\ (g\ x)\ s) \longrightarrow (\forall x{::}?'a{::}type.\ IN\ x\ s \longrightarrow POWER\ f\ n\ x = POWER\ g\ n\ x)$

thm Fan.into_domain_power_efn_fan:

$\forall (x{::}(real,\ 3)\ cart)\ (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (n{::}nat)\ p{::}((real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge p = (\lambda t{::}(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ x\ V\ E)\ (d1\_fan\ (x,\ V,\ E))) \longrightarrow (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow POWER\ (p\ e\_fan)\ n\ y = POWER\ (e\_fan\ x\ V\ E)\ n\ y) \wedge (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow POWER\ (p\ n\_fan)\ n\ y = POWER\ (n\_fan\ x\ V\ E)\ n\ y) \wedge (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow POWER\ (p\ f1\_fan)\ n\ y = POWER\ (f1\_fan\ x\ V\ E)\ n\ y)$

thm Fan.power_fun_in_domain:

$\forall (n{::}nat)\ (f{::}?'a{::}type \Rightarrow ?'a{::}type)\ s{::}?'a{::}type \Rightarrow bool.\ (\forall y{::}?'a{::}type.\ IN\ y\ s \longrightarrow IN\ (f\ y)\ s) \longrightarrow (\forall y{::}?'a{::}type.\ IN\ y\ s \longrightarrow IN\ (POWER\ f\ n\ y)\ s)$

thm Fan.into_domain1_power_efn_fan:

$\forall (x{::}(real,\ 3)\ cart)\ (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ n{::}nat.\ FAN\ (x,\ V,\ E) \longrightarrow (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow IN\ (POWER\ (e\_fan\ x\ V\ E)\ n\ y)\ (d1\_fan\ (x,\ V,\ E))) \wedge (\forall y{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow IN\ (POWER\ (n\_fan\ x$

V E) n y) (d1_fan (x, V, E))) ∧ (∀ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d1_fan (x, V, E)) ⟶ IN (POWER (f1_fan x V E) n y) (d1_fan (x, V, E)))

thm Fan.lemma_hypermap1_of_fanx:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (?p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ ?p e_fan ∘ (?p n_fan ∘ ?p f1_fan) = id

thm Fan.hypermap_of_fan_rep:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) p::((real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ p = (λt::(real, 3) cart ⇒ ((real, 3) cart ⇒ bool) ⇒ (((real, 3) cart ⇒ bool) ⇒ bool) ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. res (t x V E) (d1_fan (x, V, E))) ⟶ dart (hypermap1_of_fanx (x, V, E)) = d_fan (x, V, E) ∧ edge_map (hypermap1_of_fanx (x, V, E)) = p e_fan ∧ node_map (hypermap1_of_fanx (x, V, E)) = p n_fan ∧ face_map (hypermap1_of_fanx (x, V, E)) = p f1_fan

thm Fan.properties_of_f1_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ y = f1_fan x V E y1 ∧ IN y1 (d1_fan (x, V, E)) ⟶ pr3 y1 = pr2 y ∧ IN (INSERT (pr2 y1) (INSERT (pr3 y1) EMPTY)) E ∧ IN (INSERT (pr2 y) (INSERT (pr3 y) EMPTY)) E ∧ pr2 y1 = sigma_fan x V E (pr2 y) (pr3 y)

thm Fan.face_subset_dart_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ⟶ SUBSET ds (d_fan (x, V, E))

thm Fan.properties_of_elements_in_face_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge IN\ y\ ds \longrightarrow IN\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY))\ E \vee pr2\ y = pr3\ y$

thm Fan.fully_surrounded_is_non_isolated_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \longrightarrow d20\_fan\ (x,\ V,\ E) = EMPTY$

thm Fan.dartset_fully_surrounded_is_non_isolated_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \longrightarrow d\_fan\ (x,\ V,\ E) = d1\_fan\ (x,\ V,\ E)$

thm Fan.properties_of_elements_in_face_fully_surroundedfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge IN\ y\ ds \longrightarrow IN\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY))\ E$

thm Fan.AAUHTVE:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ p::((real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge p = (\lambda t::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ x\ V\ E)\ (d1\_fan\ (x,\ V,\ E))) \longrightarrow FINITE\ (d\_fan\ (x,\ V,\ E)) \wedge permutes\ (p\ e\_fan)\ (d\_fan\ (x,\ V,\ E)) \wedge permutes\ (p\ n\_fan)\ (d\_fan\ (x,\ V,\ E)) \wedge permutes\ (p\ f1\_fan)\ (d\_fan\ (x,\ V,\ E)) \wedge p\ e\_fan \circ (p\ n\_fan \circ p\ f1\_fan) = id \wedge (\forall a::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ a\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow e\_fan\ x\ V\ E\ (e\_fan\ x\ V\ E\ a) = a) \wedge (\forall a::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ a\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow e\_fan\ x\ V\ E\ a \neq a) \wedge (\forall a::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ a\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow f1\_fan\ x\ V\ E\ a \neq a) \wedge (\forall (k::nat)\ (l::nat)\ a::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ a\ (d1\_fan\ (x,\ V,\ E)) \longrightarrow (POWER\ (n\_fan\ x\ V\ E)\ k \circ e\_fan\ x\ V\ E)\ a = (e\_fan\ x\ V\ E \circ POWER\ (n\_fan\ x\ V\ E)\ l)\ a \longrightarrow POWER\ (n\_fan\ x\ V\ E)\ l\ a = a) \wedge (\forall (n::nat)\ a::(real,\ 3)\ cart \times (real,$

3) cart × (real, 3) cart × (real, 3) cart. IN a (d1_fan (x, V, E)) ⟶ e_fan x V E a ≠ POWER (n_fan x V E) n a)

thm Gmlwkpk.AFF_GE_1_2_0:

∀ (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. v ≠ vec (0::nat) ∧ w ≠ vec (0::nat) ⟶ aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT w EMPTY)) = GSPEC (λGEN%PVAR%318::(real, ?'a::type) cart. ∃ (a::real) b::real. SETSPEC GEN%PVAR%318 ((0::real) ≤ a ∧ (0::real) ≤ b) (vector_add (% a v) (% b w)))

thm Gmlwkpk.AFF_GE_1_1_0:

∀ v::(real, ?'a::type) cart. v ≠ vec (0::nat) ⟶ aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v EMPTY) = GSPEC (λGEN%PVAR%319::(real, ?'a::type) cart. ∃ a::real. SETSPEC GEN%PVAR%319 ((0::real) ≤ a) (% a v))

thm Gmlwkpk.GMLWKPK:

∀ (x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. graph E ⟶ fan7 (x, V, E) = (∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%320::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%320 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%321::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%321 (IN v V) (INSERT v EMPTY)))) ⟶ (HOL_Light_Import.INTER e1 e2 = EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = INSERT x EMPTY) ∧ (∀ v::(real, ?'a::type) cart. HOL_Light_Import.INTER e1 e2 = INSERT v EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (INSERT v EMPTY)))

thm Gmlwkpk.GMLWKPK_ALT:

∀ (x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. graph E ∧ (∀ e::(real, ?'a::type) cart ⇒ bool. IN e E ⟶ ¬ IN x e) ⟶ fan7 (x, V, E) = ((∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%322::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%322 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%323::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%323 (IN v V) (INSERT v EMPTY)))) ∧ HOL_Light_Import.INTER e1 e2 = EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = INSERT x EMPTY) ∧ (∀ (e1::(real, ?'a::type) cart ⇒ bool) (e2::(real, ?'a::type) cart ⇒ bool) v::(real, ?'a::type) cart. IN e1 E ∧ IN e2 E ∧ HOL_Light_Import.INTER e1 e2 = INSERT v EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (INSERT v EMPTY)))

thm Gmlwkpk.GMLWKPK_SIMPLE:

$\forall (E::((real, ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ x::(real, ?'a::type)\ cart.\ SUBSET\ (UNIONS\ E)\ V \wedge graph\ E \wedge fan6\ (x, V, E) \wedge (\forall e::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e\ E \longrightarrow \neg\ IN\ x\ e) \longrightarrow fan7\ (x, V, E) = (\forall (e1::(real, ?'a::type)\ cart \Rightarrow bool)\ e2::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ e1\ (HOL\_Light\_Import.UNION\ E\ (GSPEC\ (\lambda GEN\%PVAR\%324::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists v::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%324\ (IN\ v\ V)\ (INSERT\ v\ EMPTY)))) \wedge IN\ e2\ (HOL\_Light\_Import.UNION\ E\ (GSPEC\ (\lambda GEN\%PVAR\%325::(real, ?'a::type)\ cart \Rightarrow bool.\ \exists v::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%325\ (IN\ v\ V)\ (INSERT\ v\ EMPTY)))) \wedge HOL\_Light\_Import.INTER\ e1\ e2 = EMPTY \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e1)\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e2) = INSERT\ x\ EMPTY)$

thm Topology.CARD_SIGMA_FAN:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real, 3)\ cart.\ FAN\ (x, V, E) \longrightarrow CARD\ (IMAGE\ (sigma\_fan\ x\ V\ E\ v)\ (set\_of\_edge\ v\ V\ E)) = CARD\ (set\_of\_edge\ v\ V\ E)$

thm Topology.MONO_AZIM_SIGMA_FAN:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ w::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ v\ w \neq u \longrightarrow azim\ x\ v\ u\ w \leq azim\ x\ v\ u\ (sigma\_fan\ x\ V\ E\ v\ w)$

thm Topology.MONO_POWER_SIGMA_FAN:

$\forall (i::nat)\ (j::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ u::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge j < i \wedge power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i = power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ j \longrightarrow u = power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (i - j)$

thm Topology.MONO_POWER_MAP_POINTS1_FAN:

$\forall (i::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ u::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ u\ (set\_of\_edge\ v\ V\ E) \wedge set\_of\_edge\ v\ V\ E \neq INSERT\ u\ EMPTY \longrightarrow power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i \neq power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (Suc\ i)$

thm Topology.MONO_AZIM_POWER_SIGMA_FAN:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ i::nat.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (Suc\ i) \neq u \longrightarrow azim\ x\ v\ u\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i) \leq azim\ x\ v\ u\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (Suc\ i))$

thm DEF_complement_set:

complement_set = (λ*_2522047*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%326*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%326* (¬ *IN y* (*aff _2522047*)) *y*))

thm Topology.complement_set:

∀ (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *complement_set* (*INSERT x* (*INSERT v EMPTY*)) = *GSPEC* (λ*GEN%PVAR%326*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%326* (¬ *IN y* (*aff* (*INSERT x* (*INSERT v EMPTY*)))) *y*)

thm Topology.subset_aff:

∀ (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SUBSET* (*aff* (*INSERT x* (*INSERT v EMPTY*))) *HOL_Light_Import.UNIV*

thm Topology.union_aff:

∀ (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *HOL_Light_Import.UNIV* = *HOL_Light_Import.UNION* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*complement_set* (*INSERT x* (*INSERT v EMPTY*)))

thm DEF_if_azims_fan:

if_azims_fan = (λ(*_2522052*::(*real*, *3*) *cart*) (*_2522053*::(*real*, *3*) *cart* ⇒ *bool*) (*_2522054*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*_2522055*::(*real*, *3*) *cart*) (*_2522056*::(*real*, *3*) *cart*) *_2522057*::*nat*. *if _2522057* = *CARD* (*set_of_edge _2522055 _2522053 _2522054*) *then real_of_nat* (*2*::*nat*) ∗ *pi else azim _2522052 _2522055 _2522056* (*power_map_points sigma_fan _2522052 _2522053 _2522054 _2522055 _2522056 _2522057*))

thm Topology.if_azims_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. *if_azims_fan x V E v u i* = (*if i* = *CARD* (*set_of_edge v V E*) *then real_of_nat* (*2*::*nat*) ∗ *pi else azim x v u* (*power_map_points sigma_fan x V E v u i*))

thm Topology.if_azims_works_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. (*0*::*real*) ≤ *if_azims_fan x V E v u i* ∧ *if_azims_fan x V E v u i* ≤ *real_of_nat* (*2*::*nat*) ∗ *pi*

thm DEF_set_of_orbits_points_fan:

set_of_orbits_points_fan = (λ(*_2522112*::(*real*, *3*) *cart*) (*_2522113*::(*real*, *3*) *cart* ⇒ *bool*) (*_2522114*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*_2522115*::(*real*, *3*) *cart*) *_2522116*::(*real*, *3*) *cart*. *GSPEC* (λ*GEN%PVAR%327*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%327* ((*0*::*nat*) ≤ *i*) (*power_map_points sigma_fan _2522112 _2522113 _2522114 _2522115 _2522116 i*)))

thm Topology.set_of_orbits_points_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. set_of_orbits_points_fan x V E v u = GSPEC (λGEN%PVAR%327::(real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%327 ((0::nat) ≤ i) (power_map_points sigma_fan x V E v u i))

thm DEF_number_of_orbits_points_fan:

number_of_orbits_points_fan = (λ(_2522157::(real, 3) cart) (_2522158::(real, 3) cart ⇒ bool) (_2522159::((real, 3) cart ⇒ bool) ⇒ bool) (_2522160::(real, 3) cart) _2522161::(real, 3) cart. CARD (set_of_orbits_points_fan _2522157 _2522158 _2522159 _2522160 _2522161))

thm Topology.number_of_orbits_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. number_of_orbits_points_fan x V E v u = CARD (set_of_orbits_points_fan x V E v u)

thm Topology.addition_sigma_fan:

∀ (m::nat) (n::nat) (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. power_map_points sigma_fan x V E v u (m + n) = power_map_points sigma_fan x V E v (power_map_points sigma_fan x V E v u n) m

thm Topology.fix_point_sigma_fan:

∀ (q::nat) (i::nat) (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. power_map_points sigma_fan x V E v u i = u ⟶ power_map_points sigma_fan x V E v u (q ∗ i) = u

thm Topology.i_IN_ORBITS_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) i::nat. IN (power_map_points sigma_fan x V E v u i) (set_of_orbits_points_fan x V E v u)

thm Topology.u_IN_ORBITS_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. IN u (set_of_orbits_points_fan x V E v u)

thm Topology.IN_ORBITS_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. IN w (set_of_orbits_points_fan x V E v u) ⟶ IN (sigma_fan x V E v w) (set_of_orbits_points_fan x V E v u)

thm Topology.ORBITS_SUBSET_EDGE_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v

(*INSERT u EMPTY*)) *E* ⟶ *SUBSET* (*set_of_orbits_points_fan x V E v u*) (*set_of_edge v V E*)

thm Topology.CARD_ORBITS_EDGE_FAN_LE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *CARD* (*set_of_orbits_points_fan x V E v u*) ≤ *CARD* (*set_of_edge v V E*)

thm Topology.FINITE_ORBITS_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *FINITE* (*set_of_orbits_points_fan x V E v u*)

thm Topology.ORBITS_SIGMA_FAN:

∀ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *power_map_points sigma_fan x V E v u i* = *u* ∧ *i* ≠ (*0*::*nat*) ⟶ *set_of_orbits_points_fan x V E v u* = *GSPEC* (λ*GEN%PVAR%328*::(*real*, *3*) *cart*. ∃ *j*::*nat*. *SETSPEC GEN%PVAR%328* (*j* < *i*) (*power_map_points sigma_fan x V E v u j*))

thm Topology.IMAGE_SEG:

∀ (*n*::*nat*) *f*::*nat* ⇒ *?'a*::*type*. *IMAGE f* (*GSPEC* (λ*GEN%PVAR%329*::*nat*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%329* (*i* < *n*) *i*)) = *GSPEC* (λ*GEN%PVAR%330*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%330* (*i* < *n*) (*f i*))

thm Topology.FINITE_SERIES:

∀ (*n*::*nat*) *f*::*nat* ⇒ *?'a*::*type*. *FINITE* (*GSPEC* (λ*GEN%PVAR%331*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%331* (*i* < *n*) (*f i*)))

thm Topology.CARD_FINITE_SERIES_LE:

∀ (*n*::*nat*) *f*::*nat* ⇒ *?'a*::*type*. *CARD* (*GSPEC* (λ*GEN%PVAR%332*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%332* (*i* < *n*) (*f i*))) ≤ *n*

thm Topology.CARD_FINITE_SERIES_EQ:

∀ (*n*::*nat*) *f*::*nat* ⇒ *?'a*::*type*. (∀ (*i*::*nat*) *j*::*nat*. *i* < *n* ∧ *j* < *i* ⟶ *f i* ≠ *f j*) ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%333*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%333* (*i* < *n*) (*f i*))) = *n*

thm Topology.CARD_ORBITS_SIGMA_FAN_LE:

∀ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *power_map_points sigma_fan x V E v u i* = *u* ∧ *i* ≠ (*0*::*nat*) ⟶ *CARD* (*set_of_orbits_points_fan x V E v u*) ≤ *i*

thm Topology.exists_inverse_in_orbits_sigma_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ $\neg$ *IN y* (*set_of_orbits_points_fan x V E v u*) $\longrightarrow$ ($\exists$ *w*::(*real*, *3*) *cart*. *IN w* (*set_of_orbits_points_fan x V E v u*) $\wedge$ *w* $\neq$ *y* $\wedge$ ($\forall$ *w1*::(*real*, *3*) *cart*. *IN w1* (*set_of_orbits_points_fan x V E v u*) $\wedge$ *w1* $\neq$ *y* $\longrightarrow$ *azim1 x v y w* $\leq$ *azim1 x v y w1*))

thm Topology.key_lemma_cyclic_fan:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ (*0*::*nat*) < *i* $\wedge$ *i* < *CARD* (*set_of_edge v V E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *power_map_points sigma_fan x V E v u i* $\neq$ *u*

thm Topology.cyclic_power_sigma_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*i*::*nat*) *j*::*nat*. *FAN* (*x*, *V*, *E*) $\wedge$ *i* < *CARD* (*set_of_edge v V E*) $\wedge$ *j* < *i* $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *power_map_points sigma_fan x V E v u i* $\neq$ *power_map_points sigma_fan x V E v u j*

thm Topology.CARD_SET_OF_ORBITS_POINTS_FAN:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *CARD* (*set_of_orbits_points_fan x V E v u*) = *CARD* (*set_of_edge v V E*)

thm Topology.ORBITS_EQ_SET_EDGE_FAN:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *set_of_edge v V E* = *set_of_orbits_points_fan x V E v u*

thm Topology.SIMP_ORBITS_POINTS_FAN:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *GSPEC* (λ*GEN%PVAR%338*::(*real*, *3*) *cart*. $\exists$ *i*::*nat*. *SETSPEC GEN%PVAR%338* (*i* < *CARD* (*set_of_edge v V E*)) (*power_map_points sigma_fan x V E v u i*)) = *set_of_orbits_points_fan x V E v u*

thm Topology.ORDER_POWER_SIGMA_FAN:

$\forall$ (*i*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *i* = *CARD* (*set_of_edge v V E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\longrightarrow$ *power_map_points sigma_fan x V E v u i* = *u*

thm Topology.SUM_IF_AZIMS_FAN:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ i::nat.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge (0::nat) < i \wedge i < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow if\_azims\_fan\ x\ V\ E\ v\ u\ (Suc\ i) = if\_azims\_fan\ x\ V\ E\ v\ u\ i + azim\ x\ v\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i)\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (Suc\ i))$

thm DEF_azim_i_fan:

$azim\_i\_fan = (\lambda(\_2529879::(real, 3)\ cart)\ (\_2529880::(real, 3)\ cart \Rightarrow bool)\ (\_2529881::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (\_2529882::(real, 3)\ cart)\ (\_2529883::(real, 3)\ cart)\ \_2529884::nat.\ azim\ \_2529879\ \_2529882\ (power\_map\_points\ sigma\_fan\ \_2529879\ \_2529880\ \_2529881\ \_2529882\ \_2529883\ \_2529884)\ (power\_map\_points\ sigma\_fan\ \_2529879\ \_2529880\ \_2529881\ \_2529882\ \_2529883\ (Suc\ \_2529884)))$

thm Topology.azim_i_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ i::nat.\ azim\_i\_fan\ x\ V\ E\ v\ u\ i = azim\ x\ v\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i)\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ (Suc\ i))$

thm Topology.SUM_EQ_IF_AZIMS_FAN:

$\forall (i::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ u::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge set\_of\_edge\ v\ V\ E \neq INSERT\ u\ EMPTY \wedge (1::nat) \neq CARD\ (set\_of\_edge\ v\ V\ E) \wedge i < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow sum\ (dotdot\ (0::nat)\ i)\ (azim\_i\_fan\ x\ V\ E\ v\ u) = if\_azims\_fan\ x\ V\ E\ v\ u\ (Suc\ i)$

thm Topology.SUM_AZIMS_EQ_2PI_FAN:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ u::(real, 3)\ cart.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge set\_of\_edge\ v\ V\ E \neq INSERT\ u\ EMPTY \wedge (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow sum\ (dotdot\ (0::nat)\ (CARD\ (set\_of\_edge\ v\ V\ E) - (1::nat)))\ (azim\_i\_fan\ x\ V\ E\ v\ u) = real\_of\_nat\ (2::nat) * pi$

thm Topology.AZIM_LE_POWER_SIGMA_FAN:

$\forall (i::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ j::nat.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge set\_of\_edge\ v\ V\ E \neq INSERT\ u\ EMPTY \wedge j < i \wedge i < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow azim\ x\ v\ u\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ j) < azim\ x\ v\ u\ (power\_map\_points\ sigma\_fan\ x\ V\ E\ v\ u\ i)$

thm Topology.SUM_AZIM_POWER_SIGMA_FAN:

$\forall (i::nat)\ (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ j::nat.\ FAN\ (x, V, E) \wedge$

IN (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *set_of_edge* *v* *V* *E* ≠ *INSERT* *u* *EMPTY* ∧ *j* < *i* ∧ *i* < *CARD* (*set_of_edge* *v* *V* *E*) ⟶ *azim* *x* *v* *u* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*) = *azim* *x* *v* *u* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *j*) + *azim* *x* *v* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *j*) (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*)

thm Topology.SUM1_IFAZIMS_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*i*::*nat*) *j*::*nat*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *set_of_edge* *v* *V* *E* ≠ *INSERT* *u* *EMPTY* ∧ *j* < *i* ∧ *i* < *CARD* (*set_of_edge* *v* *V* *E*) ⟶ *if_azims_fan* *x* *V* *E* *v* *u* *i* = *if_azims_fan* *x* *V* *E* *v* *u* *j* + *azim* *x* *v* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *j*) (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*)

thm Topology.ULEKUUB:

(∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*i*::*nat*) *j*::*nat*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *set_of_edge* *v* *V* *E* ≠ *INSERT* *u* *EMPTY* ∧ *j* < *i* ∧ *i* < *CARD* (*set_of_edge* *v* *V* *E*) ⟶ *if_azims_fan* *x* *V* *E* *v* *u* *i* = *if_azims_fan* *x* *V* *E* *v* *u* *j* + *azim* *x* *v* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *j*) (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*)) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *set_of_edge* *v* *V* *E* ≠ *INSERT* *u* *EMPTY* ∧ (*1*::*nat*) < *CARD* (*set_of_edge* *v* *V* *E*) ⟶ *sum* (*dotdot* (*0*::*nat*) (*CARD* (*set_of_edge* *v* *V* *E*) − (*1*::*nat*))) (*azim_i_fan* *x* *V* *E* *v* *u*) = *real_of_nat* (*2*::*nat*) ∗ *pi*)

thm DEF_wedge2_fan:

wedge2_fan = (λ(*_2530930*::(*real*, *3*) *cart*) (*_2530931*::(*real*, *3*) *cart* ⇒ *bool*) (*_2530932*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*_2530933*::(*real*, *3*) *cart*) (*_2530934*::(*real*, *3*) *cart*) *_2530935*::*nat*. *GSPEC* (λ*GEN%PVAR%339*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%339* (*if_azims_fan* *_2530930* *_2530931* *_2530932* *_2530933* *_2530934* *_2530935* = *azim* *_2530930* *_2530933* *_2530934* *y* ∧ *IN* *y* (*complement_set* (*INSERT* *_2530930* (*INSERT* *_2530933* *EMPTY*)))) *y*))

thm Topology.wedge2_fan:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*i*::*nat*) (*u*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *wedge2_fan* *x* *V* *E* *v* *u* *i* = *GSPEC* (λ*GEN%PVAR%339*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%339* (*if_azims_fan* *x* *V* *E* *v* *u* *i* = *azim* *x* *v* *u* *y* ∧ *IN* *y* (*complement_set* (*INSERT* *x* (*INSERT* *v* *EMPTY*)))) *y*)

thm Topology.affine_hull_2_fan:

∀ (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *aff* (*INSERT* *x* (*INSERT* *v* *EMPTY*)) = *GSPEC* (λ*GEN%PVAR%340*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC*

GEN%PVAR%340 (∃ (*t1*::*real*) *t2*::*real*. *t1* + *t2* = (*1*::*real*) ∧ *y* = *vector_add* (% *t1* *x*) (% *t2* *v*)) *y*)

thm Topology.AFF_GT_1_1:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*) ⟶ *aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*) = *GSPEC* (λ*GEN%PVAR%341*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%341* (∃ (*t1*::*real*) *t2*::*real*. (*0*::*real*) < *t2* ∧ *t1* + *t2* = (*1*::*real*) ∧ *y* = *vector_add* (% *t1* *x*) (% *t2* *v*)) *y*)

thm Topology.th:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *w* *EMPTY*))) ⟶ *GSPEC* (λ*GEN%PVAR%343*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%343* (¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *y* *EMPTY*))) ∧ *azim* *x* *v* *u* *w* = *azim* *x* *v* *u* *y*) *y*) = *aff_gt* (*INSERT* *x* (*INSERT* *v* *EMPTY*)) (*INSERT* *w* *EMPTY*)

thm Topology.th1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t3* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ∧ *DISJOINT* (*INSERT* *x* (*INSERT* *v* *EMPTY*)) (*INSERT* *w* *EMPTY*) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *w* *EMPTY*))) ⟶ *azim* *x* *v* *u* *w* = *azim* *x* *v* *u* (*vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* *w*)))

thm Topology.th2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *x* ≠ *v* ⟶ *IN* *w* (*complement_set* (*INSERT* *x* (*INSERT* *v* *EMPTY*))) ⟶ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *w* *EMPTY*)))

thm Topology.COMPLEMENT_SET_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. ¬ *IN* *w* (*aff* (*INSERT* *x* (*INSERT* *v* *EMPTY*))) ∧ *t3* ≠ (*0*::*real*) ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ⟶ *IN* (*vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* *w*))) (*complement_set* (*INSERT* *x* (*INSERT* *v* *EMPTY*)))

thm Topology.aff_gt_subset_wedge_fan2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. *i* ≠ *CARD* (*set_of_edge* *v* *V* *E*) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*) *EMPTY*))) ⟶ *SUBSET* (*aff_gt* (*INSERT* *x* (*INSERT* *v* *EMPTY*)) (*INSERT* (*power_map_points* *sigma_fan* *x* *V* *E* *v* *u* *i*) *EMPTY*)) (*wedge2_fan* *x* *V* *E* *v* *u* *i*)

thm Topology.wedge_fan2_subset_aff_gt:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*))) $\wedge$ *i* $\neq$ *CARD* (*set_of_edge v V E*) $\longrightarrow$ *SUBSET* (*wedge2_fan x V E v u i*) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*))

thm Topology.wedge_fan2_equal_aff_gt:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*))) $\wedge$ *i* $\neq$ *CARD* (*set_of_edge v V E*) $\longrightarrow$ *wedge2_fan x V E v u i* = *aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*)

thm Topology.wedge_fan2_equal_aff_gt_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *i* $\neq$ *CARD* (*set_of_edge v V E*) $\longrightarrow$ *wedge2_fan x V E v u i* = *aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT* (*power_map_points sigma_fan x V E v u i*) *EMPTY*)

thm DEF_wedge3_fan:

wedge3_fan = (λ(*_2533065*::(*real*, *3*) *cart*) (*_2533066*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*_2533067*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*_2533068*::(*real*, *3*) *cart*) (*_2533069*::(*real*, *3*) *cart*) *_2533070*::*nat*. *GSPEC* (λ*GEN%PVAR%344*::(*real*, *3*) *cart*. $\exists$ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%344* (*if_azims_fan _2533065 _2533066 _2533067 _2533068 _2533069 _2533070* < *azim _2533065 _2533068 _2533069 y* $\wedge$ *azim _2533065 _2533068 _2533069 y* < *if_azims_fan _2533065 _2533066 _2533067 _2533068 _2533069* (*Suc _2533070*) $\wedge$ *IN y* (*complement_set* (*INSERT _2533065* (*INSERT _2533068 EMPTY*)))) *y*))

thm Topology.wedge3_fan:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) (*i*::*nat*) (*x*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *wedge3_fan x V E v u i* = *GSPEC* (λ*GEN%PVAR%344*::(*real*, *3*) *cart*. $\exists$ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%344* (*if_azims_fan x V E v u i* < *azim x v u y* $\wedge$ *azim x v u y* < *if_azims_fan x V E v u* (*Suc i*) $\wedge$ *IN y* (*complement_set* (*INSERT x* (*INSERT v EMPTY*)))) *y*)

thm Topology.w_dart_eq_wedge3_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT* *v* (*INSERT u EMPTY*)) *E* ∧ *i* < *CARD* (*set_of_edge v V E*) ∧ (*1*::*nat*) < *CARD* (*set_of_edge v V E*) ⟶ *w_dart_fan x V E* (*x*, *v*, *power_map_points sigma_fan x V E v u i*, *power_map_points sigma_fan x V E v u* (*Suc i*)) = *wedge3_fan x V E v u i*

thm Topology.UNION_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *HOL_Light_Import*.*UNIV* = *HOL_Light_Import*.*UNION* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*HOL_Light_Import*.*UNION* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%347*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%347* ((*0*::*nat*) ≤ *i* ∧ *i* < *CARD* (*set_of_edge v V E*)) (*wedge3_fan x V E v u i*)))) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%348*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%348* ((*0*::*nat*) ≤ *i* ∧ *i* < *CARD* (*set_of_edge v V E*)) (*wedge2_fan x V E v u i*)))))

thm Topology.aff_subset_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*) ⟶ *SUBSET* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*))

thm Topology.eq_set_wdart_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *GSPEC* (λ*GEN%PVAR%351*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%351* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*))) = *GSPEC* (λ*GEN%PVAR%352*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%352* ((*0*::*nat*) ≤ *i* ∧ *i* < *CARD* (*set_of_edge v V E*)) (*wedge3_fan x V E v u i*))

thm Topology.eq_set_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *GSPEC* (λ*GEN%PVAR%353*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%353* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*))) = *GSPEC* (λ*GEN%PVAR%354*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%354* ((*0*::*nat*) ≤ *i* ∧ *i* < *CARD* (*set_of_edge v V E*)) (*wedge2_fan x V E v u i*))

thm Topology.UNION1_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *HOL_Light_Import*.*UNIV* = *HOL_Light_Import*.*UNION*

$(aff\ (INSERT\ x\ (INSERT\ v\ EMPTY)))\ (HOL\_Light\_Import.UNION\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%355::(real,\ 3)\ cart \Rightarrow bool.\ \exists\, w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%355\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)\ (w\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w)))))\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%356::(real,\ 3)\ cart \Rightarrow bool.\ \exists\, w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%356\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY))))))$

thm Topology.CARD_SING:

$\forall (x::(real,\ 3)\ cart)\ s::(real,\ 3)\ cart \Rightarrow bool.\ FINITE\ s \wedge s = INSERT\ x\ EMPTY \longrightarrow CARD\ s = (1::nat)$

thm Topology.disjoint_set_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w1\ EMPTY))\ E \longrightarrow HOL\_Light\_Import.INTER\ (w\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w))\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w1\ EMPTY)) = EMPTY$

thm Topology.disjiont1_cor6dot1:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ i::nat.\ HOL\_Light\_Import.INTER\ (wedge3\_fan\ x\ V\ E\ v\ u\ i)\ (aff\ (INSERT\ x\ (INSERT\ v\ EMPTY))) = EMPTY$

thm Topology.disjoint_fan1:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow HOL\_Light\_Import.INTER\ (w\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w))\ (aff\ (INSERT\ x\ (INSERT\ v\ EMPTY))) = EMPTY$

thm Topology.disjoint_fan2:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w1\ EMPTY))\ E \wedge w \neq w1 \longrightarrow HOL\_Light\_Import.INTER\ (w\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w))\ (w\_dart\_fan\ x\ V\ E\ (x,\ v,\ w1,\ sigma\_fan\ x\ V\ E\ v\ w1)) = EMPTY$

thm Topology.disjoint_fan3:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow HOL\_Light\_Import.INTER\ (aff\ (INSERT\ x\ (INSERT\ v\ EMPTY)))\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w\ EMPTY)) = EMPTY$

thm Topology.remark3_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ∧ *w* ≠ *w1* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w1 EMPTY*)) = *EMPTY*

thm Topology.VBTIKLP:

(∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *HOL_Light_Import.UNIV* = *HOL_Light_Import.UNION* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*HOL_Light_Import.UNION* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%357*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%357* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*))))) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%358*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%358* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*))))))) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *HOL_Light_Import.INTER* (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*)) (*aff* (*INSERT x* (*INSERT v EMPTY*))) = *EMPTY*) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ⟶ *HOL_Light_Import.INTER* (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*)) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w1 EMPTY*)) = *EMPTY*) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ∧ *w* ≠ *w1* ⟶ *HOL_Light_Import.INTER* (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*)) (*w_dart_fan x V E* (*x*, *v*, *w1*, *sigma_fan x V E v w1*)) = *EMPTY*) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ∧ *w* ≠ *w1* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w1 EMPTY*)) = *EMPTY*) ∧ (∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *HOL_Light_Import.INTER* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) = *EMPTY*)

thm Topology.disjiont_union_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒

bool) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ⟶ *HOL_Light_Import.INTER* (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*)) (*HOL_Light_Import.UNION* (*aff* (*INSERT x* (*INSERT v EMPTY*))) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w1 EMPTY*))) = *EMPTY*

thm Topology.aff_ge_subset_aff_gt_union_aff:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*HOL_Light_Import.UNION* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff* (*INSERT x* (*INSERT v EMPTY*))))

thm Topology.IBZWFFH:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ⟶ *HOL_Light_Import.INTER* (*w_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w1 EMPTY*))) = *EMPTY*

thm Topology.aff_ge_inter_aff_ge:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ⟶ *aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) = *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff_ge* (*INSERT x* (*INSERT w EMPTY*)) (*INSERT v EMPTY*))

thm Topology.rcone_fan:

∀(*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *h*::*real*. *rcone_fan x v h* = *GSPEC* (λ*GEN%PVAR%359*::(*real*, *3*) *cart*. ∃*y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%359* (*distance* (*y*, *x*) ∗ (*distance* (*v*, *x*) ∗ *h*) < *dot* (*vector_sub y x*) (*vector_sub v x*)) *y*)

thm Topology.exp_aff_ge_by_dot:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*) = *GSPEC* (λ*GEN%PVAR%360*::(*real*, *3*) *cart*. ∃*w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%360* (*dot* (*vector_sub w x*) (*e2_fan x v u*) = (*0*::*real*) ∧ (*0*::*real*) ≤ *dot* (*vector_sub w x*) (*e1_fan x v u*)) *w*)

thm Topology.closed_aff_ge_2_1:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *HOL_Light_Import.closed* (*aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*))

thm Topology.closed_aff_ge_1_2:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \longrightarrow HOL\_Light\_Import.closed\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Topology.AFF_GE_1_1:

$\forall (x::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ x \neq v \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%366::(real,\ 3)\ cart.\ \exists\ y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%366\ (\exists (t1::real)\ t2::real.\ (0::real) \leq t2 \wedge t1 + t2 = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (\%\ t2\ v))\ y)$

thm Topology.exp_aff_ge_by_dot_1_1:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%367::(real,\ 3)\ cart.\ \exists\ w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%367\ (dot\ (vector\_sub\ w\ x)\ (e2\_fan\ x\ v\ u) = (0::real) \wedge (0::real) \leq dot\ (vector\_sub\ w\ x)\ (e3\_fan\ x\ v\ u) \wedge dot\ (vector\_sub\ w\ x)\ (e1\_fan\ x\ v\ u) = (0::real))\ w)$

thm Topology.closed_halfline_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow HOL\_Light\_Import.closed\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY))$

thm DEF_ballnorm_fan:

$ballnorm\_fan = (\lambda\_2552952::(real,\ 3)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%377::(real,\ 3)\ cart.\ \exists\ y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%377\ (distance\ (\_2552952,\ y) = (1::real))\ y))$

thm Topology.ballnorm_fan:

$\forall\ x::(real,\ 3)\ cart.\ ballnorm\_fan\ x = GSPEC\ (\lambda GEN\%PVAR\%377::(real,\ 3)\ cart.\ \exists\ y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%377\ (distance\ (x,\ y) = (1::real))\ y)$

thm Topology.closed_ballnorm_fan:

$\forall\ x::(real,\ 3)\ cart.\ HOL\_Light\_Import.closed\ (ballnorm\_fan\ x)$

thm Topology.bounded_ballnorm_fan:

$\forall\ x::(real,\ 3)\ cart.\ bounded\ (ballnorm\_fan\ x)$

thm Topology.bounded_ballnorm_fans:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ bounded\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (ballnorm\_fan\ x))$

thm Topology.closed_aff_ge_ballnorm_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \longrightarrow HOL\_Light\_Import.closed\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (ballnorm\_fan\ x))$

thm Topology.compact_aff_ge_ballnorm_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \longrightarrow compact\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (ballnorm\_fan\ x))$

thm Topology.closed_point_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow HOL\_Light\_Import.closed\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY))\ (ballnorm\_fan\ x))$

thm Topology.exist_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge \neg\ IN\ v\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)) \wedge IN\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow (\exists h{>}0::real.\ \forall (y1::(real,\ 3)\ cart)\ y2::(real,\ 3)\ cart.\ IN\ y1\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY))\ (ballnorm\_fan\ x)) \wedge IN\ y2\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)))\ (ballnorm\_fan\ x)) \longrightarrow h \le distance\ (y1,\ y2))$

thm DEF_ballsets_fan:

$ballsets\_fan = (\lambda(\_2555035::(real,\ 3)\ cart \Rightarrow bool)\ \_2555036::real.\ GSPEC\ (\lambda GEN\%PVAR\%379::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%379\ (\exists x::(real,\ 3)\ cart.\ distance\ (x,\ y) < \_2555036 \wedge IN\ x\ \_2555035)\ y))$

thm Topology.ballsets_fan:

$\forall (h::real)\ s::(real,\ 3)\ cart \Rightarrow bool.\ ballsets\_fan\ s\ h = GSPEC\ (\lambda GEN\%PVAR\%379::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%379\ (\exists x::(real,\ 3)\ cart.\ distance\ (x,\ y) < h \wedge IN\ x\ s)\ y)$

thm Topology.exists_ballsets_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge \neg\ IN\ v\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)) \wedge IN\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow (\exists h{>}0::real.\ HOL\_Light\_Import.INTER\ (ballsets\_fan\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ EMPTY))\ (ballnorm\_fan\ x))\ h)\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)))\ (ballnorm\_fan\ x)) = EMPTY)$

thm DEF_cone_ge_fan:

cone_ge_fan = (λ(*_2555047*::(*real*, *3*) *cart*) *_2555048*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%380*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%380* (∃ (*a*::*real*) *z*::(*real*, *3*) *cart*. (*0*::*real*) ≤ *a* ∧ *IN* *z* *_2555048* ∧ *y* = *vector_add* (% *a* (*vector_sub* *z* *_2555047*)) *_2555047*) *y*))

thm Topology.cone_ge_fan:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) *x*::(*real*, *3*) *cart*. *cone_ge_fan* *x* *s* = *GSPEC* (λ*GEN%PVAR%380*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%380* (∃ (*a*::*real*) *z*::(*real*, *3*) *cart*. (*0*::*real*) ≤ *a* ∧ *IN* *z* *s* ∧ *y* = *vector_add* (% *a* (*vector_sub* *z* *x*)) *x*) *y*)

thm Topology.cone_ge_fan_inter_aff_ge_is_empty:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ ¬ *IN* *v* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) ∧ *IN* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v* (*INSERT* *w* *EMPTY*)) *E* ⟶ (∃ *h*>*0*::*real*. *HOL_Light_Import*.*INTER* (*cone_ge_fan* *x* (*HOL_Light_Import*.*INTER* (*ballsets_fan* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*)) (*ballnorm_fan* *x*)) *h*) (*ballnorm_fan* *x*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *w1* *EMPTY*))) = *INSERT* *x* *EMPTY*)

thm Topology.subset_by_inequality_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*h*::*real*) *h1*::*real*. *FAN* (*x*, *V*, *E*) ∧ ¬ *IN* *v* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) ∧ *IN* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v* (*INSERT* *w* *EMPTY*)) *E* ∧ *h* < *h1* ⟶ *SUBSET* (*HOL_Light_Import*.*INTER* (*cone_ge_fan* *x* (*HOL_Light_Import*.*INTER* (*ballsets_fan* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*)) (*ballnorm_fan* *x*)) *h*) (*ballnorm_fan* *x*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)))) (*HOL_Light_Import*.*INTER* (*cone_ge_fan* *x* (*HOL_Light_Import*.*INTER* (*ballsets_fan* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*)) (*ballnorm_fan* *x*)) *h1*) (*ballnorm_fan* *x*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *w1* *EMPTY*))))

thm Topology.cone_ge_fan_inter_aff_ge_is_empty_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ ¬ *IN* *v* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) ∧ *IN* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v* (*INSERT* *w* *EMPTY*)) *E* ⟶ (∃ *h*<*1*::*real*. (*0*::*real*) < *h* ∧ *SUBSET* (*HOL_Light_Import*.*INTER* (*cone_ge_fan* *x* (*HOL_Light_Import*.*INTER* (*ballsets_fan* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*)) (*ballnorm_fan* *x*)) *h*) (*ballnorm_fan* *x*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)))) (*INSERT* *x* *EMPTY*))

thm Topology.rcone_subset_cone:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *h*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ (*0*::*real*) < *h* ∧ *h* < (*1*::*real*) ⟶ (∃ *h1*<*1*::*real*. (*0*::*real*) < *h1* ∧ *SUBSET* (*rcone_fan x v h1*) (*cone_ge_fan x* (*HOL_Light_Import.INTER* (*ballsets_fan* (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)) (*ballnorm_fan x*)) *h*) (*ballnorm_fan x*))))

thm Topology.origin_not_in_rcone_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *h*::*real*. ¬ *IN x* (*rcone_fan x v h*)

thm Topology.inter_is_empty:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ ¬ *IN v* (*INSERT v1* (*INSERT w1 EMPTY*)) ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ (∃ *h1*<*1*::*real*. (*0*::*real*) < *h1* ∧ *HOL_Light_Import.INTER* (*rcone_fan x v h1*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*))) = *EMPTY*)

thm Topology.avoids_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ ¬ *IN v* (*INSERT v1* (*INSERT w1 EMPTY*)) ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ (∃ *h*<*1*::*real*. (*0*::*real*) < *h* ∧ *HOL_Light_Import.INTER* (*rw_dart_fan x V E* (*x*, *v*, *w*, *w2*) *h*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*))) = *EMPTY*)

thm Topology.avoids1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ⟶ (∃ *h*<*1*::*real*. (*0*::*real*) < *h* ∧ *HOL_Light_Import.INTER* (*rw_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*) *h*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w1 EMPTY*))) = *EMPTY*)

thm Topology.finish_avoids_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ⟶ (∃ *h*<*1*::*real*. (*0*::*real*) < *h* ∧ *HOL_Light_Import.INTER* (*rw_dart_fan x V E* (*x*, *v*, *w*, *sigma_fan x V E v w*) *h*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*))) = *EMPTY*)

thm Topology.continuous_set_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (h::real)\ h1::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge h1 \leq h \longrightarrow SUBSET\ (rw\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w)\ h)\ (rw\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w)\ h1)$

thm Topology.CTVTAQA:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge SUBSET\ E1\ E \longrightarrow FAN\ (x,\ V,\ E1)$

thm Topology.expand_edge_graph_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ e::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge IN\ e\ E \longrightarrow (\exists (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ e = INSERT\ v\ (INSERT\ w\ EMPTY))$

thm Topology.finish_avoids1_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E'::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge SUBSET\ E'\ E \longrightarrow (\exists h<1::real.\ (0::real) < h \wedge HOL\_Light\_Import.INTER\ (rw\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w)\ h)\ (GSPEC\ (\lambda GEN\%PVAR\%381::(real,\ 3)\ cart.\ \exists v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%381\ (\exists e::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e\ E' \wedge IN\ v\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e))\ v)) = EMPTY)$

thm Topology.rw_dart_avoids_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow (\exists h<1::real.\ (0::real) < h \wedge SUBSET\ (rw\_dart\_fan\ x\ V\ E\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E\ v\ w)\ h)\ (yfan\ (x,\ V,\ E)))$

thm DEF_r_fan:

$r\_fan = (\lambda(\_2569933::real)\ (\_2569934::real)\ \_2569935::real.\ GSPEC\ (\lambda GEN\%PVAR\%382::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%382\ ((0::real) < \$\ y\ (1::nat) \wedge \_2569933 < \$\ y\ (2::nat) \wedge \$\ y\ (2::nat) < \_2569934 \wedge (0::real) < \$\ y\ (3::nat) \wedge \$\ y\ (3::nat) < \_2569935)\ y))$

thm Topology.r_fan:

$\forall (a::real)\ (b::real)\ c::real.\ r\_fan\ a\ b\ c = GSPEC\ (\lambda GEN\%PVAR\%382::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%382\ ((0::real) < \$\ y\ (1::nat) \wedge a < \$\ y\ (2::nat) \wedge \$\ y\ (2::nat) < b \wedge (0::real) < \$\ y\ (3::nat) \wedge \$\ y\ (3::nat) < c)\ y)$

thm DEF_r1_le_fan:

r1_le_fan = (λ*_2569954*::*real*. *GSPEC* (λ*GEN%PVAR%383*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%383* (*_2569954* < $ *y* (*1*::*nat*)) *y*))

thm Topology.r1_le_fan:

∀ *a*::*real*. *r1_le_fan a* = *GSPEC* (λ*GEN%PVAR%383*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%383* (*a* < $ *y* (*1*::*nat*)) *y*)

thm DEF_r2_le_fan:

r2_le_fan = (λ*_2569959*::*real*. *GSPEC* (λ*GEN%PVAR%384*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%384* (*_2569959* < $ *y* (*2*::*nat*)) *y*))

thm Topology.r2_le_fan:

∀ *a*::*real*. *r2_le_fan a* = *GSPEC* (λ*GEN%PVAR%384*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%384* (*a* < $ *y* (*2*::*nat*)) *y*)

thm DEF_r3_le_fan:

r3_le_fan = (λ*_2569964*::*real*. *GSPEC* (λ*GEN%PVAR%385*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%385* (*_2569964* < $ *y* (*3*::*nat*)) *y*))

thm Topology.r3_le_fan:

∀ *a*::*real*. *r3_le_fan a* = *GSPEC* (λ*GEN%PVAR%385*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%385* (*a* < $ *y* (*3*::*nat*)) *y*)

thm DEF_r1_ge_fan:

r1_ge_fan = (λ*_2569969*::*real*. *GSPEC* (λ*GEN%PVAR%386*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%386* ($ *y* (*1*::*nat*) < *_2569969*) *y*))

thm Topology.r1_ge_fan:

∀ *a*::*real*. *r1_ge_fan a* = *GSPEC* (λ*GEN%PVAR%386*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%386* ($ *y* (*1*::*nat*) < *a*) *y*)

thm DEF_r2_ge_fan:

r2_ge_fan = (λ*_2569974*::*real*. *GSPEC* (λ*GEN%PVAR%387*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%387* ($ *y* (*2*::*nat*) < *_2569974*) *y*))

thm Topology.r2_ge_fan:

∀ *a*::*real*. *r2_ge_fan a* = *GSPEC* (λ*GEN%PVAR%387*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%387* ($ *y* (*2*::*nat*) < *a*) *y*)

thm DEF_r3_ge_fan:

r3_ge_fan = (λ*_2569979*::*real*. *GSPEC* (λ*GEN%PVAR%388*::(*real*, *3*) *cart*. $\exists$ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%388* ($ *y* (*3*::*nat*) < *_2569979*) *y*))

thm Topology.r3_ge_fan:

$\forall$ *a*::*real*. *r3_ge_fan a* = *GSPEC* (λ*GEN%PVAR%388*::(*real*, *3*) *cart*. $\exists$ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%388* ($ *y* (*3*::*nat*) < *a*) *y*)

thm Topology.r_fan_is_inter_halfspace:

$\forall$ (*a*::*real*) (*b*::*real*) *c*::*real*. *r_fan a b c* = *HOL_Light_Import*.*INTER* (*r1_le_fan* (*0*::*real*)) (*HOL_Light_Import*.*INTER* (*r2_le_fan a*) (*HOL_Light_Import*.*INTER* (*r2_ge_fan b*) (*HOL_Light_Import*.*INTER* (*r3_le_fan* (*0*::*real*)) (*r3_ge_fan c*))))

thm Topology.r1_ge_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r1_ge_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r1_ge_fan a*)

thm Topology.r2_ge_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r2_ge_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r2_ge_fan a*)

thm Topology.r3_ge_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r3_ge_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r3_ge_fan a*)

thm Topology.r1_le_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r1_le_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r1_le_fan a*)

thm Topology.r2_le_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r2_le_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r2_le_fan a*)

thm Topology.r3_le_is_convex_fan:

$\forall$ *a*::*real*. *convex* (*r3_le_fan a*) $\wedge$ *HOL_Light_Import*.*open* (*r3_le_fan a*)

thm Topology.r_is_connected_fan:

$\forall$ (*a*::*real*) (*b*::*real*) *c*::*real*. *connected* (*r_fan a b c*) $\wedge$ *convex* (*r_fan a b c*) $\wedge$ *HOL_Light_Import*.*open* (*r_fan a b c*)

thm DEF_change_spherical_coordinate_fan:

change_spherical_coordinate_fan = (λ(*_2570105*::(*real*, *3*) *cart*) (*_2570106*::(*real*, *3*) *cart*) (*_2570107*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *LET* (λ(*r*::*real*) (*theta*::*real*) *phi*::*real*. *LET_END* (*vector_add _2570105* (*vector_add* (% (*r* ∗ (*cos theta* ∗ *sin phi*)) (*e1_fan _2570105 _2570106 _2570107*)) (*vector_add* (% (*r* ∗ (*sin theta* ∗ *sin phi*)) (*e2_fan _2570105 _2570106 _2570107*)) (% (*r* ∗ *cos phi*) (*e3_fan _2570105 _2570106 _2570107*)))))) ($ *t* (*1*::*nat*)) ($ *t* (*2*::*nat*)) ($ *t* (*3*::*nat*)))

thm Topology.change_spherical_coordinate_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *change_spherical_coordinate_fan x v u* = (λ*t*::(*real*, *3*) *cart*. *LET* (λ(*r*::*real*) (*theta*::*real*) *phi*::*real*. *LET_END* (*vector_add x* (*vector_add* (% (*r* ∗ (*cos theta* ∗ *sin phi*)) (*e1_fan x v u*)) (*vector_add* (% (*r* ∗ (*sin theta* ∗ *sin phi*)) (*e2_fan x v u*)) (% (*r* ∗ *cos phi*) (*e3_fan x v u*)))))) ($ *t* (*1*::*nat*)) ($ *t* (*2*::*nat*)) ($ *t* (*3*::*nat*)))

thm Topology.continuous_change_spherical_coordinate_fan:

∀ (*x'*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *continuous* (λ*t*::(*real*, *3*) *cart*. *LET* (λ(*r*::*real*) (*theta*::*real*) *phi*::*real*. *LET_END* (*vector_add* (% (*r* ∗ (*cos theta* ∗ *sin phi*)) (*e1_fan x' v u*)) (*vector_add* (% (*r* ∗ (*sin theta* ∗ *sin phi*)) (*e2_fan x' v u*)) (% (*r* ∗ *cos phi*) (*e3_fan x' v u*))))) ($ *t* (*1*::*nat*)) ($ *t* (*2*::*nat*)) ($ *t* (*3*::*nat*))) (*at x*)

thm Topology.one_edge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ ¬ (*1*::*nat*) < *CARD* (*set_of_edge v V E*) ⟶ *set_of_edge v V E* = *INSERT u EMPTY*

thm Topology.expand_elements_by_azim_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ (*0*::*real*) < *x1* ∧ (*0*::*real*) ≤ *x2* ∧ *x2* < *real_of_nat* (*2*::*nat*) ∗ *pi* ∧ (*0*::*real*) < *x3* ∧ *x3* < *pi* / *real_of_nat* (*2*::*nat*) ⟶ *azim x v u* (*vector_add x* (*vector_add* (% (*x1* ∗ (*cos x2* ∗ *sin x3*)) (*e1_fan x v u*)) (*vector_add* (% (*x1* ∗ (*sin x2* ∗ *sin x3*)) (*e2_fan x v u*)) (% (*x1* ∗ *cos x3*) (*e3_fan x v u*))))) = *x2*

thm Topology.rw_dart_is_image_set_spherical_coordinate:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *h*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ (*0*::*real*) < *h* ∧ *h* < *pi* / *real_of_nat* (*2*::*nat*) ⟶ *IMAGE* (*change_spherical_coordinate_fan x v u*) (*r_fan* (*azim x v u u*) (*azim_fan x V E v u*) *h*) = *rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*)

thm Topology.connected_rw_dart_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *h*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ (*0*::*real*) < *h* ∧ *h* < *pi* / *real_of_nat* (*2*::*nat*) ⟶ *connected* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*))

thm Topology.not_empty_rw_dart_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *pi* / *real_of_nat*

(*2*::*nat*) ⟶ *rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*) ≠ *EMPTY*)

thm Topology.JGIYDLE:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ (∀*h*::*real*. (*0*::*real*) < *h* ∧ *h* < *pi* / *real_of_nat* (*2*::*nat*) ⟶ *rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*) ≠ *EMPTY*) ∧ (∀(*h*::*real*) *h1*::*real*. *h1* ≤ *h* ⟶ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) *h*) (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) *h1*)) ∧ (∃*h*<*1*::*real*. (*0*::*real*) < *h* ∧ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) *h*) (*yfan* (*x*, *V*, *E*))) ∧ (∀*h*::*real*. (*0*::*real*) < *h* ∧ *h* < *pi* / *real_of_nat* (*2*::*nat*) ⟶ *connected* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*)))

thm DEF_dart_leads_into:

dart_leads_into = (λ(*_2573823*::(*real*, *3*) *cart*) (*_2573824*::(*real*, *3*) *cart* ⇒ *bool*) (*_2573825*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*_2573826*::(*real*, *3*) *cart*) *_2573827*::(*real*, *3*) *cart*. *SOME U*::(*real*, *3*) *cart* ⇒ *bool*. ∃*h*>*0*::*real*. ∀(*s*::*real*) *y*::(*real*, *3*) *cart*. (*0*::*real*) < *s* ∧ *s* < *h* ∧ *IN y* (*rw_dart_fan _2573823 _2573824 _2573825* (*_2573823*, *_2573826*, *_2573827*, *sigma_fan _2573823 _2573824 _2573825 _2573826 _2573827*) (*cos s*)) ⟶ *SUBSET* (*rw_dart_fan _2573823 _2573824 _2573825* (*_2573823*, *_2573826*, *_2573827*, *sigma_fan _2573823 _2573824 _2573825 _2573826 _2573827*) (*cos s*)) *U* ∧ *connected_component* (*yfan* (*_2573823*, *_2573824*, *_2573825*)) *y* = *U*)

thm Topology.dart_leads_into:

∀(*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *dart_leads_into x V E v u* = (*SOME U*::(*real*, *3*) *cart* ⇒ *bool*. ∃*h*>*0*::*real*. ∀(*s*::*real*) *y*::(*real*, *3*) *cart*. (*0*::*real*) < *s* ∧ *s* < *h* ∧ *IN y* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) ⟶ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) *U* ∧ *connected_component* (*yfan* (*x*, *V*, *E*)) *y* = *U*)

thm Topology.exists_leads_into_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ (∃(*U*::(*real*, *3*) *cart* ⇒ *bool*) *h*::*real*. (*0*::*real*) < *h* ∧ (∀(*s*::*real*) *y*::(*real*, *3*) *cart*. (*0*::*real*) < *s* ∧ *s* < *h* ∧ *IN y* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) ⟶ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) *U* ∧ *connected_component* (*yfan* (*x*, *V*, *E*)) *y* = *U*))

thm Topology.DART_LEADS_INTO:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v*

(*INSERT u EMPTY*)) *E* ⟶ (∃ *h*>*0*::*real*. ∀ (*s*::*real*) *y*::(*real*, *3*) *cart*. (*0*::*real*) < *s* ∧ *s* < *h* ∧ *IN y* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) ⟶ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) (*dart_leads_into x V E v u*) ∧ *connected_component* (*yfan* (*x*, *V*, *E*)) *y* = *dart_leads_into x V E v u*)

thm Topology.unique_dart_leads_into:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ (∃ *h*>*0*::*real*. ∀ (*s*::*real*) *y*::(*real*, *3*) *cart*. (*0*::*real*) < *s* ∧ *s* < *h* ∧ *IN y* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) ⟶ *SUBSET* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos s*)) *U* ∧ *connected_component* (*yfan* (*x*, *V*, *E*)) *y* = *U*) ⟶ *dart_leads_into x V E v u* = *U*

thm Topology.dart_leads_into_fan_in_topological_component_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *IN* (*dart_leads_into x V E v u*) (*topological_component_yfan* (*x*, *V*, *E*))

thm Topology.in_topological_component_yfan_is_connected:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ⟶ *connected U*

thm Topology.connected_dart_leads_into_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ *connected* (*dart_leads_into x V E v u*)

thm Fan_misc.dart1_of_fan:

∀ (*V*::*?'a*::*type* ⇒ *bool*) *E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *dart1_of_fan* (*V*, *E*) = *GSPEC* (λ*GEN%PVAR%389*::*?'a*::*type* × *?'a*::*type*. ∃ (*v*::*?'a*::*type*) *w*::*?'a*::*type*. *SETSPEC GEN%PVAR%389* (*IN* (*INSERT v* (*INSERT w EMPTY*)) *E*) (*v*, *w*))

thm Fan_misc.EXTENSION_SIGMA_FAN_EQ_RES:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *v*::(*real*, *3*) *cart*. *extension_sigma_fan x V E v* = *res* (*sigma_fan x V E v*) (*set_of_edge v V E*)

thm Fan_misc.INVERSE_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ⟶ *extension_sigma_fan x V E v* ∘

inverse_sigma_fan x V E v = id ∧ inverse_sigma_fan x V E v ∘ extension_sigma_fan x V E v = id

thm Fan_misc.EXTENSION_SIGMA_FAN_INJECTIVE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ⟶ (∀ (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *extension_sigma_fan x V E v u* = *extension_sigma_fan x V E v w* ⟶ *u* = *w*)

thm Fan_misc.IN_SET_OF_EDGE:

∀ (*V*::*?'a*::*type* ⇒ *bool*) (*E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*v*::*?'a*::*type*) *w*::*?'a*::*type*. *SUBSET* (*UNIONS E*) *V* ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *IN v V* ∧ *IN w V* ∧ *IN w* (*set_of_edge v V E*) ∧ *IN v* (*set_of_edge w V E*)

thm Fan_misc.FAN_IN_SET_OF_EDGE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *IN v V* ∧ *IN w V* ∧ *IN w* (*set_of_edge v V E*) ∧ *IN v* (*set_of_edge w V E*)

thm Fan_misc.INVERSE_SIGMA_FAN_EQ_INVERSE1_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *inverse1_sigma_fan x V E v w* = *inverse_sigma_fan x V E v w*

thm Planarity.collinear_continuous_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *c*::*real*. *continuous_on* (λ*t*::(*real*, *unit*) *cart*. *vector_add* (% ((*1*::*real*) − *HOL_Light_Import.drop t*) *u*) (*vector_sub* (*vector_sub* (% (*HOL_Light_Import.drop t*) *w*) (% ((*1*::*real*) − *c*) *x*)) (% *c v*))) *HOL_Light_Import.UNIV*

thm Planarity.collinear1_continuous_fan:

∀ (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *t*::(*real*, *unit*) *cart*. *continuous* (λ*t*::(*real*, *unit*) *cart*. *vector_add* (% ((*1*::*real*) − *HOL_Light_Import.drop t*) *u*) (% (*HOL_Light_Import.drop t*) *w*)) (*at t*)

thm Planarity.CONTINUOUS_CLOSED_PREIMAGE_CONSTANT:

∀ (*f*::(*real*, *?'b*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous_on f s* ∧ *HOL_Light_Import.closed s* ⟶ *HOL_Light_Import.closed* (*GSPEC* (λ*GEN%PVAR%393*::(*real*, *?'b*::*type*) *cart*. ∃ *x*::(*real*, *?'b*::*type*) *cart*. *SETSPEC GEN%PVAR%393* (*IN x s* ∧ *f x* = *a*) *x*))

thm Planarity.open_collinear_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *c*::*real*. *HOL_Light_Import.open* (*GSPEC* (λ*GEN%PVAR%394*::(*real*, *unit*) *cart*. ∃ *t*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%394* (*vector_add* (% ((*1*::*real*) − *HOL_Light_Import*.*drop t*) *u*) (*vector_sub* (*vector_sub* (% (*HOL_Light_Import*.*drop t*) *w*) (% ((*1*::*real*) − *c*) *x*)) (% *c v*)) ≠ *vec* (*0*::*nat*)) *t*))

thm Planarity.open_vector_angle_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*c*::*real*) *a*::*real*. (∀ *t*::*real*. *vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*) ≠ *x*) ⟶ *HOL_Light_Import*.*open* (*GSPEC* (λ*GEN%PVAR%395*::(*real*, *unit*) *cart*. ∃ *t*::(*real*, *unit*) *cart*. *SETSPEC GEN%PVAR%395* (*vector_angle* (*vector_sub v x*) (*vector_sub* (*vector_add* (% ((*1*::*real*) − *HOL_Light_Import*.*drop t*) *u*) (% (*HOL_Light_Import*.*drop t*) *w*)) *x*) ≠ *a*) *t*))

thm Planarity.exists_open_not_collinear:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ⟶ (∃ *t1*>*0*::*real*. *t1* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) ≤ *t* ∧ *t* ≤ *t1* ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*)))))

thm Planarity.exist_close_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import*.*INTER* (*INSERT v* (*INSERT w EMPTY*)) (*INSERT v1* (*INSERT w1 EMPTY*)) = *EMPTY* ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ (∃ *h*>*0*::*real*. ∀ (*y1*::(*real*, *3*) *cart*) *y2*::(*real*, *3*) *cart*. *IN y1* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*ballnorm_fan x*)) ∧ *IN y2* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*))) (*ballnorm_fan x*)) ⟶ *h* ≤ *distance* (*y1*, *y2*))

thm Planarity.AFF_GT_1_2:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) ⟶ *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) = *GSPEC* (λ*GEN%PVAR%399*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%399* (∃ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t2* ∧ (*0*::*real*) < *t3* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 v*) (% *t3 w*))) *y*)

thm Planarity.linear_aff_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *linear* (λ*t*::(*real*, *2*) *cart*. *vector_add* (% ($ *t* (*1*::*nat*)) (*vector_sub v x*)) (% ($ *t* (*2*::*nat*)) (*vector_sub u x*)))

thm Planarity.linear1_aff_fan:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. linear (λt::(real, 3) cart. vector_add (% ($ t (1::nat)) (vector_sub v x)) (vector_add (% ($ t (2::nat)) (vector_sub u x)) (% ($ t (3::nat)) (vector_sub w u))))

thm Planarity.linear_inj_fan:

∀ (x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ (∀ (a::(real, 2) cart) b::(real, 2) cart. vector_add (% ($ a (1::nat)) (vector_sub v x)) (% ($ a (2::nat)) (vector_sub u x)) = vector_add (% ($ b (1::nat)) (vector_sub v x)) (% ($ b (2::nat)) (vector_sub u x)) ⟶ a = b)

thm Planarity.origin_point_not_in_convex_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ⟶ ¬ IN x (hull convex (INSERT v (INSERT u (INSERT w EMPTY))))

thm Planarity.separate_point_convex_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ⟶ (∃ h>0::real. ∀ y::(real, 3) cart. IN y (hull convex (INSERT v (INSERT u (INSERT w EMPTY)))) ⟶ h < vector_norm (vector_sub y x))

thm Planarity.expansion_convex_fan:

∀ (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (t::real) s::real. (0::real) ≤ t ∧ t ≤ (1::real) ∧ (0::real) ≤ s ∧ s ≤ (1::real) ⟶ IN (vector_add (% ((1::real) − s) v) (% s (vector_add (% ((1::real) − t) u) (% t w)))) (hull convex (INSERT v (INSERT u (INSERT w EMPTY))))

thm Planarity.expansion1_convex_fan:

∀ (v::(real, 3) cart) (u::(real, 3) cart) s::real. (0::real) ≤ s ∧ s ≤ (1::real) ⟶ IN (vector_add (% ((1::real) − s) v) (% s u)) (hull convex (INSERT v (INSERT u EMPTY)))

thm Planarity.norm_origin_fan:

∀ x::(real, 3) cart. continuous_on (λy::(real, 3) cart. lift (vector_norm (vector_sub y x))) HOL_Light_Import.UNIV

thm Planarity.origin_point_not1_in_convex_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v

(*INSERT u EMPTY*)) *E* ⟶ ¬ *IN x* (*hull convex* (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.inequality1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *d*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *d* ⟶ (∃ *h*>*0*::*real*. *h* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) ≤ *t* ∧ *t* < *h* ⟶ (∀ *s*::*real*. (*0*::*real*) ≤ *s* ∧ *s* ≤ (*1*::*real*) ⟶ *s* ∗ (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*))) ∗ *vector_norm* (*vector_sub u* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)))) < *d*)))

thm Planarity.bounded_convex_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ⟶ (∃ *h*>*0*::*real*. ∀ *y*::(*real*, *3*) *cart*. *IN y* (*hull convex* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *vector_norm* (*vector_sub y x*) < *h*)

thm Planarity.REAL_ABS_SUB_NORM:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. |*vector_norm x* − *vector_norm y*| ≤ *vector_norm* (*vector_sub x y*)

thm Planarity.inequaility2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *d*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *d* ⟶ (∃ *h*>*0*::*real*. *h* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) ≤ *t* ∧ *t* < *h* ⟶ (∀ *s*::*real*. (*0*::*real*) ≤ *s* ∧ *s* ≤ (*1*::*real*) ⟶ *vector_norm* (*vector_sub* (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s u*) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s u*) *x*))) (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s u*) *x*)))) < *d*)))

thm Planarity.exists_point_small_edges_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *d*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *d* ⟶ (∃ *h*>*0*::*real*. *h* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) ≤ *t* ∧ *t* < *h* ⟶ (∀ *s*::*real*. (*0*::*real*) ≤ *s* ∧ *s* ≤ (*1*::*real*) ⟶ *vector_norm* (*vector_sub* (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s u*) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s u*) *x*))) (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*))) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) < *d*)))

thm Planarity.same_projective_sphere_ge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*t*::*real*) *y1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) ∧ *y1* ≠ *x* ∧ *IN y1* (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) (*ballnorm_fan* *x*)) ⟶ (∃ *s*≥*0*::*real*. *s* ≤ (*1*::*real*) ∧ *y1* = *vector_add* (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*))) *x*)

thm Planarity.same_projective_sphere_gt_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*t*::*real*) *y1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) ∧ *IN y1* (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) (*ballnorm_fan x*)) ⟶ (∃ *s*≥*0*::*real*. *s* ≤ (*1*::*real*) ∧ *y1* = *vector_add* (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*))) *x*)

thm Planarity.separate1_sphere_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*INSERT v* (*INSERT u EMPTY*)) (*INSERT v1* (*INSERT u1 EMPTY*)) = *EMPTY* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v1* (*INSERT u1 EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ (∃ *h*>*0*::*real*. *h* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < *h* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u1 EMPTY*))) (*ballnorm_fan x*)) = *EMPTY*))

thm Planarity.scale_aff_ge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ⟶ (∀ (*y*::(*real*, *3*) *cart*) *a*::*real*.

IN y (aff_ge (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ∧ (0::real) ≤ a ⟶ IN (vector_add (% a (vector_sub y x)) x) (aff_ge (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))))

thm Planarity.scale_aff_gt_fan:

∀(x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. DISJOINT (INSERT x EMPTY) (INSERT v (INSERT u EMPTY)) ⟶ (∀(y::(real, 3) cart) a::real. IN y (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ∧ (0::real) < a ⟶ IN (vector_add (% a (vector_sub y x)) x) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))))

thm Planarity.origin_is_not_aff_gt_fan:

∀(x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. ¬ IN u (aff (INSERT x (INSERT v EMPTY))) ∧ DISJOINT (INSERT x EMPTY) (INSERT v (INSERT u EMPTY)) ⟶ ¬ IN x (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY)))

thm Planarity.fan_run_in_small1_is_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (v1::(real, 3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ HOL_Light_Import.INTER (INSERT v (INSERT u EMPTY)) (INSERT v1 (INSERT u1 EMPTY)) = EMPTY ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT v1 (INSERT u1 EMPTY)) E ∧ ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ⟶ (∃t1>0::real. t1 ≤ (1::real) ∧ (∀t::real. (0::real) < t ∧ t < t1 ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − t) u) (% t w)) EMPTY))) (aff_ge (INSERT x EMPTY) (INSERT v1 (INSERT u1 EMPTY))) = EMPTY))

thm Planarity.azim_line_fan:

∀(x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) t::(real, unit) cart. ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT (vector_add (% ((1::real) − HOL_Light_Import.drop t) u) (% (HOL_Light_Import.drop t) w)) EMPTY)))) ⟶ real_continuous (λt::(real, unit) cart. azim x v u (vector_add (% ((1::real) − HOL_Light_Import.drop t) u) (% (HOL_Light_Import.drop t) w))) (at t)

thm DEF_fan81:

fan81 = (λ_2581825::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). ∀(v::(real, 3) cart) u::(real, 3) cart. IN (INSERT v (INSERT u EMPTY)) (snd (snd _2581825)) ⟶ azim_fan (fst _2581825) (fst (snd _2581825)) (snd (snd _2581825)) v u < pi)

thm Planarity.fan81:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ fan81\ (x,\ V,\ E) = (\forall (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \longrightarrow azim\_fan\ x\ V\ E\ v\ u < pi)$

thm DEF_fan80:

$fan80 = (\lambda\_2581838::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ (snd\ (snd\ \_2581838)) \longrightarrow (0::real) < azim\ (fst\ \_2581838)\ v\ u\ (sigma\_fan\ (fst\ \_2581838)\ (fst\ (snd\ \_2581838))\ (snd\ (snd\ \_2581838))\ v\ u) \wedge azim\ (fst\ \_2581838)\ v\ u\ (sigma\_fan\ (fst\ \_2581838)\ (fst\ (snd\ \_2581838))\ (snd\ (snd\ \_2581838))\ v\ u) < pi)$

thm Planarity.fan80:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ fan80\ (x,\ V,\ E) = (\forall (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \longrightarrow (0::real) < azim\ x\ v\ u\ (sigma\_fan\ x\ V\ E\ v\ u) \wedge azim\ x\ v\ u\ (sigma\_fan\ x\ V\ E\ v\ u) < pi)$

thm Planarity.continuous_coplanar_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow (\forall t::real.\ t \neq (0::real) \longrightarrow \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w))\ EMPTY)))))$

thm Planarity.open_is_not_zero_fan:

$HOL\_Light\_Import.open\ (GSPEC\ (\lambda GEN\%PVAR\%404::(real,\ unit)\ cart.\ \exists y::(real,\ unit)\ cart.\ SETSPEC\ GEN\%PVAR\%404\ (\exists x::real.\ x \neq (0::real) \wedge y = lift\ x)\ y))$

thm Planarity.azim_continuous_when_not_coplanar:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow real\_continuous\_on\ (\lambda t::real.\ azim\ x\ v\ u\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w)))\ (GSPEC\ (\lambda GEN\%PVAR\%406::real.\ \exists t::real.\ SETSPEC\ GEN\%PVAR\%406\ (t \neq (0::real))\ t))$

thm Planarity.injective_azim_coplanar:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow (\forall (a::real)\ b::real.\ a \neq (0::real) \wedge b \neq (0::real) \wedge azim\ x\ v\ u\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w)) = azim\ x\ v\ u\ (vector\_add\ (\%\ ((1::real) - b)\ u)\ (\%\ b\ w)) \longrightarrow a = b)$

thm Planarity.fan_run_in_small21_is_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E)$

∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v* (*INSERT* (*?w1.0*::(*real*, *3*) *cart*) *EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* (*INSERT* *w* *EMPTY*)))) ⟶ (∃ *t1*>*0*::*real*. *t1* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* ≤ *t1* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*)) *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *?w1.0* *EMPTY*))) = *EMPTY*))

thm Planarity.fan_run_in_small2_is_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v* (*INSERT* *w1* *EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* (*INSERT* *w* *EMPTY*)))) ⟶ (∃ *t1*>*0*::*real*. *t1* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < *t1* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*)) *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *w1* *EMPTY*))) = *EMPTY*))

thm Planarity.AFF_GT_2_2:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT* *x* (*INSERT* *u* *EMPTY*)) (*INSERT* *v* (*INSERT* *w* *EMPTY*)) ⟶ *aff_gt* (*INSERT* *x* (*INSERT* *u* *EMPTY*)) (*INSERT* *v* (*INSERT* *w* *EMPTY*)) = *GSPEC* (λ*GEN%PVAR%407*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%407* (∃ (*t1*::*real*) (*t2*::*real*) (*t3*::*real*) *t4*::*real*. (*0*::*real*) < *t3* ∧ (*0*::*real*) < *t4* ∧ *t1* + (*t2* + (*t3* + *t4*)) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1* *x*) (*vector_add* (% *t2* *u*) (*vector_add* (% *t3* *v*) (% *t4* *w*)))) *y*)

thm Planarity.extension_in_aff_2_2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *?E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *?E* ⟶ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ (∀ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t3* ∧ (*0*::*real*) < *t2* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ⟶ *IN* (*vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*))))) (*aff_gt* (*INSERT* *x* (*INSERT* *u* *EMPTY*)) (*INSERT* *w* (*INSERT* *v* *EMPTY*)))))

thm Planarity.inequality3_aim_in_convex_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ (*0*::*real*) < *azim* *x* *u* *w* *v* ∧ *azim* *x* *u* *w* *v* < *pi* ⟶ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ (∀ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t3* ∧ (*0*::*real*) < *t2* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ⟶ (*0*::*real*) < *azim* *x* *u* *w* (*vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*))))) ∧ *azim* *x* *u* *w* (*vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*))))) < *azim* *x* *u* *w* *v*))

thm Planarity.fan_run_in_small3_is_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w1* *EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* (*INSERT* *w* *EMPTY*)))) ∧ *sigma_fan* *x* *V* *E* *u* *w* = *v* ∧ (*0*::*real*) < *azim* *x* *u* *w* *v* ∧ *azim* *x* *u* *w* *v* < *pi* ⟶ (∃*t1*>*0*::*real*. *t1* ≤ (*1*::*real*) ∧ (∀*t*::*real*. (*0*::*real*) < *t* ∧ *t* < *t1* ⟶ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*)) *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *u* (*INSERT* *w1* *EMPTY*))) = *EMPTY*))

thm Planarity.properties_fully_surrounded:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ (*0*::*real*) < *azim* *x* *u* *w* *v* ∧ *azim* *x* *u* *w* *v* < *pi* ⟶ ¬ *coplanar* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* (*INSERT* *w* *EMPTY*))))

thm Planarity.fan_run_in_small_is_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *IN* (*INSERT* *v1* (*INSERT* *w1* *EMPTY*)) *E* ∧ (*0*::*real*) < *azim* *x* *u* *w* *v* ∧ *azim* *x* *u* *w* *v* < *pi* ∧ *sigma_fan* *x* *V* *E* *u* *w* = *v* ⟶ (∃*t1*>*0*::*real*. *t1* ≤ (*1*::*real*) ∧ (∀*t*::*real*. (*0*::*real*) < *t* ∧ *t* < *t1* ⟶ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t* *w*)) *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *w1* *EMPTY*))) = *EMPTY*))

thm Planarity.fan_run1_in_small_is_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E'*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *SUBSET* *E'* *E* ∧ (*0*::*real*) < *azim* *x* *u* *w* *v* ∧ *azim* *x* *u* *w* *v* < *pi* ∧ *sigma_fan* *x* *V* *E* *u* *w* = *v* ⟶ (∃*h*>*0*::*real*. *h* ≤ (*1*::*real*) ∧ (∀*s*::*real*. (*0*::*real*) < *s* ∧ *s* < *h* ⟶ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *s*) *u*) (% *s* *w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%408*::(*real*, *3*) *cart*. ∃*v*::(*real*, *3*) *cart*. *SETSPEC* *GEN%PVAR%408* (∃*e*::(*real*, *3*) *cart* ⇒ *bool*. *IN* *e* *E'* ∧ *IN* *v* (*aff_ge* (*INSERT* *x* *EMPTY*) *e*)) *v*)) = *EMPTY*))

thm Planarity.fan_run_in_small_is_not_meet_xfan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ (*0*::*real*) < *azim x u w v* $\wedge$ *azim x u w v* < *pi* $\wedge$ *sigma_fan x V E u w* = *v* $\longrightarrow$ ($\exists$ *h*>*0*::*real*. *h* $\leq$ (*1*::*real*) $\wedge$ ($\forall$ *s*::*real*. (*0*::*real*) < *s* $\wedge$ *s* < *h* $\longrightarrow$ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *s*) *u*) (% *s w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%409*::(*real*, *3*) *cart*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%409* ($\exists$ *e*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN e E* $\wedge$ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY*))

thm Planarity.fan_run_in_small_is_subset_yfan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ (*0*::*real*) < *azim x u w v* $\wedge$ *azim x u w v* < *pi* $\wedge$ *sigma_fan x V E u w* = *v* $\longrightarrow$ ($\exists$ *h*>*0*::*real*. *h* $\leq$ (*1*::*real*) $\wedge$ ($\forall$ *s*::*real*. (*0*::*real*) < *s* $\wedge$ *s* < *h* $\longrightarrow$ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *s*) *u*) (% *s w*)) *EMPTY*))) (*yfan* (*x*, *V*, *E*))))

thm Planarity.not_collinear_is_properties_fully_surrounded1:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *t*::*real*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ (*0*::*real*) < *azim x u w v* $\wedge$ *azim x u w v* < *pi* $\wedge$ (*0*::*real*) $\leq$ *t* $\wedge$ *t* $\leq$ (*1*::*real*) $\longrightarrow$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*)))

thm Planarity.not_collinear_is_properties_fully_surrounded:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *t*::*real*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ (*0*::*real*) < *azim x u w v* $\wedge$ *azim x u w v* < *pi* $\wedge$ (*0*::*real*) < *t* $\wedge$ *t* < (*1*::*real*) $\longrightarrow$ $\neg$ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*)))

thm Planarity.exists_inf_element_fix_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN v V* $\wedge$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*) $\longrightarrow$ ($\exists$ *u*::(*real*, *3*) *cart*. *IN u* (*set_of_edge v V E*) $\wedge$ ($\forall$ *w*::(*real*, *3*) *cart*. *IN w* (*set_of_edge v V E*) $\longrightarrow$ *azim x v u1 u* $\leq$ *azim x v u1 w*))

thm Planarity.exists_element_in_half_sapace_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*)

∧ *IN v V* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u1* (*INSERT w1* *EMPTY*)))) ∧ (*1*::*nat*) < *CARD* (*set_of_edge v V E*) ∧ *fan80* (*x*, *V*, *E*) ⟶ (∃ *u*::(*real*, *3*) *cart*. *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ (*0*::*real*) < *azim x v u1 u* ∧ *azim x v u1 u* < *pi*)

thm Planarity.JBDNJJB:

∀ (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT u* (*INSERT v EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT u* (*INSERT w EMPTY*))) ⟶ (∃ *t*>*0*::*real*. *sin* (*azim* (*vec* (*0*::*nat*)) *u v w*) = *t* ∗ *dot* (*cross u v*) *w*)

thm Planarity.independent_run_edges_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *fan80* (*x*, *V*, *E*) ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ⟶ *independent* (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub u x*) (*INSERT* (*vector_sub* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *x*) *EMPTY*)))

thm Planarity.span_run_edges_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *fan80* (*x*, *V*, *E*) ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ (∃ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. *vector_sub* (*?u1.0*::(*real*, *3*) *cart*) *x* = *vector_add* (% *t1* (*vector_sub v x*)) (*vector_add* (% *t2* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (*vector_sub* (% *a w*) *x*))) (% *t3* (*vector_sub u x*))))

thm Planarity.IMP_NORM_FAN:

∀ (*va*::(*real*, *3*) *cart*) *vb*::(*real*, *3*) *cart*. *va* ≠ *vb* ⟶ *vector_norm* (*vector_sub va vb*) ≠ (*0*::*real*) ∧ (*0*::*real*) ≤ *vector_norm* (*vector_sub va vb*) ∧ (*0*::*real*) < *vector_norm* (*vector_sub va vb*) ∧ (*0*::*real*) ≤ *inverse_class.inverse* (*vector_norm* (*vector_sub va vb*)) ∧ (*0*::*real*) < *inverse_class.inverse* (*vector_norm* (*vector_sub va vb*)) ∧ *inverse_class.inverse* (*vector_norm* (*vector_sub va vb*)) ∗ *vector_norm* (*vector_sub va vb*) = (*1*::*real*)

thm Planarity.cross_dot_fully_surrounded_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT v EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub v1 x*) (*vector_sub v x*)) (*vector_sub u1 x*)

thm Planarity.cross_dot_fully_surrounded_ge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ ¬ *collinear*

(*INSERT x* (*INSERT v1* (*INSERT v EMPTY*))) ∧ (*0*::*real*) ≤ *azim x v1 v u1* ∧ *azim x v1 v u1* ≤ *pi* ⟶ (*0*::*real*) ≤ *dot* (*cross* (*vector_sub v1 x*) (*vector_sub v x*)) (*vector_sub u1 x*)

thm Planarity.AFF_LT_2_1:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*) ⟶ *aff_lt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*) = *GSPEC* (λ*GEN%PVAR%410*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%410* (∃ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. *t3* < (*0*::*real*) ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 v*) (% *t3 w*))) *y*)

thm Planarity.properties_of_collinear4_points_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT v EMPTY*)))

thm Planarity.cross_dot_fully_surrounded1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub v1 x*) (*vector_sub u1 x*)) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (*vector_sub* (% *a w*) *x*))

thm Planarity.exists_cross_dot_fully_surrounded1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub v1 x*) (*vector_sub u1 x*)) (*vector_sub* (*vector_add* (% ((*1*::*real*) − *h*) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*))) (% *h u*)) *x*)))

thm Planarity.cross_dot_fully_surrounded2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ fan80 (x, V, E) ∧ ¬ collinear (INSERT x (INSERT v1 (INSERT u1 EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) ∧ (0::real) < azim x v1 v u1 ∧ azim x v1 v u1 < pi ⟶ (0::real) < dot (cross (vector_add (% ((1::real) − a) u) (vector_sub (% a w) x)) (vector_sub v x)) (vector_sub u1 x)*

thm Planarity.exists_cross_dot_fully_surrounded2_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (a::real) (v1::(real, 3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ fan80 (x, V, E) ∧ ¬ collinear (INSERT x (INSERT v1 (INSERT u1 EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) ∧ (0::real) < azim x v1 v u1 ∧ azim x v1 v u1 < pi ⟶ (∃t>0::real. t < (1::real) ∧ (∀h::real. (0::real) < h ∧ h < t ⟶ (0::real) < dot (cross (vector_sub (vector_add (% ((1::real) − h) (vector_add (% ((1::real) − a) u) (% a w))) (% h u)) x) (vector_sub v x)) (vector_sub u1 x)))

thm Planarity.properties_of_coplanar:

∀(x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) v1::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ⟶ coplanar (INSERT x (INSERT v1 (INSERT v (INSERT u EMPTY))))

thm Planarity.coplanar_is_cross_fan:

∀(x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) v1::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ⟶ dot (cross (vector_sub v x) (vector_sub u x)) (vector_sub v1 x) = (0::real)

thm Planarity.lie_in_half_space_and_azim:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (a::real) (v1::(real, 3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ fan80 (x, V, E) ∧ ¬ collinear (INSERT x (INSERT v1 (INSERT u1 EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) ∧ (0::real) < azim x v1 v u1 ∧ azim x v1 v u1 < pi ⟶ (0::real) < dot (cross (vector_sub v x) (vector_sub u x)) (vector_sub v1 x)

thm Planarity.exists_cut_small_edges_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT* *v* (*INSERT* *u* *EMPTY*)) *E* ∧ *IN* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) *E* ∧ *sigma_fan* *x* *V* *E* *u* *w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v1* (*INSERT* *u1* *EMPTY*))) ∧ *IN* *v1* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a* *w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim* *x* *v1* *v* *u1* ∧ *azim* *x* *v1* *v* *u1* < *pi* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a* *w*))) (% *t* *u*)) *EMPTY*))) (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *u1* *EMPTY*))) ≠ *EMPTY*)

thm Planarity.aff_gt_subset_aff_ge:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)) ⟶ *SUBSET* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)))

thm Planarity.aff_gt1_subset_aff_ge:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v1* (*INSERT* *u* *EMPTY*))) ∧ *IN* *v1* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ⟶ *SUBSET* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *u* *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)))

thm Planarity.aff_gt12_subset_aff_ge:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v1* (*INSERT* *v* *EMPTY*))) ∧ *IN* *v1* (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ⟶ *SUBSET* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v1* (*INSERT* *v* *EMPTY*))) (*aff_ge* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)))

thm Planarity.aff_gt2_subset_aff_ge:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*)) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v1* (*INSERT* *u* *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *x* (*INSERT* *v1* (*INSERT* *v* *EMPTY*))) ∧ *IN* *v1* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ⟶ *azim* *x* *v1* *v* *u* = *pi*

thm Planarity.remove_variable_fan:

∀(*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t3* ∧ *w* = *vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* *u*)) ⟶ *u* = *vector_sub* (*vector_sub* (% (*inverse_class.inverse* *t3*) *w*) (% (*inverse_class.inverse t3* ∗ *t1*) *x*)) (% (*inverse_class.inverse t3* ∗ *t2*) *v*)

thm Planarity.aff_gt_inter_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ⟶ *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) = *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT w EMPTY*)) (*INSERT v EMPTY*))

thm Planarity.aff_gt3_subset_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT v1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT v1 EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.aff_ge1_subset_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u EMPTY*))) ∧ *IN v1* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u EMPTY*))) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.aff_ge1_1_subset_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ *x* ≠ *v1* ∧ *IN v1* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1 EMPTY*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.decomposition_planar_by_angle_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ∧ *IN w* (*aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*)) ⟶ *IN u* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) ∨ *IN w* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.properties_of_fan7:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT* *v1* (*INSERT u1 EMPTY*)) *E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u1 EMPTY*))) ⟶ *v* = *v1* ∨ *v* = *u1*

thm Planarity.properties1_of_fan7:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT v1* (*INSERT u1 EMPTY*)) *E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1 EMPTY*)) ⟶ *v* = *v1*

thm Planarity.point_in_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ⟶ *IN x* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) ∧ *IN w* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))

thm Planarity.aff_ge_subset_aff_gt_union_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ⟶ *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*HOL_Light_Import.UNION* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)))

thm Planarity.pos_in_aff_ge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *a*::*real*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *a*) *v*) (% *a u*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.aff_gt1_subset_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.not_cut_inside_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *a* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*)

$-$ *h*) *u*) (% *h w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%413*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%413* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY*) ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) $-$ *a*) *u*) (% *a w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%414*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%414* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY*

thm Planarity.aff_ge_eq_aff_gt_union_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) ⟶ *aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) = *HOL_Light_Import.UNION* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT w EMPTY*)))

thm Planarity.AFFINE_HULL_1:

∀ *a*::(*real*, *?'a*::*type*) *cart*. *hull affine* (*INSERT a EMPTY*) = *GSPEC* (λ*GEN%PVAR%415*::(*real*, *?'a*::*type*) *cart*. ∃ *u*::*real*. *SETSPEC GEN%PVAR%415* (*u* = (*1*::*real*)) (% *u a*))

thm Planarity.exist_close1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*INSERT v* (*INSERT u EMPTY*)) (*INSERT v1* (*INSERT w1 EMPTY*)) = *EMPTY* ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ (*0*::*real*) < *azim x u w v* ∧ *azim x u w v* < *pi* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) $-$ *a*) *u*) (% *a w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%416*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%416* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY* ⟶ (∃ *h*>*0*::*real*. ∀ (*y1*::(*real*, *3*) *cart*) *y2*::(*real*, *3*) *cart*. *IN y1* (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) $-$ *a*) *u*) (% *a w*)) *EMPTY*))) (*ballnorm_fan x*)) ∧ *IN y2* (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*))) (*ballnorm_fan x*)) ⟶ *h* ≤ *distance* (*y1*, *y2*))

thm Planarity.properties_inside_collinear0_fan:

∀ (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ ¬ *collinear* (*INSERT x* (*INSERT w* (*INSERT u EMPTY*))) ⟶ ¬ *collinear* (*INSERT x* (*INSERT* (*vector_add* (% ((*1*::*real*) $-$ *a*) *u*) (% *a w*)) (*INSERT u EMPTY*)))

thm Planarity.properties1_inside_fan:

$\forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ \wedge\ (0::real) < (?a::real)\ \wedge\ ?a < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - ?a)\ u)\ (\%\ ?a\ w))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ w\ EMPTY)))$

thm Planarity.properties_inside_collinear_fan:

$\forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ a::real.\ (0::real) < a \wedge a < (1::real) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \longrightarrow \neg\ collinear\ (INSERT\ x\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ (INSERT\ u\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ (INSERT\ w\ EMPTY)))$

thm Planarity.properties_inside_collinear1_fan:

$\forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \wedge (0::real) < (?a::real) \wedge ?a < (1::real) \longrightarrow SUBSET\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ EMPTY))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ (vector\_add\ (\%\ ((1::real) - ?a)\ u)\ (\%\ ?a\ w))\ (INSERT\ w\ EMPTY))))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ EMPTY)$

thm Planarity.lemma_proof0_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ a::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ u\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)) \wedge (0::real) < a \wedge a < (1::real) \longrightarrow SUBSET\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ EMPTY))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))))\ (INSERT\ x\ EMPTY)$

thm Planarity.lemma_proof1_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ a::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ w\ (INSERT\ v1\ (INSERT\ w1\ EMPTY)) \wedge (0::real) < a \wedge a < (1::real) \longrightarrow SUBSET\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ EMPTY))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ w1\ EMPTY))))\ (INSERT\ x\ EMPTY)$

thm Planarity.GRAPH:

$\forall E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ graph\ E = (\forall e::?'a::type \Rightarrow bool.\ IN\ e\ E \longrightarrow HAS\_SIZE\ e\ (2::nat))$

thm Planarity.CARD_2_FAN:

$\forall (v::?'a::type)\ w::?'a::type.\ v \neq w \longrightarrow CARD\ (INSERT\ v\ (INSERT\ w\ EMPTY)) = (2::nat)$

thm Planarity.lemma_proof_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *INSERT u* (*INSERT w EMPTY*) ≠ *INSERT v1* (*INSERT w1 EMPTY*) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ *SUBSET* (*HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*)) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT w1 EMPTY*)))) (*INSERT x EMPTY*)

thm Planarity.remark012_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ ¬ *IN* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *V*

thm Planarity.remark01_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v1* (*INSERT w1 EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *INSERT u* (*INSERT w EMPTY*) ≠ *INSERT v1* (*INSERT w1 EMPTY*) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ ¬ *IN* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) (*INSERT v1* (*INSERT w1 EMPTY*))

thm Planarity.remark0_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *e*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *IN e E* ⟶ *HOL_Light_Import*.*INTER* (*INSERT x EMPTY*) *e* = *EMPTY*

thm Planarity.case2_proof_fan:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v* (*INSERT* (*?w1.0*::(*real*, *3*) *cart*) *EMPTY*)) *E* ∧ (*0*::*real*) < *azim x u w v* ∧ *azim x u w v* < *pi* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*) = (*?ua*::(*real*, *3*) *cart*) ∧ *?ua* ≠ *w* ∧ ¬ *collinear* (*INSERT x* (*INSERT w* (*INSERT ?ua EMPTY*))) ∧ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT ?ua EMPTY*))) (*GSPEC* (λ*GEN%PVAR%417*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%417* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY* ⟶ (∀ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e* (*DELETE E* (*INSERT u* (*INSERT w EMPTY*))) ⟶ *HOL_Light_Import*.*INTER* (*aff_ge* (*INSERT x EMPTY*) *e*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT ?ua* (*INSERT w EMPTY*))) = *aff_ge* (*INSERT x EMPTY*) (*HOL_Light_Import*.*INTER e* (*INSERT ?ua* (*INSERT w EMPTY*))))

thm Planarity.inequality1_not0_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (d::real)\ a::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < d \wedge (0::real) < a \wedge a < (1::real) \longrightarrow (\exists h>a.\ h \leq (1::real) \wedge (\forall t::real.\ a \leq t \wedge t < h \longrightarrow (\forall s::real.\ (0::real) \leq s \wedge s \leq (1::real) \longrightarrow s * (inverse\_class.inverse\ (vector\_norm\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w)))\ x))) * vector\_norm\ (vector\_sub\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w)))) < d)))$

thm Planarity.bounded_convex1_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \longrightarrow (\exists h>0::real.\ \forall y::(real,\ 3)\ cart.\ IN\ y\ (hull\ convex\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow vector\_norm\ (vector\_sub\ y\ x) < h)$

thm Planarity.inequaility2_not0_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (d::real)\ a::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < d \wedge (0::real) < a \wedge a < (1::real) \longrightarrow (\exists h>a.\ h \leq (1::real) \wedge (\forall t::real.\ a \leq t \wedge t < h \longrightarrow (\forall s::real.\ (0::real) \leq s \wedge s \leq (1::real) \longrightarrow vector\_norm\ (vector\_sub\ (\%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w)))\ x))))\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w)))\ x)))\ (\%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w)))\ x))))\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w)))\ x)))) < d)))$

thm Planarity.exists_point_small_edges_not0_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (d::real)\ a::real.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < d \wedge (0::real) < a \wedge a < (1::real) \longrightarrow (\exists h>a.\ h \leq (1::real) \wedge (\forall t::real.\ a \leq t \wedge t < h \longrightarrow (\forall s::real.\ (0::real) \leq s \wedge s \leq (1::real) \longrightarrow vector\_norm\ (vector\_sub\ (\%\ (inverse\_class.inverse\ (vector\_norm\ (vector\_add\ (\%\ ((1::real) - s)\ v)\ (vector\_sub\ (\%\ s\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w)))\ x))))\ (vector\_add\ (\%\ ((1::real) - s)\ v)$

(*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*))) *x*))) (% (*inverse_class.inverse* (*vector_norm* (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) (*vector_add* (% ((*1*::*real*) − *s*) *v*) (*vector_sub* (% *s* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*))) *x*)))) < *d*)))

thm Planarity.separate1_sphere_not0_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*INSERT v* (*INSERT u EMPTY*)) (*INSERT v1* (*INSERT u1 EMPTY*)) = *EMPTY* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v1* (*INSERT u1 EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *azim x u w v* ∧ *azim x u w v* < *pi* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%418*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%418* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY* ⟶ (∃ *h*>*a*. *h* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. *a* < *t* ∧ *t* < *h* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) (*HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u1 EMPTY*))) (*ballnorm_fan x*)) = *EMPTY*))

thm Planarity.fan_run_in_small11_not0_is_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*INSERT v* (*INSERT u EMPTY*)) (*INSERT v1* (*INSERT u1 EMPTY*)) = *EMPTY* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT v1* (*INSERT u1 EMPTY*)) *E* ∧ ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *azim x u w v* ∧ *azim x u w v* < *pi* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%419*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%419* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) = *EMPTY* ⟶ (∃ *t1*>*a*. *t1* ≤ (*1*::*real*) ∧ (∀ *t*::*real*. *a* < *t* ∧ *t* < *t1* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u1 EMPTY*))) = *EMPTY*))

thm Planarity.fan_run_in_small1_not0_is_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*INSERT v* (*INSERT u EMPTY*)) (*INSERT v1* (*INSERT u1 EMPTY*)) =

$EMPTY \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ v1\ (INSERT\ u1\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u\ w\ v < pi \wedge (0::real) < a \wedge a < (1::real) \wedge HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ EMPTY)))\ (GSPEC\ (\lambda GEN\%PVAR\%420::(real, 3)\ cart.\ \exists v::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%420\ (\exists e::(real, 3)\ cart \Rightarrow bool.\ IN\ e\ E \wedge IN\ v\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e))\ v)) = EMPTY \wedge (\forall s::real.\ (0::real) < s \wedge s < a \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - s)\ u)\ (\%\ s\ w))\ EMPTY)))\ (GSPEC\ (\lambda GEN\%PVAR\%421::(real, 3)\ cart.\ \exists v::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%421\ (\exists e::(real, 3)\ cart \Rightarrow bool.\ IN\ e\ E \wedge IN\ v\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e))\ v)) = EMPTY) \longrightarrow (\exists t1>a.\ t1 \leq (1::real) \wedge (\forall t::real.\ (0::real) < t \wedge t < t1 \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w))\ EMPTY)))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ u1\ EMPTY))) = EMPTY))$

thm Planarity.fan_run_in_small2_not0_is_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ a::real.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ v\ (INSERT\ w1\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u\ w\ v < pi \wedge (0::real) < a \wedge a < (1::real) \wedge HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ EMPTY)))\ (GSPEC\ (\lambda GEN\%PVAR\%422::(real, 3)\ cart.\ \exists v::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%422\ (\exists e::(real, 3)\ cart \Rightarrow bool.\ IN\ e\ E \wedge IN\ v\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e))\ v)) = EMPTY \wedge (\forall s::real.\ (0::real) < s \wedge s < a \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - s)\ u)\ (\%\ s\ w))\ EMPTY)))\ (GSPEC\ (\lambda GEN\%PVAR\%423::(real, 3)\ cart.\ \exists v::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%423\ (\exists e::(real, 3)\ cart \Rightarrow bool.\ IN\ e\ E \wedge IN\ v\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e))\ v)) = EMPTY) \longrightarrow (\exists t1>a.\ t1 \leq (1::real) \wedge (\forall t::real.\ (0::real) < t \wedge t < t1 \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w))\ EMPTY)))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w1\ EMPTY))) = EMPTY))$

thm Planarity.fan_run_in_small3_not0_is_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ a::real.\ FAN\ (x, V, E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w1\ EMPTY))\ E \wedge \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u$

w v < pi ∧ (0::real) < a ∧ a < (1::real) ∧ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) (GSPEC (λGEN%PVAR%424::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%424 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY ∧ (∀ s::real. (0::real) < s ∧ s < a ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − s) u) (% s w)) EMPTY))) (GSPEC (λGEN%PVAR%425::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%425 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY) ⟶ (∃ t1>a. t1 ≤ (1::real) ∧ (∀ t::real. (0::real) < t ∧ t < t1 ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − t) u) (% t w)) EMPTY))) (aff_ge (INSERT x EMPTY) (INSERT u (INSERT w1 EMPTY))) = EMPTY))

thm Planarity.fan_run_in_small_not0_is_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (v1::(real, 3) cart) (w1::(real, 3) cart) a::real. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT v1 (INSERT w1 EMPTY)) E ∧ (0::real) < azim x u w v ∧ azim x u w v < pi ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) (GSPEC (λGEN%PVAR%426::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%426 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY ∧ (∀ s::real. (0::real) < s ∧ s < a ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − s) u) (% s w)) EMPTY))) (GSPEC (λGEN%PVAR%427::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%427 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY) ⟶ (∃ t1>a. t1 ≤ (1::real) ∧ (∀ t::real. (0::real) < t ∧ t < t1 ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − t) u) (% t w)) EMPTY))) (aff_ge (INSERT x EMPTY) (INSERT v1 (INSERT w1 EMPTY))) = EMPTY))

thm Planarity.fan_run1_in_small_not0_is_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E′::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) a::real. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ SUBSET E′ E ∧ (0::real) < azim x u w v ∧ azim x u w v < pi ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) (GSPEC (λGEN%PVAR%428::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%428 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧

IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY ∧ (∀ s::real. (0::real) < s ∧ s < a ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − s) u) (% s w)) EMPTY))) (GSPEC (λGEN%PVAR%429::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%429 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY) ⟶ (∃ h>a. h ≤ (1::real) ∧ (∀ s::real. (0::real) < s ∧ s < h ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − s) u) (% s w)) EMPTY))) (GSPEC (λGEN%PVAR%430::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%430 (∃ e::(real, 3) cart ⇒ bool. IN e E′ ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY))

thm Planarity.cut_in_edges_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) a::real. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a ≤ (1::real) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) (GSPEC (λGEN%PVAR%434::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%434 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) ≠ EMPTY ⟶ a = (1::real)

thm Planarity.not_cut_in_edges_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) a::real. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a < (1::real) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) (GSPEC (λGEN%PVAR%440::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%440 (∃ e::(real, 3) cart ⇒ bool. IN e E ∧ IN v (aff_ge (INSERT x EMPTY) e)) v)) = EMPTY

thm Planarity.lie_in_half_space_and_azim_le:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (a::real) (v1::(real, 3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < a ∧ a ≤ (1::real) ∧ fan80 (x, V, E) ∧ ¬ collinear (INSERT x (INSERT v1 (INSERT u1 EMPTY))) ∧ IN v1 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − a) u) (% a w)) EMPTY))) ∧ (0::real) < azim x v1 v u1 ∧ azim x v1 v u1 < pi ⟶ (0::real) < dot (cross (vector_sub v x) (vector_sub u x)) (vector_sub v1 x)

thm Planarity.cross_dot_fully_surrounded1_fan_le:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub v1 x*) (*vector_sub u1 x*)) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (*vector_sub* (% *a w*) *x*))

thm Planarity.exists_cross_dot_fully_surrounded1_fan_le:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub v1 x*) (*vector_sub u1 x*)) (*vector_sub* (*vector_add* (% ((*1*::*real*) − *h*) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*))) (% *h u*)) *x*)))

thm Planarity.cross_dot_fully_surrounded2_fan_le:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (*vector_sub* (% *a w*) *x*)) (*vector_sub v x*)) (*vector_sub u1 x*)

thm Planarity.exists_cross_dot_fully_surrounded2_fan_le:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a* *w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_sub* (*vector_add* (% ((*1*::*real*) − *h*) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*))) (% *h u*)) *x*) (*vector_sub v x*)) (*vector_sub u1 x*)))

thm Planarity.exists_cut_small_edges_fan_le:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*v1*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v1* (*INSERT u1 EMPTY*))) ∧ *IN v1* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) ∧ (*0*::*real*) < *azim x v1 v u1* ∧ *azim x v1 v u1* < *pi* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*))) (% *t u*)) *EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v1* (*INSERT u1 EMPTY*))) ≠ *EMPTY*)

thm Planarity.AFF_GT_CUT_XFAN_IMP_EDGE_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*GSPEC* (λ*GEN%PVAR%449*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%449* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) ≠ *EMPTY* ⟶ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E*

thm Planarity.DHVFGBC:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (*0*::*real*) < *a* ∧ *a* ≤ (*1*::*real*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*))) (*GSPEC* (λ*GEN%PVAR%450*::(*real*, *3*) *cart*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%450* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *IN v* (*aff_ge* (*INSERT x EMPTY*) *e*)) *v*)) ≠ *EMPTY* ⟶ *a* = (*1*::*real*) ∧ *independent* (*INSERT* (*vector_sub v x*) (*INSERT* (*vector_sub u x*) (*INSERT* (*vector_sub* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *x*) *EMPTY*)))

thm Planarity.exists_in_aff_gt:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow (\exists\, y::(real,\ 3)\ cart.\ IN\ y\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY))))$

thm Planarity.cut_inside_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w1\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u\ w\ v < pi \wedge (0::real) < azim\ x\ v\ u\ w1 \wedge azim\ x\ v\ u\ w1 < pi \wedge (0::real) < azim\ x\ v\ w1\ w \wedge azim\ x\ v\ w1\ w < pi \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w1\ EMPTY))\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \neq EMPTY$

thm Planarity.exists_cut_in_edge_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w1\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u\ w\ v < pi \wedge (0::real) < azim\ x\ v\ u\ w1 \wedge azim\ x\ v\ u\ w1 < pi \wedge (0::real) < azim\ x\ v\ w1\ w \wedge azim\ x\ v\ w1\ w < pi \longrightarrow (\exists\, a>0::real.\ a < (1::real) \wedge IN\ (vector\_add\ (\%\ ((1::real) - a)\ u)\ (\%\ a\ w))\ (aff\_ge\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ w1\ EMPTY)))$

thm Planarity.notcoplanar_imp_notcollinear_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow \neg\ collinear\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Planarity.properties_of_fully_surrounded1_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \wedge (0::real) < azim\ x\ u\ w\ v \wedge azim\ x\ u\ w\ v < pi \longrightarrow (0::real) < azim\ x\ v\ u\ w \wedge azim\ x\ v\ u\ w < pi$

thm Planarity.in_aff_2_2_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow (\forall\, t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow (\forall (t1::real)\ (t2::real)\ t3::real.\ (0::real) < t3 \wedge t1 + (t2 + t3) = (1::real) \longrightarrow IN\ (vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (\%\ t3\ (vector\_add\ (\%\ ((1::real) - t)\ u)\ (\%\ t\ w)))))\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ EMPTY))\ (INSERT\ u\ (INSERT\ w\ EMPTY)))))$

thm Planarity.inequality4_aim_in_convex_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *azim x u w v* ∧ *azim x u w v* < *pi* ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ (*0*::*real*) < *azim x v u* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) ∧ *azim x v u* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) < *azim x v u w*

thm Planarity.condition_to_in_aff_gt_by_angle:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *s1*::*real*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ (*0*::*real*) < *dot* (*vector_sub v x*) (*vector_sub u x*) ∧ (*0*::*real*) < *s1* ∧ *s1* < *atn* (*vector_norm* (*cross* (*vector_sub v x*) (*vector_sub u x*)) ∗ *inverse_class.inverse* (*dot* (*vector_sub v x*) (*vector_sub u x*))) ⟶ *IN* (*vector_add* (% (*sin s1*) (*e1_fan x v u*)) (*vector_add* (% (*cos s1*) (*e3_fan x v u*)) *x*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.condition1_to_in_aff_gt_by_angle:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *s1*::*real*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ (*0*::*real*) < *s1* ∧ *s1* < *pi* / *real_of_nat* (*2*::*nat*) ∧ *dot* (*vector_sub v x*) (*vector_sub u x*) ≤ (*0*::*real*) ⟶ *IN* (*vector_add* (% (*sin s1*) (*e1_fan x v u*)) (*vector_add* (% (*cos s1*) (*e3_fan x v u*)) *x*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.angle_is_small_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *fan80* (*x*, *V*, *E*) ∧ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ *azim x v u w* ≤ *azim x v u* (*sigma_fan x V E v u*)

thm Planarity.angle_is_smallpi_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *fan80* (*x*, *V*, *E*) ∧ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ (*0*::*real*) < *azim x v u w* ∧ *azim x v u w* < *pi*

thm Planarity.exists_rw_dart_inter_aff_gt_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *fan80* (*x*, *V*, *E*) ∧ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ ($\exists$ *h*>*0*::*real*. $\forall$ *t*::*real*. (*0*::*real*)

< t ∧ t < h ⟶ (∀ s::real. (0::real) < s ∧ s < pi / real_of_nat (2::nat) ⟶ HOL_Light_Import.INTER (rw_dart_fan x V E (x, v, u, sigma_fan x V E v u) (cos s)) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT (vector_add (% ((1::real) − t) u) (% t w)) EMPTY))) ≠ EMPTY))

thm Planarity.scale_in_edges_fan:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. DISJOINT (INSERT x EMPTY) (INSERT v (INSERT u EMPTY)) ∧ IN w (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ⟶ (∃ (a::real) t::real. (0::real) < a ∧ (0::real) < t ∧ t < (1::real) ∧ % a (vector_sub w x) = vector_add (% ((1::real) − t) v) (vector_sub (% t u) x))

thm Planarity.aff_gt_imp_not_collinear:

∀ (x::(real, 3) cart) (u::(real, 3) cart) (v::(real, 3) cart) w::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ IN w (aff_gt (INSERT x (INSERT v EMPTY)) (INSERT u EMPTY)) ⟶ ¬ collinear (INSERT x (INSERT v (INSERT w EMPTY)))

thm Planarity.conditions_in_rcone_fan:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) s::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ IN w (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ∧ (0::real) < s ∧ s < pi / real_of_nat (2::nat) ∧ IN u (rcone_fan x v (cos s)) ⟶ IN w (rcone_fan x v (cos s))

thm Planarity.exists_point_inside_domain_cone_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) s::real. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (0::real) < s ∧ s < pi / real_of_nat (2::nat) ∧ fan80 (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ (∃ y::(real, 3) cart. IN y (rw_dart_fan x V E (x, u, w, ?w2.0::(real, 3) cart) (cos s)) ∧ azim x v u y < azim x v u w)

thm Planarity.cut_in_angle_fan:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) y::(real, 3) cart. ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ∧ ¬ collinear (INSERT x (INSERT u (INSERT y EMPTY))) ∧ (0::real) < azim x u w v ∧ azim x u w v < pi ∧ azim x u w y < azim x u w v ∧ (0::real) < azim x u w y ⟶ LET (λa1::(real, 3) cart. LET_END (LET (λa2::(real, 3) cart. LET_END (LET (λa3::(real, 3) cart. LET_END (LET (λa4::(real, 3) cart. LET_END (LET (λva::(real, 3) cart. LET_END (LET (λvb::(real, 3) cart. LET_END (LET (λv3::(real, 3) cart. LET_END (IN v3 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT w EMPTY))))) (vector_add (cross vb va) x))) (cross a3 a4))) (cross a1 a2))) (vector_sub u x))) (vector_sub y x))) (vector_sub w x))) (vector_sub v x)

thm Planarity.aff_gt_1_2_scale_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. (*0*::*real*) < *a* $\wedge$ % *a* (*vector_sub u x*) = *vector_sub w x* $\wedge$ $\neg$ *collinear* (*INSERT x* (*INSERT w* (*INSERT v EMPTY*))) $\longrightarrow$ *aff_gt* (*INSERT x EMPTY*) (*INSERT u* (*INSERT v EMPTY*)) = *aff_gt* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))

thm Planarity.exists_cut_rcone_fan_with_edge_run_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *s*::*real*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ (*0*::*real*) < *s* $\wedge$ *s* < *pi* / *real_of_nat* (*2*::*nat*) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\longrightarrow$ ($\exists$ *t*>*0*::*real*. *t* < (*1*::*real*) $\wedge$ *HOL_Light_Import.INTER* (*rw_dart_fan x V E* (*x*, *u*, *w*, *sigma_fan x V E u w*) (*cos s*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) $\neq$ *EMPTY*)

thm Planarity.aff_gt_in_rw_dart_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *s*::*real*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ (*0*::*real*) < *s* $\wedge$ *s* < *pi* / *real_of_nat* (*2*::*nat*) $\wedge$ *IN y* (*rw_dart_fan x V E* (*x*, *u*, *w*, *v*) (*cos s*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\longrightarrow$ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT u* (*INSERT y EMPTY*))) (*rw_dart_fan x V E* (*x*, *u*, *w*, *sigma_fan x V E u w*) (*cos s*))

thm Planarity.in_aff_gt_1_2:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *t*::*real*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) $\wedge$ (*0*::*real*) < *t* $\wedge$ *t* < (*1*::*real*) $\longrightarrow$ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *v*) (% *t u*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.exists_rw_dart_inter_aff_gt1_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *s*::*real*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ (*0*::*real*) < *s* $\wedge$ *s* < *pi* / *real_of_nat* (*2*::*nat*) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\longrightarrow$ ($\exists$ *h*>*0*::*real*. $\forall$ *t*::*real*. (*0*::*real*) < *t* $\wedge$ *t* < *h* $\longrightarrow$ *HOL_Light_Import.INTER* (*rw_dart_fan x V E* (*x*, *u*, *w*, *sigma_fan x V E u w*) (*cos s*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))) $\neq$ *EMPTY*)

thm Planarity.there_exists_component_contain_aff_gt_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ (∃ *h*>*0*::*real*. ∃ *y*::(*real*, *3*) *cart*. *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *h*) *u*) (% *h w*)) *EMPTY*))) (*connected_component* (*yfan* (*x*, *V*, *E*)) *y*))

thm Planarity.CONNECTED_COMPONENT_OF_SUBSET:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SUBSET s t* ∧ *connected_component s x y* ⟶ *connected_component t x y*

thm Planarity.connected_component_of_faces_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *dart_leads_into x V E v u* = *dart_leads_into x V E u w*

thm Planarity.exists_dartset_leads_into_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ (∃ *s*::(*real*, *3*) *cart* ⇒ *bool*. ∀ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y ds* ⟶ *s* = *dart_leads_into x V E* (*pr2 y*) (*pr3 y*))

thm DEF_dartset_leads_into_fan:

dartset_leads_into_fan = (λ(*_2957091*::(*real*, *3*) *cart*) (*_2957092*::(*real*, *3*) *cart* ⇒ *bool*) (*_2957093*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *_2957094*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *SOME s*::(*real*, *3*) *cart* ⇒ *bool*. ∀ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y _2957094* ⟶ *s* = *dart_leads_into _2957091 _2957092 _2957093* (*pr2 y*) (*pr3 y*))

thm Planarity.dartset_leads_into_fan:

∀ (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *dartset_leads_into_fan x V E ds* = (*SOME s*::(*real*, *3*) *cart* ⇒ *bool*. ∀ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y ds* ⟶ *s* = *dart_leads_into x V E* (*pr2 y*) (*pr3 y*))

thm Planarity.DARTSET_LEADS_INTO_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow (\forall y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ ds \longrightarrow dartset\_leads\_into\_fan\ x\ V\ E\ ds = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y))$

thm Planarity.UNIQUE_DARTSET_LEADS_INTO_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ s::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (\forall y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ ds \longrightarrow s = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y)) \longrightarrow dartset\_leads\_into\_fan\ x\ V\ E\ ds = s$

thm Planarity.equality_dart_leads_into:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ y1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge IN\ y\ ds \wedge IN\ y1\ ds \longrightarrow dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y) = dart\_leads\_into\ x\ V\ E\ (pr2\ y1)\ (pr3\ y1)$

thm Planarity.UNIQUE_DARTSET_LEADS_INTO1_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (s::(real,\ 3)\ cart \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge IN\ y\ ds \wedge s = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y) \longrightarrow dartset\_leads\_into\_fan\ x\ V\ E\ ds = s$

thm Planarity.exists_point_dart_leads_into_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow (\exists y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ ds \wedge dartset\_leads\_into\_fan\ x\ V\ E\ ds = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y))$

thm Planarity.dartset_leads_into_is_topological_component_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow IN\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)\ (topological\_component\_yfan\ (x,\ V,\ E))$

thm Planarity.dartset_leads_into_subset_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow SUBSET\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)\ (yfan\ (x,\ V,\ E))$

thm Planarity.RWXUYZZ:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow (\exists s::(real,\ 3)\ cart \Rightarrow bool.\ \forall y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ y\ ds \longrightarrow s = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y)) \wedge IN\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)\ (topological\_component\_yfan\ (x,\ V,\ E))$

thm Planarity.add_edge_graph_fan:

$\forall (V::?'a::type \Rightarrow bool)\ (E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (v::?'a::type)\ u::?'a::type.\ IN\ v\ V \wedge IN\ u\ V \wedge (?E1.0::(?'a::type \Rightarrow bool) \Rightarrow bool) = HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ EMPTY) \wedge SUBSET\ (UNIONS\ E)\ V \longrightarrow SUBSET\ (UNIONS\ ?E1.0)\ V$

thm Planarity.garph_add_edge_is_garph:

$\forall (V::?'a::type \Rightarrow bool)\ (E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (v::?'a::type)\ u::?'a::type.\ (?E1.0::(?'a::type \Rightarrow bool) \Rightarrow bool) = HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ EMPTY) \wedge v \neq u \wedge graph\ E \longrightarrow graph\ ?E1.0$

thm Planarity.add_edge_into_collinear_fan:

$\forall (x::(real,\ 3)\ cart)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge (\forall e::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e\ E \longrightarrow \neg\ collinear\ (HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ e)) \longrightarrow (\forall e::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e\ (HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ EMPTY)) \longrightarrow \neg\ collinear\ (HOL\_Light\_Import.UNION\ (INSERT\ x\ EMPTY)\ e))$

thm Planarity.condition_not_edge_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN v V ∧ IN u V ∧ ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ IN (?e1.0::(real, 3) cart ⇒ bool) E ∧ (?e2.0::(real, 3) cart ⇒ bool) = INSERT v (INSERT u EMPTY) ∧ SUBSET (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (dartset_leads_into_fan x V E ds) ⟶ ?e1.0 ≠ ?e2.0

thm Planarity.properties_edges_eq_fan:

∀ (e::?'a::type ⇒ bool) (v::?'a::type) u::?'a::type. FINITE e ∧ e ≠ INSERT v (INSERT u EMPTY) ∧ v ≠ u ∧ CARD e = (2::nat) ⟶ ¬ IN v e ∨ ¬ IN u e

thm Planarity.condition_not_intersection_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (e1::(real, 3) cart ⇒ bool) e2::(real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN v V ∧ IN u V ∧ ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ SUBSET (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (dartset_leads_into_fan x V E ds) ∧ IN e1 E ∧ e2 = INSERT v (INSERT u EMPTY) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER e1 e2)

thm Planarity.exists_edge_fully_surround_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) w::(real, 3) cart. FAN (x, V, E) ∧ IN w V ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ (∃ v::(real, 3) cart. IN (INSERT w (INSERT v EMPTY)) E ∧ IN v V)

thm Planarity.condition_not_intersection_point_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (e1::(real, 3) cart ⇒ bool) w::(real, 3) cart. FAN (x, V, E) ∧ IN v V ∧ IN u V ∧ ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ SUBSET (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (dartset_leads_into_fan x V E ds) ∧ IN w V ∧ e1 = INSERT v (INSERT u EMPTY) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) (INSERT w EMPTY)) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER e1 (INSERT w EMPTY))

thm Planarity.DWWUTKW:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge IN\ v\ V \wedge IN\ u\ V \wedge \neg\ collinear\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge SUBSET\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds) \wedge (?E1.0::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) = HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ EMPTY) \longrightarrow FAN\ (x,\ V,\ ?E1.0)$

thm Planarity.exists_point_in_component_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ U::(real,\ 3)\ cart \Rightarrow bool.\ IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow (\exists z::(real,\ 3)\ cart.\ IN\ z\ U)$

thm Planarity.nonsetedge_fully_surround_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge FAN\ (x,\ V,\ E) \longrightarrow E \neq EMPTY$

thm Planarity.exists_point_notx_in_xfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge E \neq EMPTY \longrightarrow (\exists v::(real,\ 3)\ cart.\ IN\ v\ (xfan\ (x,\ V,\ E)) \wedge v \neq x)$

thm Planarity.x_in_xfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge E \neq EMPTY \longrightarrow IN\ x\ (xfan\ (x,\ V,\ E))$

thm Planarity.xfan_notempty_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge E \neq EMPTY \longrightarrow xfan\ (x,\ V,\ E) \neq EMPTY$

thm Planarity.xfan_closed_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \longrightarrow HOL\_Light\_Import.closed\ (xfan\ (x,\ V,\ E))$

thm Planarity.topological_component_subset_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ U::(real,\ 3)\ cart \Rightarrow bool.\ IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow SUBSET\ U\ (yfan\ (x,\ V,\ E))$

thm Planarity.aff_gt_connect_bound_not_inter_edges_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (y::(real, 3) cart) z::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ DISJOINT (INSERT x EMPTY) (INSERT y (INSERT z EMPTY)) ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT y (INSERT z EMPTY))) (aff_ge (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) = EMPTY

thm Planarity.aff_gt_connect_bound_subset_yfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart) z::(real, 3) cart. FAN (x, V, E) ∧ DISJOINT (INSERT x EMPTY) (INSERT y (INSERT z EMPTY)) ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT y (INSERT z EMPTY))) (yfan (x, V, E))

thm Planarity.sym_line1_fan:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. IN x (aff (INSERT y (INSERT z EMPTY))) ∧ x ≠ y ⟶ IN z (aff (INSERT x (INSERT y EMPTY)))

thm Planarity.POINT_IN_LINE:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x (aff (INSERT x (INSERT y EMPTY)))

thm Planarity.POINT_IN_LINE1:

∀ (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN y (aff (INSERT x (INSERT y EMPTY)))

thm Planarity.AFFINE_HULL_AFFINE_EQ:

∀ s::(real, ?'a::type) cart ⇒ bool. hull affine (hull affine s) = hull affine s

thm Planarity.sym_line0_fan:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. IN x (aff (INSERT y (INSERT z EMPTY))) ∧ DISJOINT (INSERT x EMPTY) (INSERT y (INSERT z EMPTY)) ⟶ SUBSET (aff (INSERT x (INSERT z EMPTY))) (aff (INSERT x (INSERT y EMPTY)))

thm Planarity.sym_line_fan:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) z::(real, ?'a::type) cart. IN x (aff (INSERT y (INSERT z EMPTY))) ∧ DISJOINT (INSERT x EMPTY) (INSERT y (INSERT z EMPTY)) ⟶ aff (INSERT x (INSERT z EMPTY)) = aff (INSERT x (INSERT y EMPTY))

thm Planarity.sym_line01_fan:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *IN* *x* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ *DISJOINT* (*INSERT* *y* *EMPTY*) (*INSERT* *x* (*INSERT* *z* *EMPTY*)) $\longrightarrow$ *SUBSET* (*aff* (*INSERT* *y* (*INSERT* *x* *EMPTY*))) (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*)))

thm Planarity.sym_line02_fan:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *IN* *x* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ *DISJOINT* (*INSERT* *y* *EMPTY*) (*INSERT* *x* (*INSERT* *z* *EMPTY*)) $\longrightarrow$ *SUBSET* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) (*aff* (*INSERT* *y* (*INSERT* *x* *EMPTY*)))

thm Planarity.sym_line_fan0:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *IN* *x* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *y* (*INSERT* *z* *EMPTY*)) $\wedge$ *DISJOINT* (*INSERT* *y* *EMPTY*) (*INSERT* *x* (*INSERT* *z* *EMPTY*)) $\longrightarrow$ *aff* (*INSERT* *x* (*INSERT* *z* *EMPTY*)) = *aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))

thm Planarity.sym_line_fan1:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *IN* *x* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ *DISJOINT* (*INSERT* *y* *EMPTY*) (*INSERT* *x* (*INSERT* *z* *EMPTY*)) $\longrightarrow$ *aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*)) = *aff* (*INSERT* *y* (*INSERT* *x* *EMPTY*))

thm Planarity.aff_ge_1_1_subset_aff_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *y* $\neq$ *z* $\wedge$ *IN* *x* (*aff_ge* (*INSERT* *y* *EMPTY*) (*INSERT* *z* *EMPTY*)) $\longrightarrow$ *IN* *x* (*aff* (*INSERT* *y* (*INSERT* *z* *EMPTY*)))

thm Planarity.exists_point_notxin_convex_in_xfan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *z*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *x* $\neq$ *z* $\wedge$ *E* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *v*::(*real*, *3*) *cart*. *IN* *v* (*xfan* (*x*, *V*, *E*)) $\wedge$ $\neg$ *IN* *x* (*hull convex* (*INSERT* *v* (*INSERT* *z* *EMPTY*))))

thm Planarity.notempty_xfan_inter_segment_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*z*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* *v* (*xfan* (*x*, *V*, *E*)) $\longrightarrow$ *HOL_Light_Import.INTER* (*xfan* (*x*, *V*, *E*)) (*closed_segment* [(*v*, *z*)]) $\neq$ *EMPTY*

thm Planarity.xfan_inter_segment_closed_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*z*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\longrightarrow$ *HOL_Light_Import.closed* (*HOL_Light_Import.INTER* (*xfan* (*x*, *V*, *E*)) (*closed_segment* [(*v*, *z*)]))

thm Planarity.point_in_yfan_not_x_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge E \neq EMPTY \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \longrightarrow x \neq z$

thm Planarity.zpoint_in_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge E \neq EMPTY \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \longrightarrow IN\ z\ (yfan\ (x,\ V,\ E))$

thm Planarity.segment_in_segment:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ z::(real,\ ?'a::type)\ cart.\ IN\ z\ (closed\_segment\ [(x,\ y)]) \longrightarrow (\forall t::real.\ (0::real) \leq t \wedge t \leq (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ z)\ (\%\ t\ y))\ (closed\_segment\ [(x,\ y)]))$

thm Planarity.connect_insidepoint_to_bound_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \longrightarrow (\exists y::(real,\ 3)\ cart.\ y \neq x \wedge IN\ y\ (xfan\ (x,\ V,\ E)) \wedge (\forall t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ (yfan\ (x,\ V,\ E))))$

thm Planarity.expand_element_in_topological_component_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \longrightarrow U = connected\_component\ (yfan\ (x,\ V,\ E))\ z$

thm Planarity.segmentsubset_aff_gt:

$\forall (x::(real,\ ?'a::type)\ cart)\ (y::(real,\ ?'a::type)\ cart)\ (z::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY)) \wedge IN\ w\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \longrightarrow (\forall t::real.\ (0::real) \leq t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ w)\ (\%\ t\ z))\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY))))$

thm Planarity.point_in_aff_gt_in_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (y::(real,\ 3)\ cart)\ (z::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY)) \wedge IN\ w\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \wedge (\forall t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ (yfan\ (x,\ V,\ E))) \longrightarrow IN\ w\ (yfan\ (x,\ V,\ E))$

thm Planarity.segment_subset_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (y::(real,\ 3)\ cart)\ (z::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY)) \wedge IN\ z\ (yfan\ (x,\ V,\ E)) \wedge IN\ w\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY))) \wedge (\forall t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ (yfan\ (x,\ V,\ E))) \longrightarrow SUBSET\ (closed\_segment\ [(w,\ z)])\ (yfan\ (x,\ V,\ E))$

thm Planarity.exists_in_aff_gt_disjoint:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) \longrightarrow (\exists y::(real,\ 3)\ cart.\ IN\ y\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY))))$

thm Planarity.aff_gt_subset_component_y_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY)) \wedge (\forall t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ (yfan\ (x,\ V,\ E))) \longrightarrow SUBSET\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ (INSERT\ z\ EMPTY)))\ U$

thm Planarity.exists_connect_point_in_xfanto_yfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ z::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \wedge IN\ z\ U \longrightarrow (\exists y::(real,\ 3)\ cart.\ y \neq x \wedge IN\ y\ (xfan\ (x,\ V,\ E)) \wedge (\forall t::real.\ (0::real) < t \wedge t < (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ U))$

thm Planarity.place_there_point_line_fan:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ x \neq y \wedge IN\ z\ (aff\ (INSERT\ x\ (INSERT\ y\ EMPTY))) \longrightarrow (\exists t>0::real.\ t < (1::real) \wedge IN\ (vector\_add\ (\%\ ((1::real) - t)\ y)\ (\%\ t\ z))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ EMPTY)))$

thm Planarity.aff_ge_1_1_subset_xfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ y::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ y\ (xfan\ (x,\ V,\ E)) \wedge x \neq y \longrightarrow SUBSET\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ EMPTY))\ (xfan\ (x,\ V,\ E))$

thm Planarity.point_in_yfan_and_point_in_xfan_indepent_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (U::(real,\ 3)\ cart \Rightarrow bool)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ FAN\ (x,$

V, E) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ∧ *IN z U* ∧ *IN y* (*xfan* (*x*, *V*, *E*)) ∧ *y* ≠ *x* ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *y*) (% *t z*)) (*yfan* (*x*, *V*, *E*))) ⟶ ¬ *collinear* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))

thm Planarity.permutes_4points_collinear:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*z*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *x* ≠ *y* ∧ *x* ≠ *z* ∧ *IN y* (*aff* (*INSERT x* (*INSERT z EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT y* (*INSERT w EMPTY*))) ⟶ ¬ *collinear* (*INSERT x* (*INSERT z* (*INSERT w EMPTY*)))

thm Planarity.permutes_4points_collinear1:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*z*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *x* ≠ *y* ∧ *x* ≠ *z* ∧ *IN y* (*aff* (*INSERT x* (*INSERT z EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT z* (*INSERT w EMPTY*))) ⟶ ¬ *collinear* (*INSERT x* (*INSERT y* (*INSERT w EMPTY*)))

thm Planarity.in_aff_gt_eq_azim:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*w0*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *x* ≠ *z* ∧ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT z EMPTY*)) ⟶ *azim x y w0 w1* = *azim x z w0 w1*

thm Planarity.no_origin_aff_ge_is_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *x* ≠ *y* ∧ *x* ≠ *z* ∧ *IN z* (*aff_ge* (*INSERT x EMPTY*) (*INSERT y EMPTY*)) ⟶ *IN z* (*aff_gt* (*INSERT x EMPTY*) (*INSERT y EMPTY*))

thm Planarity.exists_edge_component_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN v V* ∧ ¬ *IN y* (*set_of_edge v V E*) ⟶ (∃ *w*::(*real*, *3*) *cart*. *IN w* (*set_of_edge v V E*) ∧ (∀ *w1*::(*real*, *3*) *cart*. *IN w1* (*set_of_edge v V E*) ⟶ *azim1 x v y w* ≤ *azim1 x v y w1*))

thm Planarity.v_subset_xfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ *SUBSET V* (*xfan* (*x*, *V*, *E*))

thm Planarity.set_of_edge_subset_edges:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SUBSET* (*set_of_edge v V E*) *V*

thm Planarity.aff_ge_2_1_is_exists_point_inaff_ge_1_2:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) w::(real, 3) cart. DISJOINT (INSERT x EMPTY) (INSERT y (INSERT w EMPTY)) ∧ DISJOINT (INSERT x (INSERT y EMPTY)) (INSERT w EMPTY) ∧ IN z (aff_ge (INSERT x (INSERT y EMPTY)) (INSERT w EMPTY)) ⟶ (∃ t>0::real. t < (1::real) ∧ IN (vector_add (% ((1::real) − t) y) (% t z)) (aff_ge (INSERT x EMPTY) (INSERT y (INSERT w EMPTY))))

thm Planarity.not_azim_points_in_yfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (U::(real, 3) cart ⇒ bool) (y::(real, 3) cart) (z::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN U (topological_component_yfan (x, V, E)) ∧ IN z U ∧ IN u V ∧ IN y (aff_ge (INSERT x EMPTY) (INSERT u EMPTY)) ∧ IN y (xfan (x, V, E)) ∧ y ≠ x ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ (∀ w1::(real, 3) cart. IN w1 (set_of_edge u V E) ⟶ azim x u z w1 ≠ (0::real))

thm Planarity.exists_edge_bounded_topological_component_yfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (U::(real, 3) cart ⇒ bool) (y::(real, 3) cart) (z::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN U (topological_component_yfan (x, V, E)) ∧ IN z U ∧ IN u V ∧ IN y (aff_ge (INSERT x EMPTY) (INSERT u EMPTY)) ∧ IN y (xfan (x, V, E)) ∧ y ≠ x ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ (∃ w::(real, 3) cart. IN (INSERT u (INSERT w EMPTY)) E ∧ IN z (w_dart_fan x V E (x, u, w, sigma_fan x V E u w)))

thm Planarity.aff_gt_in_w_dart_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (u::(real, 3) cart) (w::(real, 3) cart) y::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN y (w_dart_fan x V E (x, u, w, sigma_fan x V E u w)) ∧ fan80 (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT u (INSERT y EMPTY))) (w_dart_fan x V E (x, u, w, sigma_fan x V E u w))

thm Planarity.not_empty_rcone_fan_inter_aff_gt:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) h::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ (0::real) < h ∧ h ≤ pi ⟶ HOL_Light_Import.INTER (rcone_fan x v (cos h)) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ≠ EMPTY

thm Planarity.condition_rw_dart_fan_inter_aff_gt_is_not_empty:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (z::(real, 3) cart) h::real. FAN (x, V, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT z EMPTY*))) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN z* (*w_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*)) ∧ (*0*::*real*) < *h* ∧ *h* ≤ *pi* ⟶ *HOL_Light_Import*.*INTER* (*rw_dart_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) (*cos h*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT z EMPTY*))) ≠ *EMPTY*

thm Planarity.exists_edge_rw_dart_fan_inter_aff_gt_not_empty_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ∧ *IN z U* ∧ *IN u V* ∧ *IN y* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u EMPTY*)) ∧ *IN y* (*xfan* (*x*, *V*, *E*)) ∧ *y* ≠ *x* ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *y*) (% *t z*)) (*yfan* (*x*, *V*, *E*))) ⟶ (∃ *w*::(*real*, *3*) *cart*. *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* ≤ *pi* ⟶ *HOL_Light_Import*.*INTER* (*rw_dart_fan x V E* (*x*, *u*, *w*, *sigma_fan x V E u w*) (*cos h*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT u* (*INSERT z EMPTY*))) ≠ *EMPTY*))

thm Planarity.exists_edge_rw_dart_fan_inter_topological_component_not_empty_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ∧ *IN z U* ∧ *IN u V* ∧ *IN y* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u EMPTY*)) ∧ *IN y* (*xfan* (*x*, *V*, *E*)) ∧ *y* ≠ *x* ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *y*) (% *t z*)) (*yfan* (*x*, *V*, *E*))) ⟶ (∃ *w*::(*real*, *3*) *cart*. *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* ≤ *pi* ⟶ *HOL_Light_Import*.*INTER* (*rw_dart_fan x V E* (*x*, *u*, *w*, *sigma_fan x V E u w*) (*cos h*)) *U* ≠ *EMPTY*))

thm Planarity.exists_dart_leads_into_edge_eq_topological_component_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ∧ *IN z U* ∧ *IN u V* ∧ *IN y* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u EMPTY*)) ∧ *IN y* (*xfan* (*x*, *V*, *E*)) ∧ *y* ≠ *x* ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *y*) (% *t z*)) (*yfan* (*x*, *V*, *E*))) ⟶ (∃ *w*::(*real*, *3*) *cart*. *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *dart_leads_into x V E u w* = *U*)

thm Planarity.not_azim_points1_in_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*u*::(*real*, *3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN U (topological_component_yfan (x, V, E)) ∧ IN z U ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN y (aff_gt (INSERT x EMPTY) (INSERT u (INSERT w EMPTY))) ∧ IN y (xfan (x, V, E)) ∧ y ≠ x ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ azim x y z w ≠ (0::real)*

thm Planarity.not_azim_points2_in_yfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (U::(real, 3) cart ⇒ bool) (y::(real, 3) cart) (z::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN U (topological_component_yfan (x, V, E)) ∧ IN z U ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN y (aff_gt (INSERT x EMPTY) (INSERT u (INSERT w EMPTY))) ∧ IN y (xfan (x, V, E)) ∧ y ≠ x ∧ (∀ t::real. (0::real) < t ∧ t < (1::real) ⟶ IN (vector_add (% ((1::real) − t) y) (% t z)) (yfan (x, V, E))) ⟶ azim x y z u ≠ (0::real)

thm Planarity.condition_cross_dot_4point:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. LET (λa1::(real, 3) cart. LET_END (LET (λa2::(real, 3) cart. LET_END (LET (λa3::(real, 3) cart. LET_END (LET (λa4::(real, 3) cart. LET_END (LET (λva::(real, 3) cart. LET_END (LET (λvb::(real, 3) cart. LET_END (LET (λv3::(real, 3) cart. LET_END (¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ∧ (0::real) < dot (cross a1 a2) a4 ∧ (0::real) < − dot (cross a1 a2) a3 ⟶ IN v3 (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))))) (vector_add (cross va vb) x))) (cross a3 a4))) (cross a1 a2))) (vector_sub u x))) (vector_sub v x))) (vector_sub z x))) (vector_sub y x)

thm Planarity.aff_gt_2_1_crossr_dot_4point:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. LET (λa1::(real, 3) cart. LET_END (LET (λa2::(real, 3) cart. LET_END (LET (λa3::(real, 3) cart. LET_END (LET (λa4::(real, 3) cart. LET_END (¬ collinear (INSERT x (INSERT y (INSERT z EMPTY))) ∧ IN u (aff_gt (INSERT x (INSERT y EMPTY)) (INSERT z EMPTY)) ∧ (0::real) < dot (cross a1 a2) a3 ⟶ (0::real) < dot (cross a1 a4) a3)) (vector_sub u x))) (vector_sub v x))) (vector_sub z x))) (vector_sub y x)

thm Planarity.aff_gt_2_1_rcross_dot_4pointl:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. LET (λa1::(real, 3) cart. LET_END (LET (λa2::(real, 3) cart. LET_END (LET (λa3::(real, 3) cart. LET_END (LET (λa4::(real, 3) cart. LET_END (¬ collinear (INSERT x (INSERT y (INSERT z EMPTY))) ∧ IN u (aff_gt (INSERT x (INSERT z EMPTY)) (INSERT y EMPTY))

∧ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a3* ⟶ (*0*::*real*) < *dot* (*cross* *a4* *a2*) *a3*)) (*vector_sub* *u* *x*))) (*vector_sub* *v* *x*))) (*vector_sub* *z* *x*))) (*vector_sub* *y* *x*)

thm Planarity.aff_gt_2_1_cross_dotr_4point:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* *v* *EMPTY*))) ∧ *IN* *u* (*aff_gt* (*INSERT* *x* (*INSERT* *y* *EMPTY*)) (*INSERT* *v* *EMPTY*)) ∧ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a3* ⟶ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a4*)) (*vector_sub* *u* *x*))) (*vector_sub* *v* *x*))) (*vector_sub* *z* *x*))) (*vector_sub* *y* *x*)

thm Planarity.aff_gt_2_1_cross_dotl_4point:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT* *x* (*INSERT* *z* (*INSERT* *v* *EMPTY*))) ∧ *IN* *u* (*aff_gt* (*INSERT* *x* (*INSERT* *z* *EMPTY*)) (*INSERT* *v* *EMPTY*)) ∧ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a3* ⟶ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a4*)) (*vector_sub* *u* *x*))) (*vector_sub* *v* *x*))) (*vector_sub* *z* *x*))) (*vector_sub* *y* *x*)

thm Planarity.aff_gt_2_1r_rcross_dotl_4point:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *y* *EMPTY*))) ∧ *IN* *u* (*aff_gt* (*INSERT* *x* (*INSERT* *v* *EMPTY*)) (*INSERT* *y* *EMPTY*)) ∧ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a3* ⟶ (*0*::*real*) < *dot* (*cross* *a4* *a2*) *a3*)) (*vector_sub* *u* *x*))) (*vector_sub* *v* *x*))) (*vector_sub* *z* *x*))) (*vector_sub* *y* *x*)

thm Planarity.aff_gt_1_2_cross_dotr_4point:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ *IN* *y* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ (*0*::*real*) < *dot* (*cross* *a1* *a2*) *a3* ⟶ (*0*::*real*) < − *dot* (*cross* *a1* *a2*) *a4*)) (*vector_sub* *u* *x*))) (*vector_sub* *v* *x*))) (*vector_sub* *z* *x*))) (*vector_sub* *y* *x*)

thm Planarity.aff_gt_1_2_cross_dotr_4point_neg:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT* *x* (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧ *IN* *y* (*aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *u* *EMPTY*))) ∧

(*0*::*real*) < − *dot* (*cross a1 a2*) *a3* ⟶ (*0*::*real*) < *dot* (*cross a1 a2*) *a4*)) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.aff_gt_1_2_cross_dotr_4point_zero:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ∧ *dot* (*cross a1 a2*) *a3* = (*0*::*real*) ⟶ *dot* (*cross a1 a2*) *a4* = (*0*::*real*))) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.exists_esilon_real:

∀ (*a*::*real*) *b*::*real*. (*0*::*real*) < *a* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *a* − *h* ∗ *b*))

thm Planarity.invariant_cross_dotr_esilon_3piont:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* ((*0*::*real*) < *dot* (*cross a1 a2*) *a3* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross a1 a2*) (*vector_add* (% ((*1*::*real*) − *h*) *v*) (*vector_sub* (% *h u*) *x*)))))) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.invariant_rcross_dot_esilon_3piont:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* ((*0*::*real*) < *dot* (*cross a1 a2*) *a3* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross* (*vector_add* (% ((*1*::*real*) − *h*) *y*) (*vector_sub* (% *h u*) *x*)) *a2*) *a3*)))) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.invariant_crossr_dot_esilon_3piont:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* ((*0*::*real*) < *dot* (*cross a1 a2*) *a3* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ (*0*::*real*) < *dot* (*cross a1* (*vector_add* (% ((*1*::*real*) − *h*) *z*) (*vector_sub* (% *h u*) *x*))) *a3*)))) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.point_in_aff_gt_2_1_change_point_in_aff_gt_1_2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ *IN y* (*aff_gt* (*INSERT* *x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ⟶ *IN u* (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT y EMPTY*))

thm Planarity.pos_in_aff_gt_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *a*::*real*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *a*) *v*) (% *a u*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Planarity.pos_in_aff_gt_2_1_fan:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *a*::*real*. *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *a*) *v*) (% *a u*)) (*aff_gt* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*))

thm Planarity.condition_4point_aff_gt_1_2inter_aff_gt_1_2:

∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *a*::*real*. *LET* (λ*a1*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a2*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a3*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a4*::(*real*, *3*) *cart*. *LET_END* (*LET* (λ*a5*::(*real*, *3*) *cart*. *LET_END* (¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) ∧ ¬ *collinear* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))) ∧ (*0*::*real*) < *a* ∧ *a* < (*1*::*real*) ∧ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ∧ (*0*::*real*) < *dot* (*cross a3 a4*) *a5* ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *a* ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *h*) *u*) (% *h w*)) *EMPTY*)))) ∧ (*0*::*real*) < *dot* (*cross a3 a1*) *a2* ⟶ (∃ *t*>*0*::*real*. *t* < (*1*::*real*) ∧ (∀ *h*::*real*. (*0*::*real*) < *h* ∧ *h* < *t* ⟶ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT y* (*INSERT z EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *h*) *u*) (% *h w*)) *EMPTY*))) ≠ *EMPTY*)))) (*vector_sub w x*))) (*vector_sub u x*))) (*vector_sub v x*))) (*vector_sub z x*))) (*vector_sub y x*)

thm Planarity.exists_dart_leads_into_edge_eq_topological1_component_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ∧ *IN z U* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) ∧ *IN y* (*xfan* (*x*, *V*, *E*)) ∧ *y* ≠ *x* ∧ (∀ *t*::*real*. (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ⟶ *IN* (*vector_add* (% ((*1*::*real*) − *t*) *y*) (% *t z*)) (*yfan* (*x*, *V*, *E*))) ∧ (*0*::*real*) < *dot* (*cross* (*vector_sub v x*) (*vector_sub y x*)) (*vector_sub z x*) ⟶ *dart_leads_into x V E v u* = *U*

thm Planarity.JUTSTKG:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ U::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ U\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow (\exists (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge dart\_leads\_into\ x\ V\ E\ v\ u = U)$

thm Planarity.AFF_GT_3_1:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (INSERT\ w\ EMPTY) \longrightarrow aff\_gt\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (INSERT\ w\ EMPTY) = GSPEC\ (\lambda GEN\%PVAR\%460::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%460\ (\exists (t1::real)\ (t2::real)\ (t3::real)\ t4::real.\ (0::real) < t4 \wedge t1 + (t2 + (t3 + t4)) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (vector\_add\ (\%\ t3\ u)\ (\%\ t4\ w))))\ y)$

thm Planarity.AFF_GT_1_3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%461::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%461\ (\exists (t1::real)\ (t2::real)\ (t3::real)\ t4::real.\ (0::real) < t2 \wedge (0::real) < t3 \wedge (0::real) < t4 \wedge t1 + (t2 + (t3 + t4)) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (vector\_add\ (\%\ t3\ u)\ (\%\ t4\ w))))\ y)$

thm Planarity.AFF_GE_1_3:

$\forall (x::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY))) \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY))) = GSPEC\ (\lambda GEN\%PVAR\%462::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%462\ (\exists (t1::real)\ (t2::real)\ (t3::real)\ t4::real.\ (0::real) \le t2 \wedge (0::real) \le t3 \wedge (0::real) \le t4 \wedge t1 + (t2 + (t3 + t4)) = (1::real) \wedge y = vector\_add\ (\%\ t1\ x)\ (vector\_add\ (\%\ t2\ v)\ (vector\_add\ (\%\ t3\ u)\ (\%\ t4\ w))))\ y)$

thm Planarity.notcoplanar_disjoint:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow x \neq v \wedge x \neq u \wedge x \neq w \wedge v \neq u \wedge v \neq w \wedge u \neq w$

thm Planarity.notcoplanar_disjoints:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow DISJOINT\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (INSERT\ w\ EMPTY) \wedge DISJOINT\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY)))\ (INSERT\ v$

EMPTY) ∧ *DISJOINT* (*INSERT x* (*INSERT w* (*INSERT v EMPTY*))) (*INSERT u EMPTY*) ∧ *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))) ∧ *DISJOINT* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v* (*INSERT w EMPTY*)) ∧ *DISJOINT* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)) ∧ *DISJOINT* (*INSERT x EMPTY*) (*INSERT u* (*INSERT w EMPTY*)) ∧ *DISJOINT* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))

thm Planarity.coplanar_imp_continuous_collinear:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ (∀ *t*::*real*. *t* ≠ (*0*::*real*) ⟶ ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *t*) *u*) (% *t w*)) *EMPTY*))))

thm Planarity.aff_gt_1_3_eq_unions_aff_gt_1_2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))) = *UNIONS* (*GSPEC* (λ*GEN%PVAR%463*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *a*::*real*. *SETSPEC GEN%PVAR%463* ((*0*::*real*) < *a* ∧ *a* < (*1*::*real*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT* (*vector_add* (% ((*1*::*real*) − *a*) *u*) (% *a w*)) *EMPTY*)))))

thm Planarity.aff_gt_1_3_subset_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) (*yfan* (*x*, *V*, *E*))

thm Planarity.aff_gt_1_3_subset_dart_leads_into_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) (*dart_leads_into x V E u w*)

thm Planarity.inter_aff_gt_3_1_is_aff_gt_1_3:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT

w (*INSERT v EMPTY*))) (*INSERT u EMPTY*))) = *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))

thm Planarity.coplanar_cross_dot:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT v1 EMPTY*)))) $\longrightarrow$ *dot* (*cross* (*vector_sub v x*) (*vector_sub u x*)) (*vector_sub v1 x*) $\neq$ (*0*::*real*)

thm Planarity.CRAMER_LEMMA1:

$\forall$ (*A*::((*real*, *?'a*::*type*) *cart*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *k*::*nat*. (*1*::*nat*) $\leq$ *k* $\wedge$ *k* $\leq$ *dimindex HOL_Light_Import.UNIV* $\longrightarrow$ *det* (*lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if j* = *k then* \$ (*matrix_vector_mul A x*) *i else* \$ (\$ *A i*) *j*))) = \$ *x k* $*$ *det A*

thm Planarity.aff_gt_3_1_rep_cross_dot:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\wedge$ (*0*::*real*) < *dot* (*cross* (*vector_sub v x*) (*vector_sub u x*)) (*vector_sub w x*) $\longrightarrow$ *aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*) = *GSPEC* (λ*GEN%PVAR%465*::(*real*, *3*) *cart*. $\exists$ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%465* ((*0*::*real*) < *dot* (*cross* (*vector_sub v x*) (*vector_sub u x*)) (*vector_sub y x*)) *y*)

thm Planarity.OPEN_AFF_GT_1_3:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\longrightarrow$ *HOL_Light_Import.open* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))))

thm Planarity.OPEN_DIFF_AFF_GE:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *HOL_Light_Import.open* (*DIFF HOL_Light_Import.UNIV* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))))

thm Planarity.aff_ge_1_3_eq_unions_aff_ge_1_2_and_aff_gt_1_3:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\longrightarrow$ *aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))) = *HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) (*HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u* (*INSERT w EMPTY*))) (*HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))))))

thm Planarity.cut_aff_gt_1_3_connected:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *connected s* $\wedge$

$\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\wedge$ *IN y s* $\wedge$ *IN z s* $\wedge$ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\wedge$ $\neg$ *IN z* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\longrightarrow$ ($\exists$ *t*::(*real*, *3*) *cart*. *IN t s* $\wedge$ *IN t* (*HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) (*HOL_Light_Import.UNION* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u* (*INSERT w EMPTY*))) (*aff_ge* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))))))

thm Planarity.AFF_GE_SUBSET_XFAN:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\longrightarrow$ *SUBSET* (*aff_ge* (*INSERT x EMPTY*) (*INSERT u* (*INSERT w EMPTY*))) (*xfan* (*x*, *V*, *E*))

thm Planarity.notcoplanar_4point_aff_gt_3_1_not_empty:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\longrightarrow$ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT w* (*INSERT v EMPTY*))) (*INSERT u EMPTY*))) $\neq$ *EMPTY*

thm Planarity.notcoplanar_4point_aff_gt_1_3_not_empty:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\longrightarrow$ *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))) $\neq$ *EMPTY*

thm Planarity.KVQWYDL_lemma1:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\longrightarrow$ *aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*))) = *dart_leads_into x V E u w*

thm Planarity.point_in_yfan_is_not_inv_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*U*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*z*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) $\wedge$ *IN z U* $\wedge$ *IN u V* $\longrightarrow$ *u* $\neq$ *z*

thm Planarity.point_in_aff_ge_1_1:

∀(x::(real, 3) cart) v::(real, 3) cart. x ≠ v ⟶ IN x (aff_ge (INSERT x EMPTY) (INSERT v EMPTY)) ∧ IN v (aff_ge (INSERT x EMPTY) (INSERT v EMPTY))

thm Planarity.POINT_IN_AFF_GE_IMP_IN_EDGE:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) u1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN u1 V ∧ x ≠ u1 ∧ IN u1 (aff_ge (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ⟶ IN u1 (INSERT v (INSERT u EMPTY))

thm Planarity.POINT_IN_CLOSURE_AFF_GT_1_2:

∀(x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. x ≠ v ∧ x ≠ u ∧ v ≠ u ⟶ IN v (closure (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))))

thm Planarity.KVQWYDL_lemma2:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (u1::(real, 3) cart) w1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT w (INSERT v EMPTY)) E ∧ IN (INSERT u1 (INSERT w1 EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ aff_gt (INSERT x EMPTY) (INSERT v (INSERT u (INSERT w EMPTY))) = dart_leads_into x V E u1 w1 ⟶ IN u1 (INSERT v (INSERT u (INSERT w EMPTY)))

thm Planarity.DISJOINT_RW_DART_FAN_SAME_NODE_FAN:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (h1::real) h2::real. FAN (x, V, E) ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT u (INSERT (?w1.0::(real, 3) cart) EMPTY)) E ∧ w ≠ ?w1.0 ⟶ HOL_Light_Import.INTER (rw_dart_fan x V E (x, u, w, sigma_fan x V E u w) (cos h1)) (rw_dart_fan x V E (x, u, ?w1.0, sigma_fan x V E u ?w1.0) (cos h2)) = EMPTY

thm Planarity.condition_unique_by_dart_leads_into:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (w::(real, 3) cart) (u::(real, 3) cart) w1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT u (INSERT w1 EMPTY)) E ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ SUBSET (dart_leads_into x V E u w) (w_dart_fan x V E (x, u, w, sigma_fan x V E u w)) ∧ dart_leads_into x V E u w = dart_leads_into x V E u w1 ⟶ w = w1

thm Planarity.PROPERTIES_TRIANGLE_FAN_lemma1:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \longrightarrow sigma\_fan\ x\ V\ E\ v\ u = w$

thm Planarity.PROPERTIES_TRIANGLE_FAN_lemma2:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \longrightarrow sigma\_fan\ x\ V\ E\ w\ v = u$

thm Planarity.PROPERTIES_TRIANGLE_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \longrightarrow sigma\_fan\ x\ V\ E\ v\ u = w \wedge sigma\_fan\ x\ V\ E\ w\ v = u$

thm Planarity.inter_aff_gt_3_1_is_aff_gt_2_2:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (INSERT\ w\ EMPTY))\ (aff\_gt\ (INSERT\ x\ (INSERT\ u\ (INSERT\ w\ EMPTY)))\ (INSERT\ v\ EMPTY)) = aff\_gt\ (INSERT\ x\ (INSERT\ u\ EMPTY))\ (INSERT\ v\ (INSERT\ w\ EMPTY))$

thm Planarity.condition_edge_in_face_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge IN\ (INSERT\ u1\ (INSERT\ w1\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge u1 = v \wedge aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY))) = dart\_leads\_into\ x\ V\ E\ u1\ w1 \longrightarrow IN\ (u1,\ w1)\ (INSERT\ (v,\ u)\ (INSERT\ (u,\ w)\ (INSERT\ (w,\ v)\ EMPTY)))$

thm Planarity.KVQWYDL_lemma3:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)$

w1::(real, 3) cart. FAN (x, V, E) ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ IN (INSERT w (INSERT v EMPTY)) E ∧ IN (INSERT u1 (INSERT w1 EMPTY)) E ∧ sigma_fan x V E u w = v ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ aff_gt (INSERT x EMPTY) (INSERT v (INSERT u (INSERT w EMPTY))) = dart_leads_into x V E u1 w1 ⟶ IN (u1, w1) (INSERT (v, u) (INSERT (u, w) (INSERT (w, v) EMPTY)))

thm Planarity.CARD_3:

∀ e::?'a::type ⇒ bool. CARD e = (3::nat) ∧ FINITE e ⟶ (∃ (v::?'a::type) (u::?'a::type) w::?'a::type. e = INSERT v (INSERT u (INSERT w EMPTY)) ∧ v ≠ u ∧ u ≠ w ∧ w ≠ v)

thm Planarity.FINITE_FACE_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ⟶ FINITE ds

thm Planarity.condition_f1_fan_in_face_set:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ y = f1_fan x V E y1 ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ d_fan (x, V, E) = d1_fan (x, V, E) ∧ IN y1 ds ⟶ IN y ds

thm Planarity.CARD_FACE_SET_GE_3_FULLY_SURROUNDED_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ⟶ (3::nat) ≤ CARD ds

thm Planarity.CARD_FACE_SET_EQ_3_FULLY_SURROUNDED_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ CARD ds = (3::nat) ⟶ (∃ (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ds = INSERT f1 (INSERT f2 (INSERT f3 EMPTY)) ∧ f2 = f1_fan x V E f1 ∧ f3 = f1_fan x V E f2)

thm Planarity.lemma_CARD_FACE_SET_EQ_3_FULLY_SURROUNDED_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge CARD\ ds = (3::nat) \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge ds = INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)) \longrightarrow f1 = f1\_fan\ x\ V\ E\ f3$

thm Planarity.CARD_FACE_SET_EQ_3_FULLY_SURROUNDED_FAN1:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge CARD\ ds = (3::nat) \longrightarrow (\exists (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ ds = INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)) \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 = f1 \wedge IN\ (INSERT\ (pr2\ f2)\ (INSERT\ (pr2\ f3)\ EMPTY))\ E \wedge IN\ (INSERT\ (pr2\ f3)\ (INSERT\ (pr2\ f1)\ EMPTY))\ E \wedge IN\ (INSERT\ (pr2\ f1)\ (INSERT\ (pr2\ f2)\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ (pr2\ f2)\ (pr2\ f3) = pr2\ f1 \wedge pr2\ f3 = pr3\ f2 \wedge pr2\ f2 = pr3\ f1 \wedge pr2\ f1 = pr3\ f3)$

thm Planarity.KVQWYDL_lemma10:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge CARD\ ds = (3::nat) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (GSPEC\ (\lambda GEN\%PVAR\%468::(real,\ 3)\ cart.\ \exists y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%468\ (IN\ y\ ds)\ (pr2\ y))) = dartset\_leads\_into\_fan\ x\ V\ E\ ds$

thm Planarity.IN_D1_FAN_IMP_EDGE_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ y\ (d\_fan\ (x,\ V,\ E)) \longrightarrow IN\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY))\ E$

thm Planarity.EQ_PAIR_IMP_EQ_4_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ y1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,$

$V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ y\ (d\_fan\ (x, V, E)) \wedge IN\ y1\ (d\_fan\ (x, V, E)) \wedge (pr2\ y, pr3\ y) = (pr2\ y1, pr3\ y1) \longrightarrow y = y1$

thm Planarity.KVQWYDL_lemma30:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge CARD\ ds = (3::nat) \longrightarrow (\forall y::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ y\ (d\_fan\ (x, V, E)) \wedge dartset\_leads\_into\_fan\ x\ V\ E\ ds = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y) \longrightarrow IN\ y\ ds)$

thm Planarity.KVQWYDL:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge CARD\ ds = (3::nat) \longrightarrow aff\_gt\ (INSERT\ x\ EMPTY)\ (GSPEC\ (\lambda GEN\%PVAR\%469::(real, 3)\ cart.\ \exists y::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%469\ (IN\ y\ ds)\ (pr2\ y))) = dartset\_leads\_into\_fan\ x\ V\ E\ ds \wedge (\forall y::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ IN\ y\ (d\_fan\ (x, V, E)) \wedge dartset\_leads\_into\_fan\ x\ V\ E\ ds = dart\_leads\_into\ x\ V\ E\ (pr2\ y)\ (pr3\ y) \longrightarrow IN\ y\ ds)$

thm Planarity.dartset_leads_into_fan_radial:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ r::real.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge CARD\ ds = (3::nat) \wedge (0::real) < r \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)\ (normball\ x\ r))$

thm Planarity.ball_eq_normball:

$\forall (x::(real, ?'a::type)\ cart)\ r::real.\ ball\ (x, r) = normball\ x\ r$

thm Planarity.dartset_leads_into_fan_eventually_radial_norm:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge CARD\ ds = (3::nat) \longrightarrow eventually\_radial\_norm\ x\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)$

thm Planarity.measurable_dartset_leads_into3_fan:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)$

e::real. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ CARD ds = (3::nat) ∧ (0::real) < e ⟶ measurable (HOL_Light_Import.INTER (dartset_leads_into_fan x V E ds) (ball (x, e)))

thm Planarity.CARD_GT1_IMP_AZIM_FAN_EQ_AZIM:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ IN y (d_fan (x, V, E)) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ azim_fan x V E (pr2 y) (pr3 y) = azim x (pr2 y) (pr3 y) (sigma_fan x V E (pr2 y) (pr3 y))

thm Planarity.CARD_GT1_IMP_AZIM_FAN_EQ_DIHV:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN y (d_fan (x, V, E)) ⟶ azim_fan x V E (pr2 y) (pr3 y) = dihV x (pr2 y) (pr3 y) (sigma_fan x V E (pr2 y) (pr3 y))

thm Planarity.solid_of_dartset_leads_into_fan_triangle_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ CARD ds = (3::nat) ⟶ sol x (dartset_leads_into_fan x V E ds) = real_of_nat (2::nat) ∗ pi + sum ds (λy::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. azim_fan x V E (pr2 y) (pr3 y) − pi)

thm Planarity.MOZNWEH:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) e::real. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ CARD ds = (3::nat) ∧ (0::real) < e ⟶ measurable (HOL_Light_Import.INTER (dartset_leads_into_fan x V E ds) (ball (x, e))) ∧ eventually_radial_norm x (dartset_leads_into_fan x V E ds) ∧ sol x (dartset_leads_into_fan x V E ds) = real_of_nat (2::nat) ∗ pi + sum ds (λy::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. azim_fan x V E (pr2 y) (pr3 y) − pi)

thm Hypermap_and_fan.IMAGE_LEMMA:

∀ (f::?'b::type ⇒ ?'a::type) s::?'b::type ⇒ bool. GSPEC (λGEN%PVAR%471::?'a::type. ∃ x::?'b::type. SETSPEC GEN%PVAR%471 (IN x s) (f x)) = IMAGE f s

thm Packing3.CHOICE_LEMMA:

∀ (y::?'a::type) P::?'a::type ⇒ bool. (∃ x::?'a::type. P x) ∧ (∀ x::?'a::type. P x ⟶ x = y) ⟶ (SOME x::?'a::type. P x) = y

thm Hypermap_and_fan.INVERSE_LEMMA:

$\forall (f::?'b::type \Rightarrow ?'a::type)\ y::?'a::type.\ (\exists!x::?'b::type.\ f\,x = y) \longrightarrow f\ (HOL\_Light\_Import.inverse\ f\ y) = y$

thm Hypermap_and_fan.PERMUTES_IMP_INSIDE:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\,f\,s \longrightarrow (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\,x)\ s)$

thm Hypermap_and_fan.RES_RES:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ res\ (res\,f\ s)\ s = res\,f\ s$

thm Hypermap_and_fan.RES_RES2:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ res\ (res\ f\ s)\ t = res\,f\ (HOL\_Light\_Import.INTER\ s\ t)$

thm Hypermap_and_fan.RES_DECOMPOSITION:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\,x)\ s) \longrightarrow f = res\,f\ (DIFF\ HOL\_Light\_Import.UNIV\ s) \circ res\,f\ s$

thm Hypermap_and_fan.RES_EMPTY:

$\forall f::?'a::type \Rightarrow ?'a::type.\ res\,f\ EMPTY = id$

thm Hypermap_and_fan.PERMUTES_IMP_RES_EQ_FUN:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ f\ s \longrightarrow res\ f\ s = f$

thm Hypermap_and_fan.RES_PERMUTES_UNION:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (A::?'a::type \Rightarrow bool)\ B::?'a::type \Rightarrow bool.\ permutes\ f\ A \longrightarrow permutes\ (res\ f\ A)\ (HOL\_Light\_Import.UNION\ A\ B)$

thm Hypermap_and_fan.RES_PERMUTES_DISJOINT_UNIONS:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ c::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ c \longrightarrow permutes\ (res\ f\ t)\ t) \land (\forall (a::?'a::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ IN\ a\ c \land IN\ b\ c \land a \neq b \longrightarrow DISJOINT\ a\ b) \longrightarrow permutes\ (res\ f\ (UNIONS\ c))\ (UNIONS\ c)$

thm Hypermap_and_fan.RES_PERMUTES:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ (\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ s) \land (\forall y::?'a::type.\ IN\ y\ s \longrightarrow (\exists x::?'a::type.\ IN\ x\ s \land y = f\ x)) \land (\forall (x1::?'a::type)\ x2::?'a::type.\ IN\ x1\ s \land IN\ x2\ s \land f\,x1 = f\,x2 \longrightarrow x1 = x2) \longrightarrow permutes\ (res\ f\ s)\ s$

thm Hypermap_and_fan.E_FAN_PAIR_EXT_PERMUTES_DART1_OF_FAN:

$\forall (V::?'a::type \Rightarrow bool)\ E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ permutes\ (e\_fan\_pair\_ext\ (V,\ E))\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.DART1_OF_FAN_EQ_DISJOINT_UNIONS:

$\forall (V::?'a::type \Rightarrow bool)\ E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ (UNIONS\ E)\ V \longrightarrow (\exists\, c::(?'a::type \times ?'a::type \Rightarrow bool) \Rightarrow bool.\ dart1\_of\_fan\ (V,\ E) = UNIONS\ c \wedge (\forall (a::?'a::type \times ?'a::type \Rightarrow bool)\ b::?'a::type \times ?'a::type \Rightarrow bool.\ IN\ a\ c \wedge IN\ b\ c \wedge a \neq b \longrightarrow DISJOINT\ a\ b) \wedge (\forall\, a::?'a::type \times ?'a::type \Rightarrow bool.\ IN\ a\ c \longrightarrow (\exists\, v::?'a::type.\ IN\ v\ V \wedge a = GSPEC\ (\lambda GEN\%PVAR\%474::?'a::type \times ?'a::type.\ \exists\, w::?'a::type.\ SETSPEC\ GEN\%PVAR\%474\ (IN\ w\ (set\_of\_edge\ v\ V\ E))\ (v,\ w)))))$

thm Hypermap_and_fan.N_FAN_PAIR_EXT_PERMUTES_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow permutes\ (n\_fan\_pair\_ext\ (V,\ E))\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.E_N_F_EQ_I:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow e\_fan\_pair\_ext\ (V,\ E) \circ (n\_fan\_pair\_ext\ (V,\ E) \circ f\_fan\_pair\_ext\ (V,\ E)) = id$

thm Hypermap_and_fan.INVERSE_PERMUTES:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (g::?'a::type \Rightarrow ?'a::type)\ s::?'a::type \Rightarrow bool.\ permutes\ f\ s \wedge f \circ g = id \longrightarrow permutes\ g\ s$

thm Hypermap_and_fan.F_FAN_PAIR_EXT_PERMUTES_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow permutes\ (f\_fan\_pair\_ext\ (V,\ E))\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.E_N_F_IN_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow IN\ (e\_fan\_pair\ (V,\ E)\ x)\ (dart1\_of\_fan\ (V,\ E)) \wedge IN\ (n\_fan\_pair\ (V,\ E)\ x)\ (dart1\_of\_fan\ (V,\ E)) \wedge IN\ (f\_fan\_pair\ (V,\ E)\ x)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.INVERSE_F_IN_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow IN\ (HOL\_Light\_Import.inverse\ (f\_fan\_pair\_ext\ (V,\ E))\ x)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.FINITE_DART_OF_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \longrightarrow FINITE\ (dart\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.E_FAN_PAIR_EXT_PERMUTES_DART_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ permutes\ (e\_fan\_pair\_ext\ (V,\ E))\ (dart\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.F_FAN_PAIR_EXT_PERMUTES_DART_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ permutes (f_fan_pair_ext (V, E)) (dart_of_fan (V, E))

thm Hypermap_and_fan.N_FAN_PAIR_EXT_PERMUTES_DART_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ permutes (n_fan_pair_ext (V, E)) (dart_of_fan (V, E))

thm Hypermap_and_fan.HYPERMAP_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ tuple_hypermap (hypermap_of_fan (V, E)) = (dart_of_fan (V, E), e_fan_pair_ext (V, E), n_fan_pair_ext (V, E), f_fan_pair_ext (V, E))

thm Hypermap_and_fan.COMPONENTS_HYPERMAP_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ dart (hypermap_of_fan (V, E)) = dart_of_fan (V, E) ∧ edge_map (hypermap_of_fan (V, E)) = e_fan_pair_ext (V, E) ∧ node_map (hypermap_of_fan (V, E)) = n_fan_pair_ext (V, E) ∧ face_map (hypermap_of_fan (V, E)) = f_fan_pair_ext (V, E)

thm Hypermap_and_fan.INVERSE_F_FAN_PAIR_EXT:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ HOL_Light_Import.inverse (f_fan_pair_ext (V, E)) = e_fan_pair_ext (V, E) ∘ n_fan_pair_ext (V, E)

thm Hypermap_and_fan.INVERSE_F_FAN_PAIR_EXT_EXPLICIT:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ HOL_Light_Import.inverse (f_fan_pair_ext (V, E)) = res (GABS (λf::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart. ∀(v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (sigma_fan (vec (0::nat)) V E v w, v))) (dart1_of_fan (V, E))

thm Hypermap_and_fan.DART_EXISTS:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) v::(real, 3) cart. IN v V ⟶ (∃w::(real, 3) cart. IN (v, w) (dart_of_fan (V, E)))

thm Hypermap_and_fan.E_N_F_DEGENERATE_CASE:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ ¬ IN x (dart1_of_fan (V, E)) ⟶ edge (hypermap_of_fan (V, E)) x = INSERT x EMPTY ∧ node (hypermap_of_fan (V, E)) x = INSERT x EMPTY ∧ face (hypermap_of_fan (V, E)) x = INSERT x EMPTY

thm Hypermap_and_fan.DART1_OF_FAN_SUBSET_DART_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. SUBSET (dart1_of_fan (V, E)) (dart_of_fan (V, E))

thm Hypermap_and_fan.NODE_SUBSET_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow SUBSET\ (node\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.NODE_SUBSET_DART_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart\_of\_fan\ (V,\ E)) \longrightarrow SUBSET\ (node\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (dart\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.FACE_SUBSET_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow SUBSET\ (face\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.FACE_SUBSET_DART_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart\_of\_fan\ (V,\ E)) \longrightarrow SUBSET\ (face\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (dart\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.EDGE_SUBSET_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow SUBSET\ (edge\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.PAIR_IN_DART_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ w)\ (dart\_of\_fan\ (V,\ E)) \longrightarrow IN\ v\ V \wedge IN\ w\ V$

thm Hypermap_and_fan.PAIR_IN_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ w)\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow IN\ v\ V \wedge IN\ w\ V \wedge IN\ w\ (set\_of\_edge\ v\ V\ E) \wedge IN\ v\ (set\_of\_edge\ w\ V\ E)$

thm Hypermap_and_fan.PAIR_IN_DART1_OF_FAN_IMP_NOT_EQ:

$\forall (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ w)\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow v \neq w$

thm Hypermap_and_fan.NOT_IN_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow \neg\ IN\ (v,\ v)\ (dart1\_of\_fan\ (V,\ E))$

thm Hypermap_and_fan.IN_DART_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN x (dart_of_fan (V, E)) ⟶ (∃(v::(real, 3) cart) w::(real, 3) cart. x = (v, w) ∧ IN (v, w) (dart_of_fan (V, E)) ∧ IN v V ∧ IN w V)

thm Hypermap_and_fan.IN_DART1_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN x (dart1_of_fan (V, E)) ⟶ (∃(v::(real, 3) cart) w::(real, 3) cart. x = (v, w) ∧ IN (v, w) (dart1_of_fan (V, E)) ∧ IN v V ∧ IN w V ∧ IN (INSERT v (INSERT w EMPTY)) E ∧ IN w (set_of_edge v V E) ∧ IN v (set_of_edge w V E))

thm Hypermap_and_fan.DART_EQ_UNIONS:

∀H::?'a::type hypermap. dart H = UNIONS (face_set H) ∧ dart H = UNIONS (node_set H) ∧ dart H = UNIONS (edge_set H)

thm Hypermap_and_fan.SUM_SET_OF_ORBITS:

∀(s::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'a::type) g::?'a::type ⇒ real. FINITE s ∧ permutes f s ⟶ sum (set_of_orbits s f) (λy::?'a::type ⇒ bool. sum y g) = sum s g

thm Hypermap_and_fan.DART_SUM_lemma:

∀(H::?'a::type hypermap) g::?'a::type ⇒ real. sum (face_set H) (λf::?'a::type ⇒ bool. sum f g) = sum (dart H) g ∧ sum (node_set H) (λn::?'a::type ⇒ bool. sum n g) = sum (dart H) g

thm Hypermap_and_fan.FINITE_ORBIT_MAP:

∀(s::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'a::type) (x::?'a::type) n::nat. FINITE s ∧ permutes f s ∧ CARD (orbit_map f x) = n ⟶ orbit_map f x = GSPEC (λGEN%PVAR%479::?'a::type. ∃k::nat. SETSPEC GEN%PVAR%479 (k < n) (POWER f k x))

thm Hypermap_and_fan.ORBIT_MAP_CARD_POS:

∀(s::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'a::type) x::?'a::type. FINITE s ∧ permutes f s ⟶ (0::nat) < CARD (orbit_map f x)

thm Hypermap_and_fan.ORBIT_MAP_INJ:

∀(s::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'a::type) (x::?'a::type) (i::nat) (j::nat) k::nat. FINITE s ∧ permutes f s ∧ CARD (orbit_map f x) = k ∧ i < k ∧ j < k ∧ POWER f i x = POWER f j x ⟶ i = j

thm Hypermap_and_fan.INVERSE_ADD_EXPONENTS:

∀(a::nat) (b::nat) (f::?'a::type ⇒ ?'a::type) s::?'a::type ⇒ bool. permutes f s ∧ b ≤ a ⟶ POWER f a ∘ POWER (HOL_Light_Import.inverse f) b = POWER f (a − b) ∧ POWER (HOL_Light_Import.inverse f) b ∘ POWER f a = POWER f (a − b)

thm Hypermap_and_fan.FINITE_ORBIT_MAP_INVERSE:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ (x::?'a::type)\ (n::nat)\ k::nat.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = n \wedge k \leq n \longrightarrow POWER\ (HOL\_Light\_Import.inverse\ f)\ k\ x = POWER\ f\ (n - k)\ x$

thm Hypermap_and_fan.ORBIT_MAP_TRANSLATION:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (k::nat)\ n::nat.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = k \longrightarrow orbit\_map\ f\ x = IMAGE\ (\lambda i::nat.\ POWER\ f\ i\ x)\ (dotdot\ n\ (n + (k - (1::nat))))$

thm Hypermap_and_fan.SUM_ORBIT:

$\forall (P::?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ (k::nat)\ n::nat.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = k \longrightarrow sum\ (orbit\_map\ f\ x)\ P = sum\ (dotdot\ n\ (n + (k - (1::nat))))\ (\lambda i::nat.\ P\ (POWER\ f\ i\ x))$

thm Hypermap_and_fan.ORBIT_MAP_3:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = (3::nat) \longrightarrow orbit\_map\ f\ x = INSERT\ x\ (INSERT\ (f\ x)\ (INSERT\ (f\ (f\ x))\ EMPTY)) \wedge f\ (f\ (f\ x)) = x$

thm Hypermap_and_fan.ORBIT_MAP_2:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = (2::nat) \longrightarrow orbit\_map\ f\ x = INSERT\ x\ (INSERT\ (f\ x)\ EMPTY) \wedge f\ (f\ x) = x$

thm Hypermap_and_fan.ORBIT_MAP_INV_3:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = (3::nat) \longrightarrow f\ x = HOL\_Light\_Import.inverse\ f\ (HOL\_Light\_Import.inverse\ f\ x) \wedge f\ (f\ x) = HOL\_Light\_Import.inverse\ f\ x \wedge HOL\_Light\_Import.inverse\ f\ (HOL\_Light\_Import.inverse\ f\ (HOL\_Light\_Import.inverse\ f\ x)) = x$

thm Hypermap_and_fan.ORBIT_MAP_INV_3_SET:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = (3::nat) \longrightarrow orbit\_map\ f\ x = INSERT\ x\ (INSERT\ (HOL\_Light\_Import.inverse\ f\ (HOL\_Light\_Import.inverse\ f\ x))\ (INSERT\ (HOL\_Light\_Import.inverse\ f\ x)\ EMPTY))$

thm Hypermap_and_fan.ORBIT_MAP_INV_2:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ x::?'a::type.\ FINITE\ s \wedge permutes\ f\ s \wedge CARD\ (orbit\_map\ f\ x) = (2::nat) \longrightarrow f\ x = HOL\_Light\_Import.inverse\ f\ x \wedge HOL\_Light\_Import.inverse\ f\ (HOL\_Light\_Import.inverse\ f\ x) = x$

thm Hypermap_and_fan.ORBIT_MAP_INV_2_SET:

∀(s::?'a::type ⇒ bool) (f::?'a::type ⇒ ?'a::type) x::?'a::type. FINITE s ∧ permutes f s ∧ CARD (orbit_map f x) = (2::nat) ⟶ orbit_map f x = INSERT x (INSERT (HOL_Light_Import.inverse f x) EMPTY)

thm Hypermap_and_fan.FAN_PAIR_FIXED_POINT:

∀(V::(real, 3) cart ⇒ bool) x::(real, 3) cart × (real, 3) cart. IN x (GSPEC (λGEN%PVAR%480::(real, 3) cart × (real, 3) cart. ∃v::(real, 3) cart. SETSPEC GEN%PVAR%480 (IN v V ∧ set_of_edge v V (?E::((real, 3) cart ⇒ bool) ⇒ bool) = EMPTY) (v, v))) ⟶ n_fan_pair_ext (V, ?E) x = x ∧ f_fan_pair (V, ?E) x = x

thm Hypermap_and_fan.CARD_FACE_GT_1:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ (1::nat) < CARD (face (hypermap_of_fan (V, E)) x) ⟶ IN x (dart1_of_fan (V, E))

thm Hypermap_and_fan.LINEAR_FACE:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ CARD (face (hypermap_of_fan (V, E)) (v, w)) = (2::nat) ⟶ face (hypermap_of_fan (V, E)) (v, w) = INSERT (v, w) (INSERT (w, v) EMPTY)

thm Hypermap_and_fan.LINEAR_FACE_2:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ CARD (face (hypermap_of_fan (V, E)) (v, w)) = (2::nat) ⟶ f_fan_pair_ext (V, E) (v, w) = (w, v)

thm Hypermap_and_fan.TRIANGULAR_FACE:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN (v, w) (dart1_of_fan (V, E)) ∧ CARD (face (hypermap_of_fan (V, E)) (v, w)) = (3::nat) ⟶ LET (λw'::(real, 3) cart. LET_END (face (hypermap_of_fan (V, E)) (v, w) = INSERT (v, w) (INSERT (w, w') (INSERT (w', v) EMPTY)) ∧ sigma_fan (vec (0::nat)) V E w w' = v ∧ sigma_fan (vec (0::nat)) V E w' v = w)) (sigma_fan (vec (0::nat)) V E v w)

thm Hypermap_and_fan.IN_FACE_IMP_IN_DART1_OF_FAN:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (x::(real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN x (dart1_of_fan (V, E)) ∧ IN y (face (hypermap_of_fan (V, E)) x) ⟶ IN y (dart1_of_fan (V, E))

thm Hypermap_and_fan.FACE_LAST_POINT:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN (v, w) (dart1_of_fan

$(V, E)) \longrightarrow$ *LET* $(\lambda w'::(real, 3)$ *cart*. *LET_END* (*IN* (w', v) (*face* (*hypermap_of_fan* $(V, E))$ $(v, w))))$ (*sigma_fan* (*vec* (*0*::*nat*)) V E v w)

thm Hypermap_and_fan.IN_DART1_OF_FAN_IMP_CARD_FACE_GT_1:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart1_of_fan* (V, E)) $\longrightarrow$ (*1*::*nat*) $<$ *CARD* (*face* (*hypermap_of_fan* (V, E)) x)

thm Hypermap_and_fan.PLAIN_HYPERMAP_OF_FAN:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FAN* (*vec* (*0*::*nat*), V, E) $\longrightarrow$ *plain_hypermap* (*hypermap_of_fan* (V, E))

thm Hypermap_and_fan.E_HAS_NO_FIXED_POINTS_IN_D1:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart1_of_fan* (V, E)) $\longrightarrow$ *e_fan_pair* (V, E) $x \neq x$

thm Hypermap_and_fan.F_HAS_NO_FIXED_POINTS_IN_D1:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart1_of_fan* (V, E)) $\longrightarrow$ *f_fan_pair* (V, E) $x \neq x$

thm Hypermap_and_fan.UNIQUE_ORBIT:

$\forall$ (s::*?'a*::*type* $\Rightarrow$ *bool*) (f::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) x::*?'a*::*type*. *FINITE* s $\wedge$ *permutes* f s $\wedge$ *IN* x s $\longrightarrow$ ($\exists$!c::*?'a*::*type* $\Rightarrow$ *bool*. *IN* c (*set_of_orbits* s f) $\wedge$ *IN* x c)

thm Hypermap_and_fan.DART_IN_UNIQUE_NODE:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart_of_fan* (V, E)) $\longrightarrow$ ($\exists$!n::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN* n (*node_set* (*hypermap_of_fan* (V, E))) $\wedge$ *IN* x n)

thm Hypermap_and_fan.DART_IN_UNIQUE_FACE:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart_of_fan* (V, E)) $\longrightarrow$ ($\exists$!f::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN* f (*face_set* (*hypermap_of_fan* (V, E))) $\wedge$ *IN* x f)

thm Hypermap_and_fan.DART_IN_UNIQUE_EDGE:

$\forall$ (V::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (E::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) x::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), V, E) $\wedge$ *IN* x (*dart_of_fan* (V, E)) $\longrightarrow$ ($\exists$!e::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN* e (*edge_set* (*hypermap_of_fan* (V, E))) $\wedge$ *IN* x e)

thm Hypermap_and_fan.EDGE_HYPERMAP_OF_FAN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *edge* (*hypermap_of_fan* (*V*, *E*)) (*v*, *w*) = *INSERT* (*v*, *w*) (*INSERT* (*w*, *v*) *EMPTY*)

thm Hypermap_and_fan.N_FAN_PAIR_EXT_IN_DART1_OF_FAN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *SUBSET* (*orbit_map* (*n_fan_pair_ext* (*V*, *E*)) (*v*, *w*)) (*dart1_of_fan* (*V*, *E*))

thm Hypermap_and_fan.N_FAN_PAIR_EXT_POWER:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *n*::*nat*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *POWER* (*n_fan_pair_ext* (*V*, *E*)) *n* (*v*, *w*) = (*v*, *POWER* (*sigma_fan* (*vec* (*0*::*nat*)) *V E v*) *n w*)

thm Hypermap_and_fan.NODE_HYPERMAP_OF_FAN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *node* (*hypermap_of_fan* (*V*, *E*)) (*v*, *w*) = *GSPEC* (λ*GEN%PVAR%481*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *k*::*nat*. *SETSPEC GEN%PVAR%481* ((*0*::*nat*) ≤ *k*) (*v*, *POWER* (*sigma_fan* (*vec* (*0*::*nat*)) *V E v*) *k w*))

thm Hypermap_and_fan.SIGMA_FAN_POWER:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *i*::*nat*. *power_map_points sigma_fan* (*vec* (*0*::*nat*)) *V E v u i* = *POWER* (*sigma_fan* (*vec* (*0*::*nat*)) *V E v*) *i u*

thm Hypermap_and_fan.NODE_HYPERMAP_OF_FAN_lemma:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *u*) (*dart1_of_fan* (*V*, *E*)) ⟶ *node* (*hypermap_of_fan* (*V*, *E*)) (*v*, *u*) = *GSPEC* (λ*GEN%PVAR%482*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%482* ((*0*::*nat*) ≤ *i*) (*v*, *power_map_points sigma_fan* (*vec* (*0*::*nat*)) *V E v u i*))

thm Hypermap_and_fan.NODE_HYPERMAP_OF_FAN_ALT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan* (*V*, *E*)) ⟶ *node* (*hypermap_of_fan* (*V*, *E*)) (*v*, *w*) = *GSPEC* (λ*GEN%PVAR%483*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *u*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%483* (*IN* *u* (*set_of_edge v V E*)) (*v*, *u*))

thm Hypermap_and_fan.CARD_NODE_HYPERMAP_OF_FAN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN* (*v*, *w*) (*dart1_of_fan*

$(V, E)) \longrightarrow CARD\ (node\ (hypermap\_of\_fan\ (V, E))\ (v, w)) = CARD\ (set\_of\_edge\ v\ V\ E)$

thm Hypermap_and_fan.HYPERMAP_OF_FAN_NODE_EQ:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (x::(real, 3)\ cart \times (real, 3)\ cart)\ y::(real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (vec\ (0::nat), V, E) \wedge IN\ x\ (dart\_of\_fan\ (V, E)) \wedge IN\ y\ (dart\_of\_fan\ (V, E)) \wedge fst\ x = fst\ y \longrightarrow node\ (hypermap\_of\_fan\ (V, E))\ x = node\ (hypermap\_of\_fan\ (V, E))\ y$

thm Hypermap_and_fan.FST_NODE_lemma:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ n::(real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ FAN\ (vec\ (0::nat), V, E) \wedge IN\ n\ (node\_set\ (hypermap\_of\_fan\ (V, E))) \longrightarrow (\forall (x::(real, 3)\ cart \times (real, 3)\ cart)\ y::(real, 3)\ cart \times (real, 3)\ cart.\ IN\ x\ n \wedge IN\ y\ n \longrightarrow fst\ x = fst\ y)$

thm Hypermap_and_fan.FAN_NODE_EQ_lemma:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (x::(real, 3)\ cart \times (real, 3)\ cart)\ y::(real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (vec\ (0::nat), V, E) \wedge node\ (hypermap\_of\_fan\ (V, E))\ x = node\ (hypermap\_of\_fan\ (V, E))\ y \longrightarrow fst\ x = fst\ y$

thm Hypermap_and_fan.NODE_SET_AS_IMAGE:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vec\ (0::nat), V, E) \longrightarrow (\exists f::(real, 3)\ cart \Rightarrow (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ (\forall (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ f\ v = f\ w \longrightarrow v = w) \wedge (\forall (v::(real, 3)\ cart)\ x::(real, 3)\ cart \times (real, 3)\ cart.\ IN\ x\ (f\ v) \longrightarrow fst\ x = v) \wedge (\forall v::(real, 3)\ cart.\ fst\ (CHOICE\ (f\ v)) = v) \wedge node\_set\ (hypermap\_of\_fan\ (V, E)) = IMAGE\ f\ V)$

thm Hypermap_and_fan.SIMPLE_HYPERMAP_IMP_FACE_INJ:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ (u::?'a::type)\ v::?'a::type.\ simple\_hypermap\ H \wedge IN\ x\ (dart\ H) \wedge IN\ u\ (node\ H\ x) \wedge IN\ v\ (node\ H\ x) \wedge face\ H\ u = face\ H\ v \longrightarrow u = v$

thm Hypermap_and_fan.SIMPLE_HYPERMAP_lemma:

$\forall (H::?'a::type\ hypermap)\ (x::?'a::type)\ P::?'a::type \Rightarrow bool.\ simple\_hypermap\ H \wedge IN\ x\ (dart\ H) \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%484::?'a::type \Rightarrow bool.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%484\ (IN\ y\ (dart\ H) \wedge P\ y \wedge IN\ y\ (node\ H\ x))\ (face\ H\ y))) = CARD\ (GSPEC\ (\lambda GEN\%PVAR\%485::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%485\ (IN\ y\ (node\ H\ x) \wedge P\ y)\ y))$

thm Hypermap_and_fan.HYPERMAP_OF_FAN_FACE_NODE_INJ:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (f::(real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ (x::(real, 3)\ cart \times (real, 3)\ cart)\ y::(real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (vec\ (0::nat), V, E) \wedge simple\_hypermap\ (hypermap\_of\_fan$

$(V,\ E)) \wedge IN\ f\ (face\_set\ (hypermap\_of\_fan\ (V,\ E))) \wedge IN\ x\ f \wedge IN\ y\ f \wedge fst\ x = fst\ y \longrightarrow x = y$

thm Hypermap_and_fan.NODE_HYPERMAP_OF_FAN_POWER_MAP_POINTS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ u)\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow node\ (hypermap\_of\_fan\ (V,\ E))\ (v,\ u) = GSPEC\ (\lambda GEN\%PVAR\%486::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%486\ (i < CARD\ (set\_of\_edge\ v\ V\ E))\ (v,\ power\_map\_points\ sigma\_fan\ (vec\ (0::nat))\ V\ E\ v\ u\ i))$

thm Hypermap_and_fan.AZIM_I_FAN_EQ_AZIM_DART:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ i::nat.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ u)\ (dart1\_of\_fan\ (V,\ E)) \wedge (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow azim\_i\_fan\ (vec\ (0::nat))\ V\ E\ v\ u\ i = azim\_dart\ (V,\ E)\ (v,\ power\_map\_points\ sigma\_fan\ (vec\ (0::nat))\ V\ E\ v\ u\ i)$

thm Hypermap_and_fan.SUM_lemma:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ u)\ (dart1\_of\_fan\ (V,\ E)) \wedge (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E) \longrightarrow sum\ (dotdot\ (0::nat)\ (CARD\ (set\_of\_edge\ v\ V\ E) - (1::nat)))\ (azim\_i\_fan\ (vec\ (0::nat))\ V\ E\ v\ u) = sum\ (node\ (hypermap\_of\_fan\ (V,\ E))\ (v,\ u))\ (azim\_dart\ (V,\ E))$

thm Hypermap_and_fan.SUM_AZIM_DART:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ x\ (dart\_of\_fan\ (V,\ E)) \longrightarrow sum\ (node\ (hypermap\_of\_fan\ (V,\ E))\ x)\ (azim\_dart\ (V,\ E)) = real\_of\_nat\ (2::nat) * pi$

thm Hypermap_and_fan.SUM_BOUND_LT_ALT:

$\forall (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow real)\ (b::real)\ n::nat.\ FINITE\ s \wedge CARD\ s \leq n \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow f\ x \leq b) \wedge (\exists x::?'a::type.\ IN\ x\ s \wedge f\ x < b) \wedge (0::real) \leq b \longrightarrow sum\ s\ f < real\_of\_nat\ n * b$

thm Hypermap_and_fan.FULLY_SURROUNDED:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow fully\_surrounded\ (V,\ E) = (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow surrounded\_node\ (V,\ E)\ v)$

thm Hypermap_and_fan.SURROUNDED_IMP_CARD_SET_OF_EDGE_GE_3:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ v::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ v\ V \wedge surrounded\_node\ (V,\ E)\ v \longrightarrow (3::nat) \leq CARD\ (set\_of\_edge\ v\ V\ E)$

thm Hypermap_and_fan.SURROUNDED_IMP_IN_DART1_OF_FAN:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN (v, w) (dart_of_fan (V, E)) ∧ surrounded_node (V, E) v ⟶ IN (v, w) (dart1_of_fan (V, E))

thm Hypermap_and_fan.SURROUNDED_IMP_CARD_NODE_GE_3:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (vec (0::nat), V, E) ∧ IN (v, w) (dart1_of_fan (V, E)) ∧ surrounded_node (V, E) v ⟶ (3::nat) ≤ CARD (node (hypermap_of_fan (V, E)) (v, w))

thm Hypermap_and_fan.CARD_SET_OF_EDGE_GT_1_IMP_CARD_FACE_GE_3:

∀ (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ (∀ x::(real, 3) cart × (real, 3) cart. IN x (dart_of_fan (V, E)) ⟶ (3::nat) ≤ CARD (face (hypermap_of_fan (V, E)) x))

thm Hypermap_and_fan.FULLY_SURROUNDED_IMP_CARD_FACE_GE_3:

∀ (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ∧ fully_surrounded (V, E) ⟶ (∀ x::(real, 3) cart × (real, 3) cart. IN x (dart_of_fan (V, E)) ⟶ (3::nat) ≤ CARD (face (hypermap_of_fan (V, E)) x))

thm Hypermap_and_fan.FULLY_SURROUNDED_NODE_DECOMPOSITION:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ fully_surrounded (V, E) ∧ IN x (dart_of_fan (V, E)) ⟶ LET (λH::((real, 3) cart × (real, 3) cart) hypermap. LET_END (LET (λA::(real, 3) cart × (real, 3) cart ⇒ bool. LET_END (LET (λB::(real, 3) cart × (real, 3) cart ⇒ bool. LET_END (LET (λC::(real, 3) cart × (real, 3) cart ⇒ bool. LET_END (LET (λD::(real, 3) cart × (real, 3) cart ⇒ bool. LET_END (node H x = HOL_Light_Import.UNION A D ∧ DISJOINT A D ∧ D = HOL_Light_Import.UNION B C ∧ DISJOINT B C ∧ FINITE D ∧ FINITE A)) (GSPEC (λGEN%PVAR%494::(real, 3) cart × (real, 3) cart. ∃ y::(real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%494 (IN y (node H x) ∧ (4::nat) ≤ CARD (face H y)) y)))) (GSPEC (λGEN%PVAR%493::(real, 3) cart × (real, 3) cart. ∃ y::(real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%493 (IN y (node H x) ∧ (5::nat) ≤ CARD (face H y)) y)))) (GSPEC (λGEN%PVAR%492::(real, 3) cart × (real, 3) cart. ∃ y::(real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%492 (IN y (node H x) ∧ CARD (face H y) = (4::nat)) y)))) (GSPEC (λGEN%PVAR%491::(real, 3) cart × (real, 3) cart. ∃ y::(real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%491 (IN y (node H x) ∧ CARD (face H y) = (3::nat)) y)))) (hypermap_of_fan (V, E))

thm Hypermap_and_fan.SUM_AZIM_DART_FULLY_SURROUNDED:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V, E) ∧ fully_surrounded (V, E) ∧ IN x (dart_of_fan (V, E)) ⟶ LET (λH::((real, 3) cart × (real, 3) cart) hypermap.

LET_END (*LET* (λ*A*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *LET_END* (*LET* (λ*B*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *LET_END* (*sum A* (*azim_dart* (*V*, *E*)) + *sum B* (*azim_dart* (*V*, *E*)) = *real_of_nat* (*2*::*nat*) ∗ *pi*)) (*GSPEC* (λ*GEN%PVAR%498*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%498* (*IN y* (*node H x*) ∧ (*4*::*nat*) ≤ *CARD* (*face H y*)) *y*)))) (*GSPEC* (λ*GEN%PVAR%497*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%497* (*IN y* (*node H x*) ∧ *CARD* (*face H y*) = (*3*::*nat*)) *y*)))) (*hypermap_of_fan* (*V*, *E*))

thm DEF_no_loops:

no_loops = (λ*_2996047*::*?'a*::*type hypermap*. ∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*edge _2996047 y*) ∧ *IN x* (*node _2996047 y*) ⟶ *x* = *y*)

thm Hypermap_and_fan.no_loops:

∀ *H*::*?'a*::*type hypermap*. *no_loops H* = (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*edge H y*) ∧ *IN x* (*node H y*) ⟶ *x* = *y*)

thm DEF_is_no_double_joints:

is_no_double_joints = (λ*_2996052*::*?'a*::*type hypermap*. ∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart _2996052*) ∧ *IN y* (*node _2996052 x*) ∧ *IN* (*edge_map _2996052 y*) (*node _2996052* (*edge_map _2996052 x*)) ⟶ *x* = *y*)

thm Hypermap_and_fan.is_no_double_joints:

∀ *H*::*?'a*::*type hypermap*. *is_no_double_joints H* = (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. *IN x* (*dart H*) ∧ *IN y* (*node H x*) ∧ *IN* (*edge_map H y*) (*node H* (*edge_map H x*)) ⟶ *x* = *y*)

thm Hypermap_and_fan.HYPERMAP_OF_FAN_EDGE_NONDEGENERATE:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *fully_surrounded* (*V*, *E*) ⟶ *is_edge_nondegenerate* (*hypermap_of_fan* (*V*, *E*))

thm Hypermap_and_fan.HYPERMAP_OF_FAN_NO_LOOPS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ⟶ *no_loops* (*hypermap_of_fan* (*V*, *E*))

thm Hypermap_and_fan.HYPERMAP_OF_FAN_NO_DOUBLE_JOINTS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ⟶ *is_no_double_joints* (*hypermap_of_fan* (*V*, *E*))

thm Hypermap_and_fan.AZIM_DART_POS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ *IN x* (*dart_of_fan* (*V*, *E*)) ⟶ (*0*::*real*) < *azim_dart* (*V*, *E*) *x*

thm Hypermap_and_fan.DART1_NOT_COLLINEAR:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ w)\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Hypermap_and_fan.DART1_NOT_COLLINEAR_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ (v,\ w)\ (dart1\_of\_fan\ (V,\ E)) \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ (sigma\_fan\ (vec\ (0::nat))\ V\ E\ v\ w)\ EMPTY)))$

thm DEF_conforming_bijection_fan:

$conforming\_bijection\_fan = (\lambda\_2996107::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall s::(real,\ 3)\ cart \Rightarrow bool.\ IN\ s\ (topological\_component\_yfan\ (fst\ \_2996107,\ fst\ (snd\ \_2996107),\ snd\ (snd\ \_2996107))) \longrightarrow (\exists!f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (fst\ \_2996107,\ fst\ (snd\ \_2996107),\ snd\ (snd\ \_2996107)))) \wedge s = dartset\_leads\_into\_fan\ (fst\ \_2996107)\ (fst\ (snd\ \_2996107))\ (snd\ (snd\ \_2996107))\ f))$

thm Conforming.conforming_bijection_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ conforming\_bijection\_fan\ (x,\ V,\ E) = (\forall s::(real,\ 3)\ cart \Rightarrow bool.\ IN\ s\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow (\exists!f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge s = dartset\_leads\_into\_fan\ x\ V\ E\ f))$

thm DEF_conforming_half_space_fan:

$conforming\_half\_space\_fan = (\lambda\_2996120::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (fst\ \_2996120,\ fst\ (snd\ \_2996120),\ snd\ (snd\ \_2996120)))) \longrightarrow dartset\_leads\_into\_fan\ (fst\ \_2996120)\ (fst\ (snd\ \_2996120))\ (snd\ (snd\ \_2996120))\ f = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%499::(real,\ 3)\ cart \Rightarrow bool.\ \exists y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%499\ (IN\ y\ f)\ (aff\_gt\ (INSERT\ (fst\ \_2996120)\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY)))\ (INSERT\ (pr3\ (f1\_fan\ (fst\ \_2996120)\ (fst\ (snd\ \_2996120))\ (snd\ (snd\ \_2996120))\ y))\ EMPTY)))))$

thm Conforming.conforming_half_space_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ conforming\_half\_space\_fan\ (x,\ V,\ E) = (\forall f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \longrightarrow dartset\_leads\_into\_fan\ x\ V\ E\ f = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%499::(real,\ 3)\ cart \Rightarrow bool.\ \exists y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)$

cart. SETSPEC GEN%PVAR%499 (IN y f) (aff_gt (INSERT x (INSERT (pr2 y) (INSERT (pr3 y) EMPTY))) (INSERT (pr3 (f1_fan x V E y)) EMPTY)))))

thm DEF_conforming_solid_angle_fan:

conforming_solid_angle_fan = (λ_2996133::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). ∀ f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN f (face_set (hypermap1_of_fanx (fst _2996133, fst (snd _2996133), snd (snd _2996133)))) ⟶ LET (λ U::(real, 3) cart ⇒ bool. LET_END ((∀ r::real. measurable (HOL_Light_Import.INTER (ball (fst _2996133, r)) U)) ∧ eventually_radial (fst _2996133) U ∧ sol (fst _2996133) U = real_of_nat (2::nat) ∗ pi + sum f (λy::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. azim_fan (fst _2996133) (fst (snd _2996133)) (snd (snd _2996133)) (pr2 y) (pr3 y) − pi))) (dartset_leads_into_fan (fst _2996133) (fst (snd _2996133)) (snd (snd _2996133)) f))

thm Conforming.conforming_solid_angle_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. conforming_solid_angle_fan (x, V, E) = (∀ f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN f (face_set (hypermap1_of_fanx (x, V, E))) ⟶ LET (λ U::(real, 3) cart ⇒ bool. LET_END ((∀ r::real. measurable (HOL_Light_Import.INTER (ball (x, r)) U)) ∧ eventually_radial x U ∧ sol x U = real_of_nat (2::nat) ∗ pi + sum f (λy::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. azim_fan x V E (pr2 y) (pr3 y) − pi))) (dartset_leads_into_fan x V E f))

thm DEF_conforming_diagonal_fan:

conforming_diagonal_fan = (λ_2996146::(real, 3) cart × ((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). ∀ (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) z::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN f (face_set (hypermap1_of_fanx (fst _2996146, fst (snd _2996146), snd (snd _2996146)))) ∧ IN y f ∧ IN z f ∧ y ≠ z ⟶ ¬ collinear (INSERT (fst _2996146) (INSERT (pr2 y) (INSERT (pr2 z) EMPTY))) ∧ (y = f1_fan (fst _2996146) (fst (snd _2996146)) (snd (snd _2996146)) z ∨ z = f1_fan (fst _2996146) (fst (snd _2996146)) (snd (snd _2996146)) y ∨ SUBSET (aff_gt (INSERT (fst _2996146) EMPTY) (INSERT (pr2 y) (INSERT (pr2 z) EMPTY))) (dartset_leads_into_fan (fst _2996146) (fst (snd _2996146)) (snd (snd _2996146)) f)))

thm Conforming.conforming_diagonal_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. conforming_diagonal_fan (x, V, E) = (∀ (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) z::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN f (face_set (hypermap1_of_fanx (x, V, E))) ∧ IN y f ∧ IN z f ∧ y ≠ z ⟶ ¬ collinear (INSERT x (INSERT (pr2 y) (INSERT

$(pr2\ z)\ EMPTY))) \wedge (y = f1\_fan\ x\ V\ E\ z \vee z = f1\_fan\ x\ V\ E\ y \vee SUBSET\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ (pr2\ y)\ (INSERT\ (pr2\ z)\ EMPTY)))\ (dartset\_leads\_into\_fan\ x\ V\ E\ f)))$

thm DEF_conforming_fan:

$conforming\_fan = (\lambda\_2996159::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ (\forall v::(real,\ 3)\ cart.\ IN\ v\ (fst\ (snd\ \_2996159)) \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ (fst\ (snd\ \_2996159))\ (snd\ (snd\ \_2996159)))) \wedge fan80\ (fst\ \_2996159, fst\ (snd\ \_2996159), snd\ (snd\ \_2996159)) \wedge conforming\_bijection\_fan\ (fst\ \_2996159, fst\ (snd\ \_2996159), snd\ (snd\ \_2996159)) \wedge conforming\_half\_space\_fan\ (fst\ \_2996159, fst\ (snd\ \_2996159), snd\ (snd\ \_2996159)) \wedge conforming\_solid\_angle\_fan\ (fst\ \_2996159, fst\ (snd\ \_2996159), snd\ (snd\ \_2996159)) \wedge conforming\_diagonal\_fan\ (fst\ \_2996159,\ fst\ (snd\ \_2996159),\ snd\ (snd\ \_2996159)))$

thm Conforming.conforming_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ conforming\_fan\ (x,\ V,\ E) = ((\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge conforming\_bijection\_fan\ (x,\ V,\ E) \wedge conforming\_half\_space\_fan\ (x,\ V,\ E) \wedge conforming\_solid\_angle\_fan\ (x,\ V,\ E) \wedge conforming\_diagonal\_fan\ (x,\ V,\ E))$

thm DEF_N_FAN:

$N\_FAN = (\lambda\_2996172::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ nsum\ (face\_set\ (hypermap1\_of\_fanx\ (fst\ \_2996172,\ fst\ (snd\ \_2996172),\ snd\ (snd\ \_2996172))))\ (\lambda f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ CARD\ f - (3::nat)))$

thm Conforming.N_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ N\_FAN\ (x,\ V,\ E) = nsum\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E)))\ (\lambda f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ CARD\ f - (3::nat))$

thm DEF_minimally_nonconforming_fan:

$minimally\_nonconforming\_fan = (\lambda\_2996185::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool).\ FAN\ (fst\ \_2996185,\ fst\ (snd\ \_2996185),\ snd\ (snd\ \_2996185)) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ (fst\ (snd\ \_2996185)) \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ (fst\ (snd\ \_2996185))\ (snd\ (snd\ \_2996185)))) \wedge fan80\ (fst\ \_2996185, fst\ (snd\ \_2996185), snd\ (snd\ \_2996185)) \wedge \neg\ conforming\_fan\ (fst\ \_2996185, fst\ (snd\ \_2996185), snd\ (snd\ \_2996185)) \wedge (\forall E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (fst\ \_2996185, fst\ (snd\ \_2996185),\ E1) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ (fst\ (snd\ \_2996185)) \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ (fst\ (snd\ \_2996185))\ E1)) \wedge fan80\ (fst\ \_2996185,\ fst\ (snd\ \_2996185),\ E1) \wedge N\_FAN\ (fst\ \_2996185,\ fst\ (snd\ \_2996185),\ E1) < N\_FAN\ (fst\ \_2996185,\ fst\ (snd\ \_2996185),\ snd\ (snd\ \_2996185)) \longrightarrow conforming\_fan\ (fst\ \_2996185,\ fst\ (snd\ \_2996185),\ E1)))$

thm Conforming.minimally_nonconforming_fan:

∀ (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*x*::(*real*, *3*) *cart*) *V*::(*real*, *3*) *cart* ⇒ *bool*. *minimally_nonconforming_fan* (*x*, *V*, *E*) = (*FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)))

thm Conforming.GINGUAP:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ∧ *IN* (*?ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ *convex* (*dartset_leads_into_fan x V E ?ds*)

thm Conforming.fully_surrounded_imp_aff_gt_3_1_of_edge_eq_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *aff_gt* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) (*INSERT* (*sigma_fan x V E v w*) *EMPTY*) = *aff_gt* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) (*INSERT* (*inverse1_sigma_fan x V E w v*) *EMPTY*)

thm Conforming.IMAGE_F1_IN_FACE_IMP_IN_FACE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y ds* ∧ *IN y1* (*d_fan* (*x*, *V*, *E*)) ∧ *f1_fan x V E y1* = *y* ⟶ *IN y1 ds*

thm Conforming.IMAGE_F1_POWER_IN_FACE_IMP_IN_FACE:

∀ (*m*::*nat*) (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y ds* ∧ *IN y1* (*d_fan* (*x*, *V*, *E*)) ∧ *POWER* (*f1_fan x V E*) *m y1* = *y* ⟶ *IN y1 ds*

thm Conforming.REP_OF_INVERSE1_SIGMA_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒

bool) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y ds* ⟶ *f1_fan x V E* (*x*, *sigma_fan x V E* (*pr2 y*) (*pr3 y*), *pr2 y*, *sigma_fan x V E* (*sigma_fan x V E* (*pr2 y*) (*pr3 y*)) (*pr2 y*)) = *y*

thm Conforming.REP_OF_INVERSE1_SIGMA_FAN_IN_D_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y ds* ⟶ *IN* (*x*, *sigma_fan x V E* (*pr2 y*) (*pr3 y*), *pr2 y*, *sigma_fan x V E* (*sigma_fan x V E* (*pr2 y*) (*pr3 y*)) (*pr2 y*)) (*d_fan* (*x*, *V*, *E*))

thm Conforming.DARTSET_LEADS_INTO_SUBSET_WDART_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y ds* ∧ *conforming_fan* (*x*, *V*, *E*) ⟶ *SUBSET* (*dartset_leads_into_fan x V E ds*) (*w_dart_fan x V E y*)

thm Conforming.power_map_points_edge_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *n*::*nat*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ⟶ *IN* (*INSERT v* (*INSERT* (*power_map_points sigma_fan x V E v w n*) *EMPTY*)) *E*

thm Conforming.SRPRNPL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ⟶ *simple_hypermap* (*hypermap1_of_fanx* (*x*, *V*, *E*))

thm Conforming.N_FAN_GE_0:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ (*0*::*nat*) ≤ *N_FAN* (*x*, *V*, *E*)

thm Conforming.NSUM_EQ_0_IFF:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *f*::*?'a*::*type* ⇒ *nat*. *FINITE s* ⟶ (*nsum s f* = (*0*::*nat*)) = (∀ *x*::*?'a*::*type*. *IN x s* ⟶ *f x* = (*0*::*nat*))

thm Conforming.N_FAN_EQ_0_IMP_CARD_FACE_EQ_3:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒

bool. FAN (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *N_FAN* (*x*, *V*, *E*) = (*0*::*nat*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ *CARD ds* = (*3*::*nat*)

thm Conforming.version_JUTSTKG:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN U* (*topological_component_yfan* (*x*, *V*, *E*)) ⟶ (∃ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *dartset_leads_into_fan x V E f* = *U*)

thm Conforming.measurable_dartset_leads_into30_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *e*::*real*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *CARD ds* = (*3*::*nat*) ⟶ *measurable* (*HOL_Light_Import.INTER* (*dartset_leads_into_fan x V E ds*) (*ball* (*x*, *e*)))

thm Conforming.DWFBRQY:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::*?'a*::*type*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *N_FAN* (*x*, *V*, *E*) = (*0*::*nat*) ⟶ *conforming_fan* (*x*, *V*, *E*)

thm Conforming.NEGLIGIBLE_AFF_3:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *negligible* (*aff* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))))

thm Conforming.NEGLIGIBLE_AFF_GE_2_1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *negligible* (*aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT u EMPTY*))

thm Conforming.NEGLIGIBLE_AFF_GE_1_2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *negligible* (*aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Conforming.NEGLIGIBLE_AFF_GT_1_2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *negligible* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*)))

thm Conforming.MEASURE_AFF_3:

∀ (x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. HOL_Light_Import.measure (aff (INSERT x (INSERT v (INSERT u EMPTY)))) = (0::real)

thm Conforming.MEASURE_AFF_GT_2_1:

∀ (x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ HOL_Light_Import.measure (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) = (0::real)

thm Conforming.NEGLIGIBLE_AFF_3_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. negligible (HOL_Light_Import.INTER (aff (INSERT x (INSERT v (INSERT u EMPTY)))) (normball x r))

thm Conforming.NEGLIGIBLE_AFF_GT_1_2_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ negligible (HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (normball x r))

thm Conforming.MEASURE_AFF_3_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. HOL_Light_Import.measure (HOL_Light_Import.INTER (aff (INSERT x (INSERT v (INSERT u EMPTY)))) (normball x r)) = (0::real)

thm Conforming.MEASURE_AFF_GT_2_1_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ HOL_Light_Import.measure (HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (normball x r)) = (0::real)

thm Conforming.HAS_MEASURE_AFF_3_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. has_measure (HOL_Light_Import.INTER (aff (INSERT x (INSERT v (INSERT u EMPTY)))) (normball x r)) (0::real)

thm Conforming.HAS_MEASURE_AFF_GT_1_2_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ has_measure (HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (normball x r)) (0::real)

thm Conforming.MEASURABLE_AFF_GT_2_1_INTER_BALL:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) r::real. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ measurable (HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) (normball x r))

thm Conforming.XFAN_EQ_UNIONS_AFF_GE_1_2:

$\forall (x::(real,\ ?'b::type)\ cart)\ (V::?'a::type)\ E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool.\ xfan\ (x,\ V,\ E) = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%505::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \exists\, y::(real,\ ?'b::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%505\ (\exists\, e::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ e\ E \wedge y = aff\_ge\ (INSERT\ x\ EMPTY)\ e)\ y))$

thm Conforming.NEGLIGIBLE_XFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \longrightarrow negligible\ (xfan\ (x,\ V,\ E))$

thm Conforming.NEGLIGIBLE_XFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ r::real.\ FAN\ (x,\ V,\ E) \longrightarrow negligible\ (HOL\_Light\_Import.INTER\ (xfan\ (x,\ V,\ E))\ (normball\ x\ r))$

thm Conforming.MEASURE_XFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \longrightarrow HOL\_Light\_Import.measure\ (xfan\ (x,\ V,\ E)) = (0::real)$

thm Conforming.HAS_MEASURE_XFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \longrightarrow has\_measure\ (xfan\ (x,\ V,\ E))\ (0::real)$

thm Conforming.MEASURE_XFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ r::real.\ FAN\ (x,\ V,\ E) \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (xfan\ (x,\ V,\ E))\ (normball\ x\ r)) = (0::real)$

thm Conforming.HAS_MEASURE_XFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ r::real.\ FAN\ (x,\ V,\ E) \longrightarrow has\_measure\ (HOL\_Light\_Import.INTER\ (xfan\ (x,\ V,\ E))\ (normball\ x\ r))\ (0::real)$

thm Conforming.MEASURABLE_BALL_INTER_UNIV:

$\forall (x::(real,\ 3)\ cart)\ r::real.\ measurable\ (HOL\_Light\_Import.INTER\ HOL\_Light\_Import.UNIV\ (normball\ x\ r))$

thm Conforming.MEASURE_YFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ r::real.\ FAN\ (x,\ V,\ E) \wedge (0::real) \le r \longrightarrow HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (yfan\ (x,\ V,\ E))\ (normball\ x\ r)) = real\_of\_nat\ (4::nat)\ /\ real\_of\_nat\ (3::nat) * (pi * r^{3::nat})$

thm MEASURABLE_RULES_conjunct6:

$\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart \Rightarrow bool.\ measurable\ s \wedge measurable\ t \longrightarrow measurable\ (DIFF\ s\ t)$

thm Conforming.MESURABLE_YFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ r::real.\ FAN\ (x,\ V,\ E) \wedge (0::real) \leq r \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ (yfan\ (x,\ V,\ E))\ (normball\ x\ r))$

thm Conforming.RADIAL_DIFF:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (A::(real,\ ?'a::type)\ cart \Rightarrow bool)\ B::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ v0\ A \wedge radial\_norm\ r\ v0\ B \wedge SUBSET\ A\ B \longrightarrow radial\_norm\ r\ v0\ (DIFF\ B\ A)$

thm Conforming.RADIAL_UNION:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ (A::(real,\ ?'a::type)\ cart \Rightarrow bool)\ B::(real,\ ?'a::type)\ cart \Rightarrow bool.\ radial\_norm\ r\ v0\ A \wedge radial\_norm\ r\ v0\ B \longrightarrow radial\_norm\ r\ v0\ (HOL\_Light\_Import.UNION\ A\ B)$

thm Conforming.RADIAL_EMPTY:

$\forall (r::real)\ v0::(real,\ ?'a::type)\ cart.\ radial\_norm\ r\ v0\ EMPTY$

thm Conforming.RADIAL_UNIONS:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow radial\_norm\ r\ v0\ s) \longrightarrow radial\_norm\ r\ v0\ (UNIONS\ f)$

thm Conforming.RADIAL_UNIV:

$\forall (r::real)\ x::(real,\ ?'a::type)\ cart.\ (0::real) < r \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ HOL\_Light\_Import.UNIV\ (normball\ x\ r))$

thm Conforming.RADIAL_AFF_GE_1_2:

$\forall (x::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ r::real.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ EMPTY)) \wedge (0::real) < r \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (normball\ x\ r))$

thm Conforming.RADIAL_AFF_GT_3_1:

$\forall (x::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ r::real.\ DISJOINT\ (INSERT\ x\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (INSERT\ w\ EMPTY) \wedge (0::real) < r \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ x\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (INSERT\ w\ EMPTY))\ (normball\ x\ r))$

thm Conforming.RADIAL_INTERS:

$\forall (r::real)\ (v0::(real,\ ?'a::type)\ cart)\ f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow radial\_norm\ r\ v0\ (HOL\_Light\_Import.INTER\ s\ (normball\ v0\ r))) \wedge (0::real) < r \longrightarrow radial\_norm\ r\ v0\ (HOL\_Light\_Import.INTER\ (INTERS\ f)\ (normball\ v0\ r))$

thm Conforming.XFAN_INTER_BALL_UNIONS:

∀ (*x*::(*real*, *?'b*::*type*) *cart*) (*V*::*?'a*::*type*) *E*::((*real*, *?'b*::*type*) *cart* ⇒ *bool*) ⇒ *bool*. *HOL_Light_Import.INTER* (*xfan* (*x*, *V*, *E*)) (*normball x* (*?r*::*real*)) = *UNIONS* (*GSPEC* (λ*GEN%PVAR%507*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%507* (∃ *e*::(*real*, *?'b*::*type*) *cart* ⇒ *bool*. *IN e E* ∧ *y* = *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) *e*) (*normball x ?r*)) *y*))

thm Conforming.RADIAL_XFAN_INTER_BALL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *r*::*real*. *FAN* (*x*, *V*, *E*) ∧ (*0*::*real*) < *r* ⟶ *radial_norm r x* (*HOL_Light_Import.INTER* (*xfan* (*x*, *V*, *E*)) (*normball x r*))

thm Conforming.RADIAL_NORM_YFAN_INTER_BALL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *r*::*real*. *FAN* (*x*, *V*, *E*) ∧ (*0*::*real*) < *r* ⟶ *radial_norm r x* (*HOL_Light_Import.INTER* (*yfan* (*x*, *V*, *E*)) (*normball x r*))

thm Conforming.SOLID_ANGLE_YFAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ *sol x* (*yfan* (*x*, *V*, *E*)) = *real_of_nat* (*4*::*nat*) ∗ *pi*

thm Conforming.SUM_SOL_IN_TOPOLOGICAL_COMPONENET_EQ_IN_FACE_SET:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ⟶ *sum* (*topological_component_yfan* (*x*, *V*, *E*)) (*sol x*) = *sum* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *sol x* (*dartset_leads_into_fan x V E f*))

thm Conforming.SOL_EMPTY:

∀ *x*::(*real*, *3*) *cart*. *sol x EMPTY* = (*0*::*real*)

thm MEASURABLE_RULES_conjunct4:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) *t*::(*real*, *3*) *cart* ⇒ *bool*. *measurable s* ∧ *measurable t* ⟶ *measurable* (*HOL_Light_Import.UNION s t*)

thm Conforming.SOL_DISJOINT_UNION:

∀ (*x*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*t*::(*real*, *3*) *cart* ⇒ *bool*) *r*::*real*. (*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import.INTER s* (*normball x r*)) ∧ *measurable* (*HOL_Light_Import.INTER t* (*normball x r*)) ∧ *DISJOINT s t* ∧ *radial_norm r x* (*HOL_Light_Import.INTER s* (*normball x r*)) ∧ *radial_norm r x* (*HOL_Light_Import.INTER t* (*normball x r*)) ⟶ *sol x* (*HOL_Light_Import.UNION s t*) = *sol x s* + *sol x t*

thm Conforming.UNIONS_INTER:

$\forall (f::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (UNIONS\ f)\ t = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%509::?'a::type \Rightarrow bool.\ \exists s::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%509\ (IN\ s\ f)\ (HOL\_Light\_Import.INTER\ s\ t)))$

thm Conforming.UNIONS_INTER1:

$\forall (f::(?'a::type \Rightarrow bool) \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (UNIONS\ f)\ t = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%511::?'a::type \Rightarrow bool.\ \exists y::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%511\ (\exists s::?'a::type \Rightarrow bool.\ IN\ s\ f \wedge y = HOL\_Light\_Import.INTER\ s\ t)\ y))$

thm Conforming.SOL_UNIONS:

$\forall (r::real)\ (x::(real,\ 3)\ cart)\ f::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge (0::real) < r \wedge (\forall s::(real,\ 3)\ cart \Rightarrow bool.\ IN\ s\ f \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ s\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ s\ (normball\ x\ r))) \wedge (\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart \Rightarrow bool.\ IN\ s\ f \wedge IN\ t\ f \wedge s \neq t \longrightarrow DISJOINT\ s\ t) \longrightarrow sol\ x\ (UNIONS\ f) = sum\ f\ (sol\ x)$

thm Conforming.BOUNDED_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \longrightarrow (\forall f::(real,\ 3)\ cart \Rightarrow bool.\ IN\ f\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow bounded\ (HOL\_Light\_Import.INTER\ f\ (normball\ x\ (?r::real))))$

thm Conforming.OPEN_AFF_GT_3_1:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ (INSERT\ w\ EMPTY)))) \longrightarrow HOL\_Light\_Import.open\ (aff\_gt\ (INSERT\ x\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (INSERT\ w\ EMPTY))$

thm Conforming.EQ_SET_THM:

$\forall (f'::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ GSPEC\ (\lambda GEN\%PVAR\%514::?'a::type.\ \exists y::?'b::type.\ SETSPEC\ GEN\%PVAR\%514\ (IN\ y\ f')\ (f\ y)) = GSPEC\ (\lambda GEN\%PVAR\%515::?'a::type.\ \exists t::?'a::type.\ SETSPEC\ GEN\%PVAR\%515\ (\exists y::?'b::type.\ IN\ y\ f' \wedge t = f\ y)\ t)$

thm Conforming.fully_surrounded_imp_aff_gt_3_1_of_dart_eq_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge IN\ y\ ds \longrightarrow aff\_gt\ (INSERT\ x\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY)))\ (INSERT\ (pr3\ (f1\_fan\ x\ V\ E\ y))\ EMPTY) = aff\_gt\ (INSERT\ x\ (INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY)))\ (INSERT\ (sigma\_fan\ x\ V\ E\ (pr2\ y)\ (pr3\ y))\ EMPTY)$

thm Conforming.OPEN_TOPOLOGICAL_COMPONENT_YFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ f::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \wedge IN\ f\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow HOL\_Light\_Import.open\ f$

thm Conforming.OPEN_TOPOLOGICAL_COMPONENT_YFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ f::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \wedge IN\ f\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow HOL\_Light\_Import.open\ (HOL\_Light\_Import.INTER\ f\ (normball\ x\ (?r::real)))$

thm Conforming.MEASURABLE_TOPOLOGICAL_COMPONENT_YFAN_INTER_BALL:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (r::real)\ f::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \wedge IN\ f\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ f\ (normball\ x\ r))$

thm Conforming.RADIAL_TOPOLOGICAL_COMPONENT_YFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (r::real)\ f::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (0::real) < r \wedge conforming\_fan\ (x,\ V,\ E) \wedge IN\ f\ (topological\_component\_yfan\ (x,\ V,\ E)) \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ f\ (normball\ x\ r))$

thm Conforming.FINITE_TOPOLOGICAL_COMPONENT_YFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \longrightarrow FINITE\ (topological\_component\_yfan\ (x,\ V,\ E))$

thm Conforming.SUM_SOL_TOPOLOGICAL_COMPONENT_YFAN_EQ_SOL_UNIONS:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \longrightarrow sol\ x\ (UNIONS\ (topological\_component\_yfan\ (x,\ V,\ E))) = sum\ (topological\_component\_yfan\ (x,\ V,\ E))\ (sol\ x)$

thm Conforming.UNIONS_TOPOLOGICAL_COMPONENT_EQ_YFAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ UNIONS\ (topological\_component\_yfan\ (x,\ V,\ E)) = yfan\ (x,\ V,\ E)$

thm Conforming.SUM_SOL_IN_FACE_SET_EQ_4PI:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge conforming\_fan\ (x,\ V,\ E) \longrightarrow sum\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E)))\ (\lambda f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ sol\ x\ (dartset\_leads\_into\_fan\ x\ V\ E\ f)) = real\_of\_nat\ (4::nat) * pi$

thm Conforming.FINITE_NODE_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN ds (node_set (hypermap1_of_fanx (x, V, E))) ⟶ FINITE ds

thm Conforming.lemma_properties_of_node_set_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN f (node_set (hypermap1_of_fanx (x, V, E))) ∧ IN y f ∧ IN y1 f ⟶ pr2 y = pr2 y1

thm Conforming.lemma_node_identity_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN f (node_set (hypermap1_of_fanx (x, V, E))) ∧ IN y f ⟶ f = node (hypermap1_of_fanx (x, V, E)) y

thm Conforming.node_subset_dart_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN ds (node_set (hypermap1_of_fanx (x, V, E))) ⟶ SUBSET ds (d_fan (x, V, E))

thm Conforming.rep_node_set_fan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN f (node_set (hypermap1_of_fanx (x, V, E))) ∧ IN y f ⟶ f = GSPEC (λGEN%PVAR%517::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∃ z::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%517 (∃ i≥0::nat. z = (x, pr2 y, power_map_points sigma_fan x V E (pr2 y) (pr3 y) i, power_map_points sigma_fan x V E (pr2 y) (pr3 y) (Suc i))) z)

thm Conforming.properties_of_elements_in_node_fully_surroundedfan:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN ds (node_set (hypermap1_of_fanx (x, V, E))) ∧ IN y ds ⟶ IN (INSERT (pr2 y) (INSERT (pr3 y) EMPTY)) E

thm Conforming.lemma_card_node_eq_set_of_orbits:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN f* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y f* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%518*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *z*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%518* (∃ *i*≥*0*::*nat*. *z* = (*x*, *pr2 y*, *power_map_points sigma_fan x V E* (*pr2 y*) (*pr3 y*) *i*, *power_map_points sigma_fan x V E* (*pr2 y*) (*pr3 y*) (*Suc i*))) *z*)) = *CARD* (*set_of_orbits_points_fan x V E* (*pr2 y*) (*pr3 y*))

thm Conforming.lemma_card_node_eq_set_of_edge:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN f* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *IN y f* ⟶ *CARD* (*set_of_edge* (*pr2 y*) *V E*) = *CARD f*

thm Conforming.mono_cyclic_power_sigma_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*i*::*nat*) *j*::*nat*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN i* (*dotdot* (*0*::*nat*) (*CARD* (*set_of_edge v V E*) − (*1*::*nat*))) ∧ *IN j* (*dotdot* (*0*::*nat*) (*CARD* (*set_of_edge v V E*) − (*1*::*nat*))) ∧ *power_map_points sigma_fan x V E v u i* = *power_map_points sigma_fan x V E v u j* ⟶ *i* = *j*

thm Conforming.SUM_AZIM_FAN_OF_NODE_EQ_SUM_AZIM_I_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN f* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y f* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ *sum* (*dotdot* (*0*::*nat*) (*CARD* (*set_of_edge* (*pr2 y*) *V E*) − (*1*::*nat*))) (*azim_i_fan x V E* (*pr2 y*) (*pr3 y*)) = *sum f* (λ*y1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_fan x V E* (*pr2 y1*) (*pr3 y1*))

thm Conforming.exists_point_in_node:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ (∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y f*)

thm Conforming.SUM_AZIM_FAN_OF_NODE_EQ_2PI_I_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒

bool. *FAN* (*x*, *V*, *E*) ∧ *IN f* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ *sum f* (λ*y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_fan x V E* (*pr2 y*) (*pr3 y*)) = *real_of_nat* (*2*::*nat*) ∗ *pi*

thm Conforming.SUM_CARD_FACE_NODE_DART_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ⟶ *real_of_nat* (*2*::*nat*) ∗ *real_of_nat* (*CARD* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*)))) + (*real_of_nat* (*2*::*nat*) ∗ *real_of_nat* (*CARD* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*)))) − *real_of_nat* (*CARD* (*dart* (*hypermap1_of_fanx* (*x*, *V*, *E*))))) = *real_of_nat* (*4*::*nat*)

thm Conforming.nonconformin_fan_imp_n_fan_ge0:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ⟶ (*0*::*nat*) < *N_FAN* (*x*, *V*, *E*)

thm Conforming.nonconformin_fan_imp_exist_face_gt_3:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ⟶ (∃ *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds*)

thm Conforming.exists_face_in_face_set:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ (∃ *f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN f1 ds*)

thm Conforming.exists_node_in_face_set:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN ds* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ (∃ *f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN f1 ds*)

thm Conforming.identity_face_in_face_set:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN f1 ds* ⟶ *ds* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E*)) *f1*

thm Conforming.identity_node_in_face_set:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*ds*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) *f1*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *IN ds* (*node_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) $\wedge$ *IN f1 ds* $\longrightarrow$ *ds* = *node* (*hypermap1_of_fanx* (*x*, *V*, *E*)) *f1*

thm Conforming.condition_f1_eq_fan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\longrightarrow$ *f1_fan x V E* (*x*, *v*, *u*, *sigma_fan x V E v u*) = (*x*, *u*, *w*, *v*)

thm Conforming.nonconformin_fan_imp_exist_3point_in_face:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *ds*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*. *FAN* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ $\neg$ *conforming_fan* (*x*, *V*, *E*) $\wedge$ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) $\wedge$ (*3*::*nat*) < *CARD ds* $\longrightarrow$ ($\exists$ (*f1*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) *f3*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* $\wedge$ *f1_fan x V E f1* = *f2* $\wedge$ *f1_fan x V E f2* = *f3* $\wedge$ *f1_fan x V E f3* $\neq$ *f1* $\wedge$ *IN* (*INSERT* (*pr2 f2*) (*INSERT* (*pr2 f3*) *EMPTY*)) *E* $\wedge$ $\neg$ *IN* (*INSERT* (*pr2 f3*) (*INSERT* (*pr2 f1*) *EMPTY*)) *E* $\wedge$ *IN* (*INSERT* (*pr2 f1*) (*INSERT* (*pr2 f2*) *EMPTY*)) *E* $\wedge$ *sigma_fan x V E* (*pr2 f2*) (*pr2 f3*) = *pr2 f1* $\wedge$ *pr2 f3* = *pr3 f2* $\wedge$ *pr2 f2* = *pr3 f1*)

thm Conforming.condition_aff_gt_subset_yfan:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ $\neg$ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* $\longrightarrow$ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*yfan* (*x*, *V*, *E*))

thm Conforming.segment_subset_aff_gt_union:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\wedge$ *IN y* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) $\wedge$ *IN z* (*aff_gt* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))) $\longrightarrow$ *SUBSET* (*closed_segment* [(*y*, *z*)]) (*HOL_Light_Import.UNION* (*aff_gt* (*INSERT x EMPTY*) (*INSERT w* (*INSERT v EMPTY*))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))))

thm Conforming.SEGMENT_CONNECTED:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ connected\ (closed\_segment\ [(a,\ b)])$

thm Conforming.AFF_GT_SUBSET_DART_LEADS_INTO_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \longrightarrow SUBSET\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ w\ (INSERT\ v\ EMPTY)))\ (dart\_leads\_into\ x\ V\ E\ u\ w)$

thm Conforming.STEP2_REDUCE_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \longrightarrow SUBSET\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (dartset\_leads\_into\_fan\ x\ V\ E\ ds)$

thm Conforming.STEP3_REDUCE_FAN:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow FAN\ (x,\ V,\ E1)$

thm Conforming.SET_OF_EDGE_UNION_GRAPH:

$\forall (v::?'a::type) (V::?'a::type \Rightarrow bool) (E1::(?'a::type \Rightarrow bool) \Rightarrow bool) E2::(?'a::type \Rightarrow bool) \Rightarrow bool. set\_of\_edge\ v\ V\ (HOL\_Light\_Import.UNION\ E1\ E2) = HOL\_Light\_Import.UNION\ (set\_of\_edge\ v\ V\ E1)\ (set\_of\_edge\ v\ V\ E2)$

thm Conforming.add_edge_imp_card_set_edge_ge1_fan:

$\forall (x::(real, 3)\ cart) (V::(real, 3)\ cart \Rightarrow bool) (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool) E1::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool. FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart. IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge E1 = HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ (?v::(real, 3)\ cart)\ (INSERT\ (?w::(real, 3)\ cart)\ EMPTY))\ EMPTY) \longrightarrow (\forall v::(real, 3)\ cart. IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E1))$

thm Conforming.pr23:

$pr23 = GABS\ (\lambda f::?'d::type \times ?'c::type \times ?'b::type \times ?'a::type \Rightarrow ?'c::type \times ?'b::type. \forall (x::?'d::type) (y::?'c::type) (z::?'b::type) t::?'a::type. GEQ\ (f\ (x, y, z, t))\ (y, z))$

thm Conforming.PR23_OF_D1_FAN:

$\forall (x::(real, 3)\ cart) (V::(real, 3)\ cart \Rightarrow bool) E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool. IMAGE\ pr23\ (d1\_fan\ (x, V, E)) = GSPEC\ (\lambda GEN\%PVAR\%523::(real, 3)\ cart \times (real, 3)\ cart. \exists (v::(real, 3)\ cart)\ w::(real, 3)\ cart. SETSPEC\ GEN\%PVAR\%523\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)\ (v, w))$

thm Conforming.PR23_OF_D20_FAN:

$\forall (x::(real, 3)\ cart) (V::(real, 3)\ cart \Rightarrow bool) E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool. IMAGE\ pr23\ (d20\_fan\ (x, V, E)) = GSPEC\ (\lambda GEN\%PVAR\%524::(real, 3)\ cart \times (real, 3)\ cart. \exists v::(real, 3)\ cart. SETSPEC\ GEN\%PVAR\%524\ (IN\ v\ V \wedge set\_of\_edge\ v\ V\ E = EMPTY)\ (v, v))$

thm Conforming.expand_set_edge_fan:

$\forall (v::?'a::type)\ w::?'a::type. GSPEC\ (\lambda GEN\%PVAR\%528::?'a::type \times ?'a::type. \exists (v'::?'a::type)\ w'::?'a::type. SETSPEC\ GEN\%PVAR\%528\ (INSERT\ v'\ (INSERT\ w'\ EMPTY) = INSERT\ v\ (INSERT\ w\ EMPTY))\ (v', w')) = INSERT\ (v, w)\ (INSERT\ (w, v)\ EMPTY)$

thm Conforming.DART_FANADD_EQ_DART_FAN_ADD_2DART:

$\forall (x::(real, 3)\ cart) (V::(real, 3)\ cart \Rightarrow bool) (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool) (E1::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool) (ds::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool) (f1::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart) (f2::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart) (f3::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart) (v::(real, 3)\ cart) (u::(real, 3)\ cart)\ w::(real, 3)\ cart. FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart. IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge \neg\ conforming\_fan\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2$

(INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ v ≠ w ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ IMAGE pr23 (dart (hypermap1_of_fanx (x, V, E1))) = HOL_Light_Import.UNION (IMAGE pr23 (dart (hypermap1_of_fanx (x, V, E)))) (INSERT (v, w) (INSERT (w, v) EMPTY))

thm Conforming.pr23_inj_in_dfan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ⟶ (∀(y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y (d_fan (x, V, E)) ∧ IN y1 (d_fan (x, V, E)) ∧ pr23 y = pr23 y1 ⟶ y = y1)

thm Conforming.condition_set_of_edge_eq_empty:

∀(v::?'b::type) (V::?'b::type ⇒ bool) (E1::?'a::type) E2::(?'b::type ⇒ bool) ⇒ bool. ¬ IN v (UNIONS E2) ⟶ set_of_edge v V E2 = EMPTY

thm Conforming.SET_OF_EDGE_INVARIANT:

∀(v::?'a::type) (V::?'a::type ⇒ bool) (E1::(?'a::type ⇒ bool) ⇒ bool) E2::(?'a::type ⇒ bool) ⇒ bool. ¬ IN v (UNIONS E2) ⟶ set_of_edge v V (HOL_Light_Import.UNION E1 E2) = set_of_edge v V E1

thm Conforming.expand_unions:

∀(v::?'a::type) w::?'a::type. UNIONS (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = INSERT v (INSERT w EMPTY)

thm Conforming.SIGMA_FAN_OF_FANADD1:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. FAN (x, V, E) ∧ FAN (x, V, E1) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ ¬ IN (INSERT v (INSERT w EMPTY)) E ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ (∀(v1::(real, 3) cart) w1::(real, 3) cart. IN (INSERT v1 (INSERT w1 EMPTY)) E ∧ ¬ IN v1 (INSERT v (INSERT w EMPTY)) ⟶ sigma_fan x V E1 v1 w1 = sigma_fan x V E v1 w1)

thm Conforming.add_edge_graph:

∀(v::?'a::type) (w::?'a::type) (E::(?'a::type ⇒ bool) ⇒ bool) E1::(?'a::type ⇒ bool) ⇒ bool. HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ IN (INSERT w (INSERT v EMPTY)) E1 ∧ IN (INSERT v (INSERT w EMPTY)) E1

thm Conforming.not_in_set_of_edge:

$\forall (v::?'a::type)\ (w::?'a::type)\ (V::?'a::type \Rightarrow bool)\ E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \longrightarrow \neg\ IN\ w\ (set\_of\_edge\ v\ V\ E)$

thm Conforming.set_of_only_edge:

$\forall (v::?'a::type)\ (w::?'a::type)\ V::?'a::type \Rightarrow bool.\ IN\ w\ V \longrightarrow set\_of\_edge\ v\ V\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = INSERT\ w\ EMPTY$

thm Conforming.set_of_only_edge1:

$\forall (v::?'a::type)\ (w::?'a::type)\ V::?'a::type \Rightarrow bool.\ IN\ v\ V \longrightarrow set\_of\_edge\ w\ V\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = INSERT\ v\ EMPTY$

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT1:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge FAN\ (x,\ V,\ E1) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge fan80\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow sigma\_fan\ x\ V\ E1\ v\ w = sigma\_fan\ x\ V\ E\ v\ u$

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT2:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge FAN\ (x,\ V,\ E1) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge fan80\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow sigma\_fan\ x\ V\ E1\ v\ u = w$

thm Conforming.XFAN_INTER_SET:

$\forall (x::(real,\ ?'b::type)\ cart)\ (V::?'a::type)\ (E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool)\ s::(real,\ ?'b::type)\ cart \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (xfan\ (x,\ V,\ E))\ s = UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%530::(real,\ ?'b::type)\ cart \Rightarrow bool.\ \exists y::(real,\ ?'b::type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%530\ (\exists e::(real,\ ?'b::type)\ cart \Rightarrow bool.\ IN\ e\ E \wedge y = HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ x\ EMPTY)\ e)\ s)\ y))$

thm Conforming.condition_azim_imp_edge_fan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ w1::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (INSERT\ w\ (INSERT\ w1\ EMPTY))\ E \wedge$

sigma_fan x V E u w = v ∧ *sigma_fan x V E w w1 = u* ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *azim x w v u = azim x w w1 u* ⟶ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E*

thm Conforming.condition_azim_le_pi:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w = v* ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ (*0*::*real*) < *azim x w v u* ∧ *azim x w v u* < *pi*

thm Conforming.azim_trangle_le_azim_face_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ *IN* (*INSERT w* (*INSERT w1 EMPTY*)) *E* ∧ ¬ *IN* (*INSERT v* (*INSERT w EMPTY*)) *E* ∧ *sigma_fan x V E u w = v* ∧ *sigma_fan x V E w w1 = u* ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ⟶ *azim x w v u* < *azim x w w1 u*

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT3:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *FAN* (*x*, *V*, *E1*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w = v* ∧ *fan80* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *sigma_fan x V E1 w v = u*

thm Conforming.elements_in_ds2_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT* (*?f1.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*INSERT* (*?f2.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*INSERT* (*?f3.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *EMPTY*))) *ds* ∧ *f1_fan x V E ?f1.0 = ?f2.0* ∧ *f1_fan x V E ?f2.0 = ?f3.0* ∧ *f1_fan x V E ?f3.0* ≠ *?f1.0* ∧ *pr2 ?f1.0* = (*?v*::(*real*, *3*) *cart*) ∧ *pr2 ?f2.0* = (*?u*::(*real*, *3*) *cart*) ∧ *pr2 ?f3.0* = (*?w*::(*real*, *3*) *cart*) ∧ *IN* (*INSERT ?v* (*INSERT ?u EMPTY*)) *E* ∧ *IN* (*INSERT ?u* (*INSERT ?w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT ?w* (*INSERT ?v EMPTY*)) *E* ∧ *sigma_fan x*

V E ?u ?w = ?v ∧ (*?ds1.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap1_of_fanx* (*x*, *V*, *?E1.0*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) (*x*, *?v*, *?w*, *sigma_fan x V ?E1.0 ?v ?w*) ∧ (*?ds2.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap1_of_fanx* (*x*, *V*, *?E1.0*)) (*x*, *?w*, *?v*, *sigma_fan x V ?E1.0 ?w ?v*) ∧ (*?f10.0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*x*, *?w*, *?v*, *?u*) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*) = *?E1.0* ⟶ *IN ?f10.0 ?ds2.0*

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT4:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *FAN* (*x*, *V*, *?E1.0*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *u* ≠ *w1* ∧ *IN* (*INSERT v* (*INSERT w1 EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *?E1.0* ⟶ *sigma_fan x V ?E1.0 v w1* = *sigma_fan x V E v w1*

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT5:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *FAN* (*x*, *V*, *?E1.0*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *w1* ≠ *inverse1_sigma_fan x V E w u* ∧ *IN* (*INSERT w* (*INSERT w1 EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *?E1.0* ⟶ *sigma_fan x V ?E1.0 w w1* = *sigma_fan x V E w w1*

thm Conforming.SIGMA_FAN_OF_FANADD_AT_POINT6:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *FAN* (*x*, *V*, *?E1.0*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *w1* = *inverse1_sigma_fan x V E w u* ∧ *IN* (*INSERT w* (*INSERT w1 EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *?E1.0* ⟶ *sigma_fan x V ?E1.0 w w1* = *v*

thm Conforming.f1_fan_of_f10_eq_f20:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart*

× (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ *f10* = (*x*, *w*, *v*, *u*) ∧ *f20* = (*x*, *v*, *u*, *w*) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *f20* = *f1_fan x V E1 f10*

thm Conforming.f1_fan_of_f20_eq_f30:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ *f20* = (*x*, *v*, *u*, *w*) ∧ *f30* = (*x*, *u*, *w*, *v*) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *f30* = *f1_fan x V E1 f20*

thm Conforming.f1_fan_of_f30_eq_f10:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ *f10* = (*x*, *w*, *v*, *u*) ∧ *f30* = (*x*, *u*, *w*, *v*) ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *f10* = *f1_fan x V E1 f30*

thm Conforming.f10_in_d1_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ *f10* = (*x*, *w*, *v*, *u*) ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*))

$EMPTY) = E1 \longrightarrow IN\, f10\ (d1\_fan\ (x,\ V,\ E1))$

thm Conforming.pair_disjoint_f10_f20_f30:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge ds1 = face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) \wedge ds2 = face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) \wedge f10 = (x,\ w,\ v,\ u) \wedge f20 = (x,\ v,\ u,\ w) \wedge f30 = (x,\ u,\ w,\ v) \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow f10 \neq f20 \wedge f20 \neq f30 \wedge f30 \neq f10$

thm Conforming.n_fan_permutes_prime:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ p::((real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge p = (\lambda t::(real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ x\ V\ E)\ (d1\_fan\ (x,\ V,\ E))) \longrightarrow permutes\ (p\ n\_fan)\ (d\_fan\ (x,\ V,\ E))$

thm Conforming.f1_fan_permutes_prime:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ p::((real,\ 3)\ cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge p = (\lambda t::(real,\ 3)$

$cart \Rightarrow ((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ res\ (t\ x\ V\ E)\ (d1\_fan\ (x,\ V,\ E))) \longrightarrow permutes\ (p\ f1\_fan)\ (d\_fan\ (x,\ V,\ E))$

thm Conforming.card_ds2_fanadd_eq3:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge ds1 = face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) \wedge ds2 = face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) \wedge (x,\ w,\ v,\ u) = f10 \wedge (x,\ v,\ u,\ w) = f20 \wedge (x,\ u,\ w,\ v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow CARD\ ds2 = (3::nat)$

thm Conforming.reperentation_of_ds2:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan$

x V E f3 ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *ds2* = *INSERT f10* (*INSERT f20* (*INSERT f30 EMPTY*))

thm Conforming.edge_not_in_ds2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ *f10* = (*x*, *w*, *v*, *u*) ∧ *f20* = (*x*, *v*, *u*, *w*) ∧ *f30* = (*x*, *u*, *w*, *v*) ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ ¬ *IN* (*x*, *v*, *w*, *sigma_fan x V E1 v w*) *ds2*

thm Conforming.disjoint_ds1_and_ds2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*.

IN v V ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *ds1* ≠ *ds2*

thm Conforming.card_eq_image_in_d_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *SUBSET ds* (*d_fan* (*x*, *V*, *E*)) ⟶ *CARD* (*IMAGE pr23 ds*) = *CARD ds*

thm DEF_trans:

HOL_Light_Import.trans = (λ(*_3023509*::(*real*, *3*) *cart*) (*_3023510*::(*real*, *3*) *cart* ⇒ *bool*) (*_3023511*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *_3023512*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *y*, *z*, *sigma_fan x V E y z*)) (*x*, *y*, *z*, *sigma_fan x V _3023512 y z*)))

thm Conforming.trans:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *HOL_Light_Import.trans x V E E1* = *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *GEQ* (*f* (*x*, *y*, *z*, *sigma_fan x V E y z*)) (*x*, *y*, *z*, *sigma_fan x V E1 y z*))

thm DEF_tran:

tran = (λ(*_3023541*::*?'b*::*type*) (*_3023542*::(*real*, *3*) *cart* ⇒ *bool*) *_3023543*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *GABS* (λ*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × *?'a*::*type* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. ∀ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *w*::*?'a*::*type*. *GEQ* (*f* (*x*, *y*, *z*, *w*)) (*x*, *y*, *z*, *sigma_fan x _3023542 _3023543 y z*)))

thm Conforming.tran:

∀(x::?'b::type) (V::(real, 3) cart ⇒ bool) E1::((real, 3) cart ⇒ bool) ⇒ bool. tran x V E1 = GABS (λf::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type ⇒ (real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. ∀(x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) w::?'a::type. GEQ (f (x, y, z, w)) (x, y, z, sigma_fan x V E1 y z))

thm DEF_tranf:

tranf = (λ(_3023562::(real, 3) cart) (_3023563::(real, 3) cart ⇒ bool) (_3023564::?'b::type) (_3023565::((real, 3) cart ⇒ bool) ⇒ bool) _3023566::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type ⇒ bool. SOME f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. ∃y::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type. f = face (hypermap1_of_fanx (_3023562, _3023563, _3023565)) (tran _3023562 _3023563 _3023565 y) ∧ IN y _3023566)

thm Conforming.tranf:

∀(E::?'b::type) (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type ⇒ bool. tranf x V E E1 ds = (SOME f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. ∃y::(real, 3) cart × (real, 3) cart × (real, 3) cart × ?'a::type. f = face (hypermap1_of_fanx (x, V, E1)) (tran x V E1 y) ∧ IN y ds)

thm Conforming.exists_tranf_fan:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ds0::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY))

EMPTY) = *E1* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ⟶ (∃ (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *f* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*tran x V E1 y*) ∧ *IN y ds0*)

thm Conforming.TRANF:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *ds0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ⟶ (∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *tranf x V E E1 ds0* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*tran x V E1 y*) ∧ *IN y ds0*)

thm Conforming.exists_edge_not_edge_in_face:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ×

(real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN y (d_fan (x, V, E)) ⟶ (∃ y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. IN y1 (d_fan (x, V, E)) ∧ ¬ IN (pr2 y1) (INSERT v (INSERT w EMPTY)) ∧ face (hypermap1_of_fanx (x, V, E)) y1 = face (hypermap1_of_fanx (x, V, E)) y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN1:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ ¬ IN (pr3 y) (INSERT v (INSERT w EMPTY)) ∧ IN y (d_fan (x, V, E)) ⟶ tran x V E1 (f1_fan x V E y) = f1_fan x V E1 (tran x V E1 y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN2:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒

bool) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *pr2 y* ≠ *u* ∧ *pr3 y* = *w* ∧ *IN y* (*d_fan* (*x*, *V*, *E*)) ⟶ *tran x V E1* (*f1_fan x V E y*) = *f1_fan x V E1* (*tran x V E1 y*)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN3:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ pr2 y ≠ sigma_fan x V E v u ∧ pr3 y = v ∧ IN y (d_fan (x, V, E)) ⟶ tran x V E1 (f1_fan x V E y) = f1_fan x V E1 (tran x V E1 y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ ¬ IN y ds ∧ IN y (d_fan (x, V, E)) ⟶ tran x V E1 (f1_fan x V E y) = f1_fan x V E1 (tran x V E1 y)

thm Conforming.f1_fan_power_in_face:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ IN y (d1_fan (x, V, E)) ∧ ¬ IN y ds ⟶ ¬ IN (POWER (f1_fan x V E) n y) ds

thm Conforming.f1_fan_power_in_face_imp_in_face:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v

V E)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y* (*d1_fan* (*x*, *V*, *E*)) ∧ *IN* (*POWER* (*f1_fan x V E*) *n y*) *ds* ⟶ *IN y ds*

thm Conforming.TRAN_COMMUTATIVE_F1_FAN0:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (*pr2 y* ≠ *sigma_fan x V E v u* ∧ *pr3 y* = *v* ∨ *pr2 y* ≠ *u* ∧ *pr3 y* = *w* ∨ ¬ *IN* (*pr3 y*) (*INSERT v* (*INSERT w EMPTY*))) ∧ *IN y* (*d_fan* (*x*, *V*, *E*)) ⟶ *tran x V E1* (*f1_fan x V E y*) = *f1_fan x V E1* (*tran x V E1 y*)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN_POWER:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *n*::*nat*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧

¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ ¬ IN y ds ∧ IN y (d_fan (x, V, E)) ⟶ tran x V E1 (POWER (f1_fan x V E) n y) = POWER (f1_fan x V E1) n (tran x V E1 y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN_POWER1:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN y (d_fan (x, V, E)) ∧ (∀m<n. ¬ IN (pr3 (POWER (f1_fan x V E) m y)) (INSERT v (INSERT w EMPTY))) ⟶ tran x V E1 (POWER (f1_fan x V E) n y) = POWER (f1_fan x V E1) n (tran x V E1 y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN_POWER2:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart

× (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN y (d_fan (x, V, E)) ∧ (∀ m<n. pr2 (POWER (f1_fan x V E) m y) ≠ sigma_fan x V E v u ∧ pr3 (POWER (f1_fan x V E) m y) = v ∨ pr2 (POWER (f1_fan x V E) m y) ≠ u ∧ pr3 (POWER (f1_fan x V E) m y) = w) ⟶ tran x V E1 (POWER (f1_fan x V E) n y) = POWER (f1_fan x V E1) n (tran x V E1 y)

thm Conforming.TRAN_COMMUTATIVE_F1_FAN_POWER3:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) n::nat. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT

w (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN y* (*d_fan* (*x*, *V*, *E*)) ∧ (∀ *m*<*n*. *pr2* (*POWER* (*f1_fan x V E*) *m y*) ≠ *sigma_fan x V E v u* ∧ *pr3* (*POWER* (*f1_fan x V E*) *m y*) = *v* ∨ *pr2* (*POWER* (*f1_fan x V E*) *m y*) ≠ *u* ∧ *pr3* (*POWER* (*f1_fan x V E*) *m y*) = *w* ∨ ¬ *IN* (*pr3* (*POWER* (*f1_fan x V E*) *m y*)) (*INSERT v* (*INSERT w EMPTY*)))) ⟶ *tran x V E1* (*POWER* (*f1_fan x V E*) *n y*) = *POWER* (*f1_fan x V E1*) *n* (*tran x V E1 y*)

thm Conforming.unique_tranf_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*ds0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *f* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*tran x V E1 y*) ∧ *IN y ds0* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ⟶ *tranf x V E E1 ds0* = *f*

thm Conforming.tran_in_dart_newfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *FAN* (*x*, *V*, *E1*) ∧ *SUBSET*

E E1 ∧ IN y (d1_fan (x, V, E)) ⟶ IN (tran x V E1 y) (d1_fan (x, V, E1))

thm Conforming.INJ_TRAN_D1_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ FAN (x, V, E1) ∧ SUBSET E E1 ∧ IN y (d1_fan (x, V, E)) ∧ IN y1 (d1_fan (x, V, E)) ∧ tran x V E1 y = tran x V E1 y1 ⟶ y = y1

thm Conforming.INJ_TRANF_FACE_DELETE_DS:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (ds0::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) ds0′::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN ds0 (DELETE (face_set (hypermap1_of_fanx (x, V, E))) ds) ∧ IN ds0′ (DELETE (face_set (hypermap1_of_fanx (x, V, E))) ds) ∧ tranf x V E E1 ds0 = tranf x V E E1 ds0′ ⟶ ds0 = ds0′

thm Conforming.ds1_in_face_set_fanadd:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *IN ds1* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E1*)))

thm Conforming.ds2_in_face_set_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *IN ds2* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E1*)))

thm Conforming.condition_f1_fan_power_in_face_set:

∀ (n::nat) (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ y = POWER (f1_fan x V E) n y1 ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ d_fan (x, V, E) = d1_fan (x, V, E) ∧ IN y1 ds ⟶ IN y ds

thm Conforming.SUR_TRANF_FACE_DELETE_DS:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN f (DELETE (DELETE (face_set (hypermap1_of_fanx (x, V, E1))) ds1) ds2) ⟶ (∃ ds0::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN ds0 (DELETE (face_set (hypermap1_of_fanx (x, V, E))) ds) ∧ tranf x V E E1 ds0 = f)

thm Conforming.DOMAIN_TRANF_FACE_DELETE_DS:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart ×

(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *ds0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ⟶ *IN* (*tranf x V E E1 ds0*) (*DELETE* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E1*))) *ds1*) *ds2*)

thm Conforming.EQ_CARD_FACE_FAN_AND_FANADD:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *CARD* (*DELETE* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E1*))) *ds1*) *ds2*) = *CARD* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*)

thm Conforming.CARD_DART_FANADD:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge v \neq w \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow CARD\ (dart\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))) = CARD\ (dart\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) + (2::nat)$

thm Conforming.ZSZIUQE_LEMMA:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge pr3\ f1 = u \wedge pr3\ f2 = w \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) = ds1 \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) = ds2 \wedge (x,\ w,\ v,\ u) = f10 \wedge (x,\ v,\ u,\ w) = f20 \wedge (x,\ u,\ w,\ v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \longrightarrow N\_FAN\ (x,\ V,\ E1) < N\_FAN\ (x,\ V,\ E)$

thm Conforming.ZSZIUQE:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times$

(real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ ds1 ≠ ds2 ∧ CARD ds2 = (3::nat) ∧ ds2 = INSERT f10 (INSERT f20 (INSERT f30 EMPTY)) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E)

thm Conforming.FAN80_FANADD:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w

EMPTY)) *EMPTY*) = *E1* ⟶ *fan80* (*x*, *V*, *E1*)

thm Conforming.FANADD_CONFORMING:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *conforming_fan* (*x*, *V*, *E1*)

thm Conforming.dartset_leads_in_fanadd_topological_component_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *ds0*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v*

V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ IN ds0 (DELETE (face_set (hypermap1_of_fanx (x, V, E))) ds) ⟶ IN (dartset_leads_into_fan x V E1 (tranf x V E E1 ds0)) (topological_component_yfan (x, V, E1))

thm Conforming.INVARANT_SIGMA_FAN_ADD:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ ¬ IN y ds ∧ IN y (d_fan (x, V, E)) ⟶ sigma_fan x V E1 (pr2 y) (pr3 y) = sigma_fan x V E (pr2 y) (pr3 y)

thm Conforming.lemma_yfanadd_aff_ge:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart ×

(real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ SUBSET (yfan (x, V, E)) (HOL_Light_Import.UNION (yfan (x, V, E1)) (aff_ge (INSERT x EMPTY) (INSERT v (INSERT w EMPTY))))

thm Conforming.lemma_yfanadd_aff_gt:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT

v (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *SUBSET* (*yfan* (*x*, *V*, *E*)) (*HOL_Light_Import.UNION* (*yfan* (*x*, *V*, *E1*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))))

thm Conforming.lemma_yfanadd_aff_gt1:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *SUBSET* (*yfan* (*x*, *V*, *E1*)) (*yfan* (*x*, *V*, *E*))

thm Conforming.YFANADD_AFF_GT:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT* *u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *yfan* (*x*, *V*, *E*) = *HOL_Light_Import.UNION* (*yfan* (*x*, *V*, *E1*)) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))

thm Conforming.dartset_leads_into_fanadd1:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *SUBSET* (*dartset_leads_into_fan x V E1 ds1*) (*dartset_leads_into_fan x V E ds*)

thm Conforming.dartset_leads_into_fanadd2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ SUBSET (dartset_leads_into_fan x V E1 ds2) (dartset_leads_into_fan x V E ds)

thm Conforming.INTERS_HALF_SPACE_DS_FANADD1:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) U1::(real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ INTERS (GSPEC (λGEN%PVAR%531::(real, 3) cart ⇒ bool. ∃ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%531 (IN y ds) (aff_gt (INSERT x (INSERT (pr2 y) (INSERT (pr3 y) EMPTY))) (INSERT (pr3 (f1_fan x V E y)) EMPTY)))) = U1 ⟶ SUBSET (HOL_Light_Import.INTER

U1 (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) (*INSERT u EMPTY*)))
(*dartset_leads_into_fan x V E1 ds2*)

thm Conforming.inverse1_sigma_fan_FANADD:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *inverse1_sigma_fan x V E1 w v* = *inverse1_sigma_fan x V E w u*

thm Conforming.aff_gt_eq_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT* v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ aff_gt (INSERT x (INSERT w (INSERT (inverse1_sigma_fan x V E1 w v) EMPTY))) (INSERT v EMPTY) = aff_gt (INSERT x (INSERT w (INSERT (inverse1_sigma_fan x V E w u) EMPTY))) (INSERT u EMPTY)

thm Conforming.f2_EQ_F30_FANADD:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ f30 = (x, u, w, v) ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ f30 = f2

thm Conforming.CONDITION_DART_IN_NODE:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ IN f (node_set (hypermap1_of_fanx (x, V, E))) ∧ IN y f ∧ IN y1 (d1_fan (x, V, E)) ∧ pr2 y1 = pr2 y ⟶ IN y1 f

thm Conforming.INTERS_HALF_SPACE_DS_FANADD2:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ds0::?'a::type. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ INTERS (GSPEC (λGEN%PVAR%532::(real, 3) cart ⇒ bool. ∃y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%532 (IN y ds) (aff_gt (INSERT x (INSERT (pr2 y) (INSERT (pr3 y) EMPTY))) (INSERT (pr3 (f1_fan x V E y)) EMPTY)))) = (?U1.0::(real, 3) cart ⇒ bool) ⟶ SUBSET (HOL_Light_Import.INTER ?U1.0 (aff_gt (INSERT x (INSERT v (INSERT w EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY))) (dartset_leads_into_fan x V E1 ds1)

thm Conforming.lemmaINTERS_HALF_SPACE_DS_FANADD1:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) U1::(real, 3) cart ⇒ bool. FAN (x, *V, E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT* (*sigma_fan x V E v u*) *EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT* (*sigma_fan x V E v u*) *EMPTY*))) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT* (*sigma_fan x V E v u*) (*INSERT w EMPTY*))) (*INSERT v EMPTY*)))) = *U1* ⟶ *SUBSET* (*HOL_Light_Import.INTER U1* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))) (*INSERT u EMPTY*))) (*dartset_leads_into_fan x V E1 ds2*)

thm Conforming.lemmaINTERS_HALF_SPACE_DS_FANADD2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U1*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT u (INSERT w EMPTY))) (INSERT v EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT (sigma_fan x V E v u) EMPTY))) (INSERT w EMPTY)) (aff_gt (INSERT x (INSERT (sigma_fan x V E v u) (INSERT w EMPTY))) (INSERT v EMPTY)))) = U1 ⟶ SUBSET (HOL_Light_Import.INTER U1 (aff_gt (INSERT x (INSERT v (INSERT w EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY))) (dartset_leads_into_fan x V E1 ds1)*

thm Conforming.aff_gt_3_1_INTER_aff_SUBSET_aff_gt_2_1:

∀ (a::(real, 3) cart) (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) w::(real, 3) cart. azim a x y z < pi ∧ (0::real) < azim a x y z ∧ azim a x y w < pi ∧ (0::real) < azim a x y w ∧ DISJOINT (INSERT a (INSERT x EMPTY)) (INSERT w EMPTY) ∧ ¬ collinear (INSERT a (INSERT x (INSERT w EMPTY))) ∧ ¬ coplanar (INSERT a (INSERT x (INSERT y (INSERT z EMPTY)))) ⟶ SUBSET (HOL_Light_Import.INTER (aff_gt (INSERT a (INSERT x (INSERT y EMPTY))) (INSERT z EMPTY)) (aff (INSERT a (INSERT x (INSERT w EMPTY))))) (aff_gt (INSERT a (INSERT x EMPTY)) (INSERT w EMPTY))

thm Conforming.aff_gt_3_1_INTER_aff_SUBSET_aff_gt_2_14:

∀ (a::(real, 3) cart) (x::(real, 3) cart) (y::(real, 3) cart) z::(real, 3) cart. ¬ coplanar (INSERT a (INSERT x (INSERT y (INSERT z EMPTY)))) ⟶ SUBSET (HOL_Light_Import.INTER (aff_gt (INSERT a (INSERT x (INSERT y EMPTY))) (INSERT z EMPTY)) (aff (INSERT a (INSERT x (INSERT z EMPTY))))) (aff_gt (INSERT a (INSERT x EMPTY)) (INSERT z EMPTY))

thm Conforming.lemmaINTERS_HALF_SPACE_DS_FANADD3:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) U1::(real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set

(hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ U1 = HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT u (INSERT w EMPTY))) (INSERT v EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT (sigma_fan x V E v u) EMPTY))) (INSERT w EMPTY)) (aff_gt (INSERT x (INSERT (sigma_fan x V E v u) (INSERT w EMPTY))) (INSERT v EMPTY)))) ⟶ SUBSET (HOL_Light_Import.INTER U1 (aff (INSERT x (INSERT v (INSERT w EMPTY))))) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT w EMPTY)))

thm Conforming.aff_3_rep_cross_dot:

∀ (x::(real, 3) cart) (v::(real, 3) cart) u::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ aff (INSERT x (INSERT v (INSERT u EMPTY))) = GSPEC (λGEN%PVAR%533::(real, 3) cart. ∃ y::(real, 3) cart. SETSPEC GEN%PVAR%533 (dot (cross (vector_sub v x) (vector_sub u x)) (vector_sub y x) = (0::real)) y)

thm Conforming.SPACE3_EQ_UNION_3SET:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT *v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ HOL_Light_Import.UNION (aff (INSERT x (INSERT v (INSERT w EMPTY)))) (HOL_Light_Import.UNION (aff_gt (INSERT x (INSERT v (INSERT w EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY)) (aff_gt (INSERT x (INSERT v (INSERT w EMPTY))) (INSERT u EMPTY))) = HOL_Light_Import.UNIV*

thm Conforming.lemmaU1_subset_U:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (U::(real, 3) cart ⇒ bool) U1::(real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ U1 = HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT u (INSERT w EMPTY))) (INSERT v EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT (sigma_fan x V E v u) EMPTY)) (HOL_Light_Import.INTER (aff_gt (INSERT x (INSERT v (INSERT (sigma_fan x V E v u) EMPTY))) (INSERT w EMPTY)) (aff_gt (INSERT x (INSERT (sigma_fan x V E v u) (INSERT w EMPTY))) (INSERT v EMPTY)))) ∧ U = HOL_Light_Import.UNION (dartset_leads_into_fan x V E1 ds1) (HOL_Light_Import.UNION (dartset_leads_into_fan

x V E1 ds2) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))))
⟶ *SUBSET U1 U*

thm Conforming.open_subsetU:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *U* = *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT* (*sigma_fan x V E v u*) *EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT* (*sigma_fan x V E v u*) *EMPTY*))) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT* (*sigma_fan x V E v u*) (*INSERT w EMPTY*))) (*INSERT v EMPTY*)))) ⟶ *HOL_Light_Import.open U*

thm Conforming.eq_aff_gt_3_fanadd_edge:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT* (*sigma_fan x V E v u*) *EMPTY*) = *aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)

thm Conforming.aff_gt_add_subset_U1:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U1*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *U1* = *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT* (*sigma_fan x V E v u*) *EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT* (*sigma_fan x V E v u*) *EMPTY*))) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT* (*sigma_fan x V E v u*) (*INSERT w EMPTY*))) (*INSERT v EMPTY*)))) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) *U1*

thm Conforming.lemma_rep_U_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) *U1*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *U* = *HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *U1* = *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT* (*sigma_fan x V E v u*) *EMPTY*)) (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT* (*sigma_fan x V E v u*) *EMPTY*))) (*INSERT w EMPTY*)) (*aff_gt* (*INSERT x* (*INSERT* (*sigma_fan x V E v u*) (*INSERT w EMPTY*))) (*INSERT v EMPTY*)))) ⟶ *U*

= *HOL_Light_Import.UNION U1* (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*dartset_leads_into_fan x V E1 ds2*))

thm Conforming.dartset_leads_into_ds_open_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))) = *U* ⟶ *HOL_Light_Import.open U*

thm Conforming.U_INTER_U2_FANADD:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*,

$V,\ E) \wedge (\forall\, v{::}(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1{::}nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3{::}nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge pr3\ f1 = u \wedge pr3\ f2 = w \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) = ds1 \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) = ds2 \wedge (x,\ w,\ v,\ u) = f10 \wedge (x,\ v,\ u,\ w) = f20 \wedge (x,\ u,\ w,\ v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \wedge (\forall\, E1{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E1) \wedge (\forall\, v{::}(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1{::}nat) < CARD\ (set\_of\_edge\ v\ V\ E1)) \wedge fan80\ (x,\ V,\ E1) \wedge N\_FAN\ (x,\ V,\ E1) < N\_FAN\ (x,\ V,\ E) \longrightarrow conforming\_fan\ (x,\ V,\ E1)) \wedge U = HOL\_Light\_Import.UNION\ (dartset\_leads\_into\_fan\ x\ V\ E1\ ds1)\ (HOL\_Light\_Import.UNION\ (dartset\_leads\_into\_fan\ x\ V\ E1\ ds2)\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))) \longrightarrow HOL\_Light\_Import.INTER\ U\ (UNIONS\ (DELETE\ (DELETE\ (topological\_component\_yfan\ (x,\ V,\ E1))\ (dartset\_leads\_into\_fan\ x\ V\ E1\ ds1))\ (dartset\_leads\_into\_fan\ x\ V\ E1\ ds2))) = EMPTY$

thm Conforming.dartset_leads_into_fan_SUBSET_U:

$\forall (x{::}(real,\ 3)\ cart)\ (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v{::}(real,\ 3)\ cart)\ (u{::}(real,\ 3)\ cart)\ (w{::}(real,\ 3)\ cart)\ (ds1{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f30{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ U{::}(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall\, v{::}(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1{::}nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3{::}nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge pr3\ f1 = u \wedge pr3\ f2 = w \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) = ds1 \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) = ds2 \wedge (x,\ w,\ v,\ u) = f10 \wedge (x,\ v,\ u,\ w) = f20 \wedge (x,\ u,\ w,\ v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \wedge (\forall\, E1{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x,\ V,\ E1) \wedge (\forall\, v{::}(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1{::}nat) < CARD$

(*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))) = *U* ⟶ *SUBSET* (*dartset_leads_into_fan x V E ds*) *U*

thm Conforming.rep_dartset_leads_into_fan_ds:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀*v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))) = *U* ⟶ *dartset_leads_into_fan x V E ds* = *U*

thm Conforming.u_in_topological_component_yfanadd1:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) U::(real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ∧ HOL_Light_Import.UNION (dartset_leads_into_fan x V E1 ds1) (HOL_Light_Import.UNION (dartset_leads_into_fan x V E1 ds2) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT w EMPTY)))) = U ⟶ IN U (topological_component_yfan (x, V, E))

thm Conforming.dartset_leads_into_fan_eq_fanadd:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ds0::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT

(*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *dartset_leads_into_fan x V E1* (*tranf x V E E1 ds0*) = *dartset_leads_into_fan x V E ds0*

thm Conforming.conforming_bijection_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ (∀ *s*::(*real*, *3*) *cart* ⇒ *bool*. *IN s* (*topological_component_yfan* (*x*, *V*, *E*)) ⟶ (∃!*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *IN f* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *s* = *dartset_leads_into_fan x V E f*))

thm Conforming.RADIAL_AFF_GT_1_2:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT u* (*INSERT v EMPTY*)) ∧ (*0*::*real*) < *r* ⟶ *radial_norm r x* (*HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT u* (*INSERT v EMPTY*))) (*normball x r*))

thm Conforming.NORMBALL_SUBSET:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *r'*::*real*. *r* ≤ *r'* ⟶ *SUBSET* (*normball x r*) (*normball x r'*)

thm Conforming.RADIAL_NORM_CO:

$\forall (r::real)\ (r'::real)\ (x::(real,\ 3)\ cart)\ C::(real,\ 3)\ cart \Rightarrow bool.\ r' \le r \wedge (0::real) < r' \longrightarrow radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r)) \longrightarrow radial\_norm\ r'\ x\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ r'))$

thm Conforming.tranf_eq_image_of_tran:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ ds0::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x,\ V,\ E) \wedge \neg\ conforming\_fan\ (x,\ V,\ E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge pr3\ f1 = u \wedge pr3\ f2 = w \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ v,\ w,\ sigma\_fan\ x\ V\ E1\ v\ w) = ds1 \wedge face\ (hypermap1\_of\_fanx\ (x,\ V,\ E1))\ (x,\ w,\ v,\ sigma\_fan\ x\ V\ E1\ w\ v) = ds2 \wedge (x,\ w,\ v,\ u) = f10 \wedge (x,\ v,\ u,\ w) = f20 \wedge (x,\ u,\ w,\ v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \wedge IN\ ds0\ (DELETE\ (face\_set\ (hypermap1\_of\_fanx\ (x,\ V,\ E)))\ ds) \longrightarrow tranf\ x\ V\ E\ E1\ ds0 = IMAGE\ (tran\ x\ V\ E1)\ ds0$

thm Conforming.azim_fanadd_eq:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f3::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (ds1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (ds2::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (f10::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f20::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (f30::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ (ds0::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart$

× (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN ds0* (*DELETE* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) *ds*) ∧ *IN y ds0* ⟶ *azim_fan x V E1* (*pr2* (*tran x V E1 y*)) (*pr3* (*tran x V E1 y*)) = *azim_fan x V E* (*pr2 y*) (*pr3 y*)

thm Conforming.eventally_measurable_fanadd:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *IN f* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ *LET* (λ*U*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* ((∀ *r*::*real*. *measurable* (*HOL_Light_Import*.*INTER* (*ball* (*x*, *r*)) *U*)) ∧ *eventually_radial x U*))

(*dartset_leads_into_fan x V E f*)

thm Conforming.SOL_AFF_GT_2_1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) ⟶ *sol x* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) = (*0*::*real*)

thm Conforming.inverse1_sigma_fan_FANADD1:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *inverse1_sigma_fan x V E1 v* (*sigma_fan x V E v u*) = *w*

thm Conforming.inverse1_sigma_fan_FANADD2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧

¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ inverse1_sigma_fan x V E1 v w = u

thm Conforming.inverse1_sigma_fan_FANADD3:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ⟶ inverse1_sigma_fan x V E1 u v = w

thm Conforming.DS1_DS2_EQ_DS_FANADD1:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*ed1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *ed2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *ed1* = (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ed2* = (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ⟶ *SUBSET* (*DELETE* (*DELETE* (*HOL_Light_Import*.*UNION ds1 ds2*) *ed1*) *ed2*) (*IMAGE* (*tran x V E1*) *ds*)

thm Conforming.DS1_DS2_EQ_DS_FANADD2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*ed1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *ed2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *ds1* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ds2* = *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import*.*UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *ed1* = (*x*, *v*, *w*, *sigma_fan x V E1 v w*) ∧ *ed2* = (*x*, *w*, *v*, sigma_fan x V E1 w v) ⟶ SUBSET (IMAGE (tran x V E1) ds) (DELETE (DELETE (HOL_Light_Import.UNION ds1 ds2) ed1) ed2)

thm Conforming.DS1_DS2_EQ_DS_FANADD:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (ed1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) ed2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ ds1 = face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) ∧ ds2 = face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ ed1 = (x, v, w, sigma_fan x V E1 v w) ∧ ed2 = (x, w, v, sigma_fan x V E1 w v) ⟶ IMAGE (tran x V E1) ds = DELETE (DELETE (HOL_Light_Import.UNION ds1 ds2) ed1) ed2

thm Conforming.azim_fanadd_eq_ds:

∀(x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬

conforming_fan (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ *IN y ds* ∧ *y* ≠ *f1* ∧ *y* ≠ *f3* ⟶ *azim_fan x V E1* (*pr2* (*tran x V E1 y*)) (*pr3* (*tran x V E1 y*)) = *azim_fan x V E* (*pr2 y*) (*pr3 y*)

thm Conforming.TXFBALB:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *conforming_solid_angle_fan* (*x*, *V*, *E*)

thm Conforming.TRAN_IN_TRANF:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*)

$cart \times (real, 3)\ cart)\ (f3::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart)\ (v::(real, 3)\ cart)\ (u::(real, 3)\ cart)\ (w::(real, 3)\ cart)\ (ds1::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ (ds2::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ (f10::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart)\ (f20::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart)\ (f30::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart)\ (f::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ y::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge \neg\ conforming\_fan\ (x, V, E) \wedge IN\ ds\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge (3::nat) < CARD\ ds \wedge SUBSET\ (INSERT\ f1\ (INSERT\ f2\ (INSERT\ f3\ EMPTY)))\ ds \wedge f1\_fan\ x\ V\ E\ f1 = f2 \wedge f1\_fan\ x\ V\ E\ f2 = f3 \wedge f1\_fan\ x\ V\ E\ f3 \neq f1 \wedge pr2\ f1 = v \wedge pr2\ f2 = u \wedge pr2\ f3 = w \wedge pr3\ f1 = u \wedge pr3\ f2 = w \wedge IN\ (INSERT\ v\ (INSERT\ u\ EMPTY))\ E \wedge IN\ (INSERT\ u\ (INSERT\ w\ EMPTY))\ E \wedge \neg\ IN\ (INSERT\ w\ (INSERT\ v\ EMPTY))\ E \wedge sigma\_fan\ x\ V\ E\ u\ w = v \wedge ds1 = face\ (hypermap1\_of\_fanx\ (x, V, E1))\ (x, v, w, sigma\_fan\ x\ V\ E1\ v\ w) \wedge ds2 = face\ (hypermap1\_of\_fanx\ (x, V, E1))\ (x, w, v, sigma\_fan\ x\ V\ E1\ w\ v) \wedge (x, w, v, u) = f10 \wedge (x, v, u, w) = f20 \wedge (x, u, w, v) = f30 \wedge HOL\_Light\_Import.UNION\ E\ (INSERT\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ EMPTY) = E1 \wedge IN\ f\ (DELETE\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E)))\ ds) \wedge IN\ y\ f \longrightarrow IN\ (tran\ x\ V\ E1\ y)\ (tranf\ x\ V\ E\ E1\ f)$

thm Conforming.TXFBALB_VERSION:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge \neg\ conforming\_fan\ (x, V, E) \wedge (\forall E1::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E1) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E1)) \wedge fan80\ (x, V, E1) \wedge N\_FAN\ (x, V, E1) < N\_FAN\ (x, V, E) \longrightarrow conforming\_fan\ (x, V, E1)) \longrightarrow conforming\_solid\_angle\_fan\ (x, V, E)$

thm Conforming.OBHTHCD:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge fan80\ (x, V, E) \wedge \neg\ conforming\_fan\ (x, V, E) \wedge (\forall E1::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FAN\ (x, V, E1) \wedge (\forall v::(real, 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E1)) \wedge fan80\ (x, V, E1) \wedge N\_FAN\ (x, V, E1) < N\_FAN\ (x, V, E) \longrightarrow conforming\_fan\ (x, V, E1)) \longrightarrow (\forall s::(real, 3)\ cart \Rightarrow bool.\ IN\ s\ (topological\_component\_yfan\ (x, V, E)) \longrightarrow (\exists! f::(real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool.\ IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (x, V, E))) \wedge s = dartset\_leads\_into\_fan\ x\ V\ E\ f))$

thm Conforming.inverse1_sigma_fan_FANADD4:

$\forall (x::(real, 3)\ cart)\ (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (E1::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (ds::(real, 3)\ cart \times (real, 3)\ cart$

× (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ⟶ *inverse1_sigma_fan x V E1* (*inverse1_sigma_fan x V E w u*) *w* = *inverse1_sigma_fan x V E* (*inverse1_sigma_fan x V E w u*) *w*

thm Conforming.conforming_diagonal_fanadd1:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *z*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *IN z ds* ∧ *f3* ≠ *z* ⟶ ¬ *collinear* (*INSERT x* (*INSERT* (*pr2 f3*) (*INSERT* (*pr2 z*) *EMPTY*))) ∧ (*f3* = *f1_fan x V E z* ∨ *z* = *f1_fan x V E f3* ∨ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT* (*pr2 f3*) (*INSERT* (*pr2 z*) *EMPTY*))) (*dartset_leads_into_fan x V E ds*))

thm Conforming.INDUCTION_FANADD:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *IN y ds* ⟶ (∃ (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *IN* (*INSERT* (*pr2 f2*) (*INSERT* (*pr2 f3*) *EMPTY*)) *E* ∧ ¬ *IN* (*INSERT* (*pr2 f3*) (*INSERT* (*pr2 f1*) *EMPTY*)) *E* ∧ *IN* (*INSERT* (*pr2 f1*) (*INSERT* (*pr2 f2*) *EMPTY*)) *E* ∧ *sigma_fan x V E* (*pr2 f2*) (*pr2 f3*) = *pr2 f1* ∧ *pr2 f3* = *pr3 f2* ∧ *pr2 f2* = *pr3 f1* ∧ *y* = *f3*)

thm Conforming.conforming_diagonal_fan_ds_fanadd:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*E1*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*ds*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) (*f1*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) (*ds2*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) (*f10*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) *z*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ $\neg$ *conforming_fan* (*x*, *V*, *E*) $\wedge$ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) $\wedge$ (*3*::*nat*) < *CARD ds* $\wedge$ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* $\wedge$ *f1_fan x V E f1* = *f2* $\wedge$ *f1_fan x V E f2* = *f3* $\wedge$ *f1_fan x V E f3* $\neq$ *f1* $\wedge$ *pr2 f1* = *v* $\wedge$ *pr2 f2* = *u* $\wedge$ *pr2 f3* = *w* $\wedge$ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* $\wedge$ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* $\wedge$ $\neg$ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* $\wedge$ *sigma_fan x V E u w* = *v* $\wedge$ *pr3 f1* = *u* $\wedge$ *pr3 f2* = *w* $\wedge$ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* $\wedge$ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* $\wedge$ (*x*, *w*, *v*, *u*) = *f10* $\wedge$ (*x*, *v*, *u*, *w*) = *f20* $\wedge$ (*x*, *u*, *w*, *v*) = *f30* $\wedge$ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* $\wedge$ ($\forall$ *E1*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FAN* (*x*, *V*, *E1*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) $\wedge$ *fan80* (*x*, *V*, *E1*) $\wedge$ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) $\longrightarrow$ *conforming_fan* (*x*, *V*, *E1*)) $\wedge$ *IN y ds* $\wedge$ *IN z ds* $\wedge$ *y* $\neq$ *z* $\longrightarrow$ $\neg$ *collinear* (*INSERT x* (*INSERT* (*pr2 y*) (*INSERT* (*pr2 z*) *EMPTY*))) $\wedge$ (*y* = *f1_fan x V E z* $\vee$ *z* = *f1_fan x V E y* $\vee$ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT* (*pr2 y*) (*INSERT* (*pr2 z*) *EMPTY*))) (*dartset_leads_into_fan x V E ds*))

thm Conforming.GGZWYRM:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FAN* (*x*, *V*, *E*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) $\wedge$ *fan80* (*x*, *V*, *E*) $\wedge$ $\neg$ *conforming_fan* (*x*, *V*, *E*) $\wedge$ ($\forall$ *E1*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*. *FAN* (*x*, *V*, *E1*) $\wedge$ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) $\wedge$ *fan80* (*x*, *V*, *E1*) $\wedge$ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) $\longrightarrow$ *conforming_fan* (*x*, *V*, *E1*)) $\longrightarrow$ *conforming_diagonal_fan* (*x*, *V*, *E*)

thm Conforming.INTERS_HALF_SPACE_DS_FANADD3:

$\forall$ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*E1*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*ds*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) (*f1*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *U1*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *U1* = *INTERS* (*GSPEC* (λ*GEN%PVAR%539*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%539* (*IN y ds*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 y*) (*INSERT* (*pr3 y*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E y*)) *EMPTY*)))) ⟶ *SUBSET* (*HOL_Light_Import.INTER U1* (*aff* (*INSERT x* (*INSERT v* (*INSERT w EMPTY*))))) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))

thm Conforming.lemma_HYUAZSE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*U*::(*real*, *3*) *cart* ⇒ *bool*) *U1*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2* *f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ∧ *INTERS* (*GSPEC* (λ*GEN%PVAR%540*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%540* (*IN y ds*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 y*) (*INSERT* (*pr3 y*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E y*)) *EMPTY*)))) = *U1* ∧ *U* = *HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds1*) (*HOL_Light_Import.UNION* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))) ⟶ *SUBSET U1 U*

thm Conforming.DART_FANADD_SUBSET_HALFSPACE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *SUBSET* (*dartset_leads_into_fan x V E1 ds1*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 f3*) (*INSERT* (*pr3 f3*) *EMPTY*)))

(*INSERT* (*pr3* (*f1_fan x V E f3*)) *EMPTY*))

thm Conforming.DART_FANADD_SUBSET_HALFSPACE1:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *SUBSET* (*dartset_leads_into_fan x V E1 ds2*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 f3*) (*INSERT* (*pr3 f3*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E f3*)) *EMPTY*))

thm Conforming.DART_FANADD_SUBSET_HALFSPACE2:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 f3*) (*INSERT* (*pr3 f3*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E f3*)) *EMPTY*))

thm Conforming.DART_FANADD_SUBSET_HALFSPACE3:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f3*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds2*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*f10*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) (*f20*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f30*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ (*3*::*nat*) < *CARD ds* ∧ *SUBSET* (*INSERT f1* (*INSERT f2* (*INSERT f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *SUBSET* (*dartset_leads_into_fan x V E ds*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 f3*) (*INSERT* (*pr3 f3*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E f3*)) *EMPTY*))

thm Conforming.DART_FANADD_SUBSET_HALFSPACE4:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT f3 EMPTY))) ds ∧ f1_fan x V E f1 = f2 ∧ f1_fan x V E f2 = f3 ∧ f1_fan x V E f3 ≠ f1 ∧ pr2 f1 = v ∧ pr2 f2 = u ∧ pr2 f3 = w ∧ IN (INSERT v (INSERT u EMPTY)) E ∧ IN (INSERT u (INSERT w EMPTY)) E ∧ ¬ IN (INSERT w (INSERT v EMPTY)) E ∧ sigma_fan x V E u w = v ∧ pr3 f1 = u ∧ pr3 f2 = w ∧ face (hypermap1_of_fanx (x, V, E1)) (x, v, w, sigma_fan x V E1 v w) = ds1 ∧ face (hypermap1_of_fanx (x, V, E1)) (x, w, v, sigma_fan x V E1 w v) = ds2 ∧ (x, w, v, u) = f10 ∧ (x, v, u, w) = f20 ∧ (x, u, w, v) = f30 ∧ HOL_Light_Import.UNION E (INSERT (INSERT v (INSERT w EMPTY)) EMPTY) = E1 ∧ (∀ E1::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E1) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E1)) ∧ fan80 (x, V, E1) ∧ N_FAN (x, V, E1) < N_FAN (x, V, E) ⟶ conforming_fan (x, V, E1)) ⟶ SUBSET (dartset_leads_into_fan x V E ds) (INTERS (GSPEC (λGEN%PVAR%544::(real, 3) cart ⇒ bool. ∃ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%544 (IN y ds) (aff_gt (INSERT x (INSERT (pr2 y) (INSERT (pr3 y) EMPTY))) (INSERT (pr3 (f1_fan x V E y)) EMPTY)))))

thm Conforming.DART_FANADD_EQ_HALFSPACE:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (E1::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f3::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (f10::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) (f20::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f30::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ ¬ conforming_fan (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ (3::nat) < CARD ds ∧ SUBSET (INSERT f1 (INSERT f2 (INSERT *f3 EMPTY*))) *ds* ∧ *f1_fan x V E f1* = *f2* ∧ *f1_fan x V E f2* = *f3* ∧ *f1_fan x V E f3* ≠ *f1* ∧ *pr2 f1* = *v* ∧ *pr2 f2* = *u* ∧ *pr2 f3* = *w* ∧ *IN* (*INSERT v* (*INSERT u EMPTY*)) *E* ∧ *IN* (*INSERT u* (*INSERT w EMPTY*)) *E* ∧ ¬ *IN* (*INSERT w* (*INSERT v EMPTY*)) *E* ∧ *sigma_fan x V E u w* = *v* ∧ *pr3 f1* = *u* ∧ *pr3 f2* = *w* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *v*, *w*, *sigma_fan x V E1 v w*) = *ds1* ∧ *face* (*hypermap1_of_fanx* (*x*, *V*, *E1*)) (*x*, *w*, *v*, *sigma_fan x V E1 w v*) = *ds2* ∧ (*x*, *w*, *v*, *u*) = *f10* ∧ (*x*, *v*, *u*, *w*) = *f20* ∧ (*x*, *u*, *w*, *v*) = *f30* ∧ *HOL_Light_Import.UNION E* (*INSERT* (*INSERT v* (*INSERT w EMPTY*)) *EMPTY*) = *E1* ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *dartset_leads_into_fan x V E ds* = *INTERS* (*GSPEC* (λ*GEN%PVAR%546*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%546* (*IN y ds*) (*aff_gt* (*INSERT x* (*INSERT* (*pr2 y*) (*INSERT* (*pr3 y*) *EMPTY*))) (*INSERT* (*pr3* (*f1_fan x V E y*)) *EMPTY*))))

thm Conforming.HYUAZSE:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ ¬ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *E1*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E1*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E1*)) ∧ *fan80* (*x*, *V*, *E1*) ∧ *N_FAN* (*x*, *V*, *E1*) < *N_FAN* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E1*)) ⟶ *conforming_half_space_fan* (*x*, *V*, *E*)

thm Conforming.PIIJBJK:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *conforming_fan* (*x*, *V*, *E*)

thm Conforming.expand_xfan_eq_aff_gt_aff_ge:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *UNIONS* (*GSPEC* (λ*GEN%PVAR%550*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%550* (∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *IN e E* ∧ *y* = *aff_ge* (*INSERT x EMPTY*) *e*) *y*)) = *HOL_Light_Import.UNION* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%551*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%551* (*IN e E*) (*aff_gt* (*INSERT x EMPTY*) *e*)))) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%552*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%552* (*IN v V*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)))))

thm Conforming.properties12_fan7:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (*λGEN%PVAR%553*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%553* (*IN e E*) (*aff_gt* (*INSERT x EMPTY*) *e*)))) (*UNIONS* (*GSPEC* (*λGEN%PVAR%554*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%554* (*IN v V*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)))))) = *EMPTY*

thm Conforming.yfan_union_aff_gt_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ⟶ *HOL_Light_Import.UNION* (*yfan* (*x*, *V*, *E*)) (*UNIONS* (*GSPEC* (*λGEN%PVAR%555*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%555* (*IN e E*) (*aff_gt* (*INSERT x EMPTY*) *e*)))) = *DIFF HOL_Light_Import.UNIV* (*UNIONS* (*GSPEC* (*λGEN%PVAR%556*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%556* (*IN v V*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)))))

thm Conforming.exists_point_in_dartset_leads_into_fan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ (∃ *y*::(*real*, *3*) *cart*. *IN y* (*dartset_leads_into_fan x V E ds*))

thm Conforming.NEGLIGIBLE_AFF_3_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *z*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ⟶ *negligible* (*UNIONS* (*GSPEC* (*λGEN%PVAR%559*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%559* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*)))))))

thm Conforming.MEASURE_AFF_3_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *z*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ⟶ *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (*λGEN%PVAR%560*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%560* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*))))))) = (*0*::*real*)

thm Conforming.NEGLIGIBLE_AFF_3_UNION_INTER_BALL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*z*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ⟶ *negligible* (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (*λGEN%PVAR%562*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%562* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*))))))) (*normball y r*))

thm Conforming.MEASURE_AFF_3_UNION_FAN:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*z*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ⟶

HOL_Light_Import.measure (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%563*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%563* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*))))))) (*normball y r*)) = (*0*::*real*)

thm Conforming.HAS_MEASURE_AFF_3_UNION_INTER_BALL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*z*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ⟶ *has_measure* (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%564*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%564* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*))))))) (*normball y r*)) (*0*::*real*)

thm Conforming.MEASURABLE_AFF_3_UNION_INTER_BALL:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*z*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ⟶ *measurable* (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%565*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%565* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*))))))) (*normball y r*))

thm Conforming.measure_ball_diff_set_negligible:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*z*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ∧ (*0*::*real*) ≤ *r* ⟶ *HOL_Light_Import.measure* (*DIFF* (*normball y r*) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%568*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%568* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*)))))))) = *real_of_nat* (*4*::*nat*) / *real_of_nat* (*3*::*nat*) ∗ (*pi* ∗ $r^{3::nat}$)

thm Conforming.exists_measure_ball_diff_set_negligible:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *r*::*real*. *FAN* (*x*, *V*, *E*) ∧ (*0*::*real*) < *r* ⟶ (∃ *a*::(*real*, *3*) *cart*. *IN a* (*DIFF* (*normball y r*) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%571*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%571* (*IN v V*) (*aff* (*INSERT x* (*INSERT z* (*INSERT v EMPTY*)))))))))

thm Conforming.connected_in_dartset_leads_into_fan_union_aff_gt:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*ds1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *Z*::(*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN ds1* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *Z* = *DIFF HOL_Light_Import.UNIV* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%572*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%572* (*IN v V*) (*aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*))))) ⟶ (∃ (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *IN y* (*dartset_leads_into_fan* x V E ds) ∧ IN z (dartset_leads_into_fan x V E ds1) ∧ SUBSET (closed_segment [(y, z)]) Z)

thm Conforming.AFF_GT_1_1_SUBSET_DARTSET_LEADS_INTO_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) y::(real, 3) cart. FAN (x, V, E) ∧ conforming_fan (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ E ≠ EMPTY ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ IN y (dartset_leads_into_fan x V E ds) ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT y EMPTY)) (dartset_leads_into_fan x V E ds)

thm Conforming.aff_gt_subset_dartset_leads_into_fan_union_aff_gt:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (ds::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (ds1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (Z::(real, 3) cart ⇒ bool) (y::(real, 3) cart) z::(real, 3) cart. FAN (x, V, E) ∧ conforming_fan (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN ds (face_set (hypermap1_of_fanx (x, V, E))) ∧ IN ds1 (face_set (hypermap1_of_fanx (x, V, E))) ∧ Z = DIFF HOL_Light_Import.UNIV (UNIONS (GSPEC (λGEN%PVAR%573::(real, 3) cart ⇒ bool. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%573 (IN v V) (aff_ge (INSERT x EMPTY) (INSERT v EMPTY))))) ∧ IN y (dartset_leads_into_fan x V E ds) ∧ IN z (dartset_leads_into_fan x V E ds1) ∧ x ≠ y ∧ x ≠ z ∧ SUBSET (closed_segment [(y, z)]) Z ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT y (INSERT z EMPTY))) Z

thm Conforming.aff_gt_1_2_subset_aff_1_3111:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ∧ IN y (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u EMPTY))) ∧ IN z (aff_gt (INSERT x EMPTY) (INSERT v (INSERT w EMPTY))) ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT y (INSERT z EMPTY))) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u (INSERT w EMPTY))))

thm Conforming.AFF_GT_1_3_SUBSET_AFF_GT_1_3:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) t::real. ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ∧ (0::real) < t ∧ t < (1::real) ⟶ SUBSET (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u (INSERT (vector_add (% ((1::real) − t) u) (% t w)) EMPTY)))) (aff_gt (INSERT x EMPTY) (INSERT v (INSERT u (INSERT w EMPTY))))

thm Conforming.lemma_connect_hypermap:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (f1::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart) f2::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. FAN (x, V, E) ∧ conforming_fan (x, V, E) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (1::nat) < CARD (set_of_edge v V E)) ∧ fan80 (x, V, E) ∧ IN f1 (d_fan (x, V, E)) ∧ IN f2 (d_fan (x, V, E)) ⟶ (∃ D::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN D (set_of_components (hypermap1_of_fanx (x, V, E))) ∧ IN f1 D ∧ IN f2 D)

thm Conforming.WGVWSKE:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ conforming_fan (x, V, E) ⟶ connected_hypermap (hypermap1_of_fanx (x, V, E))

thm Conforming.CARD_EDGE_SET_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) e::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. FAN (x, V, E) ∧ IN e (edge_set (hypermap1_of_fanx (x, V, E))) ∧ conforming_fan (x, V, E) ⟶ CARD e = (2::nat)

thm Conforming.REP_CARD_EDGE_SET_FAN:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ conforming_fan (x, V, E) ⟶ real_of_nat (CARD (edge_set (hypermap1_of_fanx (x, V, E)))) ∗ real_of_nat (2::nat) = real_of_nat (CARD (dart (hypermap1_of_fanx (x, V, E))))

thm Conforming.GGRLKHP:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (x, V, E) ∧ conforming_fan (x, V, E) ⟶ planar_hypermap (hypermap1_of_fanx (x, V, E))

thm Polyhedron.GRAPH:

∀ E::(?'a::type ⇒ bool) ⇒ bool. graph E = (∀ e::?'a::type ⇒ bool. IN e E ⟶ HAS_SIZE e (2::nat))

thm Polyhedron.CYCLIC_SET:

cyclic_set (?W::(real, ?'a::type) cart ⇒ bool) (?v::(real, ?'a::type) cart) (?w::(real, ?'a::type) cart) = (?v ≠ ?w ∧ FINITE ?W ∧ (∀ (p::(real, ?'a::type) cart) (q::(real, ?'a::type) cart) h::real. IN p ?W ∧ IN q ?W ∧ vector_sub p q = % h (vector_sub ?v ?w) ⟶ p = q) ∧ HOL_Light_Import.INTER ?W (hull affine (INSERT ?v (INSERT ?w EMPTY))) = EMPTY)

thm Polyhedron.POLYHEDRON_FAN:

∀ (p::(real, 3) cart ⇒ bool) z::(real, 3) cart. bounded p ∧ polyhedron p ∧ IN z (interior p) ⟶ FAN (z, vertices p, edges p)

thm DEF_fchanged:

fchanged = (λ_3396354::(real, ?'a::type) cart ⇒ bool. GSPEC (λGEN%PVAR%605::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%605 (∃ (v1::(real,

?'a::type) cart) t::real. v = % t v1 ∧ IN v1 (relative_interior _3396354) ∧ (0::real) < t) v))

thm Polyhedron.fchanged:

∀ f::(real, ?'a::type) cart ⇒ bool. fchanged f = GSPEC (λGEN%PVAR%605::(real, ?'a::type) cart. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%605 (∃ (v1::(real, ?'a::type) cart) t::real. v = % t v1 ∧ IN v1 (relative_interior f) ∧ (0::real) < t) v)

thm Polyhedron.CONVEX_RELATIVE_INTERIOR:

∀ p::(real, 3) cart ⇒ bool. polyhedron p ⟶ convex (relative_interior p)

thm Polyhedron.CONVEX_RELATIVE_INTERIOR_FACE:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. polyhedron p ∧ face_of f p ⟶ convex (relative_interior f)

thm Polyhedron.CONVEX_RELATIVE_INTERIOR_FACET:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. polyhedron p ∧ facet_of f p ⟶ convex (relative_interior f)

thm Polyhedron.CONNECTED_RELATIVE_INTERIOR_FACET:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. polyhedron p ∧ facet_of f p ⟶ connected (relative_interior f)

thm Polyhedron.CONNECTED_HALF_LINE:

∀ (x::(real, 3) cart) v::(real, 3) cart. connected (aff_gt (INSERT x EMPTY) (INSERT v EMPTY))

thm Polyhedron.RELATIVE_SUBSET_FCHANGE:

∀ (p::(real, 3) cart ⇒ bool) (x::(real, 3) cart) f::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN x (interior p) ∧ facet_of f p ⟶ SUBSET (relative_interior f) (fchanged f)

thm Polyhedron.AFF_GT_SUBSET_FCHANGED:

∀ (p::(real, 3) cart ⇒ bool) (x::(real, 3) cart) (f::(real, 3) cart ⇒ bool) y::(real, 3) cart. bounded p ∧ polyhedron p ∧ IN x (interior p) ∧ facet_of f p ∧ IN y (relative_interior f) ⟶ SUBSET (GSPEC (λGEN%PVAR%615::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%615 (∃ t>0::real. v = % t y) v)) (fchanged f)

thm Polyhedron.CONNECTED_HALF_LINE1:

∀ y::(real, 3) cart. connected (GSPEC (λGEN%PVAR%618::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%618 (∃ t>0::real. v = % t y) v))

thm Polyhedron.CONNECTED_COMPONENT_OF_SUBSET:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *SUBSET s t* $\wedge$ *connected_component s x y* $\longrightarrow$ *connected_component t x y*

thm Polyhedron.CONNECTED_COMPONENT_TRANS:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *z*::(*real*, *?'a*::*type*) *cart*. *connected_component s x y* $\wedge$ *connected_component s y z* $\longrightarrow$ *connected_component s x z*

thm Polyhedron.CONNECTED_FCHANGED:

$\forall$ (*p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*x*::(*real*, *3*) *cart*) *f*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN x* (*interior p*) $\wedge$ *facet_of f p* $\longrightarrow$ *connected* (*fchanged f*)

thm Polyhedron.CONTINUOUS_ON_LIFT_DOT:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *a*::(*real*, *?'a*::*type*) *cart*. *continuous_on* (*lift* $\circ$ *dot a*) *s*

thm DEF_delta_func:

delta_func = (λ(*_3396433*::*?'a*::*type*) *_3396434*::*?'a*::*type*. *if _3396433* = *_3396434 then 1*::*real else* (*0*::*real*))

thm DEF_func1:

func1 = (λ(*_3396445*::*real*) (*_3396446*::*?'a*::*type*) (*_3396447*::*?'a*::*type*) *_3396448*::*?'a*::*type*. *if _3396446* = *_3396448 then _3396445 else if _3396447* = *_3396448 then* (*1*::*real*) $-$ *_3396445 else* (*0*::*real*))

thm Polyhedron.AFFINITE_HULL_BALL_EQ_UNIV:

$\forall$ (*x*::(*real*, *3*) *cart*) *e*::*real*. (*0*::*real*) < *e* $\longrightarrow$ *hull affine* (*ball* (*x*, *e*)) = *HOL_Light_Import.UNIV*

thm Polyhedron.INTERIOR_AFFINIE_HUL_EQ_UNIV:

$\forall$ (*x*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN x* (*interior p*) $\longrightarrow$ *hull affine p* = *HOL_Light_Import.UNIV*

thm Polyhedron.AFF_DIM_INTERIOR_EQ_3:

$\forall$ (*x*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN x* (*interior p*) $\longrightarrow$ *aff_dim p* = *int* (*3*::*nat*)

thm Polyhedron.INTERIOR_IMP_RELATIVE_INTERIOR:

$\forall$ (*x*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN x* (*interior p*) $\longrightarrow$ *IN x* (*relative_interior p*)

thm Polyhedron.IN_RELATIVE_INTERIOR1:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *IN x* (*relative_interior s*) $\longrightarrow$ ($\exists$ *e*>*0*::*real*. *SUBSET* (*HOL_Light_Import.INTER* (*ball* (*x*, *e*)) (*hull affine s*)) (*relative_interior s*))

thm Polyhedron.FCHANGED_OPEN:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ f::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow HOL\_Light\_Import.open\ (fchanged\ f)$

thm Polyhedron.FCHANGED_ONE_TO_ONE:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (f1::(real,\ 3)\ cart \Rightarrow bool)\ f2::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f1\ p \wedge facet\_of\ f2\ p \wedge HOL\_Light\_Import.INTER\ (fchanged\ f1)\ (fchanged\ f2) \neq EMPTY \longrightarrow f1 = f2$

thm Polyhedron.CARD_EXISTS_2:

$\forall e::?'a::type \Rightarrow bool.\ FINITE\ e \wedge CARD\ e = (2::nat) \longrightarrow (\exists (v::?'a::type)\ w::?'a::type.\ e = INSERT\ v\ (INSERT\ w\ EMPTY))$

thm Polyhedron.EXISTS_EDGE_POLYTOPE:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow (\exists e::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e\ (edges\ p))$

thm Polyhedron.EXISTS_EDGE_POLYTOPE1:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow edges\ p \neq EMPTY$

thm Polyhedron.REDUCE_POINT_FACET:

$\forall (x::(real,\ 3)\ cart)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ x\ (yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)) \longrightarrow (\exists (f::(real,\ 3)\ cart \Rightarrow bool)\ t::real.\ (0::real) < t \wedge facet\_of\ f\ p \wedge IN\ (\%\ t\ x)\ f)$

thm Polyhedron.aff_ge_1_1_subset_xfan:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ y::(real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge IN\ y\ (xfan\ (x,\ V,\ E)) \wedge x \neq y \longrightarrow SUBSET\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ EMPTY))\ (xfan\ (x,\ V,\ E))$

thm Polyhedron.YFAN_SUBSET_UNIONS_FCHANGED:

$\forall (y::(real,\ 3)\ cart)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ y\ (yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)) \longrightarrow IN\ y\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%639::(real,\ 3)\ cart \Rightarrow bool.\ \exists f::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%639\ (facet\_of\ f\ p)\ (fchanged\ f))))$

thm Polyhedron.in_aff_ge_fan:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ a::real.\ DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) \wedge (0::real) \leq a \wedge a \leq (1::real) \longrightarrow IN\ (vector\_add\ (\%\ ((1::real) - a)\ v)\ (\%\ a\ u))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)))$

thm Polyhedron.REDUCE_POINT_FACET_EXISTS:

$\forall (x::(real,\ 3)\ cart)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge x \neq vec\ (0::nat) \longrightarrow (\exists (f::(real,\ 3)\ cart \Rightarrow bool)\ t::real.\ (0::real) < t \wedge facet\_of\ f\ p \wedge IN\ (\%\ t\ x)\ f)$

thm Polyhedron.FCHANGED_SUBSET_YFAN:

$\forall (x::(real,\ 3)\ cart)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ x\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%642::(real,\ 3)\ cart \Rightarrow bool.\ \exists f::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%642\ (facet\_of\ f\ p)\ (fchanged\ f)))) \longrightarrow IN\ x\ (yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))$

thm Polyhedron.FCHANGED_EQ_YFAN:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%643::(real,\ 3)\ cart \Rightarrow bool.\ \exists f::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%643\ (facet\_of\ f\ p)\ (fchanged\ f))) = yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)$

thm Polyhedron.EXISTS_POINT_IN_FCHANGED:

$\forall (f::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow (\exists y::(real,\ 3)\ cart.\ IN\ y\ (fchanged\ f))$

thm Polyhedron.FCHANGED_IN_COMPONENT:

$\forall (f::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow IN\ (fchanged\ f)\ (topological\_component\_yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))$

thm Polyhedron.SUR_FCHANGED:

$\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ s\ (topological\_component\_yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)) \longrightarrow (\exists f::(real,\ 3)\ cart \Rightarrow bool.\ facet\_of\ f\ p \wedge s = fchanged\ f)$

thm Polyhedron.AMHFNXP:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow (\forall s::(real,\ 3)\ cart \Rightarrow bool.\ IN\ s\ (topological\_component\_yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)) \longrightarrow (\exists! f::(real,\ 3)\ cart \Rightarrow bool.\ facet\_of\ f\ p \wedge s = fchanged\ f))$

thm Polyhedron.AMHFNXP_BIJ:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow BIJ\ fchanged\ (\lambda f::(real,\ 3)\ cart \Rightarrow bool.\ facet\_of\ f\ p)\ (topological\_component\_yfan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))$

thm Polyhedron.EXPAND_EDGE_POLYTOPE:

$\forall (f::(real,\ ?'a::type)\ cart \Rightarrow bool)\ p::(real,\ ?'a::type)\ cart \Rightarrow bool.\ polytope\ p \wedge face\_of\ f\ p \wedge aff\_dim\ f = int\ (1::nat) \longrightarrow (\exists (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ f = closed\_segment\ [(a,\ b)])$

thm Polyhedron.EXISTS_EDGE_AT_VERTICES:

$\forall p$::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\longrightarrow$ ($\forall v$::(*real*, *3*) *cart*. *IN v* (*vertices p*) $\longrightarrow$ *set_of_edge v* (*vertices p*) (*edges p*) $\neq$ *EMPTY*)

thm SUBSET_SEGMENT_conjunct3:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*open_segment* (*a*, *b*)) (*open_segment* (*c*, *d*)) = (*a* = *b* $\vee$ *IN a* (*closed_segment* [(*c*, *d*)]) $\wedge$ *IN b* (*closed_segment* [(*c*, *d*)]))

thm SUBSET_SEGMENT_conjunct2:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*c*::(*real*, *?'a*::*type*) *cart*) *d*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*open_segment* (*a*, *b*)) (*closed_segment* [(*c*, *d*)]) = (*a* = *b* $\vee$ *IN a* (*closed_segment* [(*c*, *d*)]) $\wedge$ *IN b* (*closed_segment* [(*c*, *d*)]))

thm Polyhedron.FLVNSME:

$\forall$ (*v*::(*real*, *3*) *cart*) (*A*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *3*) *cart*) (*b*::*real*) *p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\wedge$ *A* = *GSPEC* (λ*GEN%PVAR%761*::(*real*, *3*) *cart*. $\exists x$::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%761* (*dot a x* < *b*) *x*) $\wedge$ *a* $\neq$ *vec* (*0*::*nat*) $\wedge$ *IN v* (*GSPEC* (λ*GEN%PVAR%762*::(*real*, *3*) *cart*. $\exists x$::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%762* (*dot a x* = *b*) *x*)) $\wedge$ *IN* (*vec* (*0*::*nat*)) (*GSPEC* (λ*GEN%PVAR%763*::(*real*, *3*) *cart*. $\exists x$::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%763* (*dot a x* = *b*) *x*)) $\wedge$ *IN v* (*vertices p*) $\longrightarrow$ ($\exists w$::(*real*, *3*) *cart*. *IN w* (*vertices p*) $\wedge$ *IN w A* $\wedge$ *IN* (*INSERT v* (*INSERT w EMPTY*)) (*edges p*))

thm Polyhedron.CARD_SET_OF_EDGE_INEQ_1_POLYHEDRON:

$\forall p$::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\longrightarrow$ ($\forall v$::(*real*, *3*) *cart*. *IN v* (*vertices p*) $\longrightarrow$ (*1*::*nat*) < *CARD* (*set_of_edge v* (*vertices p*) (*edges p*)))

thm Polyhedron.BSXAQBQ:

$\forall$ (*p*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *x*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\wedge$ *IN x* (*d_fan* (*vec* (*0*::*nat*), *vertices p*, *edges p*)) $\longrightarrow$ *azim_fan* (*vec* (*0*::*nat*)) (*vertices p*) (*edges p*) (*pr2 x*) (*pr3 x*) < *pi*

thm Polyhedron.POLYTOPE_FAN80:

$\forall p$::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\longrightarrow$ *fan80* (*vec* (*0*::*nat*), *vertices p*, *edges p*)

thm Polyhedron.WBLARHH:

$\forall p$::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *bounded p* $\wedge$ *polyhedron p* $\wedge$ *IN* (*vec* (*0*::*nat*)) (*interior p*) $\longrightarrow$ ($\forall f$::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\times$ (*real*, *3*)

$cart \Rightarrow bool.\ IN\, f\ (face\_set\ (hypermap1\_of\_fanx\ (vec\ (0{::}nat),\ vertices\ p,\ edges\ p))) \longrightarrow (\exists!f1{::}(real,\ 3)\ cart \Rightarrow bool.\ facet\_of\ f1\ p \wedge dartset\_leads\_into\_fan\ (vec\ (0{::}nat))\ (vertices\ p)\ (edges\ p)\ f = fchanged\ f1))$

thm Cfyxfty.WBLARHH_BIJ:

$\forall\, p{::}(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0{::}nat))\ (interior\ p) \longrightarrow BIJ\ (dartset\_leads\_into\_fan\ (vec\ (0{::}nat))\ (vertices\ p)\ (edges\ p))\ (face\_set\ (hypermap1\_of\_fanx\ (vec\ (0{::}nat),\ vertices\ p,\ edges\ p)))\ (topological\_component\_yfan\ (vec\ (0{::}nat),\ vertices\ p,\ edges\ p))$

thm DEF_map3:

$map3 = (\lambda(\_3427058{::}(real,\ 3)\ cart)\ \_3427059{::}(real,\ 3)\ cart.\ lambda\ (\lambda i{::}nat.\ HOL\_Light\_Import.floor\ (real\_of\_nat\ (2{::}nat) * (\$\ \_3427058\ i - \$\ \_3427059\ i))))$

thm Pack1.map3:

$\forall\, (x{::}(real,\ 3)\ cart)\ p{::}(real,\ 3)\ cart.\ map3\ x\ p = lambda\ (\lambda i{::}nat.\ HOL\_Light\_Import.floor\ (real\_of\_nat\ (2{::}nat) * (\$\ x\ i - \$\ p\ i)))$

thm Pack1.bound_square:

$\forall\, (a{::}real)\ (b{::}real)\ c{::}real.\ a \leq b \wedge b \leq c \longrightarrow b^2 \leq max\ (a^2)\ (c^2)$

thm Pack1.cauchy_ineq:

$\forall\, (a{::}real)\ b{::}real.\ (a + b)^2 \leq real\_of\_nat\ (2{::}nat) * (a^2 + b^2)$

thm Pack1.bdt_emveque:

$\forall\, r{::}real.\ (0{::}real) \leq real\_of\_nat\ (8{::}nat) * r^2 + real\_of\_nat\ (6{::}nat)$

thm Pack1.norm_abs:

$\forall\, x{::}(real,\ 3)\ cart.\ vector\_norm\ x = |vector\_norm\ x|$

thm Pack1.bdt_emnguchua:

$\forall\, k{::}real.\ HOL\_Light\_Import.floor\ (real\_of\_nat\ (2{::}nat) * k) * HOL\_Light\_Import.floor\ (real\_of\_nat\ (2{::}nat) * k) \leq real\_of\_nat\ (2{::}nat) * (real\_of\_nat\ (4{::}nat) * k^2 + (1{::}real))$

thm Pack1.map3_define:

$\forall\, (v{::}(real,\ 3)\ cart)\ (p{::}(real,\ 3)\ cart)\ r{::}real.\ (0{::}real) \leq r \wedge IN\ v\ (ball\ (p,\ r)) \longrightarrow IN\ (map3\ p\ v)\ (ball\ (vec\ (0{::}nat),\ sqrt\ (real\_of\_nat\ (8{::}nat) * r^2 + real\_of\_nat\ (6{::}nat))))$

thm Pack1.floor_ineq:

$\forall\, (x{::}real)\ y{::}real.\ HOL\_Light\_Import.floor\ x = HOL\_Light\_Import.floor\ y \longrightarrow |x - y| < (1{::}real)$

thm Pack1.bdt_canbatrenbon:

sqrt (*real_of_nat* (*3*::*nat*) / *real_of_nat* (*4*::*nat*)) < *real_of_nat* (*2*::*nat*)

thm Counting_spheres.inj_int_ball:

∀ (*p*::(*real*, *3*) *cart*) (*r*::*real*) *S*::(*real*, *3*) *cart* ⇒ *bool*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *INJ* (*map3 p*) (*HOL_Light_Import.INTER S* (*ball* (*p*, *r*))) (*int_ball* (*vec* (*0*::*nat*)) (*sqrt* (*real_of_nat* (*8*::*nat*) ∗ r^2 + *real_of_nat* (*6*::*nat*))))

thm Pack1.KIUMVTC:

∀ (*p*::(*real*, *3*) *cart*) (*r*::*real*) *S*::(*real*, *3*) *cart* ⇒ *bool*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *FINITE* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r*)))

thm Pack1.voronoi_open:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *voronoi_open S v* = *GSPEC* (λ*GEN%PVAR%767*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%767* (∀ *w*::(*real*, *?'a*::*type*) *cart*. *S w* ∧ *w* ≠ *v* ⟶ *distance* (*x*, *v*) < *distance* (*x*, *w*)) *x*)

thm Pack1.bis:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *bis u v* = *GSPEC* (λ*GEN%PVAR%768*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%768* (*distance* (*x*, *u*) = *distance* (*x*, *v*)) *x*)

thm DEF_nua_kg:

nua_kg = (λ(*_3427637*::(*real*, *3*) *cart*) *_3427638*::(*real*, *3*) *cart*. *GSPEC* (λ*GEN%PVAR%769*::(*real*, *3*) *cart*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%769* (*distance* (*x*, *_3427637*) < *distance* (*x*, *_3427638*)) *x*))

thm Pack1.nua_kg:

∀ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *nua_kg u v* = *GSPEC* (λ*GEN%PVAR%769*::(*real*, *3*) *cart*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%769* (*distance* (*x*, *u*) < *distance* (*x*, *v*)) *x*)

thm Pack1.voronoi_version2:

∀ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* ⇒ *bool*. *voronoi_open S v* = *INTERS* (*GSPEC* (λ*GEN%PVAR%774*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%774* (*IN y* (*DELETE S v*) ∧ *x* = *v*) (*nua_kg x y*)))

thm Pack1.norm_ineq_lt:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. (*vector_norm x* < *vector_norm y*) = (*dot x x* < *dot y y*)

thm Pack1.nua_kg_version2:

∀ (*v*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. ∃ (*a*::(*real*, *3*) *cart*) *b*::*real*. *nua_kg v y* = *GSPEC* (λ*GEN%PVAR%775*::(*real*, *3*) *cart*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%775* (*dot x a* < *b*) *x*)

thm Pack1.convex_nua_kg:

$\forall (v::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ convex\ (nua\_kg\ v\ y)$

thm Pack1.convex_voronoi:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ convex\ (voronoi\_open\ S\ v)$

thm Pack1.bound_voronoi:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ saturated\ S \longrightarrow bounded\ (voronoi\_open\ S\ v)$

thm Pack1.open_nua_kg:

$\forall (v::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ HOL\_Light\_Import.open\ (nua\_kg\ v\ y)$

thm DEF_map_to_nua_kg:

$map\_to\_nua\_kg = (\lambda\_3427878::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ nua\_kg\ (fst\ \_3427878)\ (snd\ \_3427878))$

thm Pack1.map_to_nua_kg:

$\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ map\_to\_nua\_kg\ (x,\ y) = nua\_kg\ x\ y$

thm Pack1.surj_map_to_nua_kg:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ IMAGE\ map\_to\_nua\_kg\ (CROSS\ (INSERT\ v\ EMPTY)\ (HOL\_Light\_Import.INTER\ (DELETE\ S\ v)\ (ball\ (v,\ real\_of\_nat\ (4::nat))))) = GSPEC\ (\lambda GEN\%PVAR\%777::(real,\ 3)\ cart \Rightarrow bool.\ \exists (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%777\ (IN\ y\ (DELETE\ S\ v) \wedge x = v \wedge IN\ y\ (ball\ (v,\ real\_of\_nat\ (4::nat))))\ (nua\_kg\ x\ y))$

thm Pack1.finite_voronoi2:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%778::(real,\ 3)\ cart \Rightarrow bool.\ \exists (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%778\ (IN\ y\ (DELETE\ S\ v) \wedge x = v \wedge IN\ y\ (ball\ (v,\ real\_of\_nat\ (4::nat))))\ (nua\_kg\ x\ y)))$

thm Pack1.real_sub_norm:

$\forall (x::(real,\ 3)\ cart)\ (y::(real,\ 3)\ cart)\ z::(real,\ 3)\ cart.\ distance\ (x,\ z) - distance\ (y,\ z) \leq distance\ (x,\ y)$

thm Pack1.not_open:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ (\neg\ HOL\_Light\_Import.open\ s) = (\exists (a::(real,\ ?'a::type)\ cart)\ x::nat \Rightarrow (real,\ ?'a::type)\ cart.\ IN\ a\ s \wedge (\forall n::nat.\ \neg\ IN\ (x\ n)\ s) \wedge --> x\ a\ sequentially)$

thm Pack1.not_open_voronoi1:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (y::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ A::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ A \wedge --> x\ w\ sequentially \wedge (\forall n::nat.\ distance\ (x\ n,\ y\ n) \leq distance$

$(x\ n,\ v)) \wedge \neg (\exists N::nat.\ \forall n>N.\ \neg\ IN\ (y\ n)\ A) \longrightarrow (\exists a::(real,\ ?'a::type)\ cart.\ IN\ a\ A \wedge distance\ (w,\ a) \leq distance\ (w,\ v))$

thm Pack1.not_open_voronoi2:

$\forall (x::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ {-}{-}{>}\ x\ w\ sequentially \wedge (\exists N::nat.\ \forall n>N.\ real\_of\_nat\ (2::nat) \leq distance\ (x\ n,\ v)) \longrightarrow real\_of\_nat\ (2::nat) \leq distance\ (w,\ v)$

thm Pack1.not_in_voronoi:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ (\neg\ IN\ x\ (voronoi\_open\ S\ v)) = (\exists y::(real,\ 3)\ cart.\ IN\ y\ S \wedge y \neq v \wedge distance\ (x,\ y) \leq distance\ (x,\ v))$

thm Pack1.not_open_voronoi3:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ \neg\ HOL\_Light\_Import.open\ (voronoi\_open\ S\ v) \longrightarrow (\exists (x::nat \Rightarrow (real,\ 3)\ cart)\ (a::(real,\ 3)\ cart)\ y::nat \Rightarrow (real,\ 3)\ cart.\ IN\ a\ (voronoi\_open\ S\ v) \wedge (\forall n::nat.\ \neg\ IN\ (x\ n)\ (voronoi\_open\ S\ v)) \wedge {-}{-}{>}\ x\ a\ sequentially \wedge (\forall n::nat.\ IN\ (y\ n)\ S \wedge y\ n \neq v \wedge distance\ (x\ n,\ y\ n) \leq distance\ (x\ n,\ v)))$

thm Pack1.voronoi_in_ball:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge saturated\ S \wedge IN\ x\ (voronoi\_open\ S\ v) \longrightarrow distance\ (x,\ v) < real\_of\_nat\ (2::nat)$

thm Pack1.open_voronoi:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge saturated\ S \longrightarrow HOL\_Light\_Import.open\ (voronoi\_open\ S\ v)$

thm Pack1.DRUQUFE:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge saturated\ S \longrightarrow convex\ (voronoi\_open\ S\ v) \wedge bounded\ (voronoi\_open\ S\ v) \wedge HOL\_Light\_Import.open\ (voronoi\_open\ S\ v) \wedge measurable\ (voronoi\_open\ S\ v)$

thm Pack1.measurable_voronoi:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge saturated\ S \longrightarrow measurable\ (voronoi\_open\ S\ v)$

thm DEF_negligible_fun_p:

$negligible\_fun\_p = (\lambda(\_3429016::(real,\ ?'a::type)\ cart \Rightarrow real)\ (\_3429017::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_3429018::(real,\ ?'a::type)\ cart.\ \exists C\geq 0::real.\ \forall r\geq 1::real.\ sum\ (HOL\_Light\_Import.INTER\ \_3429017\ (ball\ (\_3429018,\ r)))\ \_3429016 \leq C * r^2)$

thm Pack1.negligible_fun_p:

$\forall (S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (p::(real,\ ?'a::type)\ cart)\ f::(real,\ ?'a::type)\ cart \Rightarrow real.\ negligible\_fun\_p\ f\ S\ p = (\exists C\geq 0::real.\ \forall r\geq 1::real.\ sum\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r)))\ f \leq C * r^2)$

thm DEF_fcc_compatible:

fcc_compatible = (λ(*_3429037*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *_3429038*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∀ *v*::(*real*, *?'a*::*type*) *cart*. *IN v _3429038* ⟶ *sqrt* (*real_of_nat* (*32*::*nat*)) ≤ *HOL_Light_Import*.*measure* (*voronoi_open _3429038* *v*) + *_3429037 v*)

thm Pack1.fcc_compatible:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*. *fcc_compatible* *f S* = (∀ *v*::(*real*, *?'a*::*type*) *cart*. *IN v S* ⟶ *sqrt* (*real_of_nat* (*32*::*nat*)) ≤ *HOL_Light_Import*.*measure* (*voronoi_open S v*) + *f v*)

thm Pack1.packing_subset_unions_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *SUBSET* (*HOL_Light_Import*.*INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%779*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%779* (*IN v S*) (*ball* (*v*, *1*::*real*))))) (*ball* (*p*, *r*))) (*UNIONS* (*GSPEC* (λ*GEN%PVAR%780*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%780* (*IN v* (*HOL_Light_Import*.*INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*)))))

thm Pack1.measurable_packing_lm1:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *measurable* (*HOL_Light_Import*.*INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%782*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%782* (*IN v S*) (*ball* (*v*, *1*::*real*))))) (*ball* (*p*, *r*)))

thm DEF_map_to_ball:

map_to_ball = (λ*_3429209*::(*real*, *3*) *cart*. *ball* (*_3429209*, *1*::*real*))

thm Pack1.map_to_ball:

∀ *x*::(*real*, *3*) *cart*. *map_to_ball x* = *ball* (*x*, *1*::*real*)

thm Pack1.surj_map_to_ball:

∀ *S*::(*real*, *3*) *cart* ⇒ *bool*. *IMAGE map_to_ball S* = *GSPEC* (λ*GEN%PVAR%783*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%783* (*IN x S*) (*ball* (*x*, *1*::*real*)))

thm Pack1.finite_set_packing_in_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%784*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%784* (*IN v* (*HOL_Light_Import*.*INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*))))

thm Pack1.measurable_packing_lm2:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%785*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%785* (*IN v* (*HOL_Light_Import*.*INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*)))))

thm Pack1.measure_ineq_lm53_1:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%786*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%786* (*IN v S*) (*ball* (*v*, *1*::*real*))))) (*ball* (*p*, *r*))) ≤ *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%787*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%787* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*)))))

thm Pack1.measure_ineq_lm53_2:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *HOL_Light_Import.measure* (*HOL_Light_Import.INTER* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%797*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%797* (*IN v S*) (*ball* (*v*, *1*::*real*))))) (*ball* (*p*, *r*))) ≤ *real_of_nat* (*CARD* (*GSPEC* (λ*GEN%PVAR%798*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%798* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*))))) ∗ (*real_of_nat* (*4*::*nat*) ∗ (*pi* / *real_of_nat* (*3*::*nat*)))

thm Pack1.card_eq_ball_point:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%799*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%799* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*))))) (*ball* (*v*, *1*::*real*)))) = *CARD* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*))))

thm Pack1.voronoi_subset_ball:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* ⇒ *bool*. *packing S* ∧ *saturated S* ⟶ *SUBSET* (*voronoi_open S v*) (*ball* (*v*, *real_of_nat* (*2*::*nat*)))

thm Pack1.all_voronoi_subset_ball:

∀ (*v*::(*real*, *3*) *cart*) (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* ∧ *saturated S* ∧ *IN v* (*ball* (*p*, *r* + (*1*::*real*))) ⟶ *SUBSET* (*voronoi_open S v*) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm Pack1.unions_voronoi_subset_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* ∧ *saturated S* ⟶ *SUBSET* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%800*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%800* (*IN v* (*ball* (*p*, *r* + (*1*::*real*)))) (*voronoi_open S v*)))) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm Pack1.unions_voronoi_center_in_ball_subset_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* ∧ *saturated S* ⟶ *SUBSET* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%801*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%801* (*IN*

v (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open* *w v*)))) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm DEF_map_to_voronoi:

map_to_voronoi = (λ*_3429443*::(*real*, *3*) *cart* × ((*real*, *3*) *cart* ⇒ *bool*). *voronoi_open* (*snd _3429443*) (*fst _3429443*))

thm Pack1.map_to_voronoi:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) *x*::(*real*, *3*) *cart*. *map_to_voronoi* (*x*, *S*) = *voronoi_open* *S x*

thm Pack1.surj_map_to_voronoi:

∀ (*M*::(*real*, *3*) *cart* ⇒ *bool*) *S*::(*real*, *3*) *cart* ⇒ *bool*. *IMAGE map_to_voronoi* (*CROSS M* (*INSERT S EMPTY*)) = *GSPEC* (λ*GEN%PVAR%802*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%802* (*IN v M*) (*voronoi_open S v*))

thm Pack1.surj_map_to_voronoi_db:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *IMAGE map_to_voronoi* (*CROSS* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) (*INSERT S* *EMPTY*)) = *GSPEC* (λ*GEN%PVAR%803*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%803* (*IN v* (*HOL_Light_Import.INTER* *S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open w v*))

thm Pack1.finite_set_voronoi_center_in_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%804*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%804* (*IN v* (*HOL_Light_Import.INTER* *S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open w v*)))

thm Pack1.measurable_unions_voronoi:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ∧ *packing S* ∧ *saturated S* ⟶ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%805*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%805* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open* *w v*))))

thm Pack1.negligible_voronoi:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) (*r*::*real*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) *t*::(*real*, *3*) *cart* ⇒ *bool*. *IN s* (*GSPEC* (λ*GEN%PVAR%806*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%806* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open* *w v*))) ∧ *IN t* (*GSPEC* (λ*GEN%PVAR%807*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%807* (*IN v* (*HOL_Light_Import.INTER* *S* (*ball* (*p*, *r* + (*1*::*real*)))) ∧ *w* = *S*) (*voronoi_open w v*))) ∧ *s* ≠ *t* ⟶ *negligible* (*HOL_Light_Import.INTER s t*)

thm Pack1.inj_map_to_voronoi:

$\forall\,(S::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ (r::real)\ (x::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool))\ y::(real,\ 3)\ cart \times ((real,\ 3)\ cart \Rightarrow bool).\ IN\ x\ (CROSS\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))\ (INSERT\ S\ EMPTY)) \land IN\ y\ (CROSS\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))\ (INSERT\ S\ EMPTY)) \land map\_to\_voronoi\ x = map\_to\_voronoi\ y \longrightarrow x = y$

thm Pack1.measure_unions_sum_voronoi:

$\forall\,(S::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ r::real.\ (0::real) \le r \land packing\ S \land saturated\ S \longrightarrow HOL\_Light\_Import.measure\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%809::(real,\ 3)\ cart \Rightarrow bool.\ \exists\,(w::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%809\ (IN\ v\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real)))) \land w = S)\ (voronoi\_open\ w\ v)))) = sum\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))\ (\lambda v::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (voronoi\_open\ S\ v))$

thm Pack1.sum_measure_voronoi_le_ball:

$\forall\,(S::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ r::real.\ (0::real) \le r \land packing\ S \land saturated\ S \longrightarrow sum\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))\ (\lambda v::(real,\ 3)\ cart.\ HOL\_Light\_Import.measure\ (voronoi\_open\ S\ v)) \le HOL\_Light\_Import.measure\ (ball\ (p,\ r + real\_of\_nat\ (3::nat)))$

thm Pack1.ineq_lm5_3_step3:

$\forall\,(S::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ (r::real)\ A::(real,\ 3)\ cart \Rightarrow real.\ (0::real) \le r \land packing\ S \land saturated\ S \land fcc\_compatible\ A\ S \longrightarrow sqrt\ (real\_of\_nat\ (32::nat)) * real\_of\_nat\ (CARD\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))) \le sum\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r + (1::real))))\ (\lambda v::(real,\ 3)\ cart.\ A\ v + HOL\_Light\_Import.measure\ (voronoi\_open\ S\ v))$

thm Pack1.ineq_lm5_3_step4:

$\forall\,c \ge 0::real.\ \exists\,c'::real.\ \forall\,r \ge 1::real.\ pi\ /\ sqrt\ (real\_of\_nat\ (18::nat)) * ((1::real) + real\_of\_nat\ (3::nat)\ /\ r)^{3::nat} + c * ((r + (1::real))^2\ /\ (r^{3::nat} * sqrt\ (real\_of\_nat\ (32::nat)))) \le pi\ /\ sqrt\ (real\_of\_nat\ (18::nat)) + c'\ /\ r$

thm Pack1.JGXZYGW:

$\forall\,(S::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart.\ packing\ S \land saturated\ S \land (\exists\,A::(real,\ 3)\ cart \Rightarrow real.\ fcc\_compatible\ A\ S \land negligible\_fun\_p\ A\ S\ p) \longrightarrow (\exists\,c::real.\ \forall\,r \ge 1::real.\ HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%815::(real,\ 3)\ cart \Rightarrow bool.\ \exists\,v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%815\ (IN\ v\ S)\ (ball\ (v,\ 1::real)))))\ (ball\ (p,\ r)))\ /\ HOL\_Light\_Import.measure\ (ball\ (p,\ r)) \le pi\ /\ sqrt\ (real\_of\_nat\ (18::nat)) + c\ /\ r)$

thm Pack2.PACKING:

$\forall\,s::(real,\ 3)\ cart \Rightarrow bool.\ packing\ s = (\forall\,(u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ s \land IN\ v\ s \land distance\ (u,\ v) < real\_of\_nat\ (2::nat) \longrightarrow u = v)$

thm Pack2.VORONOI_OPEN:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_open\ s\ v = GSPEC\ (\lambda GEN\%PVAR\%816::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%816\ (\forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ s \land w \neq v \longrightarrow distance\ (x,\ v) < distance\ (x,\ w))\ x)$

thm Pack2.VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_closed\ s\ v = GSPEC\ (\lambda GEN\%PVAR\%817::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%817\ (\forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ s \longrightarrow distance\ (x,\ v) \leq distance\ (x,\ w))\ x)$

thm Pack2.KIUMVTC:

$\forall (p::(real,\ 3)\ cart)\ (r::real)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \longrightarrow FINITE\ (HOL\_Light\_Import.INTER\ S\ (ball\ (p,\ r)))$

thm Pack2.BIS_LE:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ bis\_le\ u\ v = GSPEC\ (\lambda GEN\%PVAR\%818::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%818\ (dot\ (\%\ (real\_of\_nat\ (2::nat))\ (vector\_sub\ v\ u))\ x \leq (vector\_norm\ v)^2 - (vector\_norm\ u)^2)\ x)$

thm Pack2.BIS_LT:

$\forall (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ bis\_lt\ u\ v = GSPEC\ (\lambda GEN\%PVAR\%819::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%819\ (dot\ (\%\ (real\_of\_nat\ (2::nat))\ (vector\_sub\ v\ u))\ x < (vector\_norm\ v)^2 - (vector\_norm\ u)^2)\ x)$

thm Pack2.VORONOI_CLOSED_ALT:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_closed\ s\ v = GSPEC\ (\lambda GEN\%PVAR\%820::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%820\ (\forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ s \land w \neq v \longrightarrow distance\ (x,\ v) \leq distance\ (x,\ w))\ x)$

thm Pack2.VORONOI_CLOSED_AS_INTERSECTION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_closed\ s\ v = INTERS\ (IMAGE\ (bis\_le\ v)\ (DELETE\ s\ v))$

thm Pack2.VORONOI_OPEN_AS_INTERSECTION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ voronoi\_open\ s\ v = INTERS\ (IMAGE\ (bis\_lt\ v)\ (DELETE\ s\ v))$

thm Pack2.CLOSED_VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ HOL\_Light\_Import.closed\ (voronoi\_closed\ s\ v)$

thm Pack2.VORONOI_CLOSED_SUBSET_BALL:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ saturated\ s \longrightarrow SUBSET\ (voronoi\_closed\ s\ v)\ (ball\ (v,\ real\_of\_nat\ (2::nat)))$

thm Pack2.BOUNDED_VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ saturated\ s \longrightarrow bounded\ (voronoi\_closed\ s\ v)$

thm Pack2.COMPACT_VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ saturated\ s \longrightarrow compact\ (voronoi\_closed\ s\ v)$

thm Pack2.CONVEX_VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ convex\ (voronoi\_closed\ s\ v)$

thm Pack2.BASE_IN_VORONOI_CLOSED:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ v::(real,\ ?'a::type)\ cart.\ IN\ v\ (voronoi\_closed\ s\ v)$

thm Pack2.CBALL_SUBSET_VORONOI_CLOSED:

$\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ packing\ s \wedge IN\ v\ s \longrightarrow SUBSET\ (cball\ (v,\ 1::real))\ (voronoi\_closed\ s\ v)$

thm Pack2.VORONOI_CLOSED_PARTITION_STRONG:

$\forall s::(real,\ 3)\ cart \Rightarrow bool.\ HOL\_Light\_Import.closed\ s \wedge s \neq EMPTY \longrightarrow UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%821::(real,\ 3)\ cart \Rightarrow bool.\ \exists v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%821\ (IN\ v\ s)\ (voronoi\_closed\ s\ v))) = HOL\_Light\_Import.UNIV$

thm Pack2.PACKING_IMP_CLOSED:

$\forall s::(real,\ 3)\ cart \Rightarrow bool.\ packing\ s \longrightarrow HOL\_Light\_Import.closed\ s$

thm Pack2.SATURATED_IMP_NONEMPTY:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ saturated\ s \longrightarrow s \neq EMPTY$

thm Pack2.VORONOI_CLOSED_PARTITION:

$\forall s::(real,\ 3)\ cart \Rightarrow bool.\ packing\ s \wedge saturated\ s \longrightarrow UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%822::(real,\ 3)\ cart \Rightarrow bool.\ \exists v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%822\ (IN\ v\ s)\ (voronoi\_closed\ s\ v))) = HOL\_Light\_Import.UNIV$

thm Pack2.VORONOI_CLOSED_AS_FINITE_INTERSECTION:

$\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ packing\ s \wedge saturated\ s \wedge IN\ v\ s \longrightarrow voronoi\_closed\ s\ v = INTERS\ (IMAGE\ (bis\_le\ v)\ (DELETE\ (HOL\_Light\_Import.INTER\ s\ (ball\ (v,\ real\_of\_nat\ (4::nat))))\ v))$

thm Pack2.POLYHEDRON_VORONOI_CLOSED:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *packing s* ∧ *saturated s* ∧ *IN v s* ⟶ *polyhedron* (*voronoi_closed s v*)

thm Pack2.POLYTOPE_VORONOI_CLOSED:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *packing s* ∧ *saturated s* ∧ *IN v s* ⟶ *polytope* (*voronoi_closed s v*)

thm Pack2.MEASURABLE_VORONOI_CLOSED:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *saturated s* ⟶ *measurable* (*voronoi_closed s v*)

thm Pack2.CLOSED_BIS_LE:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.closed* (*bis_le u v*)

thm Pack2.OPEN_BIS_LT:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.open* (*bis_lt u v*)

thm Pack2.INTERIOR_BIS_LE:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *v* ≠ *u* ⟶ *interior* (*bis_le u v*) = *bis_lt u v*

thm Pack2.CLOSURE_BIS_LT:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *v* ≠ *u* ⟶ *closure* (*bis_lt u v*) = *bis_le u v*

thm Pack2.CLOSURE_VORONOI_OPEN:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *closure* (*voronoi_open S v*) = *voronoi_closed S v*

thm Pack2.INTERIOR_VORONOI_CLOSED_INTERIOR:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *interior* (*voronoi_closed S v*) = *interior* (*voronoi_open S v*)

thm Pack2.INTERIOR_VORONOI_CLOSED:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *packing S* ∧ *saturated S* ⟶ *interior* (*voronoi_closed S v*) = *voronoi_open S v*

thm Pack2.VORONOI_OPEN_AS_FINITE_INTERSECTION:

∀ (*s*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *packing s* ∧ *saturated s* ∧ *IN v s* ⟶ *voronoi_open s v* = *INTERS* (*IMAGE* (*bis_lt v*) (*DELETE* (*HOL_Light_Import.INTER s* (*ball* (*v*, *real_of_nat* (*4*::*nat*)))) *v*))

thm Pack2.VORONOI_OPEN_SUBSET_CLOSED:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*voronoi_open S v*) (*voronoi_closed S v*)

thm Pack2.MEASURE_VORONOI_CLOSED_OPEN:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import*.*measure* (*voronoi_closed s v*) = *HOL_Light_Import*.*measure* (*voronoi_open s v*)

thm Pack2.INTER_VORONOI_SUBSET_BISECTOR:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u s* $\wedge$ *IN v s* $\longrightarrow$ *SUBSET* (*HOL_Light_Import*.*INTER* (*voronoi_closed s u*) (*voronoi_closed s v*)) (*GSPEC* (λ*GEN%PVAR%823*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%823* (*dot* (% (*real_of_nat* (*2*::*nat*)) (*vector_sub u v*)) *x* = (*vector_norm u*)^2 − (*vector_norm v*)^2) *x*))

thm Pack2.NEGLIGIBLE_INTER_VORONOI_CLOSED:

$\forall$ (*s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN u s* $\wedge$ *IN v s* $\wedge$ *u* $\neq$ *v* $\longrightarrow$ *negligible* (*HOL_Light_Import*.*INTER* (*voronoi_closed s u*) (*voronoi_closed s v*))

thm Pack2.voronoi_version2:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *voronoi_closed S v* = *INTERS* (*GSPEC* (λ*GEN%PVAR%825*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%825* (*IN y* (*DELETE S v*) $\wedge$ *x* = *v*) (*bis_le x y*)))

thm Pack2.convex_voronoi:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *convex* (*voronoi_closed S v*)

thm Pack2.bound_voronoi:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *saturated S* $\longrightarrow$ *bounded* (*voronoi_closed S v*)

thm Pack2.finite_voronoi2:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *packing S* $\longrightarrow$ *FINITE* (*GSPEC* (λ*GEN%PVAR%828*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%828* (*IN y* (*DELETE S v*) $\wedge$ *x* = *v* $\wedge$ *IN y* (*ball* (*v*, *real_of_nat* (*4*::*nat*)))) (*bis_le x y*)))

thm Pack2.not_in_voronoi:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. ($\neg$ *IN x* (*voronoi_closed S v*)) = ($\exists$ *y*::(*real*, *3*) *cart*. *IN y S* $\wedge$ *y* $\neq$ *v* $\wedge$ *distance* (*x*, *y*) < *distance* (*x*, *v*))

thm Pack2.voronoi_in_ball:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *packing S* $\wedge$ *saturated S* $\wedge$ *IN x* (*voronoi_closed S v*) $\longrightarrow$ *distance* (*x*, *v*) < *real_of_nat* (*2*::*nat*)

thm Pack2.DRUQUFE:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *packing S* $\wedge$ *saturated S* $\longrightarrow$ *convex* (*voronoi_closed S v*) $\wedge$ *bounded* (*voronoi_closed S v*) $\wedge$ *HOL_Light_Import.closed* (*voronoi_closed S v*) $\wedge$ *measurable* (*voronoi_closed S v*)

thm Pack2.measurable_voronoi:

$\forall$ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *packing S* $\wedge$ *saturated S* $\longrightarrow$ *measurable* (*voronoi_closed S v*)

thm Pack2.fcc_compatible:

fcc_compatible (*?f*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *real*) (*?S*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) = ($\forall$ *v*::(*real*, *?'a*::*type*) *cart*. *IN v ?S* $\longrightarrow$ *sqrt* (*real_of_nat* (*32*::*nat*)) $\leq$ *HOL_Light_Import.measure* (*voronoi_closed ?S v*) + *?f v*)

thm Pack2.voronoi_subset_ball:

$\forall$ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *packing S* $\wedge$ *saturated S* $\longrightarrow$ *SUBSET* (*voronoi_closed S v*) (*ball* (*v*, *real_of_nat* (*2*::*nat*)))

thm Pack2.all_voronoi_subset_ball:

$\forall$ (*v*::(*real*, *3*) *cart*) (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* $\wedge$ *saturated S* $\wedge$ *IN v* (*ball* (*p*, *r* + (*1*::*real*))) $\longrightarrow$ *SUBSET* (*voronoi_closed S v*) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm Pack2.unions_voronoi_subset_ball:

$\forall$ (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* $\wedge$ *saturated S* $\longrightarrow$ *SUBSET* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%829*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%829* (*IN v* (*ball* (*p*, *r* + (*1*::*real*)))) (*voronoi_closed S v*)))) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm Pack2.unions_voronoi_center_in_ball_subset_ball:

$\forall$ (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *packing S* $\wedge$ *saturated S* $\longrightarrow$ *SUBSET* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%831*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*w*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%831* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) $\wedge$ *w* = *S*) (*voronoi_closed w v*)))) (*ball* (*p*, *r* + *real_of_nat* (*3*::*nat*)))

thm Pack2.finite_set_voronoi_center_in_ball:

$\forall$ (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) $\leq$ *r* $\wedge$ *packing S* $\longrightarrow$ *FINITE* (*GSPEC* (λ*GEN%PVAR%833*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*w*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%833* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1*::*real*)))) $\wedge$ *w* = *S*) (*voronoi_closed w v*)))

thm Pack2.measurable_unions_voronoi:

$\forall$ (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0*::*real*) $\leq$ *r* $\wedge$ *packing S* $\wedge$ *saturated S* $\longrightarrow$ *measurable* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%834*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ (*w*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%834*

(*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) ∧ *w* = *S*) (*voronoi_closed w v*))))

thm Pack2.negligible_voronoi:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) (*r*::*real*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) *t*::(*real*, *3*) *cart* ⇒ *bool*. *IN s* (*GSPEC* (λ*GEN%PVAR%835*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%835* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) ∧ *w* = *S*) (*voronoi_closed w v*))) ∧ *IN t* (*GSPEC* (λ*GEN%PVAR%836*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%836* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) ∧ *w* = *S*) (*voronoi_closed w v*))) ∧ *s* ≠ *t* ⟶ *negligible* (*HOL_Light_Import.INTER s t*)

thm Pack2.measure_unions_sum_voronoi:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0::real*) ≤ *r* ∧ *packing S* ∧ *saturated S* ⟶ *HOL_Light_Import.measure* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%838*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*w*::(*real*, *3*) *cart* ⇒ *bool*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%838* (*IN v* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) ∧ *w* = *S*) (*voronoi_closed w v*)))) = *sum* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) (λ*v*::(*real*, *3*) *cart*. *HOL_Light_Import.measure* (*voronoi_closed S v*))

thm Pack2.sum_measure_voronoi_le_ball:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) *r*::*real*. (*0::real*) ≤ *r* ∧ *packing S* ∧ *saturated S* ⟶ *sum* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) (λ*v*::(*real*, *3*) *cart*. *HOL_Light_Import.measure* (*voronoi_closed S v*)) ≤ *HOL_Light_Import.measure* (*ball* (*p*, *r* + *real_of_nat* (*3::nat*)))

thm Pack2.ineq_lm5_3_step3:

∀ (*S*::(*real*, *3*) *cart* ⇒ *bool*) (*p*::(*real*, *3*) *cart*) (*r*::*real*) *A*::(*real*, *3*) *cart* ⇒ *real*. (*0::real*) ≤ *r* ∧ *packing S* ∧ *saturated S* ∧ *fcc_compatible A S* ⟶ *sqrt* (*real_of_nat* (*32::nat*)) ∗ *real_of_nat* (*CARD* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*))))) ≤ *sum* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r* + (*1::real*)))) (λ*v*::(*real*, *3*) *cart*. *A v* + *HOL_Light_Import.measure* (*voronoi_closed S v*))

thm Pack_defs.negligible_fun_p:

∀ (*S*::(*real*, *?'a::type*) *cart* ⇒ *bool*) (*p*::(*real*, *?'a::type*) *cart*) *f*::(*real*, *?'a::type*) *cart* ⇒ *real*. *negligible_fun_p f S p* = (∃ *C*≥*0*::*real*. ∀ *r*≥*1*::*real*. *sum* (*HOL_Light_Import.INTER S* (*ball* (*p*, *r*))) *f* ≤ *C* ∗ r^2)

thm DEF_negligible_fun_0:

negligible_fun_0 = (λ(*_3433113*::(*real*, *3*) *cart* ⇒ *real*) *_3433114*::(*real*, *3*) *cart* ⇒ *bool*. *negligible_fun_p _3433113 _3433114* (*vec* (*0::nat*)))

thm Pack_defs.negligible_fun_0:

∀ (*f*::(*real*, *3*) *cart* ⇒ *real*) *S*::(*real*, *3*) *cart* ⇒ *bool*. *negligible_fun_0 f S* = *negligible_fun_p f S* (*vec* (*0::nat*))

thm Pack_defs.fcc_compatible:

$\forall (S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ f::(real,\ ?'a::type)\ cart \Rightarrow real.\ fcc\_compatible\ f\ S = (\forall v::(real,\ ?'a::type)\ cart.\ IN\ v\ S \longrightarrow sqrt\ (real\_of\_nat\ (32::nat)) \leq HOL\_Light\_Import.measure\ (voronoi\_open\ S\ v) + f\ v)$

thm Pack_defs.kepler_conjecture:

$kepler\_conjecture = (\forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \longrightarrow (\exists c::real.\ \forall r\geq 1::real.\ HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%839::(real,\ 3)\ cart \Rightarrow bool.\ \exists v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%839\ (IN\ v\ V)\ (ball\ (v,\ 1::real)))))\ (ball\ (vec\ (0::nat),\ r)))\ /\ HOL\_Light\_Import.measure\ (ball\ (vec\ (0::nat),\ r)) \leq pi\ /\ sqrt\ (real\_of\_nat\ (18::nat)) + c\ /\ r))$

thm DEF_hl:

$hl = (\lambda\_3433125::(real,\ ?'a::type)\ cart\ list.\ radV\ (set\_of\_list\ \_3433125))$

thm Pack_defs.HL:

$\forall ul::(real,\ ?'a::type)\ cart\ list.\ hl\ ul = radV\ (set\_of\_list\ ul)$

thm DEF_REVERSE_TABLE:

$REVERSE\_TABLE = (SOME\ REVERSE\_TABLE::nat \Rightarrow (nat \Rightarrow ?'a::type) \Rightarrow nat \Rightarrow ?'a::type\ list.\ \forall \_3433514::nat.\ (\forall f::nat \Rightarrow ?'a::type.\ REVERSE\_TABLE\ \_3433514\ f\ (0::nat) = []) \wedge (\forall (f::nat \Rightarrow ?'a::type)\ i::nat.\ REVERSE\_TABLE\ \_3433514\ f\ (Suc\ i) = f\ i\ \#\ REVERSE\_TABLE\ \_3433514\ f\ i))\ (138::nat)$

thm Pack_defs.REVERSE_TABLE:

$REVERSE\_TABLE\ (?f::nat \Rightarrow ?'a::type)\ (0::nat) = [] \wedge REVERSE\_TABLE\ ?f\ (Suc\ (?i::nat)) = ?f\ ?i\ \#\ REVERSE\_TABLE\ ?f\ ?i$

thm DEF_TABLE:

$TABLE = (\lambda(\_3433515::nat \Rightarrow ?'a::type)\ \_3433516::nat.\ rev\ (REVERSE\_TABLE\ \_3433515\ \_3433516))$

thm Pack_defs.TABLE:

$\forall (f::nat \Rightarrow ?'a::type)\ k::nat.\ TABLE\ f\ k = rev\ (REVERSE\_TABLE\ f\ k)$

thm DEF_left_action:

$left\_action = (\lambda(\_3433527::?'c::type \Rightarrow ?'b::type)\ (\_3433528::?'c::type \Rightarrow ?'a::type)\ \_3433529::?'b::type.\ \_3433528\ (HOL\_Light\_Import.inverse\ \_3433527\ \_3433529))$

thm Pack_defs.left_action:

$\forall (p::?'c::type \Rightarrow ?'b::type)\ (f::?'c::type \Rightarrow ?'a::type)\ x::?'b::type.\ left\_action\ p\ f\ x = f\ (HOL\_Light\_Import.inverse\ p\ x)$

thm DEF_left_action_list:

left_action_list = (λ(*_3433548*::*nat* ⇒ *nat*) *_3433549*::*?'a*::*type list*. *TABLE* (λ*i*::*nat*. *EL* (*HOL_Light_Import*.*inverse* *_3433548* *i*) *_3433549*) (*length* *_3433549*))

thm Pack_defs.left_action_list:

∀ (*p*::*nat* ⇒ *nat*) *ul*::*?'a*::*type list*. *left_action_list* *p* *ul* = *TABLE* (λ*i*::*nat*. *EL* (*HOL_Light_Import*.*inverse* *p* *i*) *ul*) (*length* *ul*)

thm DEF_DROP:

DROP = (*SOME DROP*::*nat* ⇒ *?'a*::*type list* ⇒ *nat* ⇒ *?'a*::*type list*. ∀ *_3434012*::*nat*. (∀ *ul*::*?'a*::*type list*. *DROP* *_3434012* *ul* (*0*::*nat*) = *tl* *ul*) ∧ (∀ (*ul*::*?'a*::*type list*) *i*::*nat*. *DROP* *_3434012* *ul* (*Suc* *i*) = *hd* *ul* # *DROP* *_3434012* (*tl* *ul*) *i*)) (*139*::*nat*)

thm Pack_defs.DROP:

DROP (*?ul*::*?'a*::*type list*) (*0*::*nat*) = *tl* *?ul* ∧ *DROP* *?ul* (*Suc* (*?i*::*nat*)) = *hd* *?ul* # *DROP* (*tl* *?ul*) *?i*

thm DEF_mxi:

mxi = (λ(*_3434013*::(*real*, *3*) *cart* ⇒ *bool*) *_3434014*::(*real*, *3*) *cart list*. *if* *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl* (*truncate_simplex* (*2*::*nat*) *_3434014*) *then* *omega_list_n* *_3434013* *_3434014* (*2*::*nat*) *else* *SOME* *p*::(*real*, *3*) *cart*. *IN* *p* (*hull convex* (*INSERT* (*omega_list_n* *_3434013* *_3434014* (*2*::*nat*)) (*INSERT* (*omega_list_n* *_3434013* *_3434014* (*3*::*nat*)) *EMPTY*))) ∧ *distance* (*p*, *hd* *_3434014*) = *sqrt* (*real_of_nat* (*2*::*nat*)))

thm Pack_defs.mxi:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mxi* *V* *ul* = (*if* *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl* (*truncate_simplex* (*2*::*nat*) *ul*) *then* *omega_list_n* *V* *ul* (*2*::*nat*) *else* *SOME* *p*::(*real*, *3*) *cart*. *IN* *p* (*hull convex* (*INSERT* (*omega_list_n* *V* *ul* (*2*::*nat*)) (*INSERT* (*omega_list_n* *V* *ul* (*3*::*nat*)) *EMPTY*))) ∧ *distance* (*p*, *hd* *ul*) = *sqrt* (*real_of_nat* (*2*::*nat*)))

thm DEF_mcell0:

mcell0 = (λ(*_3434025*::(*real*, *3*) *cart* ⇒ *bool*) *_3434026*::(*real*, *3*) *cart list*. *DIFF* (*rogers* *_3434025* *_3434026*) (*ball* (*hd* *_3434026*, *sqrt* (*real_of_nat* (*2*::*nat*)))))

thm Pack_defs.mcell0:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mcell0* *V* *ul* = *DIFF* (*rogers* *V* *ul*) (*ball* (*hd* *ul*, *sqrt* (*real_of_nat* (*2*::*nat*))))

thm DEF_mcell1:

mcell1 = (λ(*_3434037*::(*real*, *3*) *cart* ⇒ *bool*) *_3434038*::(*real*, *3*) *cart list*. *if* *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl* *_3434038* *then* *DIFF* (*HOL_Light_Import*.*INTER* (*rogers* *_3434037* *_3434038*) (*cball* (*hd* *_3434038*, *sqrt* (*real_of_nat* (*2*::*nat*))))) (*rcone_gt* (*hd* *_3434038*) (*hd* (*tl* *_3434038*)) (*hl* (*truncate_simplex* (*1*::*nat*) *_3434038*) / *sqrt* (*real_of_nat* (*2*::*nat*)))) *else* *EMPTY*)

thm Pack_defs.mcell1:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mcell1 V ul* = (*if sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl ul then DIFF* (*HOL_Light_Import.INTER* (*rogers V ul*) (*cball* (*hd ul*, *sqrt* (*real_of_nat* (*2*::*nat*))))) (*rcone_gt* (*hd ul*) (*hd* (*tl ul*)) (*hl* (*truncate_simplex* (*1*::*nat*) *ul*) / *sqrt* (*real_of_nat* (*2*::*nat*)))) *else EMPTY*)

thm DEF_mcell2:

mcell2 = (λ(*_3434049*::(*real*, *3*) *cart* ⇒ *bool*) *_3434050*::(*real*, *3*) *cart list*. *if hl* (*truncate_simplex* (*1*::*nat*) *_3434050*) < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl _3434050 then LET* (λ*a*::*real*. *LET_END* (*HOL_Light_Import.INTER* (*rcone_ge* (*hd _3434050*) (*hd* (*tl _3434050*)) *a*) (*HOL_Light_Import.INTER* (*rcone_ge* (*hd* (*tl _3434050*)) (*hd _3434050*) *a*) (*aff_ge* (*INSERT* (*hd _3434050*) (*INSERT* (*hd* (*tl _3434050*)) *EMPTY*)) (*INSERT* (*mxi _3434049 _3434050*) (*INSERT* (*omega_list_n _3434049 _3434050* (*3*::*nat*)) *EMPTY*)))))) (*hl* (*truncate_simplex* (*1*::*nat*) *_3434050*) / *sqrt* (*real_of_nat* (*2*::*nat*))) *else EMPTY*)

thm Pack_defs.mcell2:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mcell2 V ul* = (*if hl* (*truncate_simplex* (*1*::*nat*) *ul*) < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl ul then LET* (λ*a*::*real*. *LET_END* (*HOL_Light_Import.INTER* (*rcone_ge* (*hd ul*) (*hd* (*tl ul*)) *a*) (*HOL_Light_Import.INTER* (*rcone_ge* (*hd* (*tl ul*)) (*hd ul*) *a*) (*aff_ge* (*INSERT* (*hd ul*) (*INSERT* (*hd* (*tl ul*)) *EMPTY*)) (*INSERT* (*mxi V ul*) (*INSERT* (*omega_list_n V ul* (*3*::*nat*)) *EMPTY*)))))) (*hl* (*truncate_simplex* (*1*::*nat*) *ul*) / *sqrt* (*real_of_nat* (*2*::*nat*))) *else EMPTY*)

thm DEF_mcell3:

mcell3 = (λ(*_3434061*::(*real*, *3*) *cart* ⇒ *bool*) *_3434062*::(*real*, *3*) *cart list*. *if hl* (*truncate_simplex* (*2*::*nat*) *_3434062*) < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl _3434062 then hull convex* (*HOL_Light_Import.UNION* (*set_of_list* (*truncate_simplex* (*2*::*nat*) *_3434062*)) (*INSERT* (*mxi _3434061 _3434062*) *EMPTY*)) *else EMPTY*)

thm Pack_defs.mcell3:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mcell3 V ul* = (*if hl* (*truncate_simplex* (*2*::*nat*) *ul*) < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *sqrt* (*real_of_nat* (*2*::*nat*)) ≤ *hl ul then hull convex* (*HOL_Light_Import.UNION* (*set_of_list* (*truncate_simplex* (*2*::*nat*) *ul*)) (*INSERT* (*mxi V ul*) *EMPTY*)) *else EMPTY*)

thm DEF_mcell4:

mcell4 = (λ(*_3434073*::(*real*, *3*) *cart* ⇒ *bool*) *_3434074*::(*real*, *3*) *cart list*. *if hl _3434074* < *sqrt* (*real_of_nat* (*2*::*nat*)) *then hull convex* (*set_of_list _3434074*) *else EMPTY*)

thm Pack_defs.mcell4:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *mcell4 V ul* = (*if hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*)) *then hull convex* (*set_of_list ul*) *else EMPTY*)

thm DEF_mcell:

mcell = (λ(_3434085::nat) (_3434086::(real, 3) cart ⇒ bool) _3434087::(real, 3) cart list. if _3434085 = (0::nat) then mcell0 _3434086 _3434087 else if _3434085 = (1::nat) then mcell1 _3434086 _3434087 else if _3434085 = (2::nat) then mcell2 _3434086 _3434087 else if _3434085 = (3::nat) then mcell3 _3434086 _3434087 else mcell4 _3434086 _3434087)

thm Pack_defs.mcell:

∀ (i::nat) (V::(real, 3) cart ⇒ bool) ul::(real, 3) cart list. mcell i V ul = (if i = (0::nat) then mcell0 V ul else if i = (1::nat) then mcell1 V ul else if i = (2::nat) then mcell2 V ul else if i = (3::nat) then mcell3 V ul else mcell4 V ul)

thm DEF_mcell_set:

mcell_set = (λ_3434106::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%840::(real, 3) cart ⇒ bool. ∃ X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%840 (∃ (i::nat) ul::(real, 3) cart list. X = mcell i _3434106 ul ∧ IN ul (barV _3434106 (3::nat))) X))

thm Pack_defs.mcell_set:

∀ V::(real, 3) cart ⇒ bool. mcell_set V = GSPEC (λGEN%PVAR%840::(real, 3) cart ⇒ bool. ∃ X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%840 (∃ (i::nat) ul::(real, 3) cart list. X = mcell i V ul ∧ IN ul (barV V (3::nat))) X)

thm DEF_cell_params:

cell_params = (λ(_3434111::(real, 3) cart ⇒ bool) _3434112::(real, 3) cart ⇒ bool. Eps (GABS (λf::nat × (real, 3) cart list ⇒ bool. ∀ (k::nat) ul::(real, 3) cart list. GEQ (f (k, ul)) (k ≤ (4::nat) ∧ IN ul (barV _3434111 (3::nat)) ∧ _3434112 = mcell k _3434111 ul))))

thm Pack_defs.cell_params:

∀ (V::(real, 3) cart ⇒ bool) X::(real, 3) cart ⇒ bool. cell_params V X = Eps (GABS (λf::nat × (real, 3) cart list ⇒ bool. ∀ (k::nat) ul::(real, 3) cart list. GEQ (f (k, ul)) (k ≤ (4::nat) ∧ IN ul (barV V (3::nat)) ∧ X = mcell k V ul)))

thm DEF_VX:

VX = (λ(_3434123::(real, 3) cart ⇒ bool) _3434124::(real, 3) cart ⇒ bool. if negligible _3434124 then EMPTY else LET (GABS (λf::nat × (real, 3) cart list ⇒ (real, 3) cart ⇒ bool. ∀ (k::nat) ul::(real, 3) cart list. GEQ (f (k, ul)) (LET_END (if k = (0::nat) then EMPTY else set_of_list (truncate_simplex (k − (1::nat)) ul))))) (cell_params _3434123 _3434124))

thm Pack_defs.VX:

∀ (V::(real, 3) cart ⇒ bool) X::(real, 3) cart ⇒ bool. VX V X = (if negligible X then EMPTY else LET (GABS (λf::nat × (real, 3) cart list ⇒ (real, 3) *cart* ⇒ *bool*. ∀ (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *GEQ* (*f* (*k*, *ul*)) (*LET_END* (*if k* = (*0*::*nat*) *then EMPTY else set_of_list* (*truncate_simplex* (*k* − (*1*::*nat*)) *ul*))))) (*cell_params V X*))

thm DEF_edgeX:

edgeX = (λ(*_3434135*::(*real*, *3*) *cart* ⇒ *bool*) *_3434136*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%841*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%841* (*VX _3434135 _3434136 u* ∧ *VX _3434135 _3434136 v* ∧ *u* ≠ *v*) (*INSERT u* (*INSERT v EMPTY*))))

thm Pack_defs.edgeX:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *X*::(*real*, *3*) *cart* ⇒ *bool*. *edgeX V X* = *GSPEC* (λ*GEN%PVAR%841*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%841* (*VX V X u* ∧ *VX V X v* ∧ *u* ≠ *v*) (*INSERT u* (*INSERT v EMPTY*)))

thm DEF_total_solid:

total_solid = (λ(*_3434147*::(*real*, *3*) *cart* ⇒ *bool*) *_3434148*::(*real*, *3*) *cart* ⇒ *bool*. *sum* (*VX _3434147 _3434148*) (λ*x*::(*real*, *3*) *cart*. *sol x _3434148*))

thm Pack_defs.total_solid:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*X*::(*real*, *3*) *cart* ⇒ *bool*) *x*::*?'a*::*type*. *total_solid V X* = *sum* (*VX V X*) (λ*x*::(*real*, *3*) *cart*. *sol x X*)

thm DEF_dihu2:

dihu2 = (λ(*_3434159*::(*real*, *3*) *cart* ⇒ *bool*) *_3434160*::(*real*, *3*) *cart list*. *dihV* (*EL* (*0*::*nat*) *_3434160*) (*EL* (*1*::*nat*) *_3434160*) (*mxi _3434159 _3434160*) (*omega_list_n _3434159 _3434160* (*3*::*nat*)))

thm Pack_defs.dihu2:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *dihu2 V ul* = *dihV* (*EL* (*0*::*nat*) *ul*) (*EL* (*1*::*nat*) *ul*) (*mxi V ul*) (*omega_list_n V ul* (*3*::*nat*))

thm DEF_dihu3:

dihu3 = (λ(*_3434171*::(*real*, *3*) *cart* ⇒ *bool*) *_3434172*::(*real*, *3*) *cart list*. *dihV* (*EL* (*0*::*nat*) *_3434172*) (*EL* (*1*::*nat*) *_3434172*) (*EL* (*2*::*nat*) *_3434172*) (*mxi _3434171 _3434172*))

thm Pack_defs.dihu3:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *dihu3 V ul* = *dihV* (*EL* (*0*::*nat*) *ul*) (*EL* (*1*::*nat*) *ul*) (*EL* (*2*::*nat*) *ul*) (*mxi V ul*)

thm DEF_dihu4:

dihu4 = (λ*_3434183*::(*real*, *3*) *cart list*. *dihV* (*EL* (*0*::*nat*) *_3434183*) (*EL* (*1*::*nat*) *_3434183*) (*EL* (*2*::*nat*) *_3434183*) (*EL* (*3*::*nat*) *_3434183*))

thm Pack_defs.dihu4:

$\forall$ *ul*::(*real*, *3*) *cart list*. *dihu4 ul* = *dihV* (*EL* (*0*::*nat*) *ul*) (*EL* (*1*::*nat*) *ul*) (*EL* (*2*::*nat*) *ul*) (*EL* (*3*::*nat*) *ul*)

thm DEF_cell_params_d:

cell_params_d = (λ(*_3434188*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*_3434189*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *_3434190*::(*real*, *3*) *cart list*. *Eps* (*GABS* (λ*f*::*nat* $\times$ (*real*, *3*) *cart list* $\Rightarrow$ *bool*. $\forall$ (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *GEQ* (*f* (*k*, *ul*)) (*k* $\leq$ (*4*::*nat*) $\wedge$ *IN ul* (*barV _3434188* (*3*::*nat*)) $\wedge$ *_3434189* = *mcell k _3434188 ul* $\wedge$ *initial_sublist _3434190 ul*))))

thm Pack_defs.cell_params_d:

$\forall$ (*vl*::(*real*, *3*) *cart list*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *X*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *cell_params_d V X vl* = *Eps* (*GABS* (λ*f*::*nat* $\times$ (*real*, *3*) *cart list* $\Rightarrow$ *bool*. $\forall$ (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *GEQ* (*f* (*k*, *ul*)) (*k* $\leq$ (*4*::*nat*) $\wedge$ *IN ul* (*barV V* (*3*::*nat*)) $\wedge$ *X* = *mcell k V ul* $\wedge$ *initial_sublist vl ul*)))

thm DEF_dihX:

dihX = (λ(*_3434209*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*_3434210*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *_3434211*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. *if negligible _3434210 then 0*::*real else LET* (*GABS* (λ*f*::*nat* $\times$ (*real*, *3*) *cart list* $\Rightarrow$ *real*. $\forall$ (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *GEQ* (*f* (*k*, *ul*)) (*LET_END* (*if k* = (*2*::*nat*) *then dihu2 _3434209 ul else if k* = (*3*::*nat*) *then dihu3 _3434209 ul else if k* = (*4*::*nat*) *then dihu4 ul else* (*0*::*real*))))) (*cell_params_d _3434209 _3434210* [*fst _3434211*, *snd _3434211*]))

thm Pack_defs.dihX:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*X*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*v0*::(*real*, *3*) *cart*) *v1*::(*real*, *3*) *cart*. *dihX V X* (*v0*, *v1*) = (*if negligible X then 0*::*real else LET* (*GABS* (λ*f*::*nat* $\times$ (*real*, *3*) *cart list* $\Rightarrow$ *real*. $\forall$ (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *GEQ* (*f* (*k*, *ul*)) (*LET_END* (*if k* = (*2*::*nat*) *then dihu2 V ul else if k* = (*3*::*nat*) *then dihu3 V ul else if k* = (*4*::*nat*) *then dihu4 ul else* (*0*::*real*))))) (*cell_params_d V X* [*v0*, *v1*]))

thm DEF_marchal:

marchal = (λ*_3434236*::*real*. *if _3434236* $\leq$ *sqrt* (*real_of_nat* (*2*::*nat*)) *then marchal_quartic _3434236 else* (*0*::*real*))

thm Pack_defs.marchal:

$\forall$ *h*::*real*. *marchal h* = (*if h* $\leq$ *sqrt* (*real_of_nat* (*2*::*nat*)) *then marchal_quartic h else* (*0*::*real*))

thm DEF_gammaX:

gammaX = (λ(*_3434241*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*_3434242*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *_3434243*::*real* $\Rightarrow$ *real*. *HOL_Light_Import.measure _3434242* $-$ *real_of_nat* (*2*::*nat*) $*$ (*mm1* / *pi*) $*$ *total_solid _3434241 _3434242* + *real_of_nat* (*8*::*nat*) $*$ (*mm2* / *pi*) $*$ *sum* (*edgeX _3434241 _3434242*) (*GABS* (λ*f*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *real*. $\forall$ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *GEQ* (*f* (*INSERT u*

(*INSERT v EMPTY*))) (*if IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX _3434241 _3434242*) *then dihX _3434241 _3434242* (*u*, *v*) ∗ *_3434243* (*hl* [*u*, *v*]) *else* (*0*::*real*)))))

thm Pack_defs.gammaX:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*X*::(*real*, *3*) *cart* ⇒ *bool*) *f*::*real* ⇒ *real*. *gammaX V X f* = *HOL_Light_Import.measure X* − *real_of_nat* (*2*::*nat*) ∗ (*mm1* / *pi*) ∗ *total_solid V X* + *real_of_nat* (*8*::*nat*) ∗ (*mm2* / *pi*) ∗ *sum* (*edgeX V X*) (*GABS* (λ*fa*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *real*. ∀ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *GEQ* (*fa* (*INSERT u* (*INSERT v EMPTY*))) (*if IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX V X*) *then dihX V X* (*u*, *v*) ∗ *f* (*hl* [*u*, *v*]) *else* (*0*::*real*))))

thm Pack_defs.marchal_inequality:

marchal_inequality = (∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *X*::(*real*, *3*) *cart* ⇒ *bool*. *saturated V* ∧ *packing V* ∧ *mcell_set V X* ⟶ (*0*::*real*) ≤ *gammaX V X marchal*)

thm DEF_critical_edgeX:

critical_edgeX = (λ(*_3434262*::(*real*, *3*) *cart* ⇒ *bool*) *_3434263*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%842*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%842* (*IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX _3434262 _3434263*) ∧ *hminus* ≤ *hl* [*u*, *v*] ∧ *hl* [*u*, *v*] ≤ *hplus*) (*INSERT u* (*INSERT v EMPTY*))))

thm Pack_defs.critical_edgeX:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *X*::(*real*, *3*) *cart* ⇒ *bool*. *critical_edgeX V X* = *GSPEC* (λ*GEN%PVAR%842*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%842* (*IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX V X*) ∧ *hminus* ≤ *hl* [*u*, *v*] ∧ *hl* [*u*, *v*] ≤ *hplus*) (*INSERT u* (*INSERT v EMPTY*)))

thm DEF_subcritical_edgeX:

subcritical_edgeX = (λ(*_3434274*::(*real*, *3*) *cart* ⇒ *bool*) *_3434275*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%843*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%843* (*IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX _3434274 _3434275*) ∧ *hl* [*u*, *v*] < *hminus*) (*INSERT u* (*INSERT v EMPTY*))))

thm Pack_defs.subcritical_edgeX:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *X*::(*real*, *3*) *cart* ⇒ *bool*. *subcritical_edgeX V X* = *GSPEC* (λ*GEN%PVAR%843*::(*real*, *3*) *cart* ⇒ *bool*. ∃ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%843* (*IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX V X*) ∧ *hl* [*u*, *v*] < *hminus*) (*INSERT u* (*INSERT v EMPTY*)))

thm DEF_critical_weight:

critical_weight = (λ(_3434286::(real, 3) cart ⇒ bool) _3434287::(real, 3) cart ⇒ bool. (1::real) / real_of_nat (CARD (critical_edgeX _3434286 _3434287)))

thm Pack_defs.critical_weight:

∀(V::(real, 3) cart ⇒ bool) X::(real, 3) cart ⇒ bool. critical_weight V X = (1::real) / real_of_nat (CARD (critical_edgeX V X))

thm DEF_cell_cluster:

cell_cluster = (λ(_3434298::(real, 3) cart ⇒ bool) _3434299::(real, 3) cart ⇒ bool. GSPEC (λGEN%PVAR%844::(real, 3) cart ⇒ bool. ∃X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%844 (IN _3434299 (critical_edgeX _3434298 X) ∧ mcell_set _3434298 X) X))

thm Pack_defs.cell_cluster:

∀(V::(real, 3) cart ⇒ bool) e::(real, 3) cart ⇒ bool. cell_cluster V e = GSPEC (λGEN%PVAR%844::(real, 3) cart ⇒ bool. ∃X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%844 (IN e (critical_edgeX V X) ∧ mcell_set V X) X)

thm DEF_beta_bump:

beta_bump = (λ(_3434310::(real, 3) cart ⇒ bool) (_3434311::(real, 3) cart ⇒ bool) _3434312::(real, 3) cart ⇒ bool. LET (GABS (λf::nat × (real, 3) cart list ⇒ real. ∀(k::nat) ul::(real, 3) cart list. GEQ (f (k, ul)) (LET_END (sum (GSPEC (λGEN%PVAR%845::((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool) × (nat ⇒ nat) × (real, 3) cart list. ∃(e′::(real, 3) cart ⇒ bool) (e′′::(real, 3) cart ⇒ bool) (p::nat ⇒ nat) vl::(real, 3) cart list. SETSPEC GEN%PVAR%845 (k = (4::nat) ∧ critical_edgeX _3434310 _3434312 = INSERT e′ (INSERT e′′ EMPTY) ∧ _3434311 = e′ ∧ permutes p (dotdot (0::nat) (3::nat)) ∧ vl = left_action_list p ul ∧ e′ = INSERT (EL (0::nat) vl) (INSERT (EL (1::nat) vl) EMPTY) ∧ e′′ = INSERT (EL (2::nat) vl) (INSERT (EL (3::nat) vl) EMPTY)) (e′, e′′, p, vl))) (GABS (λf::((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool) × (nat ⇒ nat) × (real, 3) cart list ⇒ real. ∀(e′::(real, 3) cart ⇒ bool) (e′′::(real, 3) cart ⇒ bool) (p::nat ⇒ nat) vl::(real, 3) cart list. GEQ (f (e′, e′′, p, vl)) ((bump (hl [EL (0::nat) vl, EL (1::nat) vl]) − bump (hl [EL (2::nat) vl, EL (3::nat) vl])) / real_of_nat (4::nat)))))))) (cell_params _3434310 _3434312))

thm Pack_defs.beta_bump:

∀(V::(real, 3) cart ⇒ bool) (e::(real, 3) cart ⇒ bool) X::(real, 3) cart ⇒ bool. beta_bump V e X = LET (GABS (λf::nat × (real, 3) cart list ⇒ real. ∀(k::nat) ul::(real, 3) cart list. GEQ (f (k, ul)) (LET_END (sum (GSPEC (λGEN%PVAR%845::((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool) × (nat ⇒ nat) × (real, 3) cart list. ∃(e′::(real, 3) cart ⇒ bool) (e′′::(real, 3) cart ⇒ bool) (p::nat ⇒ nat) vl::(real, 3) cart list. SETSPEC GEN%PVAR%845 (k = (4::nat) ∧ critical_edgeX V X = INSERT e′ (INSERT e′′ EMPTY) ∧ e = e′ ∧ permutes p (dotdot (0::nat) (3::nat)) ∧ vl = left_action_list p ul ∧ e′ = INSERT (EL (0::nat) vl) (INSERT (EL (1::nat) vl) EMPTY) ∧ e′′ =

$INSERT\ (EL\ (2::nat)\ vl)\ (INSERT\ (EL\ (3::nat)\ vl)\ EMPTY))\ (e',\ e'',\ p,\ vl)))\ (GABS\ (\lambda f::((real,\ 3)\ cart \Rightarrow bool) \times ((real,\ 3)\ cart \Rightarrow bool) \times (nat \Rightarrow nat) \times (real,\ 3)\ cart\ list \Rightarrow real.\ \forall (e'::(real,\ 3)\ cart \Rightarrow bool)\ (e''::(real,\ 3)\ cart \Rightarrow bool)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ GEQ\ (f\ (e',\ e'',\ p,\ vl))\ ((bump\ (hl\ [EL\ (0::nat)\ vl,\ EL\ (1::nat)\ vl]) - bump\ (hl\ [EL\ (2::nat)\ vl,\ EL\ (3::nat)\ vl]))\ /\ real\_of\_nat\ (4::nat))))))))\ (cell\_params\ V\ X)$

thm DEF_cluster_gammaX:

$cluster\_gammaX = (\lambda(\_3434331::(real,\ 3)\ cart \Rightarrow bool)\ (\_3434332::(real,\ 3)\ cart \Rightarrow bool)\ \_3434333::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ sum\ \_3434333\ (\lambda X::(real,\ 3)\ cart \Rightarrow bool.\ gammaX\ \_3434331\ X\ lmfun * critical\_weight\ \_3434331\ X + beta\_bump\ \_3434331\ \_3434332\ X))$

thm Pack_defs.cluster_gammaX:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (e::(real,\ 3)\ cart \Rightarrow bool)\ Z::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ cluster\_gammaX\ V\ e\ Z = sum\ Z\ (\lambda X::(real,\ 3)\ cart \Rightarrow bool.\ gammaX\ V\ X\ lmfun * critical\_weight\ V\ X + beta\_bump\ V\ e\ X)$

thm DEF_cell_cluster_estimate:

$cell\_cluster\_estimate = (\lambda\_3434352::(real,\ 3)\ cart \Rightarrow bool.\ \forall e::(real,\ 3)\ cart \Rightarrow bool.\ (0::real) \leq cluster\_gammaX\ \_3434352\ e\ (cell\_cluster\ \_3434352\ e))$

thm Pack_defs.cell_cluster_estimate:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ cell\_cluster\_estimate\ V = (\forall e::(real,\ 3)\ cart \Rightarrow bool.\ (0::real) \leq cluster\_gammaX\ V\ e\ (cell\_cluster\ V\ e))$

thm DEF_beta_bumpA:

$beta\_bumpA = (\lambda(\_3434357::(real,\ 3)\ cart \Rightarrow bool)\ (\_3434358::(real,\ 3)\ cart \Rightarrow bool)\ \_3434359::(real,\ 3)\ cart \Rightarrow bool.\ LET\ (GABS\ (\lambda f::nat \times (real,\ 3)\ cart\ list \Rightarrow real.\ \forall (k::nat)\ ul::(real,\ 3)\ cart\ list.\ GEQ\ (f\ (k,\ ul))\ (LET\_END\ (sum\ (GSPEC\ (\lambda GEN\%PVAR\%846::((real,\ 3)\ cart \Rightarrow bool) \times ((real,\ 3)\ cart \Rightarrow bool) \times (nat \Rightarrow nat) \times (real,\ 3)\ cart\ list.\ \exists (e'::(real,\ 3)\ cart \Rightarrow bool)\ (e''::(real,\ 3)\ cart \Rightarrow bool)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ SETSPEC\ GEN\%PVAR\%846\ (k = (4::nat) \wedge critical\_edgeX\ \_3434357\ \_3434359 = INSERT\ e'\ (INSERT\ e''\ EMPTY) \wedge \_3434358 = e' \wedge permutes\ p\ (dotdot\ (0::nat)\ (3::nat)) \wedge (\forall e'''::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e'''\ (edgeX\ \_3434357\ \_3434359) \longrightarrow e''' = e' \vee e''' = e'' \vee IN\ e'''\ (subcritical\_edgeX\ \_3434357\ \_3434359)) \wedge vl = left\_action\_list\ p\ ul \wedge e' = INSERT\ (EL\ (0::nat)\ vl)\ (INSERT\ (EL\ (1::nat)\ vl)\ EMPTY) \wedge e'' = INSERT\ (EL\ (2::nat)\ vl)\ (INSERT\ (EL\ (3::nat)\ vl)\ EMPTY))\ (e',\ e'',\ p,\ vl)))\ (GABS\ (\lambda f::((real,\ 3)\ cart \Rightarrow bool) \times ((real,\ 3)\ cart \Rightarrow bool) \times (nat \Rightarrow nat) \times (real,\ 3)\ cart\ list \Rightarrow real.\ \forall (e'::(real,\ 3)\ cart \Rightarrow bool)\ (e''::(real,\ 3)\ cart \Rightarrow bool)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ GEQ\ (f\ (e',\ e'',\ p,\ vl))\ ((bump\ (hl\ [EL\ (0::nat)\ vl,\ EL\ (1::nat)\ vl]) - bump\ (hl\ [EL\ (2::nat)\ vl,\ EL\ (3::nat)\ vl]))\ /\ real\_of\_nat\ (4::nat))))))))\ (cell\_params\ \_3434357\ \_3434359))$

thm Pack_defs.beta_bumpA:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (e::(real,\ 3)\ cart \Rightarrow bool)\ X::(real,\ 3)\ cart \Rightarrow bool.\ beta\_bumpA\ V\ e\ X = LET\ (GABS\ (\lambda f::nat \times (real,\ 3)\ cart\ list \Rightarrow real.\ \forall (k::nat)\ ul::(real,\ 3)\ cart\ list.\ GEQ\ (f\ (k,\ ul))\ (LET\_END\ (sum\ (GSPEC\ (\lambda GEN\%PVAR\%846::((real,\ 3)\ cart \Rightarrow bool) \times ((real,\ 3)\ cart \Rightarrow bool) \times (nat \Rightarrow nat) \times (real,\ 3)\ cart\ list.\ \exists (e'::(real,\ 3)\ cart \Rightarrow bool)\ (e''::(real,\ 3)\ cart \Rightarrow bool)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ SETSPEC\ GEN\%PVAR\%846\ (k = (4::nat) \wedge critical\_edgeX\ V\ X = INSERT\ e'\ (INSERT\ e''\ EMPTY) \wedge e = e' \wedge permutes\ p\ (dotdot\ (0::nat)\ (3::nat)) \wedge (\forall e'''::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e'''\ (edgeX\ V\ X) \longrightarrow e''' = e' \vee e''' = e'' \vee IN\ e'''\ (subcritical\_edgeX\ V\ X)) \wedge vl = left\_action\_list\ p\ ul \wedge e' = INSERT\ (EL\ (0::nat)\ vl)\ (INSERT\ (EL\ (1::nat)\ vl)\ EMPTY) \wedge e'' = INSERT\ (EL\ (2::nat)\ vl)\ (INSERT\ (EL\ (3::nat)\ vl)\ EMPTY))\ (e',\ e'',\ p,\ vl)))\ (GABS\ (\lambda f::((real,\ 3)\ cart \Rightarrow bool) \times ((real,\ 3)\ cart \Rightarrow bool) \times (nat \Rightarrow nat) \times (real,\ 3)\ cart\ list \Rightarrow real.\ \forall (e'::(real,\ 3)\ cart \Rightarrow bool)\ (e''::(real,\ 3)\ cart \Rightarrow bool)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ GEQ\ (f\ (e',\ e'',\ p,\ vl))\ ((bump\ (hl\ [EL\ (0::nat)\ vl,\ EL\ (1::nat)\ vl]) - bump\ (hl\ [EL\ (2::nat)\ vl,\ EL\ (3::nat)\ vl]))\ /\ real\_of\_nat\ (4::nat))))))))\ (cell\_params\ V\ X)$

thm DEF_cluster_gamma_AX:

$cluster\_gamma\_AX = (\lambda(\_3434378::(real,\ 3)\ cart \Rightarrow bool)\ (\_3434379::(real,\ 3)\ cart \Rightarrow bool)\ \_3434380::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ sum\ \_3434380\ (\lambda X::(real,\ 3)\ cart \Rightarrow bool.\ gammaX\ \_3434378\ X\ lmfun * critical\_weight\ \_3434378\ X + beta\_bumpA\ \_3434378\ \_3434379\ X))$

thm Pack_defs.cluster_gamma_AX:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (e::(real,\ 3)\ cart \Rightarrow bool)\ Z::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ cluster\_gamma\_AX\ V\ e\ Z = sum\ Z\ (\lambda X::(real,\ 3)\ cart \Rightarrow bool.\ gammaX\ V\ X\ lmfun * critical\_weight\ V\ X + beta\_bumpA\ V\ e\ X)$

thm DEF_cell_cluster_estimate_A:

$cell\_cluster\_estimate\_A = (\lambda\_3434399::(real,\ 3)\ cart \Rightarrow bool.\ \forall e::(real,\ 3)\ cart \Rightarrow bool.\ (0::real) \le cluster\_gamma\_AX\ \_3434399\ e\ (cell\_cluster\ \_3434399\ e))$

thm Pack_defs.cell_cluster_estimate_A:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ cell\_cluster\_estimate\_A\ V = (\forall e::(real,\ 3)\ cart \Rightarrow bool.\ (0::real) \le cluster\_gamma\_AX\ V\ e\ (cell\_cluster\ V\ e))$

thm DEF_lmfun_inequality:

$lmfun\_inequality = (\lambda\_3434404::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \forall u::(real,\ ?'a::type)\ cart.\ IN\ u\ \_3434404 \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%847::(real,\ ?'a::type)\ cart.\ \exists v::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%847\ (IN\ v\ \_3434404 \wedge u \neq v \wedge distance\ (u,\ v) \le real\_of\_nat\ (2::nat) * h0)\ v))\ (\lambda v::(real,\ ?'a::type)\ cart.\ lmfun\ (hl\ [u,\ v])) \le real\_of\_nat\ (12::nat))$

thm Pack_defs.lmfun_inequality:

$\forall V::(real,\ ?'a::type)\ cart \Rightarrow bool.\ lmfun\_inequality\ V = (\forall u::(real,\ ?'a::type)\ cart.\ IN\ u\ V \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%847::(real,\ ?'a::type)\ cart.$

$\exists$ *v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%847 (IN v V* $\wedge$ *u* $\neq$ *v* $\wedge$ *distance (u, v)* $\leq$ *real_of_nat (2::nat)* $*$ *h0) v)) (λv::(real, ?'a::type) cart. lmfun (hl [u, v]))* $\leq$ *real_of_nat (12::nat))*

thm Pack_defs.ball_annulus:

ball_annulus = DIFF (cball (vec (0::nat), real_of_nat (2::nat) $*$ *h0)) (ball (vec (0::nat), real_of_nat (2::nat)))*

thm DEF_lmfun_ineq_center:

lmfun_ineq_center = (λ_3434409::(real, ?'a::type) cart $\Rightarrow$ *bool. sum _3434409 (λv::(real, ?'a::type) cart. lmfun (hl [vec (0::nat), v]))* $\leq$ *real_of_nat (12::nat))*

thm Pack_defs.lmfun_ineq_center:

$\forall$ *V::(real, ?'a::type) cart* $\Rightarrow$ *bool. lmfun_ineq_center V = (sum V (λv::(real, ?'a::type) cart. lmfun (hl [vec (0::nat), v]))* $\leq$ *real_of_nat (12::nat))*

thm DEF_fan_of_polyhedron:

fan_of_polyhedron = (λ_3434414::(real, 3) cart $\Rightarrow$ *bool. (GSPEC (λGEN%PVAR%848::(real, 3) cart.* $\exists$ *v::(real, 3) cart. SETSPEC GEN%PVAR%848 (extreme_point_of v _3434414) v), GSPEC (λGEN%PVAR%849::(real, 3) cart* $\Rightarrow$ *bool.* $\exists$ *(v::(real, 3) cart) w::(real, 3) cart. SETSPEC GEN%PVAR%849 (v* $\neq$ *w* $\wedge$ *face_of (hull convex (INSERT v (INSERT w EMPTY))) _3434414) (INSERT v (INSERT w EMPTY)))))*

thm Pack_defs.fan_of_polyhedron:

$\forall$ *s::(real, 3) cart* $\Rightarrow$ *bool. fan_of_polyhedron s = (GSPEC (λGEN%PVAR%848::(real, 3) cart.* $\exists$ *v::(real, 3) cart. SETSPEC GEN%PVAR%848 (extreme_point_of v s) v), GSPEC (λGEN%PVAR%849::(real, 3) cart* $\Rightarrow$ *bool.* $\exists$ *(v::(real, 3) cart) w::(real, 3) cart. SETSPEC GEN%PVAR%849 (v* $\neq$ *w* $\wedge$ *face_of (hull convex (INSERT v (INSERT w EMPTY))) s) (INSERT v (INSERT w EMPTY))))*

thm DEF_discrete:

discrete = (λ_3434475::(real, ?'a::type) cart $\Rightarrow$ *bool.* $\exists$ *e>0::real.* $\forall$ *(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x _3434475* $\wedge$ *IN y _3434475* $\wedge$ *distance (x, y) < e* $\longrightarrow$ *x = y)*

thm Packing3.discrete:

$\forall$ *S::(real, ?'a::type) cart* $\Rightarrow$ *bool. discrete S = (*$\exists$ *e>0::real.* $\forall$ *(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. IN x S* $\wedge$ *IN y S* $\wedge$ *distance (x, y) < e* $\longrightarrow$ *x = y)*

thm Packing3.IMAGE_LEMMA:

$\forall$ *(s::?'b::type* $\Rightarrow$ *bool) f::?'b::type* $\Rightarrow$ *?'a::type. IMAGE f s = GSPEC (λGEN%PVAR%865::?'a::type.* $\exists$ *x::?'b::type. SETSPEC GEN%PVAR%865 (IN x s) (f x))*

thm Packing3.SING_GSPEC_APP:

∀ (*f*::*?'b*::*type* ⇒ *?'a*::*type*) *a*::*?'b*::*type*. *GSPEC* (λ*GEN%PVAR%866*::*?'a*::*type*. ∃ *x*::*?'b*::*type*. *SETSPEC GEN%PVAR%866* (*x* = *a*) (*f x*)) = *INSERT* (*f a*) *EMPTY*

thm Packing3.SING_UNION_EQ_INSERT:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *x*::*?'a*::*type*. *HOL_Light_Import.UNION* (*INSERT x EMPTY*) *s* = *INSERT x s*

thm Packing3.IN_TRANS:

∀ (*x*::*?'a*::*type*) (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *IN x t* ∧ *SUBSET t s* ⟶ *IN x s*

thm Packing3.PROJECTION_ORTHOGONAL:

∀ (*d*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *dot* (*projection d v*) *d* = (*0*::*real*)

thm Packing3.LENGTH_IMP_CONS:

∀ *l*::*?'a*::*type list*. (*1*::*nat*) ≤ *length l* ⟶ (∃ (*h*::*?'a*::*type*) *t*::*?'a*::*type list*. *l* = *h* # *t*)

thm Packing3.LENGTH_1_LEMMA:

∀ *ul*::*?'a*::*type list*. *length ul* = (*1*::*nat*) ⟶ *ul* = [*hd ul*]

thm Packing3.PERMUTES_TRIVIAL:

∀ *p*::*nat* ⇒ *nat*. *permutes p* (*dotdot* (*0*::*nat*) (*0*::*nat*)) = (*p* = *id*)

thm Packing3.CONTAINS_BALL_AFFINE_HULL:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*x*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) < *r* ∧ *SUBSET* (*ball* (*x*, *r*)) *s* ⟶ *hull affine s* = *HOL_Light_Import.UNIV*

thm Packing3.CONV_UNION_lemma:

∀ (*A*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *B*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *hull convex* (*HOL_Light_Import.UNION A B*) = *hull convex* (*HOL_Light_Import.UNION A* (*hull convex B*))

thm Packing3.CONVEX_HULL_EQ_EQ_SET_EQ:

∀ (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ¬ *affine_dependent s* ∧ ¬ *affine_dependent t* ⟶ (*hull convex s* = *hull convex t*) = (*s* = *t*)

thm Packing3.HULL_INTER_SUBSET_INTER_HULL:

∀ (*P*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *SUBSET* (*hull P* (*HOL_Light_Import.INTER s t*)) (*HOL_Light_Import.INTER* (*hull P s*) (*hull P t*))

thm Packing3.HULL_INTER_EQ_INTER:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (s::?'a::type \Rightarrow bool)\ t::?'a::type \Rightarrow bool.\ P\ s \wedge P\ t \longrightarrow hull\ P\ (HOL\_Light\_Import.INTER\ s\ t) = HOL\_Light\_Import.INTER\ s\ t$

thm Packing3.SUBSET_INTERS:

$\forall (s::?'a::type \Rightarrow bool)\ f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ s\ (INTERS\ f) = (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ f \longrightarrow SUBSET\ s\ t)$

thm Packing3.HULL_INTERS_SUBSET_INTERS_HULL:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ (hull\ P\ (INTERS\ s))\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%869::?'a::type \Rightarrow bool.\ \exists t::?'a::type \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%869\ (IN\ t\ s)\ (hull\ P\ t))))$

thm Packing3.HULL_INTERS_EQ_INTERS:

$\forall (P::(?'a::type \Rightarrow bool) \Rightarrow bool)\ s::(?'a::type \Rightarrow bool) \Rightarrow bool.\ (\forall t::?'a::type \Rightarrow bool.\ IN\ t\ s \longrightarrow P\ t) \longrightarrow hull\ P\ (INTERS\ s) = INTERS\ s$

thm Packing3.INTERS_2_LEMMA:

$\forall (a::?'b::type)\ (b::?'b::type)\ f::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%871::?'a::type \Rightarrow bool.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%871\ (IN\ x\ (INSERT\ a\ (INSERT\ b\ EMPTY)))\ (f\ x))) = HOL\_Light\_Import.INTER\ (f\ a)\ (f\ b)$

thm Packing3.INTER_INTERS:

$\forall (s::?'b::type \Rightarrow bool)\ (f::?'a::type \Rightarrow bool)\ P::?'a::type \Rightarrow ?'b::type \Rightarrow bool.\ f \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER\ s\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%872::?'b::type \Rightarrow bool.\ \exists t::?'a::type.\ SETSPEC\ GEN\%PVAR\%872\ (IN\ t\ f)\ (P\ t)))) = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%873::?'b::type \Rightarrow bool.\ \exists t::?'a::type.\ SETSPEC\ GEN\%PVAR\%873\ (IN\ t\ f)\ (HOL\_Light\_Import.INTER\ s\ (P\ t))))$

thm Packing3.INTERS_UNIV:

$\forall f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ INTERS\ f = INTERS\ (DELETE\ f\ HOL\_Light\_Import.UNIV)$

thm Packing3.INTERS_INTER_INTERS:

$\forall (f::(?'a::type \Rightarrow bool) \Rightarrow bool)\ g::(?'a::type \Rightarrow bool) \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (INTERS\ f)\ (INTERS\ g) = INTERS\ (HOL\_Light\_Import.UNION\ f\ g)$

thm Packing3.INTERS_INTER_INTERS_ALT:

$\forall (f::?'b::type \Rightarrow bool)\ (g::?'b::type \Rightarrow bool)\ P::?'b::type \Rightarrow ?'a::type \Rightarrow bool.\ HOL\_Light\_Import.INTER\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%874::?'a::type \Rightarrow bool.\ \exists x::?'b::type.\ SETSPEC\ GEN\%PVAR\%874\ (IN\ x\ f)\ (P\ x))))\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%875::?'a::type \Rightarrow bool.\ \exists y::?'b::type.\ SETSPEC\ GEN\%PVAR\%875\ (IN\ y\ g)\ (P\ y)))) = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%876::?'a::type \Rightarrow bool.\ \exists u::?'b::type.\ SETSPEC\ GEN\%PVAR\%876\ (IN\ u\ (HOL\_Light\_Import.UNION\ f\ g))\ (P\ u)))$

thm Packing3.REAL_DIV_LE_1:

$\forall$ (*a*::*real*) *b*::*real*. (*0*::*real*) < *b* $\longrightarrow$ (*a* / *b* $\leq$ (*1*::*real*)) = (*a* $\leq$ *b*)

thm Packing3.UNIONS_FINITE_LEMMA:

$\forall$ (*g*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *P*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *?'a*::*type* $\Rightarrow$ *bool*. *FINITE g* $\wedge$ ($\forall$ *t*::*?'a*::*type* $\Rightarrow$ *bool*. *IN t g* $\longrightarrow$ *FINITE* (*P t*)) $\longrightarrow$ *FINITE* (*UNIONS* (*GSPEC* (λ*GEN%PVAR%879*::*?'a*::*type* $\Rightarrow$ *bool*. $\exists$ *t*::*?'a*::*type* $\Rightarrow$ *bool*. *SETSPEC GEN%PVAR%879* (*IN t g*) (*P t*))))

thm Packing3.REAL_FINITE_MIN_EXISTS:

$\forall$ *S*::*real* $\Rightarrow$ *bool*. *FINITE S* $\wedge$ *S* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *m*::*real*. *IN m S* $\wedge$ ($\forall$ *x*::*real*. *IN x S* $\longrightarrow$ *m* $\leq$ *x*))

thm Packing3.REAL_FINITE_MAX_EXISTS:

$\forall$ *S*::*real* $\Rightarrow$ *bool*. *FINITE S* $\wedge$ *S* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *m*::*real*. *IN m S* $\wedge$ ($\forall$ *x*::*real*. *IN x S* $\longrightarrow$ *x* $\leq$ *m*))

thm Packing3.REAL_FINITE_ARGMIN:

$\forall$ (*f*::*?'a*::*type* $\Rightarrow$ *real*) *S*::*?'a*::*type* $\Rightarrow$ *bool*. *FINITE S* $\wedge$ *S* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *a*::*?'a*::*type*. *IN a S* $\wedge$ ($\forall$ *x*::*?'a*::*type*. *IN x S* $\longrightarrow$ *f a* $\leq$ *f x*))

thm Packing3.REAL_FINITE_ARGMAX:

$\forall$ (*f*::*?'a*::*type* $\Rightarrow$ *real*) *S*::*?'a*::*type* $\Rightarrow$ *bool*. *FINITE S* $\wedge$ *S* $\neq$ *EMPTY* $\longrightarrow$ ($\exists$ *a*::*?'a*::*type*. *IN a S* $\wedge$ ($\forall$ *x*::*?'a*::*type*. *IN x S* $\longrightarrow$ *f x* $\leq$ *f a*))

thm Packing3.BIS_EQ_HYPERPLANE:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *bis u v* = *GSPEC* (λ*GEN%PVAR%880*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%880* (*dot* (% (*real_of_nat* (*2*::*nat*)) (*vector_sub v u*)) *x* = *dot v v* − *dot u u*) *x*)

thm Packing3.BIS_LE_EQ_HALFSPACE:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *bis_le u v* = *GSPEC* (λ*GEN%PVAR%881*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%881* (*dot* (% (*real_of_nat* (*2*::*nat*)) (*vector_sub v u*)) *x* $\leq$ *dot v v* − *dot u u*) *x*)

thm Packing3.CONVEX_BIS_LE:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *convex* (*bis_le u v*)

thm Packing3.CLOSED_BIS_LE:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *HOL_Light_Import.closed* (*bis_le u v*)

thm Packing3.CONVEX_BIS:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *convex* (*bis u v*)

thm Packing3.POLYHEDRON_BIS:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *polyhedron* (*bis u v*)

thm Packing3.AFFINE_BIS:

$\forall\,(a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ affine\ (bis\ a\ b)$

thm Packing3.AFFINE_HULL_INTERS_BIS:

$\forall\,(p::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ hull\ affine\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%882::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\,u::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%882\ (IN\ u\ s)\ (bis\ p\ u)))) = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%883::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \exists\,u::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%883\ (IN\ u\ s)\ (bis\ p\ u)))$

thm Packing3.MID_POINT_EXISTS:

$\forall\,(v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ d::real.\ (0::real) \leq d \wedge d \leq distance\ (v,\ w) \longrightarrow (\exists\,x::(real,\ ?'a::type)\ cart.\ between\ x\ (v,\ w) \wedge distance\ (v,\ x) = d)$

thm Packing3.CLOSED_DISCRETE:

$\forall\,A::(real,\ ?'a::type)\ cart \Rightarrow bool.\ discrete\ A \longrightarrow HOL\_Light\_Import.closed\ A$

thm Packing3.DISCRETE_SUBSET:

$\forall\,(A::(real,\ ?'a::type)\ cart \Rightarrow bool)\ B::(real,\ ?'a::type)\ cart \Rightarrow bool.\ discrete\ A \wedge SUBSET\ B\ A \longrightarrow discrete\ B$

thm Packing3.DISCRETE_IMP_BOUNDED_EQ_COMPACT:

$\forall\,S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ discrete\ S \longrightarrow bounded\ S = compact\ S$

thm Packing3.DISCRETE_OPEN_COVER:

$\forall\,S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ discrete\ S \longrightarrow (\exists\,f::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ (\forall\,b::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ b\ f \longrightarrow HOL\_Light\_Import.open\ b \wedge HAS\_SIZE\ (HOL\_Light\_Import.INTER\ b\ S)\ (1::nat)) \wedge SUBSET\ S\ (UNIONS\ f))$

thm Packing3.DISCRETE_BOUNDED_IMP_FINITE:

$\forall\,S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ discrete\ S \wedge bounded\ S \longrightarrow FINITE\ S$

thm Packing3.PACKING_IMP_DISCRETE:

$\forall\,V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \longrightarrow discrete\ V$

thm Upfzbzm_support_lemmas.FINITE_PACK_LEMMA:

$\forall\,(p::(real,\ 3)\ cart)\ (r::real)\ V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \longrightarrow FINITE\ (HOL\_Light\_Import.INTER\ V\ (ball\ (p,\ r)))$

thm Packing3.TIWWFYQ:

$\forall\,(V::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart.\ packing\ V \wedge saturated\ V \longrightarrow (\exists\,v::(real,\ 3)\ cart.\ IN\ v\ V \wedge IN\ p\ (voronoi\_closed\ V\ v))$

thm Packing3.CENTER_IN_VORONOI_CELL:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. IN v ($voronoi\_closed$ V v) $\wedge$ IN v ($voronoi\_open$ V v)

thm Packing3.VORONOI_CLOSED_CONTAINS_BALL:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. $packing$ V $\longrightarrow$ ($\exists$ r>0::$real$. $SUBSET$ ($ball$ (v, r)) ($voronoi\_closed$ V v))

thm Packing3.AFF_DIM_VORONOI_CLOSED:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. $packing$ V $\longrightarrow$ aff_dim ($voronoi\_closed$ V v) = int (3::nat)

thm Packing3.VORONOI_BALL2:

$\forall$ (V::($real$, ?'a::$type$) $cart$ $\Rightarrow$ $bool$) v::($real$, ?'a::$type$) $cart$. $saturated$ V $\longrightarrow$ $SUBSET$ ($voronoi\_closed$ V v) ($ball$ (v, $real\_of\_nat$ (2::nat)))

thm Packing3.BOUNDED_VORONOI_CLOSED:

$\forall$ (V::($real$, ?'a::$type$) $cart$ $\Rightarrow$ $bool$) v::($real$, ?'a::$type$) $cart$. $saturated$ V $\longrightarrow$ $bounded$ ($voronoi\_closed$ V v)

thm Packing3.VORONOI_CLOSED_EQ_INTERS_BIS_LE:

$\forall$ (S::($real$, ?'a::$type$) $cart$ $\Rightarrow$ $bool$) v::($real$, ?'a::$type$) $cart$. $voronoi\_closed$ S v = $INTERS$ ($GSPEC$ ($\lambda$$GEN\%PVAR\%885$::($real$, ?'$a$::$type$) $cart$ $\Rightarrow$ $bool$. $\exists$ w::($real$, ?'a::$type$) $cart$. $SETSPEC$ $GEN\%PVAR\%885$ (IN w S) (bis_le v w)))

thm Packing3.VORONOI_CLOSED_EQ_INTERS_BIS_LE_ALT:

$\forall$ (S::($real$, ?'a::$type$) $cart$ $\Rightarrow$ $bool$) v::($real$, ?'a::$type$) $cart$. $voronoi\_closed$ S v = $INTERS$ ($GSPEC$ ($\lambda$$GEN\%PVAR\%886$::($real$, ?'$a$::$type$) $cart$ $\Rightarrow$ $bool$. $\exists$ w::($real$, ?'a::$type$) $cart$. $SETSPEC$ $GEN\%PVAR\%886$ (IN w S $\wedge$ w $\neq$ v) (bis_le v w)))

thm Packing3.VORONOI_INTER_BIS_LE:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. $packing$ V $\wedge$ $saturated$ V $\wedge$ IN v V $\longrightarrow$ $voronoi\_closed$ V v = $INTERS$ ($GSPEC$ ($\lambda$$GEN\%PVAR\%887$::($real$, 3) $cart$ $\Rightarrow$ $bool$. $\exists$ u::($real$, 3) $cart$. $SETSPEC$ $GEN\%PVAR\%887$ (IN u V $\wedge$ IN u ($ball$ (v, $real\_of\_nat$ (4::nat))) $\wedge$ u $\neq$ v) (bis_le v u)))

thm Packing3.VORONOI_CLOSED_EQ_FINITE_INTERS_BIS_LE:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. $packing$ V $\wedge$ $saturated$ V $\wedge$ IN v V $\longrightarrow$ ($\exists$ W::($real$, 3) $cart$ $\Rightarrow$ $bool$. $SUBSET$ W V $\wedge$ $\neg$ IN v W $\wedge$ $FINITE$ W $\wedge$ $voronoi\_closed$ V v = $INTERS$ ($GSPEC$ ($\lambda$$GEN\%PVAR\%888$::($real$, 3) $cart$ $\Rightarrow$ $bool$. $\exists$ u::($real$, 3) $cart$. $SETSPEC$ $GEN\%PVAR\%888$ (IN u W) (bis_le v u))))

thm Packing3.VORONOI_POLYHEDRON:

$\forall$ (V::($real$, 3) $cart$ $\Rightarrow$ $bool$) v::($real$, 3) $cart$. $packing$ V $\wedge$ $saturated$ V $\wedge$ IN v V $\longrightarrow$ $polyhedron$ ($voronoi\_closed$ V v)

thm Packing3.CONVEX_VORONOI_CLOSED:

$\forall (S::(real, ?'a::type) cart \Rightarrow bool)\ v::(real, ?'a::type) cart.\ convex\ (voronoi\_closed\ S\ v)$

thm Packing3.CLOSED_VORONOI_CLOSED:

$\forall (S::(real, ?'a::type) cart \Rightarrow bool)\ v::(real, ?'a::type) cart.\ HOL\_Light\_Import.closed\ (voronoi\_closed\ S\ v)$

thm Packing3.COMPACT_VORONOI_CLOSED:

$\forall (S::(real, ?'a::type) cart \Rightarrow bool)\ v::(real, ?'a::type) cart.\ saturated\ S \longrightarrow compact\ (voronoi\_closed\ S\ v)$

thm Packing3.DRUQUFE:

$\forall (V::(real, 3) cart \Rightarrow bool)\ v::(real, 3) cart.\ packing\ V \wedge saturated\ V \longrightarrow compact\ (voronoi\_closed\ V\ v) \wedge convex\ (voronoi\_closed\ V\ v) \wedge measurable\ (voronoi\_closed\ V\ v)$

thm Packing3.INITIAL_SUBLIST_APPEND:

$\forall (ul::?'a::type\ list)\ vl::?'a::type\ list.\ initial\_sublist\ ul\ (ul\ @\ vl)$

thm Packing3.INITIAL_SUBLIST_HEAD_EQ:

$\forall (xl::?'a::type\ list)\ (zl::?'a::type\ list)\ (hx::?'a::type)\ (tx::?'a::type\ list)\ (hz::?'a::type)\ tz::?'a::type\ list.\ xl = hx\ \#\ tx \wedge zl = hz\ \#\ tz \wedge initial\_sublist\ xl\ zl \longrightarrow hx = hz$

thm Packing3.INITIAL_SUBLIST_HEAD_EQ_2:

$\forall (xl::?'a::type\ list)\ (yl::?'a::type\ list)\ (zl::?'a::type\ list)\ (hx::?'a::type)\ (tx::?'a::type\ list)\ (hy::?'a::type)\ ty::?'a::type\ list.\ xl = hx\ \#\ tx \wedge yl = hy\ \#\ ty \wedge initial\_sublist\ xl\ zl \wedge initial\_sublist\ yl\ zl \longrightarrow hx = hy$

thm Packing3.INITIAL_SUBLIST_TAIL:

$\forall (xl::?'a::type\ list)\ (zl::?'a::type\ list)\ (hx::?'a::type)\ tx::?'a::type\ list.\ xl = hx\ \#\ tx \wedge initial\_sublist\ xl\ zl \longrightarrow initial\_sublist\ tx\ (tl\ zl)$

thm Packing3.INITIAL_SUBLIST_UNIQUE:

$\forall (n::nat)\ (xl::?'a::type\ list)\ (yl::?'a::type\ list)\ zl::?'a::type\ list.\ initial\_sublist\ xl\ zl \wedge initial\_sublist\ yl\ zl \wedge length\ xl = n \wedge length\ yl = n \longrightarrow xl = yl$

thm Packing3.INITIAL_SUBLIST_TRANS:

$\forall (xl::?'a::type\ list)\ (yl::?'a::type\ list)\ zl::?'a::type\ list.\ initial\_sublist\ xl\ yl \wedge initial\_sublist\ yl\ zl \longrightarrow initial\_sublist\ xl\ zl$

thm Packing3.INITIAL_SUBLIST_REFL:

$\forall ul::?'a::type\ list.\ initial\_sublist\ ul\ ul$

thm Packing3.INITIAL_SUBLIST_NIL:

$\forall$ *zl::?'a::type list. initial_sublist* [] *zl*

thm Packing3.INITIAL_SUBLIST_EXISTS_ALT:

$\forall$ (*n::nat*) (*zl::?'a::type list*) *k::nat. length zl* = *n* $\wedge$ *k* $\leq$ *n* $\longrightarrow$ ($\exists$ *xl::?'a::type list. initial_sublist xl zl* $\wedge$ *length xl* = *k*)

thm Packing3.INITIAL_SUBLIST_EXISTS:

$\forall$ (*zl::?'a::type list*) *k::nat. k* $\leq$ *length zl* $\longrightarrow$ ($\exists$ *xl::?'a::type list. initial_sublist xl zl* $\wedge$ *length xl* = *k*)

thm Packing3.INITIAL_SUBLIST_LENGTH_LE:

$\forall$ (*xl::?'a::type list*) *zl::?'a::type list. initial_sublist xl zl* $\longrightarrow$ *length xl* $\leq$ *length zl*

thm Packing3.INITIAL_SUBLIST_APPEND_2:

$\forall$ (*xl::?'a::type list*) (*ul::?'a::type list*) *vl::?'a::type list. initial_sublist xl* (*ul* @ *vl*) = (*initial_sublist xl ul* $\vee$ ($\exists$ *yl::?'a::type list. initial_sublist yl vl* $\wedge$ *xl* = *ul* @ *yl*))

thm Packing3.INITIAL_SUBLIST_SING:

$\forall$ (*v::?'a::type*) *xl::?'a::type list. initial_sublist xl* [*v*] = (*xl* = [] $\vee$ *xl* = [*v*])

thm Packing3.INITIAL_SUBLIST_APPEND_SING:

$\forall$ (*xl::?'a::type list*) (*ul::?'a::type list*) *v::?'a::type. initial_sublist xl* (*ul* @ [*v*]) = (*initial_sublist xl ul* $\vee$ *xl* = *ul* @ [*v*])

thm Packing3.INITIAL_SUBLIST_HD:

$\forall$ *ul::?'a::type list.* (*1::nat*) $\leq$ *length ul* $\longrightarrow$ *initial_sublist* [*hd ul*] *ul*

thm Packing3.BUTLAST_INITIAL_SUBLIST:

$\forall$ *ul::?'a::type list.* (*1::nat*) $\leq$ *length ul* $\longrightarrow$ *initial_sublist* (*butlast ul*) *ul*

thm Packing3.LENGTH_BUTLAST:

$\forall$ *ul::?'a::type list.* (*1::nat*) $\leq$ *length ul* $\longrightarrow$ *length* (*butlast ul*) = *length ul* − (*1::nat*)

thm Packing3.HD_IN_SET_OF_LIST:

$\forall$ *ul::?'a::type list.* (*1::nat*) $\leq$ *length ul* $\longrightarrow$ *IN* (*hd ul*) (*set_of_list ul*)

thm Packing3.HD_INITIAL_SUBLIST:

$\forall$ (*xl::?'a::type list*) *yl::?'a::type list.* (*1::nat*) $\leq$ *length yl* $\wedge$ *initial_sublist yl xl* $\longrightarrow$ *hd yl* = *hd xl*

thm Packing3.SET_OF_LIST_INITIAL_SUBLIST_SUBSET:

$\forall$ (*vl::?'a::type list*) *ul::?'a::type list. initial_sublist vl ul* $\longrightarrow$ *SUBSET* (*set_of_list vl*) (*set_of_list ul*)

thm Packing3.LENGTH_REVERSE:

$\forall\, ul::?'a::type\ list.\ length\ (rev\ ul) = length\ ul$

thm Packing3.EL_REVERSE:

$\forall\, (ul::?'a::type\ list)\ i::nat.\ i < length\ ul \longrightarrow EL\ i\ (rev\ ul) = EL\ (length\ ul - (1::nat) - i)\ ul$

thm Pack_defs.REVERSE_TABLE_conjunct1:

$REVERSE\_TABLE\ (?f::nat \Rightarrow ?'a::type)\ (Suc\ (?i::nat)) = ?f\ ?i\ \#\ REVERSE\_TABLE\ ?f\ ?i$

thm Pack_defs.REVERSE_TABLE_conjunct0:

$REVERSE\_TABLE\ (?f::nat \Rightarrow ?'a::type)\ (0::nat) = []$

thm Packing3.LENGTH_TABLE:

$\forall\, (f::nat \Rightarrow ?'a::type)\ n::nat.\ length\ (TABLE\ f\ n) = n$

thm Packing3.EL_TABLE:

$\forall\, (f::nat \Rightarrow ?'a::type)\ (n::nat)\ i::nat.\ i < n \longrightarrow EL\ i\ (TABLE\ f\ n) = f\ i$

thm Packing3.LENGTH_LEFT_ACTION_LIST:

$\forall\, (ul::?'a::type\ list)\ p::nat \Rightarrow nat.\ length\ (left\_action\_list\ p\ ul) = length\ ul$

thm Packing3.EL_LEFT_ACTION_LIST:

$\forall\, (ul::?'a::type\ list)\ (p::nat \Rightarrow nat)\ i::nat.\ permutes\ p\ (dotdot\ (0::nat)\ (length\ ul - (1::nat))) \wedge i < length\ ul \longrightarrow EL\ i\ ul = EL\ (p\ i)\ (left\_action\_list\ p\ ul)$

thm Packing3.MEM_LEFT_ACTION_LIST:

$\forall\, (ul::?'a::type\ list)\ (p::nat \Rightarrow nat)\ x::?'a::type.\ permutes\ p\ (dotdot\ (0::nat)\ (length\ ul - (1::nat))) \longrightarrow MEM\ x\ (left\_action\_list\ p\ ul) = MEM\ x\ ul$

thm Packing3.SET_OF_LIST_LEFT_ACTION_LIST:

$\forall\, (ul::?'a::type\ list)\ p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (0::nat)\ (length\ ul - (1::nat))) \longrightarrow set\_of\_list\ (left\_action\_list\ p\ ul) = set\_of\_list\ ul$

thm Packing3.CARD_SET_OF_LIST_EQ_LENGTH_IMP_ALL_DISTINCT:

$\forall\, ul::?'a::type\ list.\ CARD\ (set\_of\_list\ ul) = length\ ul \longrightarrow (\forall\, (i::nat)\ j::nat.\ i < length\ ul \wedge j < length\ ul \wedge i \neq j \longrightarrow EL\ i\ ul \neq EL\ j\ ul)$

thm Pack_defs.DROP_conjunct1:

$DROP\ (?ul::?'a::type\ list)\ (Suc\ (?i::nat)) = hd\ ?ul\ \#\ DROP\ (tl\ ?ul)\ ?i$

thm Pack_defs.DROP_conjunct0:

$DROP\ (?ul::?'a::type\ list)\ (0::nat) = tl\ ?ul$

thm Packing3.LENGTH_DROP:

$\forall (i::nat)\ ul::?'a::type\ list.\ i < length\ ul \longrightarrow length\ (DROP\ ul\ i) = length\ ul - (1::nat)$

thm Packing3.EL_DROP:

$\forall (i::nat)\ (j::nat)\ ul::?'a::type\ list.\ i < length\ ul \wedge j < length\ ul - (1::nat) \longrightarrow EL\ j\ (DROP\ ul\ i) = (if\ j < i\ then\ EL\ j\ ul\ else\ EL\ (j + (1::nat))\ ul)$

thm Packing3.LIST_EL_EQ:

$\forall (ul::?'a::type\ list)\ vl::?'a::type\ list.\ (ul = vl) = (length\ ul = length\ vl \wedge (\forall j<length\ ul.\ EL\ j\ ul = EL\ j\ vl))$

thm Packing3.LEFT_ACTION_LIST_I:

$\forall ul::?'a::type\ list.\ left\_action\_list\ id\ ul = ul$

thm Packing3.SET_OF_LIST_DELETE_SUBSET_DROP:

$\forall (j::nat)\ ul::?'a::type\ list.\ j < length\ ul \longrightarrow SUBSET\ (DELETE\ (set\_of\_list\ ul)\ (EL\ j\ ul))\ (set\_of\_list\ (DROP\ ul\ j))$

thm Packing3.SET_OF_LIST_DELETE_EQ_DROP:

$\forall (j::nat)\ ul::?'a::type\ list.\ CARD\ (set\_of\_list\ ul) = length\ ul \wedge j < length\ ul \longrightarrow DELETE\ (set\_of\_list\ ul)\ (EL\ j\ ul) = set\_of\_list\ (DROP\ ul\ j)$

thm Packing3.BARV_SUBSET:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (k::nat)\ ul::(real,\ 3)\ cart\ list.\ barV\ V\ k\ ul \longrightarrow SUBSET\ (set\_of\_list\ ul)\ V$

thm Packing3.BARV_CONS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (k::nat)\ ul::(real,\ 3)\ cart\ list.\ barV\ V\ k\ ul \longrightarrow (\exists (h::(real,\ 3)\ cart)\ t::(real,\ 3)\ cart\ list.\ ul = h\ \#\ t \wedge h = hd\ ul)$

thm Packing3.BARV_INITIAL_SUBLIST:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (k::nat)\ (ul::(real,\ 3)\ cart\ list)\ vl::(real,\ 3)\ cart\ list.\ barV\ V\ k\ ul \wedge initial\_sublist\ vl\ ul \wedge (0::nat) < length\ vl \longrightarrow barV\ V\ (length\ vl - (1::nat))\ vl$

thm Packing3.BARV_0:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ packing\ V \wedge IN\ v\ V \longrightarrow barV\ V\ (0::nat)\ [v]$

thm Packing3.BARV_IMP_K_LE_3:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ barV\ V\ k\ ul \longrightarrow k \le (3::nat)$

thm Packing3.BARV_IMP_HD_IN_SET_OF_LIST:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (k::nat)\ ul::(real,\ 3)\ cart\ list.\ barV\ V\ k\ ul \longrightarrow IN\ (hd\ ul)\ (set\_of\_list\ ul)$

thm Packing3.TRUNCATE_SIMPLEX_INITIAL_SUBLIST:

$\forall (k::nat)\ (xl::?'a::type\ list)\ zl::?'a::type\ list.\ (truncate\_simplex\ k\ zl = xl \wedge k + (1::nat) \leq length\ zl) = (initial\_sublist\ xl\ zl \wedge length\ xl = k + (1::nat))$

thm Packing3.TRUNCATE_SIMPLEX_BARV:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (r::nat)\ (k::nat)\ zl::(real,\ 3)\ cart\ list.\ barV\ V\ k\ zl \wedge r \leq k \longrightarrow barV\ V\ r\ (truncate\_simplex\ r\ zl)$

thm Packing3.TRUNCATE_SIMPLEX_REFL:

$\forall (k::nat)\ ul::?'a::type\ list.\ length\ ul = k + (1::nat) \longrightarrow truncate\_simplex\ k\ ul = ul$

thm Packing3.TRUNCATE_0_EQ_HEAD:

$\forall ul::?'a::type\ list.\ (1::nat) \leq length\ ul \longrightarrow truncate\_simplex\ (0::nat)\ ul = [hd\ ul]$

thm Packing3.LENGTH_TRUNCATE_SIMPLEX:

$\forall (k::nat)\ ul::?'a::type\ list.\ k + (1::nat) \leq length\ ul \longrightarrow length\ (truncate\_simplex\ k\ ul) = k + (1::nat)$

thm Packing3.TRUNCATE_SIMPLEX_EQ_BUTLAST:

$\forall ul::?'a::type\ list.\ (2::nat) \leq length\ ul \longrightarrow truncate\_simplex\ (length\ ul - (2::nat))\ ul = butlast\ ul$

thm Packing3.HD_TRUNCATE_SIMPLEX:

$\forall (ul::?'a::type\ list)\ j::nat.\ j + (1::nat) \leq length\ ul \longrightarrow hd\ (truncate\_simplex\ j\ ul) = hd\ ul$

thm Packing3.TRUNCATE_TRUNCATE_SIMPLEX:

$\forall (ul::?'a::type\ list)\ (i::nat)\ j::nat.\ i \leq j \wedge j + (1::nat) \leq length\ ul \longrightarrow truncate\_simplex\ i\ (truncate\_simplex\ j\ ul) = truncate\_simplex\ i\ ul$

thm Packing3.INITIAL_SUBLIST_IMP_TRUNCATE_SIMPLEX:

$\forall (xl::?'a::type\ list)\ yl::?'a::type\ list.\ initial\_sublist\ yl\ xl \wedge (1::nat) \leq length\ yl \longrightarrow yl = truncate\_simplex\ (length\ yl - (1::nat))\ xl \wedge length\ yl \leq length\ xl$

thm Packing3.LIST_EQ_TRUNCATE_SIMPLEX_APPEND_LAST:

$\forall ul::?'a::type\ list.\ (2::nat) \leq length\ ul \longrightarrow ul = truncate\_simplex\ (length\ ul - (2::nat))\ ul\ @\ [last\ ul]$

thm Packing3.TRUNCATE_SIMPLEX_ADD1:

$\forall (ul::?'a::type\ list)\ k::nat.\ k + (2::nat) \leq length\ ul \longrightarrow truncate\_simplex\ (k + (1::nat))\ ul = truncate\_simplex\ k\ ul\ @\ [last\ (truncate\_simplex\ (k + (1::nat))\ ul)]$

thm Packing3.EL_TRUNCATE_SIMPLEX:

$\forall$ (*ul*::*?'a*::*type list*) (*k*::*nat*) *j*::*nat*. *k* + (*1*::*nat*) $\leq$ *length ul* $\wedge$ *j* $\leq$ *k* $\longrightarrow$ *EL j* (*truncate_simplex k ul*) = *EL j ul*

thm Packing3.TRUNCATE_SIMPLEX_ADD1_ALT:

$\forall$ (*ul*::*?'a*::*type list*) *k*::*nat*. *k* + (*2*::*nat*) $\leq$ *length ul* $\longrightarrow$ *truncate_simplex* (*k* + (*1*::*nat*)) *ul* = *truncate_simplex k ul* @ [*EL* (*k* + (*1*::*nat*)) *ul*]

thm Packing3.VORONOI_SET_SING:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *3*) *cart*. *voronoi_set V* (*INSERT u EMPTY*) = *voronoi_closed V u*

thm Packing3.VORONOI_LIST_SING:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *u*::(*real*, *3*) *cart*. *voronoi_list V* [*u*] = *voronoi_closed V u*

thm Packing3.VORONOI_SET_2:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *voronoi_set V* (*INSERT u* (*INSERT v EMPTY*)) = *HOL_Light_Import.INTER* (*voronoi_closed V v*) (*voronoi_closed V u*)

thm Packing3.VORONOI_SET_2_BIS:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *IN u V* $\wedge$ *IN v V* $\longrightarrow$ *voronoi_set V* (*INSERT u* (*INSERT v EMPTY*)) = *HOL_Light_Import.INTER* (*voronoi_closed V v*) (*bis u v*)

thm Packing3.VORONOI_SET_2_BIS_LE:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *IN u V* $\wedge$ *IN v V* $\longrightarrow$ *voronoi_set V* (*INSERT u* (*INSERT v EMPTY*)) = *HOL_Light_Import.INTER* (*voronoi_closed V v*) (*bis_le u v*)

thm Packing3.VORONOI_LIST_BIS:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*ul*::(*real*, *3*) *cart list*) (*h*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart list*. *SUBSET* (*set_of_list ul*) *V* $\wedge$ *ul* = *h* # *t* $\longrightarrow$ *voronoi_list V ul* = *HOL_Light_Import.INTER* (*voronoi_closed V h*) (*INTERS* (*GSPEC* (λ*GEN%PVAR%890*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ *u*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%890* (*IN u* (*set_of_list t*)) (*bis h u*))))

thm Packing3.VORONOI_INTER_BIS_EQ_INTER_BIS_LE:

$\forall$ (*V*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*v*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. *IN v V* $\wedge$ *IN u V* $\longrightarrow$ *HOL_Light_Import.INTER* (*voronoi_closed V v*) (*bis u v*) = *HOL_Light_Import.INTER* (*voronoi_closed V v*) (*bis_le u v*)

thm Packing3.LIST_SUBSET:

$\forall$ (*V*::*?'a*::*type* $\Rightarrow$ *bool*) (*ul*::*?'a*::*type list*) (*h*::*?'a*::*type*) *t*::*?'a*::*type list*. *SUBSET* (*set_of_list ul*) *V* $\wedge$ *ul* = *h* # *t* $\longrightarrow$ *IN h V* $\wedge$ ($\forall$ *u*::*?'a*::*type*. *IN u* (*set_of_list t*) $\longrightarrow$ *IN u V*)

thm Packing3.VORONOI_LIST_BIS_LE:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*h*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart list*. *SUBSET* (*set_of_list ul*) *V* ∧ *ul* = *h* # *t* ⟶ *voronoi_list V ul* = *HOL_Light_Import.INTER* (*voronoi_closed V h*) (*INTERS* (*GSPEC* (λ*GEN%PVAR%891*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *u*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%891* (*IN u* (*set_of_list t*)) (*bis_le u h*))))

thm Packing3.BOUNDED_VORONOI_LIST:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *barV V k ul* ⟶ *bounded* (*voronoi_list V ul*)

thm Packing3.VORONOI_LIST_INTER_BIS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*v*::(*real*, *3*) *cart*) (*h*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart list*. *SUBSET* (*set_of_list ul*) *V* ∧ *IN v V* ∧ *ul* = *h* # *t* ⟶ *HOL_Light_Import.INTER* (*voronoi_list V ul*) (*bis h v*) = *voronoi_list V* (*ul* @ [*v*])

thm Packing3.SUPSET_INTER:

∀ (*s*::*?'a*::*type* ⇒ *bool*) (*t*::*?'a*::*type* ⇒ *bool*) *u*::*?'a*::*type* ⇒ *bool*. *SUBSET s t* ∧ *s* = *u* ⟶ *s* = *HOL_Light_Import.INTER t u*

thm Packing3.INTER_AFFINE_HULL:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *s* = *HOL_Light_Import.INTER* (*hull affine s*) *s*

thm Packing3.VORONOI_LIST_ALMOST_CANONICAL_0:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*h*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart list*. *packing V* ∧ *saturated V* ∧ *SUBSET* (*set_of_list ul*) *V* ∧ *ul* = *h* # *t* ⟶ (∃ *K*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE K* ∧ *voronoi_list V ul* = *INTERS K* ∧ (∀ *a*::(*real*, *3*) *cart* ⇒ *bool*. *IN a K* ⟶ (∃ *v*::(*real*, *3*) *cart*. *IN v V* ∧ (*a* = *bis_le v h* ∨ *a* = *bis_le h v*))))

thm Packing3.VORONOI_LIST_ALMOST_CANONICAL:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*h*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart list*. *packing V* ∧ *saturated V* ∧ *SUBSET* (*set_of_list ul*) *V* ∧ *ul* = *h* # *t* ⟶ (∃ *K*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FINITE K* ∧ *voronoi_list V ul* = *HOL_Light_Import.INTER* (*hull affine* (*voronoi_list V ul*)) (*INTERS K*) ∧ (∀ *a*::(*real*, *3*) *cart* ⇒ *bool*. *IN a K* ⟶ (∃ *v*::(*real*, *3*) *cart*. *IN v V* ∧ *v* ≠ *h* ∧ (*a* = *bis_le v h* ∨ *a* = *bis_le h v*))))

thm Packing3.lemma1:

∀ (*f*::*?'a*::*type* ⇒ *bool*) *P*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*. *FINITE f* ∧ *P f* ⟶ (∃ (*n*::*nat*) *g*::*?'a*::*type* ⇒ *bool*. *SUBSET g f* ∧ *HAS_SIZE g n* ∧ *P g*)

thm Packing3.MINIMAL_INTERS_EXISTS:

$\forall (s::?'a::type \Rightarrow bool)\ f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = INTERS\ f \longrightarrow (\exists g::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ g\ f \wedge s = INTERS\ g \wedge (\forall g'::(?'a::type \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ g'\ g \longrightarrow PSUBSET\ s\ (INTERS\ g')))$

thm Packing3.MINIMAL_INTER_INTERS_EXISTS:

$\forall (s::?'a::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::(?'a::type \Rightarrow bool) \Rightarrow bool.\ FINITE\ f \wedge s = HOL\_Light\_Import.INTER\ t\ (INTERS\ f) \longrightarrow (\exists g::(?'a::type \Rightarrow bool) \Rightarrow bool.\ SUBSET\ g\ f \wedge s = HOL\_Light\_Import.INTER\ t\ (INTERS\ g) \wedge (\forall g'::(?'a::type \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ g'\ g \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ t\ (INTERS\ g'))))$

thm Packing3.VORONOI_LIST_CANONICAL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (h::(real, 3)\ cart)\ t::(real, 3)\ cart\ list.\ packing\ V \wedge saturated\ V \wedge SUBSET\ (set\_of\_list\ ul)\ V \wedge ul = h\ \#\ t \longrightarrow (\exists K::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ K \wedge voronoi\_list\ V\ ul = HOL\_Light\_Import.INTER\ (hull\ affine\ (voronoi\_list\ V\ ul))\ (INTERS\ K) \wedge (\forall a::(real, 3)\ cart \Rightarrow bool.\ IN\ a\ K \longrightarrow (\exists v::(real, 3)\ cart.\ IN\ v\ V \wedge v \neq h \wedge (a = bis\_le\ v\ h \vee a = bis\_le\ h\ v))) \wedge (\forall K'::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ K'\ K \longrightarrow PSUBSET\ (voronoi\_list\ V\ ul)\ (HOL\_Light\_Import.INTER\ (hull\ affine\ (voronoi\_list\ V\ ul))\ (INTERS\ K'))))$

thm Packing3.POLYHEDRON_VORONOI_LIST:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ packing\ V \wedge saturated\ V \wedge SUBSET\ (set\_of\_list\ ul)\ V \longrightarrow polyhedron\ (voronoi\_list\ V\ ul)$

thm Packing3.POLYTOPE_VORONOI_LIST:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ packing\ V \wedge saturated\ V \wedge SUBSET\ (set\_of\_list\ ul)\ V \wedge ul \neq [] \longrightarrow polytope\ (voronoi\_list\ V\ ul)$

thm Packing3.POLYTOPE_VORONOI_LIST_BARV:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge saturated\ V \wedge barV\ V\ k\ ul \longrightarrow polytope\ (voronoi\_list\ V\ ul)$

thm Packing3.CLOSED_VORONOI_LIST:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ HOL\_Light\_Import.closed\ (voronoi\_list\ V\ ul)$

thm Packing3.CONVEX_VORONOI_LIST:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ convex\ (voronoi\_list\ V\ ul)$

thm Packing3.AFF_DIM_VORONOI_LIST:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ barV\ V\ k\ ul \longrightarrow aff\_dim\ (voronoi\_list\ V\ ul) = int\ (3::nat) - int\ k$

thm Packing3.VORONOI_LIST_SUBSET_VORONOI_CLOSED:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) *vl*::(*real*, *3*) *cart list*. (*1*::*nat*) ≤ *length vl* ⟶ *SUBSET* (*voronoi_list V vl*) (*voronoi_closed V* (*hd vl*))

thm Sphere.OMEGA_LIST_N_conjunct1:

omega_list_n (*?V*::(*real*, *3*) *cart* ⇒ *bool*) (*?ul*::(*real*, *3*) *cart list*) (*Suc* (*?i*::*nat*)) = *closest_point* (*voronoi_list ?V* (*truncate_simplex* (*Suc ?i*) *?ul*)) (*omega_list_n ?V ?ul ?i*)

thm Sphere.OMEGA_LIST_N_conjunct0:

omega_list_n (*?V*::(*real*, *3*) *cart* ⇒ *bool*) (*?ul*::(*real*, *3*) *cart list*) (*0*::*nat*) = *hd ?ul*

thm Packing3.OMEGA_LIST_N_LEMMA:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) *i*::*nat*. *k* + (*i* + (*1*::*nat*)) ≤ *length ul* ⟶ *omega_list_n V ul k* = *omega_list_n V* (*truncate_simplex* (*k* + *i*) *ul*) *k*

thm Packing3.OMEGA_LIST_LEMMA:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *k* + (*1*::*nat*) ≤ *length ul* ⟶ *omega_list V* (*truncate_simplex k ul*) = *omega_list_n V ul k*

thm Packing3.BARV_IMP_VORONOI_LIST_NOT_EMPTY:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *barV V k ul* ⟶ *voronoi_list V ul* ≠ *EMPTY*

thm Packing3.OMEGA_LIST_N_IN_VORONOI_LIST:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) *i*::*nat*. *barV V k ul* ∧ *i* ≤ *k* ⟶ *IN* (*omega_list_n V ul i*) (*voronoi_list V* (*truncate_simplex i ul*))

thm Packing3.OMEGA_LIST_IN_VORONOI_LIST:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *barV V k ul* ⟶ *IN* (*omega_list V ul*) (*voronoi_list V ul*)

thm Rogers.BIS_FACE_OF_BIS_LE:

∀(*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *face_of* (*bis u v*) (*bis_le u v*)

thm Rogers.KHEJKCI_GEN:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*k*::*nat*) (*r*::*nat*) (*ul*::(*real*, *3*) *cart list*) *vl*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V k ul* ∧ *barV V r vl* ∧ *initial_sublist ul vl* ⟶ *face_of* (*voronoi_list V vl*) (*voronoi_list V ul*)

thm Rogers.KHEJKCI:

∀(*V*::(*real*, *3*) *cart* ⇒ *bool*) (*k*::*nat*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V k ul* ⟶ *face_of* (*voronoi_list V ul*) (*voronoi_closed V* (*hd ul*))

thm Rogers.VORONOI_BARV_CANONICAL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (k::nat)\ ul::(real, 3)\ cart\ list.\ packing\ V \wedge saturated\ V \wedge barV\ V\ k\ ul \longrightarrow (\exists K::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ FINITE\ K \wedge voronoi\_list\ V\ ul = HOL\_Light\_Import.INTER\ (hull\ affine\ (voronoi\_list\ V\ ul))\ (INTERS\ K) \wedge (\forall a::(real, 3)\ cart \Rightarrow bool.\ IN\ a\ K \longrightarrow (\exists v::(real, 3)\ cart.\ IN\ v\ V \wedge v \neq hd\ ul \wedge (a = bis\_le\ v\ (hd\ ul) \vee a = bis\_le\ (hd\ ul)\ v))) \wedge (\forall K'::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ K'\ K \longrightarrow PSUBSET\ (voronoi\_list\ V\ ul)\ (HOL\_Light\_Import.INTER\ (hull\ affine\ (voronoi\_list\ V\ ul))\ (INTERS\ K'))))$

thm Rogers.REAL_LINE_BOUNDED:

$\forall (a::real)\ b::real.\ (\forall t::real.\ t * a \leq b) \longrightarrow a = (0::real)$

thm Rogers.REAL_NEG_LE_RMUL:

$\forall (x::real)\ (y::real)\ z::real.\ z < (0::real) \longrightarrow (x \leq y) = (y * z \leq x * z)$

thm Rogers.HALFSPACE_EQ:

$\forall (a::(real, ?'a::type)\ cart)\ (b::real)\ (c::(real, ?'a::type)\ cart)\ d::real.\ (GSPEC\ (\lambda GEN\%PVAR\%897::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%897\ (dot\ a\ x \leq b)\ x) = GSPEC\ (\lambda GEN\%PVAR\%898::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%898\ (dot\ c\ x \leq d)\ x)) = ((\exists t::real.\ c = \%\ t\ a \wedge d = t * b \wedge (0::real) < t) \vee a = vec\ (0::nat) \wedge c = vec\ (0::nat) \wedge ((0::real) \leq b \wedge (0::real) \leq d \vee b < (0::real) \wedge d < (0::real)))$

thm Rogers.HALFSPACE_EQ_BIS_LE_IMP_HYPERPLANE_EQ_BIS:

$\forall (a::(real, ?'a::type)\ cart)\ (b::real)\ (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ a \neq vec\ (0::nat) \wedge GSPEC\ (\lambda GEN\%PVAR\%899::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%899\ (dot\ a\ x \leq b)\ x) = bis\_le\ v\ w \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%900::(real, ?'a::type)\ cart.\ \exists x::(real, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%900\ (dot\ a\ x = b)\ x) = bis\ v\ w$

thm Rogers.FACET_OF_POLYHEDRON_EXPLICIT_BIS:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (K::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (s::(real, 3)\ cart \Rightarrow bool)\ u::(real, 3)\ cart.\ FINITE\ K \wedge s = HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ K) \wedge (\forall a::(real, 3)\ cart \Rightarrow bool.\ IN\ a\ K \longrightarrow (\exists v::(real, 3)\ cart.\ IN\ v\ V \wedge v \neq u \wedge (a = bis\_le\ v\ u \vee a = bis\_le\ u\ v))) \wedge (\forall K'::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ PSUBSET\ K'\ K \longrightarrow PSUBSET\ s\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (INTERS\ K'))) \longrightarrow (\forall c::(real, 3)\ cart \Rightarrow bool.\ facet\_of\ c\ s = (\exists v::(real, 3)\ cart.\ IN\ v\ V \wedge (IN\ (bis\_le\ v\ u)\ K \vee IN\ (bis\_le\ u\ v)\ K) \wedge c = HOL\_Light\_Import.INTER\ s\ (bis\ u\ v)))$

thm Rogers.IDBEZAL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ F::(real, 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge barV\ V\ k\ ul \wedge k < (3::nat) \longrightarrow facet\_of\ F\ (voronoi\_list\ V\ ul) = (\exists vl::(real, 3)\ cart\ list.\ F = voronoi\_list\ V\ vl \wedge barV\ V\ (k + (1::nat))\ vl \wedge truncate\_simplex\ k\ vl = ul)$

thm Rogers.VORONOI_LIST_EQ_UNION_CONVEX_HULL_FACETS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) *p*::(*real*, *3*) *cart*. *packing V* ∧ *saturated V* ∧ *barV V k ul* ∧ *k* < (*3*::*nat*) ∧ *IN p* (*voronoi_list V ul*) ⟶ *voronoi_list V ul* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%906*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *vl*::(*real*, *3*) *cart list*. *SETSPEC GEN%PVAR%906* (*barV V* (*k* + (*1*::*nat*)) *vl* ∧ *truncate_simplex k vl* = *ul*) (*hull convex* (*INSERT p* (*voronoi_list V vl*)))))

thm Rogers.NUMSEG_SUBSET_INDUCT:

∀ (*s*::*nat* ⇒ *bool*) (*a*::*nat*) *b*::*nat*. *IN a s* ∧ (∀ *k*::*nat*. *a* ≤ *k* ∧ *Suc k* ≤ *b* ∧ *IN k s* ⟶ *IN* (*Suc k*) *s*) ⟶ *SUBSET* (*dotdot a b*) *s*

thm Rogers.BARV_EXISTS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*wl*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *saturated V* ∧ *k* < (*3*::*nat*) ∧ *barV V k wl* ⟶ (∃ *vl*::(*real*, *3*) *cart list*. *barV V* (*Suc k*) *vl* ∧ *truncate_simplex k vl* = *wl*)

thm Rogers.BARV_EXISTS_ALT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *k*::*nat*. *packing V* ∧ *saturated V* ∧ *k* ≤ (*3*::*nat*) ⟶ (∃ *ul*::(*real*, *3*) *cart list*. *barV V k ul*)

thm Rogers.GLTVHUM_lemma1:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *j*::*nat*. *packing V* ∧ *saturated V* ∧ *j* < (*3*::*nat*) ∧ *barV V j ul* ⟶ *GSPEC* (λ*GEN%PVAR%934*::*nat*. ∃ *k*::*nat*. *SETSPEC GEN%PVAR%934* (*IN k* (*dotdot j* (*3*::*nat*)) ∧ *voronoi_list V ul* = *UNIONS* (*GSPEC* (λ*GEN%PVAR%933*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *vl*::(*real*, *3*) *cart list*. *SETSPEC GEN%PVAR%933* (*barV V k vl* ∧ *truncate_simplex j vl* = *ul*) (*hull convex* (*HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%932*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%932* (*IN i* (*dotdot j* (*k* − (*1*::*nat*)))) (*omega_list_n V vl i*))) (*voronoi_list V vl*)))))) *k*) = *dotdot j* (*3*::*nat*)

thm Rogers.GLTVHUM:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*u0*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart*. *packing V* ∧ *saturated V* ∧ *IN u0 V* ⟶ *IN p* (*voronoi_closed V u0*) = (∃ *vl*::(*real*, *3*) *cart list*. *IN vl* (*barV V* (*3*::*nat*)) ∧ *IN p* (*rogers V vl*) ∧ *truncate_simplex* (*0*::*nat*) *vl* = [*u0*])

thm Rogers.VORONOI_CLOSED_EQ_LEMMA:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *packing V* ∧ *IN u V* ∧ *IN v V* ∧ *voronoi_closed V u* = *voronoi_closed V v* ⟶ *u* = *v*

thm Rogers.ODIGPXU_lemma:

∀ (*P*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*f'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*p0*::(*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'a*::*type*) *cart*) (*q*::(*real*, *?'a*::*type*) *cart*) (*t*::*real*) *s*::*real*. *polyhedron P* ∧ *IN p0 P* ∧ ¬ *IN p0* (*HOL_Light_Import*.*UNION f f'*) ∧ *facet_of f P* ∧ *facet_of f' P* ∧ *IN p f* ∧

IN q f′ ∧ (*0*::*real*) < *t* ∧ (*0*::*real*) < *s* ∧ *vector_add* (% ((*1*::*real*) − *t*) *p0*) (% *t p*) = *vector_add* (% ((*1*::*real*) − *s*) *p0*) (% *s q*) ⟶ *s* ≤ *t*

thm Rogers.ODIGPXU:

∀ (*P*::(*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*f*::(*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*f′*::(*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*p0*::(*real*, *?′a*::*type*) *cart*) (*p*::(*real*, *?′a*::*type*) *cart*) (*q*::(*real*, *?′a*::*type*) *cart*) (*t*::*real*) *s*::*real*. *polyhedron P* ∧ *IN p0 P* ∧ ¬ *IN p0* (*HOL_Light_Import.UNION f f′*) ∧ *facet_of f P* ∧ *facet_of f′ P* ∧ *IN p f* ∧ *IN q f′* ∧ (*0*::*real*) < *t* ∧ (*0*::*real*) < *s* ∧ *vector_add* (% ((*1*::*real*) − *t*) *p0*) (% *t p*) = *vector_add* (% ((*1*::*real*) − *s*) *p0*) (% *s q*) ⟶ *s* = *t*

thm Rogers.OMEGA_LIST_N_EQ:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*i*::*nat*) *j*::*?′a*::*type*. *IN* (*omega_list_n V ul i*) (*voronoi_list V* (*truncate_simplex* (*Suc i*) *ul*)) ⟶ *omega_list_n V ul* (*Suc i*) = *omega_list_n V ul i*

thm Rogers.OMEGA_LIST_N_IN_FACET:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) *i*::*nat*. *packing V* ∧ *saturated V* ∧ *barV V k ul* ∧ *i* < *k* ⟶ (∃ *F*::(*real*, *3*) *cart* ⇒ *bool*. *facet_of F* (*voronoi_list V* (*truncate_simplex i ul*)) ∧ *voronoi_list V* (*truncate_simplex* (*i* + (*1*::*nat*)) *ul*) = *F* ∧ (∀ *j*::*nat*. *i* < *j* ∧ *j* ≤ *k* ⟶ *IN* (*omega_list_n V ul j*) *F*))

thm Rogers.OMEGA_LIST_N_IN_VORONOI_LIST_GEN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) (*i*::*nat*) *j*::*nat*. *packing V* ∧ *saturated V* ∧ *barV V k ul* ∧ *i* ≤ *j* ∧ *j* ≤ *k* ⟶ *IN* (*omega_list_n V ul j*) (*voronoi_list V* (*truncate_simplex i ul*))

thm Rogers.VORONOI_SET_SUBSET:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) *t*::(*real*, *3*) *cart* ⇒ *bool*. *SUBSET s t* ⟶ *SUBSET* (*voronoi_set V t*) (*voronoi_set V s*)

thm Rogers.TRUNCATE_SIMPLEX_SUBSET:

∀ (*ul*::*?′a*::*type list*) (*i*::*nat*) *j*::*nat*. *j* ≤ *i* ∧ *i* + (*1*::*nat*) ≤ *length ul* ⟶ *SUBSET* (*set_of_list* (*truncate_simplex j ul*)) (*set_of_list* (*truncate_simplex i ul*))

thm Rogers.OMEGA_LIST_N_EQ_GEN:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*k*::*nat*) (*i*::*nat*) *j*::*nat*. *packing V* ∧ *saturated V* ∧ *barV V k ul* ∧ *i* < *j* ∧ *j* ≤ *k* ∧ *IN* (*omega_list_n V ul i*) (*voronoi_list V* (*truncate_simplex j ul*)) ⟶ *omega_list_n V ul* (*Suc i*) = *omega_list_n V ul i*

thm Rogers.CARD_LE_3:

∀ *s*::*?′a*::*type* ⇒ *bool*. *s* ≠ *EMPTY* ∧ *FINITE s* ∧ *CARD s* ≤ (*3*::*nat*) ⟶ (∃ (*x*::*?′a*::*type*) (*y*::*?′a*::*type*) *z*::*?′a*::*type*. *s* = *INSERT x* (*INSERT y* (*INSERT z EMPTY*)))

thm Rogers.AFF_DIM_LE_2_IMP_COPLANAR:

$\forall s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ aff\_dim\ s \leq int\ (2::nat) \longrightarrow coplanar\ s$

thm Rogers.SET_OF_LIST_TRUNCATE_SIMPLEX_SUBSET:

$\forall (ul::?'a::type\ list)\ k::nat.\ k + (1::nat) \leq length\ ul \longrightarrow SUBSET\ (set\_of\_list\ (truncate\_simplex\ k\ ul))\ (set\_of\_list\ ul)$

thm Rogers.ROGERS_AFF_DIM_FULL:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ barV\ V\ (3::nat)\ ul \wedge aff\_dim\ (rogers\ V\ ul) = int\ (3::nat) \longrightarrow (\forall (i::nat)\ j::nat.\ i < (4::nat) \wedge j < (4::nat) \wedge i \neq j \longrightarrow omega\_list\_n\ V\ ul\ i \neq omega\_list\_n\ V\ ul\ j)$

thm Rogers.VORONOI_LIST_AFF_DIM:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (k::nat)\ i::nat.\ barV\ V\ k\ ul \wedge i \leq k \longrightarrow aff\_dim\ (voronoi\_list\ V\ (truncate\_simplex\ i\ ul)) = int\ (3::nat) - int\ i$

thm Rogers.AFF_DIM_FINITE_UNION_LE:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ FINITE\ s \longrightarrow aff\_dim\ (HOL\_Light\_Import.UNION\ s\ t) \leq int\ (CARD\ s) + aff\_dim\ t$

thm Rogers.DUUNHOR:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ vl::(real,\ 3)\ cart\ list.\ packing\ V \wedge saturated\ V \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge IN\ vl\ (barV\ V\ (3::nat)) \wedge rogers\ V\ ul \neq rogers\ V\ vl \longrightarrow coplanar\ (HOL\_Light\_Import.INTER\ (rogers\ V\ ul)\ (rogers\ V\ vl))$

thm Rogers.AFFINE_INDEPENDENT_IMP_INDEPENDENT:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ S \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ S \longrightarrow independent\ (GSPEC\ (\lambda GEN\%PVAR\%971::(real,\ ?'a::type)\ cart.\ \exists y::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%971\ (IN\ y\ (DELETE\ S\ x))\ (vector\_sub\ y\ x))))$

thm Rogers.ORTHOGONAL_TO_SPAN_EXISTS:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ s\ (span\ t) \wedge dim\ s < dim\ t \longrightarrow (\exists v::(real,\ ?'a::type)\ cart.\ v \neq vec\ (0::nat) \wedge IN\ v\ (span\ t) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow dot\ x\ v = (0::real)))$

thm Rogers.INDEPENDENT_EXPLICIT_NUMSEG:

$\forall (v::nat \Rightarrow (real,\ ?'a::type)\ cart)\ (f::nat \Rightarrow real)\ n::nat.\ (\forall (i::nat)\ j::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \wedge IN\ j\ (dotdot\ (1::nat)\ n) \wedge v\ i = v\ j \longrightarrow i = j) \wedge independent\ (IMAGE\ v\ (dotdot\ (1::nat)\ n)) \wedge vsum\ (dotdot\ (1::nat)\ n)\ (\lambda i::nat.\ \%\ (f\ i)\ (v\ i)) = vec\ (0::nat) \longrightarrow (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow f\ i = (0::real))$

thm Rogers.ORTHOGONAL_TO_ALL_IMP_ZERO:

$\forall (v::(real,\ ?'a::type)\ cart)\ s::(real,\ ?'a::type)\ cart \Rightarrow bool.\ IN\ v\ (span\ s) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ s \longrightarrow dot\ v\ x = (0::real)) \longrightarrow v = vec\ (0::nat)$

thm Rogers.UNIQUE_SOLUTION_lemma:

$\forall (S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ b::(real,\ ?'a::type)\ cart \Rightarrow real.\ independent\ S \longrightarrow (\exists!p::(real,\ ?'a::type)\ cart.\ IN\ p\ (span\ S) \wedge (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ S \longrightarrow dot\ p\ x = b\ x))$

thm Rogers.UNIQUE_SOLUTION_AFFINE_INDEPENDENT:

$\forall (S::(real,\ ?'a::type)\ cart \Rightarrow bool)\ b::(real,\ ?'a::type)\ cart \Rightarrow real.\ S \neq EMPTY \wedge \neg\ affine\_dependent\ S \longrightarrow (\exists!p::(real,\ ?'a::type)\ cart.\ IN\ p\ (hull\ affine\ S) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ S \wedge IN\ y\ S \longrightarrow dot\ p\ (vector\_sub\ x\ y) = b\ x - b\ y))$

thm Rogers.QXSKIIT:

$\forall (vf::?'b::type \Rightarrow (real,\ ?'a::type)\ cart)\ b::?'b::type \Rightarrow real.\ FINITE\ (IMAGE\ vf\ HOL\_Light\_Import.UNIV) \wedge \neg\ affine\_dependent\ (IMAGE\ vf\ HOL\_Light\_Import.UNIV) \wedge (\forall (i::?'b::type)\ j::?'b::type.\ vf\ i = vf\ j \longrightarrow b\ i = b\ j) \longrightarrow (\exists!p::(real,\ ?'a::type)\ cart.\ IN\ p\ (hull\ affine\ (IMAGE\ vf\ HOL\_Light\_Import.UNIV)) \wedge (\forall (i::?'b::type)\ j::?'b::type.\ dot\ p\ (vector\_sub\ (vf\ i)\ (vf\ j)) = b\ i - b\ j))$

thm Rogers.CIRCUMCENTER_lemma:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ S \neq EMPTY \wedge \neg\ affine\_dependent\ S \longrightarrow (\exists!p::(real,\ ?'a::type)\ cart.\ IN\ p\ (hull\ affine\ S) \wedge (\forall (x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ S \wedge IN\ y\ S \longrightarrow distance\ (p,\ x) = distance\ (p,\ y)))$

thm Rogers.CIRCUMCENTER_LEMMA:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ S \neq EMPTY \wedge \neg\ affine\_dependent\ S \longrightarrow (\exists!p::(real,\ ?'a::type)\ cart.\ IN\ p\ (hull\ affine\ S) \wedge (\exists c::real.\ \forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ S \longrightarrow c = distance\ (p,\ w)))$

thm Rogers.OAPVION1:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ S \neq EMPTY \wedge \neg\ affine\_dependent\ S \longrightarrow IN\ (circumcenter\ S)\ (hull\ affine\ S)$

thm Rogers.OAPVION2:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ S \longrightarrow (\forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ S \longrightarrow radV\ S = distance\ (circumcenter\ S,\ w))$

thm Rogers.OAPVION3:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ S \longrightarrow (\forall p::(real,\ ?'a::type)\ cart.\ IN\ p\ (hull\ affine\ S) \wedge (\exists c::real.\ \forall w::(real,\ ?'a::type)\ cart.\ IN\ w\ S \longrightarrow distance\ (p,\ w) = c) \longrightarrow p = circumcenter\ S)$

thm Rogers.CIRCUMCENTER_1:

$\forall x::(real,\ ?'a::type)\ cart.\ circumcenter\ (INSERT\ x\ EMPTY) = x$

thm Rogers.CIRCUMCENTER_NOT_EQ:

$\forall S::(real,\ ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ S \wedge (1::nat) < CARD\ S \longrightarrow (\forall x::(real,\ ?'a::type)\ cart.\ IN\ x\ S \longrightarrow circumcenter\ S \neq x)$

thm Rogers.CIRCUMCENTER_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ s \neq EMPTY \wedge \neg\ affine\_dependent\ s \longrightarrow circumcenter\ (IMAGE\ (vector\_add\ a)\ s) = vector\_add\ a\ (circumcenter\ s)$

thm Rogers.RADV_TRANSLATION:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ a::(real,\ ?'a::type)\ cart.\ \neg\ affine\_dependent\ s \longrightarrow radV\ (IMAGE\ (vector\_add\ a)\ s) = radV\ s$

thm Rogers.AFF_INTER_SUBSET_INTER_AFF:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ t::(real,\ ?'a::type)\ cart \Rightarrow bool.\ SUBSET\ (hull\ affine\ (HOL\_Light\_Import.INTER\ s\ t))\ (HOL\_Light\_Import.INTER\ (hull\ affine\ s)\ (hull\ affine\ t))$

thm Rogers.MHFTTZN_lemma:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow hull\ affine\ (voronoi\_list\ V\ ul) = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%977::(real,\ 3)\ cart \Rightarrow bool.\ \exists u::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%977\ (IN\ u\ (set\_of\_list\ ul))\ (bis\ (hd\ ul)\ u)))$

thm Rogers.MHFTTZN_lemma2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow aff\_dim\ (set\_of\_list\ ul) = int\ k \wedge (\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ (hull\ affine\ (voronoi\_list\ V\ ul)) \wedge IN\ v\ (hull\ affine\ (set\_of\_list\ ul)) \longrightarrow dot\ (vector\_sub\ u\ (circumcenter\ (set\_of\_list\ ul)))\ (vector\_sub\ v\ (circumcenter\ (set\_of\_list\ ul))) = (0::real))$

thm Rogers.MHFTTZN1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow aff\_dim\ (set\_of\_list\ ul) = int\ k$

thm Rogers.MHFTTZN2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow (\forall p::(real,\ 3)\ cart.\ IN\ p\ (hull\ affine\ (voronoi\_list\ V\ ul)) = (\forall u::(real,\ 3)\ cart.\ IN\ u\ (set\_of\_list\ ul) \longrightarrow IN\ p\ (bis\ (hd\ ul)\ u)))$

thm Rogers.MHFTTZN3:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow HOL\_Light\_Import.INTER\ (hull\ affine\ (voronoi\_list\ V\ ul))\ (hull\ affine\ (set\_of\_list\ ul)) = INSERT\ (circumcenter\ (set\_of\_list\ ul))\ EMPTY$

thm Rogers.MHFTTZN4:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (k::nat)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ packing\ V \wedge barV\ V\ k\ ul \wedge IN\ u\ (hull\ affine\ (voronoi\_list\ V\ ul)) \wedge IN\ v\ (hull\ affine\ (set\_of\_list\ ul)) \longrightarrow dot\ (vector\_sub\ u\ (circumcenter\ (set\_of\_list\ ul)))\ (vector\_sub\ v\ (circumcenter\ (set\_of\_list\ ul))) = (0::real)$

thm Rogers.ARCV_GT_PI2:

$\forall (p::(real,\ ?'a::type)\ cart)\ (u::(real,\ ?'a::type)\ cart)\ v::(real,\ ?'a::type)\ cart.\ (pi\ /\ real\_of\_nat\ (2::nat) < arcV\ p\ u\ v) = (cos\ (arcV\ p\ u\ v) < (0::real))$

thm Rogers.XYOFCGX_lemma0:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (S::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge p = circumcenter\ S \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \wedge IN\ w\ (DIFF\ V\ S) \wedge distance\ (p,\ w) \leq radV\ S \wedge (1::nat) < CARD\ S \longrightarrow (\forall u::(real,\ 3)\ cart.\ IN\ u\ S \longrightarrow pi\ /\ real\_of\_nat\ (2::nat) < arcV\ p\ w\ u) \wedge (\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ S \wedge IN\ v\ S \wedge u \neq v \longrightarrow pi\ /\ real\_of\_nat\ (2::nat) < arcV\ p\ u\ v)$

thm Rogers.CARD_1_IMP_SING:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge CARD\ s = (1::nat) \longrightarrow (\exists x::?'a::type.\ s = INSERT\ x\ EMPTY)$

thm Rogers.XYOFCGX_1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (S::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge p = circumcenter\ S \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \wedge CARD\ S \leq (1::nat) \longrightarrow (\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u,\ p) < distance\ (v,\ p))$

thm Rogers.CIRCUMCENTER_2:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::(real,\ ?'a::type)\ cart.\ circumcenter\ (INSERT\ a\ (INSERT\ b\ EMPTY)) = midpoint\ (a,\ b)$

thm Rogers.CARD_2_IMP_DOUBLE:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \wedge CARD\ s = (2::nat) \longrightarrow (\exists (a::?'a::type)\ b::?'a::type.\ s = INSERT\ a\ (INSERT\ b\ EMPTY) \wedge a \neq b)$

thm Rogers.XYOFCGX_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (S::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge p = circumcenter\ S \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \wedge CARD\ S = (2::nat) \longrightarrow (\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u,\ p) < distance\ (v,\ p))$

thm Rogers.ANGLE_GT_PI2:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. (*pi* / *real_of_nat* (*2*::*nat*) < *angle* (*a*, *b*, *c*)) = (*dot* (*vector_sub a b*) (*vector_sub c b*) < (*0*::*real*))

thm Rogers.AZIM_COMPL_EXT:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *azim v w b a* = (*if azim v w a b* = (*0*::*real*) *then 0*::*real else real_of_nat* (*2*::*nat*) ∗ *pi* − *azim v w a b*)

thm Rogers.AZIM_EQ_SYM:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) *c*::(*real*, *3*) *cart*. (*azim v w b a* = *azim v w c a*) = (*azim v w a b* = *azim v w a c*)

thm Collect_geom.PER_SET3_conjunct4:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) = *INSERT ?c* (*INSERT ?b* (*INSERT ?a EMPTY*))

thm Collect_geom.PER_SET3_conjunct3:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) = *INSERT ?b* (*INSERT ?c* (*INSERT ?a EMPTY*))

thm Collect_geom.PER_SET3_conjunct2:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) = *INSERT ?c* (*INSERT ?a* (*INSERT ?b EMPTY*))

thm Collect_geom.PER_SET3_conjunct1:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) = *INSERT ?b* (*INSERT ?a* (*INSERT ?c EMPTY*))

thm Collect_geom.PER_SET3_conjunct0:

INSERT (*?a*::*?'a*::*type*) (*INSERT* (*?b*::*?'a*::*type*) (*INSERT* (*?c*::*?'a*::*type*) *EMPTY*)) = *INSERT ?a* (*INSERT ?c* (*INSERT ?b EMPTY*))

thm Rogers.STRICT_CYCLIC_IMP_FAN:

$\forall$ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *p*::(*real*, *3*) *cart*. *FINITE V* ∧ (*2*::*nat*) ≤ *CARD V* ∧ ($\forall$ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v V* ∧ *IN w V* ∧ *v* ≠ *w* ⟶ (*0*::*real*) < *azim* (*vec* (*0*::*nat*)) *p v w*) ⟶ *FAN* (*vec* (*0*::*nat*), *HOL_Light_Import*.*UNION V* (*INSERT p EMPTY*), *GSPEC* (λ*GEN%PVAR%991*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%991* (*IN v V*) (*INSERT p* (*INSERT v EMPTY*))))

thm Rogers.STRICT_CYCLIC_FAN_PROPERTIES:

$\forall$ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *p*::(*real*, *3*) *cart*. *LET* (λ*W*::(*real*, *3*) *cart* ⇒ *bool*. *LET_END* (*LET* (λ*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *LET_END* (*FAN* (*vec* (*0*::*nat*), *W*, *E*) ⟶ ($\forall$ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ *IN* (*p*, *v*) (*dart1_of_fan*

$(W,\ E))) \wedge$ *set_of_edge p W E* $= V \wedge (\forall v$::(*real*, *3*) *cart*. *IN v V* $\longrightarrow$ *node* (*hypermap_of_fan* (*W*, *E*)) (*p*, *v*) = *GSPEC* (λ*GEN%PVAR%994*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart*. $\exists$ *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%994* (*IN w V*) (*p*, *w*))))) (*GSPEC* (λ*GEN%PVAR%993*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%993* (*IN v V*) (*INSERT p* (*INSERT v EMPTY*)))))) (*HOL_Light_Import.UNION V* (*INSERT p EMPTY*))

thm Rogers.ANGLE_SUM_lemma:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *p*::(*real*, *3*) *cart*. *FINITE V* $\wedge$ (*2*::*nat*) $\leq$ *CARD V* $\wedge$ ($\forall$ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v V* $\wedge$ *IN w V* $\wedge$ $v \neq w \longrightarrow$ (*0*::*real*) < *azim* (*vec* (*0*::*nat*)) *p v w*) $\longrightarrow$ ($\exists$ *f*::(*real*, *3*) *cart* $\Rightarrow$ (*real*, *3*) *cart*. ($\forall$ *x*::(*real*, *3*) *cart*. *IN x V* $\longrightarrow$ *IN* (*f x*) *V* $\wedge$ $x \neq f\ x$) $\wedge$ *sum V* (λ*x*::(*real*, *3*) *cart*. *azim* (*vec* (*0*::*nat*)) *p x* (*f x*)) = *real_of_nat* (*2*::*nat*) $*$ *pi*)

thm Rogers.ANGLE_SUM_BOUND:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*p*::(*real*, *3*) *cart*) *a*::*real*. (*0*::*real*) $\leq$ *a* $\wedge$ *FINITE V* $\wedge$ (*2*::*nat*) $\leq$ *CARD V* $\wedge$ ($\forall$ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v V* $\wedge$ *IN w V* $\wedge$ $v \neq w \longrightarrow a <$ *azim* (*vec* (*0*::*nat*)) *p v w*) $\longrightarrow$ *a* $*$ *real_of_nat* (*CARD V*) < *real_of_nat* (*2*::*nat*) $*$ *pi*

thm Rogers.DIHV_LE_AZIM:

$\forall$ (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT x EMPTY*))) $\wedge$ $\neg$ *collinear* (*INSERT v* (*INSERT w* (*INSERT y EMPTY*))) $\longrightarrow$ *dihV v w x y* $\leq$ *azim v w x y*

thm Rogers.IN_PLANE_NOT_COLLINEAR:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. $v \neq$ *vec* (*0*::*nat*) $\wedge$ $n \neq$ *vec* (*0*::*nat*) $\wedge$ *dot n v* = (*0*::*real*) $\longrightarrow$ $\neg$ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT n* (*INSERT v EMPTY*)))

thm Rogers.ANGLE_EQ_DIHV:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. $v \neq$ *vec* (*0*::*nat*) $\wedge$ $w \neq$ *vec* (*0*::*nat*) $\wedge$ $n \neq$ *vec* (*0*::*nat*) $\wedge$ *dot n v* = (*0*::*real*) $\wedge$ *dot n w* = (*0*::*real*) $\longrightarrow$ *angle* (*v*, *vec* (*0*::*nat*), *w*) = *dihV* (*vec* (*0*::*nat*)) *n v w*

thm Rogers.PYTHAGORAS_PROJECTION:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *dot x n* = (*0*::*real*) $\longrightarrow$ (*distance* (*x*, *y*))^2 = (*distance* (*x*, *projection n y*))^2 + (*distance* (*projection n y*, *y*))^2

thm Rogers.OBTUSE_ANGLE_PROJECTION:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *pi* / *real_of_nat* (*2*::*nat*) < *angle* (*a*, *vec* (*0*::*nat*), *w*) $\wedge$ *dot a n* = (*0*::*real*) $\longrightarrow$ *pi* / *real_of_nat* (*2*::*nat*) < *angle* (*a*, *vec* (*0*::*nat*), *projection n w*)

thm Rogers.XYOFCGX_3_0:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ S::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge circumcenter\ S = vec\ (0::nat) \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \wedge CARD\ S = (3::nat) \longrightarrow (\forall (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u, vec\ (0::nat)) < distance\ (v, vec\ (0::nat)))$

thm Rogers.DIHV_GT_PI2:

$\forall (v::(real, ?'a::type)\ cart)\ (w::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ y::(real, ?'a::type)\ cart.\ (pi\ /\ real\_of\_nat\ (2::nat) < dihV\ v\ w\ x\ y) = (cos\ (dihV\ v\ w\ x\ y) < (0::real))$

thm Rogers.XYOFCGX_4_0:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ S::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge circumcenter\ S = vec\ (0::nat) \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \wedge CARD\ S = (4::nat) \longrightarrow (\forall (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u, vec\ (0::nat)) < distance\ (v, vec\ (0::nat)))$

thm Rogers.XYOFCGX:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (S::(real, 3)\ cart \Rightarrow bool)\ p::(real, 3)\ cart.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge p = circumcenter\ S \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow (\forall (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u, p) < distance\ (v, p))$

thm Rogers.BARV_AFFINE_INDEPENDENT:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow \neg\ affine\_dependent\ (set\_of\_list\ ul)$

thm Rogers.BARV_IMP_LENGTH_EQ_CARD:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow length\ ul = k + (1::nat) \wedge CARD\ (set\_of\_list\ ul) = k + (1::nat)$

thm Rogers.AFFINE_HULL_PROJECTION_EXISTS:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ p::(real, ?'a::type)\ cart.\ S \neq EMPTY \longrightarrow (\exists (x::(real, ?'a::type)\ cart)\ n::(real, ?'a::type)\ cart.\ p = vector\_add\ x\ n \wedge IN\ x\ (hull\ affine\ S) \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)))$

thm Rogers.AFFINE_HULL_PROJECTION_DIST_EQ:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ (p::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ (w::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ n::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \wedge distance\ (p, v) = distance\ (p, w) \wedge p = vector\_add\ x\ n \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)) \longrightarrow distance\ (x, v) = distance\ (x, w)$

thm Rogers.ORTHOGONAL_TO_AFFINE_HULL_EQ:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ n::(real, ?'a::type)\ cart.\ (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)) = (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ (hull\ affine\ S) \wedge IN\ w\ (hull\ affine\ S) \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real))$

thm Rogers.AFFINE_HULL_PROJECTION_DIST_LE:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ (p::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ n::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge p = vector\_add\ x\ n \wedge IN\ x\ (hull\ affine\ S) \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)) \longrightarrow distance\ (x, v) \leq distance\ (p, v)$

thm Rogers.AFFINE_HULL_PROJECTION_DIST_LT:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ (p::(real, ?'a::type)\ cart)\ (v::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ n::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge \neg\ IN\ p\ (hull\ affine\ S) \wedge p = vector\_add\ x\ n \wedge IN\ x\ (hull\ affine\ S) \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)) \longrightarrow distance\ (x, v) < distance\ (p, v)$

thm Rogers.AFFINE_HULL_CIRCUMCENTER_PROJECTION:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \wedge SUBSET\ t\ s \wedge t \neq EMPTY \longrightarrow (\exists\, n::(real, ?'a::type)\ cart.\ circumcenter\ s = vector\_add\ (circumcenter\ t)\ n \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ t \wedge IN\ w\ t \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)))$

thm Rogers.AFFINE_HULL_CIRCUMCENTER_EQ:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \wedge SUBSET\ t\ s \wedge t \neq EMPTY \wedge IN\ (circumcenter\ s)\ (hull\ affine\ t) \longrightarrow circumcenter\ s = circumcenter\ t$

thm Rogers.AFFINE_HULL_RADV:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \wedge SUBSET\ t\ s \wedge t \neq EMPTY \longrightarrow (radV\ s)^2 = (radV\ t)^2 + (distance\ (circumcenter\ t, circumcenter\ s))^2 \wedge (0::real) \leq radV\ s \wedge (0::real) \leq radV\ t$

thm Rogers.RADV_MONO:

$\forall (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ \neg\ affine\_dependent\ s \wedge SUBSET\ t\ s \wedge t \neq EMPTY \longrightarrow radV\ t \leq radV\ s$

thm Rogers.HL_PROPERTIES:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow (\forall\, w::(real, 3)\ cart.\ IN\ w\ (set\_of\_list\ ul) \longrightarrow distance\ (circumcenter\ (set\_of\_list\ ul), w) = hl\ ul)$

thm Rogers.BARV_CIRCUMCENTER_EXISTS:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow IN\ (circumcenter\ (set\_of\_list\ ul))\ (hull\ affine\ (set\_of\_list\ ul))$

thm Rogers.HL_EQ_DIST0:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (k::nat)\ ul::(real, 3)\ cart\ list.\ packing\ V \wedge barV\ V\ k\ ul \longrightarrow hl\ ul = distance\ (circumcenter\ (set\_of\_list\ ul), hd\ ul)$

thm Rogers.BARV_CIRCUMCENTER_PROJECTION:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ i::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge i \leq k \longrightarrow LET\ (\lambda S::(real, 3)\ cart \Rightarrow bool.\ LET\_END\ (\exists\, n::(real, 3)\ cart.\ circumcenter\ (set\_of\_list\ ul) = vector\_add\ (circumcenter\ S)\ n \wedge (\forall (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ IN\ v\ S \wedge IN\ w\ S \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real))))\ (set\_of\_list\ (truncate\_simplex\ i\ ul))$

thm Rogers.HL_DECREASE:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ i::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge i \leq k \longrightarrow hl\ (truncate\_simplex\ i\ ul) \leq hl\ ul$

thm Rogers.XNHPWAB1:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow omega\_list\ V\ ul = circumcenter\ (set\_of\_list\ ul)$

thm Rogers.AFFINE_HULL_PROJECTION_SEPARATES:

$\forall (S::(real, ?'a::type)\ cart \Rightarrow bool)\ p::(real, ?'a::type)\ cart.\ FINITE\ S \wedge IN\ p\ (hull\ affine\ S) \wedge \neg\ IN\ p\ (hull\ convex\ S) \longrightarrow (\exists\, u::(real, ?'a::type)\ cart.\ IN\ u\ S \wedge (\forall (x::(real, ?'a::type)\ cart)\ n::(real, ?'a::type)\ cart.\ p = vector\_add\ x\ n \wedge IN\ x\ (hull\ affine\ (DELETE\ S\ u)) \wedge (\forall (v::(real, ?'a::type)\ cart)\ w::(real, ?'a::type)\ cart.\ IN\ v\ (DELETE\ S\ u) \wedge IN\ w\ (DELETE\ S\ u) \longrightarrow dot\ (vector\_sub\ v\ w)\ n = (0::real)) \longrightarrow dot\ (vector\_sub\ p\ x)\ (vector\_sub\ u\ x) \leq (0::real)))$

thm Rogers.XNHPWAB2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow IN\ (omega\_list\ V\ ul)\ (hull\ convex\ (set\_of\_list\ ul))$

thm Rogers.XNHPWAB4:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow (\forall (i::nat)\ j::nat.\ i < j \wedge j \leq k \longrightarrow hl\ (truncate\_simplex\ i\ ul) < hl\ (truncate\_simplex\ j\ ul))$

thm Rogers.OMEGA_LIST_N_IN_CONVEX_HULL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ i::nat.\ packing\ V \wedge barV\ V\ k\ ul \wedge i \leq k \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow IN\ (omega\_list\_n\ V\ ul\ i)\ (hull\ convex\ (set\_of\_list\ ul))$

thm Rogers.XNHPWAB3:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow aff\_dim\ (GSPEC\ (\lambda GEN\%PVAR\%1002::(real,\ 3)\ cart.\ \exists j::nat.\ SETSPEC\ GEN\%PVAR\%1002\ (IN\ j\ (dotdot\ (0::nat)\ k))\ (omega\_list\_n\ V\ ul\ j))) = int\ k$

thm Rogers.IN_VORONOI_LIST_IMP_IN_BIS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (k::nat)\ x::(real,\ 3)\ cart.\ barV\ V\ k\ ul \wedge IN\ x\ (voronoi\_list\ V\ ul) \longrightarrow (\forall u::(real,\ 3)\ cart.\ IN\ u\ (set\_of\_list\ ul) \longrightarrow IN\ x\ (bis\ (hd\ ul)\ u))$

thm Rogers.WAUFCHE1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \longrightarrow hl\ ul \leq distance\ (omega\_list\ V\ ul,\ hd\ ul)$

thm Rogers.WAUFCHE2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ packing\ V \wedge IN\ ul\ (barV\ V\ k) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow hl\ ul = distance\ (omega\_list\ V\ ul,\ hd\ ul)$

thm Rogers.CIRCUMCENTER_IN_VORONOI_SET:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ S\ V \wedge \neg\ affine\_dependent\ S \wedge radV\ S < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow IN\ (circumcenter\ S)\ (voronoi\_set\ V\ S)$

thm Rogers.NEIGHBORHOOD_lemma:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (S::(real,\ 3)\ cart \Rightarrow bool)\ p::(real,\ 3)\ cart.\ packing\ V \wedge SUBSET\ S\ V \wedge (\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u,\ p) < distance\ (v,\ p)) \longrightarrow (\exists r>0::real.\ \forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ IN\ x\ (ball\ (p,\ r)) \wedge IN\ u\ S \wedge IN\ v\ (DIFF\ V\ S) \longrightarrow distance\ (u,\ x) < distance\ (v,\ x))$

thm Rogers.SUBSPACES_INTER_BALL_EQ_IMP_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ r::real.\ subspace\ s \wedge subspace\ t \wedge (0::real) < r \wedge HOL\_Light\_Import.INTER\ s\ (ball\ (vec\ (0::nat),\ r)) = HOL\_Light\_Import.INTER\ t\ (ball\ (vec\ (0::nat),\ r)) \longrightarrow s = t$

thm Rogers.AFFINES_INTER_BALL_EQ_IMP_EQ:

$\forall (s::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (t::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (x::(real,\ ?'a::type)\ cart)\ r::real.\ affine\ s \wedge affine\ t \wedge (0::real) < r \wedge HOL\_Light\_Import.INTER\ s\ (ball\ (x,\ r)) = HOL\_Light\_Import.INTER\ t\ (ball\ (x,\ r)) \wedge IN\ x\ s \longrightarrow s = t$

thm Rogers.VORONOI_LIST_EQ_INTERS_BIS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ SUBSET\ (set\_of\_list\ ul)\ V \wedge (1::nat) \leq length\ ul \longrightarrow voronoi\_list\ V\ ul = HOL\_Light\_Import.INTER\ (voronoi\_closed\ V\ (hd\ ul))\ (INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%1006::(real,$

3) cart ⇒ bool. ∃ u::(real, 3) cart. SETSPEC GEN%PVAR%1006 (IN u (set_of_list ul)) (bis (hd ul) u))))

thm Rogers.AFFINE_HULL_VORONOI_LIST_SUBSET_INTERS_BIS:

∀ (V::(real, 3) cart ⇒ bool) ul::(real, 3) cart list. SUBSET (set_of_list ul) V ⟶ SUBSET (hull affine (voronoi_list V ul)) (INTERS (GSPEC (λGEN%PVAR%1010::(real, 3) cart ⇒ bool. ∃ u::(real, 3) cart. SETSPEC GEN%PVAR%1010 (IN u (set_of_list ul)) (bis (hd ul) u))))

thm Rogers.YIFVQDV_lemma_aff_dim:

∀ (V::(real, 3) cart ⇒ bool) vl::(real, 3) cart list. packing V ∧ SUBSET (set_of_list vl) V ∧ ¬ affine_dependent (set_of_list vl) ∧ hl vl < sqrt (real_of_nat (2::nat)) ⟶ aff_dim (voronoi_list V vl) = aff_dim (INTERS (GSPEC (λGEN%PVAR%1013::(real, 3) cart ⇒ bool. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%1013 (IN v (set_of_list vl)) (bis (hd vl) v))))

thm Rogers.YIFVQDV_1:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) (k::nat) p::nat ⇒ nat. packing V ∧ IN ul (barV V k) ∧ hl ul < sqrt (real_of_nat (2::nat)) ∧ permutes p (dotdot (0::nat) k) ⟶ IN (left_action_list p ul) (barV V k)

thm Rogers.YIFVQDV:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) (k::nat) p::nat ⇒ nat. packing V ∧ IN ul (barV V k) ∧ hl ul < sqrt (real_of_nat (2::nat)) ∧ permutes p (dotdot (0::nat) k) ⟶ IN (left_action_list p ul) (barV V k) ∧ omega_list V (left_action_list p ul) = omega_list V ul

thm Rogers.HL_TRUNCATE_SIMPLEX_OMEGA_N:

∀ (V::(real, 3) cart ⇒ bool) (k::nat) (ul::(real, 3) cart list) j::nat. packing V ∧ barV V k ul ∧ j ≤ k ∧ hl ul < sqrt (real_of_nat (2::nat)) ⟶ hl (truncate_simplex j ul) = distance (omega_list_n V ul j, hd ul)

thm Rogers.KSOQKWL_lemma0:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) (vl::(real, 3) cart list) k::nat. packing V ∧ barV V k ul ∧ barV V k vl ∧ hd ul ≠ hd vl ⟶ GSPEC (λGEN%PVAR%1019::(real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%1019 (i ≤ k) (omega_list_n V ul i)) ≠ GSPEC (λGEN%PVAR%1020::(real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%1020 (i ≤ k) (omega_list_n V vl i))

thm Rogers.KSOQKWL_lemma1:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) (vl::(real, 3) cart list) (k::nat) j::nat. packing V ∧ barV V k ul ∧ barV V k vl ∧ hl ul < sqrt (real_of_nat (2::nat)) ∧ hl vl < sqrt (real_of_nat (2::nat)) ∧ (0::nat) < j ∧ j ≤ k ∧ truncate_simplex (j − (1::nat)) ul = truncate_simplex (j − (1::nat)) vl ∧ hl (truncate_simplex j ul) ≤ hl (truncate_simplex j vl) ∧ omega_list_n V ul j ≠ omega_list_n V vl j ⟶ GSPEC (λGEN%PVAR%1021::(real, 3)

cart. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1021* (*i* ≤ *k*) (*omega_list_n V ul i*)) ≠ *GSPEC* (λ*GEN%PVAR%1022*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1022* (*i* ≤ *k*) (*omega_list_n V vl i*))

thm Rogers.AFFINE_INDEPENDENT_OMEGA_LIST_N:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *barV V k ul* ∧ *hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*)) ⟶ ¬ *affine_dependent* (*GSPEC* (λ*GEN%PVAR%1024*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1024* (*i* ≤ *k*) (*omega_list_n V ul i*)))

thm Rogers.ROGERS_EQ:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*vl*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *barV V k ul* ∧ *barV V k vl* ∧ *hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *hl vl* < *sqrt* (*real_of_nat* (*2*::*nat*)) ⟶ (*rogers V ul* = *rogers V vl*) = (*GSPEC* (λ*GEN%PVAR%1025*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1025* (*i* ≤ *k*) (*omega_list_n V ul i*)) = *GSPEC* (λ*GEN%PVAR%1026*::(*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1026* (*i* ≤ *k*) (*omega_list_n V vl i*)))

thm Rogers.NUM_FINITE_IMP_MAX_EXISTS:

∀ *K*::*nat* ⇒ *bool*. *FINITE K* ∧ *K* ≠ *EMPTY* ⟶ (∃ *m*::*nat*. *IN m K* ∧ (∀ *j*::*nat*. *IN j K* ⟶ *j* ≤ *m*))

thm Rogers.NOT_ID_IMP_LISTS_NOT_EQ:

∀ (*ul*::*?'a*::*type list*) (*p*::*nat* ⇒ *nat*) *k*::*nat*. *length ul* = *k* + (*1*::*nat*) ∧ *CARD* (*set_of_list ul*) = *k* + (*1*::*nat*) ∧ *permutes p* (*dotdot* (*0*::*nat*) *k*) ∧ *p* ≠ *id* ⟶ *ul* ≠ *left_action_list p ul*

thm Rogers.NOT_ID_IMP_EXISTS_MAX_EQ_TRUNCATE_SIMPLEX:

∀ (*ul*::*?'a*::*type list*) (*p*::*nat* ⇒ *nat*) *k*::*nat*. *length ul* = *k* + (*1*::*nat*) ∧ *CARD* (*set_of_list ul*) = *k* + (*1*::*nat*) ∧ *permutes p* (*dotdot* (*0*::*nat*) *k*) ∧ *p* ≠ *id* ⟶ *hd ul* ≠ *hd* (*left_action_list p ul*) ∨ (∃ *j*<*k*. *truncate_simplex j ul* = *truncate_simplex j* (*left_action_list p ul*) ∧ *EL* (*j* + (*1*::*nat*)) *ul* ≠ *EL* (*j* + (*1*::*nat*)) (*left_action_list p ul*))

thm Rogers.KSOQKWL:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) (*p*::*nat* ⇒ *nat*) *k*::*nat*. *packing V* ∧ *IN ul* (*barV V k*) ∧ *hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*)) ∧ *permutes p* (*dotdot* (*0*::*nat*) *k*) ∧ *rogers V ul* = *rogers V* (*left_action_list p ul*) ⟶ *p* = *id*

thm Rogers.IVFICRK:

∀ *k*::*nat*. ∃ *g*::*nat* × (*nat* ⇒ *nat*) ⇒ *nat* ⇒ *nat*. *BIJ g* (*GSPEC* (λ*GEN%PVAR%1029*::*nat* × (*nat* ⇒ *nat*). ∃ (*i*::*nat*) *sigma*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1029* (*IN i* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*))) ∧ *permutes sigma* (*dotdot* (*0*::*nat*) *k*)) (*i*, *sigma*))) (*GSPEC* (λ*GEN%PVAR%1030*::*nat* ⇒ *nat*. ∃ *p*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1030* (*permutes p* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*)))) *p*)) ∧ (∀ (*ul*::*?'a*::*type list*) (*i*::*nat*) (*sigma*::*nat* ⇒ *nat*) *j*::*nat*. *length ul* = *k* +

(*2*::*nat*) ∧ *j* ≤ *k* ∧ *IN i* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*))) ∧ *permutes sigma* (*dotdot* (*0*::*nat*) *k*) ⟶ *EL j* (*left_action_list* (*g* (*i*, *sigma*)) *ul*) = *EL j* (*left_action_list sigma* (*DROP ul i*)))

thm Rogers.IVFICRK_real3:

∀ *k*::*nat*. ∃ *g*::*nat* × (*nat* ⇒ *nat*) ⇒ *nat* ⇒ *nat*. *BIJ g* (*GSPEC* (λ*GEN%PVAR%1031*::*nat* × (*nat* ⇒ *nat*). ∃ (*i*::*nat*) *sigma*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1031* (*IN i* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*))) ∧ *permutes sigma* (*dotdot* (*0*::*nat*) *k*)) (*i*, *sigma*))) (*GSPEC* (λ*GEN%PVAR%1032*::*nat* ⇒ *nat*. ∃ *p*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1032* (*permutes p* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*)))) *p*)) ∧ (∀ (*ul*::(*real*, *3*) *cart list*) (*i*::*nat*) (*sigma*::*nat* ⇒ *nat*) *j*::*nat*. *length ul* = *k* + (*2*::*nat*) ∧ *j* ≤ *k* ∧ *IN i* (*dotdot* (*0*::*nat*) (*k* + (*1*::*nat*))) ∧ *permutes sigma* (*dotdot* (*0*::*nat*) *k*) ⟶ *EL j* (*left_action_list* (*g* (*i*, *sigma*)) *ul*) = *EL j* (*left_action_list sigma* (*DROP ul i*)))

thm Rogers.WQPRRDY:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *IN ul* (*barV V k*) ∧ *hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*)) ⟶ *hull convex* (*set_of_list ul*) = *UNIONS* (*GSPEC* (λ*GEN%PVAR%1035*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *p*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1035* (*permutes p* (*dotdot* (*0*::*nat*) *k*)) (*rogers V* (*left_action_list p ul*))))

thm DEF_weakly_saturated:

weakly_saturated = (λ(*_3450731*::(*real*, *3*) *cart* ⇒ *bool*) (*_3450732*::*real*) *_3450733*::*real*. ∀ *v*::(*real*, *3*) *cart*. *real_of_nat* (*2*::*nat*) ≤ *distance* (*vec* (*0*::*nat*), *v*) ∧ *distance* (*vec* (*0*::*nat*), *v*) ≤ *_3450733* ⟶ (∃ *u*::(*real*, *3*) *cart*. *IN u _3450731* ∧ *vec* (*0*::*nat*) ≠ *u* ∧ *distance* (*u*, *v*) < *_3450732*))

thm Tarjjuw.weakly_saturated:

∀ (*r'*::*real*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *r*::*real*. *weakly_saturated V r r'* = (∀ *v*::(*real*, *3*) *cart*. *real_of_nat* (*2*::*nat*) ≤ *distance* (*vec* (*0*::*nat*), *v*) ∧ *distance* (*vec* (*0*::*nat*), *v*) ≤ *r'* ⟶ (∃ *u*::(*real*, *3*) *cart*. *IN u V* ∧ *vec* (*0*::*nat*) ≠ *u* ∧ *distance* (*u*, *v*) < *r*))

thm DEF_half_spaces:

half_spaces = (λ(*_3450752*::(*real*, *3*) *cart*) *_3450753*::*real*. *GSPEC* (λ*GEN%PVAR%1036*::(*real*, *3*) *cart*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1036* (*dot _3450752 x* ≤ *_3450753*) *x*))

thm Tarjjuw.half_spaces:

∀ (*a*::(*real*, *3*) *cart*) *b*::*real*. *half_spaces a b* = *GSPEC* (λ*GEN%PVAR%1036*::(*real*, *3*) *cart*. ∃ *x*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1036* (*dot a x* ≤ *b*) *x*)

thm Tarjjuw.CHANGE_TARJJUW_1:

∀ (*v*::(*real*, *3*) *cart*) (*r'*::*real*) *p*::(*real*, *3*) *cart*. (*0*::*real*) < *r'* ∧ *p* ≠ *vec* (*0*::*nat*) ∧ *v* = % (*r'* / *vector_norm p*) *p* ⟶ *r'* = *vector_norm v*

thm Tarjjuw.CHANGE_TARJJUW_11:

$\forall (v::(real, 3)\ cart)\ (r'::real)\ p::(real, 3)\ cart.\ (0::real) < r' \wedge p \neq vec\ (0::nat) \wedge v = \%\ (r'\ /\ vector\_norm\ p)\ p \longrightarrow vector\_norm\ (\%\ (r'\ /\ vector\_norm\ p)\ p) = r'$

thm Tarjjuw.CHANGE_TARJJUW_12:

$\forall (v::(real, 3)\ cart)\ (r'::real)\ p::(real, 3)\ cart.\ (0::real) < r' \wedge p \neq vec\ (0::nat) \longrightarrow vector\_norm\ (\%\ (r'\ /\ vector\_norm\ p)\ p) = r'$

thm Tarjjuw.CHANGE_TARJJUW_2:

$\forall (v::(real, 3)\ cart)\ (r'::real)\ p::(real, 3)\ cart.\ p \neq vec\ (0::nat) \wedge v = \%\ (r'\ /\ vector\_norm\ p)\ p \longrightarrow \%\ r'\ p = \%\ (vector\_norm\ p)\ v$

thm Tarjjuw.CHANGE_TARJJUW_3:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ u::(real, 3)\ cart.\ packing\ V \wedge SUBSET\ V\ (DIFF\ HOL\_Light\_Import.UNIV\ (ball\ (vec\ (0::nat),\ real\_of\_nat\ (2::nat)))) \wedge IN\ u\ V \longrightarrow real\_of\_nat\ (2::nat) \leq vector\_norm\ u$

thm Tarjjuw.CHANGE_TARJJUW_31:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ u::(real, 3)\ cart.\ packing\ V \wedge SUBSET\ V\ (DIFF\ HOL\_Light\_Import.UNIV\ (ball\ (vec\ (0::nat),\ real\_of\_nat\ (2::nat)))) \wedge IN\ u\ V \longrightarrow u \neq vec\ (0::nat)$

thm Tarjjuw.CHANGE_TARJJUW_32:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ packing\ V \wedge IN\ (vec\ (0::nat))\ V \wedge IN\ u\ V \wedge vec\ (0::nat) \neq u \longrightarrow real\_of\_nat\ (2::nat) \leq vector\_norm\ u$

thm Tarjjuw.CHANGE_TARJJUW_4:

$\forall (u::(real, 3)\ cart)\ (v::(real, 3)\ cart)\ r::real.\ distance\ (u,\ v) < r \longrightarrow (distance\ (u,\ v))^2 < r^2$

thm Tarjjuw.CHANGE_TARJJUW_5:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (g::(real, 3)\ cart \Rightarrow real)\ (r::real)\ (r'::real)\ (u::(real, 3)\ cart)\ (v::(real, 3)\ cart)\ p::(real, 3)\ cart.\ packing\ V \wedge SUBSET\ V\ (DIFF\ HOL\_Light\_Import.UNIV\ (ball\ (vec\ (0::nat),\ real\_of\_nat\ (2::nat)))) \wedge real\_of\_nat\ (2::nat) \leq r \wedge r \leq r' \wedge p \neq vec\ (0::nat) \wedge g\ u * r'\ /\ real\_of\_nat\ (2::nat) < vector\_norm\ p \wedge v = \%\ (r'\ /\ vector\_norm\ p)\ p \wedge dot\ u\ p \leq g\ u \wedge vec\ (0::nat) \neq u \wedge distance\ (u,\ v) < r \wedge IN\ u\ V \longrightarrow vector\_norm\ p < vector\_norm\ p$

thm Tarjjuw.CHANGE_TARJJUW_6:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (P::(real, 3)\ cart \Rightarrow bool)\ (g::(real, 3)\ cart \Rightarrow real)\ (p::(real, 3)\ cart)\ (r::real)\ r'::real.\ real\_of\_nat\ (2::nat) \leq r \wedge r \leq r' \wedge SUBSET\ V\ (DIFF\ HOL\_Light\_Import.UNIV\ (ball\ (vec\ (0::nat),\ real\_of\_nat\ (2::nat)))) \wedge FINITE\ V \wedge packing\ V \wedge weakly\_saturated\ V\ r\ r' \wedge P = INTERS\ (GSPEC\ (\lambda GEN\%PVAR\%1039::(real, 3)\ cart \Rightarrow bool.\ \exists u::(real, 3)$

cart. SETSPEC GEN%PVAR%1039 (IN u V) (half_spaces u (g u)))) ∧ IN p P ⟶ (∀u::(real, 3) cart. IN u V ⟶ IN p (half_spaces u (g u)))

thm Tarjjuw.CHANGE_TARJJUW_7:

∀(V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) (u::(real, 3) cart) (r::real) r′::real. real_of_nat (2::nat) ≤ r ∧ r ≤ r′ ∧ SUBSET V (DIFF HOL_Light_Import.UNIV (ball (vec (0::nat), real_of_nat (2::nat)))) ∧ FINITE V ∧ packing V ∧ weakly_saturated V r r′ ∧ P = INTERS (GSPEC (λGEN%PVAR%1040::(real, 3) cart ⇒ bool. ∃u::(real, 3) cart. SETSPEC GEN%PVAR%1040 (IN u V) (half_spaces u (g u)))) ∧ IN u V ⟶ (∀p::(real, 3) cart. IN p P ⟶ dot u p ≤ g u)

thm Tarjjuw.CHANGE_TARJJUW_71:

∀(V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) (r::real) r′::real. real_of_nat (2::nat) ≤ r ∧ r ≤ r′ ∧ SUBSET V (DIFF HOL_Light_Import.UNIV (ball (vec (0::nat), real_of_nat (2::nat)))) ∧ FINITE V ∧ packing V ∧ weakly_saturated V r r′ ∧ P = INTERS (GSPEC (λGEN%PVAR%1041::(real, 3) cart ⇒ bool. ∃u::(real, 3) cart. SETSPEC GEN%PVAR%1041 (IN u V) (half_spaces u (g u)))) ⟶ (∀(p::(real, 3) cart) u::(real, 3) cart. IN p P ∧ IN u V ⟶ dot u p ≤ g u)

thm Tarjjuw.CHANGE_TARJJUW_8:

∀(g::(real, 3) cart ⇒ real) (r′::real) u::(real, 3) cart. real_of_nat (2::nat) ≤ (?r::real) ∧ ?r ≤ r′ ∧ (0::real) ≤ g u ⟶ (0::real) ≤ g u ∗ r′ / real_of_nat (2::nat)

thm Tarjjuw.FININTE_GFUN:

∀(V::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) r′::real. FINITE V ∧ V ≠ EMPTY ⟶ FINITE (GSPEC (λGEN%PVAR%1043::real. ∃u::(real, 3) cart. SETSPEC GEN%PVAR%1043 (IN u V) (g u ∗ r′ / real_of_nat (2::nat)))) ∧ GSPEC (λGEN%PVAR%1044::real. ∃u::(real, 3) cart. SETSPEC GEN%PVAR%1044 (IN u V) (g u ∗ r′ / real_of_nat (2::nat))) ≠ EMPTY

thm Tarjjuw.CHANGE_TARJJUW_9:

∀(V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) (r::real) r′::real. real_of_nat (2::nat) ≤ r ∧ r ≤ r′ ∧ SUBSET V (DIFF HOL_Light_Import.UNIV (ball (vec (0::nat), real_of_nat (2::nat)))) ∧ FINITE V ∧ packing V ∧ V ≠ EMPTY ∧ weakly_saturated V r r′ ∧ P = INTERS (GSPEC (λGEN%PVAR%1047::(real, 3) cart ⇒ bool. ∃u::(real, 3) cart. SETSPEC GEN%PVAR%1047 (IN u V) (half_spaces u (g u)))) ⟶ polyhedron P

thm Tarjjuw.CHANGE_TARJJUW_10:

∀(V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) (r::real) (r′::real) u::(real, 3) cart. real_of_nat (2::nat) ≤ r ∧ r ≤ r′ ∧

SUBSET V (DIFF HOL_Light_Import.UNIV (ball (vec (0::nat), real_of_nat (2::nat)))) ∧ FINITE V ∧ packing V ∧ V ≠ EMPTY ∧ weakly_saturated V r r′ ∧ P = INTERS (GSPEC (λGEN%PVAR%1062::(real, 3) cart ⇒ bool. ∃ u::(real, 3) cart. SETSPEC GEN%PVAR%1062 (IN u V) (half_spaces u (g u)))) ∧ polyhedron P ∧ IN u V ⟶ bounded P

thm Tarjjuw.TARJJUW:

∀ (V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (g::(real, 3) cart ⇒ real) (r::real) r′::real. real_of_nat (2::nat) ≤ r ∧ r ≤ r′ ∧ SUBSET V (DIFF HOL_Light_Import.UNIV (ball (vec (0::nat), real_of_nat (2::nat)))) ∧ FINITE V ∧ packing V ∧ V ≠ EMPTY ∧ weakly_saturated V r r′ ∧ P = INTERS (GSPEC (λGEN%PVAR%1063::(real, 3) cart ⇒ bool. ∃ u::(real, 3) cart. SETSPEC GEN%PVAR%1063 (IN u V) (half_spaces u (g u)))) ⟶ bounded P

thm Marchal_cells.TRUNCATE_SIMPLEX_GENERAL_0:

∀ xl::(real, 3) cart list. xl ≠ [] ⟶ truncate_simplex (0::nat) xl = [hd xl]

thm Marchal_cells.TRUNCATE_SIMPLEX_EXPLICIT_0:

∀ (u0::(real, 3) cart) (u1::(real, 3) cart) (u2::(real, 3) cart) u3::(real, 3) cart. truncate_simplex (0::nat) [u0] = [u0] ∧ truncate_simplex (0::nat) [u0, u1] = [u0] ∧ truncate_simplex (0::nat) [u0, u1, u2] = [u0] ∧ truncate_simplex (0::nat) [u0, u1, u2, u3] = [u0]

thm Marchal_cells.TRUNCATE_SIMPLEX_GENERAL_1:

∀ (ul::(real, 3) cart list) (vl::(real, 3) cart list) (a::(real, 3) cart) b::(real, 3) cart. ul = [a, b] @ vl ⟶ truncate_simplex (1::nat) ul = [a, b]

thm Marchal_cells.TRUNCATE_SIMPLEX_EXPLICIT_1:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. truncate_simplex (1::nat) [a, b] = [a, b] ∧ truncate_simplex (1::nat) [a, b, c] = [a, b] ∧ truncate_simplex (1::nat) [a, b, c, d] = [a, b]

thm Marchal_cells.TRUNCATE_SIMPLEX_GENERAL_2:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) (ul::(real, 3) cart list) vl::(real, 3) cart list. ul = [a, b, c] @ vl ⟶ truncate_simplex (2::nat) ul = [a, b, c]

thm Marchal_cells.TRUNCATE_SIMPLEX_EXPLICIT_2:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. truncate_simplex (2::nat) [a, b, c] = [a, b, c] ∧ truncate_simplex (2::nat) [a, b, c, d] = [a, b, c]

thm Marchal_cells.TRUNCATE_SIMPLEX_EXPLICIT_3:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. truncate_simplex (3::nat) [a, b, c, d] = [a, b, c, d]

thm Marchal_cells.OMEGA_LIST_TRUNCATE_0:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ omega\_list\_n\ V\ [u0,\ u1,\ u2,\ u3]\ (0::nat) = omega\_list\ V\ [u0]$

thm Marchal_cells.OMEGA_LIST_TRUNCATE_1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ omega\_list\_n\ V\ [u0,\ u1,\ u2,\ u3]\ (1::nat) = omega\_list\ V\ [u0,\ u1]$

thm Marchal_cells.OMEGA_LIST_TRUNCATE_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ omega\_list\_n\ V\ [u0,\ u1,\ u2,\ u3]\ (2::nat) = omega\_list\ V\ [u0,\ u1,\ u2]$

thm Marchal_cells.OMEGA_LIST_0_EXPLICIT:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \wedge ul = [a,\ b,\ c,\ ?d::(real,\ 3)\ cart] \longrightarrow omega\_list\_n\ V\ ul\ (0::nat) = a$

thm Marchal_cells.OMEGA_LIST_1_EXPLICIT:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ (d::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \wedge ul = [a,\ b,\ c,\ d] \longrightarrow omega\_list\_n\ V\ ul\ (1::nat) = circumcenter\ (INSERT\ a\ (INSERT\ b\ EMPTY))$

thm Marchal_cells.OMEGA_LIST_2_EXPLICIT:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ (d::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \wedge ul = [a,\ b,\ c,\ d] \longrightarrow omega\_list\_n\ V\ ul\ (2::nat) = circumcenter\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))$

thm Marchal_cells.OMEGA_LIST_3_EXPLICIT:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ (d::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge hl\ ul < sqrt\ (real\_of\_nat\ (2::nat)) \wedge ul = [a,\ b,\ c,\ d] \longrightarrow omega\_list\_n\ V\ ul\ (3::nat) = circumcenter\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))))$

thm Marchal_cells.BARV_3_EXPLICIT:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ vl::(real,\ 3)\ cart\ list.\ barV\ V\ (3::nat)\ vl \longrightarrow (\exists (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ vl = [u0,\ u1,\ u2,\ u3])$

thm Marchal_cells.BARV_K_EXPLICIT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) *d*::(*real*, *3*) *cart*. *barV V* (*3*::*nat*) [*a*, *b*, *c*, *d*] ⟶ *barV V* (*2*::*nat*) [*a*, *b*, *c*] ∧ *barV V* (*1*::*nat*) [*a*, *b*] ∧ *barV V* (*0*::*nat*) [*a*]

thm Marchal_cells.AFF_DIM_LE_LENGTH:

∀ (*xl*::(*real*, *3*) *cart list*) *n*::*nat*. *length xl* = *n* ⟶ *aff_dim* (*set_of_list xl*) < *int n*

thm Marchal_cells.CONVEX_HULL_SUBSET:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *S'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET S S'* ⟶ *SUBSET* (*hull convex S*) (*hull convex S'*)

thm Marchal_cells.BALL_CONVEX_HULL_LEMMA:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *x*::(*real*, *?'a*::*type*) *cart*. (∀ *x*::(*real*, *?'a*::*type*) *cart*. *S x* ⟶ *distance* (*s*, *x*) < *r*) ⟶ *hull convex S x* ⟶ *distance* (*s*, *x*) < *r*

thm Marchal_cells.CONVEX_RCONE_GT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) ≤ *r* ⟶ *convex* (*rcone_gt a b r*)

thm Marchal_cells.RCONE_GT_CONVEX_HULL_LEMMA:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET s* (*rcone_gt a b r*) ∧ (*0*::*real*) ≤ *r* ⟶ *SUBSET* (*hull convex s*) (*rcone_gt a b r*)

thm Upfzbzm_support_lemmas.tau0_not_zero:

tau0 ≠ (*0*::*real*)

thm Upfzbzm_support_lemmas.ZERO_LT_MM2_LEMMA:

(*0*::*real*) < *mm2*

thm Upfzbzm_support_lemmas.FINITE_PERMUTE_3:

FINITE (*GSPEC* (λ*GEN%PVAR%1067*::*nat* ⇒ *nat*. ∃ *p*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1067* (*permutes p* (*INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) *EMPTY*)))) *p*))

thm Upfzbzm_support_lemmas.FINITE_PERMUTE_4:

FINITE (*GSPEC* (λ*GEN%PVAR%1068*::*nat* ⇒ *nat*. ∃ *p*::*nat* ⇒ *nat*. *SETSPEC GEN%PVAR%1068* (*permutes p* (*INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) (*INSERT* (*3*::*nat*) *EMPTY*))))) *p*))

thm Upfzbzm_support_lemmas.john_harrison_lemma1:

GABS (λ*f*::*?'c*::*type* × *?'b*::*type* ⇒ *?'a*::*type*. ∀ (*x*::*?'c*::*type*) *y*::*?'b*::*type*. *GEQ* (*f* (*x*, *y*)) ((*?P*::*?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type*) *x y*)) = (λ*p*::*?'c*::*type* × *?'b*::*type*. *?P* (*fst p*) (*snd p*))

thm Upfzbzm_support_lemmas.john_harrison_lemma2:

$(\exists\, x{::}?'a{::}type.\ (?P{::}?'a{::}type \Rightarrow bool)\ x) \wedge (SOME\, x{::}?'a{::}type.\ ?P\ x) = (?a{::}?'a{::}type) \longrightarrow ?P\ ?a$

thm Upfzbzm_support_lemmas.JOHN_SELECT_THM:

$(\exists\, (x{::}?'b{::}type)\ y{::}?'a{::}type.\ (?P{::}?'b{::}type \Rightarrow bool)\ x \wedge (?Q{::}?'a{::}type \Rightarrow bool)\ y \wedge (?R{::}?'b{::}type \Rightarrow ?'a{::}type \Rightarrow bool)\ x\ y) \wedge Eps\ (GABS\ (\lambda f{::}?'b{::}type \times ?'a{::}type \Rightarrow bool.\ \forall\, (x{::}?'b{::}type)\ y{::}?'a{::}type.\ GEQ\ (f\ (x,\ y))\ (?P\ x \wedge ?Q\ y \wedge ?R\ x\ y))) = (?a{::}?'b{::}type,\ ?b{::}?'a{::}type) \longrightarrow ?P\ ?a \wedge ?Q\ ?b \wedge ?R\ ?a\ ?b$

thm Upfzbzm_support_lemmas.SQRT_OF_32_lemma:

$sqrt\ (real\_of\_nat\ (32{::}nat)) = real\_of\_nat\ (8{::}nat) * sqrt\ ((1{::}real)\ /\ real\_of\_nat\ (2{::}nat))$

thm Upfzbzm_support_lemmas.m1_minus_12m2:

$mm1 - real\_of\_nat\ (12{::}nat) * mm2 = sqrt\ ((1{::}real)\ /\ real\_of\_nat\ (2{::}nat))$

thm Upfzbzm_support_lemmas.ZERO_LE_MM2_LEMMA:

$(0{::}real) \leq mm2$

thm Upfzbzm_support_lemmas.FINITE_edgeX:

$\forall\, (V{::}(real,\ 3)\ cart \Rightarrow bool)\ X{::}(real,\ 3)\ cart \Rightarrow bool.\ FINITE\ (edgeX\ V\ X)$

thm Upfzbzm_support_lemmas.FINITE_critical_edgeX:

$\forall\, (V{::}(real,\ 3)\ cart \Rightarrow bool)\ X{::}(real,\ 3)\ cart \Rightarrow bool.\ FINITE\ (critical\_edgeX\ V\ X)$

thm Upfzbzm_support_lemmas.DIHV_LE_0:

$\forall\, (x{::}(real,\ ?'a{::}type)\ cart)\ (y{::}(real,\ ?'a{::}type)\ cart)\ (z{::}(real,\ ?'a{::}type)\ cart)\ t{::}(real,\ ?'a{::}type)\ cart.\ (0{::}real) \leq dihV\ x\ y\ z\ t$

thm Marchal_cells_2.DIHV_SYM:

$\forall\, (x{::}(real,\ ?'a{::}type)\ cart)\ (y{::}(real,\ ?'a{::}type)\ cart)\ (z{::}(real,\ ?'a{::}type)\ cart)\ t{::}(real,\ ?'a{::}type)\ cart.\ dihV\ x\ y\ z\ t = dihV\ y\ x\ z\ t$

thm Upfzbzm_support_lemmas.DIHX_POS:

$\forall\, (u{::}(real,\ 3)\ cart)\ (v{::}(real,\ 3)\ cart)\ (V{::}(real,\ 3)\ cart \Rightarrow bool)\ X{::}(real,\ 3)\ cart \Rightarrow bool.\ (0{::}real) \leq dihX\ V\ X\ (u,\ v)$

thm Upfzbzm_support_lemmas.SUM_SET_OF_2_ELEMENTS:

$\forall\, (s{::}?'a{::}type)\ (t{::}?'a{::}type)\ f{::}?'a{::}type \Rightarrow real.\ s \neq t \longrightarrow sum\ (INSERT\ s\ (INSERT\ t\ EMPTY))\ f = f\ s + f\ t$

thm Upfzbzm_support_lemmas.pos_lemma:

$\forall\, Q{::}real \Rightarrow real.\ (\exists\, C{\geq}0{::}real.\ \forall\, r{\geq}1{::}real.\ Q\ r \leq C * r^2) = (\exists\, C{::}real.\ \forall\, r{\geq}1{::}real.\ Q\ r \leq C * r^2)$

thm Upfzbzm_support_lemmas.negligible_fun_any_C:

$\forall$ (*f*::(*real*, *3*) *cart* $\Rightarrow$ *real*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *negligible_fun_0 f S* = ($\exists$ *C*::*real*. $\forall$ *r*$\geq$*1*::*real*. *sum* (*HOL_Light_Import*.*INTER S* (*ball* (*vec* (*0*::*nat*), *r*))) *f* $\leq$ *C* $*$ r^2)

thm Emnwuus.EMNWUUS1:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* $\wedge$ *packing V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\longrightarrow$ (*hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*))) = (*mcell4 V ul* $\neq$ *EMPTY*)

thm Emnwuus.EMNWUUS2:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* $\wedge$ *packing V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\longrightarrow$ (*hl ul* < *sqrt* (*real_of_nat* (*2*::*nat*))) = (*mcell0 V ul* = *EMPTY* $\wedge$ *mcell1 V ul* = *EMPTY* $\wedge$ *mcell2 V ul* = *EMPTY* $\wedge$ *mcell3 V ul* = *EMPTY*)

thm Marchal_cells_2.AFF_GE_2_2:

$\forall$ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w* (*INSERT* (*?z*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) $\longrightarrow$ *aff_ge* (*INSERT x* (*INSERT v EMPTY*)) (*INSERT w* (*INSERT ?z EMPTY*)) = *GSPEC* (λ*GEN%PVAR%1073*::(*real*, *?'a*::*type*) *cart*. $\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1073* ($\exists$ (*t1*::*real*) (*t2*::*real*) (*t3*::*real*) *t4*::*real*. (*0*::*real*) $\leq$ *t3* $\wedge$ (*0*::*real*) $\leq$ *t4* $\wedge$ *t1* + (*t2* + (*t3* + *t4*)) = (*1*::*real*) $\wedge$ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 v*) (*vector_add* (% *t3 w*) (% *t4 ?z*)))) *y*)

thm Marchal_cells_2.MEASURABLE_ROGERS:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*?'a*::*type*. *saturated V* $\wedge$ *packing V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\longrightarrow$ *measurable* (*rogers V ul*)

thm Marchal_cells_2_new.CONVEX_RCONE_GE:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *r*::*real*. (*0*::*real*) $\leq$ *r* $\longrightarrow$ *convex* (*rcone_ge a b r*)

thm Marchal_cells_2.FINITE_PERMUTE_3:

FINITE (*GSPEC* (λ*GEN%PVAR%1074*::*nat* $\Rightarrow$ *nat*. $\exists$ *p*::*nat* $\Rightarrow$ *nat*. *SETSPEC GEN%PVAR%1074* (*permutes p* (*INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) *EMPTY*)))) *p*))

thm Marchal_cells_2.FINITE_PERMUTE_4:

FINITE (*GSPEC* (λ*GEN%PVAR%1075*::*nat* $\Rightarrow$ *nat*. $\exists$ *p*::*nat* $\Rightarrow$ *nat*. *SETSPEC GEN%PVAR%1075* (*permutes p* (*INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) (*INSERT* (*3*::*nat*) *EMPTY*))))) *p*))

thm Marchal_cells_2.DISJOINT_KY_LEMMA:

(*?x*::(*real*, *3*) *cart*) ≠ (*?y*::(*real*, *3*) *cart*) ∧ *?x* ≠ (*?z*::(*real*, *3*) *cart*) ⟶ *DISJOINT* (*INSERT ?x EMPTY*) (*INSERT ?y* (*INSERT ?z EMPTY*))

thm Marchal_cells_2_new.RCONE_GT_SUBSET_RCONE_GE:

∀ (*z*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *h*::*real*. *SUBSET* (*rcone_gt z w h*) (*rcone_ge z w h*)

thm Marchal_cells_2_new.MCELL_EXPLICIT:

∀ (*k*::*nat*) (*ul*::(*real*, *3*) *cart list*) *V*::(*real*, *3*) *cart* ⇒ *bool*. *mcell* (*0*::*nat*) *V ul* = *mcell0 V ul* ∧ *mcell* (*1*::*nat*) *V ul* = *mcell1 V ul* ∧ *mcell* (*2*::*nat*) *V ul* = *mcell2 V ul* ∧ *mcell* (*3*::*nat*) *V ul* = *mcell3 V ul* ∧ ((*4*::*nat*) ≤ *k* ⟶ *mcell k V ul* = *mcell4 V ul*)

thm Marchal_cells_2.EVENTUALLY_RADIAL_EMPTY:

∀ *v*::(*real*, *3*) *cart*. *eventually_radial v EMPTY*

thm Marchal_cells_2.EVENTUALLY_RADIAL_NOT_IN_CLOSED_SET:

∀ (*v*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* ⇒ *bool*. ¬ *S v* ∧ *HOL_Light_Import.closed S* ⟶ *eventually_radial v S*

thm Marchal_cells_2.CLOSED_CONVEX_HULL_FINITE:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ⟶ *HOL_Light_Import.closed* (*hull convex s*)

thm Marchal_cells_2_new.CLOSED_ROGERS:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ *HOL_Light_Import.closed* (*rogers V ul*)

thm Marchal_cells_2.CLOSED_SET_OF_LIST_KY_LEMMA_1:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ *HOL_Light_Import.closed* (*hull convex* (*HOL_Light_Import.UNION* (*set_of_list* (*truncate_simplex* (*2*::*nat*) *ul*)) (*INSERT* (*mxi V ul*) *EMPTY*)))

thm Marchal_cells_2.CLOSED_SET_OF_LIST_KY_LEMMA_2:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ *HOL_Light_Import.closed* (*hull convex* (*set_of_list ul*))

thm Marchal_cells_2.CLOSED_RCONE_GE:

∀ (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) *a*::*real*. (*0*::*real*) < *a* ⟶ *HOL_Light_Import.closed* (*rcone_ge v0 v1 a*)

thm Marchal_cells_2_new.BARV_IMP_HL_1_POS_LT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* ∧ *packing V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ (*0*::*real*) < *hl* (*truncate_simplex* (*1*::*nat*) *ul*)

thm Marchal_cells_2.CLOSED_MCELL:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (k::nat)\ v::(real,\ 3)\ cart.$ $saturated\ V \land packing\ V \land barV\ V\ (3::nat)\ ul \longrightarrow HOL\_Light\_Import.closed$ $(mcell\ k\ V\ ul)$

thm Marchal_cells_2_new.BARV_IMP_u0_IN_V:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (u0::(real,\ 3)\ cart)\ (u1::(real,$ $3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ saturated\ V \land packing\ V \land barV$ $V\ (3::nat)\ ul \land ul = [u0,\ u1,\ u2,\ u3] \longrightarrow IN\ u0\ V$

thm Marchal_cells_2_new.ROGERS_INTER_V_LEMMA:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ v::(real,\ 3)\ cart.\ saturated$ $V \land packing\ V \land barV\ V\ (3::nat)\ ul \land IN\ v\ V \land rogers\ V\ ul\ v \longrightarrow v = hd\ ul$

thm Marchal_cells_2.CONVEX_HULL_4:

$hull\ convex\ (INSERT\ (?a::(real,\ ?'a::type)\ cart)\ (INSERT\ (?b::(real,\ ?'a::type)$ $cart)\ (INSERT\ (?c::(real,\ ?'a::type)\ cart)\ (INSERT\ (?d::(real,\ ?'a::type)\ cart)$ $EMPTY)))) = GSPEC\ (\lambda GEN\%PVAR\%1083::(real,\ ?'a::type)\ cart.\ \exists (u::real)$ $(v::real)\ (w::real)\ z::real.\ SETSPEC\ GEN\%PVAR\%1083\ ((0::real) \le u \land (0::real)$ $\le v \land (0::real) \le w \land (0::real) \le z \land u + (v + (w + z)) = (1::real))$ $(vector\_add\ (\%\ u\ ?a)\ (vector\_add\ (\%\ v\ ?b)\ (vector\_add\ (\%\ w\ ?c)\ (\%\ z\ ?d)))))$

thm Marchal_cells_2.REAL_LE_DIV_SIMPLIFY_KY_LEMMA:

$\forall (a::real)\ (b::real)\ c::real.\ (0::real) < a \land b \le c\ /\ a \longrightarrow a * b \le c$

thm Marchal_cells_2.EVENTUALLY_RADIAL_CONVEX_HULL_4_sub1:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ d::(real,\ 3)\ cart.\ \neg\ IN\ a$ $(hull\ convex\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \longrightarrow eventually\_radial$ $a\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))))$

thm Marchal_cells_2.U0_NOT_IN_CONVEX_HULL_FROM_ROGERS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \land packing\ V \land$ $barV\ V\ (3::nat)\ ul \longrightarrow \neg\ IN\ (hd\ ul)\ (hull\ convex\ (INSERT\ (omega\_list\_n\ V$ $ul\ (1::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (2::nat))\ (INSERT\ (omega\_list\_n\ V$ $ul\ (3::nat))\ EMPTY))))$

thm Marchal_cells_2_new.RADIAL_VS_RADIAL_NORM:

$\forall (x::(real,\ 3)\ cart)\ (r::real)\ C::(real,\ 3)\ cart \Rightarrow bool.\ radial\ r\ x\ C = radial\_norm$ $r\ x\ C$

thm Marchal_cells_2.EVENTUALLY_RADIAL_INTER:

$\forall (x::(real,\ 3)\ cart)\ (C::(real,\ 3)\ cart \Rightarrow bool)\ C'::(real,\ 3)\ cart \Rightarrow bool.\ eventually\_radial$ $x\ C \land eventually\_radial\ x\ C' \longrightarrow eventually\_radial\ x\ (HOL\_Light\_Import.INTER$ $C\ C')$

thm Marchal_cells_2.SET_EQ_LEMMA:

$((?A::?'a::type \Rightarrow bool) = (?B::?'a::type \Rightarrow bool)) = (\forall x::?'a::type.\ (IN\ x\ ?A$ $\longrightarrow IN\ x\ ?B) \land (IN\ x\ ?B \longrightarrow IN\ x\ ?A))$

thm Marchal_cells_2.SET_OF_0_TO_3:

GSPEC (λ*GEN%PVAR%1093*::*nat*. $\exists$ *j*::*nat*. *SETSPEC GEN%PVAR%1093* (*j* < (*4*::*nat*)) *j*) = *INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) (*INSERT* (*3*::*nat*) *EMPTY*)))

thm Marchal_cells_2.SET_OF_0_TO_2:

GSPEC (λ*GEN%PVAR%1094*::*nat*. $\exists$ *j*::*nat*. *SETSPEC GEN%PVAR%1094* (*j* $\leq$ (*2*::*nat*)) *j*) = *INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) *EMPTY*))

thm Marchal_cells_2.ZERO_LT_SQRT_2:

(*1*::*real*) < *sqrt* (*real_of_nat* (*2*::*nat*))

thm Marchal_cells_2.RCONE_GE_TRANS:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*r*::*real*) (*x*::(*real*, *3*) *cart*) *t*::*real*. (*0*::*real*) $\leq$ *t* $\wedge$ *IN* (*vector_add a x*) (*rcone_ge a b r*) $\longrightarrow$ *IN* (*vector_add a* (% *t x*)) (*rcone_ge a b r*)

thm Marchal_cells_2.RCONE_GE_INTERS_PROJECTION_KY_LEMMA:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*r*::*real*) *x*::(*real*, *3*) *cart*. (*0*::*real*) < *r* $\wedge$ *r* < (*1*::*real*) $\wedge$ *a* $\neq$ *b* $\wedge$ *IN x* (*HOL_Light_Import.INTER* (*rcone_ge a b r*) (*rcone_ge b a r*)) $\longrightarrow$ ($\exists$ *s*::(*real*, *3*) *cart*. *IN s* (*hull convex* (*INSERT a* (*INSERT b EMPTY*))) $\wedge$ *dot* (*vector_sub x s*) (*vector_sub a b*) = (*0*::*real*))

thm Marchal_cells_2_new.RCONE_GE_INTER_VORONOI_CLOSED_PROJECTION_KY_LEMMA:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*r*::*real*) (*x*::(*real*, *3*) *cart*) *V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. (*0*::*real*) < *r* $\wedge$ *a* $\neq$ *b* $\wedge$ *IN a V* $\wedge$ *IN b V* $\wedge$ *IN x* (*HOL_Light_Import.INTER* (*rcone_ge a b r*) (*voronoi_closed V a*)) $\longrightarrow$ ($\exists$ *s*::(*real*, *3*) *cart*. *IN s* (*hull convex* (*INSERT a* (*INSERT b EMPTY*))) $\wedge$ *dot* (*vector_sub x s*) (*vector_sub a b*) = (*0*::*real*))

thm Marchal_cells_2.RCONEGE_INTER_VORONOI_CLOSED_IMP_RCONEGE:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*r*::*real*) *x*::(*real*, *3*) *cart*. *packing V* $\wedge$ *saturated V* $\wedge$ *IN a V* $\wedge$ *IN b V* $\wedge$ *a* $\neq$ *b* $\wedge$ (*0*::*real*) < *r* $\wedge$ *r* $\leq$ (*1*::*real*) $\wedge$ *IN x* (*rcone_ge a b r*) $\wedge$ *IN x* (*voronoi_closed V a*) $\longrightarrow$ *IN x* (*rcone_ge b a r*)

thm Marchal_cells_2.OMEGA_LIST_1_EXPLICIT_NEW:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* $\wedge$ *packing V* $\wedge$ *IN ul* (*barV V* (*3*::*nat*)) $\wedge$ *ul* = [*a*, *b*, *c*, *d*] $\wedge$ *hl* [*a*, *b*] < *sqrt* (*real_of_nat* (*2*::*nat*)) $\longrightarrow$ *omega_list_n V ul* (*1*::*nat*) = *circumcenter* (*INSERT a* (*INSERT b EMPTY*))

thm Marchal_cells_2.IN_SET_IMP_IN_CONVEX_HULL_SET:

$\forall$ (*a*::(*real*, *3*) *cart*) *S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *IN a S* $\longrightarrow$ *IN a* (*hull convex S*)

thm Marchal_cells_2_new.CONVEX_HULL_BREAK_KY_LEMMA:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *between x* (*a*, *b*) ⟶ *hull convex* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*)))) = *HOL_Light_Import.UNION* (*hull convex* (*INSERT a* (*INSERT x* (*INSERT c* (*INSERT d EMPTY*))))) (*hull convex* (*INSERT x* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*)))))

thm Marchal_cells_2.AFF_GE_BREAK_KY_LEMMA:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) *x*::(*real*, *3*) *cart*. *between x* (*c*, *d*) ∧ *DISJOINT* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT c* (*INSERT d EMPTY*)) ∧ *DISJOINT* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT c* (*INSERT x EMPTY*)) ∧ *DISJOINT* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT x* (*INSERT d EMPTY*)) ⟶ *aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT c* (*INSERT d EMPTY*)) = *HOL_Light_Import.UNION* (*aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT c* (*INSERT x EMPTY*))) (*aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT x* (*INSERT d EMPTY*)))

thm Marchal_cells_2_new.CONVEX_HULL_4_SUBSET_AFF_GE_2_2:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) *d*::(*real*, *3*) *cart*. *SUBSET* (*hull convex* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*))))) (*aff_ge* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT c* (*INSERT d EMPTY*)))

thm Marchal_cells_2.AFF_INDEPENDENT_SET_OF_LIST_BARV:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *packing V* ∧ *saturated V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ ¬ *affine_dependent* (*set_of_list ul*)

thm Marchal_cells_2_new.VORONOI_LIST_3_SINGLETON_EXPLICIT:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) *ul*::(*real*, *3*) *cart list*. *packing V* ∧ *saturated V* ∧ *barV V* (*3*::*nat*) *ul* ⟶ (∃ *a*::(*real*, *3*) *cart*. *voronoi_list V ul* = *INSERT a EMPTY* ∧ *a* = *circumcenter* (*set_of_list ul*) ∧ *hl ul* = *distance* (*hd ul*, *a*))

thm Marchal_cells_2.ORTHOGONAL_AFF_HULL_2_KY_LEMMA:

∀ (*n*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart*. *orthogonal* (*vector_sub a b*) *n* ∧ *IN s* (*aff* (*INSERT a* (*INSERT b EMPTY*))) ∧ *IN p* (*aff* (*INSERT a* (*INSERT b EMPTY*))) ⟶ *orthogonal* (*vector_sub s p*) *n*

thm Marchal_cells_2_new.DIST_PROJECTION_LT_LEMMA:

∀ (*x*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. ∃ *s*::(*real*, *3*) *cart*. *IN s* (*aff* (*INSERT a* (*INSERT b EMPTY*))) ∧ (∀ (*m*::(*real*, *3*) *cart*) *n*::(*real*, *3*) *cart*. *IN m* (*aff* (*INSERT a* (*INSERT b EMPTY*))) ∧ *IN n* (*aff* (*INSERT a* (*INSERT b EMPTY*))) ⟶ (*distance* (*x*, *m*) < *distance* (*x*, *n*)) = (*distance* (*s*, *m*) < *distance* (*s*, *n*)) ∧ (*distance* (*x*, *m*) ≤ *distance* (*x*, *n*)) = (*distance* (*s*, *m*) ≤ *distance* (*s*, *n*)))

thm Marchal_cells_2_new.SIMPLEX_FURTHEST_LT_2:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*hull convex s*) $\wedge$ $\neg$ *IN x s* $\longrightarrow$ ($\exists$ *y*::(*real*, *?'a*::*type*) *cart*. *IN y s* $\wedge$ *vector_norm* (*vector_sub x a*) < *vector_norm* (*vector_sub y a*)))

thm Marchal_cells_2.DIST_BETWEEN_FURTHEST_LT:

$\forall$ (*x*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart*. *between s* (*a*, *b*) $\wedge$ *s* $\neq$ *a* $\wedge$ *s* $\neq$ *b* $\wedge$ *a* $\neq$ *b* $\wedge$ *distance* (*x*, *b*) $\leq$ *distance* (*x*, *a*) $\longrightarrow$ *distance* (*x*, *s*) < *distance* (*x*, *a*)

thm Marchal_cells_2_new.ROGERS_EXPLICIT:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* $\wedge$ *packing V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\longrightarrow$ *rogers V ul* = *hull convex* (*INSERT* (*hd ul*) (*INSERT* (*omega_list_n V ul* (*1*::*nat*)) (*INSERT* (*omega_list_n V ul* (*2*::*nat*)) (*INSERT* (*omega_list_n V ul* (*3*::*nat*)) *EMPTY*))))

thm Marchal_cells_2_new.SEGMENT_INTER_CBALL_LEMMA:

$\forall$ (*x*::(*real*, *3*) *cart*) (*r*::*real*) (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *distance* (*x*, *a*) $\leq$ *r* $\wedge$ *r* $\leq$ *distance* (*x*, *b*) $\longrightarrow$ ($\exists$ *c*::(*real*, *3*) *cart*. *between c* (*a*, *b*) $\wedge$ *distance* (*x*, *c*) = *r*)

thm Marchal_cells_2.CLOSEST_POINT_SING:

$\forall$ (*a*::(*real*, *3*) *cart*) *b*::(*real*, *3*) *cart*. *closest_point* (*INSERT a EMPTY*) *b* = *a*

thm Marchal_cells_2.MXI_EXPLICIT:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *packing V* $\wedge$ *saturated V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\wedge$ *ul* = [*?u0.0*::(*real*, *3*) *cart*, *?u1.0*::(*real*, *3*) *cart*, *?u2.0*::(*real*, *3*) *cart*, *?u3.0*::(*real*, *3*) *cart*] $\wedge$ *hl* (*truncate_simplex* (*2*::*nat*) *ul*) < *sqrt* (*real_of_nat* (*2*::*nat*)) $\wedge$ *sqrt* (*real_of_nat* (*2*::*nat*)) $\leq$ *hl ul* $\longrightarrow$ ($\exists$ *s*::(*real*, *3*) *cart*. *between s* (*omega_list_n V ul* (*2*::*nat*), *omega_list_n V ul* (*3*::*nat*)) $\wedge$ *distance* (*?u0.0*, *s*) = *sqrt* (*real_of_nat* (*2*::*nat*)) $\wedge$ *mxi V ul* = *s*)

thm Marchal_cells_2_new.CONVEX_HULL_4_IMP_2_2:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart*. *IN p* (*hull convex* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*))))) $\longrightarrow$ ($\exists$ (*m*::(*real*, *3*) *cart*) *n*::(*real*, *3*) *cart*. *between p* (*m*, *n*) $\wedge$ *between m* (*a*, *b*) $\wedge$ *between n* (*c*, *d*))

thm DEF_proj_point:

proj_point = (λ(*_3527368*::(*real*, *?'a*::*type*) *cart*) *_3527369*::(*real*, *?'a*::*type*) *cart*. *vector_sub _3527369* (*projection _3527368 _3527369*))

thm Marchal_cells_2.proj_point:

$\forall$ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *proj_point e x* = *vector_sub x* (*projection e x*)

thm Marchal_cells_2.projection_proj_point:

∀ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *projection e x* = *vector_sub x* (*proj_point e x*)

thm Marchal_cells_2.PRO_EXP:

∀ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *proj_point e x* = % (*dot x e* / *dot e e*) *e*

thm Marchal_cells_2.BETWEEN_PROJ_POINT:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *e*::(*real*, *?'a*::*type*) *cart*. *between x* (*a*, *b*) ⟶ *between* (*proj_point e x*) (*proj_point e a*, *proj_point e b*)

thm Marchal_cells_2.PARALLEL_PROJECTION:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *between x* (*a*, *y*) ∧ *a* ≠ *b* ⟶ (∃ *k*≤*1*::*real*. (*0*::*real*) ≤ *k* ∧ *projection* (*vector_sub b a*) (*vector_sub x a*) = % *k* (*projection* (*vector_sub b a*) (*vector_sub y a*)))

thm Marchal_cells_2.NORM_PROJECTION_LE:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *between x* (*a*, *y*) ∧ *a* ≠ *b* ⟶ *vector_norm* (*projection* (*vector_sub b a*) (*vector_sub x a*)) ≤ *vector_norm* (*projection* (*vector_sub b a*) (*vector_sub y a*))

thm Marchal_cells_2.OMEGA_LIST_TRUNCATE_1_NEW1:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::(*real*, *3*) *cart*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*, *u2*] (*1*::*nat*) = *omega_list V* [*u0*, *u1*]

thm Marchal_cells_2.OMEGA_LIST_TRUNCATE_1_NEW2:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::*?'b*::*type*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*] (*1*::*nat*) = *omega_list V* [*u0*, *u1*]

thm Marchal_cells_2.OMEGA_LIST_TRUNCATE_2_NEW1:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::(*real*, *3*) *cart*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*, *u2*] (*2*::*nat*) = *omega_list V* [*u0*, *u1*, *u2*]

thm Marchal_cells_2.IN_AFFINE_KY_LEMMA1:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *IN x s* ⟶ *IN x* (*hull affine s*)

thm Marchal_cells_2_new.AFFINE_SUBSET_KY_LEMMA:

∀ (*S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *B*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *SUBSET S B* ⟶ *SUBSET* (*hull affine S*) (*hull affine B*)

thm Marchal_cells_2.TRANSLATE_AFFINE_KY_LEMMA1:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *k*::*real*. *IN a* (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *IN b* (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *IN c* (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\longrightarrow$ *IN* (*vector_add a* (% *k* (*vector_sub b c*))) (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))))

thm Marchal_cells_2_new.IN_AFFINE_HULL_KY_LEMMA3:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*p*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) *r*::*real*. *IN* (*vector_add p a*) (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *IN* (*vector_add p* (% *r a*)) (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *r* $\neq$ (*1*::*real*) $\longrightarrow$ *IN p* (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))))

thm Marchal_cells_2_new.IN_AFFINE_HULL_KY_LEMMA3_alt:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*p*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) *r*::*real*. *IN* (*vector_sub p a*) (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *IN* (*vector_sub p* (% *r a*)) (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*)))) $\wedge$ *r* $\neq$ (*1*::*real*) $\longrightarrow$ *IN p* (*hull affine* (*INSERT x* (*INSERT y* (*INSERT z EMPTY*))))

thm Marchal_cells_2.IN_AFFINE_HULL_3_KY_LEMMA2:

$\forall$ (*X*::(*real*, *3*) *cart*) (*Y*::(*real*, *3*) *cart*) (*Z*::(*real*, *3*) *cart*) (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) *c*::(*real*, *3*) *cart*. *IN X* (*hull affine* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))) $\wedge$ *IN Y* (*hull affine* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))) $\wedge$ *between Z* (*X*, *Y*) $\longrightarrow$ *IN Z* (*hull affine* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*))))

thm Marchal_cells_2.SUM_CLAUSES_alt:

$\forall$ (*x*::*?'a*::*type*) (*f*::*?'a*::*type* $\Rightarrow$ *real*) *s*::*?'a*::*type* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ *sum* (*INSERT x s*) *f* = (*if IN x s then sum s f else f x* + *sum s f*)

thm Marchal_cells_2_new.SUM_DIS4:

$\forall$ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) (*z*::*?'a*::*type*) (*t*::*?'a*::*type*) *f*::*?'a*::*type* $\Rightarrow$ *real*. *CARD* (*INSERT x* (*INSERT y* (*INSERT z* (*INSERT t EMPTY*)))) = (*4*::*nat*) $\longrightarrow$ *sum* (*INSERT x* (*INSERT y* (*INSERT z* (*INSERT t EMPTY*)))) *f* = *f x* + (*f y* + (*f z* + *f t*))

thm Marchal_cells_2.CARD4_IMP_DISTINCT:

$\forall$ (*a*::*?'a*::*type*) (*b*::*?'a*::*type*) (*c*::*?'a*::*type*) *d*::*?'a*::*type*. *CARD* (*INSERT a* (*INSERT b* (*INSERT c* (*INSERT d EMPTY*)))) = (*4*::*nat*) $\longrightarrow$ *a* $\neq$ *b*

thm Marchal_cells_2.VSUM_CLAUSES_alt:

$\forall$ (*x*::*?'b*::*type*) (*f*::*?'b*::*type* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) *s*::*?'b*::*type* $\Rightarrow$ *bool*. *FINITE s* $\longrightarrow$ *vsum* (*INSERT x s*) *f* = (*if IN x s then vsum s f else vector_add* (*f x*) (*vsum s f*))

thm Marchal_cells_2.VSUM_DIS4:

$\forall (x::?'b::type)\ (y::?'b::type)\ (z::?'b::type)\ (t::?'b::type)\ f::?'b::type \Rightarrow (real, ?'a::type)\ cart.\ CARD\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ (INSERT\ t\ EMPTY)))) = (4::nat) \longrightarrow vsum\ (INSERT\ x\ (INSERT\ y\ (INSERT\ z\ (INSERT\ t\ EMPTY))))\ f = vector\_add\ (f\ x)\ (vector\_add\ (f\ y)\ (vector\_add\ (f\ z)\ (f\ t)))$

thm Marchal_cells_2.AFFINE_DEPENDENT_KY_LEMMA1:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (c::(real, 3)\ cart)\ (d::(real, 3)\ cart)\ (p::(real, 3)\ cart)\ (k1::real)\ (k2::real)\ (k3::real)\ k4::real.\ \neg\ affine\_dependent\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \wedge CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) = (4::nat) \wedge IN\ p\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))))) \wedge k1 + (k2 + (k3 + k4)) = (1::real) \wedge p = vector\_add\ (\%\ k1\ a)\ (vector\_add\ (\%\ k2\ b)\ (vector\_add\ (\%\ k3\ c)\ (\%\ k4\ d))) \wedge k1 \leq (0::real) \longrightarrow k1 = (0::real)$

thm Marchal_cells_2.IN_2_2_IMP_CONVEX_HULL_4:

$\forall (a::(real, ?'a::type)\ cart)\ (b::(real, ?'a::type)\ cart)\ (x::(real, ?'a::type)\ cart)\ (y::(real, ?'a::type)\ cart)\ (m::(real, ?'a::type)\ cart)\ (n::(real, ?'a::type)\ cart)\ p::(real, ?'a::type)\ cart.\ between\ p\ (m, n) \wedge between\ m\ (a, b) \wedge between\ n\ (x, y) \longrightarrow IN\ p\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ (INSERT\ y\ EMPTY)))))$

thm Marchal_cells_2.BETWEEN_TRANS_3_CASES:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (x::(real, 3)\ cart)\ y::(real, 3)\ cart.\ between\ x\ (a, b) \wedge between\ y\ (a, b) \longrightarrow between\ x\ (a, y) \vee between\ x\ (y, b)$

thm Marchal_cells_2.OMEGA_LIST_UP_TO_2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ GSPEC\ (\lambda GEN\%PVAR\%1103::(real, 3)\ cart.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%1103\ (i \leq (2::nat))\ (omega\_list\_n\ V\ ul\ i)) = INSERT\ (omega\_list\_n\ V\ ul\ (0::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (1::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (2::nat))\ EMPTY))$

thm Marchal_cells_2.CONVEX_HULL_KY_LEMMA_5:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (c::(real, 3)\ cart)\ (d::?'a::type)\ (x::(real, 3)\ cart)\ (y::(real, 3)\ cart)\ (da::(real, 3)\ cart)\ p::(real, 3)\ cart.\ \neg\ affine\_dependent\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))) \wedge CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))) = (4::nat) \wedge da \neq x \wedge IN\ x\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))) \wedge between\ da\ (x, y) \wedge \neg\ IN\ p\ (hull\ affine\ (INSERT\ a\ (INSERT\ b\ (INSERT\ da\ EMPTY)))) \wedge IN\ p\ (HOL\_Light\_Import.INTER\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))))\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ (INSERT\ y\ EMPTY)))))) \longrightarrow IN\ p\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ (INSERT\ da\ EMPTY)))))$

thm Marchal_cells_2_new.AFF_GE_2_2:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. DISJOINT (INSERT x (INSERT v EMPTY)) (INSERT w (INSERT (?z::(real, ?'a::type) cart) EMPTY)) ⟶ aff_ge (INSERT x (INSERT v EMPTY)) (INSERT w (INSERT ?z EMPTY)) = GSPEC (λGEN%PVAR%1104::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1104 (∃ (t1::real) (t2::real) (t3::real) t4::real. (0::real) ≤ t3 ∧ (0::real) ≤ t4 ∧ t1 + (t2 + (t3 + t4)) = (1::real) ∧ y = vector_add (% t1 x) (vector_add (% t2 v) (vector_add (% t3 w) (% t4 ?z)))) y)

thm Marchal_cells_2_new.MEASURABLE_ROGERS:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) k::?'a::type. saturated V ∧ packing V ∧ barV V (3::nat) ul ⟶ measurable (rogers V ul)

thm Marchal_cells_2_new.FINITE_PERMUTE_3:

FINITE (GSPEC (λGEN%PVAR%1105::nat ⇒ nat. ∃ p::nat ⇒ nat. SETSPEC GEN%PVAR%1105 (permutes p (INSERT (0::nat) (INSERT (1::nat) (INSERT (2::nat) EMPTY)))) p))

thm Marchal_cells_2_new.FINITE_PERMUTE_4:

FINITE (GSPEC (λGEN%PVAR%1106::nat ⇒ nat. ∃ p::nat ⇒ nat. SETSPEC GEN%PVAR%1106 (permutes p (INSERT (0::nat) (INSERT (1::nat) (INSERT (2::nat) (INSERT (3::nat) EMPTY))))) p))

thm Marchal_cells_2_new.CLOSED_CONVEX_HULL_FINITE:

∀ s::(real, ?'a::type) cart ⇒ bool. FINITE s ⟶ HOL_Light_Import.closed (hull convex s)

thm Marchal_cells_2_new.CLOSED_MCELL:

∀ (V::(real, 3) cart ⇒ bool) (ul::(real, 3) cart list) k::nat. saturated V ∧ packing V ∧ barV V (3::nat) ul ⟶ HOL_Light_Import.closed (mcell k V ul)

thm Marchal_cells_2_new.CONVEX_HULL_4:

hull convex (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) (INSERT (?d::(real, ?'a::type) cart) EMPTY)))) = GSPEC (λGEN%PVAR%1114::(real, ?'a::type) cart. ∃ (u::real) (v::real) (w::real) z::real. SETSPEC GEN%PVAR%1114 ((0::real) ≤ u ∧ (0::real) ≤ v ∧ (0::real) ≤ w ∧ (0::real) ≤ z ∧ u + (v + (w + z)) = (1::real)) (vector_add (% u ?a) (vector_add (% v ?b) (vector_add (% w ?c) (% z ?d)))))

thm Marchal_cells_2_new.SET_EQ_LEMMA:

((?A::?'a::type ⇒ bool) = (?B::?'a::type ⇒ bool)) = (∀ x::?'a::type. (IN x ?A ⟶ IN x ?B) ∧ (IN x ?B ⟶ IN x ?A))

thm Marchal_cells_2_new.SET_OF_0_TO_3:

GSPEC (λGEN%PVAR%1124::nat. ∃ j::nat. SETSPEC GEN%PVAR%1124 (j < (4::nat)) j) = INSERT (0::nat) (INSERT (1::nat) (INSERT (2::nat) (INSERT (3::nat) EMPTY)))

thm Marchal_cells_2_new.SET_OF_0_TO_2:

GSPEC (λ*GEN%PVAR%1125*::*nat*. $\exists$ *j*::*nat*. *SETSPEC GEN%PVAR%1125* (*j* $\leq$ (*2*::*nat*)) *j*) = *INSERT* (*0*::*nat*) (*INSERT* (*1*::*nat*) (*INSERT* (*2*::*nat*) *EMPTY*))

thm Marchal_cells_2_new.MXI_EXPLICIT:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*ul*::(*real*, *3*) *cart list*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::(*real*, *3*) *cart*) *u3*::(*real*, *3*) *cart*. *packing V* $\wedge$ *saturated V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\wedge$ *ul* = [*u0*, *u1*, *u2*, *u3*] $\wedge$ *hl* (*truncate_simplex* (*2*::*nat*) *ul*) < *sqrt* (*real_of_nat* (*2*::*nat*)) $\wedge$ *sqrt* (*real_of_nat* (*2*::*nat*)) $\leq$ *hl ul* $\longrightarrow$ ($\exists$ *s*::(*real*, *3*) *cart*. *between s* (*omega_list_n V ul* (*2*::*nat*), *omega_list_n V ul* (*3*::*nat*)) $\wedge$ *distance* (*u0*, *s*) = *sqrt* (*real_of_nat* (*2*::*nat*)) $\wedge$ *mxi V ul* = *s*)

thm Marchal_cells_2_new.proj_point:

$\forall$ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *proj_point e x* = *vector_sub x* (*projection e x*)

thm Marchal_cells_2_new.projection_proj_point:

$\forall$ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *projection e x* = *vector_sub x* (*proj_point e x*)

thm Marchal_cells_2_new.PRO_EXP:

$\forall$ (*e*::(*real*, *?'a*::*type*) *cart*) *x*::(*real*, *?'a*::*type*) *cart*. *proj_point e x* = % (*dot x e* / *dot e e*) *e*

thm Marchal_cells_2_new.BETWEEN_PROJ_POINT:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) *e*::(*real*, *?'a*::*type*) *cart*. *between x* (*a*, *b*) $\longrightarrow$ *between* (*proj_point e x*) (*proj_point e a*, *proj_point e b*)

thm Marchal_cells_2_new.OMEGA_LIST_TRUNCATE_1_NEW1:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::(*real*, *3*) *cart*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*, *u2*] (*1*::*nat*) = *omega_list V* [*u0*, *u1*]

thm Marchal_cells_2_new.OMEGA_LIST_TRUNCATE_1_NEW2:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::*?'b*::*type*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*] (*1*::*nat*) = *omega_list V* [*u0*, *u1*]

thm Marchal_cells_2_new.OMEGA_LIST_TRUNCATE_2_NEW1:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*u2*::(*real*, *3*) *cart*) *u3*::*?'a*::*type*. *omega_list_n V* [*u0*, *u1*, *u2*] (*2*::*nat*) = *omega_list V* [*u0*, *u1*, *u2*]

thm Marchal_cells_2_new.IN_AFFINE_KY_LEMMA1:

$\forall (x::(real, ?'a::type)\ cart)\ s::(real, ?'a::type)\ cart \Rightarrow bool.\ IN\ x\ s \longrightarrow IN\ x\ (hull\ affine\ s)$

thm Marchal_cells_2_new.SUM_CLAUSES_alt:

$\forall (x::?'a::type)\ (f::?'a::type \Rightarrow real)\ s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow sum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ sum\ s\ f\ else\ f\ x + sum\ s\ f)$

thm Marchal_cells_2_new.CARD4_IMP_DISTINCT:

$\forall (a::?'a::type)\ (b::?'a::type)\ (c::?'a::type)\ d::?'a::type.\ CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) = (4::nat) \longrightarrow a \neq b$

thm Marchal_cells_2_new.VSUM_CLAUSES_alt:

$\forall (x::?'b::type)\ (f::?'b::type \Rightarrow (real, ?'a::type)\ cart)\ s::?'b::type \Rightarrow bool.\ FINITE\ s \longrightarrow vsum\ (INSERT\ x\ s)\ f = (if\ IN\ x\ s\ then\ vsum\ s\ f\ else\ vector\_add\ (f\ x)\ (vsum\ s\ f))$

thm Marchal_cells_2_new.OMEGA_LIST_UP_TO_2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ ul::(real, 3)\ cart\ list.\ GSPEC\ (\lambda GEN\%PVAR\%1134::(real, 3)\ cart.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%1134\ (i \leq (2::nat))\ (omega\_list\_n\ V\ ul\ i)) = INSERT\ (omega\_list\_n\ V\ ul\ (0::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (1::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (2::nat))\ EMPTY))$

thm Marchal_cells_2_new.CONVEX_HULL_KY_LEMMA_5:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (c::(real, 3)\ cart)\ (d::?'a::type)\ (x::(real, 3)\ cart)\ (y::(real, 3)\ cart)\ (da::(real, 3)\ cart)\ p::(real, 3)\ cart.\ \neg\ affine\_dependent\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))) \wedge CARD\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))) = (4::nat) \wedge da \neq x \wedge IN\ x\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))) \wedge between\ da\ (x, y) \wedge \neg\ IN\ p\ (hull\ affine\ (INSERT\ a\ (INSERT\ b\ (INSERT\ da\ EMPTY)))) \wedge IN\ p\ (HOL\_Light\_Import.INTER\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ da\ EMPTY)))))\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ (INSERT\ y\ EMPTY)))))) \longrightarrow IN\ p\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ x\ (INSERT\ da\ EMPTY)))))$

thm Marchal_cells_2_new.KY_PERMUTES_2_PERMUTES_3:

$\forall p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (0::nat)\ (2::nat)) \longrightarrow permutes\ p\ (dotdot\ (0::nat)\ (3::nat))$

thm Marchal_cells_2_new.TABLE_4:

$\forall f::nat \Rightarrow ?'a::type.\ TABLE\ f\ (4::nat) = [f\ (0::nat), f\ (1::nat), f\ (2::nat), f\ (3::nat)] \wedge TABLE\ f\ (3::nat) = [f\ (0::nat), f\ (1::nat), f\ (2::nat)] \wedge TABLE\ f\ (2::nat) = [f\ (0::nat), f\ (1::nat)] \wedge TABLE\ f\ (1::nat) = [f\ (0::nat)] \wedge TABLE\ f\ (0::nat) = []$

thm Marchal_cells_2_new.MEM_LEFT_ACTION_LIST_2:

$\forall$ (*ul*::*?'a*::*type list*) (*p*::*nat* $\Rightarrow$ *nat*) *x*::*?'a*::*type*. (*2*::*nat*) $\leq$ *length ul* $\wedge$ *permutes p* (*dotdot* (*0*::*nat*) (*length ul* $-$ (*2*::*nat*))) $\longrightarrow$ *MEM x* (*left_action_list p ul*) = *MEM x ul*

thm Marchal_cells_2_new.SET_OF_LIST_LEFT_ACTION_LIST_2:

$\forall$ (*ul*::*?'a*::*type list*) *p*::*nat* $\Rightarrow$ *nat*. (*2*::*nat*) $\leq$ *length ul* $\wedge$ *permutes p* (*dotdot* (*0*::*nat*) (*length ul* $-$ (*2*::*nat*))) $\longrightarrow$ *set_of_list* (*left_action_list p ul*) = *set_of_list ul*

thm Marchal_cells_2_new.OMEGA_LIST_2_EXPLICIT_NEW:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *ul*::(*real*, *3*) *cart list*. *saturated V* $\wedge$ *packing V* $\wedge$ *IN ul* (*barV V* (*3*::*nat*)) $\wedge$ *ul* = [*a*, *b*, *c*, *d*] $\wedge$ *hl* [*a*, *b*, *c*] $<$ *sqrt* (*real_of_nat* (*2*::*nat*)) $\longrightarrow$ *omega_list_n V ul* (*2*::*nat*) = *circumcenter* (*INSERT a* (*INSERT b* (*INSERT c EMPTY*)))

thm Marchal_cells_2_new.INTER_RCONE_GE_IMP_BETWEEN_PROJ_POINT:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*p*::(*real*, *3*) *cart*) *r*::*real*. *a* $\neq$ *b* $\wedge$ (*0*::*real*) $\leq$ *r* $\wedge$ *IN p* (*HOL_Light_Import*.*INTER* (*rcone_ge a b r*) (*rcone_ge b a r*)) $\longrightarrow$ *between* (*vector_add* (*proj_point* (*vector_sub b a*) (*vector_sub p a*)) *a*) (*a*, *b*)

thm Marchal_cells_2_new.INTER_RCONE_GE_LT_lemma:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*p*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart*) (*h*::*?'a*::*type*) *r*::*real*. *a* $\neq$ *b* $\wedge$ *s* = *midpoint* (*a*, *b*) $\wedge$ (*0*::*real*) $<$ *r* $\wedge$ *p* $\neq$ *a* $\wedge$ *p* $\neq$ *b* $\wedge$ *inverse_class*.*inverse* (*real_of_nat* (*2*::*nat*)) $<$ r^2 $\wedge$ *between* (*vector_add* (*proj_point* (*vector_sub b a*) (*vector_sub p a*)) *a*) (*a*, *s*) $\wedge$ *IN p* (*HOL_Light_Import*.*INTER* (*rcone_ge a b r*) (*rcone_ge b a r*)) $\longrightarrow$ *distance* (*s*, *p*) $<$ *distance* (*s*, *a*)

thm Marchal_cells_2_new.INTER_RCONE_GE_LE_lemma:

$\forall$ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*p*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart*) (*h*::*?'a*::*type*) *r*::*real*. *a* $\neq$ *b* $\wedge$ *s* = *midpoint* (*a*, *b*) $\wedge$ (*0*::*real*) $<$ *r* $\wedge$ *p* $\neq$ *a* $\wedge$ *p* $\neq$ *b* $\wedge$ *inverse_class*.*inverse* (*real_of_nat* (*2*::*nat*)) $\leq$ r^2 $\wedge$ *between* (*vector_add* (*proj_point* (*vector_sub b a*) (*vector_sub p a*)) *a*) (*a*, *s*) $\wedge$ *IN p* (*HOL_Light_Import*.*INTER* (*rcone_ge a b r*) (*rcone_ge b a r*)) $\longrightarrow$ *distance* (*s*, *p*) $\leq$ *distance* (*s*, *a*)

thm Marchal_cells_2_new.LEFT_ACTION_LIST_PROPERTIES:

$\forall$ (*V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*ul*::(*real*, *3*) *cart list*) (*p*::*nat* $\Rightarrow$ *nat*) (*u0*::*?'e*::*type*) (*u1*::*?'d*::*type*) (*u2*::*?'c*::*type*) (*u3*::*?'b*::*type*) *k*::*?'a*::*type*. *packing V* $\wedge$ *saturated V* $\wedge$ *barV V* (*3*::*nat*) *ul* $\wedge$ *hl* (*truncate_simplex* (*2*::*nat*) *ul*) $<$ *sqrt* (*real_of_nat* (*2*::*nat*)) $\wedge$ *sqrt* (*real_of_nat* (*2*::*nat*)) $\leq$ *hl ul* $\wedge$ *permutes p* (*dotdot* (*0*::*nat*) (*2*::*nat*)) $\wedge$ (*?xl*::(*real*, *3*) *cart list*) = *left_action_list p ul* $\longrightarrow$ *IN ?xl* (*barV V* (*3*::*nat*)) $\wedge$ *omega_list_n V ?xl* (*2*::*nat*) = *omega_list_n V ul* (*2*::*nat*)

$\wedge$ *omega_list_n V ?xl (3::nat) = omega_list_n V ul (3::nat)* $\wedge$ *mxi V ?xl = mxi V ul*

thm Marchal_cells_2_new.MEM_LEFT_ACTION_LIST_3:

$\forall$ *(ul::?'a::type list) (p::nat* $\Rightarrow$ *nat) x::?'a::type. (3::nat)* $\leq$ *length ul* $\wedge$ *permutes p (dotdot (0::nat) (length ul* $-$ *(3::nat)))* $\longrightarrow$ *MEM x (left_action_list p ul) = MEM x ul*

thm Marchal_cells_2_new.SET_OF_LIST_LEFT_ACTION_LIST_3:

$\forall$ *(ul::?'a::type list) p::nat* $\Rightarrow$ *nat. (3::nat)* $\leq$ *length ul* $\wedge$ *permutes p (dotdot (0::nat) (length ul* $-$ *(3::nat)))* $\longrightarrow$ *set_of_list (left_action_list p ul) = set_of_list ul*

thm Marchal_cells_2_new.LEFT_ACTION_LIST_1_EXPLICIT:

$\forall$ *(ul::(real, 3) cart list) (u0::(real, 3) cart) (u1::(real, 3) cart) (u2::(real, 3) cart) (u3::(real, 3) cart) p::nat* $\Rightarrow$ *nat. packing (?V::(real, 3) cart* $\Rightarrow$ *bool)* $\wedge$ *barV ?V (3::nat) ul* $\wedge$ *ul = [u0, u1, u2, u3]* $\wedge$ *permutes p (dotdot (0::nat) (1::nat))* $\longrightarrow$ *left_action_list p ul = ul* $\vee$ *left_action_list p ul = [u1, u0, u2, u3]*

thm Marchal_cells_2_new.LEFT_ACTION_LIST_1_PROPERTIES:

$\forall$ *(V::(real, 3) cart* $\Rightarrow$ *bool) (ul::(real, 3) cart list) p::nat* $\Rightarrow$ *nat. packing V* $\wedge$ *saturated V* $\wedge$ *barV V (3::nat) ul* $\wedge$ *permutes p (dotdot (0::nat) (1::nat))* $\wedge$ *hl (truncate_simplex (1::nat) ul) < sqrt (real_of_nat (2::nat))* $\wedge$ *sqrt (real_of_nat (2::nat))* $\leq$ *hl ul* $\wedge$ *(?xl::(real, 3) cart list) = left_action_list p ul* $\longrightarrow$ *IN ?xl (barV V (3::nat))* $\wedge$ *omega_list_n V ?xl (1::nat) = omega_list_n V ul (1::nat)* $\wedge$ *omega_list_n V ?xl (2::nat) = omega_list_n V ul (2::nat)* $\wedge$ *omega_list_n V ?xl (3::nat) = omega_list_n V ul (3::nat)* $\wedge$ *mxi V ?xl = mxi V ul*

thm Marchal_cells_2_new.NUMSEG_012:

INSERT (0::nat) (INSERT (1::nat) (INSERT (2::nat) EMPTY)) = dotdot (0::nat) (2::nat)

thm Marchal_cells_2_new.SET_OF_LIST_TRUN2_LEFT_ACTION_LIST2:

$\forall$ *(V::(real, 3) cart* $\Rightarrow$ *bool) (ul::(real, 3) cart list) p::nat* $\Rightarrow$ *nat. packing V* $\wedge$ *saturated V* $\wedge$ *barV V (3::nat) ul* $\wedge$ *permutes p (dotdot (0::nat) (2::nat))* $\longrightarrow$ *set_of_list (truncate_simplex (2::nat) (left_action_list p ul)) = set_of_list (truncate_simplex (2::nat) ul)*

thm Marchal_cells_2_new.SQRT2_LT_2:

sqrt (real_of_nat (2::nat)) < real_of_nat (2::nat)

thm Sltstlo.SLTSTLO1:

$\forall$ *(V::(real, 3) cart* $\Rightarrow$ *bool) (ul::(real, 3) cart list) p::(real, 3) cart. saturated V* $\wedge$ *packing V* $\wedge$ *barV V (3::nat) ul* $\wedge$ *IN p (rogers V ul)* $\longrightarrow$ *(* $\exists$ *i* $\leq$ *4::nat. IN p (mcell i V ul))*

thm NULLSET_RULES_conjunct1:

$\forall (s::(real,\ 3)\ cart \Rightarrow bool)\ t::(real,\ 3)\ cart \Rightarrow bool.\ negligible\ s \wedge negligible\ t \longrightarrow negligible\ (HOL\_Light\_Import.UNION\ s\ t)$

thm Trigonometry2.UV_IN_AFF2_conjunct1:

$IN\ (?v::(real,\ ?'a::type)\ cart)\ (hull\ affine\ (INSERT\ (?u::(real,\ ?'a::type)\ cart)\ (INSERT\ ?v\ EMPTY)))$

thm Trigonometry2.UV_IN_AFF2_conjunct0:

$IN\ (?u::(real,\ ?'a::type)\ cart)\ (hull\ affine\ (INSERT\ ?u\ (INSERT\ (?v::(real,\ ?'a::type)\ cart)\ EMPTY)))$

thm Sltstlo.SLTSTLO2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ \exists Z::(real,\ 3)\ cart \Rightarrow bool.\ \forall p::(real,\ 3)\ cart.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow negligible\ Z \wedge (IN\ p\ (DIFF\ (rogers\ V\ ul)\ Z) \longrightarrow (\exists !i::nat.\ i \leq (4::nat) \wedge IN\ p\ (mcell\ i\ V\ ul)))$

thm Lepjbdj.LEPJBDJ:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge (1::nat) \leq k \wedge k \leq (4::nat) \wedge mcell\ k\ V\ ul \neq EMPTY \longrightarrow HOL\_Light\_Import.INTER\ V\ (mcell\ k\ V\ ul) = set\_of\_list\ (truncate\_simplex\ (k - (1::nat))\ ul)$

thm Lepjbdj.LEPJBDJ_0:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow HOL\_Light\_Import.INTER\ V\ (mcell\ (0::nat)\ V\ ul) = EMPTY$

thm Urrphbz1.INTER_RCONE_GE_LE_lemma:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (p::(real,\ 3)\ cart)\ (s::(real,\ 3)\ cart)\ (h::?'a::type)\ r::real.\ a \neq b \wedge s = midpoint\ (a,\ b) \wedge (0::real) < r \wedge p \neq a \wedge p \neq b \wedge inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) \leq r^2 \wedge between\ (vector\_add\ (proj\_point\ (vector\_sub\ b\ a)\ (vector\_sub\ p\ a))\ a)\ (a,\ s) \wedge IN\ p\ (HOL\_Light\_Import.INTER\ (rcone\_ge\ a\ b\ r)\ (rcone\_ge\ b\ a\ r)) \longrightarrow distance\ (s,\ p) \leq distance\ (s,\ a)$

thm Urrphbz1.MCELL_2_PROPERTIES_lemma1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (k::?'a::type)\ p::(real,\ 3)\ cart.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ p\ (mcell2\ V\ ul) \longrightarrow distance\ (midpoint\ (hd\ ul,\ hd\ (tl\ ul)),\ p) \leq hl\ (truncate\_simplex\ (1::nat)\ ul)$

thm Urrphbz1.BOUNDED_MCELL:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow bounded\ (mcell\ k\ V\ ul)$

thm Urrphbz1.MEASURABLE_MCELL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow measurable\ (mcell\ k\ V\ ul)$

thm Urrphbz2.EVENTUALLY_RADIAL_RCONE_GE_ABC_A:

$\forall (a::real)\ (u0::(real, 3)\ cart)\ u1::(real, 3)\ cart.\ eventually\_radial\ u0\ (rcone\_ge\ u0\ u1\ a)$

thm Urrphbz2.EVENTUALLY_RADIAL_RCONE_GE_ABC_B:

$\forall (a::real)\ (u0::(real, 3)\ cart)\ u1::(real, 3)\ cart.\ u0 \neq u1 \wedge (0::real) < a \wedge a < (1::real) \longrightarrow eventually\_radial\ u1\ (rcone\_ge\ u0\ u1\ a)$

thm Urrphbz2.EVENTUALLY_RADIAL_AFF_GE:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (c::(real, 3)\ cart)\ d::(real, 3)\ cart.\ DISJOINT\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ (INSERT\ c\ (INSERT\ d\ EMPTY)) \longrightarrow eventually\_radial\ a\ (aff\_ge\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ (INSERT\ c\ (INSERT\ d\ EMPTY)))$

thm Urrphbz2.FUN_AFFINE_KLEMMA:

$\forall (a::(real, 3)\ cart)\ (b::(real, 3)\ cart)\ (c::(real, 3)\ cart)\ d::(real, 3)\ cart.\ aff\_dim\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY))) = int\ (2::nat) \wedge \neg\ IN\ d\ (hull\ affine\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))) \longrightarrow \neg\ IN\ a\ (hull\ convex\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))))$

thm Urrphbz2.URRPHBZ2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ v::(real, 3)\ cart.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ v\ V \longrightarrow eventually\_radial\ v\ (mcell\ k\ V\ ul)$

thm Hdtfnfz.HDTFNFZ:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ (v::?'a::type)\ X::(real, 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge X = mcell\ k\ V\ ul \wedge \neg\ negligible\ X \longrightarrow VX\ V\ X = HOL\_Light\_Import.INTER\ V\ X$

thm Urrphbz3.URRPHBZ3:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (k::nat)\ v::(real, 3)\ cart.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge \neg\ negligible\ (mcell\ k\ V\ ul) \wedge IN\ v\ (DIFF\ V\ (VX\ V\ (mcell\ k\ V\ ul))) \longrightarrow (\exists\, t>0::real.\ \forall\, p::(real, 3)\ cart.\ IN\ p\ (mcell\ k\ V\ ul) \longrightarrow t < distance\ (p, v))$

thm Rvfxzbu.RVFXZBU:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (i::nat)\ p::nat \Rightarrow nat.\ IN\ i\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \wedge saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge permutes\ p\ (dotdot\ (0::nat)\ (i - (1::nat))) \longrightarrow mcell\ i\ V\ (left\_action\_list\ p\ ul) = mcell\ i\ V\ ul$

thm Ynhyjit.YNHYJIT:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (i::nat)\ (p::nat \Rightarrow nat)\ vl::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ i\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))) \wedge hl\ (truncate\_simplex\ (i - (1::nat))\ ul) < sqrt\ (real\_of\_nat\ (2::nat)) \wedge sqrt\ (real\_of\_nat\ (2::nat)) \leq hl\ ul \wedge permutes\ p\ (dotdot\ (0::nat)\ (i - (1::nat))) \wedge vl = left\_action\_list\ p\ ul \longrightarrow barV\ V\ (3::nat)\ vl \wedge (\forall j::nat.\ i - (1::nat) \leq j \wedge j \leq (3::nat) \longrightarrow omega\_list\_n\ V\ vl\ j = omega\_list\_n\ V\ ul\ j)$

thm Njiutiu.CLOSEST_POINT_SUBSET_lemma:

$\forall (a::(real,\ 3)\ cart)\ (x::(real,\ 3)\ cart)\ (S::(real,\ 3)\ cart \Rightarrow bool)\ P::(real,\ 3)\ cart \Rightarrow bool.\ a = closest\_point\ S\ x \wedge IN\ a\ P \wedge SUBSET\ P\ S \wedge convex\ P \wedge HOL\_Light\_Import.closed\ P \wedge HOL\_Light\_Import.closed\ S \wedge P \neq EMPTY \longrightarrow a = closest\_point\ P\ x$

thm Njiutiu.AFF_DEPENDENT_AFF_DIM_4:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ d::(real,\ 3)\ cart.\ affine\_dependent\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \longrightarrow aff\_dim\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY)))) \leq int\ (2::nat)$

thm Njiutiu.NJIUTIU:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ vl::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge barV\ V\ (3::nat)\ vl \wedge rogers\ V\ ul = rogers\ V\ vl \wedge aff\_dim\ (rogers\ V\ ul) = int\ (3::nat) \longrightarrow (\forall i::nat.\ (0::nat) \leq i \wedge i \leq (3::nat) \longrightarrow omega\_list\_n\ V\ ul\ i = omega\_list\_n\ V\ vl\ i)$

thm Tezffsk.TEZFFSK:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (vl::(real,\ 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge barV\ V\ (3::nat)\ vl \wedge rogers\ V\ ul = rogers\ V\ vl \wedge aff\_dim\ (rogers\ V\ ul) = int\ (3::nat) \wedge k \leq (3::nat) \wedge hl\ (truncate\_simplex\ k\ ul) < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow truncate\_simplex\ k\ ul = truncate\_simplex\ k\ vl$

thm Sum_beta_bump.SUM_BETA_BUMP_LEMMA:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ X::(real,\ 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge mcell\_set\ V\ X \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%1174::(real,\ 3)\ cart \Rightarrow bool.\ \exists e::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1174\ (IN\ e\ (critical\_edgeX\ V\ X))\ e))\ (\lambda e::(real,\ 3)\ cart \Rightarrow bool.\ beta\_bump\ V\ e\ X) = (0::real)$

thm Qzksykg.CONVEX_HULL_4_IMP_3_1:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ (c::(real,\ 3)\ cart)\ (d::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ IN\ x\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ (INSERT\ d\ EMPTY))))) \longrightarrow (\exists (x1::(real,\ 3)\ cart)\ (t1::real)\ t2::real.\ IN\ x1\ (hull\ convex\ (INSERT\ a\ (INSERT\ b\ (INSERT\ c\ EMPTY)))) \wedge (0::real) \leq t1 \wedge (0::real) \leq t2 \wedge t1 + t2 = (1::real) \wedge x = vector\_add\ (\%\ t1\ x1)\ (\%\ t2\ d))$

thm Qzksykg.BARV_2_EXPLICIT:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ vl::(real,\ 3)\ cart\ list.\ barV\ V\ (2::nat)\ vl \longrightarrow (\exists (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ u2::(real,\ 3)\ cart.\ vl = [u0,\ u1,\ u2])$

thm Qzksykg.ROGERS_EXPLICIT_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (2::nat)\ ul \longrightarrow rogers\ V\ ul = hull\ convex\ (INSERT\ (hd\ ul)\ (INSERT\ (omega\_list\_n\ V\ ul\ (1::nat))\ (INSERT\ (omega\_list\_n\ V\ ul\ (2::nat))\ EMPTY)))$

thm Qzksykg.TWO_REARRANGEMENT_LEMMA:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (p::?'a::type)\ (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ u3::(real,\ 3)\ cart.\ packing\ V \wedge saturated\ V \wedge barV\ V\ (3::nat)\ ul \wedge ul = [u0,\ u1,\ u2,\ u3] \longrightarrow (\exists p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (0::nat)\ (1::nat)) \wedge [u1,\ u0,\ u2,\ u3] = left\_action\_list\ p\ ul)$

thm Qzksykg.SET_SUBSET_AFFINE_HULL:

$\forall S::(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ S\ (hull\ affine\ S)$

thm Qzksykg.QZKSYKG1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (vl::(real,\ 3)\ cart\ list)\ (k::nat)\ p::nat \Rightarrow nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ k\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \wedge mcell\ k\ V\ ul \neq EMPTY \wedge permutes\ p\ (dotdot\ (0::nat)\ (k - (1::nat))) \wedge vl = left\_action\_list\ p\ ul \longrightarrow barV\ V\ (3::nat)\ vl$

thm Qzksykg.QZKSYKG2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ k\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \longrightarrow SUBSET\ (mcell\ k\ V\ ul)\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%1177::(real,\ 3)\ cart \Rightarrow bool.\ \exists p::nat \Rightarrow nat.\ SETSPEC\ GEN\%PVAR\%1177\ (permutes\ p\ (dotdot\ (0::nat)\ (k - (1::nat))))\ (rogers\ V\ (left\_action\_list\ p\ ul)))))$

thm Ddzuphj.RCONE_GT_EQ_EMPTY_LEMMA:

$\forall (a::(real,\ 3)\ cart)\ (b::(real,\ 3)\ cart)\ r::real.\ (1::real) \leq r \longrightarrow rcone\_gt\ a\ b\ r = EMPTY$

thm Ddzuphj.DDZUPHJ:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (vl::(real,\ 3)\ cart\ list)\ k::nat.\ saturated\ V \wedge packing\ V \wedge IN\ k\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \wedge barV\ V\ (3::nat)\ ul \wedge barV\ V\ (3::nat)\ vl \wedge rogers\ V\ ul = rogers\ V\ vl \wedge aff\_dim\ (rogers\ V\ ul) = int\ (3::nat) \wedge mcell\ k\ V\ ul \neq EMPTY \longrightarrow mcell\ k\ V\ ul = mcell\ k\ V\ vl$

thm Ajripqn.VOL_POS_LT_AFF_DIM_3:

$\forall S::(real, 3)\ cart \Rightarrow bool.\ measurable\ S \wedge (0::real) < HOL\_Light\_Import.measure\ S \longrightarrow aff\_dim\ S = int\ (3::nat)$

thm Ajripqn.UP_TO_4_KY_LEMMA:

$\forall i::nat.\ (i \leq (4::nat)) = IN\ i\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY)))))$

thm Ajripqn.FINITE_SET_LIST_LEMMA:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1186::?'a::type\ list.\ \exists y::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%1186\ (\exists (u0::?'a::type)\ (u1::?'a::type)\ (u2::?'a::type)\ u3::?'a::type.\ IN\ u0\ s \wedge IN\ u1\ s \wedge IN\ u2\ s \wedge IN\ u3\ s \wedge y = [u0, u1, u2, u3])\ y))$

thm Ajripqn.AJRIPQN:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (vl::(real, 3)\ cart\ list)\ (i::nat)\ j::nat.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \wedge barV\ V\ (3::nat)\ vl \wedge IN\ i\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \wedge IN\ j\ (INSERT\ (0::nat)\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ (INSERT\ (3::nat)\ (INSERT\ (4::nat)\ EMPTY))))) \wedge \neg\ negligible\ (HOL\_Light\_Import.INTER\ (mcell\ i\ V\ ul)\ (mcell\ j\ V\ vl)) \longrightarrow i = j \wedge mcell\ i\ V\ ul = mcell\ j\ V\ vl$

thm Qzyzmjc.mcell_set_2:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ mcell\_set\ V = GSPEC\ (\lambda GEN\%PVAR\%1218::(real, 3)\ cart \Rightarrow bool.\ \exists X::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1218\ (\exists (i::nat)\ ul::(real, 3)\ cart\ list.\ X = mcell\ i\ V\ ul \wedge IN\ ul\ (barV\ V\ (3::nat)) \wedge i \leq (4::nat))\ X)$

thm Qzyzmjc.BARV_3_IMP_FINITE_lemma1:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (u::(real, 3)\ cart)\ v::(real, 3)\ cart.\ packing\ V \wedge saturated\ V \wedge barV\ V\ (3::nat)\ ul \wedge SUBSET\ (INSERT\ u\ (INSERT\ v\ EMPTY))\ (set\_of\_list\ ul) \longrightarrow distance\ (u, v) < real\_of\_nat\ (4::nat)$

thm Qzyzmjc.BARV_3_IMP_FINITE_lemma2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (ul::(real, 3)\ cart\ list)\ (v::(real, 3)\ cart)\ k::?'a::type.\ packing\ V \wedge saturated\ V \wedge barV\ V\ (3::nat)\ ul \wedge IN\ v\ (set\_of\_list\ ul) \longrightarrow SUBSET\ (set\_of\_list\ ul)\ (ball\ (v, real\_of\_nat\ (4::nat)))$

thm Qzyzmjc.lemma_r_r′_fix2:

$\forall (C::(real, 3)\ cart \Rightarrow bool)\ (x::(real, 3)\ cart)\ (r::real)\ s::real.\ measurable\ C \wedge radial\_norm\ r\ x\ C \wedge (0::real) < s \wedge s \leq r \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ C\ (normball\ x\ s)) = HOL\_Light\_Import.measure\ C * (s\ /\ r)^{3::nat}$

thm Qzyzmjc.MCELL_SET_NOT_EMPTY:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ X::?'a::type.\ saturated\ V \wedge packing\ V \wedge IN\ v\ V \longrightarrow GSPEC\ (\lambda GEN\%PVAR\%1228::(real,\ 3)\ cart \Rightarrow bool.\ \exists X::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1228\ (mcell\_set\ V\ X \wedge \neg\ negligible\ X \wedge IN\ v\ (HOL\_Light\_Import.INTER\ V\ X))\ X) \neq EMPTY$

thm Qzyzmjc.QZYZMJC:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ X::?'a::type.\ saturated\ V \wedge packing\ V \wedge IN\ v\ V \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%1217::(real,\ 3)\ cart \Rightarrow bool.\ \exists X::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1217\ (mcell\_set\ V\ X \wedge IN\ v\ (VX\ V\ X))\ X))\ (sol\ v) = real\_of\_nat\ (4::nat) * pi$

thm Marchal_cells_3.HD_IN_ROGERS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow IN\ (hd\ ul)\ (rogers\ V\ ul)$

thm Marchal_cells_3.ROGERS_SUBSET_VORONOI_CLOSED:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ ul::(real,\ 3)\ cart\ list.\ saturated\ V \wedge packing\ V \wedge barV\ V\ (3::nat)\ ul \longrightarrow SUBSET\ (rogers\ V\ ul)\ (voronoi\_closed\ V\ (hd\ ul))$

thm Marchal_cells_3.HD_IN_MCELL:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (i::nat)\ (r::?'a::type)\ X::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge barV\ V\ (3::nat)\ ul \wedge X = mcell\ i\ V\ ul \wedge X \neq EMPTY \wedge i \neq (0::nat) \longrightarrow IN\ (hd\ ul)\ X$

thm Marchal_cells_3.FINITE_MCELL_SET_lemma1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (ul::(real,\ 3)\ cart\ list)\ (i::nat)\ (r::real)\ X::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge barV\ V\ (3::nat)\ ul \wedge X = mcell\ i\ V\ ul \wedge SUBSET\ X\ (ball\ (vec\ (0::nat),\ r)) \wedge X \neq EMPTY \longrightarrow (\forall u::(real,\ 3)\ cart.\ IN\ u\ (set\_of\_list\ ul) \longrightarrow IN\ u\ (ball\ (vec\ (0::nat),\ r + real\_of\_nat\ (6::nat))))$

thm Marchal_cells_3.FINITE_MCELL_SET_LEMMA:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ packing\ V \wedge saturated\ V \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1253::(real,\ 3)\ cart \Rightarrow bool.\ \exists X::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1253\ (SUBSET\ X\ (ball\ (vec\ (0::nat),\ r)) \wedge mcell\_set\ V\ X)\ X))$

thm Marchal_cells_3.CARD_BOUNDARY_INT_BALL_BOUND_1:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ k2::real.\ (0::real) < k1 \wedge (0::real) < k2 \longrightarrow (\exists C::real.\ \forall r \geq 1::real.\ real\_of\_nat\ (CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2)))) \leq C * r^2)$

thm Marchal_cells_3.CARD_BOUNDARY_INT_BALL_BOUND:

$\forall (x::(real,\ 3)\ cart)\ (k1::real)\ k2::real.\ \exists C::real.\ \forall r \geq 1::real.\ real\_of\_nat\ (CARD\ (DIFF\ (int\_ball\ x\ (r + k1))\ (int\_ball\ x\ (r - k2)))) \leq C * r^2$

thm Marchal_cells_3.CARD_INJ_LE:

$\forall (s::?'b::type \Rightarrow bool)\ (t::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ s \wedge FINITE\ t \wedge INJ\ f\ s\ t \longrightarrow CARD\ s \leq CARD\ t$

thm Marchal_cells_3.BOUNDARY_VOLUME:

$\forall (p::(real,\ 3)\ cart)\ (k1::real)\ k2::real.\ \exists C::real.\ \forall r \geq 1::real.\ HOL\_Light\_Import.measure\ (DIFF\ (ball\ (p,\ r\ +\ k1))\ (ball\ (p,\ r\ -\ k2))) \leq C * r^2$

thm Marchal_cells_3.PACKING_BALL_BOUNDARY:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (p::(real,\ 3)\ cart)\ (k1::real)\ k2::real.\ packing\ V \longrightarrow (\exists C::real.\ \forall r \geq 1::real.\ real\_of\_nat\ (CARD\ (DIFF\ (HOL\_Light\_Import.INTER\ V\ (ball\ (p,\ r\ +\ k1)))\ (HOL\_Light\_Import.INTER\ V\ (ball\ (p,\ r\ -\ k2))))) \leq C * r^2)$

thm Marchal_cells_3.MCELL_SUBSET_BALL_4:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ X::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \longrightarrow (\exists p::(real,\ 3)\ cart.\ SUBSET\ X\ (ball\ (p,\ real\_of\_nat\ (4::nat))))$

thm Marchal_cells_3.HL_2:

$\forall (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ hl\ [u,\ v] = inverse\_class.inverse\ (real\_of\_nat\ (2::nat)) * distance\ (u,\ v)$

thm Marchal_cells_3.HL_LE_SQRT2_IMP_BARV_1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (u0::(real,\ 3)\ cart)\ u1::(real,\ 3)\ cart.\ saturated\ V \wedge packing\ V \wedge IN\ u0\ V \wedge IN\ u1\ V \wedge u0 \neq u1 \wedge hl\ [u0,\ u1] < sqrt\ (real\_of\_nat\ (2::nat)) \longrightarrow barV\ V\ (1::nat)\ [u0,\ u1]$

thm Marchal_cells_3.RCONE_GE_SUBSET:

$\forall (a::real)\ (b::real)\ (u0::(real,\ ?'a::type)\ cart)\ u1::(real,\ ?'a::type)\ cart.\ a \leq b \longrightarrow SUBSET\ (rcone\_ge\ u0\ u1\ b)\ (rcone\_ge\ u0\ u1\ a)$

thm Marchal_cells_3.RCONE_GT_SUBSET:

$\forall (a::real)\ (b::real)\ (u0::(real,\ ?'a::type)\ cart)\ u1::(real,\ ?'a::type)\ cart.\ a \leq b \longrightarrow SUBSET\ (rcone\_gt\ u0\ u1\ b)\ (rcone\_gt\ u0\ u1\ a)$

thm Marchal_cells_3.BOUNDED_SING:

$\forall a::(real,\ ?'a::type)\ cart.\ bounded\ (INSERT\ a\ EMPTY)$

thm Marchal_cells_3.BOUNDED_HYPERPLANE_EQ_TRIVIAL:

$\forall (a::(real,\ ?'a::type)\ cart)\ b::real.\ bounded\ (GSPEC\ (\lambda GEN\%PVAR\%1272::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%1272\ (dot\ a\ x = b)\ x)) = (if\ a = vec\ (0::nat)\ then\ b \neq (0::real)\ else\ dimindex\ HOL\_Light\_Import.UNIV = (1::nat))$

thm Marchal_cells_3.UNBOUNDED_HYPERPLANE:

∀ (a::(real, 3) cart) b::real. a ≠ vec (0::nat) ⟶ ¬ bounded (GSPEC (λGEN%PVAR%1273::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%1273 (dot a x = b) x))

thm Marchal_cells_3.DIHV_SYM_2:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) t::(real, 3) cart. dihV x y z t = dihV x y t z

thm Marchal_cells_3.REAL_DIV_LE_1_TACTICS:

∀ (m::real) n::real. (0::real) < n ∧ m ≤ n ⟶ m / n ≤ (1::real)

thm Marchal_cells_3.REAL_DIV_GE_1_TACTICS:

∀ (m::real) n::real. (0::real) < n ∧ n ≤ m ⟶ (1::real) ≤ m / n

thm Marchal_cells_3.REAL_DIV_LT_1_TACTICS:

∀ (m::real) n::real. (0::real) < n ∧ m < n ⟶ m / n < (1::real)

thm Marchal_cells_3.REAL_DIV_GT_1_TACTICS:

∀ (m::real) n::real. (0::real) < n ∧ n < m ⟶ (1::real) < m / n

thm Marchal_cells_3.MCELL_ID_OMEGA_LIST_N:

∀ (V::(real, 3) cart ⇒ bool) (i::nat) (j::nat) (ul::(real, 3) cart list) vl::(real, 3) cart list. packing V ∧ saturated V ∧ barV V (3::nat) ul ∧ barV V (3::nat) vl ∧ mcell i V ul = mcell j V vl ∧ ¬ negligible (mcell i V ul) ∧ IN i (INSERT (2::nat) (INSERT (3::nat) (INSERT (4::nat) EMPTY))) ∧ IN j (INSERT (2::nat) (INSERT (3::nat) (INSERT (4::nat) EMPTY))) ⟶ i = j ∧ (∀ k::nat. i − (1::nat) ≤ k ∧ k ≤ (3::nat) ⟶ omega_list_n V ul k = omega_list_n V vl k)

thm Marchal_cells_3.MCELL_ID_MXI:

∀ (V::(real, 3) cart ⇒ bool) (i::nat) (j::nat) (ul::(real, 3) cart list) vl::(real, 3) cart list. packing V ∧ saturated V ∧ barV V (3::nat) ul ∧ barV V (3::nat) vl ∧ hd ul = hd vl ∧ mcell i V ul = mcell j V vl ∧ ¬ negligible (mcell i V ul) ∧ IN i (INSERT (2::nat) (INSERT (3::nat) EMPTY)) ∧ IN j (INSERT (2::nat) (INSERT (3::nat) EMPTY)) ⟶ mxi V ul = mxi V vl

thm Marchal_cells_3.AFFINE_HULL_3_INSERT:

∀ (a::(real, 3) cart) S::(real, 3) cart ⇒ bool. IN a (hull affine S) ⟶ hull affine (INSERT a S) = hull affine S

thm Marchal_cells_3.FINITE_EDGE_X2:

∀ (V::(real, 3) cart ⇒ bool) (e::(real, 3) cart ⇒ bool) (u0::(real, 3) cart) u1::(real, 3) cart. packing V ∧ saturated V ∧ e = INSERT u0 (INSERT u1 EMPTY) ⟶ FINITE (GSPEC (λGEN%PVAR%1313::(real, 3) cart ⇒ bool. ∃ X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%1313 (mcell_set V X ∧ edgeX V X e) X))

thm Marchal_cells_3.LIFT_MUL_CONTINUOUS_ON:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) (*g*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *continuous_on* (*lift* ∘ *f*) *s* ∧ *continuous_on* (*lift* ∘ *g*) *s* ⟶ *continuous_on* (*lift* ∘ (λ*x*::(*real*, *?'a*::*type*) *cart*. *f x* ∗ *g x*)) *s*

thm Marchal_cells_3.LIFT_DIV_CONTINUOUS_ON:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) (*g*::(*real*, *?'a*::*type*) *cart* ⇒ *real*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *continuous_on* (*lift* ∘ *f*) *s* ∧ *continuous_on* (*lift* ∘ *g*) *s* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x s* ⟶ *g x* ≠ (*0*::*real*)) ⟶ *continuous_on* (*lift* ∘ (λ*x*::(*real*, *?'a*::*type*) *cart*. *f x* / *g x*)) *s*

thm Marchal_cells_3.LIFT_DOT_CONTINUOUS_ON:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *continuous_on* (*lift* ∘ (λ*x*::(*real*, *?'a*::*type*) *cart*. *dot* (*vector_sub x a*) *b*)) *s*

thm Marchal_cells_3.LIFT_NORM_CONTINUOUS_ON:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *continuous_on* (*lift* ∘ (λ*x*::(*real*, *?'a*::*type*) *cart*. *vector_norm* (*vector_sub x a*))) *s*

thm DEF_aff_ge_alt:

aff_ge_alt = (λ(*_5007969*::(*real*, *3*) *cart* ⇒ *bool*) (*_5007970*::(*real*, *3*) *cart* ⇒ *bool*) *_5007971*::(*real*, *3*) *cart*. ∃ (*f*::(*real*, *3*) *cart* ⇒ *real*) *q*::(*real*, *3*) *cart* ⇒ *bool*. *FINITE q* ∧ *SUBSET q _5007970* ∧ *_5007971* = *lin_combo* (*HOL_Light_Import.UNION _5007969 q*) *f* ∧ (∀ *w*::(*real*, *3*) *cart*. *q w* ⟶ (*0*::*real*) ≤ *f w*) ∧ *sum* (*HOL_Light_Import.UNION _5007969 q*) *f* = (*1*::*real*))

thm Marchal_cells_3.aff_ge_alt:

∀ (*t*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *aff_ge_alt s t v* = (∃ (*f*::(*real*, *3*) *cart* ⇒ *real*) *q*::(*real*, *3*) *cart* ⇒ *bool*. *FINITE q* ∧ *SUBSET q t* ∧ *v* = *lin_combo* (*HOL_Light_Import.UNION s q*) *f* ∧ (∀ *w*::(*real*, *3*) *cart*. *q w* ⟶ (*0*::*real*) ≤ *f w*) ∧ *sum* (*HOL_Light_Import.UNION s q*) *f* = (*1*::*real*))

thm DEF_smallest_angle_set:

smallest_angle_set = (λ(*_5007990*::(*real*, *3*) *cart* ⇒ *bool*) (*_5007991*::(*real*, *3*) *cart*) *_5007992*::(*real*, *3*) *cart*. *SOME x*::(*real*, *3*) *cart*. *IN x _5007990* ∧ (∀ *y*::(*real*, *3*) *cart*. *IN y _5007990* ⟶ *dot* (*vector_sub y _5007991*) (*vector_sub _5007992 _5007991*) / (*vector_norm* (*vector_sub y _5007991*) ∗ *vector_norm* (*vector_sub _5007992 _5007991*)) ≤ *dot* (*vector_sub x _5007991*) (*vector_sub _5007992 _5007991*) / (*vector_norm* (*vector_sub x _5007991*) ∗ *vector_norm* (*vector_sub _5007992 _5007991*))))

thm Marchal_cells_3.smallest_angle_set:

∀ (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* ⇒ *bool*. *smallest_angle_set s u0 u1* = (*SOME x*::(*real*, *3*) *cart*. *IN x s* ∧ (∀ *y*::(*real*, *3*) *cart*. *IN y s* ⟶ *dot* (*vector_sub y u0*) (*vector_sub u1 u0*) / (*vector_norm* (*vector_sub y u0*) ∗

vector_norm (vector_sub u1 u0)) ≤ dot (vector_sub x u0) (vector_sub u1 u0) / (vector_norm (vector_sub x u0) ∗ vector_norm (vector_sub u1 u0))))

thm DEF_smallest_angle_line:

smallest_angle_line = (λ(_5008011::(real, 3) cart) _5008012::(real, 3) cart. smallest_angle_set (hull convex (INSERT _5008011 (INSERT _5008012 EMPTY))))

thm Marchal_cells_3.smallest_angle_line:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (c::(real, 3) cart) d::(real, 3) cart. smallest_angle_line a b c d = smallest_angle_set (hull convex (INSERT a (INSERT b EMPTY))) c d

thm Marchal_cells_3.SMALLEST_ANGLE_LINE_EXISTS:

∀ (a::(real, 3) cart) (b::(real, 3) cart) (u0::(real, 3) cart) u1::(real, 3) cart. u0 ≠ u1 ∧ ¬ IN u0 (hull convex (INSERT a (INSERT b EMPTY))) ⟶ (∃ x::(real, 3) cart. IN x (hull convex (INSERT a (INSERT b EMPTY))) ∧ (∀ y::(real, 3) cart. IN y (hull convex (INSERT a (INSERT b EMPTY))) ⟶ dot (vector_sub y u0) (vector_sub u1 u0) / (vector_norm (vector_sub y u0) ∗ vector_norm (vector_sub u1 u0)) ≤ dot (vector_sub x u0) (vector_sub u1 u0) / (vector_norm (vector_sub x u0) ∗ vector_norm (vector_sub u1 u0))))

thm Marchal_cells_3.SMALLEST_ANGLE_IN_CONVEX_HULL:

∀ (m::(real, 3) cart) (n::(real, 3) cart) (p::(real, 3) cart) (q::(real, 3) cart) x::(real, 3) cart. p ≠ q ∧ ¬ IN p (hull convex (INSERT m (INSERT n EMPTY))) ∧ x = smallest_angle_line m n p q ⟶ IN x (hull convex (INSERT m (INSERT n EMPTY)))

thm Marchal_cells_3.SMALLEST_ANGLE_LINE_PROPERTY:

∀ (m::(real, 3) cart) (n::(real, 3) cart) (u0::(real, 3) cart) (u1::(real, 3) cart) (x::(real, 3) cart) y::(real, 3) cart. u0 ≠ u1 ∧ ¬ IN u0 (hull convex (INSERT m (INSERT n EMPTY))) ∧ x = smallest_angle_line m n u0 u1 ∧ IN y (hull convex (INSERT m (INSERT n EMPTY))) ⟶ dot (vector_sub y u0) (vector_sub u1 u0) / (vector_norm (vector_sub y u0) ∗ vector_norm (vector_sub u1 u0)) ≤ dot (vector_sub x u0) (vector_sub u1 u0) / (vector_norm (vector_sub x u0) ∗ vector_norm (vector_sub u1 u0))

thm Marchal_cells_3.MCELL_SUBSET_BALL8_1:

∀ (v::(real, 3) cart) (ul::(real, 3) cart list) (i::nat) V::(real, 3) cart ⇒ bool. i ≤ (4::nat) ∧ packing V ∧ saturated V ∧ barV V (3::nat) ul ∧ IN v (mcell i V ul) ⟶ SUBSET (mcell i V ul) (ball (v, real_of_nat (8::nat)))

thm Marchal_cells_3.MCELL_SUBSET_BALL8:

∀ (v::(real, 3) cart) (ul::(real, 3) cart list) (i::nat) V::(real, 3) cart ⇒ bool. packing V ∧ saturated V ∧ barV V (3::nat) ul ∧ IN v (mcell i V ul) ⟶ SUBSET (mcell i V ul) (ball (v, real_of_nat (8::nat)))

thm Marchal_cells_3.FINITE_MCELL_SET_LEMMA_2:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (r::real)\ s::(real, 3)\ cart.\ packing\ V \wedge saturated\ V \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1324::(real, 3)\ cart \Rightarrow bool.\ \exists X::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1324\ (SUBSET\ X\ (ball\ (s, r)) \wedge mcell\_set\ V\ X)\ X))$

thm Marchal_cells_3.CONIC_CAP_WEDGE_EQ_0:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (a::real)\ (r::real)\ (w1::(real, 3)\ cart)\ w2::(real, 3)\ cart.\ a < (1::real) \wedge (0::real) < r \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (wedge\ v0\ v1\ w1\ w2)) = (0::real) \longrightarrow coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w1\ (INSERT\ w2\ EMPTY))))$

thm Marchal_cells_3.CONIC_CAP_AFF_GT_EQ_0:

$\forall (v0::(real, 3)\ cart)\ (v1::(real, 3)\ cart)\ (a::real)\ (r::real)\ (w1::(real, 3)\ cart)\ w2::(real, 3)\ cart.\ a < (1::real) \wedge (0::real) < r \wedge HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ v0\ v1\ r\ a)\ (aff\_gt\ (INSERT\ v0\ (INSERT\ v1\ EMPTY))\ (INSERT\ w1\ (INSERT\ w2\ EMPTY)))) = (0::real) \longrightarrow coplanar\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ w1\ (INSERT\ w2\ EMPTY))))$

thm Marchal_cells_3.CONIC_CAP_INTER_CONVEX_HULL_4_GT_0:

$\forall (u0::(real, 3)\ cart)\ (u1::(real, 3)\ cart)\ (w1::(real, 3)\ cart)\ (w2::(real, 3)\ cart)\ (r::real)\ a::real.\ (0::real) < r \wedge a < (1::real) \wedge (0::real) \leq a \wedge \neg\ coplanar\ (INSERT\ u0\ (INSERT\ u1\ (INSERT\ w1\ (INSERT\ w2\ EMPTY)))) \longrightarrow (0::real) < HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (conic\_cap\ u0\ u1\ r\ a)\ (hull\ convex\ (INSERT\ u0\ (INSERT\ u1\ (INSERT\ w1\ (INSERT\ w2\ EMPTY))))))$

thm Marchal_cells_3.MEASURABLE_BALL_AFF_GE:

$\forall (z::(real, ?'a::type)\ cart)\ (r::real)\ (s::(real, ?'a::type)\ cart \Rightarrow bool)\ t::(real, ?'a::type)\ cart \Rightarrow bool.\ measurable\ (HOL\_Light\_Import.INTER\ (ball\ (z, r))\ (aff\_ge\ s\ t))$

thm Marchal_cells_3.FINITE_LIST_KY_LEMMA_2:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1330::?'a::type\ list.\ \exists y::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%1330\ (\exists (u0::?'a::type)\ u1::?'a::type.\ IN\ u0\ s \wedge IN\ u1\ s \wedge y = [u0, u1])\ y))$

thm Marchal_cells_3.FINITE_SET_PRODUCT_KY_LEMMA:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1331::?'a::type \Rightarrow bool.\ \exists (u0::?'a::type)\ u1::?'a::type.\ SETSPEC\ GEN\%PVAR\%1331\ (IN\ u0\ s \wedge IN\ u1\ s)\ (INSERT\ u0\ (INSERT\ u1\ EMPTY))))$

thm Marchal_cells_3.pre_beta:

$\forall (g::?'b::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (u'::?'b::type)\ v'::?'b::type.\ (\exists f::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ \forall (u::?'b::type)\ v::?'b::type.\ f\ (INSERT\ u\ (INSERT\ v\ EMPTY)) = g\ u\ v) \longrightarrow GABS\ (\lambda f::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ \forall (u::?'b::type)$

$v::?'b::type.\ GEQ\ (f\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (g\ u\ v))\ (INSERT\ u'\ (INSERT\ v'\ EMPTY)) = g\ u'\ v'$

thm Marchal_cells_3.WELLDEFINED_FUNCTION_2:

$(\exists f::?'d::type \Rightarrow ?'c::type.\ \forall (x::?'b::type)\ y::?'a::type.\ f\ ((?s::?'b::type \Rightarrow ?'a::type \Rightarrow ?'d::type)\ x\ y) = (?t::?'b::type \Rightarrow ?'a::type \Rightarrow ?'c::type)\ x\ y) = (\forall (x::?'b::type)\ (x'::?'b::type)\ (y::?'a::type)\ y'::?'a::type.\ ?s\ x\ y = ?s\ x'\ y' \longrightarrow ?t\ x\ y = ?t\ x'\ y')$

thm Marchal_cells_3.well_defined_unordered_pair:

$(\exists f::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ \forall (u::?'b::type)\ v::?'b::type.\ f\ (INSERT\ u\ (INSERT\ v\ EMPTY)) = (?g::?'b::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ u\ v) = (\forall (u::?'b::type)\ v::?'b::type.\ ?g\ u\ v = ?g\ v\ u)$

thm Marchal_cells_3.BETA_PAIR_THM:

$\forall (g::?'b::type \Rightarrow ?'b::type \Rightarrow ?'a::type)\ (u'::?'b::type)\ v'::?'b::type.\ (\forall (u::?'b::type)\ v::?'b::type.\ g\ u\ v = g\ v\ u) \longrightarrow GABS\ (\lambda f::(?'b::type \Rightarrow bool) \Rightarrow ?'a::type.\ \forall (u::?'b::type)\ v::?'b::type.\ GEQ\ (f\ (INSERT\ u\ (INSERT\ v\ EMPTY)))\ (g\ u\ v))\ (INSERT\ u'\ (INSERT\ v'\ EMPTY)) = g\ u'\ v'$

thm Marchal_cells_3.DIHX_RANGE:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ (0::real) \leq dihX\ V\ X\ (u,\ v) \wedge dihX\ V\ X\ (u,\ v) \leq pi$

thm Marchal_cells_3.DIHX_LE_PI:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ dihX\ V\ X\ (u,\ v) \leq pi$

thm Marchal_cells_3.BOUNDED_VOLUME_MCELL:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ \exists c::real.\ \forall X::(real,\ 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge mcell\_set\ V\ X \longrightarrow HOL\_Light\_Import.measure\ X \leq c$

thm Marchal_cells_3.LEFT_ACTION_LIST_2_EXISTS:

$\forall (u0::?'a::type)\ (u1::?'a::type)\ (u2::?'a::type)\ (x::?'a::type)\ (y::?'a::type)\ (z::?'a::type)\ d::?'a::type.\ CARD\ (INSERT\ u0\ (INSERT\ u1\ (INSERT\ u2\ (INSERT\ d\ EMPTY)))) = (4::nat) \wedge INSERT\ x\ (INSERT\ y\ (INSERT\ z\ EMPTY)) = INSERT\ u0\ (INSERT\ u1\ (INSERT\ u2\ EMPTY)) \longrightarrow (\exists p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (0::nat)\ (2::nat)) \wedge [x,\ y,\ z,\ d] = left\_action\_list\ p\ [u0,\ u1,\ u2,\ d])$

thm Marchal_cells_3.LEFT_ACTION_LIST_3_EXISTS:

$\forall (u0::?'a::type)\ (u1::?'a::type)\ (u2::?'a::type)\ (u3::?'a::type)\ (x::?'a::type)\ (y::?'a::type)\ (z::?'a::type)\ t::?'a::type.\ CARD\ (INSERT\ u0\ (INSERT\ u1\ (INSERT\ u2\ (INSERT\ u3\ EMPTY)))) = (4::nat) \wedge INSERT\ x\ (INSERT\ y\ (INSERT\ z\ (INSERT\ t\ EMPTY))) = INSERT\ u0\ (INSERT\ u1\ (INSERT\ u2\ (INSERT\ u3\ EMPTY))) \longrightarrow (\exists p::nat \Rightarrow nat.\ permutes\ p\ (dotdot\ (0::nat)\ (3::nat)) \wedge [x,\ y,\ z,\ t] = left\_action\_list\ p\ [u0,\ u1,\ u2,\ u3])$

thm Marchal_cells_3.lmfun_bounded:

∀ *h*≥*0*::*real*. *lmfun h* ≤ *h0* / (*h0* − (*1*::*real*))

thm Marchal_cells_3.lmfun_pos_le:

∀ *h*::*real*. (*0*::*real*) ≤ *lmfun h*

thm Marchal_cells_3.BARV_CARD_LEMMA:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *barV V k ul* ⟶ *CARD* (*set_of_list ul*) = *k* + (*1*::*nat*)

thm Marchal_cells_3.BARV_LENGTH_LEMMA:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*ul*::(*real*, *3*) *cart list*) *k*::*nat*. *packing V* ∧ *barV V k ul* ⟶ *length ul* = *k* + (*1*::*nat*)

thm Marchal_cells_3.MCELL_ID_MXI_2:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*i*::*nat*) (*j*::*nat*) (*ul*::(*real*, *3*) *cart list*) *vl*::(*real*, *3*) *cart list*. *packing V* ∧ *saturated V* ∧ *barV V* (*3*::*nat*) *ul* ∧ *barV V* (*3*::*nat*) *vl* ∧ *mcell i V ul* = *mcell j V vl* ∧ ¬ *negligible* (*mcell i V ul*) ∧ *IN i* (*INSERT* (*2*::*nat*) (*INSERT* (*3*::*nat*) *EMPTY*)) ∧ *IN j* (*INSERT* (*2*::*nat*) (*INSERT* (*3*::*nat*) *EMPTY*)) ⟶ *mxi V ul* = *mxi V vl*

thm Marchal_cells_3.DIHV_SYM_3:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*c*::(*real*, *3*) *cart*) (*d*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *INSERT a* (*INSERT b EMPTY*) = *INSERT c* (*INSERT d EMPTY*) ∧ *INSERT x* (*INSERT y EMPTY*) = *INSERT z* (*INSERT t EMPTY*) ⟶ *dihV a b x y* = *dihV c d z t*

thm Marchal_cells_3.DIHX_SYM:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*X*::(*real*, *3*) *cart* ⇒ *bool*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *packing V* ∧ *saturated V* ∧ *mcell_set V X* ∧ *IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX V X*) ⟶ *dihX V X* (*u*, *v*) = *dihX V X* (*v*, *u*)

thm DEF_gammaY:

gammaY = (λ(*_6230159*::(*real*, *3*) *cart* ⇒ *bool*) (*_6230160*::(*real*, *3*) *cart* ⇒ *bool*) *_6230161*::*real* ⇒ *real*. *GABS* (λ*f*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *real*. ∀ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *GEQ* (*f* (*INSERT u* (*INSERT v EMPTY*))) (*if IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX _6230159 _6230160*) *then dihX _6230159 _6230160* (*u*, *v*) ∗ *_6230161* (*hl* [*u*, *v*]) *else* (*0*::*real*))))

thm Marchal_cells_3.gammaY:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*X*::(*real*, *3*) *cart* ⇒ *bool*) *f*::*real* ⇒ *real*. *gammaY V X f* = *GABS* (λ*fa*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *real*. ∀ (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *GEQ* (*fa* (*INSERT u* (*INSERT v EMPTY*))) (*if IN* (*INSERT u* (*INSERT v EMPTY*)) (*edgeX V X*) *then dihX V X* (*u*, *v*) ∗ *f* (*hl* [*u*, *v*]) *else* (*0*::*real*)))

thm Marchal_cells_3.gamma_y_lmfun_bound:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *packing V* ∧ *saturated V* ⟶ (∃ *d*::*real*. ∀ (*X*::(*real*, *3*) *cart* ⇒ *bool*) *e*::(*real*, *3*) *cart* ⇒ *bool*. *mcell_set V X* ∧ *edgeX V X e* ⟶ *gammaY V X lmfun e* ≤ *d*)

thm Marchal_cells_3.BETA_SET_2_THM:

∀ (*g*::(*?'b*::*type* ⇒ *bool*) ⇒ *?'a*::*type*) (*u0*::*?'b*::*type*) *v0*::*?'b*::*type*. *GABS* (λ*f*::(*?'b*::*type* ⇒ *bool*) ⇒ *?'a*::*type*. ∀ (*u*::*?'b*::*type*) *v*::*?'b*::*type*. *GEQ* (*f* (*INSERT u* (*INSERT v EMPTY*))) (*g* (*INSERT u* (*INSERT v EMPTY*)))) (*INSERT u0* (*INSERT v0 EMPTY*)) = *g* (*INSERT u0* (*INSERT v0 EMPTY*))

thm Marchal_cells_3.SUM_PAIR_2_SET:

∀ (*f*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *real*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *d*::*real*. *FINITE s* ⟶ *sum* (*GSPEC* (λ*GEN%PVAR%1335*::(*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*. ∃ (*m*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1335* (*IN m s* ∧ *IN n s* ∧ *m* ≠ *n* ∧ *distance* (*m*, *n*) ≤ *d*) (*m*, *n*))) (*GABS* (λ*fa*::(*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart* ⇒ *real*. ∀ (*m*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *GEQ* (*fa* (*m*, *n*)) (*f* (*INSERT m* (*INSERT n EMPTY*))))) = *real_of_nat* (*2*::*nat*) ∗ *sum* (*GSPEC* (λ*GEN%PVAR%1336*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. ∃ (*m*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1336* (*IN m s* ∧ *IN n s* ∧ *m* ≠ *n* ∧ *distance* (*m*, *n*) ≤ *d*) (*INSERT m* (*INSERT n EMPTY*)))) *f*

thm Marchal_cells_3.H0_LT_SQRT2:

h0 < *sqrt* (*real_of_nat* (*2*::*nat*))

thm Marchal_cells_3.gamma_y_pos_le:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*X*::(*real*, *3*) *cart* ⇒ *bool*) *e*::(*real*, *3*) *cart* ⇒ *bool*. *packing V* ∧ *saturated V* ∧ *mcell_set V X* ∧ *edgeX V X e* ⟶ (*0*::*real*) ≤ *gammaY V X lmfun e*

thm Marchal_cells_3.CARD_LIST_klemma:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type list* ⇒ *bool*. *FINITE s* ∧ *FINITE t* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%1339*::*?'a*::*type list*. ∃ (*u0*::*?'a*::*type*) *y1*::*?'a*::*type list*. *SETSPEC GEN%PVAR%1339* (*IN u0 s* ∧ *IN y1 t*) (*u0* # *y1*))) = *CARD s* ∗ *CARD t*

thm Marchal_cells_3.CARD_LIST_klemma_2:

∀ (*s*::*?'a*::*type* ⇒ *bool*) *t*::*?'a*::*type* ⇒ *bool*. *FINITE s* ∧ *FINITE t* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%1342*::*?'a*::*type list*. ∃ (*u0*::*?'a*::*type*) *y1*::*?'a*::*type*. *SETSPEC GEN%PVAR%1342* (*IN u0 s* ∧ *IN y1 t*) [*u0*, *y1*])) = *CARD s* ∗ *CARD t*

thm Marchal_cells_3.FINITE_LIST_klemma:

$\forall (s::?'a::type \Rightarrow bool)\ t::?'a::type\ list \Rightarrow bool.\ FINITE\ s \wedge FINITE\ t \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%1343::?'a::type\ list.\ \exists (u0::?'a::type)\ y1::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%1343\ (IN\ u0\ s \wedge IN\ y1\ t)\ (u0\ \#\ y1)))$

thm Marchal_cells_3.CARD_LIST_4_klemma:

$\forall s::?'a::type \Rightarrow bool.\ FINITE\ s \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1363::?'a::type\ list.\ \exists y::?'a::type\ list.\ SETSPEC\ GEN\%PVAR\%1363\ (\exists (u0::?'a::type)\ (u1::?'a::type)\ (u2::?'a::type)\ u3::?'a::type.\ IN\ u0\ s \wedge IN\ u1\ s \wedge IN\ u2\ s \wedge IN\ u3\ s \wedge y = [u0,\ u1,\ u2,\ u3])\ y)) = CARD\ s * (CARD\ s * (CARD\ s * CARD\ s))$

thm Marchal_cells_3.BOUNDS_VGEN_klemma:

$\forall (u0::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ (0::real) \leq r \wedge packing\ V \longrightarrow real\_of\_nat\ (CARD\ (HOL\_Light\_Import.INTER\ V\ (ball\ (u0,\ r)))) \leq (r + (1::real))^{3::nat}$

thm Marchal_cells_3.BOUNDS_V4_klemma:

$\forall (u0::(real,\ 3)\ cart)\ V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \longrightarrow CARD\ (HOL\_Light\_Import.INTER\ V\ (ball\ (u0,\ real\_of\_nat\ (4::nat)))) \leq (125::nat)$

thm Marchal_cells_3.CARD_MCELL_CONTAINS_POINT_klemma:

$\exists c::nat.\ \forall (V::(real,\ 3)\ cart \Rightarrow bool)\ u0::(real,\ 3)\ cart.\ saturated\ V \wedge packing\ V \wedge IN\ u0\ V \longrightarrow CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1379::(real,\ 3)\ cart \Rightarrow bool.\ \exists X::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1379\ (mcell\_set\ V\ X \wedge VX\ V\ X\ u0)\ X)) \leq c$

thm Marchal_cells_3.BOUND_BETA_BUMP:

$\exists c::real.\ \forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ e::(real,\ 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge mcell\_set\ V\ X \wedge critical\_edgeX\ V\ X\ e \longrightarrow beta\_bump\ V\ e\ X \leq c$

thm Marchal_cells_3.MCELL_SUBSET_BALL8_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \wedge IN\ v\ X \longrightarrow SUBSET\ X\ (ball\ (v,\ real\_of\_nat\ (8::nat)))$

thm Marchal_cells_3.critical_edge_subset_mcell:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \wedge critical\_edgeX\ V\ X\ x \longrightarrow SUBSET\ x\ X$

thm Marchal_cells_3.EDGEX_SUBSET_MCELL:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (X::(real,\ 3)\ cart \Rightarrow bool)\ e::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \wedge edgeX\ V\ X\ e \longrightarrow SUBSET\ e\ X$

thm Marchal_cells_3.CRITICAL_EDGEX_SUBSET_MCELL:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (X::(real, 3)\ cart \Rightarrow bool)\ e::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \wedge critical\_edgeX\ V\ X\ e \longrightarrow SUBSET\ e\ X$

thm Marchal_cells_3.FINITE_VX:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ X::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \longrightarrow FINITE\ (VX\ V\ X)$

thm Marchal_cells_3.CARD_EDGEX_LE_16:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ X::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \longrightarrow CARD\ (edgeX\ V\ X) \leq (16::nat)$

thm Marchal_cells_3.gamma_y_lmfun_bound2:

$\exists d::real.\ \forall (V::(real, 3)\ cart \Rightarrow bool)\ (X::(real, 3)\ cart \Rightarrow bool)\ e::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \wedge edgeX\ V\ X\ e \longrightarrow gammaY\ V\ X\ lmfun\ e \leq d$

thm Marchal_cells_3.BOUND_GAMMA_X_lmfun:

$\exists c::real.\ \forall (V::(real, 3)\ cart \Rightarrow bool)\ X::(real, 3)\ cart \Rightarrow bool.\ packing\ V \wedge saturated\ V \wedge mcell\_set\ V\ X \longrightarrow gammaX\ V\ X\ lmfun \leq c$

thm Grutoti.GRUTOTI:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (u0::(real, 3)\ cart)\ (u1::(real, 3)\ cart)\ e::(real, 3)\ cart \Rightarrow bool.\ saturated\ V \wedge packing\ V \wedge IN\ u0\ V \wedge IN\ u1\ V \wedge u0 \neq u1 \wedge hl\ [u0, u1] < sqrt\ (real\_of\_nat\ (2::nat)) \wedge e = INSERT\ u0\ (INSERT\ u1\ EMPTY) \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%1431::(real, 3)\ cart \Rightarrow bool.\ \exists X::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%1431\ (mcell\_set\ V\ X \wedge IN\ e\ (edgeX\ V\ X))\ X))\ (\lambda t::(real, 3)\ cart \Rightarrow bool.\ dihX\ V\ t\ (u0, u1)) = real\_of\_nat\ (2::nat) * pi$

thm Kizhltl.KIZHLTL1:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ \exists c::real.\ \forall r::real.\ saturated\ V \wedge packing\ V \wedge (1::real) \leq r \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%855::(real, 3)\ cart \Rightarrow bool.\ \exists X::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%855\ (SUBSET\ X\ (ball\ (vec\ (0::nat), r)) \wedge mcell\_set\ V\ X)\ X))\ HOL\_Light\_Import.measure + c * r^2 \leq sum\ (HOL\_Light\_Import.INTER\ V\ (ball\ (vec\ (0::nat), r)))\ (\lambda u::(real, 3)\ cart.\ HOL\_Light\_Import.measure\ (voronoi\_open\ V\ u))$

thm Kizhltl.KIZHLTL2:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ \exists c::real.\ \forall r::real.\ saturated\ V \wedge packing\ V \wedge (1::real) \leq r \longrightarrow real\_of\_nat\ (CARD\ (HOL\_Light\_Import.INTER\ V\ (ball\ (vec\ (0::nat), r)))) * (real\_of\_nat\ (8::nat) * mm1) + c * r^2 \leq real\_of\_nat\ (2::nat) * (mm1\ /\ pi) * sum\ (GSPEC\ (\lambda GEN\%PVAR\%856::(real, 3)\ cart \Rightarrow bool.\ \exists X::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%856\ (SUBSET\ X\ (ball\ (vec\ (0::nat), r)) \wedge mcell\_set\ V\ X)\ X))\ (total\_solid\ V)$

thm Kizhltl.KIZHLTL4:

∀ V::(real, 3) cart ⇒ bool. ∃ c::real. ∀ r::real. saturated V ∧ packing V ∧ (1::real) ≤ r ⟶ real_of_nat (8::nat) ∗ (mm2 / pi) ∗ sum (GSPEC (λGEN%PVAR%1566::(real, 3) cart ⇒ bool. ∃ X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%1566 (SUBSET X (ball (vec (0::nat), r)) ∧ mcell_set V X) X)) (λX::(real, 3) cart ⇒ bool. sum (edgeX V X) (GABS (λf::((real, 3) cart ⇒ bool) ⇒ real. ∀ (u::(real, 3) cart) v::(real, 3) cart. GEQ (f (INSERT u (INSERT v EMPTY))) (if IN (INSERT u (INSERT v EMPTY)) (edgeX V X) then dihX V X (u, v) ∗ lmfun (hl [u, v]) else (0::real))))) + c ∗ r^2 ≤ real_of_nat (8::nat) ∗ (mm2 ∗ sum (HOL_Light_Import.INTER V (ball (vec (0::nat), r))) (λu::(real, 3) cart. sum (GSPEC (λGEN%PVAR%1567::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%1567 (IN v V ∧ u ≠ v ∧ distance (u, v) ≤ real_of_nat (2::nat) ∗ h0) v)) (λv::(real, 3) cart. lmfun (hl [u, v]))))

thm Sum_gammax_lmfun_estimate.TSKAJXY_statement:

TSKAJXY_statement = (∀ (V::(real, 3) cart ⇒ bool) X::(real, 3) cart ⇒ bool. saturated V ∧ packing V ∧ mcell_set V X ∧ critical_edgeX V X = EMPTY ⟶ (0::real) ≤ gammaX V X lmfun)

thm Sum_gammax_lmfun_estimate.SUM_GAMMAX_LMFUN_ESTIMATE:

∀ V::(real, 3) cart ⇒ bool. ∃ c::real. ∀ r::real. saturated V ∧ packing V ∧ (1::real) ≤ r ∧ cell_cluster_estimate V ∧ TSKAJXY_statement ⟶ c ∗ r^2 ≤ sum (GSPEC (λGEN%PVAR%1603::(real, 3) cart ⇒ bool. ∃ X::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%1603 (SUBSET X (ball (vec (0::nat), r)) ∧ mcell_set V X) X)) (λX::(real, 3) cart ⇒ bool. gammaX V X lmfun)

thm Upfzbzm.FCC_COMPATABILITY_FUNC:

∀ V::(real, 3) cart ⇒ bool. saturated V ∧ packing V ∧ cell_cluster_estimate V ∧ TSKAJXY_statement ∧ lmfun_inequality V ∧ (?G::(real, 3) cart ⇒ real) = (λu::(real, 3) cart. − HOL_Light_Import.measure (voronoi_open V u) + (real_of_nat (8::nat) ∗ mm1 − real_of_nat (8::nat) ∗ (mm2 ∗ sum (GSPEC (λGEN%PVAR%1722::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%1722 (IN v V ∧ u ≠ v ∧ distance (u, v) ≤ real_of_nat (2::nat) ∗ h0) v)) (λv::(real, 3) cart. lmfun (hl [u, v]))))) ⟶ fcc_compatible ?G V

thm Upfzbzm.NEGLIGIBLE_FUNC:

∀ V::(real, 3) cart ⇒ bool. saturated V ∧ packing V ∧ cell_cluster_estimate V ∧ TSKAJXY_statement ∧ lmfun_inequality V ∧ (?G::(real, 3) cart ⇒ real) = (λu::(real, 3) cart. − HOL_Light_Import.measure (voronoi_open V u) + (real_of_nat (8::nat) ∗ mm1 − real_of_nat (8::nat) ∗ (mm2 ∗ sum (GSPEC (λGEN%PVAR%1723::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%1723 (IN v V ∧ u ≠ v ∧ distance (u, v) ≤ real_of_nat (2::nat) ∗ h0) v)) (λv::(real, 3) cart. lmfun (hl [u, v]))))) ⟶ negligible_fun_0 ?G V

thm Upfzbzm.UPFZBZM:

∀ V::(real, 3) cart ⇒ bool. saturated V ∧ packing V ∧ cell_cluster_estimate V ∧ TSKAJXY_statement ∧ lmfun_inequality V ⟶ (∃ G::(real, 3) cart ⇒ real. negligible_fun_0 G V ∧ fcc_compatible G V)

thm Rdwkarc.JGXZYGW_KY:

$\forall S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge saturated\ S \wedge (\exists A::(real,\ 3)\ cart \Rightarrow real.\ fcc\_compatible\ A\ S \wedge negligible\_fun\_0\ A\ S) \longrightarrow (\exists c::real.\ \forall r \geq 1::real.\ HOL\_Light\_Import.measure\ (HOL\_Light\_Import.INTER\ (UNIONS\ (GSPEC\ (\lambda GEN\%PVAR\%1731::(real,\ 3)\ cart \Rightarrow bool.\ \exists v::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%1731\ (IN\ v\ S)\ (ball\ (v,\ 1::real))))) (ball\ (vec\ (0::nat),\ r))) /\ HOL\_Light\_Import.measure\ (ball\ (vec\ (0::nat),\ r)) \leq pi\ /\ sqrt\ (real\_of\_nat\ (18::nat)) + c\ /\ r)$

thm Rdwkarc.PACKING_SUBSET:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ S\ V \longrightarrow packing\ S$

thm Rdwkarc.PACKING_TRANS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart.\ packing\ V \longrightarrow packing\ (GSPEC\ (\lambda GEN\%PVAR\%1732::(real,\ 3)\ cart.\ \exists u::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%1732\ (IN\ (vector\_add\ u\ x)\ V)\ u))$

thm Rdwkarc.SATURATED_TRANS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart.\ saturated\ V \longrightarrow saturated\ (GSPEC\ (\lambda GEN\%PVAR\%1733::(real,\ 3)\ cart.\ \exists u::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%1733\ (IN\ (vector\_add\ u\ x)\ V)\ u))$

thm Rdwkarc.RADV_TRANS_EQ:

$\forall (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ x::(real,\ 3)\ cart.\ u \neq v \longrightarrow radV\ (INSERT\ u\ (INSERT\ v\ EMPTY)) = radV\ (INSERT\ (vector\_add\ u\ x)\ (INSERT\ (vector\_add\ v\ x)\ EMPTY))$

thm Rdwkarc.RDWKARC:

$\neg\ kepler\_conjecture \wedge (\forall V::(real,\ 3)\ cart \Rightarrow bool.\ cell\_cluster\_estimate\ V) \wedge TSKAJXY\_statement \longrightarrow (\exists V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \wedge \neg\ lmfun\_ineq\_center\ V)$

thm Wrgcvdr_cizmrrh.all_hy_map:

$(((\forall H::?'a::type\ hypermap.\ dart\ H = fst\ (tuple\_hypermap\ H)) \wedge (\forall H::?'a::type\ hypermap.\ edge\_map\ H = fst\ (snd\ (tuple\_hypermap\ H)))) \wedge (\forall H::?'a::type\ hypermap.\ node\_map\ H = fst\ (snd\ (snd\ (tuple\_hypermap\ H))))) \wedge (\forall H::?'a::type\ hypermap.\ face\_map\ H = snd\ (snd\ (snd\ (tuple\_hypermap\ H))))$

thm Wrgcvdr_cizmrrh.SPEC_HY_ELEMS:

$\forall H::?'a::type\ hypermap.\ (tuple\_hypermap\ H = (?D::?'a::type \Rightarrow bool,\ ?e::?'a::type \Rightarrow ?'a::type,\ ?n::?'a::type \Rightarrow ?'a::type,\ ?f::?'a::type \Rightarrow ?'a::type)) = (dart\ H = ?D \wedge edge\_map\ H = ?e \wedge node\_map\ H = ?n \wedge face\_map\ H = ?f)$

thm Wrgcvdr_cizmrrh.hypermap:

$\forall H::?'a::type\ hypermap.\ tuple\_hypermap\ H = (?D::?'a::type \Rightarrow bool,\ ?e::?'a::type \Rightarrow ?'a::type,\ ?n::?'a::type \Rightarrow ?'a::type,\ ?f::?'a::type \Rightarrow ?'a::type) \longrightarrow permutes\ ?e\ ?D \wedge permutes\ ?n\ ?D \wedge permutes\ ?f\ ?D \wedge ?e \circ (?n \circ ?f) = id$

thm DEF_has_orders:

$has\_orders = (\lambda(\_6497814::?'a::type \Rightarrow ?'a::type)\ \_6497815::nat.\ (\forall i::nat.\ (0::nat) < i \wedge i < \_6497815 \longrightarrow ITER\ i\ \_6497814 \neq id) \wedge ITER\ \_6497815\ \_6497814 = id)$

thm Wrgcvdr_cizmrrh.has_orders:

$\forall (k::nat)\ f::?'a::type \Rightarrow ?'a::type.\ has\_orders\ f\ k = ((\forall i::nat.\ (0::nat) < i \wedge i < k \longrightarrow ITER\ i\ f \neq id) \wedge ITER\ k\ f = id)$

thm DEF_cyclic_on:

$cyclic\_on = (\lambda(\_6497826::?'a::type \Rightarrow ?'a::type)\ \_6497827::?'a::type \Rightarrow bool.\ \forall x::?'a::type.\ IN\ x\ \_6497827 \longrightarrow \_6497827 = GSPEC\ (\lambda GEN\%PVAR\%1742::?'a::type.\ \exists z::?'a::type.\ SETSPEC\ GEN\%PVAR\%1742\ (\exists n::nat.\ z = ITER\ n\ \_6497826\ x)\ z))$

thm Wrgcvdr_cizmrrh.cyclic_on:

$\forall (S::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow ?'a::type.\ cyclic\_on\ f\ S = (\forall x::?'a::type.\ IN\ x\ S \longrightarrow S = GSPEC\ (\lambda GEN\%PVAR\%1742::?'a::type.\ \exists z::?'a::type.\ SETSPEC\ GEN\%PVAR\%1742\ (\exists n::nat.\ z = ITER\ n\ f\ x)\ z))$

thm DEF_dih2k:

$dih2k = (\lambda(\_6497838::?'a::type\ hypermap)\ \_6497839::nat.\ CARD\ (dart\ \_6497838) = (2::nat) * \_6497839 \wedge (\forall x::?'a::type.\ IN\ x\ (dart\ \_6497838) \longrightarrow LET\ (\lambda S::?'a::type \Rightarrow bool.\ LET\_END\ (dart\ \_6497838 = HOL\_Light\_Import.UNION\ S\ (IMAGE\ (node\_map\ \_6497838)\ S)))\ (face\ \_6497838\ x)) \wedge has\_orders\ (face\_map\ \_6497838)\ \_6497839 \wedge has\_orders\ (edge\_map\ \_6497838)\ (2::nat) \wedge has\_orders\ (node\_map\ \_6497838)\ (2::nat))$

thm Wrgcvdr_cizmrrh.dih2k:

$\forall (k::nat)\ H::?'a::type\ hypermap.\ dih2k\ H\ k = (CARD\ (dart\ H) = (2::nat) * k \wedge (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow LET\ (\lambda S::?'a::type \Rightarrow bool.\ LET\_END\ (dart\ H = HOL\_Light\_Import.UNION\ S\ (IMAGE\ (node\_map\ H)\ S)))\ (face\ H\ x)) \wedge has\_orders\ (face\_map\ H)\ k \wedge has\_orders\ (edge\_map\ H)\ (2::nat) \wedge has\_orders\ (node\_map\ H)\ (2::nat))$

thm DEF_EE:

$EE = (\lambda(\_6497850::?'a::type)\ \_6497851::(?'a::type \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1743::?'a::type.\ \exists w::?'a::type.\ SETSPEC\ GEN\%PVAR\%1743\ (IN\ (INSERT\ \_6497850\ (INSERT\ w\ EMPTY))\ \_6497851)\ w))$

thm Wrgcvdr_cizmrrh.EE:

$\forall (v::?'a::type)\ S::(?'a::type \Rightarrow bool) \Rightarrow bool.\ EE\ v\ S = GSPEC\ (\lambda GEN\%PVAR\%1743::?'a::type.\ \exists w::?'a::type.\ SETSPEC\ GEN\%PVAR\%1743\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ S)\ w)$

thm DEF_ord_pairs:

$ord\_pairs = (\lambda\_6497862::(?'a::type \Rightarrow bool) \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1744::?'a::type \times ?'a::type.\ \exists (a::?'a::type)\ b::?'a::type.\ SETSPEC\ GEN\%PVAR\%1744\ (IN\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ \_6497862)\ (a,\ b)))$

thm Wrgcvdr_cizmrrh.ord_pairs:

$\forall E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ ord\_pairs\ E = GSPEC\ (\lambda GEN\%PVAR\%1744::?'a::type \times ?'a::type.\ \exists (a::?'a::type)\ b::?'a::type.\ SETSPEC\ GEN\%PVAR\%1744\ (IN\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ E)\ (a,\ b))$

thm DEF_self_pairs:

$self\_pairs = (\lambda(\_6497867::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_6497868::?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1745::?'a::type \times ?'a::type.\ \exists v::?'a::type.\ SETSPEC\ GEN\%PVAR\%1745\ (IN\ v\ \_6497868 \land EE\ v\ \_6497867 = EMPTY)\ (v,\ v)))$

thm Wrgcvdr_cizmrrh.self_pairs:

$\forall (V::?'a::type \Rightarrow bool)\ E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ self\_pairs\ E\ V = GSPEC\ (\lambda GEN\%PVAR\%1745::?'a::type \times ?'a::type.\ \exists v::?'a::type.\ SETSPEC\ GEN\%PVAR\%1745\ (IN\ v\ V \land EE\ v\ E = EMPTY)\ (v,\ v))$

thm DEF_darts_of_hyp:

$darts\_of\_hyp = (\lambda(\_6497879::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_6497880::?'a::type \Rightarrow bool.\ HOL\_Light\_Import.UNION\ (ord\_pairs\ \_6497879)\ (self\_pairs\ \_6497879\ \_6497880))$

thm Wrgcvdr_cizmrrh.darts_of_hyp:

$\forall (E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ V::?'a::type \Rightarrow bool.\ darts\_of\_hyp\ E\ V = HOL\_Light\_Import.UNION\ (ord\_pairs\ E)\ (self\_pairs\ E\ V)$

thm DEF_ee_of_hyp:

$ee\_of\_hyp = (\lambda(\_6497891::?'a::type \times ((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool))\ \_6497892::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ if\ IN\ (fst\ \_6497892,\ snd\ \_6497892)\ (darts\_of\_hyp\ (snd\ (snd\ \_6497891))\ (fst\ (snd\ \_6497891)))\ then\ (snd\ \_6497892,\ fst\ \_6497892)\ else\ (fst\ \_6497892,\ snd\ \_6497892))$

thm Wrgcvdr_cizmrrh.ee_of_hyp:

$\forall (x::?'a::type)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (a::(real,\ 3)\ cart)\ b::(real,\ 3)\ cart.\ ee\_of\_hyp\ (x,\ V,\ E)\ (a,\ b) = (if\ IN\ (a,\ b)\ (darts\_of\_hyp\ E\ V)\ then\ (b,\ a)\ else\ (a,\ b))$

thm Wrgcvdr_cizmrrh.ee_of_hyp2:

ee_of_hyp (*?x*::*?'a*::*type*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?u*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*if IN ?u* (*darts_of_hyp ?E ?V*) *then* (*snd ?u*, *fst ?u*) *else ?u*)

thm DEF_nn_of_hyp:

nn_of_hyp = (λ(*_6497918*::(*real*, *3*) *cart* × ((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) *_6497919*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *if IN* (*fst _6497919*, *snd _6497919*) (*darts_of_hyp* (*snd* (*snd _6497918*)) (*fst* (*snd _6497918*))) *then* (*fst _6497919*, *azim_cycle* (*EE* (*fst _6497919*) (*snd* (*snd _6497918*))) (*fst _6497918*) (*fst _6497919*) (*snd _6497919*)) *else* (*fst _6497919*, *snd _6497919*))

thm Wrgcvdr_cizmrrh.nn_of_hyp:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *nn_of_hyp* (*x*, *V*, *E*) (*v*, *u*) = (*if IN* (*v*, *u*) (*darts_of_hyp E V*) *then* (*v*, *azim_cycle* (*EE v E*) *x v u*) *else* (*v*, *u*))

thm Wrgcvdr_cizmrrh.nn_of_hyp2:

nn_of_hyp (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?u*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*if IN ?u* (*darts_of_hyp ?E ?V*) *then* (*fst ?u*, *azim_cycle* (*EE* (*fst ?u*) *?E*) *?x* (*fst ?u*) (*snd ?u*)) *else ?u*)

thm DEF_ivs_azim_cycle:

ivs_azim_cycle = (λ(*_6497945*::(*real*, *3*) *cart* ⇒ *bool*) (*_6497946*::(*real*, *3*) *cart*) (*_6497947*::(*real*, *3*) *cart*) *_6497948*::(*real*, *3*) *cart*. *if _6497945* = *EMPTY then _6497948 else SOME x*::(*real*, *3*) *cart*. *IN x _6497945* ∧ *azim_cycle _6497945 _6497946 _6497947 x* = *_6497948*)

thm Wrgcvdr_cizmrrh.ivs_azim_cycle:

∀ (*W*::(*real*, *3*) *cart* ⇒ *bool*) (*v0*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *ivs_azim_cycle W v0 v w* = (*if W* = *EMPTY then w else SOME x*::(*real*, *3*) *cart*. *IN x W* ∧ *azim_cycle W v0 v x* = *w*)

thm DEF_ff_of_hyp:

ff_of_hyp = (λ(*_6497977*::(*real*, *3*) *cart* × ((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) *_6497978*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *if IN* (*fst _6497978*, *snd _6497978*) (*darts_of_hyp* (*snd* (*snd _6497977*)) (*fst* (*snd _6497977*))) *then* (*snd _6497978*, *ivs_azim_cycle* (*EE* (*snd _6497978*) (*snd* (*snd _6497977*))) (*fst _6497977*) (*snd _6497978*) (*fst _6497978*)) *else* (*fst _6497978*, *snd _6497978*))

thm Wrgcvdr_cizmrrh.ff_of_hyp:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *u*::(*real*, *3*) *cart*. *ff_of_hyp* (*x*, *V*, *E*) (*v*, *u*) = (*if IN* (*v*, *u*) (*darts_of_hyp E V*) *then* (*u*, *ivs_azim_cycle* (*EE u E*) *x u v*) *else* (*v*, *u*))

thm Wrgcvdr_cizmrrh.ff_of_hyp2:

ff_of_hyp (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?u*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*if IN ?u* (*darts_of_hyp ?E ?V*) *then* (*snd ?u*, *ivs_azim_cycle* (*EE* (*snd ?u*) *?E*) *?x* (*snd ?u*) (*fst ?u*)) *else ?u*)

thm DEF_HYP:

HYP = (λ*_6498004*::(*real*, *3*) *cart* × ((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*). (*darts_of_hyp* (*snd* (*snd _6498004*)) (*fst* (*snd _6498004*)), *ee_of_hyp* (*fst _6498004*, *fst* (*snd _6498004*), *snd* (*snd _6498004*)), *nn_of_hyp* (*fst _6498004*, *fst* (*snd _6498004*), *snd* (*snd _6498004*)), *ff_of_hyp* (*fst _6498004*, *fst* (*snd _6498004*), *snd* (*snd _6498004*))))

thm Wrgcvdr_cizmrrh.HYP:

∀(*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *HYP* (*x*, *V*, *E*) = (*darts_of_hyp E V*, *ee_of_hyp* (*x*, *V*, *E*), *nn_of_hyp* (*x*, *V*, *E*), *ff_of_hyp* (*x*, *V*, *E*))

thm DEF_local_fan:

local_fan = (λ*_6498017*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) × ((*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*). *LET* (λ*H*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*. *LET_END* (*FAN* (*vec* (*0*::*nat*), *fst _6498017*, *fst* (*snd _6498017*)) ∧ (∃*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*dart H*) ∧ *snd* (*snd _6498017*) = *face H x*) ∧ *dih2k H* (*CARD* (*snd* (*snd _6498017*))))) (*hypermap* (*HYP* (*vec* (*0*::*nat*), *fst _6498017*, *fst* (*snd _6498017*)))))

thm Wrgcvdr_cizmrrh.local_fan:

∀(*FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *local_fan* (*V*, *E*, *FF*) = *LET* (λ*H*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*. *LET_END* (*FAN* (*vec* (*0*::*nat*), *V*, *E*) ∧ (∃*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*dart H*) ∧ *FF* = *face H x*) ∧ *dih2k H* (*CARD FF*))) (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V*, *E*)))

thm DEF_azim_in_fan:

azim_in_fan = (λ(*_6498030*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *_6498031*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *LET* (λ*d*::(*real*, *3*) *cart*. *LET_END* (*if* (*1*::*nat*) < *CARD* (*EE* (*fst _6498030*) *_6498031*) *then azim* (*vec* (*0*::*nat*)) (*fst _6498030*) (*snd _6498030*) *d else real_of_nat* (*2*::*nat*) ∗ *pi*)) (*azim_cycle* (*EE* (*fst _6498030*) *_6498031*) (*vec* (*0*::*nat*)) (*fst _6498030*) (*snd _6498030*)))

thm Wrgcvdr_cizmrrh.azim_in_fan:

∀(*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *azim_in_fan* (*v*, *w*) *E* = *LET* (λ*d*::(*real*, *3*) *cart*. *LET_END* (*if* (*1*::*nat*) <

*CARD (EE v E) then azim (vec (0::nat)) v w d else real_of_nat (2::nat) * pi))
(azim_cycle (EE v E) (vec (0::nat)) v w)*

thm DEF_wedge_in_fan_gt:

wedge_in_fan_gt = (λ(_6498047::(real, 3) cart × (real, 3) cart) _6498048::((real, 3) cart ⇒ bool) ⇒ bool. if (1::nat) < CARD (EE (fst _6498047) _6498048) then wedge (vec (0::nat)) (fst _6498047) (snd _6498047) (azim_cycle (EE (fst _6498047) _6498048) (vec (0::nat)) (fst _6498047) (snd _6498047)) else if EE (fst _6498047) _6498048 = INSERT (snd _6498047) EMPTY then GSPEC (λGEN%PVAR%1746::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1746 (¬ IN x (aff_ge (INSERT (vec (0::nat)) (INSERT (fst _6498047) EMPTY)) (INSERT (snd _6498047) EMPTY))) x) else GSPEC (λGEN%PVAR%1747::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1747 (¬ IN x (aff (INSERT (vec (0::nat)) (INSERT (fst _6498047) EMPTY)))) x))

thm Wrgcvdr_cizmrrh.wedge_in_fan_gt:

∀(E::((real, 3) cart ⇒ bool) ⇒ bool) (w::(real, 3) cart) v::(real, 3) cart. wedge_in_fan_gt (v, w) E = (if (1::nat) < CARD (EE v E) then wedge (vec (0::nat)) v w (azim_cycle (EE v E) (vec (0::nat)) v w) else if EE v E = INSERT w EMPTY then GSPEC (λGEN%PVAR%1746::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1746 (¬ IN x (aff_ge (INSERT (vec (0::nat)) (INSERT v EMPTY)) (INSERT w EMPTY))) x) else GSPEC (λGEN%PVAR%1747::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1747 (¬ IN x (aff (INSERT (vec (0::nat)) (INSERT v EMPTY)))) x))

thm DEF_wedge_ge:

wedge_ge = (λ(_6498064::(real, 3) cart) (_6498065::(real, 3) cart) (_6498066::(real, 3) cart) _6498067::(real, 3) cart. GSPEC (λGEN%PVAR%1748::(real, 3) cart. ∃z::(real, 3) cart. SETSPEC GEN%PVAR%1748 ((0::real) ≤ azim _6498064 _6498065 _6498066 z ∧ azim _6498064 _6498065 _6498066 z ≤ azim _6498064 _6498065 _6498066 _6498067) z))

thm Wrgcvdr_cizmrrh.wedge_ge:

∀(v0::(real, 3) cart) (v1::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. wedge_ge v0 v1 w1 w2 = GSPEC (λGEN%PVAR%1748::(real, 3) cart. ∃z::(real, 3) cart. SETSPEC GEN%PVAR%1748 ((0::real) ≤ azim v0 v1 w1 z ∧ azim v0 v1 w1 z ≤ azim v0 v1 w1 w2) z)

thm DEF_wedge_in_fan_ge:

wedge_in_fan_ge = (λ(_6498096::(real, 3) cart × (real, 3) cart) _6498097::((real, 3) cart ⇒ bool) ⇒ bool. if (1::nat) < CARD (EE (fst _6498096) _6498097) then wedge_ge (vec (0::nat)) (fst _6498096) (snd _6498096) (azim_cycle (EE (fst _6498096) _6498097) (vec (0::nat)) (fst _6498096) (snd _6498096)) else GSPEC (λGEN%PVAR%1749::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1749 True x))

thm Wrgcvdr_cizmrrh.wedge_in_fan_ge:

∀(E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. wedge_in_fan_ge (v, w) E = (if (1::nat) < CARD (EE v E) then wedge_ge (vec (0::nat)) v w (azim_cycle (EE v E) (vec (0::nat)) v w) else GSPEC (λGEN%PVAR%1749::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1749 True x))

thm DEF_convex_local_fan:

convex_local_fan = (λ_6498113::((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool) × ((real, 3) cart × (real, 3) cart ⇒ bool). local_fan (fst _6498113, fst (snd _6498113), snd (snd _6498113)) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x (snd (snd _6498113)) ⟶ azim_in_fan x (fst (snd _6498113)) ≤ pi ∧ SUBSET (fst _6498113) (wedge_in_fan_ge x (fst (snd _6498113)))))

thm Wrgcvdr_cizmrrh.convex_local_fan:

∀(FF::(real, 3) cart × (real, 3) cart ⇒ bool) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. convex_local_fan (V, E, FF) = (local_fan (V, E, FF) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x FF ⟶ azim_in_fan x E ≤ pi ∧ SUBSET V (wedge_in_fan_ge x E)))

thm Wrgcvdr_cizmrrh.FST_SND_FORM_OF_4_TUPLE:

∀(X::?'d::type × ?'c::type × ?'b::type × ?'a::type) (D::?'d::type) (e::?'c::type) (n::?'b::type) f::?'a::type. (fst X = D ∧ fst (snd X) = e ∧ fst (snd (snd X)) = n ∧ snd (snd (snd X)) = f) = (X = (D, e, n, f))

thm Wrgcvdr_cizmrrh.V_IN_DARTS_IFF_E_V_IN_DARTS:

IN (?v::(real, 3) cart × (real, 3) cart) (darts_of_hyp (?E::((real, 3) cart ⇒ bool) ⇒ bool) (?V::(real, 3) cart ⇒ bool)) = IN (ee_of_hyp (?x::?'a::type, ?V, ?E) ?v) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.V_IN_DARTS_IMP_SWICH_SO_DO:

IN (?v::?'a::type × ?'a::type) (darts_of_hyp (?E::(?'a::type ⇒ bool) ⇒ bool) (?V::?'a::type ⇒ bool)) ⟶ IN (snd ?v, fst ?v) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.IN_V_OF_FAN_EXISTS_DART:

∀(u::(real, 3) cart) (E::((real, 3) cart ⇒ bool) ⇒ bool) V::(real, 3) cart ⇒ bool. SUBSET (UNIONS E) V ∧ IN u V ⟶ (∃v::(real, 3) cart. IN v V ∧ IN (u, v) (darts_of_hyp E V))

thm Wrgcvdr_cizmrrh.X_IN_HYP_ORBITS:

∀x::?'a::type. IN x (edge (?H::?'a::type hypermap) x) ∧ IN x (node ?H x) ∧ IN x (face ?H x)

thm Wrgcvdr_cizmrrh.IN_DARTS_HYP_IMP_FST_SND_IN_V:

SUBSET (UNIONS (?E::(?'a::type ⇒ bool) ⇒ bool)) (?V::?'a::type ⇒ bool) ∧ IN (?y::?'a::type × ?'a::type) (darts_of_hyp ?E ?V) ⟶ IN (fst ?y) ?V ∧ IN (snd ?y) ?V

thm Wrgcvdr_cizmrrh.POWER_SND:

$\forall$ *n*::*nat*. *POWER* (*?f*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) (*Suc n*) = *?f* $\circ$ *POWER ?f n*

thm Wrgcvdr_cizmrrh.POWER_TO_ITER:

$\forall$ *n*::*nat*. *POWER* (*?f*::*?'a*::*type* $\Rightarrow$ *?'a*::*type*) *n* = *ITER n ?f*

thm Wrgcvdr_cizmrrh.SND_IN_SET_OF_DART_OF_FRST:

SUBSET (*UNIONS* (*?E*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) (*?V*::*?'a*::*type* $\Rightarrow$ *bool*) $\wedge$ *IN* (*?y*::*?'a*::*type* $\times$ *?'a*::*type*) (*darts_of_hyp ?E ?V*) $\longrightarrow$ *fst ?y* = *snd ?y* $\vee$ *IN* (*snd ?y*) (*set_of_edge* (*fst ?y*) *?V ?E*)

thm Wrgcvdr_cizmrrh.UNI_E_IMP_EE_EQ_SET_OF_EDGE:

SUBSET (*UNIONS* (*?E*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) (*?V*::*?'a*::*type* $\Rightarrow$ *bool*) $\longrightarrow$ *EE* (*?v*::*?'a*::*type*) *?E* = *set_of_edge ?v ?V ?E*

thm Wrgcvdr_cizmrrh.IN_ORD_PAIRS_IMP_IMP_IN_TOO:

IN (*?y*::*?'a*::*type* $\times$ *?'a*::*type*) (*ord_pairs* (*?E*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) $\longrightarrow$ *IN* (*fst ?y*, *?d*::*?'a*::*type*) (*darts_of_hyp ?E* (*?V*::*?'a*::*type* $\Rightarrow$ *bool*)) $\longrightarrow$ *IN* (*fst ?y*, *?d*) (*ord_pairs ?E*)

thm Wrgcvdr_cizmrrh.IN_ORD_PAIRS_IMP_SND_IN_EE_FST:

IN (*?y*::*?'a*::*type* $\times$ *?'a*::*type*) (*ord_pairs* (*?E*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) $\longrightarrow$ *IN* (*snd ?y*) (*EE* (*fst ?y*) *?E*)

thm Wrgcvdr_cizmrrh.IN_SELF_PAIRS_IMP_FST_EQ_SND_FORALL:

$\forall$ *v*::*?'a*::*type*. *IN* (*?y*::*?'a*::*type* $\times$ *?'a*::*type*) (*self_pairs* (*?E*::(*?'a*::*type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) (*?V*::*?'a*::*type* $\Rightarrow$ *bool*)) $\wedge$ *IN* (*fst ?y*, *v*) (*darts_of_hyp ?E ?V*) $\longrightarrow$ *fst ?y* = *v*

thm DEF_choose_nd_point:

choose_nd_point = (*SOME v*::*nat* $\Rightarrow$ (*real*, *3*) *cart* $\Rightarrow$ (((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) $\Rightarrow$ ((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ (*real*, *3*) *cart*. $\forall$ (*_6498580*::*nat*) (*u*::(*real*, *3*) *cart*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *SUBSET* (*UNIONS E*) *V* $\wedge$ *IN u V* $\longrightarrow$ *IN* (*v _6498580 u E V*) *V* $\wedge$ *IN* (*u*, *v _6498580 u E V*) (*darts_of_hyp E V*)) (*140*::*nat*)

thm Wrgcvdr_cizmrrh.choose_nd_point:

$\forall$ (*u*::(*real*, *3*) *cart*) (*E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *SUBSET* (*UNIONS E*) *V* $\wedge$ *IN u V* $\longrightarrow$ *IN* (*choose_nd_point u E V*) *V* $\wedge$ *IN* (*u*, *choose_nd_point u E V*) (*darts_of_hyp E V*)

thm Wrgcvdr_cizmrrh.ITER_N_I:

$\forall$ *n*::*nat*. *ITER n id* = *id*

thm Wrgcvdr_cizmrrh.HAS_ORDERS_IMP_ORBIT_MAP_FIRST_ROW:

has_orders (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?k*::*nat*) ∧ *?k* ≠ (*0*::*nat*) ⟶ (∀ *x*::*?'a*::*type*. *orbit_map ?f x* = *GSPEC* (λ*GEN%PVAR%1750*::*?'a*::*type*. ∃ *y*::*?'a*::*type*. *SETSPEC GEN%PVAR%1750* (∃ *n*≥*0*::*nat*. *n* < *?k* ∧ *y* = *ITER n ?f x*) *y*))

thm Wrgcvdr_cizmrrh.FINITE_OF_N_FIRST_ELMS:

∀ (*k*::*nat*) *x*::*?'a*::*type*. *FINITE* (*GSPEC* (λ*GEN%PVAR%1755*::*?'a*::*type*. ∃ *y*::*?'a*::*type*. *SETSPEC GEN%PVAR%1755* (∃ *n*<*k*. *y* = *ITER n* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*) *y*))

thm Wrgcvdr_cizmrrh.F_HAS_ORDERS_IMP_FINITE_ORBIT:

has_orders (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?k*::*nat*) ∧ *?k* ≠ (*0*::*nat*) ⟶ (∀ *x*::*?'a*::*type*. *FINITE* (*orbit_map ?f x*))

thm Wrgcvdr_cizmrrh.CARD_CLAUSES2:

FINITE (*?S*::*?'a*::*type* ⇒ *bool*) ⟶ (∀ *x*::*?'a*::*type*. *IN x ?S* ⟶ *CARD* (*INSERT x ?S*) = *CARD ?S*) ∧ (∀ *x*::*?'a*::*type*. ¬ *IN x ?S* ⟶ *CARD* (*INSERT x ?S*) = *Suc* (*CARD ?S*))

thm Wrgcvdr_cizmrrh.CARD_INSERT_GE_AND_LE:

FINITE (*?S*::*?'a*::*type* ⇒ *bool*) ⟶ *CARD ?S* ≤ *CARD* (*INSERT* (*?x*::*?'a*::*type*) *?S*) ∧ *CARD* (*INSERT ?x ?S*) ≤ *Suc* (*CARD ?S*)

thm Wrgcvdr_cizmrrh.HAVING_ORDERS_K_IMP_CARD_ORBIT_LE_K:

has_orders (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?k*::*nat*) ∧ *?k* ≠ (*0*::*nat*) ⟶ (∀ *x*::*?'a*::*type*. *CARD* (*orbit_map ?f x*) ≤ *?k*)

thm Wrgcvdr_cizmrrh.CARD_LE_K_OF_SET_K_FIRST_ELMS:

CARD (*GSPEC* (λ*GEN%PVAR%332*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%332* (*i* < (*?k*::*nat*)) (*ITER i* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?x*::*?'a*::*type*)))) ≤ *?k*

thm Wrgcvdr_cizmrrh.CARD_K_FIRST_ELMS_EQ_K:

∀ (*k*::*nat*) *f*::*?'a*::*type* ⇒ *?'a*::*type*. *CARD* (*GSPEC* (λ*GEN%PVAR%1772*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1772* (*n* < *k*) (*ITER n f* (*?x*::*?'a*::*type*)))) = *k* ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%1773*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1773* (*n* < *k* − (*1*::*nat*)) (*ITER n f ?x*))) = *k* − (*1*::*nat*) ∧ (∀ *i*<*k* − (*1*::*nat*). *ITER i f ?x* ≠ *ITER* (*k* − (*1*::*nat*)) *f ?x*)

thm Wrgcvdr_cizmrrh.FINITENESS_OF_K_FIRST_ELMS:

∀ (*k*::*nat*) *f*::*nat* ⇒ *?'a*::*type*. *FINITE* (*GSPEC* (λ*GEN%PVAR%1777*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1777* (*n* < *k*) (*f n*)))

thm Wrgcvdr_cizmrrh.LT_SUC_E:

((*?i*::*nat*) < *Suc* (*?k*::*nat*)) = (*?i* < *?k* ∨ *?i* = *?k*)

thm Wrgcvdr_cizmrrh.CARD_K_FIRST_ELMS_LE_K:

∀ *k*::*nat*. *CARD* (*GSPEC* (λ*GEN%PVAR%332*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%332* (*i* < *k*) ((*?f*::*nat* ⇒ *?'a*::*type*) *i*))) ≤ *k*

thm Wrgcvdr_cizmrrh.CARD_N_FIRST_ELMS_UNDUCTIVE:

$\forall (k::nat)\ f::nat \Rightarrow ?'a::type.\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1781::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1781\ (n < k)\ (f\ n))) = k) = (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1782::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1782\ (n < k - (1::nat))\ (f\ n))) = k - (1::nat) \wedge (\forall i<k - (1::nat).\ f\ i \neq f\ (k - (1::nat))))$

thm Wrgcvdr_cizmrrh.CARD_ADD1_LE:

$\forall f::nat \Rightarrow ?'a::type.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1785::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1785\ (n < (?k::nat) + (1::nat))\ (f\ n))) \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1786::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1786\ (n < ?k)\ (f\ n))) + (1::nat)$

thm Wrgcvdr_cizmrrh.CARD_LT_KT_LE_ADDT:

$CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1787::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1787\ (n < (?k::nat) + (?t::nat))\ ((?f::nat \Rightarrow ?'a::type)\ n))) \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1788::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1788\ (n < ?k)\ (?f\ n))) + ?t$

thm Wrgcvdr_cizmrrh.CARD_KS_EQ_K_EQ_ALL_LE:

$\forall (k::nat)\ f::nat \Rightarrow ?'a::type.\ (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1793::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1793\ (n < k)\ (f\ n))) = k) = (\forall kk \leq k.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1794::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1794\ (n < kk)\ (f\ n))) = kk)$

thm Wrgcvdr_cizmrrh.CARD_K_ELMS_EQ_K_IMP_ALL_DISTINCT:

$\forall (k::nat)\ f::nat \Rightarrow ?'a::type.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1795::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1795\ (n < k)\ (f\ n))) = k \longrightarrow (\forall (i::nat)\ j::nat.\ i < k \wedge j < k \wedge i \neq j \longrightarrow f\ i \neq f\ j)$

thm Wrgcvdr_cizmrrh.CARD_UNION_NOT_DISTJ_LT:

$FINITE\ (?s::?'a::type \Rightarrow bool) \wedge FINITE\ (?t::?'a::type \Rightarrow bool) \wedge HOL\_Light\_Import.INTER\ ?s\ ?t \neq EMPTY \longrightarrow CARD\ (HOL\_Light\_Import.UNION\ ?s\ ?t) < CARD\ ?s + CARD\ ?t$

thm Wrgcvdr_cizmrrh.CARD_ITER_K_EK_IMP_DIST:

$\forall (k::nat)\ f::?'a::type \Rightarrow ?'a::type.\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%1796::?'a::type.\ \exists n::nat.\ SETSPEC\ GEN\%PVAR\%1796\ (n < k)\ (ITER\ n\ f\ (?x::?'a::type)))) = k \longrightarrow (\forall (i::nat)\ j::nat.\ i < k \wedge j < k \wedge i \neq j \longrightarrow ITER\ i\ f\ ?x \neq ITER\ j\ f\ ?x)$

thm Wrgcvdr_cizmrrh.DIH2K_IMP_PRE_SIMPLE_HYP:

$FINITE\ (dart\ (?H::?'a::type\ hypermap)) \wedge dih2k\ ?H\ (?k::nat) \wedge ?k \neq (0::nat) \longrightarrow (\forall x::?'a::type.\ IN\ x\ (dart\ ?H) \longrightarrow \neg\ IN\ (node\_map\ ?H\ x)\ (face\ ?H\ x))$

thm Wrgcvdr_cizmrrh.ITER1:

ITER (1::nat) (?f::?'a::type ⇒ ?'a::type) = ?f

thm Wrgcvdr_cizmrrh.DIH2K_IMP_SIMPLE_HYPERMAP:

FINITE (dart (?H::?'a::type hypermap)) ∧ dih2k ?H (?k::nat) ∧ ?k ≠ (0::nat) ⟶ simple_hypermap ?H

thm Wrgcvdr_cizmrrh.IN_ORBIT_IMP_ORBIT_SUBSET:

∀(x::?'a::type) (y::?'a::type) f::?'a::type ⇒ ?'a::type. IN x (orbit_map f y) ⟶ SUBSET (orbit_map f x) (orbit_map f y)

thm Wrgcvdr_cizmrrh.IN_FACE_IMP_SUBSET_FACE:

∀x::?'a::type. IN x (face (?H::?'a::type hypermap) (?y::?'a::type)) ⟶ SUBSET (face ?H x) (face ?H ?y)

thm Wrgcvdr_cizmrrh.HAS_ORDK_IN_ORBIT_IMP_SAME_ORBIT:

has_orders (?f::?'a::type ⇒ ?'a::type) (?k::nat) ∧ IN (?x::?'a::type) (orbit_map ?f (?y::?'a::type)) ∧ ?k ≠ (0::nat) ⟶ orbit_map ?f ?x = orbit_map ?f ?y

thm Wrgcvdr_cizmrrh.DIH_IMP_EVERY_NODE_INTER_FACE:

dih2k (?H::?'a::type hypermap) (?k::nat) ⟶ (∀(x::?'a::type) y::?'a::type. SUBSET (INSERT x (INSERT y EMPTY)) (dart ?H) ⟶ (∃d::?'a::type. IN d (node ?H x) ∧ IN d (face ?H y)))

thm Wrgcvdr_cizmrrh.F_INVERSE_F:

(∃y::?'b::type. (?f::?'b::type ⇒ ?'a::type) y = (?x::?'a::type)) ⟶ ?f (HOL_Light_Import.inverse ?f ?x) = ?x

thm Wrgcvdr_cizmrrh.F_INVERSE_F_F:

(?f::?'a::type ⇒ ?'b::type) (HOL_Light_Import.inverse ?f (?f (?x::?'a::type))) = ?f ?x

thm Wrgcvdr_cizmrrh.INJ_IMP_INVERSE_FF:

(∀y::?'b::type. (?f::?'b::type ⇒ ?'a::type) y = ?f (?x::?'b::type) ⟶ y = ?x) ⟶ HOL_Light_Import.inverse ?f (?f ?x) = ?x

thm Wrgcvdr_cizmrrh.BIJ_AND_BIJ_INVERSE:

BIJ (?f::?'b::type ⇒ ?'a::type) (?S1.0::?'b::type ⇒ bool) (?S2.0::?'a::type ⇒ bool) ∧ (∀x::?'b::type. IN (?f x) ?S2.0 ⟶ IN x ?S1.0) ⟶ BIJ (HOL_Light_Import.inverse ?f) ?S2.0 ?S1.0

thm Wrgcvdr_cizmrrh.INVERSE_FUNCTION_OF_BIJ:

BIJ (?f::?'b::type ⇒ ?'a::type) (?S1.0::?'b::type ⇒ bool) (?S2.0::?'a::type ⇒ bool) ∧ (?g::?'a::type ⇒ ?'b::type) = (λx::?'a::type. if IN x ?S2.0 then SOME t::?'b::type. IN t ?S1.0 ∧ ?f t = x else (?tt::?'b::type)) ⟶ BIJ ?g ?S2.0 ?S1.0

thm Wrgcvdr_cizmrrh.TOW_BIJS_IMP_BIJ_BETWEEN_FIRST:

BIJ (?f::?'c::type ⇒ ?'b::type) (?S1.0::?'c::type ⇒ bool) (?V::?'b::type ⇒ bool) ∧ BIJ (?g::?'a::type ⇒ ?'b::type) (?S2.0::?'a::type ⇒ bool) ?V ∧ (?ff::?'c::type ⇒ ?'a::type) = (λx::?'c::type. if IN x ?S1.0 then SOME a::?'a::type. IN a ?S2.0 ∧ ?f x = ?g a else (?aa::?'a::type)) ⟶ BIJ ?ff ?S1.0 ?S2.0

thm Wrgcvdr_cizmrrh.INDENT_IN_S1_IMP_BIJ:

BIJ (?f::?'b::type ⇒ ?'a::type) (?S1.0::?'b::type ⇒ bool) (?S2.0::?'a::type ⇒ bool) ∧ (∀x::?'b::type. IN x ?S1.0 ⟶ ?f x = (?g::?'b::type ⇒ ?'a::type) x) ⟶ BIJ ?g ?S1.0 ?S2.0

thm Wrgcvdr_cizmrrh.LOCAL_FAN_IMP_FAN:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ FAN (vec (0::nat), ?V, ?E)

thm Wrgcvdr_cizmrrh.IN_ORBIT_MAP_IMP_F_Y:

IN (?y::?'a::type) (orbit_map (?f::?'a::type ⇒ ?'a::type) (?x::?'a::type)) ⟶ IN (?f ?y) (orbit_map ?f ?x)

thm Wrgcvdr_cizmrrh.SURJ_IMP_S2_EQ_IMAGE_S1:

SURJ (?f::?'b::type ⇒ ?'a::type) (?S1.0::?'b::type ⇒ bool) (?S2.0::?'a::type ⇒ bool) ⟶ IMAGE ?f ?S1.0 = ?S2.0

thm Wrgcvdr_cizmrrh.CYCLIC_SET_IMP_NOT_COLLINEAR:

cyclic_set (?W::(real, 3) cart ⇒ bool) (?x::(real, 3) cart) (?y::(real, 3) cart) ⟶ (∀v::(real, 3) cart. IN v ?W ⟶ ¬ collinear (INSERT v (INSERT ?x (INSERT ?y EMPTY))))

thm Wrgcvdr_cizmrrh.SLIDABLE_PROJECTION:

(?e::(real, ?'a::type) cart) ≠ vec (0::nat) ⟶ projection ?e (vector_add (% (?t::real) ?e) (?x::(real, ?'a::type) cart)) = projection ?e ?x

thm Wrgcvdr_cizmrrh.LINEAR_PROJECTION:

projection (?e::(real, ?'a::type) cart) (% (?t::real) (?x::(real, ?'a::type) cart)) = % ?t (projection ?e ?x)

thm Wrgcvdr_cizmrrh.IDENTIFY_AZIM_CYCLE:

¬ SUBSET (?W::(real, 3) cart ⇒ bool) (INSERT (?p::(real, 3) cart) EMPTY) ∧ ¬ collinear (INSERT ?p (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY))) ∧ cyclic_set ?W ?v ?w ∧ ((?u::(real, 3) cart) ≠ ?p ∧ ?W ?u) ∧ (∀q::(real, 3) cart. q ≠ ?p ∧ ?W q ⟶ azim ?v ?w ?p ?u < azim ?v ?w ?p q ∨ azim ?v ?w ?p ?u = azim ?v ?w ?p q ∧ vector_norm (projection (vector_sub ?w ?v) (vector_sub ?u ?v)) ≤ vector_norm (projection (vector_sub ?w ?v) (vector_sub q ?v))) ⟶ azim_cycle ?W ?v ?w ?p = ?u

thm Wrgcvdr_cizmrrh.EXISTS_SMALLEST_ELMS:

FINITE (?W::?'a::type ⇒ bool) ∧ ?W ≠ EMPTY ∧ (∀(x::?'a::type) y::?'a::type. (?ll::?'a::type ⇒ ?'a::type ⇒ bool) x y ∨ ?ll y x) ∧ (∀(x::?'a::type) (y::?'a::type) *z::?'a::type. ?ll x y ∧ ?ll y z ⟶ ?ll x z) ⟶ (∃ v::?'a::type. IN v ?W ∧ (∀ w::?'a::type. IN w ?W ⟶ ?ll v w))*

thm Wrgcvdr_cizmrrh.EXIS_SMALLEST_WITH_AZIM_ORD:

¬ SUBSET (?W::(real, 3) cart ⇒ bool) (INSERT (?p::(real, 3) cart) EMPTY) ∧ FINITE ?W ⟶ (∃ u::(real, 3) cart. (u ≠ ?p ∧ ?W u) ∧ (∀ q::(real, 3) cart. q ≠ ?p ∧ ?W q ⟶ azim (?v::(real, 3) cart) (?w::(real, 3) cart) ?p u < azim ?v ?w ?p q ∨ azim ?v ?w ?p u = azim ?v ?w ?p q ∧ vector_norm (projection (vector_sub ?w ?v) (vector_sub u ?v)) ≤ vector_norm (projection (vector_sub ?w ?v) (vector_sub q ?v))))

thm Wrgcvdr_cizmrrh.AZIM_CYCLE_PROPERTIES:

¬ SUBSET (?W::(real, 3) cart ⇒ bool) (INSERT (?p::(real, 3) cart) EMPTY) ∧ FINITE ?W ⟶ azim_cycle ?W (?v::(real, 3) cart) (?w::(real, 3) cart) ?p ≠ ?p ∧ ?W (azim_cycle ?W ?v ?w ?p) ∧ (∀ q::(real, 3) cart. q ≠ ?p ∧ ?W q ⟶ azim ?v ?w ?p (azim_cycle ?W ?v ?w ?p) < azim ?v ?w ?p q ∨ azim ?v ?w ?p (azim_cycle ?W ?v ?w ?p) = azim ?v ?w ?p q ∧ vector_norm (projection (vector_sub ?w ?v) (vector_sub (azim_cycle ?W ?v ?w ?p) ?v)) ≤ vector_norm (projection (vector_sub ?w ?v) (vector_sub q ?v)))

thm Wrgcvdr_cizmrrh.PROJECT_EQ_VEC0_IMP_PARALLED:

projection (?e::(real, ?'a::type) cart) (?x::(real, ?'a::type) cart) = vec (0::nat) ⟶ (∃ t::real. ?x = % t ?e)

thm Wrgcvdr_cizmrrh.PROJECTION_VEC0:

projection (?e::(real, ?'a::type) cart) (vec (0::nat)) = vec (0::nat)

thm Wrgcvdr_cizmrrh.LE_ORD_IS_ASSYMETRY:

LET (λle::(real, 3) cart ⇒ (real, 3) cart ⇒ bool. LET_END (cyclic_set (?W::(real, 3) cart ⇒ bool) (?v::(real, 3) cart) (?w::(real, 3) cart) ∧ SUBSET (INSERT (?x::(real, 3) cart) (INSERT (?y::(real, 3) cart) EMPTY)) ?W ∧ le ?x ?y ∧ le ?y ?x ⟶ ?x = ?y)) (λ(u::(real, 3) cart) q::(real, 3) cart. azim ?v ?w (CHOICE ?W) u < azim ?v ?w (CHOICE ?W) q ∨ azim ?v ?w (CHOICE ?W) u = azim ?v ?w (CHOICE ?W) q ∧ vector_norm (projection (vector_sub ?w ?v) (vector_sub u ?v)) ≤ vector_norm (projection (vector_sub ?w ?v) (vector_sub q ?v)))

thm Wrgcvdr_cizmrrh.LE_ORD_IS_ASSYMETRY2:

cyclic_set (?W::(real, 3) cart ⇒ bool) (?v::(real, 3) cart) (?w::(real, 3) cart) ∧ SUBSET (INSERT (?x::(real, 3) cart) (INSERT (?y::(real, 3) cart) EMPTY)) ?W ∧ azim ?v ?w (CHOICE ?W) ?x = azim ?v ?w (CHOICE ?W) ?y ∧ vector_norm (projection (vector_sub ?w ?v) (vector_sub ?x ?v)) = vector_norm (projection (vector_sub ?w ?v) (vector_sub ?y ?v)) ⟶ ?x = ?y

thm Lvducxu.W_SUBSET_SINGLETON_IMP_IDE:

SUBSET (?W::(real, 3) cart ⇒ bool) (INSERT (?p::(real, 3) cart) EMPTY) ⟶ azim_cycle ?W (?v::(real, 3) cart) (?w::(real, 3) cart) ?p = ?p

thm Wrgcvdr_cizmrrh.AZIM_CYCLE_EQ_SIGMA_FAN:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?u*::(*real*, *3*) *cart*) (*set_of_edge* (*?v*::(*real*, *3*) *cart*) *?V ?E*) ⟶ *azim_cycle* (*EE ?v ?E*) *?x ?v ?u* = *sigma_fan ?x ?V ?E ?v ?u*

thm Wrgcvdr_cizmrrh.IVS_AZIM_AS_SIGMA_FAN:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?u*::(*real*, *3*) *cart*) (*set_of_edge* (*?v*::(*real*, *3*) *cart*) *?V ?E*) ⟶ *ivs_azim_cycle* (*set_of_edge ?v ?V ?E*) *?x ?v ?u* = (*SOME xx*::(*real*, *3*) *cart*. *IN xx* (*set_of_edge ?v ?V ?E*) ∧ *sigma_fan ?x ?V ?E ?v xx* = *?u*)

thm Wrgcvdr_cizmrrh.IVS_AZIM_PROPERTIES:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?u*::(*real*, *3*) *cart*) (*set_of_edge* (*?v*::(*real*, *3*) *cart*) *?V ?E*) ⟶ *IN* (*ivs_azim_cycle* (*set_of_edge ?v ?V ?E*) *?x ?v ?u*) (*set_of_edge ?v ?V ?E*) ∧ *sigma_fan ?x ?V ?E ?v* (*ivs_azim_cycle* (*set_of_edge ?v ?V ?E*) *?x ?v ?u*) = *?u*

thm Wrgcvdr_cizmrrh.IVS_AZIM_EQ_INVERSE_SIGMA_FAN:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) *EMPTY*)) *?E* ⟶ *ivs_azim_cycle* (*EE ?v ?E*) *?x ?v ?u* = *inverse1_sigma_fan ?x ?V ?E ?v ?u*

thm Wrgcvdr_cizmrrh.IVS_AZIM_EQ_INVERSE_SIGMA_FAN2:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) *EMPTY*)) *?E* ⟶ *ivs_azim_cycle* (*set_of_edge ?v ?V ?E*) *?x ?v ?u* = *inverse1_sigma_fan ?x ?V ?E ?v ?u*

thm Wrgcvdr_cizmrrh.EE_OF_HYP_PERMUTES_DARTS:

permutes (*ee_of_hyp* (*?x*::*?'a*::*type*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)) (*darts_of_hyp ?E ?V*)

thm Wrgcvdr_cizmrrh.EE_SUBSET_UNIONS_E:

SUBSET (*EE* (*?v*::*?'a*::*type*) (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) (*UNIONS ?E*)

thm Wrgcvdr_cizmrrh.FAN_IMP_FINITE_EE:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ *FINITE* (*EE* (*?v*::(*real*, *3*) *cart*) *?E*)

thm Wrgcvdr_cizmrrh.W_SUBSET_SINGLETON_IMP_IDE:

∀ (*W*::(*real*, *3*) *cart* ⇒ *bool*) *p*::(*real*, *3*) *cart*. *SUBSET W* (*INSERT p EMPTY*) ⟶ (∀ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *azim_cycle W v w p* = *p*)

thm Wrgcvdr_cizmrrh.IN_SELF_PAIRS_IMP_EE_EMPTY:

IN (?x::?'a::type × ?'a::type) (self_pairs (?E::(?'a::type ⇒ bool) ⇒ bool) (?V::?'a::type ⇒ bool)) ⟶ EE (fst ?x) ?E = EMPTY

thm Wrgcvdr_cizmrrh.IN_DARTS_IFF_NN_OF_HYP_TOO:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ IN (?y::(real, 3) cart × (real, 3) cart) (darts_of_hyp ?E ?V) = IN (nn_of_hyp (?x, ?V, ?E) ?y) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.IN_E_IMP_IMP_IN_DARTS:

IN (INSERT (?a::?'a::type) (INSERT (?b::?'a::type) EMPTY)) (?E::(?'a::type ⇒ bool) ⇒ bool) ⟶ IN (?a, ?b) (darts_of_hyp ?E (?V::?'a::type ⇒ bool))

thm Wrgcvdr_cizmrrh.PAIR_EQ2:

((?a::?'b::type × ?'a::type) = (?b::?'b::type × ?'a::type)) = (fst ?a = fst ?b ∧ snd ?a = snd ?b)

thm Wrgcvdr_cizmrrh.IN_ORD_E_EQ_IN_E:

IN (?x::?'a::type × ?'a::type) (ord_pairs (?E::(?'a::type ⇒ bool) ⇒ bool)) = IN (INSERT (fst ?x) (INSERT (snd ?x) EMPTY)) ?E

thm Wrgcvdr_cizmrrh.FAN_IMP_NN_OF_HYP_PERMUTES_DARTS:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ permutes (nn_of_hyp (?x, ?V, ?E)) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.IVS_AZIM_EMPTY_IDE:

ivs_azim_cycle EMPTY (?x::(real, 3) cart) (?y::(real, 3) cart) (?t::(real, 3) cart) = ?t

thm Wrgcvdr_cizmrrh.FAN_IMP_IN_DARTS_IFF_FF_TOO:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ IN (?y::(real, 3) cart × (real, 3) cart) (darts_of_hyp ?E ?V) = IN (ff_of_hyp (?x, ?V, ?E) ?y) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.IN_E_IFF_IN_ORD_E:

IN (INSERT (?a::?'a::type) (INSERT (?b::?'a::type) EMPTY)) (?E::(?'a::type ⇒ bool) ⇒ bool) = IN (?a, ?b) (ord_pairs ?E)

thm Wrgcvdr_cizmrrh.IN_ORD_E_IFF_SWITCH_TOO:

IN (?x::?'a::type × ?'a::type) (ord_pairs (?E::(?'a::type ⇒ bool) ⇒ bool)) = IN (snd ?x, fst ?x) (ord_pairs ?E)

thm Wrgcvdr_cizmrrh.SIG_AND_INVERSE1_SIG:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (INSERT (?u::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY)) ?E ⟶ sigma_fan ?x ?V ?E ?u ?w = (?v::(real, 3) cart) ⟶ inverse1_sigma_fan ?x ?V ?E ?u ?v = ?w

thm Wrgcvdr_cizmrrh.INVERSE1_SIG_AND_SIG:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (INSERT (?u::(real, 3) cart) (INSERT (?v::(real, 3) cart) EMPTY)) ?E ⟶ inverse1_sigma_fan ?x ?V ?E ?u ?v = (?w::(real, 3) cart) ⟶ sigma_fan ?x ?V ?E ?u ?w = ?v

thm Wrgcvdr_cizmrrh.FAN_IMP_FACE_MAP_PERMUTES_DARTS:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ permutes (ff_of_hyp (?x, ?V, ?E)) (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.nn_of_hyp3:

nn_of_hyp (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) (?y::(real, 3) cart × (real, 3) cart) = (if ¬ IN ?y (darts_of_hyp ?E ?V) ∨ IN ?y (self_pairs ?E ?V) then ?y else (fst ?y, azim_cycle (EE (fst ?y) ?E) ?x (fst ?y) (snd ?y)))

thm Wrgcvdr_cizmrrh.ff_of_hyp3:

ff_of_hyp (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) (?u::(real, 3) cart × (real, 3) cart) = (if ¬ IN ?u (darts_of_hyp ?E ?V) ∨ IN ?u (self_pairs ?E ?V) then ?u else (snd ?u, ivs_azim_cycle (EE (snd ?u) ?E) ?x (snd ?u) (fst ?u)))

thm Wrgcvdr_cizmrrh.FAN_IMP_FIMITE_DARTS:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ⟶ FINITE (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.FAN_IMP_EE_EQ_SET_OF_EDGE:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ⟶ EE (?v::(real, ?'a::type) cart) ?E = set_of_edge ?v ?V ?E

thm Wrgcvdr_cizmrrh.FAN_IMP_IN_SELF_PAIRS_IFF_FF_OF_HYP:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ IN (?y::(real, 3) cart × (real, 3) cart) (self_pairs ?E ?V) = IN (ff_of_hyp (?x, ?V, ?E) ?y) (self_pairs ?E ?V)

thm Wrgcvdr_cizmrrh.FAN_IMP_FINITE_DARTS:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ FINITE (darts_of_hyp ?E ?V)

thm Wrgcvdr_cizmrrh.FIRST_AAUHTVE:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ HYP (?x, ?V, ?E) = (?D::(real, 3) cart × (real, 3) cart ⇒ bool, ?e::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart, ?n::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart, ?f::(real, 3) cart ×

(real, 3) cart ⇒ (real, 3) cart × (real, 3) cart) ⟶ FINITE ?D ∧ permutes ?e ?D ∧ permutes ?n ?D ∧ permutes ?f ?D ∧ ?e ∘ (?n ∘ ?f) = id ∧ ?e ∘ ?e = id

thm Wrgcvdr_cizmrrh.HYP_LEMMA:

∀ x::(real, 3) cart. FAN (x, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ tuple_hypermap (hypermap (HYP (x, ?V, ?E))) = HYP (x, ?V, ?E)

thm Wrgcvdr_cizmrrh.ELMS_OF_HYPERMAP_HYP:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ dart (hypermap (HYP (?x, ?V, ?E))) = darts_of_hyp ?E ?V ∧ edge_map (hypermap (HYP (?x, ?V, ?E))) = ee_of_hyp (?x, ?V, ?E) ∧ node_map (hypermap (HYP (?x, ?V, ?E))) = nn_of_hyp (?x, ?V, ?E) ∧ face_map (hypermap (HYP (?x, ?V, ?E))) = ff_of_hyp (?x, ?V, ?E)

thm Wrgcvdr_cizmrrh.NN_OF_HYP_POWER_IDE:

¬ IN (?y::(real, 3) cart × (real, 3) cart) (darts_of_hyp (?E::((real, 3) cart ⇒ bool) ⇒ bool) (?V::(real, 3) cart ⇒ bool)) ∨ IN ?y (self_pairs ?E ?V) ⟶ (∀ n::nat. POWER (nn_of_hyp (?x::(real, 3) cart, ?V, ?E)) n ?y = ?y)

thm Wrgcvdr_cizmrrh.XX_SS_TT:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?y::(real, 3) cart × (real, 3) cart) (ord_pairs ?E) ⟶ IN (snd (nn_of_hyp (?x, ?V, ?E) ?y)) (EE (fst ?y) ?E)

thm Wrgcvdr_cizmrrh.IN_ORD_PAIRS_IMP_NN_OF_HYP_IN_ORD:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?y::(real, 3) cart × (real, 3) cart) (ord_pairs ?E) ⟶ IN (nn_of_hyp (?x, ?V, ?E) ?y) (ord_pairs ?E)

thm Wrgcvdr_cizmrrh.NN_OF_HYP_POWER_IN_ORD_PAIRS:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?y::(real, 3) cart × (real, 3) cart) (ord_pairs ?E) ⟶ (∀ n::nat. IN (POWER (nn_of_hyp (?x, ?V, ?E)) n ?y) (ord_pairs ?E))

thm Wrgcvdr_cizmrrh.N_HYP_TO_AZIM_CYCLE_LEM:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?u::(real, 3) cart, ?v::(real, 3) cart) (darts_of_hyp ?E ?V) ⟶ (∀ n::nat. POWER (nn_of_hyp (?x, ?V, ?E)) n (?u, ?v) = (?u, POWER (azim_cycle (EE ?u ?E) ?x ?u) n ?v))

thm Wrgcvdr_cizmrrh.iter_sigma_fan_in_set_of_edge:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (v::(real, 3) cart) u::(real, 3) cart. FAN (x, V, E) ∧ IN u (set_of_edge

v V E) ⟶ (∀ *n*::*nat*. *IN* (*ITER n* (*sigma_fan x V E v*) *u*) (*set_of_edge v V E*))

thm Wrgcvdr_cizmrrh.ITER_AZIM_CYCLE_EQ_ITER_SIGMA:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) *EMPTY*)) *?E* ⟶ (∀ *a*::(*real*, *3*) *cart*. *IN a* (*EE ?v ?E*) ⟶ (∀ *n*::*nat*. *ITER n* (*azim_cycle* (*EE ?v ?E*) *?x ?v*) *a* = *ITER n* (*sigma_fan ?x ?V ?E ?v*) *a*))

thm Wrgcvdr_cizmrrh.pmp_to_iter:

∀ *n*::*nat*. *power_map_points* (*?f*::*?'d*::*type* ⇒ *?'c*::*type* ⇒ *?'b*::*type* ⇒ *?'a*::*type* ⇒ *?'e*::*type* ⇒ *?'e*::*type*) (*?x*::*?'d*::*type*) (*?V*::*?'c*::*type*) (*?E*::*?'b*::*type*) (*?v*::*?'a*::*type*) (*?w*::*?'e*::*type*) *n* = *ITER n* (*?f ?x ?V ?E ?v*) *?w*

thm Wrgcvdr_cizmrrh.CYCLIC_SET_IMP_STABLE_SET2:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) *EMPTY*)) *?E* ⟶ (∀ *a*::(*real*, *3*) *cart*. *IN a* (*EE ?v ?E*) ⟶ *EE ?v ?E* = *GSPEC* (λ*GEN%PVAR%1798*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1798* (∃ *n*::*nat*. *y* = *ITER n* (*azim_cycle* (*EE ?v ?E*) *?x ?v*) *a*) *y*))

thm Lvducxu.FAN_DART_DARTS:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ *dart* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) = *darts_of_hyp ?E ?V*

thm Wrgcvdr_cizmrrh.FAN_IMP_BIJ_V_NODE_OF_HYP:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ (*?f*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = (λ*u*::(*real*, *3*) *cart*. *if IN u ?V then node* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) (*u*, *choose_nd_point u ?E ?V*) *else EMPTY*) ⟶ *BIJ ?f ?V* (*GSPEC* (λ*GEN%PVAR%1799*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1799* (*IN y* (*darts_of_hyp ?E ?V*)) (*node* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) *y*)))

thm Wrgcvdr_cizmrrh.LOCAL_FAN_IMP_FF_SUBSET_DARTS:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *SUBSET ?FF* (*darts_of_hyp ?E ?V*)

thm Wrgcvdr_cizmrrh.LOCAL_IMP_FINITE_DARTS:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *FINITE* (*darts_of_hyp ?E ?V*)

thm Wrgcvdr_cizmrrh.LOCAL_FAN_FINITE_FF:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *FINITE ?FF*

thm Wrgcvdr_cizmrrh.LOCAL_FAN_IMP_BIJ_FF_NODES:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ (*?f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *node* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) ⟶ *BIJ ?f ?FF* (*GSPEC* (λ*GEN%PVAR%1800*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1800* (*IN y* (*darts_of_hyp ?E ?V*)) (*node* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) *y*)))

thm Wrgcvdr_cizmrrh.NOT_IN_DARTS_NN_IDE:

¬ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?V*::(*real*, *3*) *cart* ⇒ *bool*)) ⟶ *nn_of_hyp* (*?x*::(*real*, *3*) *cart*, *?V*, *?E*) *?y* = *?y*

thm Wrgcvdr_cizmrrh.ITER_FIXPOINT2:

∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*::*?'a*::*type*. *f x* = *x* ⟶ (∀ *n*::*nat*. *ITER n f x* = *x*)

thm Wrgcvdr_cizmrrh.NOT_IN_DARTS_NN_OF_HYP_POWER_IDE:

¬ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?V*::(*real*, *3*) *cart* ⇒ *bool*)) ⟶ (∀ *n*::*nat*. *POWER* (*nn_of_hyp* (*?x*::(*real*, *3*) *cart*, *?V*, *?E*)) *n ?y* = *?y*)

thm Wrgcvdr_cizmrrh.IN_NODE_IMP_FIRST_EQ:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?a*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*node* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) (*?b*::(*real*, *3*) *cart* × (*real*, *3*) *cart*)) ⟶ *fst ?a* = *fst ?b*

thm Wrgcvdr_cizmrrh.BIJ_BETWEEN_FF_AND_V:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ (*?k*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*) = *fst* ⟶ *BIJ ?k ?FF ?V*

thm Wrgcvdr_cizmrrh.WRGCVDR:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ (*?k*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*) = *fst* ⟶ *BIJ ?k ?FF ?V* ∧ ((∀ *x*::(*real*, *3*) *cart*. *IN x ?V* ⟶ *IN* (*x*, (*?hro*::(*real*, *3*) *cart* ⇒ (*real*, *3*) *cart*) *x*) *?FF*) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *x* = (*fst x*, *?hro* (*fst x*))) ⟶ (∀ *x*::(*real*, *3*) *cart*. *IN x ?V* ⟶ *?V* = *orbit_map ?hro x*))

thm DEF_v_prime:

$v\_prime = (\lambda(\_6513466::?'b::type \Rightarrow bool)\ \_6513467::?'b::type \times ?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1803::?'b::type.\ \exists\, v::?'b::type.\ SETSPEC\ GEN\%PVAR\%1803\ (IN\ v\ \_6513466 \wedge (\exists\, w::?'a::type.\ IN\ (v,\ w)\ \_6513467))\ v))$

thm Wrgcvdr_cizmrrh.v_prime:

$\forall\,(V::?'b::type \Rightarrow bool)\ FF::?'b::type \times ?'a::type \Rightarrow bool.\ v\_prime\ V\ FF = GSPEC\ (\lambda GEN\%PVAR\%1803::?'b::type.\ \exists\, v::?'b::type.\ SETSPEC\ GEN\%PVAR\%1803\ (IN\ v\ V \wedge (\exists\, w::?'a::type.\ IN\ (v,\ w)\ FF))\ v)$

thm DEF_e_prime:

$e\_prime = (\lambda(\_6513478::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_6513479::?'a::type \times ?'a::type \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%1804::?'a::type \Rightarrow bool.\ \exists\,(v::?'a::type)\ w::?'a::type.\ SETSPEC\ GEN\%PVAR\%1804\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ \_6513478 \wedge IN\ (v,\ w)\ \_6513479)\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm Wrgcvdr_cizmrrh.e_prime:

$\forall\,(E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ FF::?'a::type \times ?'a::type \Rightarrow bool.\ e\_prime\ E\ FF = GSPEC\ (\lambda GEN\%PVAR\%1804::?'a::type \Rightarrow bool.\ \exists\,(v::?'a::type)\ w::?'a::type.\ SETSPEC\ GEN\%PVAR\%1804\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ (v,\ w)\ FF)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Wrgcvdr_cizmrrh.IMP_FAN_V_PRIME_E_PRIME:

$FAN\ (?v::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge (\exists\, x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ x\ (dart\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))) \wedge (?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) = face\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))\ x) \longrightarrow FAN\ (?v,\ v\_prime\ ?V\ ?FF,\ e\_prime\ ?E\ ?FF)$

thm Wrgcvdr_cizmrrh.E_PRIME_SUBSET_E:

$SUBSET\ (e\_prime\ (?E::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (?FF::?'a::type \times ?'a::type \Rightarrow bool))\ ?E$

thm Wrgcvdr_cizmrrh.SUBSET_IMP_SO_DO_EE:

$SUBSET\ (?W1.0::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (?W2.0::(?'a::type \Rightarrow bool) \Rightarrow bool) \longrightarrow SUBSET\ (EE\ (?v::?'a::type)\ ?W1.0)\ (EE\ ?v\ ?W2.0)$

thm Wrgcvdr_cizmrrh.FST_SND_X_IN_EE_E_PRIME:

$IN\ (?x::?'a::type \times ?'a::type)\ (?FF::?'a::type \times ?'a::type \Rightarrow bool) \wedge IN\ (INSERT\ (fst\ ?x)\ (INSERT\ (snd\ ?x)\ EMPTY))\ (?E::(?'a::type \Rightarrow bool) \Rightarrow bool) \longrightarrow IN\ (fst\ ?x)\ (EE\ (snd\ ?x)\ (e\_prime\ ?E\ ?FF)) \wedge IN\ (snd\ ?x)\ (EE\ (fst\ ?x)\ (e\_prime\ ?E\ ?FF))$

thm Wrgcvdr_cizmrrh.CYCLIC_MAP_IMP_CIRCLE_ITSELF:

$(\forall\, x::?'a::type.\ IN\ x\ (?W::?'a::type \Rightarrow bool) \longrightarrow ?W = orbit\_map\ (?f::?'a::type \Rightarrow ?'a::type)\ x) \wedge IN\ (?y::?'a::type)\ ?W \longrightarrow (\exists\, x::?'a::type.\ IN\ x\ ?W \wedge ?y = ?f\ x)$

thm DEF_generic:

$generic = (\lambda(\_6517201::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_6517202::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ \forall (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ \_6517202 \wedge IN\ u\ \_6517201 \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (aff\_lt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ u\ EMPTY)) = EMPTY)$

thm Wrgcvdr_cizmrrh.generic:

$\forall (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ V::(real,\ ?'a::type)\ cart \Rightarrow bool.\ generic\ V\ E = (\forall (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ u\ V \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (aff\_lt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ u\ EMPTY)) = EMPTY)$

thm DEF_circular:

$circular = (\lambda(\_6517213::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_6517214::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ \exists (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ \_6517214 \wedge IN\ u\ \_6517213 \wedge HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (aff\_lt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ u\ EMPTY)) \neq EMPTY)$

thm Wrgcvdr_cizmrrh.circular:

$\forall (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ V::(real,\ ?'a::type)\ cart \Rightarrow bool.\ circular\ V\ E = (\exists (v::(real,\ ?'a::type)\ cart)\ (w::(real,\ ?'a::type)\ cart)\ u::(real,\ ?'a::type)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \wedge IN\ u\ V \wedge HOL\_Light\_Import.INTER\ (aff\_gt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (aff\_lt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ u\ EMPTY)) \neq EMPTY)$

thm DEF_lunar:

$lunar = (\lambda(\_6517225::(real,\ ?'a::type)\ cart \times (real,\ ?'a::type)\ cart)\ (\_6517226::(real,\ ?'a::type)\ cart \Rightarrow bool)\ \_6517227::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ \neg\ circular\ \_6517226\ \_6517227 \wedge SUBSET\ (INSERT\ (fst\ \_6517225)\ (INSERT\ (snd\ \_6517225)\ EMPTY))\ \_6517226 \wedge fst\ \_6517225 \neq snd\ \_6517225 \wedge collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (fst\ \_6517225)\ (INSERT\ (snd\ \_6517225)\ EMPTY))))$

thm Wrgcvdr_cizmrrh.lunar:

$\forall (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ lunar\ (v,\ w)\ V\ E = (\neg\ circular\ V\ E \wedge SUBSET\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ V \wedge v \neq w \wedge collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm Wrgcvdr_cizmrrh.DIH2K_IMP_NODE_MAP_X_DIFF_X:

$FINITE\ (dart\ (?H::?'a::type\ hypermap)) \wedge dih2k\ ?H\ (?k::nat) \wedge ?k \neq (0::nat) \longrightarrow (\forall x::?'a::type.\ IN\ x\ (dart\ ?H) \longrightarrow node\_map\ ?H\ x \neq x)$

thm Wrgcvdr_cizmrrh.FAN_IMP_NOT_EMPTY_DARTS:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ⟶ darts_of_hyp ?E ?V ≠ EMPTY

thm Wrgcvdr_cizmrrh.FAN7_SIMPLE:

∀ x::(real, ?'a::type) cart. fan7 (x, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ⟶ (∀ (a::(real, ?'a::type) cart) b::(real, ?'a::type) cart. IN a ?V ∧ IN b ?V ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) (INSERT a EMPTY)) (aff_ge (INSERT x EMPTY) (INSERT b EMPTY)) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER (INSERT a EMPTY) (INSERT b EMPTY)))

thm Wrgcvdr_cizmrrh.FAN_IMP_DIFF:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ⟶ (∀ v::(real, ?'a::type) cart. IN v ?V ∨ IN v (UNIONS ?E) ⟶ v ≠ ?x)

thm Wrgcvdr_cizmrrh.AFF_GE_TO_AFF_GT2_GE1:

(?u::(real, ?'a::type) cart) ≠ (?x::(real, ?'a::type) cart) ∧ (?v::(real, ?'a::type) cart) ≠ ?x ⟶ aff_ge (INSERT ?x EMPTY) (INSERT ?u (INSERT ?v EMPTY)) = HOL_Light_Import.UNION (aff_gt (INSERT ?x EMPTY) (INSERT ?u (INSERT ?v EMPTY))) (HOL_Light_Import.UNION (aff_ge (INSERT ?x EMPTY) (INSERT ?u EMPTY)) (aff_ge (INSERT ?x EMPTY) (INSERT ?v EMPTY)))

thm Wrgcvdr_cizmrrh.AFF_GE_INTER_AFF_LT_IMP_NOT_EQ_COL:

(?v::(real, 3) cart) ≠ vec (0::nat) ∧ (?u::(real, 3) cart) ≠ vec (0::nat) ∧ HOL_Light_Import.INTER (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT ?v EMPTY)) (aff_lt (INSERT (vec (0::nat)) EMPTY) (INSERT ?u EMPTY)) ≠ EMPTY ⟶ ?u ≠ ?v ∧ collinear (INSERT (vec (0::nat)) (INSERT ?u (INSERT ?v EMPTY)))

thm Wrgcvdr_cizmrrh.CIZMRRH:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ generic ?V ?E ∧ ¬ (circular ?V ?E ∨ (∃ (v::(real, 3) cart) w::(real, 3) cart. lunar (v, w) ?V ?E)) ∨ circular ?V ?E ∧ ¬ (generic ?V ?E ∨ (∃ (v::(real, 3) cart) w::(real, 3) cart. lunar (v, w) ?V ?E)) ∨ (∃ (v::(real, 3) cart) w::(real, 3) cart. lunar (v, w) ?V ?E) ∧ ¬ (generic ?V ?E ∨ circular ?V ?E)

thm Lvducxu.IN_FF_IMP_FST_SND_IN_V:

SUBSET (UNIONS (?E::(?'a::type ⇒ bool) ⇒ bool)) (?V::?'a::type ⇒ bool) ∧ SUBSET (?FF::?'a::type × ?'a::type ⇒ bool) (darts_of_hyp ?E ?V) ∧ IN (?x::?'a::type × ?'a::type) ?FF ⟶ IN (fst ?x) ?V ∧ IN (snd ?x) ?V

thm Lvducxu.W_EQ_ITS_ORBIT_IMP_EQ_ITS_IMAGE:

(∀ *x*::*?'a*::*type*. *IN x* (*?W*::*?'a*::*type* ⇒ *bool*) ⟶ *?W* = *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*) ⟶ *?W* = *IMAGE ?f ?W*

thm Lvducxu.FINITE_AND_LOOP_IMP_BIJ_S_IM_S:

∀ *S*::*?'a*::*type* ⇒ *bool*. *FINITE S* ∧ (∀ *x*::*?'a*::*type*. *IN x S* ⟶ *S* = *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*) ⟶ *BIJ ?f S* (*IMAGE ?f S*)

thm Lvducxu.CYCLIC_SET_IMP_SELF_LOPP2:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) *EMPTY*)) *?E* ⟶ (∀ *a*::(*real*, *3*) *cart*. *IN a* (*EE ?v ?E*) ⟶ *EE ?v ?E* = *orbit_map* (*azim_cycle* (*EE ?v ?E*) *?x ?v*) *a*)

thm Lvducxu.FIN_LOOP_IMP_BIJ_ITSELF:

∀ *S*::*?'a*::*type* ⇒ *bool*. *FINITE S* ∧ (∀ *x*::*?'a*::*type*. *IN x S* ⟶ *S* = *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*) ⟶ *BIJ ?f S S*

thm Lvducxu.IN_DARTS_FF_IMP_DARTS_E_PRIME_V_PRIME:

IN (*?x*::*?'a*::*type* × *?'a*::*type*) (*darts_of_hyp* (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*?V*::*?'a*::*type* ⇒ *bool*)) ∧ *IN ?x* (*?FF*::*?'a*::*type* × *?'a*::*type* ⇒ *bool*) ⟶ *IN ?x* (*darts_of_hyp* (*e_prime ?E ?FF*) (*v_prime ?V ?FF*))

thm Lvducxu.PROPERTIES_OF_IVS_AZIM_CYCLE2:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?w*::(*real*, *3*) *cart*) (*EE* (*?v*::(*real*, *3*) *cart*) *?E*) ⟶ *IN* (*ivs_azim_cycle* (*EE ?v ?E*) *?x ?v ?w*) (*EE ?v ?E*) ∧ *azim_cycle* (*EE ?v ?E*) *?x ?v* (*ivs_azim_cycle* (*EE ?v ?E*) *?x ?v ?w*) = *?w*

thm Lvducxu.IN_EE_IFF_IN_E:

IN (*?x*::*?'a*::*type*) (*EE* (*?v*::*?'a*::*type*) (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) = *IN* (*INSERT ?v* (*INSERT ?x EMPTY*)) *?E*

thm Lvducxu.CYCLIC_SET_IMP_SELF_LOPP3:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ (∀ *a*::(*real*, *3*) *cart*. *IN a* (*EE* (*?v*::(*real*, *3*) *cart*) *?E*) ⟶ *EE ?v ?E* = *orbit_map* (*azim_cycle* (*EE ?v ?E*) *?x ?v*) *a*)

thm Lvducxu.BIJ_AZIM_CYCLE_EE:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ *BIJ* (*azim_cycle* (*EE* (*?v*::(*real*, *3*) *cart*) *?E*) *?x ?v*) (*EE ?v ?E*) (*EE ?v ?E*)

thm Lvducxu.IN_FF_FACE_MAP_IDE:

FAN (*?v*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ (∃ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*dart* (*hypermap* (*HYP* (*?v*, *?V*, *?E*)))) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap*

$(HYP\ (?v,\ ?V,\ ?E)))\ x) \longrightarrow (\forall x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ x\ ?FF \longrightarrow face\_map\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))\ x = face\_map\ (hypermap\ (HYP\ (?v,\ v\_prime\ ?V\ ?FF,\ e\_prime\ ?E\ ?FF)))\ x)$

thm Lvducxu.LOCALIZE_PRESERVE_FACE:

$FAN\ (?v::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge IN\ (?x::(real,\ 3)\ cart \times (real,\ 3)\ cart)\ (dart\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))) \wedge (?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) = face\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))\ ?x \longrightarrow ?FF = face\ (hypermap\ (HYP\ (?v,\ v\_prime\ ?V\ ?FF,\ e\_prime\ ?E\ ?FF)))\ ?x$

thm Lvducxu.FAN_IN_SEFL_PAIRS_IMP_PROGORIOUS_DEGENERATE:

$FAN\ (?x::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge IN\ (?v::(real,\ 3)\ cart \times (real,\ 3)\ cart)\ (self\_pairs\ ?E\ ?V) \longrightarrow edge\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))\ ?v = ?v \wedge face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))\ ?v = ?v \wedge node\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))\ ?v = ?v$

thm Lvducxu.FAN_FACE_IMP_IVS_F_IN_F:

$FAN\ (?x::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge (?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) = face\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))\ (?xx::(real,\ 3)\ cart \times (real,\ 3)\ cart) \wedge IN\ (?y::(real,\ 3)\ cart \times (real,\ 3)\ cart)\ ?FF \longrightarrow IN\ (HOL\_Light\_Import.inverse\ (face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E))))\ ?y)\ ?FF \wedge face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E))) \circ HOL\_Light\_Import.inverse\ (face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))) = id \wedge HOL\_Light\_Import.inverse\ (face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E)))) \circ face\_map\ (hypermap\ (HYP\ (?x,\ ?V,\ ?E))) = id$

thm Lvducxu.INVERSE_FACE_EQ_INV_FACE_LOCALLIZED:

$FAN\ (?v::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge IN\ (?x::(real,\ 3)\ cart \times (real,\ 3)\ cart)\ (dart\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))) \wedge (?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) = face\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))\ ?x \longrightarrow (\forall x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ x\ ?FF \longrightarrow HOL\_Light\_Import.inverse\ (face\_map\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E))))\ x = HOL\_Light\_Import.inverse\ (face\_map\ (hypermap\ (HYP\ (?v,\ v\_prime\ ?V\ ?FF,\ e\_prime\ ?E\ ?FF))))\ x)$

thm Lvducxu.ED_MA_O_NO_MA_EQ_INV_FA:

$\forall H::?'a::type\ hypermap.\ edge\_map\ H \circ node\_map\ H = HOL\_Light\_Import.inverse\ (face\_map\ H)$

thm Lvducxu.IN_FF_IMP_AZIM_CYCLE_EQ:

$FAN\ (?v::(real,\ 3)\ cart,\ ?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \wedge IN\ (?x::(real,\ 3)\ cart \times (real,\ 3)\ cart)\ (dart\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))) \wedge (?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) = face\ (hypermap\ (HYP\ (?v,\ ?V,\ ?E)))\ ?x \longrightarrow (\forall x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ x\ ?FF$

⟶ (*azim_cycle* (*EE* (*fst x*) *?E*) *?v* (*fst x*) (*snd x*), *fst x*) = (*azim_cycle* (*EE* (*fst x*) (*e_prime ?E ?FF*)) *?v* (*fst x*) (*snd x*), *fst x*))

thm Lvducxu.IN_DARTS_IMP_NN_OF_HYP_TOO:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *FAN* (*x*, *V*, *E*) ⟶ (∀ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y* (*darts_of_hyp E V*) ⟶ *IN* (*nn_of_hyp* (*x*, *V*, *E*) *y*) (*darts_of_hyp E V*))

thm Lvducxu.CARD_E_GT1_EQ_CARD_E_PRIME:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*dart* (*hypermap* (*HYP* (*?x*, *?V*, *?E*)))) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) *?v* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ ((*1*::*nat*) < *CARD* (*EE* (*fst ?y*) *?E*)) = ((*1*::*nat*) < *CARD* (*EE* (*fst ?y*) (*e_prime ?E ?FF*)))

thm Lvducxu.AZIM_IN_FAN_EQ_IZIM_E_PRIME:

FAN (*vec* (*0*::*nat*), *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*dart* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*)))) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) *?v* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ *azim_in_fan ?y ?E* = *azim_in_fan ?y* (*e_prime ?E ?FF*)

thm Lvducxu.AZIM_CY_FST_Y_IN_FF:

IN (*?xx*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*dart* (*hypermap* (*HYP* (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*)))) ∧ *FAN* (*?x*, *?V*, *?E*) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) *?xx* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ *IN* (*azim_cycle* (*EE* (*fst ?y*) *?E*) *?x* (*fst ?y*) (*snd ?y*), *fst ?y*) *?FF* ∧ *HOL_Light_Import.inverse* (*face_map* (*hypermap* (*HYP* (*?x*, *?V*, *?E*)))) *?y* = (*azim_cycle* (*EE* (*fst ?y*) *?E*) *?x* (*fst ?y*) (*snd ?y*), *fst ?y*)

thm Lvducxu.EE_FST_Y_EQ_SET_SET_SNDY:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*dart* (*hypermap* (*HYP* (*?x*, *?V*, *?E*)))) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?x*, *?V*, *?E*))) *?v* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ (*EE* (*fst ?y*) *?E* = *INSERT* (*snd ?y*) *EMPTY*) = (*EE* (*fst ?y*) (*e_prime ?E ?FF*) = *INSERT* (*snd ?y*) *EMPTY*)

thm Lvducxu.WEDGE_IN_FAN_EQ_WITH_E_PRIME:

FAN (*vec* (*0*::*nat*), *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*dart* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*)))) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) *?v* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3) cart) ?FF* ⟶ *wedge_in_fan_ge ?y ?E* = *wedge_in_fan_ge ?y* (*e_prime ?E ?FF*) ∧ *wedge_in_fan_gt ?y ?E* = *wedge_in_fan_gt ?y* (*e_prime ?E ?FF*)

thm Lvducxu.FACE_NODE_EDGE_ORBIT_INVERSE:

face (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *orbit_map* (*HOL_Light_Import.inverse* (*face_map ?H*)) *?x* ∧ *node ?H ?x* = *orbit_map* (*HOL_Light_Import.inverse* (*node_map ?H*)) *?x* ∧ *edge ?H ?x* = *orbit_map* (*HOL_Light_Import.inverse* (*edge_map ?H*)) *?x*

thm Lvducxu.DARTS_E_PRIME_EQ_FF_UNION_SWITCH:

(∀ *x*::*?'a*::*type* × *?'a*::*type*. *IN x* (*?FF*::*?'a*::*type* × *?'a*::*type* ⇒ *bool*) ⟶ *IN* (*INSERT* (*fst x*) (*INSERT* (*snd x*) *EMPTY*)) (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) ⟶ *darts_of_hyp* (*e_prime ?E ?FF*) (*v_prime* (*?V*::*?'a*::*type* ⇒ *bool*) *?FF*) = *HOL_Light_Import.UNION ?FF* (*GSPEC* (λ*GEN%PVAR%1816*::*?'a*::*type* × *?'a*::*type*. ∃ (*v*::*?'a*::*type*) *w*::*?'a*::*type*. *SETSPEC GEN%PVAR%1816* (*IN* (*w*, *v*) *?FF*) (*v*, *w*)))

thm Lvducxu.CARD_FF_GT1_FF_SUBSET:

FAN (*?v*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?v*, *?V*, *?E*))) (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) ∧ (*1*::*nat*) < *CARD ?FF* ⟶ (∀ *y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN y ?FF* ⟶ *IN* (*INSERT* (*fst y*) (*INSERT* (*snd y*) *EMPTY*)) *?E*)

thm Lvducxu.DARTS_E_PRIME_GT1_SWITCH:

FAN (*?v*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?v*, *?V*, *?E*))) (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) ∧ (*1*::*nat*) < *CARD ?FF* ⟶ *darts_of_hyp* (*e_prime ?E ?FF*) (*v_prime ?V ?FF*) = *HOL_Light_Import.UNION ?FF* (*GSPEC* (λ*GEN%PVAR%1817*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1817* (*IN* (*w*, *v*) *?FF*) (*v*, *w*)))

thm Lvducxu.FAN_FACE_GT1_IMAGE_EE_OF_HYP:

IN (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?V*::(*real*, *3*) *cart* ⇒ *bool*)) ∧ *FAN* (*?v*::(*real*, *3*) *cart*, *?V*, *?E*) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?v*, *?V*, *?E*))) *?x* ∧ (*1*::*nat*) < *CARD ?FF* ⟶ *darts_of_hyp* (*e_prime ?E ?FF*) (*v_prime ?V ?FF*) = *HOL_Light_Import.UNION ?FF* (*IMAGE* (*ee_of_hyp* (*?v*, *?V*, *?E*)) *?FF*)

thm Lvducxu.IN_ORD_PAIRS_E_PRIME:

IN (*?x*::*?'a*::*type* × *?'a*::*type*) (*ord_pairs* (*e_prime* (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*?FF*::*?'a*::*type* × *?'a*::*type* ⇒ *bool*))) ⟶ *IN ?x* (*ord_pairs ?E*) ∧ *IN ?x* (*darts_of_hyp ?E* (*?V*::*?'a*::*type* ⇒ *bool*))

thm Lvducxu.CARD_GT1_EE_OF_HYP_E_PRIME_EQ:

FAN (?v::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) (dart (hypermap (HYP (?v, ?V, ?E)))) ∧ (?FF::(real, 3) cart × (real, 3) cart ⇒ bool) = face (hypermap (HYP (?v, ?V, ?E))) ?x ∧ (1::nat) < CARD ?FF ⟶ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ ee_of_hyp (?v, ?V, ?E) x = ee_of_hyp (?v, v_prime ?V ?FF, e_prime ?E ?FF) x)

thm Lvducxu.FF_IMAGE_EE_OF_HYP_EQ_E_PRIME:

FAN (?v::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) (dart (hypermap (HYP (?v, ?V, ?E)))) ∧ (?FF::(real, 3) cart × (real, 3) cart ⇒ bool) = face (hypermap (HYP (?v, ?V, ?E))) ?x ∧ (1::nat) < CARD ?FF ⟶ IMAGE (ee_of_hyp (?v, ?V, ?E)) ?FF = IMAGE (ee_of_hyp (?v, v_prime ?V ?FF, e_prime ?E ?FF)) ?FF

thm Lvducxu.FIRST_AAUHTVE2:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ FINITE (darts_of_hyp ?E ?V) ∧ permutes (ee_of_hyp (?x, ?V, ?E)) (darts_of_hyp ?E ?V) ∧ permutes (nn_of_hyp (?x, ?V, ?E)) (darts_of_hyp ?E ?V) ∧ permutes (ff_of_hyp (?x, ?V, ?E)) (darts_of_hyp ?E ?V) ∧ ee_of_hyp (?x, ?V, ?E) ∘ (nn_of_hyp (?x, ?V, ?E) ∘ ff_of_hyp (?x, ?V, ?E)) = id ∧ ee_of_hyp (?x, ?V, ?E) ∘ ee_of_hyp (?x, ?V, ?E) = id

thm Lvducxu.NOT_IN_DART_IMP_IDE:

¬ IN (?x::?'a::type) (dart (?H::?'a::type hypermap)) ⟶ edge_map ?H ?x = ?x ∧ node_map ?H ?x = ?x ∧ face_map ?H ?x = ?x

thm Lvducxu.SIMPLE_IMP_DISTINCT_FF_NODE_IMAGE:

simple_hypermap (?H::?'a::type hypermap) ∧ (?FF::?'a::type ⇒ bool) = face ?H (?x::?'a::type) ∧ (∀x::?'a::type. IN x ?FF ⟶ node_map ?H x ≠ x) ⟶ HOL_Light_Import.INTER ?FF (IMAGE (node_map ?H) ?FF) = EMPTY

thm Lvducxu.FX_IN_S_IMP_F_POWER_TOO:

(∀x::?'a::type. IN x (?S::?'a::type ⇒ bool) ⟶ IN ((?f::?'a::type ⇒ ?'a::type) x) ?S) ∧ IN (?x::?'a::type) ?S ⟶ (∀n::nat. IN (POWER ?f n ?x) ?S)

thm Lvducxu.FX_IN_S_IMP_ORBIT_SUBSET:

(∀x::?'a::type. IN x (?S::?'a::type ⇒ bool) ⟶ IN ((?f::?'a::type ⇒ ?'a::type) x) ?S) ∧ IN (?x::?'a::type) ?S ⟶ SUBSET (orbit_map ?f ?x) ?S

thm Lvducxu.FAN_DARTS_OF_IN_D:

FAN (?v::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) (darts_of_hyp ?E ?V) ∧ (?FF::(real, 3) cart × (real, 3) cart ⇒ bool) = face (hypermap (HYP (?v, ?V, ?E))) ?x ∧ IN (?x'::(real, 3) cart × (real, 3) cart) ?FF ⟶ IN ?x' (darts_of_hyp ?E ?V)

thm Lvducxu.FACE_SUBSET_DARTS:

FAN (*?v*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp ?E ?V*) ⟶ *SUBSET* (*face* (*hypermap* (*HYP* (*?v*, *?V*, *?E*))) *?x*) (*darts_of_hyp ?E ?V*)

thm Lvducxu.FF_DISJOINT_ITS_IMAGE_CARD_EQ:

FAN (*?v*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp ?E ?V*) ∧ (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*?v*, *?V*, *?E*))) *?x* ∧ *simple_hypermap* (*hypermap* (*HYP* (*?v*, *v_prime ?V ?FF*, *e_prime ?E ?FF*))) ∧ (*2*::*nat*) < *CARD ?FF* ⟶ *HOL_Light_Import.INTER ?FF* (*IMAGE* (*nn_of_hyp* (*?v*, *v_prime ?V ?FF*, *e_prime ?E ?FF*)) *?FF*) = *EMPTY* ∧ *CARD ?FF* = *CARD* (*IMAGE* (*nn_of_hyp* (*?v*, *v_prime ?V ?FF*, *e_prime ?E ?FF*)) *?FF*)

thm Lvducxu.HYP_MAPS_INJ:

(∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. (*edge_map* (*?H*::*?'a*::*type hypermap*) *x* = *edge_map ?H y*) = (*x* = *y*)) ∧ (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. (*node_map ?H x* = *node_map ?H y*) = (*x* = *y*)) ∧ (∀ (*x*::*?'a*::*type*) *y*::*?'a*::*type*. (*face_map ?H x* = *face_map ?H y*) = (*x* = *y*))

thm Lvducxu.SIMPLE_FACE_DISJOINT_NODE_MAP_2:

IN (*?x*::*?'a*::*type*) (*dart* (*?H*::*?'a*::*type hypermap*)) ∧ *SUBSET* (*face ?H ?x*) (*dart ?H*) ∧ *simple_hypermap ?H* ∧ *HOL_Light_Import.INTER* (*face ?H ?x*) (*IMAGE* (*node_map ?H*) (*face ?H ?x*)) = *EMPTY* ∧ *dart ?H* = *HOL_Light_Import.UNION* (*face ?H ?x*) (*IMAGE* (*node_map ?H*) (*face ?H ?x*)) ⟶ (∀ *x*::*?'a*::*type*. *IN x* (*dart ?H*) ⟶ *node_map ?H x* ≠ *x* ∧ *node_map ?H* (*node_map ?H x*) = *x*)

thm Lvducxu.ITER12:

∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) *x*::*?'a*::*type*. *ITER* (*1*::*nat*) *f x* = *f x* ∧ *ITER* (*2*::*nat*) *f x* = *f* (*f x*)

thm Lvducxu.EDGE_MAP_RESO_INVERSE_conjunct0:

(*edge_map* (*?H*::*?'a*::*type hypermap*) ∘ *edge_map ?H* = *id*) = (*HOL_Light_Import.inverse* (*edge_map ?H*) = *edge_map ?H*)

thm Lvducxu.EDGE_MAP_RESO_INVERSE_conjunct1:

(*node_map* (*?H*::*?'a*::*type hypermap*) ∘ *node_map ?H* = *id*) = (*HOL_Light_Import.inverse* (*node_map ?H*) = *node_map ?H*)

thm Lvducxu.EDGE_MAP_RESO_INVERSE:

(*edge_map* (*?H*::*?'b*::*type hypermap*) ∘ *edge_map ?H* = *id*) = (*HOL_Light_Import.inverse* (*edge_map ?H*) = *edge_map ?H*) ∧ (*node_map* (*?Ha*::*?'a*::*type hypermap*) ∘ *node_map ?Ha* = *id*) = (*HOL_Light_Import.inverse* (*node_map ?Ha*) = *node_map ?Ha*)

thm Lvducxu.HAS_ORD2_INTERPRET:

has_orders (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*2*::*nat*) = (*?f* ∘ *?f* = *id* ∧ *?f* ≠ *id*)

thm Lvducxu.HYP_MAPS_INVS:

edge_map (*?H*::*?'a*::*type hypermap*) (*HOL_Light_Import.inverse* (*edge_map ?H*) (*?x*::*?'a*::*type*)) = *?x* ∧ *HOL_Light_Import.inverse* (*edge_map ?H*) (*edge_map ?H ?x*) = *?x* ∧ *face_map ?H* (*HOL_Light_Import.inverse* (*face_map ?H*) *?x*) = *?x* ∧ *HOL_Light_Import.inverse* (*face_map ?H*) (*face_map ?H ?x*) = *?x* ∧ *node_map ?H* (*HOL_Light_Import.inverse* (*node_map ?H*) *?x*) = *?x* ∧ *HOL_Light_Import.inverse* (*node_map ?H*) (*node_map ?H ?x*) = *?x*

thm Lvducxu.ITER_CYCLIC_ORBIT:

(*0*::*nat*) < (*?i*::*nat*) ∧ *ITER ?i* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?x*::*?'a*::*type*) = *?x* ⟶ *orbit_map ?f ?x* = *GSPEC* (λ*GEN%PVAR%1818*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1818* (*n* < *?i*) (*ITER n ?f ?x*))

thm Lvducxu.FACE_CYCLE_CARD:

∀ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *H*::*?'a*::*type hypermap*. *IN y* (*face H x*) ⟶ *POWER* (*face_map H*) (*CARD* (*face H x*)) *y* = *y*

thm Lvducxu.FACE_NODE_EDGE_ORBIT_INVERSE_conjunct2:

edge (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *orbit_map* (*HOL_Light_Import.inverse* (*edge_map ?H*)) *?x*

thm Lvducxu.FACE_NODE_EDGE_ORBIT_INVERSE_conjunct1:

node (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *orbit_map* (*HOL_Light_Import.inverse* (*node_map ?H*)) *?x*

thm Lvducxu.FACE_NODE_EDGE_ORBIT_INVERSE_conjunct0:

face (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *orbit_map* (*HOL_Light_Import.inverse* (*face_map ?H*)) *?x*

thm Lvducxu.INVERSE_FACE_CYCLE:

∀ (*x*::*?'a*::*type*) *H*::*?'a*::*type hypermap*. *POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*CARD* (*face H x*)) *x* = *x*

thm Lvducxu.INVERSE_FACE_CYCLE_ALL:

∀ (*x*::*?'a*::*type*) (*y*::*?'a*::*type*) *H*::*?'a*::*type hypermap*. *IN y* (*face H x*) ⟶ *POWER* (*HOL_Light_Import.inverse* (*face_map H*)) (*CARD* (*face H x*)) *y* = *y*

thm Lvducxu.DIH2K_IMP_SIMPLE_HYPERMAP2:

dih2k (*?H*::*?'a*::*type hypermap*) (*?k*::*nat*) ∧ *?k* ≠ (*0*::*nat*) ⟶ *simple_hypermap ?H*

thm Lvducxu.HYP_MAPS_INVS_conjunct5:

HOL_Light_Import.inverse (*node_map* (*?H*::*?'a*::*type hypermap*)) (*node_map* *?H* (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.HYP_MAPS_INVS_conjunct4:

node_map (*?H*::*?'a*::*type hypermap*) (*HOL_Light_Import.inverse* (*node_map* *?H*) (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.HYP_MAPS_INVS_conjunct3:

HOL_Light_Import.inverse (*face_map* (*?H*::*?'a*::*type hypermap*)) (*face_map ?H* (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.HYP_MAPS_INVS_conjunct2:

face_map (*?H*::*?'a*::*type hypermap*) (*HOL_Light_Import.inverse* (*face_map ?H*) (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.HYP_MAPS_INVS_conjunct1:

HOL_Light_Import.inverse (*edge_map* (*?H*::*?'a*::*type hypermap*)) (*edge_map* *?H* (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.HYP_MAPS_INVS_conjunct0:

edge_map (*?H*::*?'a*::*type hypermap*) (*HOL_Light_Import.inverse* (*edge_map ?H*) (*?x*::*?'a*::*type*)) = *?x*

thm Lvducxu.DIH2K_FACE_SIMPLIZED:

(*has_orders* (*edge_map* (*?H*::*?'a*::*type hypermap*)) (*2*::*nat*) ∧ *IN* (*?x*::*?'a*::*type*) (*dart ?H*) ∧ *simple_hypermap ?H* ∧ *HOL_Light_Import.INTER* (*face ?H ?x*) (*IMAGE* (*node_map ?H*) (*face ?H ?x*)) = *EMPTY* ∧ *dart ?H* = *HOL_Light_Import.UNION* (*face ?H ?x*) (*IMAGE* (*node_map ?H*) (*face ?H ?x*))) = (*dih2k ?H* (*CARD* (*face ?H ?x*)) ∧ *IN ?x* (*dart ?H*))

thm Lvducxu.EE_OF_HYP_IDE_FST_SND_EQ:

IN (*?z*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*?V*::(*real*, *3*) *cart* ⇒ *bool*)) ⟶ (*ee_of_hyp* (*?x*::*?'a*::*type*, *?V*, *?E*) *?z* = *?z*) = (*fst ?z* = *snd ?z*)

thm Lvducxu.ENF_IMAGE_ITSELF_conjunct2:

node (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *IMAGE* (*node_map ?H*) (*node* *?H ?x*)

thm Lvducxu.ENF_IMAGE_ITSELF_conjunct1:

face (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *IMAGE* (*face_map ?H*) (*face* *?H ?x*)

thm Lvducxu.ENF_IMAGE_ITSELF_conjunct0:

edge (*?H*::*?'a*::*type hypermap*) (*?x*::*?'a*::*type*) = *IMAGE* (*edge_map ?H*) (*edge* *?H ?x*)

thm Lvducxu.ENF_IMAGE_ITSELF:

edge (?H::?'c::type hypermap) (?x::?'c::type) = IMAGE (edge_map ?H) (edge ?H ?x) ∧ face (?Ha::?'b::type hypermap) (?xa::?'b::type) = IMAGE (face_map ?Ha) (face ?Ha ?xa) ∧ node (?Hb::?'a::type hypermap) (?xb::?'a::type) = IMAGE (node_map ?Hb) (node ?Hb ?xb)

thm Lvducxu.IDE_ON_S_IMP_SAME_IMAGE:

(∀ x::?'b::type. IN x (?S::?'b::type ⇒ bool) ⟶ (?f::?'b::type ⇒ ?'a::type) x = (?g::?'b::type ⇒ ?'a::type) x) ⟶ IMAGE ?f ?S = IMAGE ?g ?S

thm Lvducxu.DIH_K_HYP_E_PRIME:

FAN (?v::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) (darts_of_hyp ?E ?V) ∧ (?FF::(real, 3) cart × (real, 3) cart ⇒ bool) = face (hypermap (HYP (?v, ?V, ?E))) ?x ∧ simple_hypermap (hypermap (HYP (?v, v_prime ?V ?FF, e_prime ?E ?FF))) ∧ (2::nat) < CARD ?FF ⟶ dih2k (hypermap (HYP (?v, v_prime ?V ?FF, e_prime ?E ?FF))) (CARD ?FF)

thm Lvducxu.LVDUCXU:

FAN (vec (0::nat), ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) (darts_of_hyp ?E ?V) ∧ (?FF::(real, 3) cart × (real, 3) cart ⇒ bool) = face (hypermap (HYP (vec (0::nat), ?V, ?E))) ?x ⟶ ?FF = face (hypermap (HYP (vec (0::nat), v_prime ?V ?FF, e_prime ?E ?FF))) ?x ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ azim_in_fan x ?E = azim_in_fan x (e_prime ?E ?FF) ∧ wedge_in_fan_ge x ?E = wedge_in_fan_ge x (e_prime ?E ?FF) ∧ wedge_in_fan_gt x ?E = wedge_in_fan_gt x (e_prime ?E ?FF)) ∧ ((2::nat) < CARD ?FF ∧ simple_hypermap (hypermap (HYP (vec (0::nat), v_prime ?V ?FF, e_prime ?E ?FF))) ⟶ local_fan (v_prime ?V ?FF, e_prime ?E ?FF, ?FF))

thm Ldurdpn.SUBSET_P_HULL:

SUBSET (?S::?'a::type ⇒ bool) (hull (?P::(?'a::type ⇒ bool) ⇒ bool) ?S)

thm Ldurdpn.IN_HULL_INSERT:

∀ x::?'a::type. IN x (hull (?P::(?'a::type ⇒ bool) ⇒ bool) (INSERT x (?S::?'a::type ⇒ bool)))

thm Ldurdpn.VECTOR_SCALE_CHANGE:

(?a::real) ≠ (0::real) ⟶ (% ?a (?x::(real, ?'a::type) cart) = (?y::(real, ?'a::type) cart)) = (?x = % ((1::real) / ?a) ?y)

thm Ldurdpn.AFF_CONV0_IN_AFF_LT:

¬ collinear (INSERT (vec (0::nat)) (INSERT (?u::(real, ?'a::type) cart) (INSERT (?v::(real, ?'a::type) cart) EMPTY))) ⟶ HOL_Light_Import.INTER (aff (INSERT (vec (0::nat)) (INSERT ?u EMPTY))) (conv0 (INSERT ?v (INSERT (?w::(real,

?'a::type) cart) EMPTY))) ≠ EMPTY ⟶ IN ?w (aff_lt (INSERT (vec (0::nat)) (INSERT ?u EMPTY)) (INSERT ?v EMPTY))

thm Ldurdpn.LDURDPN:

¬ collinear (INSERT (vec (0::nat)) (INSERT (?u::(real, 3) cart) (INSERT (?v::(real, 3) cart) EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?u (INSERT (?w::(real, 3) cart) EMPTY))) ⟶ (azim (vec (0::nat)) ?u ?v ?w = pi) = ((∃ A::(real, 3) cart ⇒ bool. plane A ∧ SUBSET (INSERT (vec (0::nat)) (INSERT ?u (INSERT ?v (INSERT ?w EMPTY)))) A) ∧ HOL_Light_Import.INTER (aff (INSERT (vec (0::nat)) (INSERT ?u EMPTY))) (conv0 (INSERT ?v (INSERT ?w EMPTY))) ≠ EMPTY)

thm DEF_rho_node1:

rho_node1 = (λ(_6528142::(real, 3) cart × (real, 3) cart ⇒ bool) _6528143::(real, 3) cart. SOME w::(real, 3) cart. IN (_6528143, w) _6528142)

thm Local_lemmas.rho_node1:

∀ (v::(real, 3) cart) FF::(real, 3) cart × (real, 3) cart ⇒ bool. rho_node1 FF v = (SOME w::(real, 3) cart. IN (v, w) FF)

thm DEF_ivs_rho_node1:

ivs_rho_node1 = (λ(_6528154::(real, 3) cart × (real, 3) cart ⇒ bool) _6528155::(real, 3) cart. SOME a::(real, 3) cart. IN (a, _6528155) _6528154)

thm Local_lemmas.ivs_rho_node1:

∀ (v::(real, 3) cart) FF::(real, 3) cart × (real, 3) cart ⇒ bool. ivs_rho_node1 FF v = (SOME a::(real, 3) cart. IN (a, v) FF)

thm Local_lemmas.LOCAL_FAN_IMP_IN_V:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?x::(real, 3) cart, ?y::(real, 3) cart) ?FF ⟶ IN ?x ?V ∧ IN ?y ?V

thm Local_lemmas.LOCAL_FAN_RHO_NODE_PROS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀ x::(real, 3) cart. IN x ?V ⟶ IN (x, rho_node1 ?FF x) ?FF) ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ x = (fst x, rho_node1 ?FF (fst x)))

thm Local_lemmas.ORTHONORMAL_CYCLIC:

(?x::(real, ?'a::type) cart) ≠ (?y::(real, ?'a::type) cart) ∧ FINITE (?U::(real, ?'a::type) cart ⇒ bool) ∧ (∀ (u1::(real, ?'a::type) cart) u2::(real, ?'a::type) cart. IN u1 ?U ∧ IN u2 ?U ⟶ dot (vector_sub u1 u2) (vector_sub ?x ?y) = (0::real)) ∧ HOL_Light_Import.INTER ?U (hull affine (INSERT ?x (INSERT ?y EMPTY))) = EMPTY ⟶ cyclic_set ?U ?x ?y

thm Local_lemmas.FAN_SINGLETON_V_DARTS:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *?V* = *INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) *EMPTY* ⟶ *darts_of_hyp ?E ?V* = *INSERT* (*?v*, *?v*) *EMPTY*

thm Local_lemmas.FAN_IN_DARTS_FST_EQ_SND_SELF_PAIRS:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *IN* (*?y*::(*real*, *?'a*::*type*) *cart* × (*real*, *?'a*::*type*) *cart*) (*darts_of_hyp ?E ?V*) ⟶ (*fst ?y* = *snd ?y*) = *IN ?y* (*self_pairs ?E ?V*)

thm Local_lemmas.FAN_FST_EQ_SND_SUPPER_EQ:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *fst* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = *snd ?y* ⟶ *ee_of_hyp* (*?x*, *?V*, *?E*) *?y* = *?y* ∧ *nn_of_hyp* (*?x*, *?V*, *?E*) *?y* = *?y* ∧ *ff_of_hyp* (*?x*, *?V*, *?E*) *?y* = *?y*

thm Local_lemmas.COLLINEAR_CROSS_0:

collinear (*INSERT* (*?x*::(*real*, *3*) *cart*) (*INSERT* (*?y*::(*real*, *3*) *cart*) (*INSERT* (*?z*::(*real*, *3*) *cart*) *EMPTY*))) = (*cross* (*vector_sub ?y ?x*) (*vector_sub ?z ?x*) = *vec* (*0*::*nat*))

thm Local_lemmas.DET_CROSS:

det (*vector* [*?x*::(*real*, *3*) *cart*, *?y*::(*real*, *3*) *cart*, *?z*::(*real*, *3*) *cart*]) = *dot* (*cross ?x ?y*) *?z*

thm Local_lemmas.COPLANAR_IFF_CROSS_DOT:

coplanar (*INSERT* (*?x*::(*real*, *3*) *cart*) (*INSERT* (*?y*::(*real*, *3*) *cart*) (*INSERT* (*?z*::(*real*, *3*) *cart*) (*INSERT* (*?t*::(*real*, *3*) *cart*) *EMPTY*)))) = (*dot* (*cross* (*vector_sub ?y ?x*) (*vector_sub ?z ?x*)) (*vector_sub ?t ?x*) = (*0*::*real*))

thm Local_lemmas.CROSS_DOT_COPLANAR:

(*dot* (*cross* (*?x*::(*real*, *3*) *cart*) (*?y*::(*real*, *3*) *cart*)) (*?z*::(*real*, *3*) *cart*) = (*0*::*real*)) = *coplanar* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?x* (*INSERT ?y* (*INSERT ?z* *EMPTY*))))

thm Local_lemmas.SUBSET_NOT_COLLINEAR_AFFINE_HULL_EQ:

SUBSET (*INSERT* (*?a*::(*real*, *3*) *cart*) (*INSERT* (*?b*::(*real*, *3*) *cart*) (*INSERT* (*?c*::(*real*, *3*) *cart*) *EMPTY*))) (*hull affine* (*INSERT* (*?x*::(*real*, *3*) *cart*) (*INSERT* (*?y*::(*real*, *3*) *cart*) (*INSERT* (*?z*::(*real*, *3*) *cart*) *EMPTY*)))) ∧ ¬ *collinear* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c* *EMPTY*))) ⟶ *hull affine* (*INSERT ?x* (*INSERT ?y* (*INSERT ?z* *EMPTY*))) = *hull affine* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c* *EMPTY*)))

thm Local_lemmas.THREE_NOT_COLL_DETER_PLANE:

plane (*?P*::(*real*, *3*) *cart* ⇒ *bool*) ∧ *SUBSET* (*INSERT* (*?a*::(*real*, *3*) *cart*) (*INSERT* (*?b*::(*real*, *3*) *cart*) (*INSERT* (*?c*::(*real*, *3*) *cart*) *EMPTY*))) *?P* ∧

¬ *collinear* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*))) ⟶ *hull affine* (*INSERT ?a* (*INSERT ?b* (*INSERT ?c EMPTY*))) = *?P*

thm Local_lemmas.LOCAL_FAN_NOT_V_SING:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ ¬ (∃ *v*::(*real*, *3*) *cart*. *?V* = *INSERT v EMPTY*)

thm Local_lemmas.LOCAL_FAN_NOT_SING_FF:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ ¬ (∃ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *?FF* = *INSERT x EMPTY*)

thm Local_lemmas.LOCAL_FAN_IN_FF_DISTINCT:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ *fst ?x* ≠ *snd ?x*

thm Local_lemmas.LOCAL_FAN_IN_FF_IN_ORD_PAIRS:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ *IN ?x* (*ord_pairs ?E*)

thm Local_lemmas.LOCAL_FAN_IN_FF_NOT_COLLINEAR:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*fst ?x*) (*INSERT* (*snd ?x*) *EMPTY*)))

thm Local_lemmas.LOCAL_FAN_CHARACTER_OF_RHO_NODE:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ (∀ *v*::(*real*, *3*) *cart*. *IN v ?V* ⟶ *rho_node1 ?FF v* ≠ *v* ∧ *IN* (*v*, *rho_node1 ?FF v*) (*ord_pairs ?E*) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT* (*rho_node1 ?FF v*) *EMPTY*))))

thm Local_lemmas.graph2:

graph (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) = (∀ *e*::*?'a*::*type* ⇒ *bool*. *IN e ?E* ⟶ *FINITE e* ∧ *CARD e* = (*2*::*nat*))

thm Local_lemmas.GRAPH_WITH_SET2:

graph (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) ⟶ (∀ (*a*::*?'a*::*type*) *b*::*?'a*::*type*. *IN* (*INSERT a* (*INSERT b EMPTY*)) *?E* ⟶ *a* ≠ *b*)

thm Local_lemmas.FAN_V_TWO_ELMS_IN_E_DARTS2:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *?V* = *INSERT* (*?v1.0*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v2.0*::(*real*, *?'a*::*type*) *cart*) *EMPTY*) ∧ *IN* (*INSERT ?v1.0* (*INSERT ?v2.0 EMPTY*)) *?E* ⟶ *darts_of_hyp ?E ?V* = *INSERT* (*?v1.0*, *?v2.0*) (*INSERT* (*?v2.0*, *?v1.0*) *EMPTY*)

thm Local_lemmas.FAN_IN_E_DIFF:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ (∀ (*x*::(*real*, *?'a*::*type*) *cart*) *y*::(*real*, *?'a*::*type*) *cart*. *IN* (*INSERT x* (*INSERT y EMPTY*)) *?E* ⟶ (*x*, *y*) ≠ (*y*, *x*))

thm Local_lemmas.LOCAL_FAN_NOT_TWO_V_IN_E:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ ¬ (∃ (*v1*::(*real*, *3*) *cart*) *v2*::(*real*, *3*) *cart*. *?V* = *INSERT v1* (*INSERT v2 EMPTY*) ∧ *IN* (*INSERT v1* (*INSERT v2 EMPTY*)) *?E*)

thm Local_lemmas.LOCAL_FAN_ORBIT_MAP_V:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ (∀ *v*::(*real*, *3*) *cart*. *IN v ?V* ⟶ *orbit_map* (*rho_node1 ?FF*) *v* = *?V*)

thm Local_lemmas.LOCAL_FAN_RHO_NODE_PROS2:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *x* = (*fst x*, *rho_node1 ?FF* (*fst x*))) ∧ (∀ *x*::(*real*, *3*) *cart*. *IN x ?V* ⟶ *IN* (*x*, *rho_node1 ?FF x*) *?FF*)

thm Local_lemmas.FINTE_OF_N_FIRST_ELMS2:

FINITE (*GSPEC* (λ*GEN%PVAR%1823*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1823* (*n* < (*?i*::*nat*)) (*ITER n* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?x*::*?'a*::*type*))))

thm Local_lemmas.PLANE_AFFINE_HUL_INTER_P:

plane (*?P*::(*real*, *3*) *cart* ⇒ *bool*) ∧ *SUBSET* (*INSERT* (*?x*::(*real*, *3*) *cart*) (*INSERT* (*?y*::(*real*, *3*) *cart*) (*INSERT* (*?z*::(*real*, *3*) *cart*) *EMPTY*))) *?P* ⟶ *HOL_Light_Import.INTER* (*hull affine* (*INSERT ?x* (*INSERT* (*vector_add ?x* (*cross* (*vector_sub ?y ?x*) (*vector_sub ?z ?x*))) *EMPTY*))) *?P* = *INSERT ?x EMPTY*

thm Local_lemmas.FAN_IMP_V_DIFF:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ⟶ (∀ *v*::(*real*, *?'a*::*type*) *cart*. *IN v ?V* ⟶ *v* ≠ *?x*)

thm Local_lemmas.LOCAL_FAN_IMP_CYCLIC_SET:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *GSPEC* (λ*GEN%PVAR%1824*::(*real*, *3*) *cart*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1824*

$(n \leq (?l::nat))$ $(ITER\ n\ (rho\_node1\ ?FF)\ ?v)) = (?U::(real,\ 3)\ cart \Rightarrow bool)$ $\wedge\ plane\ (?P::(real,\ 3)\ cart \Rightarrow bool) \wedge IN\ (vec\ (0::nat))\ ?P \wedge SUBSET\ ?U\ ?P$ $\wedge\ cross\ ?v\ (rho\_node1\ ?FF\ ?v) = (?e::(real,\ 3)\ cart) \longrightarrow cyclic\_set\ ?U\ (vec\ (0::nat))\ ?e$

thm Local_lemmas.LOCAL_FAN_ITER_RHO_NODE_IN_V:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \wedge IN\ (?v::(real,\ 3)\ cart)\ ?V \longrightarrow (\forall\, i::nat.\ IN\ (ITER\ i\ (rho\_node1\ ?FF)\ ?v)\ ?V)$

thm Local_lemmas.ORD2_ORBIT_MAP:

$(?f::?'a::type \Rightarrow ?'a::type)\ (?f\ (?x::?'a::type)) = ?x \longrightarrow orbit\_map\ ?f\ ?x = INSERT\ ?x\ (INSERT\ (?f\ ?x)\ EMPTY)$

thm Local_lemmas.LOCAL_FAN_IMP_NOT_SEMI_IDE:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \longrightarrow (\forall\, v::(real,\ 3)\ cart.\ IN\ v\ ?V \longrightarrow rho\_node1\ ?FF\ (rho\_node1\ ?FF\ v) \neq v)$

thm ENDS_IN_HALFLINE_conjunct1:

$\forall\,(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ y\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ EMPTY))$

thm ENDS_IN_HALFLINE_conjunct0:

$\forall\,(x::(real,\ ?'a::type)\ cart)\ y::(real,\ ?'a::type)\ cart.\ IN\ x\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ y\ EMPTY))$

thm Local_lemmas.RHO_NODE_SET_IN_A_PLANE_IMP_POS_DIRECT:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \wedge IN\ (?v::(real,\ 3)\ cart)\ ?V \wedge GSPEC\ (\lambda GEN\%PVAR\%1825::(real,\ 3)\ cart.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%1825\ (n \leq (?l::nat))\ (ITER\ n\ (rho\_node1\ ?FF)\ ?v)) = (?U::(real,\ 3)\ cart \Rightarrow bool) \wedge plane\ (?P::(real,\ 3)\ cart \Rightarrow bool) \wedge IN\ (vec\ (0::nat))\ ?P \wedge SUBSET\ ?U\ ?P \wedge cross\ ?v\ (rho\_node1\ ?FF\ ?v) = (?e::(real,\ 3)\ cart) \longrightarrow (\forall\, i{<}?l.\ (0::real) < dot\ (cross\ (ITER\ i\ (rho\_node1\ ?FF)\ ?v)\ (ITER\ (i + (1::nat))\ (rho\_node1\ ?FF)\ ?v))\ ?e)$

thm Local_lemmas.AZIM_RANGE:

$\forall\,(v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (w1::(real,\ 3)\ cart)\ w2::(real,\ 3)\ cart.\ (0::real) \leq azim\ v\ w\ w1\ w2 \wedge azim\ v\ w\ w1\ w2 < real\_of\_nat\ (2::nat) * pi$

thm Local_lemmas.PI_TO_TWO_PI_NEG_SIN:

$\forall\, x::real.\ pi < x \wedge x < real\_of\_nat\ (2::nat) * pi \longrightarrow sin\ x < (0::real)$

thm Local_lemmas.MIXED_PROD_POS_IMP_RANGE_AZIM:

$\neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (?u::(real,\ 3)\ cart)\ (INSERT\ (?v::(real,\ 3)\ cart)\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT$

?u (INSERT (?w::(real, 3) cart) EMPTY))) ∧ (0::real) < dot (cross ?u ?v) ?w ⟶ (0::real) < azim (vec (0::nat)) ?u ?v ?w ∧ azim (vec (0::nat)) ?u ?v ?w < pi

thm Local_lemmas.COLLINEAR_ONCE_VEC_0:

(?x::(real, ?'a::type) cart) ≠ vec (0::nat) ⟶ (∀ y::(real, ?'a::type) cart. collinear (INSERT (vec (0::nat)) (INSERT ?x (INSERT y EMPTY))) = (∃ t::real. y = % t ?x))

thm Local_lemmas.AFF_GE11:

∀ (x::(real, ?'a::type) cart) v::(real, ?'a::type) cart. x ≠ v ⟶ aff_ge (INSERT x EMPTY) (INSERT v EMPTY) = GSPEC (λGEN%PVAR%1826::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1826 (∃ (t1::real) t2::real. (0::real) ≤ t2 ∧ t1 + t2 = (1::real) ∧ y = vector_add (% t1 x) (% t2 v)) y)

thm Local_lemmas.X_IN_AFF_GE11:

∀ x::(real, ?'a::type) cart. (?c::(real, ?'a::type) cart) ≠ x ⟶ IN x (aff_ge (INSERT ?c EMPTY) (INSERT x EMPTY))

thm Local_lemmas.FAN_IN_AFF_GE_IMP_EQ:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ IN (?v::(real, ?'a::type) cart) ?V ∧ IN (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) EMPTY)) ?E ∧ IN ?v (aff_ge (INSERT ?x EMPTY) (INSERT ?a (INSERT ?b EMPTY))) ⟶ ?v = ?a ∨ ?v = ?b

thm Local_lemmas.AFF_GE22:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) (x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. DISJOINT (INSERT a (INSERT b EMPTY)) (INSERT x (INSERT y EMPTY)) ⟶ aff_ge (INSERT a (INSERT b EMPTY)) (INSERT x (INSERT y EMPTY)) = GSPEC (λGEN%PVAR%1827::(real, ?'a::type) cart. ∃ z::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1827 (∃ (aa::real) (bb::real) (xx::real) yy::real. (0::real) ≤ xx ∧ (0::real) ≤ yy ∧ aa + (bb + (xx + yy)) = (1::real) ∧ z = vector_add (% aa a) (vector_add (% bb b) (vector_add (% xx x) (% yy y)))) z)

thm Local_lemmas.RHO_NODE_IS_SUCCESEOR_AZIM:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1828::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1828 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ (∀ (x::(real, 3) cart) n::nat. x = ITER n (rho_node1 ?FF) ?v ∧ n < ?l ⟶ (∀ y::(real, 3) *cart. IN y ?U ∧ y ≠ x ∧ y ≠ rho_node1 ?FF x ⟶ azim (vec (0::nat)) ?e x (rho_node1 ?FF x) < azim (vec (0::nat)) ?e x y))*

thm Local_lemmas.FIRST_AZIM_CYCLE_EQ_RHO_NODE:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1829::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1829 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ (∀ (x::(real, 3) cart) n::nat. x = ITER n (rho_node1 ?FF) ?v ∧ n < ?l ⟶ azim_cycle ?U (vec (0::nat)) ?e x = rho_node1 ?FF x)

thm Local_lemmas.SEQUENCE_OF_RHO_NODE_IS_SUC:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1830::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1830 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ (∀ (x::(real, 3) cart) n::nat. x = ITER n (rho_node1 ?FF) ?v ∧ n < ?l ⟶ (∀ y::(real, 3) cart. IN y ?U ∧ y ≠ x ∧ y ≠ rho_node1 ?FF x ⟶ azim (vec (0::nat)) ?e y x < azim (vec (0::nat)) ?e y (rho_node1 ?FF x)))

thm Local_lemmas.V_AZIM_SMALLEST_ELMS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1831::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1831 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ ITER ?l (rho_node1 ?FF) ?v = (?ls::(real, 3) cart) ∧ (∀ i<?l. ?ls ≠ ITER i (rho_node1 ?FF) ?v) ⟶ (∀ y::(real, 3) cart. IN y ?U ∧ y ≠ ?v ∧ y ≠ ?ls ⟶ azim (vec (0::nat)) ?e ?ls ?v < azim (vec (0::nat)) ?e ?ls y)

thm Local_lemmas.AZIM_LAST_POINT_IN_RHO_SET:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1834::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1834 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ ITER ?l (rho_node1 ?FF) ?v = (?ls::(real, 3) cart) ∧ (∀ i<?l. ?ls ≠ ITER i (rho_node1 ?FF) ?v) ⟶ azim_cycle ?U (vec (0::nat)) ?e ?ls = ?v

thm Local_lemmas.LOOP_MAP_IMP_DIFF_FIRST_ELMS:

(∀ v::?'a::type. IN v (?V::?'a::type ⇒ bool) ⟶ orbit_map (?f::?'a::type ⇒ ?'a::type) v = ?V) ∧ IN (?v::?'a::type) ?V ∧ (?k::nat) < CARD ?V ∧ (?l::nat) < ?k ⟶ ITER ?k ?f ?v ≠ ITER ?l ?f ?v

thm Local_lemmas.CARD_IMAGE_INJ2:

INJ (?f::?'b::type ⇒ ?'a::type) (?A::?'b::type ⇒ bool) (?B::?'a::type ⇒ bool) ∧ FINITE ?A ⟶ CARD (IMAGE ?f ?A) = CARD ?A

thm Local_lemmas.BIJ_IMP_CARD_EQ:

BIJ (?f::?'b::type ⇒ ?'a::type) (?A::?'b::type ⇒ bool) (?B::?'a::type ⇒ bool) ∧ FINITE ?A ⟶ CARD ?A = CARD ?B

thm Local_lemmas.LOFA_IMP_DIS_ELMS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ (?l::nat) < CARD ?FF ⟶ (∀ i<?l. ITER ?l (rho_node1 ?FF) ?v ≠ ITER i (rho_node1 ?FF) ?v)

thm Local_lemmas.AZIM_LAST_POINT_IN_RHO_SET2:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1835::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1835 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ∧ cross ?v (rho_node1 ?FF ?v) = (?e::(real, 3) cart) ⟶ ITER ?l (rho_node1 ?FF) ?v = (?ls::(real, 3) cart) ∧ ?l < CARD ?FF ⟶ azim_cycle ?U (vec (0::nat)) ?e ?ls = ?v

thm Local_lemmas.KOMWBWC:

∀ (E::((real, 3) cart ⇒ bool) ⇒ bool) (V::(real, 3) cart ⇒ bool) (P::(real, 3) cart ⇒ bool) (l::nat) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (U::(real, 3) cart ⇒ bool) (e::(real, 3) cart) (ls::(real, 3) cart) v::(real, 3) cart. local_fan (V, E, FF) ∧ IN v V ∧ GSPEC (λGEN%PVAR%1836::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1836 (n ≤ l) (ITER n (rho_node1 FF) v)) = U ∧ plane P ∧ IN (vec (0::nat)) P ∧ SUBSET U P ∧ cross v (rho_node1 FF v) = e ⟶ cyclic_set U (vec (0::nat)) e ∧ (∀ i<l. (0::real) < dot (cross (ITER i (rho_node1 FF) v) (ITER (i + (1::nat)) (rho_node1 FF) v)) e) ∧ (∀ (x::(real, 3) cart) n::nat. x = ITER n (rho_node1 FF) v ∧ n < l ⟶ azim_cycle U (vec (0::nat)) e x = rho_node1 FF x) ∧ (ITER l (rho_node1 FF) v = ls ∧ l < CARD FF ⟶ azim_cycle U (vec (0::nat)) e ls = v)

thm DEF_interior_angle1:

interior_angle1 = (λ(_6536709::(real, 3) cart) (_6536710::(real, 3) cart × (real, 3) cart ⇒ bool) _6536711::(real, 3) cart. azim _6536709 _6536711 (rho_node1 _6536710 _6536711) (SOME a::(real, 3) cart. IN (a, _6536711) _6536710))

thm Local_lemmas.interior_angle1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *interior_angle1 x FF v* = *azim x v* (*rho_node1 FF v*) (*SOME a*::(*real*, *3*) *cart*. *IN* (*a*, *v*) *FF*)

thm Local_lemmas.WEDGE_GE_AZIM_LE:

∀ (*x*::(*real*, *3*) *cart*) (*v0*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *IN x* (*wedge_ge v0 v1 w1 w2*) = (*azim v0 v1 w1 x* ≤ *azim v0 v1 w1 w2*)

thm Local_lemmas.IN_WEDGE_IMP_AZIM_LE:

∀ (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *IN y* (*wedge_ge* (*?v0.0*::(*real*, *3*) *cart*) (*?v1.0*::(*real*, *3*) *cart*) (*?w1.0*::(*real*, *3*) *cart*) (*?w2.0*::(*real*, *3*) *cart*)) ∧ *azim ?v0.0 ?v1.0 ?w1.0 x* ≤ *azim ?v0.0 ?v1.0 ?w1.0 y* ∧ ¬ *collinear* (*INSERT ?v0.0* (*INSERT ?v1.0* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT ?v0.0* (*INSERT ?v1.0* (*INSERT y EMPTY*))) ∧ ¬ *collinear* (*INSERT ?v0.0* (*INSERT ?v1.0* (*INSERT ?w1.0 EMPTY*))) ⟶ *azim ?v0.0 ?v1.0 x y* ≤ *azim ?v0.0 ?v1.0 ?w1.0 ?w2.0*

thm Local_lemmas.LOFA_IMAGE_RHO_NODE_IDE:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *IMAGE* (*rho_node1 ?FF*) *?V* = *?V*

thm Local_lemmas.EXISTS_INVERSE_OF_V:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ (∃ *vv*::(*real*, *3*) *cart*. *IN vv ?V* ∧ *rho_node1 ?FF vv* = *?v*)

thm Local_lemmas.LOFA_IN_V_SO_DO_RHO_NODE_V:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ *IN* (*rho_node1 ?FF ?v*) *?V*

thm Local_lemmas.HYP_MAPS_INVERSABLE:

∀ *H*::*?'a*::*type hypermap*. *HOL_Light_Import.inverse* (*face_map H*) ∘ *face_map H* = *id* ∧ *HOL_Light_Import.inverse* (*node_map H*) ∘ *node_map H* = *id* ∧ *HOL_Light_Import.inverse* (*edge_map H*) ∘ *edge_map H* = *id* ∧ *face_map H* ∘ *HOL_Light_Import.inverse* (*face_map H*) = *id* ∧ *node_map H* ∘ *HOL_Light_Import.inverse* (*node_map H*) = *id* ∧ *edge_map H* ∘ *HOL_Light_Import.inverse* (*edge_map H*) = *id*

thm Local_lemmas.LOFA_DARTS_FF_UNION_SWITCH_FF:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *darts_of_hyp ?E ?V* = *HOL_Light_Import.UNION ?FF* (*GSPEC* (λ*GEN%PVAR%1837*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ (*v*::(*real*, *3) cart) w::(real, 3) cart. SETSPEC GEN%PVAR%1837 (IN (w, v) ?FF) (v, w)))*

thm Local_lemmas.SELF_CYCLIC_IMP_FINITE:

(∀ x::?'a::type. IN x (?V::?'a::type ⇒ bool) ⟶ ?V = orbit_map (?f::?'a::type ⇒ ?'a::type) x) ⟶ FINITE ?V

thm Local_lemmas.SELF_CYCLIC_IMP_BIJ:

(∀ x::?'a::type. IN x (?S::?'a::type ⇒ bool) ⟶ ?S = orbit_map (?f::?'a::type ⇒ ?'a::type) x) ⟶ BIJ ?f ?S ?S

thm Local_lemmas.LOFA_IN_E_IMP_IN_FF:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (INSERT (?a::(real, 3) cart) (INSERT (?b::(real, 3) cart) EMPTY)) ?E ⟶ IN (?a, ?b) ?FF ∨ IN (?b, ?a) ?FF

thm Local_lemmas.LOFA_IMP_EE_TWO_ELMS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?vv::(real, 3) cart) ?V ∧ rho_node1 ?FF ?vv = (?v::(real, 3) cart) ⟶ EE ?v ?E = INSERT (rho_node1 ?FF ?v) (INSERT ?vv EMPTY)

thm Local_lemmas.LOFA_CARD_EE_V_2:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?vv::(real, 3) cart) ?V ∧ rho_node1 ?FF ?vv = (?v::(real, 3) cart) ⟶ CARD (EE ?v ?E) = (2::nat)

thm Local_lemmas.LOFA_CARD_EE_V_1:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ CARD (EE ?v ?E) = (2::nat)

thm Local_lemmas.RHO_NODE_INVERSE_POINT:

IN (?w::?'b::type, ?v::?'a::type) (?FF::?'b::type × ?'a::type ⇒ bool) ∧ (∀ ww::?'b::type. IN (ww, ?v) ?FF ⟶ ww = ?w) ⟶ (SOME a::?'b::type. IN (a, ?v) ?FF) = ?w

thm Local_lemmas.AZIM_CYCLE_TWO_POINT_SET:

azim_cycle (INSERT (?a::(real, 3) cart) (INSERT (?b::(real, 3) cart) EMPTY)) (?v::(real, 3) cart) (?w::(real, 3) cart) ?a = ?b

thm Local_lemmas.LOFA_IMP_BIJ_VV:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ BIJ (rho_node1 ?FF) ?V ?V

thm Local_lemmas.MOST_EXPAND_IN_WEDGE_GE:

¬ *collinear* (*INSERT* (*?v0.0*::(*real*, *3*) *cart*) (*INSERT* (*?v1.0*::(*real*, *3*) *cart*) (*INSERT* (*?w1.0*::(*real*, *3*) *cart*) *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *?v0.0* (*INSERT* *?v1.0* (*INSERT* (*?w2.0*::(*real*, *3*) *cart*) *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *?v0.0* (*INSERT* *?v1.0* (*INSERT* (*?x*::(*real*, *3*) *cart*) *EMPTY*))) ∧ ¬ *collinear* (*INSERT* *?v0.0* (*INSERT* *?v1.0* (*INSERT* (*?y*::(*real*, *3*) *cart*) *EMPTY*))) ∧ *IN* *?y* (*wedge_ge* *?v0.0* *?v1.0* *?w1.0* *?w2.0*) ∧ *azim* *?v0.0* *?v1.0* *?w1.0* *?x* ≤ *azim* *?v0.0* *?v1.0* *?w1.0* *?y* ∧ *azim* *?v0.0* *?v1.0* *?x* *?y* = *azim* *?v0.0* *?v1.0* *?w1.0* *?w2.0* ⟶ *azim* *?v0.0* *?v1.0* *?w1.0* *?x* = (*0*::*real*) ∧ *azim* *?v0.0* *?v1.0* *?y* *?w2.0* = (*0*::*real*)

thm Local_lemmas.OZQVSFF:

∀ (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *convex_local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *SUBSET* (*INSERT* *u* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* *w* *EMPTY*))) *?V* ∧ *plane* (*?P*::(*real*, *3*) *cart* ⇒ *bool*) ∧ *SUBSET* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *u* (*INSERT* *?v* (*INSERT* *w* *EMPTY*)))) *?P* ∧ *HOL_Light_Import.INTER* (*INSERT* *u* (*INSERT* *w* *EMPTY*)) (*aff* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *?v* *EMPTY*))) = *EMPTY* ∧ *HOL_Light_Import.INTER* (*aff* (*INSERT* *?v* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*))) (*conv0* (*INSERT* *w* (*INSERT* *u* *EMPTY*))) ≠ *EMPTY* ⟶ *interior_angle1* (*vec* (*0*::*nat*)) *?FF* *?v* = *pi* ∧ *IN* (*rho_node1* *?FF* *?v*) *?P* ∧ *IN* (*ivs_rho_node1* *?FF* *?v*) *?P*

thm Local_lemmas.REAL_LT_DIV_NEG:

(*?a*::*real*) < (*0*::*real*) ∧ (*?b*::*real*) < (*0*::*real*) ⟶ (*0*::*real*) < *?a* / *?b*

thm Local_lemmas.IN_CONV0:

(*0*::*real*) < (*?a*::*real*) ∧ (*0*::*real*) < (*?b*::*real*) ⟶ *IN* (% ((*1*::*real*) / (*?a* + *?b*)) (*vector_add* (% *?a* (*?x*::(*real*, *?'a*::*type*) *cart*)) (% *?b* (*?y*::(*real*, *?'a*::*type*) *cart*)))) (*conv0* (*INSERT* *?x* (*INSERT* *?y* *EMPTY*)))

thm Local_lemmas.INTERSECTION_LEMMA:

(*?z*::(*real*, *?'a*::*type*) *cart*) ≠ (*?x*::(*real*, *?'a*::*type*) *cart*) ∧ *DISJOINT* (*INSERT* *?x* *EMPTY*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) ∧ *?x* ≠ (*?u*::(*real*, *?'a*::*type*) *cart*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* *?x* *EMPTY*) (*INSERT* *?v* (*INSERT* *?w* *EMPTY*))) (*aff_lt* (*INSERT* *?x* *EMPTY*) (*INSERT* *?u* *EMPTY*)) ≠ *EMPTY* ∧ *IN* *?z* (*hull* *affine* (*INSERT* *?x* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*)))) ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*INSERT* *a* (*INSERT* *b* *EMPTY*)) (*INSERT* *?u* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*))) ∧ *IN* *t* (*HOL_Light_Import.INTER* (*aff* (*INSERT* *?x* (*INSERT* *?z* *EMPTY*))) (*conv0* (*INSERT* *a* (*INSERT* *b* *EMPTY*)))))

thm Local_lemmas.CVX_LO_IMP_LO:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *local_fan* (*?V*, *?E*, *?FF*)

thm Local_lemmas.S_SUBSET_IMP_AFF_S_TOO:

SUBSET (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*aff* (*?SS*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*)) ⟶ *SUBSET* (*aff ?S*) (*aff ?SS*)

thm Local_lemmas.AFF2_DET_BY_TWO_POINTS:

SUBSET (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) (*aff* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ∧ *?x* ≠ *?y* ⟶ *aff* (*INSERT ?a* (*INSERT ?b EMPTY*)) = *aff* (*INSERT ?x* (*INSERT ?y EMPTY*))

thm Local_lemmas.IN_CONV0_EQ_EQ:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *IN x* (*conv0* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ (*x* = *?a*) = (*?a* = *?b*)

thm Local_lemmas.CONV02_SUBSET_AFF2:

SUBSET (*conv0* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) (*aff* (*INSERT ?a* (*INSERT ?b EMPTY*)))

thm Local_lemmas.IN_CONV0_IMP_AFF_EQ:

IN (*?a*::(*real*, *?'a*::*type*) *cart*) (*conv0* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *aff* (*INSERT ?x* (*INSERT ?y EMPTY*)) = *aff* (*INSERT ?x* (*INSERT ?a EMPTY*))

thm Local_lemmas.IN_CONV0_AFF_SUBSET:

∀ *t*::(*real*, *?'a*::*type*) *cart*. *IN t* (*HOL_Light_Import.INTER* (*aff* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) (*conv0* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)))) ∧ *IN ?x* (*aff* (*INSERT ?a* (*INSERT ?b EMPTY*))) ⟶ *SUBSET* (*aff* (*INSERT ?x* (*INSERT ?y EMPTY*))) (*aff* (*INSERT ?a* (*INSERT ?b EMPTY*)))

thm Local_lemmas.CONDS_FOR_INTER_AFF_CONV0:

(*0*::*real*) < (*?t2.0*::*real*) ∧ (*0*::*real*) < (*?t3.0*::*real*) ∧ (*?t1.0*::*real*) + (*?t2.0* + *?t3.0*) = (*?ss*::*real*) ∧ (*?t*::(*real*, *?'a*::*type*) *cart*) = *vector_add* (% *?t1.0* (*?x*::(*real*, *?'a*::*type*) *cart*)) (*vector_add* (% *?t2.0* (*?v*::(*real*, *?'a*::*type*) *cart*)) (% *?t3.0* (*?w*::(*real*, *?'a*::*type*) *cart*))) ∧ (*?t1'*::*real*) + (*?t2'*::*real*) = *?ss* ∧ *?t* = *vector_add* (% *?t1' ?x*) (% *?t2'* (*?u*::(*real*, *?'a*::*type*) *cart*)) ⟶ (∃ *tt*::(*real*, *?'a*::*type*) *cart*. *IN tt* (*aff* (*INSERT ?x* (*INSERT ?u EMPTY*))) ∧ *IN tt* (*conv0* (*INSERT ?v* (*INSERT ?w EMPTY*))))

thm Local_lemmas.SUBSET_AFF2_IMP_COLL:

SUBSET (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*aff* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *collinear ?S*

thm Local_lemmas.COLLINEAR_SUBSET_AFF2:

IN (*?a*::(*real*, *?'a*::*type*) *cart*) (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ∧ *IN* (*?b*::(*real*, *?'a*::*type*) *cart*) *?S* ∧ *?a* ≠ *?b* ∧ *collinear ?S* ⟶ *SUBSET ?S* (*aff* (*INSERT ?a* (*INSERT ?b EMPTY*)))

thm Local_lemmas.AFF_CONV0_COLL4:

IN (?t::(real, ?'a::type) cart) (HOL_Light_Import.INTER (aff (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) EMPTY))) (conv0 (INSERT (?x::(real, ?'a::type) cart) (INSERT (?y::(real, ?'a::type) cart) EMPTY)))) ∧ IN ?x (aff (INSERT ?a (INSERT ?b EMPTY))) ⟶ collinear (INSERT ?a (INSERT ?b (INSERT ?x (INSERT ?y EMPTY))))

thm Local_lemmas.CONDS_IN_HAFL_LINE:

(0::real) ≤ (?t::real) ∧ vector_sub (?a::(real, ?'a::type) cart) (?x::(real, ?'a::type) cart) = % ?t (vector_sub (?b::(real, ?'a::type) cart) ?x) ⟶ IN ?a (aff_ge (INSERT ?x EMPTY) (INSERT ?b EMPTY))

thm Local_lemmas.PRESERABLE_AFF_GE_SUBSET:

IN (?a::(real, ?'a::type) cart) (aff_ge (INSERT (?x::(real, ?'a::type) cart) EMPTY) (INSERT (?b::(real, ?'a::type) cart) EMPTY)) ⟶ SUBSET (aff_ge (INSERT ?x EMPTY) (INSERT ?a EMPTY)) (aff_ge (INSERT ?x EMPTY) (INSERT ?b EMPTY))

thm Local_lemmas.AFF_GE_EQ:

(0::real) < (?t::real) ∧ vector_sub (?a::(real, ?'a::type) cart) (?x::(real, ?'a::type) cart) = % ?t (vector_sub (?b::(real, ?'a::type) cart) ?x) ⟶ aff_ge (INSERT ?x EMPTY) (INSERT ?a EMPTY) = aff_ge (INSERT ?x EMPTY) (INSERT ?b EMPTY)

thm Local_lemmas.FAN_INTERSECTION_PRO_EXPRESS:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ IN (?a::(real, ?'a::type) cart) ?V ∧ IN (?b::(real, ?'a::type) cart) ?V ∧ ?a ≠ ?b ⟶ ¬ (∃ t>0::real. vector_sub ?a ?x = % t (vector_sub ?b ?x))

thm Local_lemmas.FAN_INTERSECTION_PRO_EXPRESS2:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ IN (?a::(real, ?'a::type) cart) ?V ∧ IN (?b::(real, ?'a::type) cart) ?V ∧ ?a ≠ ?b ∧ (0::real) < (?t::real) ⟶ vector_sub ?a ?x ≠ % ?t (vector_sub ?b ?x)

thm Local_lemmas.FAN_AFF2_INTER_CONV0_IMP_NO_IN_AF:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ SUBSET (INSERT (?u::(real, ?'a::type) cart) (INSERT (?v::(real, ?'a::type) cart) (INSERT (?w::(real, ?'a::type) cart) EMPTY))) ?V ∧ IN (?t::(real, ?'a::type) cart) (HOL_Light_Import.INTER (aff (INSERT ?x (INSERT ?v EMPTY))) (conv0 (INSERT ?u (INSERT ?w EMPTY)))) ∧ CARD (INSERT ?u (INSERT ?v (INSERT ?w EMPTY))) = (3::nat) ⟶ ¬ IN ?u (aff (INSERT ?x (INSERT ?v EMPTY)))

thm Local_lemmas.AFF2_ITR_CONV0_IMP_SAME_ENDS:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *IN* (*?t*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import*.*INTER* (*aff* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) (*conv0* (*INSERT* *a* (*INSERT* *b* *EMPTY*)))) ⟶ *IN* *a* (*aff* (*INSERT* *?x* (*INSERT* *?y* *EMPTY*))) = *IN* *b* (*aff* (*INSERT* *?x* (*INSERT* *?y* *EMPTY*)))

thm Local_lemmas.AFF_XX_CASES:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *aff_lt* (*INSERT* *x* *EMPTY*) (*INSERT* *x* *EMPTY*) = *EMPTY* ∧ *aff_le* (*INSERT* *x* *EMPTY*) (*INSERT* *x* *EMPTY*) = *EMPTY* ∧ *aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *x* *EMPTY*) = *INSERT* *x* *EMPTY*

thm Local_lemmas.NOT_X_IN_AFF_X_A:

¬ *IN* (*?x*::(*real*, *?'a*::*type*) *cart*) (*aff_lt* (*INSERT* *?x* *EMPTY*) (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))

thm Local_lemmas.INTER_EQ_EM_EXPAND:

(*HOL_Light_Import*.*INTER* (*?A*::*?'a*::*type* ⇒ *bool*) (*?B*::*?'a*::*type* ⇒ *bool*) = *EMPTY*) = (¬ (∃ *x*::*?'a*::*type*. *IN* *x* *?A* ∧ *IN* *x* *?B*))

thm Local_lemmas.NOT_INTER_EQ_EM_IMP_AFF_SUBSET:

IN (*?x'*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) *EMPTY*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) (*aff_lt* (*INSERT* *?x* *EMPTY*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *SUBSET* (*aff* (*INSERT* *?x* (*INSERT* *?u* *EMPTY*))) (*hull* *affine* (*INSERT* *?x* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*))))

thm Local_lemmas.IN_AFF_HULL_3:

IN (*?x'*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) *EMPTY*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) (*aff_lt* (*INSERT* *?x* *EMPTY*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *IN* *?u* (*hull* *affine* (*INSERT* *?x* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*))))

thm Local_lemmas.lemma_in_orbit_iter:

∀ (*f*::*?'a*::*type* ⇒ *?'a*::*type*) (*n*::*nat*) *x*::*?'a*::*type*. *IN* (*ITER* *n* *f* *x*) (*orbit_map* *f* *x*)

thm Local_lemmas.LOCAL_FAN_ORBIT_MAP_VITER:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ (∀ (*v*::(*real*, *3*) *cart*) *n*::*nat*. *IN* *v* *?V* ⟶ *IN* (*ITER* *n* (*rho_node1* *?FF*) *v*) *?V*)

thm Local_lemmas.INTER_AFF_GT_LT_IMP_INTER_AFF_CONV0:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. (*DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *w* *EMPTY*)) ∧ *u* ≠ *x*) ∧ *IN* (*?t*::(*real*, *?'a*::*type*) *cart*) (*HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*aff_lt* (*INSERT x EMPTY*) (*INSERT u EMPTY*))) ⟶ (∃ *tt*::(*real*, *?'a*::*type*) *cart*. *IN tt* (*HOL_Light_Import.INTER* (*aff* (*INSERT x* (*INSERT u EMPTY*))) (*conv0* (*INSERT v* (*INSERT w EMPTY*)))))

thm Local_lemmas.AFF_GT_AFF_LT_INTERPRET:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) ∧ *DISJOINT* (*INSERT x EMPTY*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) *EMPTY*) ⟶ (∃ (*a*::*real*) (*b*::*real*) *c*::*real*. *a* < (*0*::*real*) ∧ (*0*::*real*) < *b* ∧ (*0*::*real*) < *c* ∧ % *a* (*vector_sub ?u x*) = *vector_add* (% *b* (*vector_sub ?v x*)) (% *c* (*vector_sub ?w x*))) = (∃ *tt*::(*real*, *?'a*::*type*) *cart*. *IN tt* (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*aff_lt* (*INSERT x EMPTY*) (*INSERT ?u EMPTY*))))

thm Local_lemmas.AFF_GT_AFF_LT_INTERPRET2:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ (∃ (*a*::*real*) (*b*::*real*) *c*::*real*. *a* < (*0*::*real*) ∧ (*0*::*real*) < *b* ∧ (*0*::*real*) < *c* ∧ % *a* (*vector_sub ?u x*) = *vector_add* (% *b* (*vector_sub ?v x*)) (% *c* (*vector_sub ?w x*))) = (∃ *tt*::(*real*, *?'a*::*type*) *cart*. *IN tt* (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*aff_lt* (*INSERT x EMPTY*) (*INSERT ?u EMPTY*))))

thm Local_lemmas.EXISTS_IN:

((*?a*::*?'a*::*type* ⇒ *bool*) ≠ *EMPTY*) = (∃ *x*::*?'a*::*type*. *IN x ?a*)

thm Local_lemmas.AFF_GT_LT_INTER_SYM:

∀ *x*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x EMPTY*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*aff_lt* (*INSERT x EMPTY*) (*INSERT ?u EMPTY*)) ≠ *EMPTY* ⟶ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT x EMPTY*) (*INSERT ?u* (*INSERT ?v EMPTY*))) (*aff_lt* (*INSERT x EMPTY*) (*INSERT ?w EMPTY*)) ≠ *EMPTY*

thm Local_lemmas.EXISTS_INTERSECTION_PROPERPLY:

((*?z*::(*real*, *?'a*::*type*) *cart*) ≠ (*?x*::(*real*, *?'a*::*type*) *cart*) ∧ *DISJOINT* (*INSERT ?x EMPTY*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) ∧ *?x* ≠ (*?u*::(*real*, *?'a*::*type*) *cart*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT ?x EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*aff_lt* (*INSERT ?x EMPTY*) (*INSERT ?u EMPTY*)) ≠ *EMPTY* ∧ *IN ?z* (*hull affine* (*INSERT ?x* (*INSERT ?v* (*INSERT ?w EMPTY*))))) ∧ ¬ *collinear* (*INSERT ?x* (*INSERT ?v* (*INSERT ?w EMPTY*))) ⟶ (∃ (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *t*::(*real*, *?'a*::*type*) *cart*. *SUBSET* (*INSERT a* (*INSERT b EMPTY*)) (*INSERT ?u* (*INSERT ?v* (*INSERT ?w EMPTY*))) ∧ *IN t* (*HOL_Light_Import.INTER* (*aff* (*INSERT ?x* (*INSERT ?z EMPTY*))) (*conv0* (*INSERT a* (*INSERT b EMPTY*)))) ∧ ¬ *IN a* (*aff* (*INSERT ?x* (*INSERT ?z EMPTY*))))

thm Local_lemmas.LOFA_V_SUBSET_AFF_HULL:

∀ (*w*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *IN* (*v*, *w*) (*?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *convex_local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*) ∧ *IN* (*?u*::(*real*, *3*) *cart*) *?V* ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*aff_lt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?u EMPTY*)) ≠ *EMPTY* ⟶ *SUBSET ?V* (*hull affine* (*INSERT v* (*INSERT w* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*)))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w EMPTY*)))

thm Local_lemmas.DETER_RHO_NODE:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?FF* ⟶ *rho_node1 ?FF ?v* = *?w*

thm Local_lemmas.CARD_RECUSIVE_EQ:

∀ (*k*::*nat*) *f*::*?'a*::*type* ⇒ *?'a*::*type*. (*0*::*nat*) < *k* ⟶ (*CARD* (*GSPEC* (λ*GEN%PVAR%1846*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1846* (*n* < *k*) (*ITER n f* (*?x*::*?'a*::*type*)))) = *k*) = (*CARD* (*GSPEC* (λ*GEN%PVAR%1847*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1847* (*n* < *k* − (*1*::*nat*)) (*ITER n f ?x*))) = *k* − (*1*::*nat*) ∧ (∀ *i*<*k* − (*1*::*nat*). *ITER i f ?x* ≠ *ITER* (*k* − (*1*::*nat*)) *f ?x*))

thm Local_lemmas.LE_CARDV_IMP_CARD_DETERED:

(∀ *v*::*?'a*::*type*. *IN v* (*?V*::*?'a*::*type* ⇒ *bool*) ⟶ *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *v* = *?V*) ∧ *IN* (*?v*::*?'a*::*type*) *?V* ⟶ (∀ *l*≤*CARD ?V*. *CARD* (*GSPEC* (λ*GEN%PVAR%1849*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1849* (*n* < *l*) (*ITER n ?f ?v*))) = *l*)

thm Local_lemmas.LEMMA_SUBSET_ORBIT_MAP:

∀ (*p*::*?'a*::*type* ⇒ *?'a*::*type*) (*x*::*?'a*::*type*) *n*::*nat*. *SUBSET* (*GSPEC* (λ*GEN%PVAR%251*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%251* (*i* ≤ *n*) (*ITER i p x*))) (*orbit_map p x*)

thm Local_lemmas.LEMMA_SUBSET_ORBIT_MAP_LT:

SUBSET (*GSPEC* (λ*GEN%PVAR%1850*::*?'a*::*type*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%1850* (*i* < (*?n*::*nat*)) (*ITER i* (*?p*::*?'a*::*type* ⇒ *?'a*::*type*) (*?x*::*?'a*::*type*)))) (*orbit_map ?p ?x*)

thm Local_lemmas.LOOP_SET_DETER_FIRTS_ELMS:

(∀ *v*::*?'a*::*type*. *IN v* (*?V*::*?'a*::*type* ⇒ *bool*) ⟶ *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *v* = *?V*) ⟶ (∀ *v*::*?'a*::*type*. *IN v ?V* ⟶ *GSPEC* (λ*GEN%PVAR%1853*::*?'a*::*type*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1853* (*n* < *CARD ?V*) (*ITER n ?f v*)) = *?V*)

thm Local_lemmas.SIN_SUB_PERIODIC:

sin (?x::real) = − sin (?x − pi)

thm Local_lemmas.KCHMAMG:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ circular ?V ?E ⟶ (∀ v::(real, 3) cart. IN v ?V ⟶ interior_angle1 (vec (0::nat)) ?FF v = pi) ∧ (∃ A::(real, 3) cart ⇒ bool. plane A ∧ IN (vec (0::nat)) A ∧ SUBSET ?V A ∧ (∃ e::(real, 3) cart. (∀ x::(real, 3) cart. IN x A ⟶ dot e x = (0::real)) ∧ cyclic_set ?V (vec (0::nat)) e ∧ (∀ v::(real, 3) cart. IN v ?V ⟶ azim_cycle ?V (vec (0::nat)) e v = rho_node1 ?FF v ∧ azim (vec (0::nat)) e v (rho_node1 ?FF v) = dihV (vec (0::nat)) e v (rho_node1 ?FF v) ∧ azim (vec (0::nat)) e v (rho_node1 ?FF v) = arcV (vec (0::nat)) v (rho_node1 ?FF v) ∧ azim (vec (0::nat)) e v (rho_node1 ?FF v) < pi)))

thm Local_lemmas.CONVEX_SOME_WEDGE:

¬ collinear (INSERT (?v0.0::(real, 3) cart) (INSERT (?v1.0::(real, 3) cart) (INSERT (?w1.0::(real, 3) cart) EMPTY))) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT (?w2.0::(real, 3) cart) EMPTY))) ∧ (0::real) < azim ?v0.0 ?v1.0 ?w1.0 ?w2.0 ∧ azim ?v0.0 ?v1.0 ?w1.0 ?w2.0 < pi ⟶ convex (wedge ?v0.0 ?v1.0 ?w1.0 ?w2.0)

thm Local_lemmas.AZIM_EQ_0_GE_ALT2:

∀ (v0::(real, 3) cart) (v1::(real, 3) cart) (w::(real, 3) cart) x::(real, 3) cart. ¬ collinear (INSERT v0 (INSERT v1 (INSERT w EMPTY))) ⟶ (azim v0 v1 w x = (0::real)) = IN x (aff_ge (INSERT v0 (INSERT v1 EMPTY)) (INSERT w EMPTY))

thm Local_lemmas.WEDGE_GE_EQ_AFF_GE:

azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) (?w2.0::(real, 3) cart) < pi ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?w1.0 EMPTY))) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?w2.0 EMPTY))) ⟶ wedge_ge ?v0.0 ?v1.0 ?w1.0 ?w2.0 = aff_ge (INSERT ?v0.0 (INSERT ?v1.0 EMPTY)) (INSERT ?w1.0 (INSERT ?w2.0 EMPTY))

thm Local_lemmas.AZIM_PI_WEDGE_GE_SIN:

azim (?u::(real, 3) cart) (?v::(real, 3) cart) (?w::(real, 3) cart) (?ww::(real, 3) cart) = pi ⟶ wedge_ge ?u ?v ?w ?ww = GSPEC (λGEN%PVAR%1859::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%1859 ((0::real) ≤ sin (azim ?u ?v ?w x)) x)

thm Local_lemmas.AZIM_PI_WEDGE_GE_CROSS_DOT:

azim (?u::(real, 3) cart) (?v::(real, 3) cart) (?w::(real, 3) cart) (?ww::(real, 3) cart) = pi ⟶ wedge_ge ?u ?v ?w ?ww = GSPEC (λGEN%PVAR%1860::(real, 3) cart. ∃ x::(real, 3) cart. SETSPEC GEN%PVAR%1860 ((0::real) ≤ dot (cross (vector_sub ?v ?u) (vector_sub ?w ?u)) (vector_sub x ?u)) x)

thm Local_lemmas.AZIM_PI_CONVEX_WEDGE:

azim (?u::(real, 3) cart) (?v::(real, 3) cart) (?w::(real, 3) cart) (?ww::(real, 3) cart) = pi ⟶ convex (wedge_ge ?u ?v ?w ?ww)

thm Local_lemmas.CONVEX_WEDGE_LE_PI:

azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) (?w2.0::(real, 3) cart) ≤ pi ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?w1.0 EMPTY))) ∧ ¬ collinear (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?w2.0 EMPTY))) ⟶ convex (wedge_ge ?v0.0 ?v1.0 ?w1.0 ?w2.0)

thm Local_lemmas.wedge_in_fan_ge2:

wedge_in_fan_ge (?x::(real, 3) cart × (real, 3) cart) (?E::((real, 3) cart ⇒ bool) ⇒ bool) = (if (1::nat) < CARD (EE (fst ?x) ?E) then wedge_ge (vec (0::nat)) (fst ?x) (snd ?x) (azim_cycle (EE (fst ?x) ?E) (vec (0::nat)) (fst ?x) (snd ?x)) else GSPEC (λGEN%PVAR%1861::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1861 True x))

thm Local_lemmas.azim_in_fan2:

azim_in_fan (?x::(real, 3) cart × (real, 3) cart) (?E::((real, 3) cart ⇒ bool) ⇒ bool) = LET (λd::(real, 3) cart. LET_END (if (1::nat) < CARD (EE (fst ?x) ?E) then azim (vec (0::nat)) (fst ?x) (snd ?x) d else real_of_nat (2::nat) ∗ pi)) (azim_cycle (EE (fst ?x) ?E) (vec (0::nat)) (fst ?x) (snd ?x))

thm Local_lemmas.LOFA_IMP_NOT_INCLUDE_VEC0:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ ¬ IN (vec (0::nat)) ?V

thm Local_lemmas.AZIM_SPEC_DEGENERATE:

azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) ?v0.0 = (0::real) ∧ azim ?v0.0 ?v1.0 ?w1.0 ?v1.0 = (0::real)

thm Local_lemmas.CONDS_IN_CONV2:

(0::real) ≤ (?t2.0::real) ∧ (0::real) ≤ (?t3.0::real) ∧ ¬ (?t2.0 = (0::real) ∧ ?t3.0 = (0::real)) ⟶ IN (vector_add (% (?t2.0 / (?t2.0 + ?t3.0)) (?v::(real, ?'a::type) cart)) (% (?t3.0 / (?t2.0 + ?t3.0)) (?w::(real, ?'a::type) cart))) (conv (INSERT ?v (INSERT ?w EMPTY)))

thm Local_lemmas.AZIM_SPEC_DEGENERATE_conjunct1:

azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) ?v1.0 = (0::real)

thm Local_lemmas.AZIM_SPEC_DEGENERATE_conjunct0:

azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) ?v0.0 = (0::real)

thm Local_lemmas.PGSQVBL:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ SUBSET (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY)) ?V ∧ IN (?x::(real, 3) cart × (real, 3) cart) ?FF ⟶ SUBSET (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_ge ?x ?E)

thm Local_lemmas.FST_EQ_IF_SAME_SND:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?w1.0::(real, 3) cart, ?v::(real, 3) cart) ?FF ∧ IN (?w2.0::(real, 3) cart, ?v) ?FF ⟶ ?w1.0 = ?w2.0

thm Local_lemmas.PRE_IVS_RHO_NODE1_DETE:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?vv::(real, 3) cart, ?v::(real, 3) cart) ?FF ⟶ (SOME a::(real, 3) cart. IN (a, ?v) ?FF) = ?vv

thm Local_lemmas.IVS_RHO_NODE1_DETE:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?vv::(real, 3) cart, ?v::(real, 3) cart) ?FF ⟶ ivs_rho_node1 ?FF ?v = ?vv

thm Local_lemmas.AZIM_EQ_0_SYM2:

∀(w1::(real, 3) cart) w2::(real, 3) cart. (azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) w1 w2 = (0::real)) = (azim ?v0.0 ?v1.0 w2 w1 = (0::real))

thm Local_lemmas.LOCAL_FAN_IN_FF_IN_ORD_PAIRS2:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?x::(real, 3) cart, ?y::(real, 3) cart) ?FF ⟶ IN (INSERT ?x (INSERT ?y EMPTY)) ?E

thm Local_lemmas.INTERIOR_ANGLE1_POS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ (0::real) < interior_angle1 (vec (0::nat)) ?FF ?v

thm Local_lemmas.FAN_IMP_NOT_IN_AFF_GE:

FAN (?x::(real, ?'a::type) cart, ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ SUBSET (INSERT (?v::(real, ?'a::type) cart) (INSERT (?w::(real, ?'a::type) cart) EMPTY)) ?V ∧ ?v ≠ ?w ⟶ ¬ IN ?v (aff_ge (INSERT ?x EMPTY) (INSERT ?w EMPTY))

thm Local_lemmas.IN_AFF_LT_IMP_IN_CONV:

DISJOINT (INSERT (?x::(real, ?'a::type) cart) EMPTY) (INSERT (?b::(real, ?'a::type) cart) EMPTY) ∧ IN (?a::(real, ?'a::type) cart) (aff_lt (INSERT ?x

EMPTY) (*INSERT ?b EMPTY*)) ⟶ *IN ?x* (*conv0* (*INSERT ?a* (*INSERT ?b EMPTY*)))

thm Local_lemmas.FAN_SUB_NOT_EQ_COLL_IN_CONV0:

FAN (*?x*::(*real*, *?'a*::*type*) *cart*, *?V*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*, *?E*::((*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *SUBSET* (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?w*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) *?V* ∧ *?v* ≠ *?w* ∧ *collinear* (*INSERT ?x* (*INSERT ?v* (*INSERT ?w EMPTY*))) ⟶ *IN ?x* (*conv0* (*INSERT ?v* (*INSERT ?w EMPTY*)))

thm Local_lemmas.IN_CONV_LINE_SEPERATABLE:

IN (*?x*::(*real*, *?'a*::*type*) *cart*) (*conv0* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *aff* (*INSERT ?a* (*INSERT ?b EMPTY*)) = *HOL_Light_Import.UNION* (*aff_ge* (*INSERT ?x EMPTY*) (*INSERT ?a EMPTY*)) (*aff_ge* (*INSERT ?x EMPTY*) (*INSERT ?b EMPTY*))

thm Local_lemmas.CVLF_LF_F:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *local_fan* (*?V*, *?E*, *?FF*) ∧ *FAN* (*vec* (*0*::*nat*), *?V*, *?E*)

thm Local_lemmas.EMPTY_NOT_EXISTS_IN:

((*?a*::*?'a*::*type* ⇒ *bool*) = *EMPTY*) = (¬ (∃ *x*::*?'a*::*type*. *IN x ?a*))

thm Local_lemmas.LUNAR_IMP_INTERIOR_ANGLE1_EQ_PI:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V ?E* ⟶ *IN* (*vec* (*0*::*nat*)) (*conv0* (*INSERT ?v* (*INSERT ?w EMPTY*))) ∧ (∀ *u*::(*real*, *3*) *cart*. *IN u* (*DIFF ?V* (*INSERT ?v* (*INSERT ?w EMPTY*))) ⟶ *interior_angle1* (*vec* (*0*::*nat*)) *?FF u* = *pi* ∧ *IN* (*rho_node1 ?FF u*) (*aff* (*INSERT u* (*INSERT ?v* (*INSERT ?w EMPTY*)))) ∧ *IN* (*ivs_rho_node1 ?FF u*) (*aff* (*INSERT u* (*INSERT ?v* (*INSERT ?w EMPTY*)))))

thm Local_lemmas.AFF_GE_MONO_TRANS:

SUBSET (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*?X*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⟶ *SUBSET* (*aff_ge* (*DIFF ?X ?S*) (*HOL_Light_Import.UNION* (*?Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *?S*)) (*aff_ge ?X ?Y*)

thm Local_lemmas.AFF_GT_MONO_TRANS:

SUBSET (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*?X*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) ⟶ *SUBSET* (*aff_gt* (*DIFF ?X ?S*) (*HOL_Light_Import.UNION* (*?Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *?S*)) (*aff_gt ?X ?Y*)

thm Local_lemmas.CONV_UNION_SUB_AFF_GE:

SUBSET (*conv* (*HOL_Light_Import.UNION* (*?X*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*?Y*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*))) (*aff_ge ?X ?Y*)

thm Local_lemmas.CONV_MONO:

SUBSET (?S::(real, ?'a::type) cart ⇒ bool) (?SS::(real, ?'a::type) cart ⇒ bool) ⟶ SUBSET (conv ?S) (conv ?SS)

thm Local_lemmas.CONV3_SUBSET_AFF_GE_3S:

SUBSET (conv (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) EMPTY)))) (aff_ge (INSERT ?a (INSERT ?b (INSERT ?c EMPTY))) (?S::(real, ?'a::type) cart ⇒ bool))

thm Local_lemmas.P_SET3_IMP_SET2:

(∀ (a::?'a::type) (b::?'a::type) c::?'a::type. (?P::(?'a::type ⇒ bool) ⇒ bool) (INSERT a (INSERT b (INSERT c EMPTY)))) ⟶ (∀ (a::?'a::type) b::?'a::type. ?P (INSERT a (INSERT b EMPTY)))

thm Local_lemmas.CONV2_SUBSET_AFF_GE_2S:

∀ (a::(real, ?'a::type) cart) (b::(real, ?'a::type) cart) S::(real, ?'a::type) cart ⇒ bool. SUBSET (conv (INSERT a (INSERT b EMPTY))) (aff_ge (INSERT a (INSERT b EMPTY)) S)

thm Local_lemmas.X_IN_AFF_GE:

IN (?x::(real, ?'a::type) cart) (aff_ge (?S::(real, ?'a::type) cart ⇒ bool) (INSERT ?x EMPTY))

thm Local_lemmas.LOFA_IMP_BIJ_FF_V:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ BIJ fst ?FF ?V

thm Local_lemmas.LOFA_IMP_CARD_FF_V_EQ:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ CARD ?FF = CARD ?V

thm Local_lemmas.FIRST_IN_AFF:

IN (?a::(real, ?'a::type) cart) (aff (INSERT ?a (?S::(real, ?'a::type) cart ⇒ bool)))

thm Local_lemmas.HALF_CIRCULAR_IN_PLANE:

(convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E) ∧ (?n::nat) < CARD ?V ∧ ?w = ITER ?n (rho_node1 ?FF) ?v ⟶ SUBSET (GSPEC (λGEN%PVAR%1863::(real, 3) cart. ∃ l::nat. SETSPEC GEN%PVAR%1863 (l ≤ ?n) (ITER l (rho_node1 ?FF) ?v))) (aff (INSERT (vec (0::nat)) (INSERT ?v (INSERT (rho_node1 ?FF ?v) EMPTY))))

thm Local_lemmas.LOFA_IMP_AZIM_RHO_NODE_ST:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧

$cross\ ?v\ (rho\_node1\ ?FF\ ?v) = (?e::(real,\ 3)\ cart) \longrightarrow \neg\ (azim\ (vec\ (0::nat))\ ?e\ ?v\ (rho\_node1\ ?FF\ ?v) = (0::real) \lor azim\ (vec\ (0::nat))\ ?e\ ?v\ (rho\_node1\ ?FF\ ?v) = pi)$

thm Local_lemmas.LOFA_IMP_DIS_ELMS2:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \land IN\ (?v::(real,\ 3)\ cart)\ ?V \longrightarrow (\forall (i::nat)\ l::nat.\ i < l \land l < CARD\ ?FF \longrightarrow ITER\ l\ (rho\_node1\ ?FF)\ ?v \neq ITER\ i\ (rho\_node1\ ?FF)\ ?v)$

thm Local_lemmas.RHO_NODE1_MONO_WITH_AZIM:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \land IN\ (?v::(real,\ 3)\ cart)\ ?V \land GSPEC\ (\lambda GEN\%PVAR\%1864::(real,\ 3)\ cart.\ \exists\, n::nat.\ SETSPEC\ GEN\%PVAR\%1864\ (n \leq (?l::nat))\ (ITER\ n\ (rho\_node1\ ?FF)\ ?v)) = (?U::(real,\ 3)\ cart \Rightarrow bool) \land plane\ (?P::(real,\ 3)\ cart \Rightarrow bool) \land IN\ (vec\ (0::nat))\ ?P \land SUBSET\ ?U\ ?P \land cross\ ?v\ (rho\_node1\ ?FF\ ?v) = (?e::(real,\ 3)\ cart) \longrightarrow (\forall (n::nat)\ m::nat.\ n < m \land m < CARD\ ?V \land m \leq ?l \longrightarrow azim\ (vec\ (0::nat))\ ?e\ ?v\ (ITER\ n\ (rho\_node1\ ?FF)\ ?v) < azim\ (vec\ (0::nat))\ ?e\ ?v\ (ITER\ m\ (rho\_node1\ ?FF)\ ?v))$

thm Local_lemmas.DISJOINT_IMP_Z_IN_AFF_GT:

$DISJOINT\ (INSERT\ (?x::(real,\ ?'a::type)\ cart)\ (INSERT\ (?y::(real,\ ?'a::type)\ cart)\ EMPTY))\ (INSERT\ (?z::(real,\ ?'a::type)\ cart)\ EMPTY) \longrightarrow IN\ ?z\ (aff\_gt\ (INSERT\ ?x\ (INSERT\ ?y\ EMPTY))\ (INSERT\ ?z\ EMPTY))$

thm Local_lemmas.LOFA_IMP_DIS_ELMS23:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \land IN\ (?v::(real,\ 3)\ cart)\ ?V \longrightarrow (\forall (i::nat)\ l::nat.\ i < l \land l < CARD\ ?V \longrightarrow ITER\ l\ (rho\_node1\ ?FF)\ ?v \neq ITER\ i\ (rho\_node1\ ?FF)\ ?v)$

thm Local_lemmas.NOT_COLL_IMP_COPL:

$\neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (?v::(real,\ 3)\ cart)\ (INSERT\ (?w::(real,\ 3)\ cart)\ EMPTY))) \longrightarrow \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?v\ (INSERT\ ?w\ (INSERT\ (cross\ ?v\ ?w)\ EMPTY))))$

thm Local_lemmas.COLL_IFF_COLL_CROSS:

$collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ (?v::(real,\ 3)\ cart)\ (INSERT\ (?w::(real,\ 3)\ cart)\ EMPTY))) = collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?v\ (INSERT\ (cross\ ?v\ ?w)\ EMPTY)))$

thm Local_lemmas.LOCAL_FAN_CHARACTER_OF_RHO_NODE2:

$local\_fan\ (?V::(real,\ 3)\ cart \Rightarrow bool,\ ?E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool,\ ?FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool) \land IN\ (?v::(real,\ 3)\ cart)\ ?V \longrightarrow$

¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *?v* (*INSERT* (*rho_node1* *?FF* *?v*) *EMPTY*)))

thm Local_lemmas.LUNAR_IMP_HALF_CIRCLE_SUBSET_AFF_GT:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V* *?E* ⟶ (∃ *i*<*CARD* *?V*. *?w* = *ITER* *i* (*rho_node1* *?FF*) *?v* ∧ *SUBSET* (*GSPEC* (λ*GEN%PVAR%1867*::(*real*, *3*) *cart*. ∃ *l*::*nat*. *SETSPEC* *GEN%PVAR%1867* ((*0*::*nat*) < *l* ∧ *l* < *i*) (*ITER* *l* (*rho_node1* *?FF*) *?v*))) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *?v* *EMPTY*)) (*INSERT* (*rho_node1* *?FF* *?v*) *EMPTY*)))

thm Local_lemmas.LOOP_SET_ITER_CARD_ID:

(∀ *v*::*?'a*::*type*. *IN* *v* (*?V*::*?'a*::*type* ⇒ *bool*) ⟶ *orbit_map* (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) *v* = *?V*) ⟶ (∀ *v*::*?'a*::*type*. *IN* *v* *?V* ⟶ *ITER* (*CARD* *?V*) *?f* *v* = *v*)

thm Local_lemmas.LOFA_IMP_ITER_RHO_NODE_ID:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ (∀ *v*::(*real*, *3*) *cart*. *IN* *v* *?V* ⟶ *ITER* (*CARD* *?V*) (*rho_node1* *?FF*) *v* = *v*)

thm Local_lemmas.CONV_SUBSET_AFF_GE:

SUBSET (*conv* (*?SS*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*)) (*aff_ge* (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *?SS*)

thm Local_lemmas.CONV0_SUBSET_AFF_GT:

SUBSET (*conv0* (*?SS*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*)) (*aff_gt* (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *?SS*)

thm Local_lemmas.collinear_fan22:

collinear (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) = (*IN* *?u* (*aff* (*INSERT* *?x* (*INSERT* *?v* *EMPTY*))) ∨ *?x* = *?v*)

thm Local_lemmas.IN_CONV0_IMP_COLL_IFF:

IN (*?x*::(*real*, *?'a*::*type*) *cart*) (*conv0* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *collinear* (*INSERT* *?a* (*INSERT* *?x* (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) = *collinear* (*INSERT* *?a* (*INSERT* *?b* (*INSERT* *?v* *EMPTY*)))

thm Local_lemmas.IN_CONV0_IMP_COLL_ENDS_AFF:

IN (*?x*::(*real*, *?'a*::*type*) *cart*) (*conv0* (*INSERT* (*?a*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?b*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ *collinear* (*INSERT* *?a* (*INSERT* *?x* (*INSERT* (*?v*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) = *collinear* (*INSERT* *?b* (*INSERT* *?x* (*INSERT* *?v* *EMPTY*)))

thm Local_lemmas.IN_CONV0_IMP_AZIM_PI:

$\neg$ *collinear* (*INSERT* (*?x*::(*real*, *3*) *cart*) (*INSERT* (*?a*::(*real*, *3*) *cart*) (*INSERT* (*?e*::(*real*, *3*) *cart*) *EMPTY*))) $\wedge$ *IN ?x* (*conv0* (*INSERT ?a* (*INSERT* (*?b*::(*real*, *3*) *cart*) *EMPTY*))) $\longrightarrow$ *azim ?x ?e ?a ?b* = *pi*

thm Local_lemmas.AFF_GT_MONO:

$\forall$ *S*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*. *SUBSET S* (*?Y*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) $\longrightarrow$ *SUBSET* (*aff_gt* (*?X*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) *?Y*) (*aff_gt* (*HOL_Light_Import.UNION ?X S*) (*DIFF ?Y S*))

thm Local_lemmas.AFF_GT_SUB_AFF_UNION:

SUBSET (*aff_gt* (*?X*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*) (*?Y*::(*real*, *?'a*::*type*) *cart* $\Rightarrow$ *bool*)) (*aff* (*HOL_Light_Import.UNION ?X ?Y*))

thm Local_lemmas.SIN_AZIM_NEG_PI_LT:

(*sin* (*azim* (*?x*::(*real*, *3*) *cart*) (*?y*::(*real*, *3*) *cart*) (*?u*::(*real*, *3*) *cart*) (*?v*::(*real*, *3*) *cart*)) < (*0*::*real*)) = (*pi* < *azim ?x ?y ?u ?v*)

thm Local_lemmas.NEXT_OPOSITE_POINT_IS_NOT_IN_AFF_GT:

convex_local_fan (*?V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*, *?E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*, *?FF*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) $\wedge$ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V ?E* $\longrightarrow$ ($\exists$ *i*::*nat*. *?w* = *ITER i* (*rho_node1 ?FF*) *?v* $\wedge$ *i* + (*1*::*nat*) < *CARD ?V* $\wedge$ $\neg$ *IN* (*ITER* (*i* + (*1*::*nat*)) (*rho_node1 ?FF*) *?v*) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?v EMPTY*)) (*INSERT* (*rho_node1 ?FF ?v*) *EMPTY*)))

thm Local_lemmas.FOR_AFF_GT_NOT_INTERSECTION:

vector_add (% (*?a1.0*::*real*) (*?x*::(*real*, *?'a*::*type*) *cart*)) (*vector_add* (% (*?b1.0*::*real*) (*?y*::(*real*, *?'a*::*type*) *cart*)) (% (*?t*::*real*) (*?u*::(*real*, *?'a*::*type*) *cart*))) = *vector_add* (% (*?a2.0*::*real*) *?x*) (*vector_add* (% (*?b2.0*::*real*) *?y*) (% (*?tt*::*real*) (*?v*::(*real*, *?'a*::*type*) *cart*))) $\wedge$ (*0*::*real*) < *?t* $\wedge$ (*0*::*real*) < *?tt* $\wedge$ *?a1.0* + (*?b1.0* + *?t*) = (*1*::*real*) $\wedge$ *?a2.0* + (*?b2.0* + *?tt*) = (*1*::*real*) $\longrightarrow$ *?u* = *vector_add* (% ((*?a2.0* − *?a1.0*) / *?t*) *?x*) (*vector_add* (% ((*?b2.0* − *?b1.0*) / *?t*) *?y*) (% (*?tt* / *?t*) *?v*)) $\wedge$ (*?a2.0* − *?a1.0*) / *?t* + ((*?b2.0* − *?b1.0*) / *?t* + *?tt* / *?t*) = (*1*::*real*) $\wedge$ (*0*::*real*) < *?tt* / *?t*

thm Local_lemmas.NOT_COLL_RHONODE_SND_POINT:

convex_local_fan (*?V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*, *?E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*, *?FF*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) $\wedge$ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V ?E* $\longrightarrow$ $\neg$ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?v* (*INSERT* (*rho_node1 ?FF ?w*) *EMPTY*)))

thm Local_lemmas.NOT_INTERSECTION_BWT_AFF_GTS:

convex_local_fan (*?V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*, *?E*::((*real*, *3*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*, *?FF*::(*real*, *3*) *cart* $\times$ (*real*, *3*) *cart* $\Rightarrow$ *bool*) $\wedge$ *lunar* (*?v*::(*real*, *3*) *cart*,

?w::(real, 3) cart) ?V ?E ⟶ HOL_Light_Import.INTER (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?w) EMPTY)) (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?v) EMPTY)) = EMPTY

thm Local_lemmas.LUNAR_COMM:

lunar (?v::(real, ?'a::type) cart, ?w::(real, ?'a::type) cart) (?V::(real, ?'a::type) cart ⇒ bool) (?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) = lunar (?w, ?v) ?V ?E

thm Local_lemmas.CONV0_AFF_GT_EQ:

IN (?x::(real, ?'a::type) cart) (conv0 (INSERT (?a::(real, ?'a::type) cart) (INSERT (?b::(real, ?'a::type) cart) EMPTY))) ∧ ¬ collinear (INSERT ?a (INSERT ?x (INSERT (?v::(real, ?'a::type) cart) EMPTY))) ⟶ aff_gt (INSERT ?x (INSERT ?a EMPTY)) (INSERT ?v EMPTY) = aff_gt (INSERT ?x (INSERT ?a (INSERT ?b EMPTY))) (INSERT ?v EMPTY)

thm Local_lemmas.AFF_GT_SAME_WITH_ENDS:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?w) EMPTY) = aff_gt (INSERT (vec (0::nat)) (INSERT ?w EMPTY)) (INSERT (rho_node1 ?FF ?w) EMPTY)

thm Local_lemmas.CRAZY_THMMM:

IN (?w::(real, ?'a::type) cart) (aff (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) EMPTY))) ∧ IN (?z::(real, ?'a::type) cart) (aff_gt (INSERT ?b (INSERT ?c EMPTY)) (INSERT ?w EMPTY)) ⟶ IN ?z (aff (INSERT ?b (INSERT ?c EMPTY)))

thm Local_lemmas.USEFULL_THHM:

IN (?z::(real, ?'a::type) cart) (aff (INSERT (?b::(real, ?'a::type) cart) (INSERT (?c::(real, ?'a::type) cart) EMPTY))) ∧ IN ?z (aff_gt (INSERT ?b (INSERT ?c EMPTY)) (INSERT (?w::(real, ?'a::type) cart) EMPTY)) ⟶ IN ?w (aff (INSERT ?b (INSERT ?c EMPTY)))

thm Local_lemmas.COLL_IN_AFF_GT_TOO:

¬ collinear (INSERT (?x::(real, ?'a::type) cart) (INSERT (?y::(real, ?'a::type) cart) (INSERT (?z::(real, ?'a::type) cart) EMPTY))) ∧ IN (?a::(real, ?'a::type) cart) (aff_gt (INSERT ?x (INSERT ?y EMPTY)) (INSERT ?z EMPTY)) ⟶ ¬ collinear (INSERT ?x (INSERT ?y (INSERT ?a EMPTY)))

thm Local_lemmas.AFF_GT_IN_IMP_SUBSET:

¬ collinear (INSERT (?x::(real, ?'a::type) cart) (INSERT (?y::(real, ?'a::type) cart) (INSERT (?z::(real, ?'a::type) cart) EMPTY))) ∧ IN (?a::(real, ?'a::type) cart) (aff_gt (INSERT ?x (INSERT ?y EMPTY)) (INSERT ?z EMPTY)) ⟶

SUBSET (*aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?a EMPTY*)) (*aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?z EMPTY*))

thm Local_lemmas.FOR_AFF_GT_NOT_INTERSECTION2:

vector_add (% (*?a1.0*::*real*) (*?x*::(*real*, *?'a*::*type*) *cart*)) (*vector_add* (% (*?b1.0*::*real*) (*?y*::(*real*, *?'a*::*type*) *cart*)) (% (*?t*::*real*) (*?u*::(*real*, *?'a*::*type*) *cart*))) = (*?v*::(*real*, *?'a*::*type*) *cart*) ∧ (*0*::*real*) < *?t* ∧ *?a1.0* + (*?b1.0* + *?t*) = (*1*::*real*) ⟶ *?u* = *vector_add* (% (((*0*::*real*) − *?a1.0*) / *?t*) *?x*) (*vector_add* (% (((*0*::*real*) − *?b1.0*) / *?t*) *?y*) (% ((*1*::*real*) / *?t*) *?v*)) ∧ ((*0*::*real*) − *?a1.0*) / *?t* + (((*0*::*real*) − *?b1.0*) / *?t* + (*1*::*real*) / *?t*) = (*1*::*real*) ∧ (*0*::*real*) < (*1*::*real*) / *?t*

thm Local_lemmas.INVS_IN_AFF_GT:

¬ *collinear* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?z*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ∧ *IN* (*?a*::(*real*, *?'a*::*type*) *cart*) (*aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?z EMPTY*)) ⟶ *IN ?z* (*aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?a EMPTY*))

thm Local_lemmas.COLL_IN_AFF_GT_AFF_GT_EQ:

¬ *collinear* (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?z*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ∧ *IN* (*?a*::(*real*, *?'a*::*type*) *cart*) (*aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?z EMPTY*)) ⟶ *aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?z EMPTY*) = *aff_gt* (*INSERT ?x* (*INSERT ?y EMPTY*)) (*INSERT ?a EMPTY*)

thm Local_lemmas.NEXT_OPOSITE_POINT_IS_NOT_IN_AFF_GT2:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V ?E* ⟶ (∃ *i*::*nat*. *?w* = *ITER i* (*rho_node1 ?FF*) *?v* ∧ *i* + (*1*::*nat*) < *CARD ?V* ∧ *i* ≠ (*0*::*nat*) ∧ *i* ≠ (*1*::*nat*) ∧ ¬ *IN* (*ITER* (*i* + (*1*::*nat*)) (*rho_node1 ?FF*) *?v*) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?v EMPTY*)) (*INSERT* (*rho_node1 ?FF ?v*) *EMPTY*)))

thm Local_lemmas.LUNAR_IMP_HALF_CIRCLE_SUBSET_AFF_GT100:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *lunar* (*?v*::(*real*, *3*) *cart*, *?w*::(*real*, *3*) *cart*) *?V ?E* ⟶ (∃ *i*<*CARD ?V*. *i* ≠ (*0*::*nat*) ∧ *i* ≠ (*1*::*nat*) ∧ *?w* = *ITER i* (*rho_node1 ?FF*) *?v* ∧ *SUBSET* (*GSPEC* (λ*GEN%PVAR%1872*::(*real*, *3*) *cart*. ∃ *l*::*nat*. *SETSPEC GEN%PVAR%1872* ((*0*::*nat*) < *l* ∧ *l* < *i*) (*ITER l* (*rho_node1 ?FF*) *?v*))) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?v EMPTY*)) (*INSERT* (*rho_node1 ?FF ?v*) *EMPTY*)))

thm Local_lemmas.IVS_RHO_IDD:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ *ivs_rho_node1 ?FF* (*rho_node1 ?FF ?v*) = *?v*

thm Local_lemmas.AFF_IVS_RHO_NODE_EQQ:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ aff_gt (INSERT (vec (0::nat)) (INSERT ?w EMPTY)) (INSERT (rho_node1 ?FF ?w) EMPTY) = aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (ivs_rho_node1 ?FF ?v) EMPTY)

thm Local_lemmas.LOFA_IMP_LT_CARD_SET_V:

∀ v::(real, 3) cart. local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN v ?V ⟶ GSPEC (λGEN%PVAR%1874::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1874 (n < CARD ?V) (ITER n (rho_node1 ?FF) v)) = ?V

thm Local_lemmas.NOT_COLL_IMP_NOT_AFF_SUB:

∀ v::(real, ?'a::type) cart. ¬ collinear (INSERT (?x::(real, ?'a::type) cart) (INSERT (?y::(real, ?'a::type) cart) (INSERT (?z::(real, ?'a::type) cart) EMPTY))) ∧ IN v (aff (INSERT ?x (INSERT ?y EMPTY))) ⟶ ¬ IN v (aff_gt (INSERT ?x (INSERT ?y EMPTY)) (INSERT ?z EMPTY))

thm Local_lemmas.HALP_CIRCLE_IS_INTERSECTION:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ (∃ i<CARD ?V. i ≠ (0::nat) ∧ i ≠ (1::nat) ∧ ?w = ITER i (rho_node1 ?FF) ?v ∧ GSPEC (λGEN%PVAR%1876::(real, 3) cart. ∃ l::nat. SETSPEC GEN%PVAR%1876 ((0::nat) < l ∧ l < i) (ITER l (rho_node1 ?FF) ?v)) = HOL_Light_Import.INTER (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?v) EMPTY)) ?V)

thm Local_lemmas.CONVEX_LOFA_IMP_INANGLE_LE_PI:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ interior_angle1 (vec (0::nat)) ?FF ?v ≤ pi

thm Local_lemmas.X_IN_AFF_GT_X:

IN (?x::(real, ?'a::type) cart) (aff_gt (?S::(real, ?'a::type) cart ⇒ bool) (INSERT ?x EMPTY))

thm Local_lemmas.IVS_RNODE_IN_AFF_V:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ IN (ivs_rho_node1 ?FF ?w) (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?v) EMPTY))

thm Local_lemmas.AZIM_LE_PI_EQ_DIHV:

¬ collinear (INSERT (?a::(real, 3) cart) (INSERT (?b::(real, 3) cart) (INSERT (?x::(real, 3) cart) EMPTY))) ∧ ¬ collinear (INSERT ?a (INSERT ?b (INSERT

(?y::(real, 3) cart) EMPTY))) ⟶ azim ?a ?b ?x ?y ≤ pi ⟶ azim ?a ?b ?x ?y = dihV ?a ?b ?x ?y

thm Local_lemmas.LOFA_IMP_NOT_COLL_IVS:

∀ v::(real, 3) cart. local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN v ?V ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT (ivs_rho_node1 ?FF v) EMPTY)))

thm Local_lemmas.DIHV_NOT_CHANGE:

(0::real) < (?c::real) ∧ (?a::real) + ((?b::real) + ?c) = (1::real) ⟶ dihV (?x::(real, ?'a::type) cart) (?y::(real, ?'a::type) cart) (vector_add (% ?a ?x) (vector_add (% ?b ?y) (% ?c (?v::(real, ?'a::type) cart)))) (?w::(real, ?'a::type) cart) = dihV ?x ?y ?v ?w

thm Local_lemmas.LUNAR_IMP_INTERIOR_ANGLE_EQQ:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ interior_angle1 (vec (0::nat)) ?FF ?v = interior_angle1 (vec (0::nat)) ?FF ?w

thm Local_lemmas.HKIRPEP:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ lunar (?v::(real, 3) cart, ?w::(real, 3) cart) ?V ?E ⟶ (∀ u::(real, 3) cart. IN u (DIFF ?V (INSERT ?v (INSERT ?w EMPTY))) ⟶ interior_angle1 (vec (0::nat)) ?FF u = pi) ∧ (0::real) < interior_angle1 (vec (0::nat)) ?FF ?v ∧ interior_angle1 (vec (0::nat)) ?FF ?v ≤ pi ∧ interior_angle1 (vec (0::nat)) ?FF ?v = interior_angle1 (vec (0::nat)) ?FF ?w ∧ (∃ i<CARD ?V. i ≠ (0::nat) ∧ i ≠ (1::nat) ∧ ?w = ITER i (rho_node1 ?FF) ?v ∧ GSPEC (λGEN%PVAR%1877::(real, 3) cart. ∃ l::nat. SETSPEC GEN%PVAR%1877 ((0::nat) < l ∧ l < i) (ITER l (rho_node1 ?FF) ?v)) = HOL_Light_Import.INTER (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (rho_node1 ?FF ?v) EMPTY)) ?V) ∧ (∃ j<CARD ?V. j ≠ (0::nat) ∧ j ≠ (1::nat) ∧ ?v = ITER j (rho_node1 ?FF) ?w ∧ GSPEC (λGEN%PVAR%1878::(real, 3) cart. ∃ l::nat. SETSPEC GEN%PVAR%1878 ((0::nat) < l ∧ l < j) (ITER l (rho_node1 ?FF) ?w)) = HOL_Light_Import.INTER (aff_gt (INSERT (vec (0::nat)) (INSERT ?v EMPTY)) (INSERT (ivs_rho_node1 ?FF ?v) EMPTY)) ?V)

thm Local_lemmas.NEXT_PRO_IMP_ALLS:

∀ le::?'a::type ⇒ ?'a::type ⇒ bool. (∀ i::nat. le ((?f::nat ⇒ ?'a::type) i) (?f (i + (1::nat)))) ∧ (∀ (a::?'a::type) (b::?'a::type) c::?'a::type. le a b ∧ le b c ⟶ le a c) ⟶ (∀ (i::nat) j::nat. i < j ⟶ le (?f i) (?f j))

thm Counting_spheres.EMPTY_NOT_EXISTS_IN:

((?a::?'a::type ⇒ bool) = EMPTY) = (¬ (∃ x::?'a::type. IN x ?a))

thm Local_lemmas.FINITE_CARD1_IMP_SINGLETON:

FINITE (?S::?'a::type ⇒ bool) ∧ CARD ?S = (1::nat) ⟶ (∃x::?'a::type. ?S = INSERT x EMPTY)

thm Local_lemmas.SURJ_IMP_FINITE:

SURJ (?f::?'b::type ⇒ ?'a::type) (?A::?'b::type ⇒ bool) (?B::?'a::type ⇒ bool) ∧ FINITE ?A ⟶ FINITE ?B

thm Local_lemmas.BIJ_FINITE_TOO:

BIJ (?f::?'b::type ⇒ ?'a::type) (?X::?'b::type ⇒ bool) (?Y::?'a::type ⇒ bool) ∧ FINITE ?X ⟶ FINITE ?Y

thm Local_lemmas.EQ_IFF_IMP:

(∀(a::?'a::type) b::?'a::type. (?P::?'a::type ⇒ ?'a::type ⇒ bool) a b = ?P b a) = (∀(a::?'a::type) b::?'a::type. ?P a b ⟶ ?P b a)

thm Local_lemmas.NOT_EMP_INJ_IMP_SURJ:

LET (λf1::?'b::type ⇒ ?'a::type. LET_END ((?X::?'a::type ⇒ bool) ≠ EMPTY ∧ INJ (?f::?'a::type ⇒ ?'b::type) ?X (?Y::?'b::type ⇒ bool) ⟶ SURJ f1 ?Y ?X)) (λy::?'b::type. if ∃x::?'a::type. IN x ?X ∧ ?f x = y then SOME x::?'a::type. IN x ?X ∧ ?f x = y else SOME x::?'a::type. IN x ?X)

thm Local_lemmas.LOFA_V_NOT_EMP:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ ?V ≠ EMPTY

thm Local_lemmas.LOCAL_FAN_FINITE_V:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ FINITE ?V

thm Local_lemmas.ITER_CARD_MINUS1_EQ_IVS_RN1:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀v::(real, 3) cart. IN v ?V ⟶ ITER (CARD ?V − (1::nat)) (rho_node1 ?FF) v = ivs_rho_node1 ?FF v)

thm Local_lemmas.FIRST_EQ0_LAST_LT_PI:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v0.0::(real, 3) cart) ?V ∧ CARD ?V = (?k::nat) ∧ (∀i::nat. ITER i (rho_node1 ?FF) ?v0.0 = (?vv::nat ⇒ (real, 3) cart) i) ∧ (∀i::nat. azim (vec (0::nat)) ?v0.0 (?vv (1::nat)) (?vv i) = (?bta::nat ⇒ real) i) ⟶ ?bta (1::nat) = (0::real) ∧ ?bta (?k − (1::nat)) ≤ pi

thm Local_lemmas.PROJEC_EQ_0_IFF_COLL:

(projection (?e::(real, ?'a::type) cart) (?x::(real, ?'a::type) cart) = vec (0::nat)) = (∃ *t::real. ?x = % t ?e*)

thm Local_lemmas.DETERMINE_WEDGE_IN_FAN:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) ?FF ⟶ wedge_in_fan_ge ?x ?E = wedge_ge (vec (0::nat)) (fst ?x) (snd ?x) (azim_cycle (EE (fst ?x) ?E) (vec (0::nat)) (fst ?x) (snd ?x))

thm Local_lemmas.LOCAL_FAN_IMP_IN_V2:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) ?FF ⟶ IN (fst ?x) ?V ∧ IN (snd ?x) ?V

thm Local_lemmas.LOFA_DETERMINE_AZIM_IN_FA:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?x::(real, 3) cart × (real, 3) cart) ?FF ⟶ azim_in_fan ?x ?E = azim (vec (0::nat)) (fst ?x) (snd ?x) (azim_cycle (EE (fst ?x) ?E) (vec (0::nat)) (fst ?x) (snd ?x))

thm Local_lemmas.PRIOR_TO_LESS_THAN_PI_LEMMA:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ (∀ *w::(real, 3) cart. IN w ?V ⟶ azim (vec (0::nat)) ?v (rho_node1 ?FF ?v) w ≤ azim (vec (0::nat)) ?v (rho_node1 ?FF ?v) (azim_cycle (EE ?v ?E) (vec (0::nat)) ?v (rho_node1 ?FF ?v)))*

thm Local_lemmas.IN_V_IMP_AZIM_LESS_PI:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ (∀ *w::(real, 3) cart. IN w ?V ⟶ azim (vec (0::nat)) ?v (rho_node1 ?FF ?v) w ≤ pi)*

thm Local_lemmas.NEXT_PRO_IMP_ALLS_STRICT:

∀ *le::?'a::type ⇒ ?'a::type ⇒ bool.* (∀ *i::nat. i + (1::nat) < (?k::nat) ⟶ le ((?f::nat ⇒ ?'a::type) i) (?f (i + (1::nat))))* ∧ (∀ *(a::?'a::type) (b::?'a::type) c::?'a::type. le a b ∧ le b c ⟶ le a c)* ⟶ (∀ *(i::nat) j::nat. i < j ∧ j < ?k ⟶ le (?f i) (?f j))*

thm Local_lemmas.LOFA_IMP_V_DIFF:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀ *v::(real, 3) cart. IN v ?V ⟶ v ≠ vec (0::nat))*

thm Local_lemmas.SIN_AZIM_POS_PI_LT:

((0::real) ≤ sin (azim (?x::(real, 3) cart) (?y::(real, 3) cart) (?u::(real, 3) cart) (?v::(real, 3) cart))) = (azim ?x ?y ?u ?v ≤ pi)

thm Local_lemmas.SIN_AZIM_MUTUAL_SROSS:

(*sin* (*azim* (*vec* (*0*::*nat*)) (*?u*::(*real*, *3*) *cart*) (*?v*::(*real*, *3*) *cart*) (*?w*::(*real*, *3*) *cart*)) < (*0*::*real*)) = (*dot* (*cross* *?u* *?v*) *?w* < (*0*::*real*)) ∧ ((*0*::*real*) < *sin* (*azim* (*vec* (*0*::*nat*)) *?u* *?v* *?w*)) = ((*0*::*real*) < *dot* (*cross* *?u* *?v*) *?w*)

thm Local_lemmas.VECTOR_ADD_LDISTRIB1:

% (*?c*::*real*) (*vector_add* (*?x*::(*real*, *?'b*::*type*) *cart*) (*?y*::(*real*, *?'b*::*type*) *cart*)) = *vector_add* (% *?c* *?x*) (% *?c* *?y*) ∧ (∀ (*a*::*real*) (*b*::*real*) *x*::(*real*, *?'a*::*type*) *cart*. % *a* (% *b* *x*) = % (*a* ∗ *b*) *x*)

thm Local_lemmas.OPPOSITE_SIDES_IMP_INTER:

IN (*?a*::(*real*, *?'a*::*type*) *cart*) (*aff_gt* (*INSERT* (*?va*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?vb*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) (*INSERT* (*?vc*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) ∧ *IN* (*?b*::(*real*, *?'a*::*type*) *cart*) (*aff_lt* (*INSERT* *?va* (*INSERT* *?vb* *EMPTY*)) (*INSERT* *?vc* *EMPTY*)) ∧ *DISJOINT* (*INSERT* *?va* (*INSERT* *?vb* *EMPTY*)) (*INSERT* *?vc* *EMPTY*) ⟶ *HOL_Light_Import.INTER* (*conv0* (*INSERT* *?a* (*INSERT* *?b* *EMPTY*))) (*aff* (*INSERT* *?va* (*INSERT* *?vb* *EMPTY*))) ≠ *EMPTY*

thm Local_lemmas.DISJOINT_DOUBLE_SING:

DISJOINT (*INSERT* (*?a*::*?'a*::*type*) *EMPTY*) (*INSERT* (*?b*::*?'a*::*type*) *EMPTY*) = (*?a* ≠ *?b*)

thm Local_lemmas.AFF_IN_TWO_PARTS:

aff (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) = *HOL_Light_Import.UNION* (*aff_ge* (*INSERT* *?x* *EMPTY*) (*INSERT* *?y* *EMPTY*)) (*aff_lt* (*INSERT* *?x* *EMPTY*) (*INSERT* *?y* *EMPTY*))

thm Local_lemmas.INTER_UNION_EMPTY:

(*HOL_Light_Import.INTER* (*?X*::*?'a*::*type* ⇒ *bool*) (*HOL_Light_Import.UNION* (*?A*::*?'a*::*type* ⇒ *bool*) (*?B*::*?'a*::*type* ⇒ *bool*)) = *EMPTY*) = (*HOL_Light_Import.INTER* *?X* *?A* = *EMPTY* ∧ *HOL_Light_Import.INTER* *?X* *?B* = *EMPTY*)

thm Local_lemmas.CONV0_SUB_CONV:

IN (*?x*::(*real*, *?'a*::*type*) *cart*) (*conv0* (*?S*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*)) ⟶ *IN* *?x* (*conv* *?S*)

thm Local_lemmas.SIN_AZIM_MUTUAL_SROSS_conjunct1:

((*0*::*real*) < *sin* (*azim* (*vec* (*0*::*nat*)) (*?u*::(*real*, *3*) *cart*) (*?v*::(*real*, *3*) *cart*) (*?w*::(*real*, *3*) *cart*))) = ((*0*::*real*) < *dot* (*cross* *?u* *?v*) *?w*)

thm Local_lemmas.SIN_AZIM_MUTUAL_SROSS_conjunct0:

(*sin* (*azim* (*vec* (*0*::*nat*)) (*?u*::(*real*, *3*) *cart*) (*?v*::(*real*, *3*) *cart*) (*?w*::(*real*, *3*) *cart*)) < (*0*::*real*)) = (*dot* (*cross* *?u* *?v*) *?w* < (*0*::*real*))

thm Local_lemmas.MONO_AZIM_AS_BTA_I:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v0.0::(real, 3) cart) ?V ∧ CARD ?V = (?k::nat) ∧ (∀ v::(real, 3) cart. IN v ?V ∧ v ≠ ?v0.0 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?v0.0 (INSERT v EMPTY)))) ∧ (∀ i::nat. ITER i (rho_node1 ?FF) ?v0.0 = (?vv::nat ⇒ (real, 3) cart) i) ∧ (∀ i::nat. azim (vec (0::nat)) ?v0.0 (?vv (1::nat)) (?vv i) = (?bta::nat ⇒ real) i) ⟶ (∀ (i::nat) j::nat. i < j ∧ j < ?k ⟶ ?bta i ≤ ?bta j)

thm Local_lemmas.SUCCESSIVE_RHO_NODE1_AFF_LT:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1883::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1883 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ⟶ (∀ i::nat. (0::nat) < i ∧ i < ?l ⟶ IN (ITER (i − (1::nat)) (rho_node1 ?FF) ?v) (aff_lt (INSERT (vec (0::nat)) (INSERT (ITER i (rho_node1 ?FF) ?v) EMPTY)) (INSERT (ITER (i + (1::nat)) (rho_node1 ?FF) ?v) EMPTY)))

thm Local_lemmas.IN_OPOSITE_SIDE_IMP_INTER:

IN (?b::(real, ?'a::type) cart) (aff_lt (INSERT (?va::(real, ?'a::type) cart) (INSERT (?vb::(real, ?'a::type) cart) EMPTY)) (INSERT (?vc::(real, ?'a::type) cart) EMPTY)) ∧ DISJOINT (INSERT ?va (INSERT ?vb EMPTY)) (INSERT ?vc EMPTY) ⟶ HOL_Light_Import.INTER (conv0 (INSERT ?vc (INSERT ?b EMPTY))) (aff (INSERT ?va (INSERT ?vb EMPTY))) ≠ EMPTY

thm Local_lemmas.TWO_SIDES_SUCESSIVE:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1884::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1884 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ⟶ (∀ i::nat. (0::nat) < i ∧ i < ?l ⟶ HOL_Light_Import.INTER (conv0 (INSERT (ITER (i − (1::nat)) (rho_node1 ?FF) ?v) (INSERT (ITER (i + (1::nat)) (rho_node1 ?FF) ?v) EMPTY))) (aff (INSERT (vec (0::nat)) (INSERT (ITER i (rho_node1 ?FF) ?v) EMPTY))) ≠ EMPTY)

thm Local_lemmas.LOCAL_FAN_CHARACTER_OF_RHO_NODE3:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ (∀ i::nat. ¬ collinear (INSERT (vec (0::nat)) (INSERT (ITER i (rho_node1 ?FF) ?v) (INSERT (ITER (i + (1::nat)) (rho_node1 ?FF) ?v) EMPTY))))

thm Local_lemmas.CNVX_IMP_INTERIOR_ANGLE_PI:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ GSPEC (λGEN%PVAR%1885::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1885 (n ≤ (?l::nat)) (ITER n (rho_node1 ?FF) ?v)) = (?U::(real, 3) cart ⇒ bool) ∧ plane (?P::(real, 3) cart ⇒ bool) ∧ IN (vec (0::nat)) ?P ∧ SUBSET ?U ?P ⟶ (∀ i::nat. (0::nat) < i ∧ i < ?l ⟶ interior_angle1 (vec (0::nat)) ?FF (ITER i (rho_node1 ?FF) ?v) = pi)

thm Local_lemmas.INSERT_UNION2:

HOL_Light_Import.UNION (INSERT (?x::?'a::type) (?S::?'a::type ⇒ bool)) (?U::?'a::type ⇒ bool) = HOL_Light_Import.UNION ?S (INSERT ?x ?U)

thm Local_lemmas.EGHNAVX:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v0.0::(real, 3) cart) ?V ∧ CARD ?V = (?k::nat) ∧ (∀ v::(real, 3) cart. IN v ?V ∧ v ≠ ?v0.0 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?v0.0 (INSERT v EMPTY)))) ∧ (∀ i::nat. ITER i (rho_node1 ?FF) ?v0.0 = (?vv::nat ⇒ (real, 3) cart) i) ∧ (∀ i::nat. azim (vec (0::nat)) ?v0.0 (?vv (1::nat)) (?vv i) = (?bta::nat ⇒ real) i) ⟶ ?bta (1::nat) = (0::real) ∧ ?bta (?k − (1::nat)) ≤ pi ∧ (∀ (i::nat) j::nat. i < j ∧ j < ?k ⟶ ?bta i ≤ ?bta j) ∧ (?bta (?i::nat) = (0::real) ∧ ?i < ?k ⟶ (∀ j::nat. (0::nat) < j ∧ j < ?i ⟶ interior_angle1 (vec (0::nat)) ?FF (?vv j) = pi) ∧ SUBSET (GSPEC (λGEN%PVAR%1893::(real, 3) cart. ∃ k::nat. SETSPEC GEN%PVAR%1893 ((0::nat) < k ∧ k ≤ ?i) (?vv k))) (aff_gt (INSERT (vec (0::nat)) (INSERT ?v0.0 EMPTY)) (INSERT (?vv (1::nat)) EMPTY))) ∧ (?bta ?i = ?bta (?k − (1::nat)) ∧ (0::nat) < ?i ∧ ?i < ?k − (1::nat) ⟶ (∀ j::nat. ?i < j ∧ j < ?k ⟶ interior_angle1 (vec (0::nat)) ?FF (?vv j) = pi) ∧ SUBSET (GSPEC (λGEN%PVAR%1894::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1894 (?i ≤ n ∧ n < ?k) (?vv n))) (aff_gt (INSERT (vec (0::nat)) (INSERT ?v0.0 EMPTY)) (INSERT (?vv (?k − (1::nat))) EMPTY)))

thm DEF_spherical_map:

spherical_map = (λ(_6553052::(real, ?'a::type) cart × (real, ?'a::type) cart × (real, ?'a::type) cart) _6553053::real × real × real. vector_add (% (fst _6553053 ∗ (cos (fst (snd _6553053)) ∗ sin (snd (snd _6553053)))) (fst _6553052)) (vector_add (% (fst _6553053 ∗ (sin (fst (snd _6553053)) ∗ sin (snd (snd _6553053)))) (fst (snd _6553052))) (% (fst _6553053 ∗ cos (snd (snd _6553053))) (snd (snd _6553052)))))

thm Local_lemmas1.spherical_map:

∀ (e1::(real, ?'a::type) cart) (theta::real) (e2::(real, ?'a::type) cart) (r::real) (phi::real) e3::(real, ?'a::type) cart. spherical_map (e1, e2, e3) (r, theta, phi) = vector_add (% (r ∗ (cos theta ∗ sin phi)) e1) (vector_add (% (r ∗ (sin theta ∗ sin phi)) e2) (% (r ∗ cos phi) e3))

thm DEF_lunar_deform:

lunar_deform = (λ(_6553084::(real, 3) cart × (real, 3) cart × (real, 3) cart) (_6553085::real) _6553086::(real, 3) cart. vector_add (% (vector_norm _6553086

∗ (cos (((1::real) − _6553085) ∗ azim (vec (0::nat)) (snd (snd _6553084)) (fst _6553084) _6553086) ∗ sin (arcV (vec (0::nat)) (snd (snd _6553084)) _6553086))) (fst _6553084)) (vector_add (% (vector_norm _6553086 ∗ (sin (((1::real) − _6553085) ∗ azim (vec (0::nat)) (snd (snd _6553084)) (fst _6553084) _6553086) ∗ sin (arcV (vec (0::nat)) (snd (snd _6553084)) _6553086))) (fst (snd _6553084))) (% (vector_norm _6553086 ∗ cos (arcV (vec (0::nat)) (snd (snd _6553084)) _6553086)) (snd (snd _6553084)))))

thm Local_lemmas1.lunar_deform:

∀ (t::real) (e1::(real, 3) cart) (e2::(real, 3) cart) (x::(real, 3) cart) e3::(real, 3) cart. lunar_deform (e1, e2, e3) t x = vector_add (% (vector_norm x ∗ (cos (((1::real) − t) ∗ azim (vec (0::nat)) e3 e1 x) ∗ sin (arcV (vec (0::nat)) e3 x))) e1) (vector_add (% (vector_norm x ∗ (sin (((1::real) − t) ∗ azim (vec (0::nat)) e3 e1 x) ∗ sin (arcV (vec (0::nat)) e3 x))) e2) (% (vector_norm x ∗ cos (arcV (vec (0::nat)) e3 x)) e3))

thm Local_lemmas1.LUNAR_DEFORM_PRESERVE_NORM:

orthonormal (?e1.0::(real, 3) cart) (?e2.0::(real, 3) cart) (?e3.0::(real, 3) cart) ⟶ vector_norm (lunar_deform (?e1.0, ?e2.0, ?e3.0) (?t::real) (?x::(real, 3) cart)) = vector_norm ?x

thm Local_lemmas1.ACR_REFL:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ⟶ arcV ?u ?v ?v = (0::real)

thm Local_lemmas1.ARC_OPPOSITE:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ⟶ arcV ?u ?v (vector_sub (% (real_of_nat (2::nat)) ?u) ?v) = pi

thm Local_lemmas1.ARCV_EQ_0:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ∧ (0::real) < (?t::real) ⟶ arcV ?u ?v (vector_add ?u (% ?t (vector_sub ?v ?u))) = (0::real)

thm Local_lemmas1.ARCV_EQ_0_ORIGIN:

(?u::(real, ?'a::type) cart) ≠ vec (0::nat) ∧ (0::real) < (?t::real) ⟶ arcV (vec (0::nat)) ?u (% ?t ?u) = (0::real)

thm Local_lemmas1.ARCV_PI_OPPOSITE:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ∧ (?t::real) < (0::real) ⟶ arcV ?u ?v (vector_add ?u (% ?t (vector_sub ?v ?u))) = pi

thm Local_lemmas1.DOT_0_ARCV:

dot (vector_sub (?v::(real, ?'a::type) cart) (?u::(real, ?'a::type) cart)) (vector_sub (?w::(real, ?'a::type) cart) ?u) = (0::real) ⟶ arcV ?u ?v ?w = pi / real_of_nat (2::nat)

thm Local_lemmas1.ARCV_DEGENERATE_conjunct0:

arcV (?u::(real, ?'a::type) cart) ?u (?v::(real, ?'a::type) cart) = pi / real_of_nat (2::nat)

thm Local_lemmas1.ARCV_DEGENERATE_conjunct1:

arcV (?u::(real, ?'a::type) cart) (?v::(real, ?'a::type) cart) ?u = pi / real_of_nat (2::nat)

thm Local_lemmas1.ARCV_DEGENERATE:

arcV (?u::(real, ?'a::type) cart) ?u (?v::(real, ?'a::type) cart) = pi / real_of_nat (2::nat) ∧ arcV ?u ?v ?u = pi / real_of_nat (2::nat)

thm Local_lemmas1.ARCV_DIRECTIONS:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ⟶ (arcV ?u ?v (vector_add ?u (% (?t::real) (vector_sub ?v ?u))) = (0::real)) = ((0::real) < ?t) ∧ (arcV ?u ?v (vector_add ?u (% ?t (vector_sub ?v ?u))) = pi) = (?t < (0::real))

thm Local_lemmas1.ARCV_ORI_DIRECTIONS:

(?v::(real, ?'a::type) cart) ≠ vec (0::nat) ⟶ (arcV (vec (0::nat)) ?v (% (?t::real) ?v) = (0::real)) = ((0::real) < ?t) ∧ (arcV (vec (0::nat)) ?v (% ?t ?v) = pi) = (?t < (0::real))

thm Local_lemmas1.REAL_NEG_MUL_EQ:

(?a::real) < (0::real) ⟶ (?a * (?x::real) < (0::real)) = ((0::real) < ?x) ∧ ((0::real) < ?a * ?x) = (?x < (0::real))

thm Local_lemmas1.NOT_EQ_0:

((?x::real) ≠ (0::real)) = ((0::real) < ?x ∨ ?x < (0::real))

thm Local_lemmas1.SIN_ARCV_EQ_0:

(?u::(real, ?'a::type) cart) ≠ (?v::(real, ?'a::type) cart) ∧ (?t::real) ≠ (0::real) ⟶ sin (arcV ?u ?v (vector_add ?u (% ?t (vector_sub ?v ?u)))) = (0::real)

thm Local_lemmas1.COLLINEAR_SIN_ARCV_0:

collinear (INSERT (?u::(real, ?'a::type) cart) (INSERT (?v::(real, ?'a::type) cart) (INSERT (?x::(real, ?'a::type) cart) EMPTY))) ∧ ?u ≠ ?x ∧ ?u ≠ ?v ⟶ sin (arcV ?u ?v ?x) = (0::real)

thm DEF_normize:

normize = (λ_6553775::(real, ?'a::type) cart. % ((1::real) / vector_norm _6553775) _6553775)

thm Local_lemmas1.normize:

∀ v::(real, ?'a::type) cart. normize v = % ((1::real) / vector_norm v) v

thm Local_lemmas1.COLL_AFF_GT_2_1:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. ¬ collinear (INSERT x (INSERT v (INSERT w EMPTY))) ⟶ aff_gt (INSERT *x* (*INSERT v EMPTY*)) (*INSERT w EMPTY*) = *GSPEC* (λ*GEN%PVAR%1895*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1895* (∃ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t3* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 v*) (% *t3 w*))) *y*)

thm Local_lemmas1.COLL_EQ_DEPENDENT:

collinear (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) = (∃ (*tx*::*real*) *ty*::*real*. ¬ (*tx* = (*0*::*real*) ∧ *ty* = (*0*::*real*)) ∧ *vector_add* (% *tx ?x*) (% *ty ?y*) = *vec* (*0*::*nat*))

thm Local_lemmas1.NOT_COLL_ORTHONORMAL:

¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*?x*::(*real*, *?'a*::*type*) *cart*) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*))) ⟶ (∃ *u*::(*real*, *?'a*::*type*) *cart*. *IN u* (*aff* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?x* (*INSERT ?y EMPTY*)))) ∧ *vector_norm u* = (*1*::*real*) ∧ *dot ?x u* = (*0*::*real*))

thm Local_lemmas1.NORM1_NOT_0:

vector_norm (*?x*::(*real*, *?'a*::*type*) *cart*) = (*1*::*real*) ⟶ *?x* ≠ *vec* (*0*::*nat*)

thm Local_lemmas1.ARCV_DETER_DIRECTION:

(*?x*::(*real*, *?'a*::*type*) *cart*) = *vector_add* (% (*?tu*::*real*) (*?u*::(*real*, *?'a*::*type*) *cart*)) (% (*?ty*::*real*) (*?y*::(*real*, *?'a*::*type*) *cart*)) ∧ (*0*::*real*) < *?ty* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?u* (*INSERT ?y EMPTY*))) ∧ *vector_norm ?x* = (*1*::*real*) ∧ *vector_norm ?y* = (*1*::*real*) ∧ *arcV* (*vec* (*0*::*nat*)) *?u ?x* = *arcV* (*vec* (*0*::*nat*)) *?u ?y* ⟶ *?x* = *?y*

thm Local_lemmas1.NORM_NORMIZE:

∀ *u*::(*real*, *?'a*::*type*) *cart*. (*u* ≠ *vec* (*0*::*nat*)) = (*vector_norm* (*normize u*) = (*1*::*real*))

thm Local_lemmas1.ARCV_EQ_IMP_NORMIZE:

IN (*?x*::(*real*, *?'a*::*type*) *cart*) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*?u*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) (*INSERT* (*?y*::(*real*, *?'a*::*type*) *cart*) *EMPTY*)) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT ?u* (*INSERT ?y EMPTY*))) ∧ *arcV* (*vec* (*0*::*nat*)) *?u ?x* = *arcV* (*vec* (*0*::*nat*)) *?u ?y* ⟶ *normize ?x* = *normize ?y*

thm Local_lemmas1.AZIM_AND_ARCV_EQ_IMP_PARA:

¬ *collinear* (*INSERT* (*?v0.0*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?v*::(*real*, *3*) *cart*) *EMPTY*))) ∧ *azim ?v0.0 ?u ?v* (*?x*::(*real*, *3*) *cart*) = *azim ?v0.0 ?u ?v* (*?y*::(*real*, *3*) *cart*) ∧ *arcV ?v0.0 ?u ?x* = *arcV ?v0.0 ?u ?y* ⟶ *?y* = *?v0.0* ∨ (∃ *t*≥*0*::*real*. *vector_sub ?x ?v0.0* = % *t* (*vector_sub ?y ?v0.0*))

thm Local_lemmas1.NORM_CAUCHY_SCHWARZ_FRAC2:

$- (1::real) \leq dot\ (?u::(real,\ ?'a::type)\ cart)\ (?v::(real,\ ?'a::type)\ cart)\ /\ (vector\_norm\ ?u * vector\_norm\ ?v) \wedge dot\ ?u\ ?v\ /\ (vector\_norm\ ?u * vector\_norm\ ?v) \leq (1::real)$

thm Local_lemmas1.COS_ARCV_EQ_ARCV:

$(cos\ (arcV\ (?x::(real,\ ?'b::type)\ cart)\ (?y::(real,\ ?'b::type)\ cart)\ (?z::(real,\ ?'b::type)\ cart)) = cos\ (arcV\ (?xx::(real,\ ?'a::type)\ cart)\ (?yy::(real,\ ?'a::type)\ cart)\ (?zz::(real,\ ?'a::type)\ cart))) = (arcV\ ?x\ ?y\ ?z = arcV\ ?xx\ ?yy\ ?zz)$

thm Local_lemmas1.NORM_CAUCHY_SCHWARZ_FRAC2_conjunct1:

$dot\ (?u::(real,\ ?'a::type)\ cart)\ (?v::(real,\ ?'a::type)\ cart)\ /\ (vector\_norm\ ?u * vector\_norm\ ?v) \leq (1::real)$

thm Local_lemmas1.NORM_CAUCHY_SCHWARZ_FRAC2_conjunct0:

$- (1::real) \leq dot\ (?u::(real,\ ?'a::type)\ cart)\ (?v::(real,\ ?'a::type)\ cart)\ /\ (vector\_norm\ ?u * vector\_norm\ ?v)$

thm Local_lemmas1.ARCV_BOUNDS:

$(0::real) \leq arcV\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)\ (?z::(real,\ ?'a::type)\ cart) \wedge arcV\ ?x\ ?y\ ?z \leq pi$

thm Local_lemmas1.SIN_ARCV_EQ_0_EQ_LAP:

$(sin\ (arcV\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)\ (?z::(real,\ ?'a::type)\ cart)) = (0::real)) = (arcV\ ?x\ ?y\ ?z = (0::real) \vee arcV\ ?x\ ?y\ ?z = pi)$

thm Local_lemmas1.ORTHONORMAL_NOT_COLLINEAR:

$orthonormal\ (?e1.0::(real,\ 3)\ cart)\ (?e2.0::(real,\ 3)\ cart)\ (?e3.0::(real,\ 3)\ cart) \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ ?e3.0\ (INSERT\ ?e1.0\ EMPTY)))$

thm Local_lemmas1.LUNAR_DEFORM_INJ:

$orthonormal\ (?e1.0::(real,\ 3)\ cart)\ (?e2.0::(real,\ 3)\ cart)\ (?e3.0::(real,\ 3)\ cart) \wedge (0::real) \leq (?t::real) \wedge ?t < (1::real) \longrightarrow (\forall (x::(real,\ 3)\ cart)\ y::(real,\ 3)\ cart.\ lunar\_deform\ (?e1.0,\ ?e2.0,\ ?e3.0)\ ?t\ x = lunar\_deform\ (?e1.0,\ ?e2.0,\ ?e3.0)\ ?t\ y \longrightarrow x = y)$

thm Local_lemmas1.GRAPH_IMAGE_IMAGE:

$graph\ (?E::((real,\ ?'b::type)\ cart \Rightarrow bool) \Rightarrow bool) \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ (?f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ x = ?f\ y \longrightarrow x = y) \longrightarrow graph\ (IMAGE\ (IMAGE\ ?f)\ ?E)$

thm Local_lemmas1.LUNAR_DEFORM_ORIGIN:

$lunar\_deform\ (?e1.0::(real,\ 3)\ cart,\ ?e2.0::(real,\ 3)\ cart,\ ?e3.0::(real,\ 3)\ cart)\ (?t::real)\ (vec\ (0::nat)) = vec\ (0::nat)$

thm Local_lemmas1.orthonormal1:

∀(e1::(real, 3) cart) (e2::(real, 3) cart) e3::(real, 3) cart. orthonormal e1 e2 e3 = (vector_norm e1 = (1::real) ∧ vector_norm e2 = (1::real) ∧ vector_norm e3 = (1::real) ∧ dot e1 e2 = (0::real) ∧ dot e1 e3 = (0::real) ∧ dot e2 e3 = (0::real) ∧ (0::real) < dot (cross e1 e2) e3)

thm Local_lemmas1.ORTHO_SPHERICAL_AZIM:

orthonormal (?e1.0::(real, 3) cart) (?e2.0::(real, 3) cart) (?e3.0::(real, 3) cart) ∧ (?x::(real, 3) cart) = vector_add (% ((?r::real) ∗ cos (?theta::real)) ?e1.0) (vector_add (% (?r ∗ sin ?theta) ?e2.0) (% (?h::real) ?e3.0)) ∧ (0::real) < ?r ∧ (0::real) ≤ ?theta ∧ ?theta < real_of_nat (2::nat) ∗ pi ⟶ azim (vec (0::nat)) ?e3.0 ?e1.0 ?x = ?theta

thm Local_lemmas1.COS_ARCV:

cos (arcV (?v0.0::(real, ?'a::type) cart) (?u::(real, ?'a::type) cart) (?w::(real, ?'a::type) cart)) = dot (vector_sub ?u ?v0.0) (vector_sub ?w ?v0.0) / (vector_norm (vector_sub ?u ?v0.0) ∗ vector_norm (vector_sub ?w ?v0.0))

thm Local_lemmas1.LUNAR_DEFORM_ARCV_PRESERVED:

orthonormal (?e1.0::(real, 3) cart) (?e2.0::(real, 3) cart) (?e3.0::(real, 3) cart) ⟶ arcV (vec (0::nat)) ?e3.0 (?x::(real, 3) cart) = arcV (vec (0::nat)) ?e3.0 (lunar_deform (?e1.0, ?e2.0, ?e3.0) (?t::real) ?x)

thm Local_lemmas1.AFF_CONV0_INTERSECTION_LEMMA:

∀(u::(real, 3) cart) (x::(real, 3) cart) (y::(real, 3) cart) z::(real, 3) cart. coplanar (INSERT u (INSERT x (INSERT y (INSERT z EMPTY)))) ∧ ¬ collinear (INSERT u (INSERT x (INSERT y EMPTY))) ∧ ¬ collinear (INSERT u (INSERT y (INSERT z EMPTY))) ∧ ¬ collinear (INSERT u (INSERT z (INSERT x EMPTY))) ⟶ ¬ (HOL_Light_Import.INTER (aff (INSERT u (INSERT x EMPTY))) (conv0 (INSERT y (INSERT z EMPTY))) = EMPTY ∧ HOL_Light_Import.INTER (aff (INSERT u (INSERT y EMPTY))) (conv0 (INSERT x (INSERT z EMPTY))) = EMPTY ∧ HOL_Light_Import.INTER (aff (INSERT u (INSERT z EMPTY))) (conv0 (INSERT x (INSERT y EMPTY))) = EMPTY)

thm Local_lemmas1.CVLF_COLLINEAR_CIRCULAR_LUNAR:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ SUBSET (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY)) ?V ∧ collinear (INSERT (vec (0::nat)) (INSERT ?v (INSERT ?w EMPTY))) ∧ ?v ≠ ?w ⟶ circular ?V ?E ∨ lunar (?v, ?w) ?V ?E

thm Local_lemmas1.CVLF_FLAT_ANGLE_LEMMA:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ SUBSET (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY)) ?V ∧ collinear (INSERT (vec (0::nat)) (INSERT ?v (INSERT ?w EMPTY))) ∧ ?v ≠ ?w ⟶ (∀z::(real, 3)

$cart.\ IN\ z\ ?V \land z \neq ?v \land z \neq ?w \longrightarrow interior\_angle1\ (vec\ (0::nat))\ ?FF\ z = pi)$

thm DEF_deformation:

$deformation = (\lambda(\_6557255::(real, ?'a::type)\ cart \Rightarrow real \Rightarrow (real, ?'a::type)\ cart)\ (\_6557256::(real, ?'a::type)\ cart \Rightarrow bool)\ \_6557257::real \times real.\ IN\ (0::real)\ (open\_real\_interval\ (fst\ \_6557257,\ snd\ \_6557257)) \land (\forall (v::(real, ?'a::type)\ cart)\ r::real.\ IN\ v\ \_6557256 \land IN\ r\ (open\_real\_interval\ (fst\ \_6557257,\ snd\ \_6557257)) \longrightarrow continuous\ (\_6557255\ v)\ (atreal\ r)) \land (\forall v::(real, ?'a::type)\ cart.\ IN\ v\ \_6557256 \longrightarrow \_6557255\ v\ (0::real) = v))$

thm Local_lemmas1.deformation:

$\forall (a::real)\ (b::real)\ (V::(real, ?'a::type)\ cart \Rightarrow bool)\ ff::(real, ?'a::type)\ cart \Rightarrow real \Rightarrow (real, ?'a::type)\ cart.\ deformation\ ff\ V\ (a,\ b) = (IN\ (0::real)\ (open\_real\_interval\ (a,\ b)) \land (\forall (v::(real, ?'a::type)\ cart)\ r::real.\ IN\ v\ V \land IN\ r\ (open\_real\_interval\ (a,\ b)) \longrightarrow continuous\ (ff\ v)\ (atreal\ r)) \land (\forall v::(real, ?'a::type)\ cart.\ IN\ v\ V \longrightarrow ff\ v\ (0::real) = v))$

thm Local_lemmas1.TOW_REAL_EXISTS_COMBINED:

$(\exists e1>0::real.\ \forall t::real.\ - e1 < t \land t < e1 \longrightarrow (?P::real \Rightarrow bool)\ t) \land (\exists e1>0::real.\ \forall t::real.\ - e1 < t \land t < e1 \longrightarrow (?Q::real \Rightarrow bool)\ t) \longrightarrow (\exists e1>0::real.\ \forall t::real.\ - e1 < t \land t < e1 \longrightarrow ?P\ t \land ?Q\ t)$

thm Local_lemmas1.CONTINUOUS_FUNS_DISTINCT_POINTS:

$continuous\ (?f::real \Rightarrow (real, ?'a::type)\ cart)\ (atreal\ (?r::real)) \land continuous\ (?g::real \Rightarrow (real, ?'a::type)\ cart)\ (atreal\ (?rr::real)) \land ?f\ ?r \neq ?g\ ?rr \longrightarrow (\exists d>0::real.\ \forall (x::real)\ y::real.\ |x - ?r| < d \land |y - ?rr| < d \longrightarrow ?f\ x \neq ?g\ y)$

thm Local_lemmas1.CONTINUOUS_TWO_POINTS_DISTINCT:

$continuous\ ((?ff::(real, ?'a::type)\ cart \Rightarrow real \Rightarrow (real, ?'b::type)\ cart)\ (?v1.0::(real, ?'a::type)\ cart))\ (atreal\ (?r::real)) \land continuous\ (?ff\ (?v2.0::(real, ?'a::type)\ cart))\ (atreal\ ?r) \land ?ff\ ?v1.0\ ?r \neq ?ff\ ?v2.0\ ?r \longrightarrow (\exists d>0::real.\ \forall t::real.\ |t - ?r| < d \longrightarrow ?ff\ ?v1.0\ t \neq ?ff\ ?v2.0\ t)$

thm Local_lemmas1.CONTINUOUS_FUN_DISTINCT_FINITE_SET:

$\forall V::(real, ?'b::type)\ cart \Rightarrow bool.\ FINITE\ V \longrightarrow (\forall v1::(real, ?'b::type)\ cart.\ IN\ v1\ V \land (?v0.0::(real, ?'b::type)\ cart) \neq v1 \longrightarrow (?ff::(real, ?'b::type)\ cart \Rightarrow real \Rightarrow (real, ?'a::type)\ cart)\ ?v0.0\ (?r::real) \neq ?ff\ v1\ ?r) \land (\forall v::(real, ?'b::type)\ cart.\ IN\ v\ (INSERT\ ?v0.0\ V) \longrightarrow continuous\ (?ff\ v)\ (atreal\ ?r)) \longrightarrow (\exists d>0::real.\ \forall t::real.\ |t - ?r| < d \longrightarrow (\forall v1::(real, ?'b::type)\ cart.\ IN\ v1\ V \land ?v0.0 \neq v1 \longrightarrow ?ff\ ?v0.0\ t \neq ?ff\ v1\ t))$

thm Local_lemmas1.CONTINUOUS_ATREAL_INJ_PRESERVED:

$\forall V::(real, ?'b::type)\ cart \Rightarrow bool.\ FINITE\ V \longrightarrow (\forall (v1::(real, ?'b::type)\ cart)\ v2::(real, ?'b::type)\ cart.\ IN\ v1\ V \land IN\ v2\ V \land v1 \neq v2 \longrightarrow (?ff::(real,$

?'b::type) cart ⇒ real ⇒ (real, ?'a::type) cart) v1 (?r::real) ≠ ?ff v2 ?r) ∧ (∀ v::(real, ?'b::type) cart. IN v V ⟶ continuous (?ff v) (atreal ?r)) ⟶ (∃ d>0::real. ∀ t::real. |t − ?r| < d ⟶ (∀ (v1::(real, ?'b::type) cart) v2::(real, ?'b::type) cart. IN v1 V ∧ IN v2 V ∧ v1 ≠ v2 ⟶ ?ff v1 t ≠ ?ff v2 t))

thm Local_lemmas1.CONTINUOUS_ATREAL_DISTINCT:

continuous ((?ff::(real, ?'a::type) cart ⇒ real ⇒ (real, ?'b::type) cart) (?v::(real, ?'a::type) cart)) (atreal (?r::real)) ∧ ?ff ?v ?r ≠ (?v0.0::(real, ?'b::type) cart) ⟶ (∃ d>0::real. ∀ t::real. |t − ?r| < d ⟶ ?ff ?v t ≠ ?v0.0)

thm Local_lemmas1.CONTINUOUS_ATREAL_DISTINCT_FINITE:

∀ V::(real, ?'b::type) cart ⇒ bool. FINITE V ⟶ (∀ v1::(real, ?'b::type) cart. IN v1 V ⟶ (?ff::(real, ?'b::type) cart ⇒ real ⇒ (real, ?'a::type) cart) v1 (?r::real) ≠ (?v0.0::(real, ?'a::type) cart)) ∧ (∀ v::(real, ?'b::type) cart. IN v V ⟶ continuous (?ff v) (atreal ?r)) ⟶ (∃ d>0::real. ∀ t::real. |t − ?r| < d ⟶ (∀ v1::(real, ?'b::type) cart. IN v1 V ⟶ ?ff v1 t ≠ ?v0.0))

thm Local_lemmas1.REAL_CONTINUOUS_SUM_FUNS:

real_continuous (?f::(real, ?'a::type) cart ⇒ real) (at (?x::(real, ?'a::type) cart)) ∧ real_continuous (?g::(real, ?'a::type) cart ⇒ real) (at ?x) ⟶ real_continuous (λx::(real, ?'a::type) cart. ?f x + ?g x) (at ?x)

thm Local_lemmas1.REAL_CON_IMP_OPP_FUN_TOO:

real_continuous (?f::(real, ?'a::type) cart ⇒ real) (at (?x::(real, ?'a::type) cart)) ⟶ real_continuous (λx::(real, ?'a::type) cart. − ?f x) (at ?x)

thm Local_lemmas1.REAL_CONTINUOUS_SUB_FUNS:

real_continuous (?f::(real, ?'a::type) cart ⇒ real) (at (?x::(real, ?'a::type) cart)) ∧ real_continuous (?g::(real, ?'a::type) cart ⇒ real) (at ?x) ⟶ real_continuous (λx::(real, ?'a::type) cart. ?f x − ?g x) (at ?x)

thm Local_lemmas1.INV_INEQUAL_GENERAL:

(0::real) < (?e::real) ∧ (0::real) < (?x::real) ∧ ?x < (?y::real) ⟶ ?e / ?y < ?e / ?x

thm Local_lemmas1.REAL_CONTINUOUS_IMP_MUL_FUN:

real_continuous (?f::(real, ?'a::type) cart ⇒ real) (at (?x::(real, ?'a::type) cart)) ∧ real_continuous (?g::(real, ?'a::type) cart ⇒ real) (at ?x) ⟶ real_continuous (λx::(real, ?'a::type) cart. ?f x ∗ ?g x) (at ?x)

thm Local_lemmas1.CON_ATREAL_REAL_CON:

∀ (f::real ⇒ (real, ?'a::type) cart) v0::(real, ?'a::type) cart. continuous f (atreal t0) ⟶ real_continuous (λt::real. distance (f t, v0)) (atreal t0)

thm Local_lemmas1.CON_ATREAL_REAL_CON2:

$\forall$ (*f::real* $\Rightarrow$ (*real*, *?'b::type*) *cart*) *v0::*(*real*, *?'a::type*) *cart*. *continuous f* (*atreal t0*) $\wedge$ *continuous* (*?g::real* $\Rightarrow$ (*real*, *?'b::type*) *cart*) (*atreal t0*) $\longrightarrow$ *real_continuous* (λ*t::real*. *distance* (*f t*, *?g t*)) (*atreal t0*)

thm DEF_localization:

localization = (λ(*_6558114::*(*?'a::type* $\Rightarrow$ *bool*) $\times$ ((*?'a::type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*)) *_6558115::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*. (*v_prime* (*fst _6558114*) *_6558115*, *e_prime* (*snd _6558114*) *_6558115*))

thm Local_lemmas1.localization:

$\forall$ (*V::?'a::type* $\Rightarrow$ *bool*) (*E::*(*?'a::type* $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *FF::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *bool*. *localization* (*V*, *E*) *FF* = (*v_prime V FF*, *e_prime E FF*)

thm DEF_v_slice:

v_slice = (λ(*_6558131::?'c::type*) *_6558132::?'b::type* $\times$ *?'a::type*. *GSPEC* (λ*GEN%PVAR%1896::?'a::type*. $\exists$ (*i::nat*) (*f::?'a::type* $\Rightarrow$ *?'a::type*) *v::?'a::type*. *SETSPEC GEN%PVAR%1896* ($\forall$ *j<i*. *ITER j f v* $\neq$ *snd _6558132*) (*ITER i f v*)))

thm Local_lemmas1.v_slice:

$\forall$ (*f::?'c::type*) (*v::?'b::type*) *w::?'a::type*. *v_slice f* (*v*, *w*) = *GSPEC* (λ*GEN%PVAR%1896::?'a::type*. $\exists$ (*i::nat*) (*f::?'a::type* $\Rightarrow$ *?'a::type*) *v::?'a::type*. *SETSPEC GEN%PVAR%1896* ($\forall$ *j<i*. *ITER j f v* $\neq$ *w*) (*ITER i f v*))

thm DEF_e_slice:

e_slice = (λ(*_6558148::?'b::type*) *_6558149::?'a::type* $\times$ *?'a::type*. *INSERT* (*INSERT* (*snd _6558149*) (*INSERT* (*fst _6558149*) *EMPTY*)) (*GSPEC* (λ*GEN%PVAR%1897::?'a::type* $\Rightarrow$ *bool*. $\exists$ (*i::nat*) (*f::?'a::type* $\Rightarrow$ *?'a::type*) *v::?'a::type*. *SETSPEC GEN%PVAR%1897* ($\forall$ *j<i* + (*1::nat*). *ITER j f v* $\neq$ *snd _6558149*) (*INSERT* (*ITER i f v*) (*INSERT* (*ITER* (*i* + (*1::nat*)) *f v*) *EMPTY*)))))

thm Local_lemmas1.e_slice:

$\forall$ (*f::?'b::type*) (*v::?'a::type*) *w::?'a::type*. *e_slice f* (*v*, *w*) = *INSERT* (*INSERT w* (*INSERT v EMPTY*)) (*GSPEC* (λ*GEN%PVAR%1897::?'a::type* $\Rightarrow$ *bool*. $\exists$ (*i::nat*) (*f::?'a::type* $\Rightarrow$ *?'a::type*) *v::?'a::type*. *SETSPEC GEN%PVAR%1897* ($\forall$ *j<i* + (*1::nat*). *ITER j f v* $\neq$ *w*) (*INSERT* (*ITER i f v*) (*INSERT* (*ITER* (*i* + (*1::nat*)) *f v*) *EMPTY*))))

thm DEF_f_slice:

f_slice = (λ(*_6558165::?'b::type*) *_6558166::?'a::type* $\times$ *?'a::type*. *INSERT* (*snd _6558166*, *fst _6558166*) (*GSPEC* (λ*GEN%PVAR%1898::?'a::type* $\times$ *?'a::type*. $\exists$ (*i::nat*) (*f::?'a::type* $\Rightarrow$ *?'a::type*) *v::?'a::type*. *SETSPEC GEN%PVAR%1898* ($\forall$ *j<i* + (*1::nat*). *ITER j f v* $\neq$ *snd _6558166*) (*ITER i f v*, *ITER* (*i* + (*1::nat*)) *f v*))))

thm Local_lemmas1.f_slice:

$\forall$ (*f::?'b::type*) (*v::?'a::type*) *w::?'a::type*. *f_slice f* (*v*, *w*) = *INSERT* (*w*, *v*) (*GSPEC* (λ*GEN%PVAR%1898::?'a::type* $\times$ *?'a::type*. $\exists$ (*i::nat*) (*f::?'a::type*

⇒ ?'a::type) v::?'a::type. SETSPEC GEN%PVAR%1898 (∀j<i + (1::nat). ITER j f v ≠ w) (ITER i f v, ITER (i + (1::nat)) f v)))

thm Local_lemmas1.SET2_HAS_SIZE2:

HAS_SIZE (INSERT (?a::?'a::type) (INSERT (?b::?'a::type) EMPTY)) (2::nat) = (?a ≠ ?b)

thm Local_lemmas1.DIJ_AFF_GE_PARTITION:

DISJOINT (INSERT (?u::(real, ?'a::type) cart) (INSERT (?v::(real, ?'a::type) cart) EMPTY)) (INSERT (?w::(real, ?'a::type) cart) EMPTY) ⟶ aff_ge (INSERT ?u (INSERT ?v EMPTY)) (INSERT ?w EMPTY) = HOL_Light_Import.UNION (aff (INSERT ?u (INSERT ?v EMPTY))) (aff_gt (INSERT ?u (INSERT ?v EMPTY)) (INSERT ?w EMPTY))

thm Local_lemmas1.AFF_GE_WEDGE_DISJOINTION:

HOL_Light_Import.INTER (aff_ge (INSERT (?v0.0::(real, 3) cart) (INSERT (?v1.0::(real, 3) cart) EMPTY)) (INSERT (?w1.0::(real, 3) cart) EMPTY)) (wedge ?v0.0 ?v1.0 ?w1.0 (?w2.0::(real, 3) cart)) = EMPTY ∧ HOL_Light_Import.INTER (aff_ge (INSERT ?v0.0 (INSERT ?v1.0 EMPTY)) (INSERT ?w2.0 EMPTY)) (wedge ?v0.0 ?v1.0 ?w1.0 ?w2.0) = EMPTY

thm Local_lemmas1.HAS_SIZE_2_EXISTS2:

HAS_SIZE (?S::?'a::type ⇒ bool) (2::nat) = (∃(x::?'a::type) y::?'a::type. x ≠ y ∧ ?S = INSERT x (INSERT y EMPTY))

thm Local_lemmas1.FAN_E_SUB_V:

FAN (vec (0::nat), ?V::(real, ?'a::type) cart ⇒ bool, ?E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) ∧ IN (INSERT (?x::(real, ?'a::type) cart) (INSERT (?y::(real, ?'a::type) cart) EMPTY)) ?E ⟶ IN ?x ?V ∧ IN ?y ?V

thm Local_lemmas1.LOCAL_E_SUB_V:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (INSERT (?x::(real, 3) cart) (INSERT (?y::(real, 3) cart) EMPTY)) ?E ⟶ IN ?x ?V ∧ IN ?y ?V

thm Local_lemmas1.EDGE_NOT_INTER_WITH_WEDGE:

HOL_Light_Import.INTER (aff (INSERT (?v0.0::(real, 3) cart) (INSERT (?v1.0::(real, 3) cart) EMPTY))) (wedge ?v0.0 ?v1.0 (?w1.0::(real, 3) cart) (?w2.0::(real, 3) cart)) = EMPTY

thm Local_lemmas1.AFF_GE11_SUB_AFF2:

SUBSET (aff_ge (INSERT (?v0.0::(real, ?'a::type) cart) EMPTY) (INSERT (?v1.0::(real, ?'a::type) cart) EMPTY)) (aff (INSERT ?v0.0 (INSERT ?v1.0 EMPTY)))

thm Local_lemmas1.PROVE_SLICING_FAN:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*)) ⟶ *FAN* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))

thm Local_lemmas1.FACE_MAP_ADD_SET2_EQ:

IN (*?x*::(*real*, *3*) *cart*, *?y*::(*real*, *3*) *cart*) (*darts_of_hyp* (*HOL_Light_Import.UNION* (*?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*INSERT* (*INSERT* (*?a*::(*real*, *3*) *cart*) (*INSERT* (*?b*::(*real*, *3*) *cart*) *EMPTY*)) *EMPTY*)) (*?V*::(*real*, *3*) *cart* ⇒ *bool*)) ∧ *?y* ≠ *?a* ∧ *?y* ≠ *?b* ∧ *FAN* (*vec* (*0*::*nat*), *?V*, *?E*) ∧ *FAN* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION ?E* (*INSERT* (*INSERT ?a* (*INSERT ?b EMPTY*)) *EMPTY*)) ⟶ *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION ?E* (*INSERT* (*INSERT ?a* (*INSERT ?b EMPTY*)) *EMPTY*)))) (*?x*, *?y*) = *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) (*?x*, *?y*)

thm Local_lemmas1.LOCAL_FACE_MAP_RHO_NODE1:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart*, *?y*::(*real*, *3*) *cart*) *?FF* ⟶ *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*))) (*?x*, *?y*) = (*rho_node1 ?FF ?x*, *rho_node1 ?FF ?y*)

thm Local_lemmas1.IN_DARTS_EXTENSION:

IN (*INSERT* (*?x*::*?'a*::*type*) (*INSERT* (*?y*::*?'a*::*type*) *EMPTY*)) (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) ⟶ *IN* (*?x*, *?y*) (*darts_of_hyp* (*HOL_Light_Import.UNION ?E* (*?S*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) (*?V*::*?'a*::*type* ⇒ *bool*))

thm Local_lemmas1.LOCAL_RHO_NODE_PAIR_E:

∀ *v*::(*real*, *3*) *cart*. *local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN v ?V* ⟶ *IN* (*INSERT v* (*INSERT* (*rho_node1 ?FF v*) *EMPTY*)) *?E*

thm Local_lemmas1.LOFA_HYP_UNION_CARD_GT2:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION* *?E* (*INSERT* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* *?HS* (*?v*, *rho_node1* *?FF* *?v*) ⟶ (*2*::*nat*) < *CARD* *?fv*

thm Local_lemmas1.LOCAL_FAN_SIMPLE_HYP:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ⟶ *simple_hypermap* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *?E*)))

thm Local_lemmas1.EE_UNION:

EE (*?v*::*?'a*::*type*) (*HOL_Light_Import.UNION* (*?E*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*) (*?S*::(*?'a*::*type* ⇒ *bool*) ⇒ *bool*)) = *HOL_Light_Import.UNION* (*EE* *?v* *?E*) (*EE* *?v* *?S*)

thm Local_lemmas1.EE_SING_SING:

EE (*?v*::*?'a*::*type*) (*INSERT* (*INSERT* *?v* (*INSERT* (*?w*::*?'a*::*type*) *EMPTY*)) *EMPTY*) = *INSERT* *?w* *EMPTY*

thm Local_lemmas1.CROSS_PAIR_NOT_IN_FF:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN* *x* *?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* *?v* (*INSERT* *?w* *EMPTY*))) (*wedge_in_fan_gt* *x* *?E*)) ⟶ ¬ *IN* (*?v*, *?w*) *?FF*

thm Local_lemmas1.AZ_REFL11:

∀ (*w1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *azim* (*?v*::(*real*, *3*) *cart*) *?v* *w1* *w2* = (*0*::*real*)

thm Local_lemmas1.AZIM_POS_IMP_CYCLIC_SET:

(*0*::*real*) < *azim* (*?v0.0*::(*real*, *3*) *cart*) (*?v1.0*::(*real*, *3*) *cart*) (*?w1.0*::(*real*, *3*) *cart*) (*?w2.0*::(*real*, *3*) *cart*) ⟶ *cyclic_set* (*INSERT* *?w1.0* (*INSERT* *?w2.0* *EMPTY*)) *?v0.0* *?v1.0*

thm Local_lemmas1.AZIM_POS_IMP_SUM_2PI:

(*0*::*real*) < *azim* (*?a*::(*real*, *3*) *cart*) (*?b*::(*real*, *3*) *cart*) (*?c*::(*real*, *3*) *cart*) (*?d*::(*real*, *3*) *cart*) ⟶ *azim* *?a* *?b* *?c* *?d* + *azim* *?a* *?b* *?d* *?c* = *real_of_nat* (*2*::*nat*) ∗ *pi*

thm Local_lemmas1.FACE_MAP_AT_TURNING_DART:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN* *z* (*INSERT* *?v* (*INSERT* *?w* *EMPTY*)) ∧ *IN* *t* (*DIFF* *?V* (*INSERT* *z* *EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *z* (*INSERT* *t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ *IN* (*?x*::(*real*, *3*) *cart*, *?v*) *?FF* ⟶ *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*)))) (*?x*, *?v*) = (*?v*, *?w*)

thm Local_lemmas1.WEDGE_IN_FAN_LOFA_DETER:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *rho_node1 ?FF ?v* = (*?w*::(*real*, *3*) *cart*) ⟶ *wedge_in_fan_gt* (*?w*, *rho_node1 ?FF ?w*) *?E* = *wedge* (*vec* (*0*::*nat*)) *?w* (*rho_node1 ?FF ?w*) *?v*

thm Local_lemmas1.FACE_MAP_SLICING_HYP_TRANS_POINT:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*)) ⟶ *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*)))) (*?v*, *?w*) = (*?w*, *rho_node1 ?FF ?w*)

thm Local_lemmas1.FACE_MAP_AT_TURNING_DART1:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ *IN* (*?x*::(*real*, *3*) *cart*, *?w*) *?FF* ⟶ *face_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*)))) (*?x*, *?w*) = (*?w*, *?v*)

thm Local_lemmas1.LOCAL_FAN_ORBIT_MAP_VITERFF:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ (∀ *n*::*nat*. *IN* (*ITER n* (*rho_node1 ?FF*) *?v*, *ITER* (*n* + (*1*::*nat*)) (*rho_node1 ?FF*) *?v*) *?FF*)

thm Local_lemmas1.DETERMINE_FV:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V*

∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ⟶ *?fv* = *INSERT* (*?w*, *?v*) (*GSPEC* (λ*GEN%PVAR%1909*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *n*::*nat*. *SETSPEC GEN%PVAR%1909* (∀ *m*<*n* + (*1*::*nat*). *ITER m* (*rho_node1 ?FF*) *?v* ≠ *?w*) (*ITER n* (*rho_node1 ?FF*) *?v*, *ITER* (*n* + (*1*::*nat*)) (*rho_node1 ?FF*) *?v*)))

thm Local_lemmas1.TWO_EQ_SYSTEM_THM:

FAN (*vec* (*0*::*nat*), *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ *graph ?E* ∧ *SUBSET* (*UNIONS ?E*) *?V* ⟶ (*dart_of_fan* (*?V*, *?E*), *res* (*e_fan_pair* (*?V*, *?E*)) (*dart1_of_fan* (*?V*, *?E*)), *res* (*n_fan_pair* (*?V*, *?E*)) (*dart1_of_fan* (*?V*, *?E*)), *res* (*f_fan_pair* (*?V*, *?E*)) (*dart1_of_fan* (*?V*, *?E*))) = (*darts_of_hyp ?E ?V*, *ee_of_hyp* (*vec* (*0*::*nat*), *?V*, *?E*), *nn_of_hyp* (*vec* (*0*::*nat*), *?V*, *?E*), *ff_of_hyp* (*vec* (*0*::*nat*), *?V*, *?E*))

thm Local_lemmas1.LOFA_FST_IDENTIFY:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ∧ *IN* (*?y*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) *?FF* ∧ *fst ?x* = *fst ?y* ⟶ *?x* = *?y*

thm Local_lemmas1.CARD_IS_LEAST_CYCLE:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *ITER* (*?n*::*nat*) (*rho_node1 ?FF*) *?v* = *?v* ∧ *?n* ≠ (*0*::*nat*) ⟶ *CARD ?V* ≤ *?n*

thm Local_lemmas1.HAFL_CIRCLE_FORM_LOCAL_FAN:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import.UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ⟶

local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv)

thm Local_lemmas1.HAFL_CIRCLE_FORM_LOCAL_FAN2:

(local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀ (z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ∧ (?fw::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?w, rho_node1 ?FF ?w) ⟶ local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv) ∧ local_fan (v_prime ?V ?fw, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?w (INSERT ?v EMPTY)) EMPTY)) ?fw, ?fw)

thm Local_lemmas1.LOCAL_FAN_RHO_NODE_IVS:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ rho_node1 ?FF (ivs_rho_node1 ?FF ?v) = ?v

thm Local_lemmas1.LOCAL_FAN_IVS_IN_V:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ IN (ivs_rho_node1 ?FF ?v) ?V

thm Local_lemmas1.LF_AZIM_CYCLE_EQ_IVS_ND:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ azim_cycle (EE ?v ?E) (vec (0::nat)) ?v (rho_node1 ?FF ?v) = ivs_rho_node1 ?FF ?v

thm Local_lemmas1.AZIM_IN_FAN_RHOND_IVS_RHOND:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ azim_in_fan (?v, rho_node1 ?FF ?v) ?E = azim (vec (0::nat)) ?v (rho_node1 ?FF ?v) (ivs_rho_node1 ?FF ?v)

thm Local_lemmas1.LOFA_IMP_EE_TWO_ELMS_INS_ND:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶

EE ?v ?E = INSERT (rho_node1 ?FF ?v) (INSERT (ivs_rho_node1 ?FF ?v) EMPTY)

thm Local_lemmas1.WEDGE_IN_FAN_RHOND_IVS_RHOND:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ wedge_in_fan_ge (?v, rho_node1 ?FF ?v) ?E = wedge_ge (vec (0::nat)) ?v (rho_node1 ?FF ?v) (ivs_rho_node1 ?FF ?v)

thm Local_lemmas1.FST_LST_IN_WEDGE_GE:

IN (?w1.0::(real, 3) cart) (wedge_ge (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) ?w1.0 (?w2.0::(real, 3) cart)) ∧ IN ?w2.0 (wedge_ge ?v0.0 ?v1.0 ?w1.0 ?w2.0)

thm Local_lemmas1.IVS_RHO_NODE_DIFF_ID:

∀ v::(real, 3) cart. local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN v ?V ⟶ ivs_rho_node1 ?FF v ≠ v

thm Local_lemmas1.POINT_PRESENTED_IN_RHOND1:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ⟶ (∃ n<CARD ?V. ITER n (rho_node1 ?FF) ?v = ?w ∧ (∀ m<n. ITER m (rho_node1 ?FF) ?v ≠ ?w))

thm Local_lemmas1.POINTS_IN_HAFL_CIRCLE:

(local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀ (z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ⟶ v_prime ?V ?fv = GSPEC (λGEN%PVAR%1922::(real, 3) cart. ∃ n::nat. SETSPEC GEN%PVAR%1922 (∀ m<n. ITER m (rho_node1 ?FF) ?v ≠ ?w) (ITER n (rho_node1 ?FF) ?v))

thm Local_lemmas1.FST_LST_IN_WEDGE_GE_conjunct1:

IN (?w2.0::(real, 3) cart) (wedge_ge (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) ?w2.0)

thm Local_lemmas1.FST_LST_IN_WEDGE_GE_conjunct0:

IN (?w1.0::(real, 3) cart) (wedge_ge (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) ?w1.0 (?w2.0::(real, 3) cart))

thm Local_lemmas1.COVEX_OF_LOFA_HALF_CIRCLE:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ (local_fan (?V, ?E, ?FF) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀(z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ⟶ convex_local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv)

thm Local_lemmas1.COVEX_OF_LOFA_HALF_CIRCLE2:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ (local_fan (?V, ?E, ?FF) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀(z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ∧ (?fw::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?w, rho_node1 ?FF ?w) ⟶ convex_local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv) ∧ convex_local_fan (v_prime ?V ?fw, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?w (INSERT ?v EMPTY)) EMPTY)) ?fw, ?fw)

thm Local_lemmas1.CARD_V_TWO_HAFL_CIRCLE:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ ITER (?n::nat) (rho_node1 ?FF) ?v = ?w ∧ ITER (?n′::nat) (rho_node1 ?FF) ?w = ?v ∧ ?n < CARD ?V ∧ ?n′ < CARD ?V ⟶ ?n + ?n′ = CARD ?V

thm Local_lemmas1.DIFFERENCE_IMP_LT_CARDV:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN

(*?w*::(*real*, *3*) *cart*) *?V* ∧ (∀ *n*<*?m*::*nat*. *ITER n* (*rho_node1 ?FF*) *?v* ≠ *?w*) ⟶ *?m* < *CARD ?V*

thm Local_lemmas1.LT_CARD_MONO_LOFA:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ (∀ (*i*::*nat*) *j*::*nat*. *i* < *CARD ?V* ∧ *j* < *CARD ?V* ∧ *ITER i* (*rho_node1 ?FF*) *?v* = *ITER j* (*rho_node1 ?FF*) *?v* ⟶ *i* = *j*)

thm Local_lemmas1.CONDS_IN_V_PRIME_NUM:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ⟶ (*?n*::*nat*) < *CARD ?V* ∧ *ITER ?n* (*rho_node1 ?FF*) *?v* = *?w* ⟶ (∀ *i*::*nat*. (*i* < *CARD ?V* ∧ *IN* (*ITER i* (*rho_node1 ?FF*) *?v*) (*v_prime ?V ?fv*)) = (*i* < *?n* + (*1*::*nat*)))

thm Local_lemmas1.LOFA_IMP_ITER_RHO_NODE_ID2:

local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ⟶ *ITER* (*CARD ?V*) (*rho_node1 ?FF*) *?v* = *?v*

thm Local_lemmas1.CONDS_IN_V_PRIME_NUM2:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀ (*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fw*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?w*, *rho_node1 ?FF ?w*) ⟶ (*?n*::*nat*) < *CARD ?V* ∧ *ITER ?n* (*rho_node1 ?FF*) *?v* = *?w* ⟶ (∀ *i*::*nat*. (*i* < *CARD ?V* ∧ *IN* (*ITER i* (*rho_node1 ?FF*) *?v*) (*v_prime ?V ?fw*)) = (*i* = (*0*::*nat*) ∨ *?n* ≤ *i* ∧ *i* < *CARD ?V*))

thm Local_lemmas1.DETERMINE_FV2:

(*local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀(*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ⟶ (*?n*::*nat*) < *CARD ?V* ∧ *ITER ?n* (*rho_node1 ?FF*) *?v* = *?w* ⟶ *?fv* = *INSERT* (*?w*, *?v*) (*GSPEC* (λ*GEN%PVAR%1928*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃*m*::*nat*. *SETSPEC GEN%PVAR%1928* (*m* < *?n*) (*ITER m* (*rho_node1 ?FF*) *?v*, *ITER* (*m* + (*1*::*nat*)) (*rho_node1 ?FF*) *?v*)))

thm Local_lemmas1.INTERIOR_ANGLE_LEM_SLICING_FAN:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ (*local_fan* (*?V*, *?E*, *?FF*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀(*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ∧ (*?fw*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?w*, *rho_node1 ?FF ?w*) ⟶ *interior_angle1* (*vec* (*0*::*nat*)) *?fv ?v* + *interior_angle1* (*vec* (*0*::*nat*)) *?fw ?v* = *interior_angle1* (*vec* (*0*::*nat*)) *?FF ?v*

thm Local_lemmas1.INTERIOR_ANGLE_LEM_SLICING_FAN2:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ (*local_fan* (*?V*, *?E*, *?FF*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀(*z*::(*real*, *3*) *cart*) *t*::(*real*, *3*) *cart*. *IN z* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN t* (*DIFF ?V* (*INSERT z EMPTY*)) ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT z* (*INSERT t EMPTY*)))) ∧ (∀*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ∧ (*?fw*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?w*, *rho_node1 ?FF ?w*) ⟶ *interior_angle1* (*vec* (*0*::*nat*)) *?fv ?v* + *interior_angle1* (*vec*

(0::nat)) ?fw ?v = interior_angle1 (vec (0::nat)) ?FF ?v ∧ interior_angle1 (vec (0::nat)) ?fw ?w + interior_angle1 (vec (0::nat)) ?fv ?w = interior_angle1 (vec (0::nat)) ?FF ?w

thm Local_lemmas1.INTERIOR_AGL_EQ:

(local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀(z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ⟶ (?n::nat) < CARD ?V ∧ ITER ?n (rho_node1 ?FF) ?v = ?w ⟶ (∀i::nat. (0::nat) < i ∧ i < ?n ⟶ interior_angle1 (vec (0::nat)) ?fv (ITER i (rho_node1 ?FF) ?v) = interior_angle1 (vec (0::nat)) ?FF (ITER i (rho_node1 ?FF) ?v))

thm Local_lemmas1.SUM_INTERIOR_AGL_LEMMA:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ (local_fan (?V, ?E, ?FF) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀(z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ∧ (?fw::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?w, rho_node1 ?FF ?w) ⟶ (∀ff::nat ⇒ real. sum (GSPEC (λGEN%PVAR%1957::nat. ∃i::nat. SETSPEC GEN%PVAR%1957 (i < CARD ?V) i)) (λi::nat. ff i * interior_angle1 (vec (0::nat)) ?FF (ITER i (rho_node1 ?FF) ?v)) = sum (GSPEC (λGEN%PVAR%1958::nat. ∃i::nat. SETSPEC GEN%PVAR%1958 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fv)) i)) (λi::nat. ff i * interior_angle1 (vec (0::nat)) ?fv (ITER i (rho_node1 ?FF) ?v)) + sum (GSPEC (λGEN%PVAR%1959::nat. ∃i::nat. SETSPEC GEN%PVAR%1959 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fw)) i)) (λi::nat. ff i * interior_angle1 (vec (0::nat)) ?fw (ITER i (rho_node1 ?FF) ?v)))

thm Local_lemmas1.THE_SLICING_INTO_2_LEMMA:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ (local_fan (?V, ?E, ?FF) ∧ IN

(?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀ (z::(real, 3) cart) t::(real, 3) cart. IN z (INSERT ?v (INSERT ?w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY)))) ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E))) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ∧ (?fw::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?w, rho_node1 ?FF ?w) ⟶ convex_local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv) ∧ convex_local_fan (v_prime ?V ?fw, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?w (INSERT ?v EMPTY)) EMPTY)) ?fw, ?fw) ∧ (∀ ff::nat ⇒ real. sum (GSPEC (λGEN%PVAR%1960::nat. ∃ i::nat. SETSPEC GEN%PVAR%1960 (i < CARD ?V) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?FF (ITER i (rho_node1 ?FF) ?v)) = sum (GSPEC (λGEN%PVAR%1961::nat. ∃ i::nat. SETSPEC GEN%PVAR%1961 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fv)) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?fv (ITER i (rho_node1 ?FF) ?v)) + sum (GSPEC (λGEN%PVAR%1962::nat. ∃ i::nat. SETSPEC GEN%PVAR%1962 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fw)) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?fw (ITER i (rho_node1 ?FF) ?v)))

thm Local_lemmas1.WEDGE_IN_FAN_LOFA_DETER2:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ⟶ wedge_in_fan_gt (?v, rho_node1 ?FF ?v) ?E = wedge (vec (0::nat)) ?v (rho_node1 ?FF ?v) (ivs_rho_node1 ?FF ?v)

thm Local_lemmas1.AZIM_COND_FOR_COPLANAR:

(azim (?v0.0::(real, 3) cart) (?v1.0::(real, 3) cart) (?w1.0::(real, 3) cart) (?w2.0::(real, 3) cart) = (0::real) ∨ azim ?v0.0 ?v1.0 ?w1.0 ?w2.0 = pi) = coplanar (INSERT ?v0.0 (INSERT ?v1.0 (INSERT ?w1.0 (INSERT ?w2.0 EMPTY))))

thm Local_lemmas1.AFF_SUB_PLANE:

plane (?P::(real, ?'a::type) cart ⇒ bool) ∧ SUBSET (?S::(real, ?'a::type) cart ⇒ bool) ?P ⟶ SUBSET (aff ?S) ?P

thm Local_lemmas1.PROVE_THE_SLICE_ASSUMPTION:

∀ (v::(real, 3) cart) w::(real, 3) cart. convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN v ?V ∧ IN w ?V ∧ v ≠ w ∧ (∀ x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT w EMPTY))) (wedge_in_fan_gt x ?E)) ⟶ (∀ (z::(real, 3) cart) *t::(real, 3) cart. IN z (INSERT v (INSERT w EMPTY)) ∧ IN t (DIFF ?V (INSERT z EMPTY)) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT z (INSERT t EMPTY))))*

thm Local_lemmas1.EJRCFJD:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ∧ ?v ≠ ?w ∧ (∀x::(real, 3) cart × (real, 3) cart. IN x ?FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY))) (wedge_in_fan_gt x ?E)) ∧ (?HS::((real, 3) cart × (real, 3) cart) hypermap) = hypermap (HYP (vec (0::nat), ?V, HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY))) ∧ (?fv::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?v, rho_node1 ?FF ?v) ∧ (?fw::(real, 3) cart × (real, 3) cart ⇒ bool) = face ?HS (?w, rho_node1 ?FF ?w) ⟶ convex_local_fan (v_prime ?V ?fv, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?v (INSERT ?w EMPTY)) EMPTY)) ?fv, ?fv) ∧ convex_local_fan (v_prime ?V ?fw, e_prime (HOL_Light_Import.UNION ?E (INSERT (INSERT ?w (INSERT ?v EMPTY)) EMPTY)) ?fw, ?fw) ∧ (∀ff::nat ⇒ real. sum (GSPEC (λGEN%PVAR%1965::nat. ∃i::nat. SETSPEC GEN%PVAR%1965 (i < CARD ?V) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?FF (ITER i (rho_node1 ?FF) ?v)) = sum (GSPEC (λGEN%PVAR%1966::nat. ∃i::nat. SETSPEC GEN%PVAR%1966 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fv)) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?fv (ITER i (rho_node1 ?FF) ?v)) + sum (GSPEC (λGEN%PVAR%1967::nat. ∃i::nat. SETSPEC GEN%PVAR%1967 (i < CARD ?V ∧ IN (ITER i (rho_node1 ?FF) ?v) (v_prime ?V ?fw)) i)) (λi::nat. ff i ∗ interior_angle1 (vec (0::nat)) ?fw (ITER i (rho_node1 ?FF) ?v)))

thm Local_lemmas1.DIST_TRIANGLE_AS_ABS:

|distance (?x::(real, ?'a::type) cart, ?y::(real, ?'a::type) cart) − distance (?x, ?z::(real, ?'a::type) cart)| ≤ distance (?y, ?z)

thm Local_lemmas1.CON_ATREAL_REAL_CON2_REDO:

continuous (?f::real ⇒ (real, ?'a::type) cart) (atreal (?r::real)) ∧ continuous (?g::real ⇒ (real, ?'a::type) cart) (atreal ?r) ⟶ real_continuous (λt::real. distance (?f t, ?g t)) (atreal ?r)

thm Local_lemmas1.REAL_POS_LT_MUL:

(0::real) ≤ (?a::real) ∧ ?a < (?b::real) ∧ (0::real) ≤ (?x::real) ∧ ?x < (?y::real) ⟶ ?a ∗ ?x < ?b ∗ ?y

thm Local_lemmas1.REAL_CONTINUOUS_ATREAL_IMP_MUL_FUN:

real_continuous (?f::real ⇒ real) (atreal (?r::real)) ∧ real_continuous (?g::real ⇒ real) (atreal ?r) ⟶ real_continuous (λt::real. ?f t ∗ ?g t) (atreal ?r)

thm Local_lemmas1.REAL_CONTINUOUS_ATREAL_POW_2:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \longrightarrow real\_continuous\ (\lambda t::real.\ (?f\ t)^2)\ (atreal\ ?r)$

thm Local_lemmas1.CONSTANCE_FUN_CONTINUOUS:

$\forall c::real.\ real\_continuous\ (\lambda t::real.\ c)\ (atreal\ (?r::real))$

thm Local_lemmas1.REAL_CONS_IMP_SCALAR_MUL:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \longrightarrow (\forall c::real.\ real\_continuous\ (\lambda t::real.\ c * ?f\ t)\ (atreal\ ?r))$

thm Local_lemmas1.CONS_IMP_SO_IVS:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \longrightarrow real\_continuous\ (\lambda t::real.\ -\ ?f\ t)\ (atreal\ ?r)$

thm Local_lemmas1.REAL_CONS_IMP_SUM_CONS:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \wedge real\_continuous\ (?g::real \Rightarrow real)\ (atreal\ ?r) \longrightarrow real\_continuous\ (\lambda t::real.\ ?f\ t + ?g\ t)\ (atreal\ ?r)$

thm Local_lemmas1.REAL_CONS_IMP_SCALAR_MUL_ALT:

$\forall c::real.\ real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \longrightarrow real\_continuous\ (\lambda t::real.\ c * ?f\ t)\ (atreal\ ?r)$

thm Local_lemmas1.UPS_X_CONTS_FUNC:

$continuous\ (?f::real \Rightarrow (real,\ ?'a::type)\ cart)\ (atreal\ (?r::real)) \wedge continuous\ (?g::real \Rightarrow (real,\ ?'a::type)\ cart)\ (atreal\ ?r) \wedge continuous\ (?h::real \Rightarrow (real,\ ?'a::type)\ cart)\ (atreal\ ?r) \longrightarrow real\_continuous\ (\lambda r::real.\ ups\_x\ ((distance\ (?f\ r,\ ?g\ r))^2)\ ((distance\ (?h\ r,\ ?f\ r))^2)\ ((distance\ (?g\ r,\ ?h\ r))^2))\ (atreal\ ?r)$

thm Local_lemmas1.CONTS_FUN_CONTINUOUS_ATREAL:

$\forall v0::(real,\ ?'a::type)\ cart.\ continuous\ (\lambda t::real.\ v0)\ (atreal\ (?r::real))$

thm Local_lemmas1.REAL_CONS_STILL_DIFF:

$real\_continuous\ (?f::real \Rightarrow real)\ (atreal\ (?r::real)) \wedge ?f\ ?r \neq (?a::real) \longrightarrow (\exists d>0::real.\ \forall rr::real.\ |rr - ?r| < d \longrightarrow ?f\ rr \neq ?a)$

thm Local_lemmas1.CONTINUOUS_PRESERVE_COLLINEAR:

$continuous\ (?f::real \Rightarrow (real,\ 3)\ cart)\ (atreal\ (?r::real)) \wedge continuous\ (?g::real \Rightarrow (real,\ 3)\ cart)\ (atreal\ ?r) \wedge \neg\ collinear\ (INSERT\ (?v0.0::(real,\ 3)\ cart)\ (INSERT\ (?f\ ?r)\ (INSERT\ (?g\ ?r)\ EMPTY))) \longrightarrow (\exists e>0::real.\ \forall r'::real.\ |?r - r'| < e \longrightarrow \neg\ collinear\ (INSERT\ ?v0.0\ (INSERT\ (?f\ r')\ (INSERT\ (?g\ r')\ EMPTY))))$

thm Local_lemmas1.EACH_ELM_PRESERVED_IMP_ALL:

$(\forall x::?'a::type.\ IN\ x\ (?E::?'a::type \Rightarrow bool) \longrightarrow (\exists e>0::real.\ \forall t::real.\ |t| < e \longrightarrow (?P::?'a::type \Rightarrow real \Rightarrow bool)\ x\ t)) \wedge FINITE\ ?E \longrightarrow (\exists e>0::real.\ \forall t::real.\ |t| < e \longrightarrow (\forall x::?'a::type.\ IN\ x\ ?E \longrightarrow ?P\ x\ t))$

thm Local_lemmas1.EACH_ELM_PRESERVED_IMP_ALLL:

(∀ x::(real, 3) cart ⇒ bool. IN x (?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ (∃ e>0::real. ∀ t::real. |t| < e ⟶ ¬ collinear (HOL_Light_Import.UNION (INSERT (vec (0::nat)) EMPTY) (IMAGE (λv::(real, 3) cart. (?phii::(real, 3) cart ⇒ real ⇒ (real, 3) cart) v t) x)))) ∧ FINITE ?E ⟶ (∃ e>0::real. ∀ t::real. |t| < e ⟶ (∀ e::(real, 3) cart ⇒ bool. IN e (IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E) ⟶ ¬ collinear (HOL_Light_Import.UNION (INSERT (vec (0::nat)) EMPTY) e)))

thm Local_lemmas1.ALL_TO_THE_NONPARALLEL_PART:

deformation (?phii::(real, 3) cart ⇒ real ⇒ (real, 3) cart) (?V::(real, 3) cart ⇒ bool) (?a::real, ?b::real) ∧ FAN (vec (0::nat), ?V, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ⟶ (∃ e>0::real. ∀ t::real. − e < t ∧ t < e ⟶ SUBSET (UNIONS (IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E)) (IMAGE (λv::(real, 3) cart. ?phii v t) ?V) ∧ graph (IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E) ∧ fan1 (vec (0::nat), IMAGE (λv::(real, 3) cart. ?phii v t) ?V, IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E) ∧ fan2 (vec (0::nat), IMAGE (λv::(real, 3) cart. ?phii v t) ?V, IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E) ∧ fan6 (vec (0::nat), IMAGE (λv::(real, 3) cart. ?phii v t) ?V, IMAGE (IMAGE (λv::(real, 3) cart. ?phii v t)) ?E))

thm DEF_sol_local_fan:

sol_local_fan = (λ(_6583110::?'a::type) (_6583111::((real, 3) cart ⇒ bool) ⇒ bool) _6583112::(real, 3) cart × (real, 3) cart ⇒ bool. real_of_nat (2::nat) ∗ pi + sum _6583112 (λe::(real, 3) cart × (real, 3) cart. azim_in_fan e _6583111 − pi))

thm Nkezbfc_local.sol_local_fan:

∀ (V::?'a::type) (f::(real, 3) cart × (real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. sol_local_fan V E f = real_of_nat (2::nat) ∗ pi + sum f (λe::(real, 3) cart × (real, 3) cart. azim_in_fan e E − pi)

thm DEF_sol_local:

sol_local = (λ(_6583131::((real, 3) cart ⇒ bool) ⇒ bool) _6583132::(real, 3) cart × (real, 3) cart ⇒ bool. real_of_nat (2::nat) ∗ pi + sum _6583132 (λe::(real, 3) cart × (real, 3) cart. azim_in_fan e _6583131 − pi))

thm Nkezbfc_local.sol_local:

∀ (f::(real, 3) cart × (real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. sol_local E f = real_of_nat (2::nat) ∗ pi + sum f (λe::(real, 3) cart × (real, 3) cart. azim_in_fan e E − pi)

thm Nkezbfc_local.CONVEX_LOFA_IMP_INANGLE_EQ_AZIM:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀ v::(real, 3) cart. IN *v ?V ⟶ interior_angle1 (vec (0::nat)) ?FF v = azim_in_fan (v, rho_node1 ?FF v) ?E)*

thm Nkezbfc_local.SOL_LOFA_EQ_SUM_INANGLE:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ sol_local ?E ?FF = real_of_nat (2::nat) ∗ pi + sum ?V (λv::(real, 3) cart. interior_angle1 (vec (0::nat)) ?FF v − pi)

thm Nkezbfc_local.CARD_VERTEX_GE_3_LOCAL_FAN:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) FF::(real, 3) cart × (real, 3) cart ⇒ bool. convex_local_fan (V, E, FF) ⟶ (3::nat) ≤ CARD V

thm Nkezbfc_local.REP_VERTEX_3_LOCAL_FAN:

CARD (?V::(real, 3) cart ⇒ bool) = (3::nat) ∧ convex_local_fan (?V, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∃ v::(real, 3) cart. ?V = INSERT v (INSERT (rho_node1 ?FF v) (INSERT (rho_node1 ?FF (rho_node1 ?FF v)) EMPTY)) ∧ v ≠ rho_node1 ?FF v ∧ rho_node1 ?FF v ≠ rho_node1 ?FF (rho_node1 ?FF v) ∧ rho_node1 ?FF (rho_node1 ?FF v) ≠ v)

thm Nkezbfc_local.CONVEX_LOFA_IMP_INANGLE_EQ_AZIM_IVS:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀ v::(real, 3) cart. IN v ?V ⟶ interior_angle1 (vec (0::nat)) ?FF v = azim (vec (0::nat)) v (rho_node1 ?FF v) (ivs_rho_node1 ?FF v))

thm Nkezbfc_local.SOL_LOCAL_FAN_POS_CASE3:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) FF::(real, 3) cart × (real, 3) cart ⇒ bool. convex_local_fan (V, E, FF) ∧ CARD V = (3::nat) ⟶ (0::real) ≤ sol_local E FF

thm Nkezbfc_local.AFF_LT_1_1:

∀ (x::(real, ?'a::type) cart) w::(real, ?'a::type) cart. x ≠ w ⟶ aff_lt (INSERT x EMPTY) (INSERT w EMPTY) = GSPEC (λGEN%PVAR%1968::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%1968 (∃ (t1::real) t2::real. t2 < (0::real) ∧ t1 + t2 = (1::real) ∧ y = vector_add (% t1 x) (% t2 w)) y)

thm Nkezbfc_local.PROPERTIES_GENERIC_LOCAL_FAN:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) v0::(real, 3) cart. local_fan (V, E, FF) ∧ IN v0 V ∧ generic V E ⟶ (∀ v::(real, 3) cart. IN v V ∧ v ≠ v0 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT v0 (INSERT v EMPTY))))

thm Nkezbfc_local.PROPERTIES_GENERIC:

local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ generic ?V ?E ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ⟶ (∀(u::(real, 3) cart) u1::(real, 3) cart. IN u (INSERT ?v (INSERT ?w EMPTY)) ∧ IN u1 ?V ∧ u ≠ u1 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT u (INSERT u1 EMPTY))))

thm Nkezbfc_local.PROPERTIES_GENERIC1:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ generic ?V ?E ∧ IN (?v::(real, 3) cart) ?V ∧ IN (?w::(real, 3) cart) ?V ⟶ (∀(u::(real, 3) cart) u1::(real, 3) cart. IN u (INSERT ?v (INSERT ?w EMPTY)) ∧ IN u1 ?V ∧ u ≠ u1 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT u (INSERT u1 EMPTY))))

thm Nkezbfc_local.AZIM_PI_WEDGE_SIN:

azim (?u::(real, 3) cart) (?v::(real, 3) cart) (?w::(real, 3) cart) (?ww::(real, 3) cart) = pi ⟶ wedge ?u ?v ?w ?ww = GSPEC (λGEN%PVAR%1969::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1969 ((0::real) < sin (azim ?u ?v ?w x)) x)

thm Nkezbfc_local.AZIM_PI_WEDGE_CROSS_DOT:

azim (?u::(real, 3) cart) (?v::(real, 3) cart) (?w::(real, 3) cart) (?ww::(real, 3) cart) = pi ⟶ wedge ?u ?v ?w ?ww = GSPEC (λGEN%PVAR%1970::(real, 3) cart. ∃x::(real, 3) cart. SETSPEC GEN%PVAR%1970 ((0::real) < dot (cross (vector_sub ?v ?u) (vector_sub ?w ?u)) (vector_sub x ?u)) x)

thm Nkezbfc_local.AFF_GT_SUBSET_WEDGE_IMP_VERTEX:

∀(x::(real, 3) cart) (v::(real, 3) cart) (w::(real, 3) cart) (y::(real, 3) cart) z::(real, 3) cart. ¬ collinear (INSERT x (INSERT v (INSERT w EMPTY))) ∧ ¬ collinear (INSERT x (INSERT v (INSERT y EMPTY))) ∧ ¬ collinear (INSERT x (INSERT v (INSERT z EMPTY))) ∧ SUBSET (aff_gt (INSERT x EMPTY) (INSERT v (INSERT w EMPTY))) (wedge x v y z) ⟶ IN w (wedge x v y z)

thm Nkezbfc_local.CONDITION_INANGLE_CROSS_DOT:

SUBSET (aff_gt (INSERT (?x::(real, 3) cart) EMPTY) (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY))) (wedge ?x ?v (?y::(real, 3) cart) (?z::(real, 3) cart)) ∧ ¬ collinear (INSERT ?x (INSERT ?v (INSERT ?w EMPTY))) ∧ ¬ collinear (INSERT ?x (INSERT ?v (INSERT ?y EMPTY))) ∧ ¬ collinear (INSERT ?x (INSERT ?v (INSERT ?z EMPTY))) ∧ (?u::(real, 3) cart) = cross (vector_sub ?v ?x) (vector_sub ?w ?x) ∧ azim ?x ?v ?y ?z < pi ⟶ (0::real) < dot (cross (vector_sub ?v ?x) (vector_sub ?y ?x)) (vector_sub ?w ?x) ∧ (0::real) < dot (cross (vector_sub ?v ?x) (vector_sub ?w ?x)) (vector_sub ?z ?x)

thm Nkezbfc_local.AFF_GE_3_1:

∀(x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) w::(real, ?'a::type) cart. DISJOINT (INSERT x (INSERT v (INSERT u EMPTY)))

(*INSERT w EMPTY*) ⟶ *aff_ge* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*) = *GSPEC* (λ*GEN%PVAR%1971*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1971* (∃ (*t1*::*real*) (*t2*::*real*) (*t3*::*real*) *t4*::*real*. (*0*::*real*) ≤ *t4* ∧ *t1* + (*t2* + (*t3* + *t4*)) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 v*) (*vector_add* (% *t3 u*) (% *t4 w*)))) *y*)

thm Nkezbfc_local.AFF_GE_2_2:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v* (*INSERT w EMPTY*)) ⟶ *aff_ge* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v* (*INSERT w EMPTY*)) = *GSPEC* (λ*GEN%PVAR%1972*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%1972* (∃ (*t1*::*real*) (*t2*::*real*) (*t3*::*real*) *t4*::*real*. (*0*::*real*) ≤ *t3* ∧ (*0*::*real*) ≤ *t4* ∧ *t1* + (*t2* + (*t3* + *t4*)) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1 x*) (*vector_add* (% *t2 u*) (*vector_add* (% *t3 v*) (% *t4 w*)))) *y*)

thm Nkezbfc_local.inter_aff_ge_3_1_is_aff_ge_2_2:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)) (*aff_ge* (*INSERT x* (*INSERT u* (*INSERT w EMPTY*))) (*INSERT v EMPTY*)) = *aff_ge* (*INSERT x* (*INSERT u EMPTY*)) (*INSERT v* (*INSERT w EMPTY*))

thm Nkezbfc_local.aff_ge_3_1_rep_cross_dot:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ∧ (*0*::*real*) < *dot* (*cross* (*vector_sub v x*) (*vector_sub u x*)) (*vector_sub w x*) ⟶ *aff_ge* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*) = *GSPEC* (λ*GEN%PVAR%1973*::(*real*, *3*) *cart*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1973* ((*0*::*real*) ≤ *dot* (*cross* (*vector_sub v x*) (*vector_sub u x*)) (*vector_sub y x*)) *y*)

thm Nkezbfc_local.lemma:

∀ *A*::*bool*. *A* ∨ ¬ *A*

thm Nkezbfc_local.PROPERTIES_AFF_GT_SUBSET_WEDGE:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ *generic ?V ?E* ∧ *azim_in_fan* (*?v*, *rho_node1 ?FF ?v*) *?E* < *pi* ∧ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt* (*?v*, *rho_node1 ?FF ?v*) *?E*) ⟶ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*))

thm Nkezbfc_local.AFF_GE_SUBSET_AFF_GE_UNION:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v1::(real,\ 3)\ cart.$ $DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ u\ EMPTY)) \wedge IN\ v1\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow SUBSET\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)))\ (HOL\_Light\_Import.UNION\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ u\ EMPTY))))$

thm Nkezbfc_local.aff_ge_subset3_aff_ge:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v1::(real,\ 3)\ cart.$ $DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)) \wedge IN\ v1\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow SUBSET\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)))$

thm Nkezbfc_local.AFF_GE_EQ_AFF_GE_UNION:

$\forall (x::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v1::(real,\ 3)\ cart.$ $DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)) \wedge DISJOINT\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ u\ EMPTY)) \wedge IN\ v1\ (aff\_gt\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY))) \longrightarrow aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ u\ EMPTY)) = HOL\_Light\_Import.UNION\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v\ (INSERT\ v1\ EMPTY)))\ (aff\_ge\ (INSERT\ x\ EMPTY)\ (INSERT\ v1\ (INSERT\ u\ EMPTY)))$

thm DEF_order:

$order = (\lambda(\_6587564::?'a::type \Rightarrow ?'a::type)\ (\_6587565::?'a::type)\ \_6587566::?'a::type.$ $SOME\ n::nat.\ ITER\ n\ \_6587564\ \_6587565 = \_6587566 \wedge (\forall i::nat.\ (0::nat) < i \wedge i < n \longrightarrow ITER\ i\ \_6587564\ \_6587565 \neq \_6587566))$

thm Nkezbfc_local.order:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (x::?'a::type)\ y::?'a::type.\ order\ f\ x\ y = (SOME\ n::nat.\ ITER\ n\ f\ x = y \wedge (\forall i::nat.\ (0::nat) < i \wedge i < n \longrightarrow ITER\ i\ f\ x \neq y))$

thm DEF_slicev:

$slicev = (\lambda(\_6587585::?'a::type)\ (\_6587586::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (\_6587587::(real,\ 3)\ cart)\ \_6587588::(real,\ 3)\ cart.\ GSPEC\ (\lambda GEN\%PVAR\%1980::(real,\ 3)\ cart.\ \exists u::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%1980\ (\exists n \geq 0::nat.\ n \leq order\ (rho\_node1\ \_6587586)\ \_6587587\ \_6587588 \wedge u = ITER\ n\ (rho\_node1\ \_6587586)\ \_6587587)\ u))$

thm Nkezbfc_local.slicev:

$\forall (E::?'a::type)\ (w::(real,\ 3)\ cart)\ (FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ slicev\ E\ FF\ v\ w = GSPEC\ (\lambda GEN\%PVAR\%1980::(real,\ 3)$

cart. ∃ *u*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1980* (∃ *n*≥*0*::*nat*. *n* ≤ *order* (*rho_node1 FF*) *v w* ∧ *u* = *ITER n* (*rho_node1 FF*) *v*) *u*)

thm DEF_slicee:

slicee = (λ(*_6587617*::*?'a*::*type*) (*_6587618*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*_6587619*::(*real*, *3*) *cart*) *_6587620*::(*real*, *3*) *cart*. *HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%1981*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%1981* (∃ *u*::(*real*, *3*) *cart*. *IN u* (*DELETE* (*slicev _6587617 _6587618 _6587619 _6587620*) *_6587620*) ∧ *e* = *INSERT u* (*INSERT* (*rho_node1 _6587618 u*) *EMPTY*)) *e*)) (*INSERT* (*INSERT _6587620* (*INSERT _6587619 EMPTY*)) *EMPTY*))

thm Nkezbfc_local.slicee:

∀ (*E*::*?'a*::*type*) (*FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*w*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *slicee E FF v w* = *HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%1981*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%1981* (∃ *u*::(*real*, *3*) *cart*. *IN u* (*DELETE* (*slicev E FF v w*) *w*) ∧ *e* = *INSERT u* (*INSERT* (*rho_node1 FF u*) *EMPTY*)) *e*)) (*INSERT* (*INSERT w* (*INSERT v EMPTY*)) *EMPTY*)

thm DEF_slicef:

slicef = (λ(*_6587649*::*?'a*::*type*) (*_6587650*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*_6587651*::(*real*, *3*) *cart*) *_6587652*::(*real*, *3*) *cart*. *HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%1982*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1982* (∃ *u*::(*real*, *3*) *cart*. *IN u* (*DELETE* (*slicev _6587649 _6587650 _6587651 _6587652*) *_6587652*) ∧ *f* = (*u*, *rho_node1 _6587650 u*)) *f*)) (*INSERT* (*_6587652*, *_6587651*) *EMPTY*))

thm Nkezbfc_local.slicef:

∀ (*E*::*?'a*::*type*) (*FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*w*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. *slicef E FF v w* = *HOL_Light_Import*.*UNION* (*GSPEC* (λ*GEN%PVAR%1982*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%1982* (∃ *u*::(*real*, *3*) *cart*. *IN u* (*DELETE* (*slicev E FF v w*) *w*) ∧ *f* = (*u*, *rho_node1 FF u*)) *f*)) (*INSERT* (*w*, *v*) *EMPTY*)

thm DEF_rho_fun:

rho_fun = (λ*_6587681*::*real*. (*1*::*real*) + *inverse_class*.*inverse* (*real_of_nat* (*2*::*nat*) ∗ *h0* − *real_of_nat* (*2*::*nat*)) ∗ (*inverse_class*.*inverse pi* ∗ (*sol0* ∗ (*_6587681* − *real_of_nat* (*2*::*nat*)))))

thm Dih2k_hypermap.rho_fun:

∀ *y*::*real*. *rho_fun y* = (*1*::*real*) + *inverse_class*.*inverse* (*real_of_nat* (*2*::*nat*) ∗ *h0* − *real_of_nat* (*2*::*nat*)) ∗ (*inverse_class*.*inverse pi* ∗ (*sol0* ∗ (*y* − *real_of_nat* (*2*::*nat*))))

thm DEF_tau_fun:

tau_fun = (λ(*_6587686*::*?'a*::*type*) (*_6587687*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *_6587688*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *sum _6587688* (λ*e*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *rho_fun* (*vector_norm* (*fst e*)) ∗ *azim_in_fan e _6587687*) − (*pi* + *sol0*) ∗ *real_of_nat* (*CARD _6587688* − (*2*::*nat*)))

thm Nkezbfc_local.tau_fun:

∀(*V*::*?'a*::*type*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *tau_fun V E f* = *sum f* (λ*e*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *rho_fun* (*vector_norm* (*fst e*)) ∗ *azim_in_fan e E*) − (*pi* + *sol0*) ∗ *real_of_nat* (*CARD f* − (*2*::*nat*))

thm Nkezbfc_local.ORDER:

ITER (*?n*::*nat*) (*?f*::*?'a*::*type* ⇒ *?'a*::*type*) (*?x*::*?'a*::*type*) = (*?y*::*?'a*::*type*) ∧ (∀*i*::*nat*. (*0*::*nat*) < *i* ∧ *i* < *?n* ⟶ *ITER i ?f ?x* ≠ *?y*) ⟶ *ITER* (*order ?f ?x ?y*) *?f ?x* = *?y* ∧ (∀*i*::*nat*. (*0*::*nat*) < *i* ∧ *i* < *order ?f ?x ?y* ⟶ *ITER i ?f ?x* ≠ *?y*)

thm Nkezbfc_local.UNIQUE_ORDER:

∀*f*::*?'a*::*type* ⇒ *?'a*::*type*. *ITER* (*?n*::*nat*) *f* (*?x*::*?'a*::*type*) = (*?y*::*?'a*::*type*) ∧ (∀*i*::*nat*. (*0*::*nat*) < *i* ∧ *i* < *?n* ⟶ *ITER i f ?x* ≠ *?y*) ∧ *?x* ≠ *?y* ⟶ *order f ?x ?y* = *?n*

thm Nkezbfc_local.COMPATIBLE_BW_TWO_LEMMAS:

convex_local_fan (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*)) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*) ⟶ *v_prime ?V ?fv* = *slicev ?E ?FF ?v ?w* ∧ *e_prime* (*HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*)) *?fv* = *slicee ?E ?FF ?v ?w* ∧ *?fv* = *slicef ?E ?FF ?v ?w*

thm Nkezbfc_local.COMPATIBLE_BW_TWO_LEMMAS2:

(*convex_local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* (*?v*::(*real*, *3*) *cart*) *?V* ∧ *IN* (*?w*::(*real*, *3*) *cart*) *?V* ∧ *?v* ≠ *?w* ∧ (∀*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x ?FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?v* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x ?E*)) ∧ (*?HS*::((*real*, *3*) *cart* × (*real*, *3*) *cart*) *hypermap*) = *hypermap* (*HYP* (*vec* (*0*::*nat*), *?V*, *HOL_Light_Import*.*UNION ?E* (*INSERT* (*INSERT ?v* (*INSERT ?w EMPTY*)) *EMPTY*))) ∧ (*?fv*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?v*, *rho_node1 ?FF ?v*)) ∧ (*?fw*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face ?HS* (*?w*, *rho_node1 ?FF ?w*) ⟶ (*v_prime ?V ?fv* = *slicev ?E ?FF ?v ?w* ∧

e_prime $(HOL\_Light\_Import.UNION$ $?E$ $(INSERT$ $(INSERT$ $?v$ $(INSERT$ $?w$ $EMPTY))$ $EMPTY))$ $?fv = slicee$ $?E$ $?FF$ $?v$ $?w \wedge ?fv = slicef$ $?E$ $?FF$ $?v$ $?w)$ $\wedge$ v_prime $?V$ $?fw = slicev$ $?E$ $?FF$ $?w$ $?v \wedge e\_prime$ $(HOL\_Light\_Import.UNION$ $?E$ $(INSERT$ $(INSERT$ $?w$ $(INSERT$ $?v$ $EMPTY))$ $EMPTY))$ $?fw = slicee$ $?E$ $?FF$ $?w$ $?v \wedge ?fw = slicef$ $?E$ $?FF$ $?w$ $?v$

thm Nkezbfc_local.lemma1:

$\forall A{::}bool.\ \neg A \vee A$

thm Nkezbfc_local.NKEZBFC:

$(\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (v{::}(real,\ 3)\ cart)\ w{::}(real,\ 3)\ cart.\ convex\_local\_fan\ (V,\ E,\ FF) \wedge IN\ v\ V \wedge IN\ w\ V \wedge (\forall (u{::}(real,\ 3)\ cart)\ u1{::}(real,\ 3)\ cart.\ IN\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)) \wedge IN\ u1\ V \wedge u \neq u1 \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0{::}nat))\ (INSERT\ u\ (INSERT\ u1\ EMPTY)))) \wedge (\forall e{::}(real,\ 3)\ cart \times (real,\ 3)\ cart.\ IN\ e\ FF \longrightarrow SUBSET\ (aff\_gt\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (wedge\_in\_fan\_gt\ e\ E)) \longrightarrow convex\_local\_fan\ (slicev\ E\ FF\ v\ w,\ slicee\ E\ FF\ v\ w,\ slicef\ E\ FF\ v\ w) \wedge convex\_local\_fan\ (slicev\ E\ FF\ w\ v,\ slicee\ E\ FF\ w\ v,\ slicef\ E\ FF\ w\ v) \wedge tau\_fun\ (slicev\ E\ FF\ v\ w)\ (slicee\ E\ FF\ v\ w)\ (slicef\ E\ FF\ v\ w) + tau\_fun\ (slicev\ E\ FF\ w\ v)\ (slicee\ E\ FF\ w\ v)\ (slicef\ E\ FF\ w\ v) \leq tau\_fun\ V\ E\ FF \wedge sol\_local\ E\ FF = sol\_local\ (slicee\ E\ FF\ v\ w)\ (slicef\ E\ FF\ v\ w) + sol\_local\ (slicee\ E\ FF\ w\ v)\ (slicef\ E\ FF\ w\ v) \wedge CARD\ (slicev\ E\ FF\ v\ w) < CARD\ V \wedge CARD\ (slicev\ E\ FF\ w\ v) < CARD\ V \wedge (generic\ V\ E \longrightarrow generic\ (slicev\ E\ FF\ v\ w)\ (slicee\ E\ FF\ v\ w) \wedge generic\ (slicev\ E\ FF\ w\ v)\ (slicee\ E\ FF\ w\ v))) \longrightarrow (\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ FF{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ convex\_local\_fan\ (V,\ E,\ FF) \wedge generic\ V\ E \longrightarrow (0{::}real) \leq sol\_local\ E\ FF)$

thm Arc_properties.REALLIM_ATREAL_LOCAL:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ (x{::}real)\ y{::}real.\ {-}{-}{-}{>}\ g\ y\ (atreal\ x) \wedge (\exists s{::}real \Rightarrow bool.\ real\_open\ s \wedge IN\ x\ s \wedge (\forall y{::}real.\ IN\ y\ s \longrightarrow f\ y = g\ y)) \longrightarrow {-}{-}{-}{>}\ f\ y\ (atreal\ x)$

thm Arc_properties.HAS_REAL_DERIVATIVE_LOCAL:

$\forall (f{::}real \Rightarrow real)\ (g{::}real \Rightarrow real)\ (x{::}real)\ g'x{::}real.\ has\_real\_derivative\ g\ g'x\ (atreal\ x) \wedge (\exists s{::}real \Rightarrow bool.\ real\_open\ s \wedge IN\ x\ s \wedge (\forall y{::}real.\ IN\ y\ s \longrightarrow f\ y = g\ y)) \longrightarrow has\_real\_derivative\ f\ g'x\ (atreal\ x)$

thm Arc_properties.REAL_LT_ONE_LDIV:

$\forall (a{::}real)\ b{::}real.\ (0{::}real) < b \wedge a < b \longrightarrow a\ /\ b < (1{::}real)$

thm Arc_properties.arc_derivative:

$\forall (x{::}real)\ b{::}real.\ (real\_of\_nat\ (2{::}nat) \leq x \wedge x \leq DECIMAL\ (252{::}nat)\ (100{::}nat)) \wedge real\_of\_nat\ (2{::}nat) \leq b \wedge b \leq DECIMAL\ (252{::}nat)\ (100{::}nat) \longrightarrow has\_real\_derivative\ (\lambda x{::}real.\ arclength\ x\ b\ (real\_of\_nat\ (2{::}nat)))\ (((x + x) * (real\_of\_nat\ (2{::}nat)$

$* (x * b)) - (x * x + (b * b - real\_of\_nat\ (2::nat) * real\_of\_nat\ (2::nat))) * (real\_of\_nat\ (2::nat) * b)) / (real\_of\_nat\ (2::nat) * (x * b))^2 * - inverse\_class.inverse\ (sqrt\ ((1::real) - ((x * x + (b * b - real\_of\_nat\ (2::nat) * real\_of\_nat\ (2::nat))) / (real\_of\_nat\ (2::nat) * (x * b)))^2))) (within\ (atreal\ x)\ (closed\_real\_interval\ [(real\_of\_nat\ (2::nat), DECIMAL\ (252::nat)\ (100::nat))]))$

thm Arc_properties.COS_EQ_NEG_SIN:

$\forall x::real.\ cos\ (x + pi / real\_of\_nat\ (2::nat)) = - sin\ x$

thm Arc_properties.COS_DERIVATIVES:

$\forall (x::real)\ n::nat.\ has\_real\_derivative\ (\lambda x::real.\ cos\ (x + real\_of\_nat\ n * (pi / real\_of\_nat\ (2::nat))))\ (cos\ (x + real\_of\_nat\ (n + (1::nat)) * (pi / real\_of\_nat\ (2::nat))))\ (atreal\ x)$

thm Arc_properties.REAL_TAYLOR_COS_RAW:

$\forall (x::real)\ n::nat.\ |cos\ x - sum\ (dotdot\ (0::nat)\ n)\ (\lambda k::nat.\ if\ even\ k\ then\ (-(1::real))^{k\ div\ (2::nat)} * x^k / real\_of\_nat\ (fact\ k)\ else\ (0::real))| \leq |x|^{n + (1::nat)} / real\_of\_nat\ (fact\ (n + (1::nat)))$

thm Arc_properties.SUM_PAIR_0:

$\forall (f::nat \Rightarrow real)\ n::nat.\ sum\ (dotdot\ (0::nat)\ ((2::nat) * n + (1::nat)))\ f = sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ f\ ((2::nat) * i) + f\ ((2::nat) * i + (1::nat)))$

thm Arc_properties.REAL_TAYLOR_COS:

$\forall (x::real)\ n::nat.\ |cos\ x - sum\ (dotdot\ (0::nat)\ n)\ (\lambda i::nat.\ (- (1::real))^i * (x^{(2::nat) * i} / real\_of\_nat\ (fact\ ((2::nat) * i))))| \leq |x|^{(2::nat) * n + (2::nat)} / real\_of\_nat\ (fact\ ((2::nat) * n + (2::nat)))$

thm Arc_properties.arc_lemma1:

$\forall (a::real)\ (b::real)\ c::real.\ real\_of\_nat\ (2::nat) \leq a \wedge a \leq DECIMAL\ (252::nat)\ (100::nat) \wedge real\_of\_nat\ (2::nat) \leq b \wedge b \leq DECIMAL\ (252::nat)\ (100::nat) \wedge real\_of\_nat\ (2::nat) \leq c \wedge c \leq DECIMAL\ (252::nat)\ (100::nat) \longrightarrow arclength\ a\ b\ (real\_of\_nat\ (2::nat)) \leq arclength\ a\ b\ c$

thm Arc_properties.ups_x_sym:

$\forall (a::real)\ (b::real)\ c::real.\ ups\_x\ a\ b\ c = ups\_x\ b\ a\ c$

thm Arc_properties.arc_sym:

$\forall (a::real)\ (b::real)\ c::real.\ arclength\ a\ b\ c = arclength\ b\ a\ c$

thm Arc_properties.arc_concave:

$\forall b::real.\ real\_of\_nat\ (2::nat) \leq b \wedge b \leq DECIMAL\ (252::nat)\ (100::nat) \longrightarrow real\_convex\_on\ (\lambda x::real.\ - arclength\ x\ b\ (real\_of\_nat\ (2::nat)))\ (closed\_real\_interval\ [(real\_of\_nat\ (2::nat), DECIMAL\ (252::nat)\ (100::nat))])$

thm Arc_properties.arc_lemma3:

$\forall (x::real)\ b::real.\ (real\_of\_nat\ (2::nat) \le x \wedge x \le DECIMAL\ (252::nat)\ (100::nat)) \wedge real\_of\_nat\ (2::nat) \le b \wedge b \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow arclength\ (DECIMAL\ (252::nat)\ (100::nat))\ b\ (real\_of\_nat\ (2::nat)) + lmfun\ (x\ /\ real\_of\_nat\ (2::nat)) * (arclength\ (real\_of\_nat\ (2::nat))\ b\ (real\_of\_nat\ (2::nat)) - arclength\ (DECIMAL\ (252::nat)\ (100::nat))\ b\ (real\_of\_nat\ (2::nat))) \le arclength\ x\ b\ (real\_of\_nat\ (2::nat))$

thm Arc_properties.ABS_LE_BOUNDS:

$\forall (x::real)\ (a::real)\ e::real.\ (|x - a| \le e) = (a - e \le x \wedge x \le a + e)$

thm Arc_properties.estimate0:

$DECIMAL\ (73::nat)\ (1000::nat) \le arclength\ (real\_of\_nat\ (2::nat))\ (DECIMAL\ (252::nat)\ (100::nat))\ (real\_of\_nat\ (2::nat)) - arclength\ (DECIMAL\ (252::nat)\ (100::nat))\ (DECIMAL\ (252::nat)\ (100::nat))\ (real\_of\_nat\ (2::nat))$

thm Arc_properties.estimate1:

$\forall x::real.\ real\_of\_nat\ (2::nat) \le x \wedge x \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow (1::real)\ /\ real\_of\_nat\ (4::nat) * - inverse\_class.inverse\ (sqrt\ ((1::real) - (x\ /\ real\_of\_nat\ (4::nat))^2)) \le - DECIMAL\ (28::nat)\ (100::nat)$

thm Arc_properties.estimate2:

$\forall x::real.\ real\_of\_nat\ (2::nat) \le x \wedge x \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow (real\_of\_nat\ (2::nat) * (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * (x * x))) - (real\_of\_nat\ (3969::nat)\ /\ real\_of\_nat\ (625::nat) + (x * x - real\_of\_nat\ (4::nat))) * (real\_of\_nat\ (126::nat)\ /\ real\_of\_nat\ (25::nat)))\ /\ (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * x))^2 \le DECIMAL\ (13::nat)\ (100::nat) \wedge (0::real) \le (real\_of\_nat\ (2::nat) * (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * (x * x))) - (real\_of\_nat\ (3969::nat)\ /\ real\_of\_nat\ (625::nat) + (x * x - real\_of\_nat\ (4::nat))) * (real\_of\_nat\ (126::nat)\ /\ real\_of\_nat\ (25::nat)))\ /\ (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * x))^2$

thm Arc_properties.estimate3:

$\forall x::real.\ real\_of\_nat\ (2::nat) \le x \wedge x \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow inverse\_class.inverse\ (sqrt\ ((1::real) - ((real\_of\_nat\ (3969::nat)\ /\ real\_of\_nat\ (625::nat) + (x * x - real\_of\_nat\ (4::nat)))\ /\ (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * x)))^2)) \le real\_of\_nat\ (2::nat) \wedge (0::real) \le inverse\_class.inverse\ (sqrt\ ((1::real) - ((real\_of\_nat\ (3969::nat)\ /\ real\_of\_nat\ (625::nat) + (x * x - real\_of\_nat\ (4::nat)))\ /\ (real\_of\_nat\ (2::nat) * (DECIMAL\ (252::nat)\ (100::nat) * x)))^2))$

thm Arc_properties.arc_lemma4:

$\forall x::real.\ real\_of\_nat\ (2::nat) \le x \wedge x \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow DECIMAL\ (73::nat)\ (1000::nat) \le arclength\ (real\_of\_nat\ (2::nat))\ x\ (real\_of\_nat\ (2::nat)) - arclength\ (DECIMAL\ (252::nat)\ (100::nat))\ x\ (real\_of\_nat\ (2::nat))$

thm Arc_properties.arc_lemma5:

DECIMAL (*816*::*nat*) (*1000*::*nat*) ≤ *arclength* (*DECIMAL* (*252*::*nat*) (*100*::*nat*)) (*DECIMAL* (*252*::*nat*) (*100*::*nat*)) (*real_of_nat* (*2*::*nat*))

thm DEF_EDGES0_FAN:

EDGES0_FAN = (λ(*_6600301*::(*real*, *3*) *cart* ⇒ *bool*) *_6600302*::(*real*, *3*) *cart* ⇒ *bool*. *GSPEC* (λ*GEN%PVAR%2009*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2009* (*IN e* (*edges _6600301*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *e*) (*closure* (*fchanged _6600302*)) ≠ *EMPTY*) *e*))

thm Cfyxfty.EDGES0_FAN:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) *f1*::(*real*, *3*) *cart* ⇒ *bool*. *EDGES0_FAN p f1* = *GSPEC* (λ*GEN%PVAR%2009*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2009* (*IN e* (*edges p*) ∧ *HOL_Light_Import.INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) *e*) (*closure* (*fchanged f1*)) ≠ *EMPTY*) *e*)

thm Cfyxfty.CONVEX_CLOSURE_DARTSET_LEADS_INTO_FAN:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) (*f1*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *bounded p* ∧ *polyhedron p* ∧ *IN* (*vec* (*0*::*nat*)) (*interior p*) ∧ *IN f1* (*face_set* (*hypermap1_of_fanx* (*vec* (*0*::*nat*), *vertices p*, *edges p*))) ∧ *dartset_leads_into_fan* (*vec* (*0*::*nat*)) (*vertices p*) (*edges p*) *f1* = *U* ⟶ *convex* (*closure U*)

thm Cfyxfty.CONVEX_CLOSURE_FCHANGED:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) (*f*::(*real*, *3*) *cart* ⇒ *bool*) *U*::(*real*, *3*) *cart* ⇒ *bool*. *bounded p* ∧ *polyhedron p* ∧ *IN* (*vec* (*0*::*nat*)) (*interior p*) ∧ *facet_of f p* ∧ *fchanged f* = *U* ⟶ *convex* (*closure U*)

thm Cfyxfty.GINGUAP:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ⟶ *convex* (*dartset_leads_into_fan x V E ds*)

thm Cfyxfty.AFF_GT_SUBSET_DARTSET_LEADS_INTO:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*ds*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*y*::(*real*, *3*) *cart*) *z*::(*real*, *3*) *cart*. *FAN* (*x*, *V*, *E*) ∧ *conforming_fan* (*x*, *V*, *E*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v V E*)) ∧ *fan80* (*x*, *V*, *E*) ∧ *IN ds* (*face_set* (*hypermap1_of_fanx* (*x*, *V*, *E*))) ∧ *IN y* (*dartset_leads_into_fan x V E ds*) ∧ *IN z* (*dartset_leads_into_fan x V E ds*) ⟶ *SUBSET* (*aff_gt* (*INSERT x EMPTY*) (*INSERT y* (*INSERT z EMPTY*))) (*dartset_leads_into_fan x V E ds*)

thm Cfyxfty.FACET_SUBSET_CLOSURE_FCHANGED:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ⟶ SUBSET f (closure (fchanged f))

thm Cfyxfty.MAP_EDGES_FACET_INTO_E1_FAN:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart ⇒ bool) e::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ∧ IN e (edges f) ⟶ IN e (EDGES0_FAN p f)

thm Cfyxfty.LIM_NULL_CMUL_BOUNDED_FAN:

∀ (f::?'b::type ⇒ real) (g::?'b::type ⇒ (real, ?'a::type) cart) (B::real) net::?'b::type net. eventually (λa::?'b::type. |f a| < B) net ∧ −−> g (vec (0::nat)) net ⟶ −−> (λn::?'b::type. % (f n) (g n)) (vec (0::nat)) net

thm Cfyxfty.MAP_EDGES_FACET_INTO_E1_FAN_INJ:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart ⇒ bool) e::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ∧ IN e (EDGES0_FAN p f) ⟶ IN e (edges f)

thm Cfyxfty.CONVEX_OPEN_AFF_GT_FACE:

FAN (?x::(real, 3) cart, ?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool) ∧ (∀ v::(real, 3) cart. IN v ?V ⟶ (1::nat) < CARD (set_of_edge v ?V ?E)) ∧ fan80 (?x, ?V, ?E) ∧ IN (?f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) (face_set (hypermap1_of_fanx (?x, ?V, ?E))) ⟶ (∀ s::(real, 3) cart ⇒ bool. IN s (GSPEC (λGEN%PVAR%2023::(real, 3) cart ⇒ bool. ∃ y::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%2023 (IN y ?f) (aff_gt (INSERT ?x (INSERT (pr2 y) (INSERT (pr3 y) EMPTY))) (INSERT (pr3 (f1_fan ?x ?V ?E y)) EMPTY)))) ⟶ convex s ∧ HOL_Light_Import.open s)

thm Cfyxfty.AFF_GE_3_1:

∀ (x::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) w::(real, ?'a::type) cart. DISJOINT (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT w EMPTY) ⟶ aff_ge (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT w EMPTY) = GSPEC (λGEN%PVAR%2024::(real, ?'a::type) cart. ∃ y::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2024 (∃ (t1::real) (t2::real) (t3::real) t4::real. (0::real) ≤ t4 ∧ t1 + (t2 + (t3 + t4)) = (1::real) ∧ y = vector_add (% t1 x) (vector_add (% t2 v) (vector_add (% t3 u) (% t4 w)))) y)

thm Cfyxfty.inter_aff_ge_3_1_is_aff_ge_1_3:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. ¬ coplanar (INSERT x (INSERT v (INSERT u (INSERT w EMPTY)))) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x (INSERT v (INSERT u EMPTY))) (INSERT w EMPTY)) (HOL_Light_Import.INTER (aff_ge (INSERT x (INSERT u (INSERT w EMPTY))) (INSERT v EMPTY)) (aff_ge (INSERT x (INSERT *w* (*INSERT v EMPTY*))) (*INSERT u EMPTY*))) = *aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))

thm Cfyxfty.CLOSURE_AFF_GT_3_1_EQ_AFF_GE_3_1:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ *closure* (*aff_gt* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)) = *aff_ge* (*INSERT x* (*INSERT v* (*INSERT u EMPTY*))) (*INSERT w EMPTY*)

thm Cfyxfty.CLOSURE_AFF_GT_1_3_EQ_AFF_GE_1_3:

∀ (*x*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *coplanar* (*INSERT x* (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) ⟶ *closure* (*aff_gt* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))) = *aff_ge* (*INSERT x EMPTY*) (*INSERT v* (*INSERT u* (*INSERT w EMPTY*)))

thm Cfyxfty.AFF_GT_SUBSET_CLOSURE_DARTSET_LEADS_INTO_FAN:

FAN (*?x*::(*real*, *3*) *cart*, *?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v ?V* ⟶ (*1*::*nat*) < *CARD* (*set_of_edge v ?V ?E*)) ∧ *fan80* (*?x*, *?V*, *?E*) ∧ *IN* (*?f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*face_set* (*hypermap1_of_fanx* (*?x*, *?V*, *?E*))) ∧ *IN* (*?e*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *?f* ⟶ *SUBSET* (*aff_gt* (*INSERT ?x EMPTY*) (*INSERT* (*pr2 ?e*) (*INSERT* (*pr3 ?e*) *EMPTY*))) (*closure* (*dartset_leads_into_fan ?x ?V ?E ?f*))

thm Cfyxfty.IN_EDGES0_FAN:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *f1*::(*real*, *3*) *cart* ⇒ *bool*. *bounded p* ∧ *polyhedron p* ∧ *IN* (*vec* (*0*::*nat*)) (*interior p*) ∧ *IN f* (*face_set* (*hypermap1_of_fanx* (*vec* (*0*::*nat*), *vertices p*, *edges p*))) ∧ *facet_of f1 p* ∧ *dartset_leads_into_fan* (*vec* (*0*::*nat*)) (*vertices p*) (*edges p*) *f* = *fchanged f1* ∧ *IN* (*?e*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*) *f* ⟶ *IN* (*INSERT* (*pr2 ?e*) (*INSERT* (*pr3 ?e*) *EMPTY*)) (*EDGES0_FAN p f1*)

thm Cfyxfty.SUBSET_EDGES0_FAN:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) (*f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) *f1*::(*real*, *3*) *cart* ⇒ *bool*. *bounded p* ∧ *polyhedron p* ∧ *IN* (*vec* (*0*::*nat*)) (*interior p*) ∧ *IN f* (*face_set* (*hypermap1_of_fanx* (*vec* (*0*::*nat*), *vertices p*, *edges p*))) ∧ *facet_of f1 p* ∧ *dartset_leads_into_fan* (*vec* (*0*::*nat*)) (*vertices p*) (*edges p*) *f* = *fchanged f1* ⟶ *SUBSET* (*GSPEC* (λ*GEN%PVAR%2025*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart* × (*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2025* (*IN e f*) (*INSERT* (*pr2 e*) (*INSERT* (*pr3 e*) *EMPTY*)))) (*EDGES0_FAN p f1*)

thm Cfyxfty.EDGES0_SUBSET_FAN:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) f1::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ IN f (face_set (hypermap1_of_fanx (vec (0::nat), vertices p, edges p))) ∧ facet_of f1 p ∧ fchanged f1 = dartset_leads_into_fan (vec (0::nat)) (vertices p) (edges p) f ⟶ SUBSET (EDGES0_FAN p f1) (GSPEC (λGEN%PVAR%2028::(real, 3) cart ⇒ bool. ∃ e::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%2028 (IN e f) (INSERT (pr2 e) (INSERT (pr3 e) EMPTY))))

thm Cfyxfty.CFYXFTY0:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ⟶ EDGES0_FAN p f = edges f

thm Cfyxfty.CFYXFTY1:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool) f1::(real, 3) cart ⇒ bool. bounded p ∧ polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ IN f (face_set (hypermap1_of_fanx (vec (0::nat), vertices p, edges p))) ∧ facet_of f1 p ∧ fchanged f1 = dartset_leads_into_fan (vec (0::nat)) (vertices p) (edges p) f ⟶ EDGES0_FAN p f1 = GSPEC (λGEN%PVAR%2029::(real, 3) cart ⇒ bool. ∃ e::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart. SETSPEC GEN%PVAR%2029 (IN e f) (INSERT (pr2 e) (INSERT (pr3 e) EMPTY)))

thm Ysskqoy.pack_ineq_def_a:

pack_ineq_def_a = ((∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. ineq [(DECIMAL (30::nat) (10::nat), x1, DECIMAL (640::nat) (10::nat)), (1::real, x2, 1::real), (1::real, x3, 1::real), (1::real, x4, 1::real), (1::real, x5, 1::real), (1::real, x6, 1::real)] (DECIMAL (591::nat) (1000::nat) − DECIMAL (331::nat) (10000::nat) ∗ x1 + (DECIMAL (506::nat) (1000::nat) ∗ lfun (1::real) + DECIMAL (10::nat) (10::nat)) < real_of_nat (2::nat) ∗ pi − real_of_nat (2::nat) ∗ asn797k x1 x2 x3 x4 x5 x6)) ∧ (∀ h::real. ineq [(DECIMAL (10::nat) (10::nat), h, DECIMAL (10::nat) (10::nat))] (DECIMAL (591::nat) (1000::nat) − DECIMAL (331::nat) (10000::nat) ∗ real_of_nat (64::nat) + (DECIMAL (506::nat) (1000::nat) ∗ lfun (1::real) + DECIMAL (10::nat) (10::nat)) < (0::real))) ∧ (∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. ineq [(real_of_nat (4::nat), x1, (DECIMAL (252::nat) (100::nat))^2), (real_of_nat (4::nat), x2, real_of_nat (4::nat)), ((real_of_nat (2::nat) ∗ h0)^2, x3, (real_of_nat (2::nat) ∗ h0)^2), (1::real, x4, 1::real), (1::real, x5, 1::real), (1::real, x6, 1::real)] (acs_sqrt_x1_d4 x1 x2 x3 x4 x5 x6 − pi / real_of_nat (6::nat) + DECIMAL (797::nat) (1000::nat) < arclength_x_123 x1 x2 x3 x4 x5 x6)) ∧ (∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. ineq [(DECIMAL (10::nat) (10::nat), x1, DECIMAL (126::nat) (100::nat)), (DECIMAL (30::nat) (10::nat), x2, DECIMAL (340::nat) (10::nat)), (1::real, x3, 1::real), (1::real, x4, 1::real), (1::real, x5, 1::real), (1::real, x6, 1::real)] (DECIMAL

(*591*::*nat*) (*1000*::*nat*) − *DECIMAL* (*331*::*nat*) (*10000*::*nat*) ∗ *x2* + *DECIMAL* (*506*::*nat*) (*1000*::*nat*) ∗ *lfun_y1 x1 x2 x3 x4 x5 x6* < *real_of_nat* (*2*::*nat*) ∗ *pi* − *real_of_nat* (*2*::*nat*) ∗ *asnFnhk x1 x2 x3 x4 x5 x6*)) ∧ (∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *ineq* [(*DECIMAL* (*10*::*nat*) (*10*::*nat*), *x1*, *DECIMAL* (*126*::*nat*) (*100*::*nat*)), (*1*::*real*, *x2*, *1*::*real*), (*1*::*real*, *x3*, *1*::*real*), (*1*::*real*, *x4*, *1*::*real*), (*1*::*real*, *x5*, *1*::*real*), (*1*::*real*, *x6*, *1*::*real*)] (*DECIMAL* (*591*::*nat*) (*1000*::*nat*) − *DECIMAL* (*331*::*nat*) (*10000*::*nat*) ∗ *real_of_nat* (*34*::*nat*) + *DECIMAL* (*506*::*nat*) (*1000*::*nat*) ∗ *lfun_y1 x1 x2 x3 x4 x5 x6* < (*0*::*real*))))

thm DEF_selectd:

selectd = (λ*_6610896*::*?'a*::*type* ⇒ *bool*. *If* (∃ *r*::*?'a*::*type*. *_6610896 r*) (*Eps _6610896*))

thm Ysskqoy.selectd:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *d*::*?'a*::*type*. *selectd P d* = (*if* ∃ *r*::*?'a*::*type*. *P r then Eps P else d*)

thm Ysskqoy.selectd_cases:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *d*::*?'a*::*type*. *P* (*selectd P d*) ∨ *selectd P d* = *d*

thm Ysskqoy.selectd_exists:

∀ (*P*::*?'a*::*type* ⇒ *bool*) *d*::*?'a*::*type*. (∃ *r*::*?'a*::*type*. *P r*) ⟶ *P* (*selectd P d*)

thm Ysskqoy.DOT_COMPLEX:

∀ (*x*::*real*) (*y*::*real*) (*x'*::*real*) *y'*::*real*. *dot* (*complex* (*x*, *y*)) (*complex* (*x'*, *y'*)) = *x* ∗ *x'* + *y* ∗ *y'*

thm Ysskqoy.DOT_RE:

∀ (*z1*::(*real*, *2*) *cart*) *z2*::(*real*, *2*) *cart*. *dot z1 z2* = *Re* (*complex_mul z1* (*cnj z2*))

thm Ysskqoy.ARG_CNJ:

∀ (*z*::(*real*, *2*) *cart*) *w*::(*real*, *2*) *cart*. *w* ≠ *Cx* (*0*::*real*) ⟶ *Arg* (*complex_div z w*) = *Arg* (*complex_mul z* (*cnj w*))

thm Ysskqoy.ARG_0_DIV:

∀ (*u*::(*real*, *2*) *cart*) *v*::(*real*, *2*) *cart*. (*complex_div u v* = *Cx* (*0*::*real*)) = (*u* = *Cx* (*0*::*real*) ∨ *v* = *Cx* (*0*::*real*))

thm Ysskqoy.CARD_UNION_EQ:

∀ (*s*::*?'a*::*type* ⇒ *bool*) (*t*::*?'a*::*type* ⇒ *bool*) *u*::*?'a*::*type* ⇒ *bool*. *FINITE u* ∧ *HOL_Light_Import.INTER s t* = *EMPTY* ∧ *HOL_Light_Import.UNION s t* = *u* ⟶ *CARD s* + *CARD t* = *CARD u*

thm Ysskqoy.INJ_SURJ:

$\forall$ (*a*::*?'b*::*type* $\Rightarrow$ *bool*) (*b*::*?'a*::*type* $\Rightarrow$ *bool*) *f*::*?'b*::*type* $\Rightarrow$ *?'a*::*type*. *FINITE* *a* $\wedge$ *FINITE b* $\wedge$ *CARD a* = *CARD b* $\longrightarrow$ *INJ f a b* $\longrightarrow$ *SURJ f a b*

thm Ysskqoy.INJ_IFF_SURJ:

$\forall$ (*a*::*?'b*::*type* $\Rightarrow$ *bool*) (*b*::*?'a*::*type* $\Rightarrow$ *bool*) *f*::*?'b*::*type* $\Rightarrow$ *?'a*::*type*. *FINITE* *a* $\wedge$ *FINITE b* $\wedge$ *CARD a* = *CARD b* $\longrightarrow$ *INJ f a b* = *SURJ f a b*

thm Ysskqoy.NORM1_NZ:

$\forall$ *a*::(*real*, *?'a*::*type*) *cart*. *vector_norm a* = (*1*::*real*) $\longrightarrow$ *a* $\neq$ *vec* (*0*::*nat*)

thm DEF_normalize:

normalize = (λ*_6612294*::(*real*, *?'a*::*type*) *cart*. % (*inverse_class.inverse* (*vector_norm* *_6612294*)) *_6612294*)

thm Ysskqoy.normalize:

$\forall$ *v*::(*real*, *?'a*::*type*) *cart*. *normalize v* = % (*inverse_class.inverse* (*vector_norm* *v*)) *v*

thm Ysskqoy.norm_normalize:

$\forall$ *v*::(*real*, *?'a*::*type*) *cart*. *v* $\neq$ *vec* (*0*::*nat*) $\longrightarrow$ *vector_norm* (*normalize v*) = (*1*::*real*)

thm Ysskqoy.NZ_IMP_NORM1:

$\forall$ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::*real*. *a* $\neq$ *vec* (*0*::*nat*) $\longrightarrow$ ($\exists$ (*a'*::(*real*, *?'a*::*type*) *cart*) *b'*::*real*. *vector_norm a'* = (*1*::*real*) $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. (*dot a x* $\leq$ *b*) = (*dot a' x* $\leq$ *b'*)) $\wedge$ ($\forall$ *x*::(*real*, *?'a*::*type*) *cart*. (*dot a x* = *b*) = (*dot a'* *x* = *b'*)))

thm Ysskqoy.RE_CEXP_CX:

$\forall$ *x*::*real*. *Re* (*cexp* (*complex_mul ii* (*Cx x*))) = *cos x*

thm Ysskqoy.RE_NORM_1:

$\forall$ *z*::(*real*, *2*) *cart*. *vector_norm z* = (*1*::*real*) $\longrightarrow$ *Re z* = *cos* (*Arg z*)

thm Ysskqoy.COS_ARG_VECTOR_ANGLE:

$\forall$ (*u*::(*real*, *2*) *cart*) *v*::(*real*, *2*) *cart*. *u* $\neq$ *Cx* (*0*::*real*) $\wedge$ *v* $\neq$ *Cx* (*0*::*real*) $\longrightarrow$ *cos* (*Arg* (*complex_div u v*)) = *cos* (*vector_angle u v*)

thm Ysskqoy.SEC_DOT:

$\forall$ (*u*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*r*::*real*) *psi*::*real*. (*0*::*real*) < *r* $\wedge$ (*0*::*real*) $\leq$ *psi* $\wedge$ *psi* < *pi* / *real_of_nat* (*2*::*nat*) $\wedge$ *vector_norm v* = *r* $*$ *inverse_class.inverse* (*cos psi*) $\wedge$ *vector_norm u* = *r* $\wedge$ *cos* (*vector_angle u v*) = *cos psi* $\longrightarrow$ *dot u* (*vector_sub v u*) = (*0*::*real*)

thm Ysskqoy.arclength2:

$\forall$ *h*::*real*. (*1*::*real*) $\leq$ *h* $\wedge$ *h* $\leq$ *h0* $\longrightarrow$ *arclength* (*real_of_nat* (*2*::*nat*)) (*real_of_nat* (*2*::*nat*) $*$ *h*) (*real_of_nat* (*2*::*nat*)) = *acs* (*h* / *real_of_nat* (*2*::*nat*))

thm Ysskqoy.yssk_reduction:

(∀ (*a1*::*real*) (*a2*::*real*) (*b1*::*real*) *b2*::*real*. *real_of_nat* (*2*::*nat*) ≤ *a1* ∧ *a1* ≤ *a2* ∧ *a2* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ∧ *real_of_nat* (*2*::*nat*) ≤ *b1* ∧ *b1* ≤ *b2* ∧ *b2* ≤ *real_of_nat* (*2*::*nat*) ∗ *h0* ⟶ (*0*::*real*) ≤ *arclength a2 b2* (*real_of_nat* (*2*::*nat*)) − *arclength a1 b2* (*real_of_nat* (*2*::*nat*)) − *arclength a2 b1* (*real_of_nat* (*2*::*nat*)) + *arclength a1 b1* (*real_of_nat* (*2*::*nat*))) ⟶ (∀ (*h*::*real*) *h*′::*real*. (*1*::*real*) ≤ *h* ∧ *h* ≤ *h0* ∧ (*1*::*real*) ≤ *h*′ ∧ *h*′ ≤ *h0* ⟶ *acs* (*h* / *real_of_nat* (*2*::*nat*)) + (*acs* (*h*′ / *real_of_nat* (*2*::*nat*)) − *pi* / *real_of_nat* (*3*::*nat*)) ≤ *arclength* (*real_of_nat* (*2*::*nat*) ∗ *h*) (*real_of_nat* (*2*::*nat*) ∗ *h*′) (*real_of_nat* (*2*::*nat*)))

thm Ysskqoy.TRI_UPS_X_STRICT_POS:

∀ (*a*::*real*) (*b*::*real*) *c*::*real*. (*0*::*real*) < *a* ∧ (*0*::*real*) < *b* ∧ (*0*::*real*) ≤ *c* ∧ *c* < *a* + *b* ∧ *a* < *b* + *c* ∧ *b* < *c* + *a* ⟶ (*0*::*real*) < *ups_x* (*a* ∗ *a*) (*b* ∗ *b*) (*c* ∗ *c*)

thm Ysskqoy.ups_x_pos:

∀ (*a*::*real*) *b*::*real*. *real_of_nat* (*2*::*nat*) ≤ *a* ∧ *a* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *b* ∧ *b* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ⟶ (*0*::*real*) < *ups_x* (a^2) (b^2) (*real_of_nat* (*4*::*nat*))

thm Ysskqoy.arc_derivative:

∀ (*a*::*real*) *b*::*real*. (*real_of_nat* (*2*::*nat*) ≤ *a* ∧ *a* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*)) ∧ *real_of_nat* (*2*::*nat*) ≤ *b* ∧ *b* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ⟶ *has_real_derivative* (λ*x*::*real*. *arclength x b* (*real_of_nat* (*2*::*nat*))) (− (*real_of_nat* (*4*::*nat*) + (a^2 − b^2)) / (*a* ∗ *sqrt* (*ups_x* (a^2) (b^2) (*real_of_nat* (*4*::*nat*))))) (*within* (*atreal a*) (*closed_real_interval* [(*real_of_nat* (*2*::*nat*), *DECIMAL* (*252*::*nat*) (*100*::*nat*))]))

thm Ysskqoy.arc_derivative2:

∀ (*a*::*real*) *b*::*real*. (*real_of_nat* (*2*::*nat*) ≤ *a* ∧ *a* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*)) ∧ *real_of_nat* (*2*::*nat*) ≤ *b* ∧ *b* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ⟶ *has_real_derivative* (λ*x*::*real*. − (*real_of_nat* (*4*::*nat*) + (a^2 − x^2)) / (*a* ∗ *sqrt* (*ups_x* (a^2) (x^2) (*real_of_nat* (*4*::*nat*))))) (*real_of_nat* (*32*::*nat*) ∗ (*a* ∗ (*b* / (*sqrt* (*ups_x* (a^2) (b^2) (*real_of_nat* (*4*::*nat*))))$^{3::nat}$))) (*within* (*atreal b*) (*closed_real_interval* [(*real_of_nat* (*2*::*nat*), *DECIMAL* (*252*::*nat*) (*100*::*nat*))]))

thm Ysskqoy.arc_length2_increasing:

∀ (*a*::*real*) (*b1*::*real*) *b2*::*real*. (*real_of_nat* (*2*::*nat*) ≤ *a* ∧ *a* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*)) ∧ *real_of_nat* (*2*::*nat*) ≤ *b1* ∧ *b1* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *b2* ∧ *b2* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *b1* ≤ *b2* ⟶ *LET* (λ*fa*::*real* ⇒ *real*. *LET_END* (*fa b1* ≤ *fa b2*)) (λ*x*::*real*. − (*real_of_nat* (*4*::*nat*) + (a^2 − x^2)) / (*a* ∗ *sqrt* (*ups_x* (a^2) (x^2) (*real_of_nat* (*4*::*nat*)))))

thm Ysskqoy.arc_length1_increasing:

∀ (*a1*::*real*) (*a2*::*real*) (*b1*::*real*) *b2*::*real*. *real_of_nat* (*2*::*nat*) ≤ *a1* ∧ *a1* ≤ *a2* ∧ *a2* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *b1* ∧

$b1 \le b2 \wedge b2 \le DECIMAL\ (252::nat)\ (100::nat) \longrightarrow LET\ (\lambda f::real \Rightarrow real.\ LET\_END\ (f\ a1 \le f\ a2))\ (\lambda x::real.\ arclength\ x\ b2\ (real\_of\_nat\ (2::nat)) - arclength\ x\ b1\ (real\_of\_nat\ (2::nat)))$

thm Ysskqoy.YSSKQOY:

$\forall (h::real)\ h'::real.\ (1::real) \le h \wedge h \le h0 \wedge (1::real) \le h' \wedge h' \le h0 \longrightarrow acs\ (h\ /\ real\_of\_nat\ (2::nat)) + (acs\ (h'\ /\ real\_of\_nat\ (2::nat)) - pi\ /\ real\_of\_nat\ (3::nat)) \le arclength\ (real\_of\_nat\ (2::nat) * h)\ (real\_of\_nat\ (2::nat) * h')\ (real\_of\_nat\ (2::nat))$

thm Counting_spheres.fat_lemma1:

$\forall S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge SUBSET\ S\ ball\_annulus \longrightarrow FINITE\ S$

thm Counting_spheres.ckq_in_ball_annulus:

$\forall v::(real,\ 3)\ cart.\ IN\ v\ ball\_annulus = (real\_of\_nat\ (2::nat) \le vector\_norm\ v \wedge vector\_norm\ v \le real\_of\_nat\ (2::nat) * h0 \wedge v \ne vec\ (0::nat))$

thm Counting_spheres.lemma:

$\forall (a2::real)\ (b::real)\ c::real.\ (0::real) < a2 \wedge (0::real) < c \wedge (\forall t::real.\ (0::real) \le t \wedge t < c \longrightarrow a2 * t \le b) \longrightarrow (\forall t::real.\ (0::real) \le t \wedge t \le c \longrightarrow a2 * t \le b)$

thm Counting_spheres.eus1:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ c::(real,\ 2)\ cart \Rightarrow bool.\ polyhedron\ P \wedge facet\_of\ c\ P \longrightarrow (\exists (a::(real,\ 2)\ cart)\ b::real.\ vector\_norm\ a = (1::real) \wedge (\forall r::real.\ (0::real) < r \wedge (\forall p::(real,\ 2)\ cart.\ vector\_norm\ p < r \longrightarrow P\ p) \longrightarrow r \le b) \wedge SUBSET\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2031::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2031\ (dot\ a\ x \le b)\ x)) \wedge c = HOL\_Light\_Import.INTER\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2032::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2032\ (dot\ a\ x = b)\ x)))$

thm Counting_spheres.facet_rep_uniq:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ (a::(real,\ 2)\ cart)\ (b1::real)\ b2::real.\ polyhedron\ P \wedge facet\_of\ (?c1.0::(real,\ 2)\ cart \Rightarrow bool)\ P \wedge facet\_of\ (?c2.0::(real,\ 2)\ cart \Rightarrow bool)\ P \wedge SUBSET\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2033::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2033\ (dot\ a\ x \le b1)\ x)) \wedge SUBSET\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2034::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2034\ (dot\ a\ x \le b2)\ x)) \wedge ?c1.0 = HOL\_Light\_Import.INTER\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2035::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2035\ (dot\ a\ x = b1)\ x)) \wedge ?c2.0 = HOL\_Light\_Import.INTER\ P\ (GSPEC\ (\lambda GEN\%PVAR\%2036::(real,\ 2)\ cart.\ \exists x::(real,\ 2)\ cart.\ SETSPEC\ GEN\%PVAR\%2036\ (dot\ a\ x = b2)\ x)) \longrightarrow b1 = b2 \wedge ?c1.0 = ?c2.0$

thm Counting_spheres.facet_rep_spec:

$\exists (a::((real,\ 2)\ cart \Rightarrow bool) \Rightarrow ((real,\ 2)\ cart \Rightarrow bool) \Rightarrow (real,\ 2)\ cart)\ b::((real,\ 2)\ cart \Rightarrow bool) \Rightarrow ((real,\ 2)\ cart \Rightarrow bool) \Rightarrow real.\ \forall (P::(real,\ 2)\ cart$

$\Rightarrow$ *bool*) *c*::(*real*, *2*) *cart* $\Rightarrow$ *bool*. *polyhedron P* $\wedge$ *facet_of c P* $\longrightarrow$ *vector_norm* (*a P c*) = (*1*::*real*) $\wedge$ ($\forall$ *r*::*real*. (*0*::*real*) < *r* $\wedge$ ($\forall$ *p*::(*real*, *2*) *cart*. *vector_norm p* < *r* $\longrightarrow$ *P p*) $\longrightarrow$ *r* $\leq$ *b P c*) $\wedge$ *SUBSET P* (*GSPEC* (λ*GEN%PVAR%2037*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2037* (*dot* (*a P c*) *x* $\leq$ *b P c*) *x*)) $\wedge$ *c* = *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2038*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2038* (*dot* (*a P c*) *x* = *b P c*) *x*))

thm DEF_facet_rep_a:

facet_rep_a = (*SOME a*::*nat* $\Rightarrow$ ((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ ((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ (*real*, *2*) *cart*. $\forall$ *_6613376*::*nat*. $\exists$ *b*::((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ ((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *real*. $\forall$ (*P*::(*real*, *2*) *cart* $\Rightarrow$ *bool*) *c*::(*real*, *2*) *cart* $\Rightarrow$ *bool*. *polyhedron P* $\wedge$ *facet_of c P* $\longrightarrow$ *vector_norm* (*a _6613376 P c*) = (*1*::*real*) $\wedge$ ($\forall$ *r*::*real*. (*0*::*real*) < *r* $\wedge$ ($\forall$ *p*::(*real*, *2*) *cart*. *vector_norm p* < *r* $\longrightarrow$ *P p*) $\longrightarrow$ *r* $\leq$ *b P c*) $\wedge$ *SUBSET P* (*GSPEC* (λ*GEN%PVAR%2037*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2037* (*dot* (*a _6613376 P c*) *x* $\leq$ *b P c*) *x*)) $\wedge$ *c* = *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2038*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2038* (*dot* (*a _6613376 P c*) *x* = *b P c*) *x*))) (*141*::*nat*)

thm DEF_facet_rep_b:

facet_rep_b = (*SOME b*::*nat* $\Rightarrow$ ((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ ((*real*, *2*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *real*. $\forall$ (*_6613377*::*nat*) (*P*::(*real*, *2*) *cart* $\Rightarrow$ *bool*) *c*::(*real*, *2*) *cart* $\Rightarrow$ *bool*. *polyhedron P* $\wedge$ *facet_of c P* $\longrightarrow$ *vector_norm* (*facet_rep_a P c*) = (*1*::*real*) $\wedge$ ($\forall$ *r*::*real*. (*0*::*real*) < *r* $\wedge$ ($\forall$ *p*::(*real*, *2*) *cart*. *vector_norm p* < *r* $\longrightarrow$ *P p*) $\longrightarrow$ *r* $\leq$ *b _6613377 P c*) $\wedge$ *SUBSET P* (*GSPEC* (λ*GEN%PVAR%2037*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2037* (*dot* (*facet_rep_a P c*) *x* $\leq$ *b _6613377 P c*) *x*)) $\wedge$ *c* = *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2038*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2038* (*dot* (*facet_rep_a P c*) *x* = *b _6613377 P c*) *x*))) (*142*::*nat*)

thm Counting_spheres.facet_rep_def:

$\forall$ (*P*::(*real*, *2*) *cart* $\Rightarrow$ *bool*) *c*::(*real*, *2*) *cart* $\Rightarrow$ *bool*. *polyhedron P* $\wedge$ *facet_of c P* $\longrightarrow$ *vector_norm* (*facet_rep_a P c*) = (*1*::*real*) $\wedge$ ($\forall$ *r*::*real*. (*0*::*real*) < *r* $\wedge$ ($\forall$ *p*::(*real*, *2*) *cart*. *vector_norm p* < *r* $\longrightarrow$ *P p*) $\longrightarrow$ *r* $\leq$ *facet_rep_b P c*) $\wedge$ *SUBSET P* (*GSPEC* (λ*GEN%PVAR%2037*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2037* (*dot* (*facet_rep_a P c*) *x* $\leq$ *facet_rep_b P c*) *x*)) $\wedge$ *c* = *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2038*::(*real*, *2*) *cart*. $\exists$ *x*::(*real*, *2*) *cart*. *SETSPEC GEN%PVAR%2038* (*dot* (*facet_rep_a P c*) *x* = *facet_rep_b P c*) *x*))

thm Counting_spheres.facet_rep_uniq_c:

$\forall$ (*P*::(*real*, *2*) *cart* $\Rightarrow$ *bool*) (*c1*::(*real*, *2*) *cart* $\Rightarrow$ *bool*) *c2*::(*real*, *2*) *cart* $\Rightarrow$ *bool*. *polyhedron P* $\wedge$ *facet_of c1 P* $\wedge$ *facet_of c2 P* $\wedge$ *facet_rep_a P c1* = *facet_rep_a P c2* $\longrightarrow$ *c1* = *c2*

thm Counting_spheres.norm1_cauchy_eq:

∀(x::(real, ?'a::type) cart) y::(real, ?'a::type) cart. vector_norm x = (1::real) ∧ vector_norm y = (1::real) ∧ dot x y = (1::real) ⟶ x = y

thm Counting_spheres.facet_rep_in_facet:

∀(P::(real, 2) cart ⇒ bool) (c1::(real, 2) cart ⇒ bool) (c2::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ facet_of c1 P ∧ facet_of c2 P ∧ (0::real) < r ∧ (∀p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ facet_rep_b P c1 ≤ dot (facet_rep_a P c1) (% r (facet_rep_a P c2)) ⟶ c1 = c2

thm Counting_spheres.facet_rep_refl:

∀(P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ facet_of c P ∧ (0::real) < r ∧ (∀p::(real, 2) cart. vector_norm p < r ⟶ P p) ⟶ dot (facet_rep_a P c) (% r (facet_rep_a P c)) ≤ facet_rep_b P c

thm Counting_spheres.DOT_EQ_IMP_INEQ_LEMMA:

∀(a::(real, ?'a::type) cart) (b::real) (a'::(real, ?'a::type) cart) b'::real. (∀x::(real, ?'a::type) cart. (dot a x = b) = (dot a' x = b')) ∧ (0::real) < b ∧ (0::real) < b' ⟶ (∀x::(real, ?'a::type) cart. dot a x ≠ (0::real) ⟶ (dot a x ≤ b) = (dot a' x ≤ b'))

thm Counting_spheres.DOT_EQ_IMP_INEQ:

∀(a::(real, ?'a::type) cart) (b::real) (a'::(real, ?'a::type) cart) b'::real. (∀x::(real, ?'a::type) cart. (dot a x = b) = (dot a' x = b')) ∧ (0::real) ≤ b ∧ (0::real) < b' ⟶ (∀x::(real, ?'a::type) cart. (dot a x ≤ b) = (dot a' x ≤ b'))

thm Counting_spheres.affine_facet_hyper:

∀(P::(real, ?'a::type) cart ⇒ bool) (c::(real, ?'a::type) cart ⇒ bool) (a::(real, ?'a::type) cart) b::real. facet_of c P ∧ polyhedron P ∧ hull affine P = HOL_Light_Import.UNIV ∧ a ≠ vec (0::nat) ∧ HOL_Light_Import.INTER P (GSPEC (λGEN%PVAR%2039::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2039 (dot a x = b) x)) = c ⟶ hull affine c = GSPEC (λGEN%PVAR%2040::(real, ?'a::type) cart. ∃x::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2040 (dot a x = b) x)

thm Counting_spheres.POLYHEDRON_MEMBER:

∀(P::(real, 2) cart ⇒ bool) (r::real) x::(real, 2) cart. polyhedron P ∧ (0::real) < r ∧ (∀p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ (∀c::(real, 2) cart ⇒ bool. facet_of c P ⟶ dot (facet_rep_a P c) x ≤ facet_rep_b P c) ⟶ P x

thm Counting_spheres.facet_rep_in_poly:

∀(P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ facet_of c P ∧ (0::real) < r ∧ (∀p::(real, 2) cart. vector_norm p < r ⟶ P p) ⟶ P (% r (facet_rep_a P c))

thm Counting_spheres.facet_rep_not_in_facet:

∀(P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) (c'::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ facet_of c P ∧ facet_of c' P ∧ (0::real) < r ∧

(∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ c′ (% r (facet_rep_a P c)) ⟶ c′ = c

thm Counting_spheres.facet_arg_lt_pi:

∀ (P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ bounded P ∧ facet_of c P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ⟶ (∃ c′::(real, 2) cart ⇒ bool. facet_of c′ P ∧ (0::real) < Arg (complex_div (facet_rep_a P c′) (facet_rep_a P c)) ∧ Arg (complex_div (facet_rep_a P c′) (facet_rep_a P c)) < pi)

thm Counting_spheres.eus_cos:

∀ (phi::real) psi::real. (0::real) ≤ psi ∧ psi ≤ phi ∧ phi ≤ real_of_nat (2::nat) ∗ pi − psi ⟶ cos phi ≤ cos psi

thm Counting_spheres.insert_v:

∀ (P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) (c′::(real, 2) cart ⇒ bool) (r::real) (v::(real, 2) cart) psi::real. polyhedron P ∧ facet_of c P ∧ facet_of c′ P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ Arg (complex_div v (facet_rep_a P c)) = psi ∧ (0::real) < psi ∧ psi < pi / real_of_nat (2::nat) ∧ Arg (complex_div (facet_rep_a P c′) (facet_rep_a P c)) = real_of_nat (2::nat) ∗ psi ∧ (∀ c′′::(real, 2) cart ⇒ bool. facet_of c′′ P ∧ Arg (complex_div (facet_rep_a P c′′) (facet_rep_a P c)) < real_of_nat (2::nat) ∗ psi ⟶ c′′ = c) ∧ vector_norm v = r / cos psi ⟶ P v

thm Counting_spheres.facet_rep_a_uniq:

∀ (P::(real, 2) cart ⇒ bool) (c1::(real, 2) cart ⇒ bool) (c2::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ facet_of c1 P ∧ facet_of c2 P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ (∃ s>0::real. facet_rep_a P c1 = % s (facet_rep_a P c2)) ⟶ c1 = c2

thm DEF_poly_sort_fn:

poly_sort_fn = (λ(_6627016::(real, 2) cart ⇒ bool) (_6627017::(real, 2) cart) (_6627018::(real, 2) cart ⇒ bool) _6627019::(real, 2) cart ⇒ bool. facet_of _6627018 _6627016 ∧ facet_of _6627019 _6627016 ∧ Arg (complex_div (facet_rep_a _6627016 _6627018) _6627017) ≤ Arg (complex_div (facet_rep_a _6627016 _6627019) _6627017))

thm Counting_spheres.poly_sort_fn:

∀ (c1::(real, 2) cart ⇒ bool) (P::(real, 2) cart ⇒ bool) (c2::(real, 2) cart ⇒ bool) u::(real, 2) cart. poly_sort_fn P u c1 c2 = (facet_of c1 P ∧ facet_of c2 P ∧ Arg (complex_div (facet_rep_a P c1) u) ≤ Arg (complex_div (facet_rep_a P c2) u))

thm Counting_spheres.poly_sort_antisym:

∀ (P::(real, 2) cart ⇒ bool) (u::(real, 2) cart) (c1::(real, 2) cart ⇒ bool) (c2::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ (0::real) < r ∧ (∀ p::(real, *2) cart. vector_norm p < r ⟶ P p) ∧ poly_sort_fn P u c1 c2 ∧ poly_sort_fn P u c2 c1 ∧ u ≠ Cx (0::real) ⟶ c1 = c2*

thm Counting_spheres.poly_sort_trans:

∀ (P::(real, 2) cart ⇒ bool) (u::(real, 2) cart) (c1::(real, 2) cart ⇒ bool) (c2::(real, 2) cart ⇒ bool) (c3::(real, 2) cart ⇒ bool) r::real. polyhedron P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ u ≠ Cx (0::real) ∧ poly_sort_fn P u c1 c2 ∧ poly_sort_fn P u c2 c3 ⟶ poly_sort_fn P u c1 c3

thm Counting_spheres.POLY_SORT_LEMMA:

∀ (P::(real, 2) cart ⇒ bool) (n::nat) (s::((real, 2) cart ⇒ bool) ⇒ bool) (r::real) u::(real, 2) cart. s = GSPEC (λGEN%PVAR%2047::(real, 2) cart ⇒ bool. ∃ c::(real, 2) cart ⇒ bool. SETSPEC GEN%PVAR%2047 (facet_of c P) c) ∧ polyhedron P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ u ≠ Cx (0::real) ∧ HAS_SIZE s n ⟶ (∃ f::nat ⇒ (real, 2) cart ⇒ bool. s = IMAGE f (dotdot (1::nat) n) ∧ (∀ (j::nat) k::nat. IN j (dotdot (1::nat) n) ∧ IN k (dotdot (1::nat) n) ∧ j < k ⟶ ¬ poly_sort_fn P u (f k) (f j)))

thm Counting_spheres.POLY_SORT:

∀ (P::(real, 2) cart ⇒ bool) (n::nat) (s::((real, 2) cart ⇒ bool) ⇒ bool) (r::real) u::(real, 2) cart. s = GSPEC (λGEN%PVAR%2048::(real, 2) cart ⇒ bool. ∃ c::(real, 2) cart ⇒ bool. SETSPEC GEN%PVAR%2048 (facet_of c P) c) ∧ polyhedron P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ u ≠ Cx (0::real) ∧ HAS_SIZE s n ⟶ (∃ f::nat ⇒ (real, 2) cart ⇒ bool. s = IMAGE f (dotdot (1::nat) n) ∧ (∀ (j::nat) k::nat. IN j (dotdot (1::nat) n) ∧ IN k (dotdot (1::nat) n) ∧ j < k ⟶ Arg (complex_div (facet_rep_a P (f j)) u) < Arg (complex_div (facet_rep_a P (f k)) u)))

thm Counting_spheres.POLY_SORT_BIJ:

∀ (P::(real, 2) cart ⇒ bool) (n::nat) (s::((real, 2) cart ⇒ bool) ⇒ bool) (r::real) u::(real, 2) cart. s = GSPEC (λGEN%PVAR%2049::(real, 2) cart ⇒ bool. ∃ c::(real, 2) cart ⇒ bool. SETSPEC GEN%PVAR%2049 (facet_of c P) c) ∧ polyhedron P ∧ (0::real) < r ∧ (∀ p::(real, 2) cart. vector_norm p < r ⟶ P p) ∧ u ≠ Cx (0::real) ∧ HAS_SIZE s n ⟶ (∃ f::nat ⇒ (real, 2) cart ⇒ bool. s = IMAGE f (dotdot (1::nat) n) ∧ BIJ f (dotdot (1::nat) n) s ∧ (∀ (j::nat) k::nat. IN j (dotdot (1::nat) n) ∧ IN k (dotdot (1::nat) n) ∧ j < k ⟶ Arg (complex_div (facet_rep_a P (f j)) u) < Arg (complex_div (facet_rep_a P (f k)) u)))

thm Counting_spheres.facet_rep_nz:

∀ (P::(real, 2) cart ⇒ bool) c::(real, 2) cart ⇒ bool. polyhedron P ∧ facet_of c P ⟶ facet_rep_a P c ≠ Cx (0::real)

thm Counting_spheres.bisector_point_exists:

∀ (P::(real, 2) cart ⇒ bool) (c::(real, 2) cart ⇒ bool) (c′::(real, 2) cart ⇒ bool) r::real. ∃ v::(real, 2) cart. ∀ psi::real. polyhedron P ∧ facet_of c P ∧ facet_of

c' $P \wedge (0::real) < r \wedge (\forall p::(real,\ 2)\ cart.\ vector\_norm\ p < r \longrightarrow P\ p) \wedge psi = Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ (facet\_rep\_a\ P\ c))\ /\ real\_of\_nat\ (2::nat) \wedge (\forall c''::(real,\ 2)\ cart \Rightarrow bool.\ facet\_of\ c''\ P \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c'')\ (facet\_rep\_a\ P\ c)) < real\_of\_nat\ (2::nat) * psi \longrightarrow c'' = c) \wedge psi < pi\ /\ real\_of\_nat\ (2::nat) \wedge c' \neq c \longrightarrow P\ v \wedge vector\_norm\ v = r * inverse\_class.inverse\ (cos\ psi) \wedge Arg\ (complex\_div\ v\ (facet\_rep\_a\ P\ c)) = psi \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ v) = psi$

thm DEF_bisector_point:

$bisector\_point = (SOME\ v::nat \Rightarrow ((real,\ 2)\ cart \Rightarrow bool) \Rightarrow ((real,\ 2)\ cart \Rightarrow bool) \Rightarrow ((real,\ 2)\ cart \Rightarrow bool) \Rightarrow real \Rightarrow (real,\ 2)\ cart.\ \forall (\_6632282::nat)\ (P::(real,\ 2)\ cart \Rightarrow bool)\ (c::(real,\ 2)\ cart \Rightarrow bool)\ (c'::(real,\ 2)\ cart \Rightarrow bool)\ (r::real)\ psi::real.\ polyhedron\ P \wedge facet\_of\ c\ P \wedge facet\_of\ c'\ P \wedge (0::real) < r \wedge (\forall p::(real,\ 2)\ cart.\ vector\_norm\ p < r \longrightarrow P\ p) \wedge psi = Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ (facet\_rep\_a\ P\ c))\ /\ real\_of\_nat\ (2::nat) \wedge (\forall c''::(real,\ 2)\ cart \Rightarrow bool.\ facet\_of\ c''\ P \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c'')\ (facet\_rep\_a\ P\ c)) < real\_of\_nat\ (2::nat) * psi \longrightarrow c'' = c) \wedge psi < pi\ /\ real\_of\_nat\ (2::nat) \wedge c' \neq c \longrightarrow P\ (v\ \_6632282\ P\ c\ c'\ r) \wedge vector\_norm\ (v\ \_6632282\ P\ c\ c'\ r) = r * inverse\_class.inverse\ (cos\ psi) \wedge Arg\ (complex\_div\ (v\ \_6632282\ P\ c\ c'\ r)\ (facet\_rep\_a\ P\ c)) = psi \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ (v\ \_6632282\ P\ c\ c'\ r)) = psi)\ (143::nat)$

thm Counting_spheres.bisector_point:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ (c::(real,\ 2)\ cart \Rightarrow bool)\ (c'::(real,\ 2)\ cart \Rightarrow bool)\ (r::real)\ psi::real.\ polyhedron\ P \wedge facet\_of\ c\ P \wedge facet\_of\ c'\ P \wedge (0::real) < r \wedge (\forall p::(real,\ 2)\ cart.\ vector\_norm\ p < r \longrightarrow P\ p) \wedge psi = Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ (facet\_rep\_a\ P\ c))\ /\ real\_of\_nat\ (2::nat) \wedge (\forall c''::(real,\ 2)\ cart \Rightarrow bool.\ facet\_of\ c''\ P \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c'')\ (facet\_rep\_a\ P\ c)) < real\_of\_nat\ (2::nat) * psi \longrightarrow c'' = c) \wedge psi < pi\ /\ real\_of\_nat\ (2::nat) \wedge c' \neq c \longrightarrow P\ (bisector\_point\ P\ c\ c'\ r) \wedge vector\_norm\ (bisector\_point\ P\ c\ c'\ r) = r * inverse\_class.inverse\ (cos\ psi) \wedge Arg\ (complex\_div\ (bisector\_point\ P\ c\ c'\ r)\ (facet\_rep\_a\ P\ c)) = psi \wedge Arg\ (complex\_div\ (facet\_rep\_a\ P\ c')\ (bisector\_point\ P\ c\ c'\ r)) = psi$

thm Counting_spheres.RCONE_LINEAR_INVARIANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ (v::(real,\ ?'b::type)\ cart)\ a::real.\ linear\ f \wedge (\forall y::(real,\ ?'a::type)\ cart.\ \exists x::(real,\ ?'b::type)\ cart.\ f\ x = y) \wedge (\forall x::(real,\ ?'b::type)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \longrightarrow rcone\_gt\ (vec\ (0::nat))\ (f\ v)\ a = IMAGE\ f\ (rcone\_gt\ (vec\ (0::nat))\ v\ a)$

thm Counting_spheres.FCHANGED_LINEAR_INVARIANT:

$\forall (f::(real,\ ?'b::type)\ cart \Rightarrow (real,\ ?'a::type)\ cart)\ c::(real,\ ?'b::type)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall (x::(real,\ ?'b::type)\ cart)\ y::(real,\ ?'b::type)\ cart.\ f\ x = f\ y \longrightarrow x = y) \longrightarrow fchanged\ (IMAGE\ f\ c) = IMAGE\ f\ (fchanged\ c)$

thm DEF_solvec0:

solvec0 = (λ*_6634892*::(*real*, *3*) *cart* ⇒ *bool*. *LET* (λ*r*::*real*. *LET_END* (*real_of_nat* (*3*::*nat*) ∗ (*HOL_Light_Import*.*measure* (*HOL_Light_Import*.*INTER* *_6634892* (*normball* (*vec* (*0*::*nat*)) *r*)) / $r^{3::nat}$))) (*selectd* (*GSPEC* (λ*GEN%PVAR%2050*::*real*. ∃ *r*::*real*. *SETSPEC* *GEN%PVAR%2050* ((*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import*.*INTER* *_6634892* (*normball* (*vec* (*0*::*nat*)) *r*)) ∧ *radial_norm* *r* (*vec* (*0*::*nat*)) (*HOL_Light_Import*.*INTER* *_6634892* (*normball* (*vec* (*0*::*nat*)) *r*))) *r*)) (*0*::*real*)))

thm Counting_spheres.solvec0:

∀ *X*::(*real*, *3*) *cart* ⇒ *bool*. *solvec0* *X* = *LET* (λ*r*::*real*. *LET_END* (*real_of_nat* (*3*::*nat*) ∗ (*HOL_Light_Import*.*measure* (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*)) / $r^{3::nat}$))) (*selectd* (*GSPEC* (λ*GEN%PVAR%2050*::*real*. ∃ *r*::*real*. *SETSPEC* *GEN%PVAR%2050* ((*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*)) ∧ *radial_norm* *r* (*vec* (*0*::*nat*)) (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*))) *r*)) (*0*::*real*))

thm Counting_spheres.solvec0_sol:

∀ *X*::(*real*, *3*) *cart* ⇒ *bool*. *LET* (λ*P*::*real* ⇒ *bool*. *LET_END* ((∃ *r*::*real*. *P* *r*) ⟶ *solvec0* *X* = *sol* (*vec* (*0*::*nat*)) *X*)) (*GSPEC* (λ*GEN%PVAR%2051*::*real*. ∃ *r*::*real*. *SETSPEC* *GEN%PVAR%2051* ((*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*)) ∧ *radial_norm* *r* (*vec* (*0*::*nat*)) (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*))) *r*))

thm Counting_spheres.solvec0_d:

∀ *X*::(*real*, *3*) *cart* ⇒ *bool*. *LET* (λ*P*::*real* ⇒ *bool*. *LET_END* (¬ (∃ *r*::*real*. *P* *r*) ⟶ *solvec0* *X* = (*0*::*real*))) (*GSPEC* (λ*GEN%PVAR%2052*::*real*. ∃ *r*::*real*. *SETSPEC* *GEN%PVAR%2052* ((*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*)) ∧ *radial_norm* *r* (*vec* (*0*::*nat*)) (*HOL_Light_Import*.*INTER* *X* (*normball* (*vec* (*0*::*nat*)) *r*))) *r*))

thm Counting_spheres.BALL_LINEAR_INVARIANT:

∀ (*f*::(*real*, *?'a*::*type*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::*real*. *linear* *f* ∧ (∀ *x*::(*real*, *?'a*::*type*) *cart*. *vector_norm* (*f* *x*) = *vector_norm* *x*) ∧ (∀ *y*::(*real*, *?'a*::*type*) *cart*. ∃ *x*::(*real*, *?'a*::*type*) *cart*. *f* *x* = *y*) ⟶ *IMAGE* *f* (*ball* (*vec* (*0*::*nat*), *r*)) = *ball* (*vec* (*0*::*nat*), *r*)

thm Counting_spheres.cos_acs_pi6:

∀ *h*::*real*. (*1*::*real*) ≤ *h* ∧ *h* ≤ *h0* ⟶ *cos* (*acs* (*h* / *real_of_nat* (*2*::*nat*)) − *pi* / *real_of_nat* (*6*::*nat*)) = *h* ∗ (*sqrt3* / *DECIMAL* (*40*::*nat*) (*10*::*nat*)) + *sqrt* ((*1*::*real*) − (*h* / *real_of_nat* (*2*::*nat*))^2) / *real_of_nat* (*2*::*nat*)

thm Counting_spheres.regular_spherical_polygon_area_asnFnhk:

∀ (*h*::*real*) *k*::*nat*. (*3*::*nat*) ≤ *k* ∧ (*1*::*real*) ≤ *h* ∧ *h* ≤ *h0* ⟶ *regular_spherical_polygon_area* (*h* ∗ (*sqrt3* / *DECIMAL* (*40*::*nat*) (*10*::*nat*)) + *sqrt* ((*1*::*real*) − (*h* / *real_of_nat* (*2*::*nat*))^2) / *real_of_nat* (*2*::*nat*)) (*real_of_nat* *k*) = *real_of_nat* (*2*::*nat*) ∗ *pi* − *real_of_nat* (*2*::*nat*) ∗ *asnFnhk* *h* (*real_of_nat* *k*) (*1*::*real*) (*1*::*real*) (*1*::*real*) (*1*::*real*)

thm Counting_spheres.regular_spherical_polygon_area_797:

$\forall (h::?'a::type)\ k::nat.\ (3::nat) \leq k \longrightarrow regular\_spherical\_polygon\_area\ (cos\ (DECIMAL\ (797::nat)\ (1000::nat)))\ (real\_of\_nat\ k) = real\_of\_nat\ (2::nat) * pi - real\_of\_nat\ (2::nat) * (real\_of\_nat\ k * asn\ (cos\ (DECIMAL\ (797::nat)\ (1000::nat)) * sin\ (pi\ /\ real\_of\_nat\ k)))$

thm Counting_spheres.BIEFJHU_explicit:

$\forall (h::real)\ k::nat.\ pack\_ineq\_def\_a \wedge (1::real) \leq h \wedge h \leq h0 \wedge (3::nat) \leq k \longrightarrow DECIMAL\ (591::nat)\ (1000::nat) - DECIMAL\ (331::nat)\ (10000::nat) * real\_of\_nat\ k + DECIMAL\ (506::nat)\ (1000::nat) * lfun\ h \leq max\ (0::real)\ (regular\_spherical\_polygon\_area\ (h * (sqrt3\ /\ DECIMAL\ (40::nat)\ (10::nat)) + sqrt\ ((1::real) - (h\ /\ real\_of\_nat\ (2::nat))^2)\ /\ real\_of\_nat\ (2::nat))\ (real\_of\_nat\ k))$

thm Counting_spheres.UKBRPFE_explicit:

$\forall k::nat.\ pack\_ineq\_def\_a \wedge (3::nat) \leq k \longrightarrow DECIMAL\ (591::nat)\ (1000::nat) - DECIMAL\ (331::nat)\ (10000::nat) * real\_of\_nat\ k + (DECIMAL\ (506::nat)\ (1000::nat) * lfun\ (1::real) + (1::real)) \leq max\ (0::real)\ (regular\_spherical\_polygon\_area\ (cos\ (DECIMAL\ (797::nat)\ (1000::nat)))\ (real\_of\_nat\ k))$

thm Counting_spheres.DLWCHEM_sum:

$\forall (h::nat \Rightarrow real)\ (k::nat \Rightarrow nat)\ n::nat.\ pack\_ineq\_def\_a \wedge (12::nat) < n \wedge (\forall i<n.\ (3::nat) \leq k\ i \wedge (1::real) \leq h\ i \wedge h\ i \leq h0) \wedge sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ real\_of\_nat\ (k\ i)) \leq real\_of\_nat\ (6::nat) * real\_of\_nat\ n - real\_of\_nat\ (12::nat) \wedge sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ max\ (0::real)\ (regular\_spherical\_polygon\_area\ (h\ i * (sqrt3\ /\ DECIMAL\ (40::nat)\ (10::nat)) + sqrt\ ((1::real) - (h\ i\ /\ real\_of\_nat\ (2::nat))^2)\ /\ real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (k\ i)))) \leq real\_of\_nat\ (4::nat) * pi \wedge real\_of\_nat\ (12::nat) < sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ lfun\ (h\ i)) \longrightarrow n < (16::nat)$

thm Counting_spheres.XULJEPR_sum:

$\forall (h::nat \Rightarrow real)\ (k::nat \Rightarrow nat)\ n::nat.\ pack\_ineq\_def\_a \wedge (12::nat) < n \wedge h\ (0::nat) = (1::real) \wedge (\forall i<n.\ (3::nat) \leq k\ i \wedge (1::real) \leq h\ i \wedge h\ i \leq h0) \wedge sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ real\_of\_nat\ (k\ i)) \leq real\_of\_nat\ (6::nat) * real\_of\_nat\ n - real\_of\_nat\ (12::nat) \wedge max\ (0::real)\ (regular\_spherical\_polygon\_area\ (cos\ (DECIMAL\ (797::nat)\ (1000::nat)))\ (real\_of\_nat\ (k\ (0::nat)))) + sum\ (dotdot\ (1::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ max\ (0::real)\ (regular\_spherical\_polygon\_area\ (h\ i * (sqrt3\ /\ DECIMAL\ (40::nat)\ (10::nat)) + sqrt\ ((1::real) - (h\ i\ /\ real\_of\_nat\ (2::nat))^2)\ /\ real\_of\_nat\ (2::nat))\ (real\_of\_nat\ (k\ i)))) \leq real\_of\_nat\ (4::nat) * pi \wedge real\_of\_nat\ (12::nat) < sum\ (dotdot\ (0::nat)\ (n - (1::nat)))\ (\lambda i::nat.\ lfun\ (h\ i)) \longrightarrow False$

thm Counting_spheres.REAL_CONVEX_ON_SECOND_SECANT:

$\forall (f::real \Rightarrow real)\ (f'::real \Rightarrow real)\ (f''::real \Rightarrow real)\ s::real \Rightarrow bool.\ is\_realinterval\ s \wedge \neg (\exists a::real.\ s = INSERT\ a\ EMPTY) \wedge (\forall x::real.\ IN\ x\ s \longrightarrow has\_real\_derivative$

$f\ (f'\ x)\ (within\ (atreal\ x)\ s)) \wedge (\forall x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f'$ $(f''\ x)\ (within\ (atreal\ x)\ s)) \wedge (\forall x{::}real.\ IN\ x\ s \longrightarrow (0{::}real) \leq f''\ x) \longrightarrow$ $(\forall (x{::}real)\ y{::}real.\ IN\ x\ s \wedge IN\ y\ s \longrightarrow f\ y - f\ x \leq f'\ y * (y - x))$

thm Counting_spheres.asn_sin_t′:

$derived\_form\ (|sin\ (?x{::}real) * (?t{::}real)| < (1{::}real))\ (\lambda x{::}real.\ x - asn\ (sin\ x * ?t))\ ((1{::}real) - cos\ ?x * ?t * inverse\_class.inverse\ (sqrt\ ((1{::}real) - (sin\ ?x * ?t)^2)))\ ?x\ (closed\_real\_interval\ [(0{::}real,\ pi)])$

thm Counting_spheres.asn_sin_t′′:

$derived\_form\ (sqrt\ ((1{::}real) - (sin\ (?x{::}real) * (?t{::}real))^2) \neq (0{::}real) \wedge (0{::}real) < (1{::}real) - (sin\ ?x * ?t)^2)\ (\lambda x{::}real.\ (1{::}real) - cos\ x * ?t * inverse\_class.inverse\ (sqrt\ ((1{::}real) - (sin\ x * ?t)^2)))\ (-\ (cos\ ?x * ?t * (-\ (real\_of\_nat\ (2{::}nat) * ((sin\ ?x * ?t)^{1{::}nat} * (cos\ ?x * ?t))) * inverse\_class.inverse\ (real\_of\_nat\ (2{::}nat) * sqrt\ ((1{::}real) - (sin\ ?x * ?t)^2)) * -\ inverse\_class.inverse\ ((sqrt\ ((1{::}real) - (sin\ ?x * ?t)^2))^2)) + -\ sin\ ?x * ?t * inverse\_class.inverse\ (sqrt\ ((1{::}real) - (sin\ ?x * ?t)^2))))\ ?x\ (closed\_real\_interval\ [(0{::}real,\ pi)])$

thm Counting_spheres.asn_sin_t′′_alt:

$\forall (x{::}real)\ (t{::}real)\ alpha{::}real.\ |sin\ x * t| < (1{::}real) \wedge cos\ alpha = sin\ x * t \longrightarrow derived\_form\ True\ (\lambda x{::}real.\ (1{::}real) - cos\ x * t * inverse\_class.inverse\ (sqrt\ ((1{::}real) - (sin\ x * t)^2)))\ (t * (((1{::}real) - t^2) * (sin\ x * inverse\_class.inverse\ |sin\ alpha|^{3{::}nat})))\ x\ (closed\_real\_interval\ [(0{::}real,\ pi)])$

thm Counting_spheres.real_interval_not_sing:

$\forall (a{::}real)\ b{::}real.\ a < b \longrightarrow \neg\ (\exists c{::}real.\ closed\_real\_interval\ [(a,\ b)] = INSERT\ c\ EMPTY)$

thm Counting_spheres.g_convex:

$\forall t{::}real.\ (0{::}real) < t \wedge t < (1{::}real) \longrightarrow (\exists (s{::}real \Rightarrow bool)\ (f'{::}real \Rightarrow real)\ f''{::}real \Rightarrow real.\ s = closed\_real\_interval\ [(0{::}real,\ pi)] \wedge is\_realinterval\ s \wedge \neg\ (\exists a{::}real.\ s = INSERT\ a\ EMPTY) \wedge (\forall x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ (\lambda x{::}real.\ x - asn\ (sin\ x * t))\ (f'\ x)\ (within\ (atreal\ x)\ s)) \wedge (\forall x{::}real.\ IN\ x\ s \longrightarrow has\_real\_derivative\ f'\ (f''\ x)\ (within\ (atreal\ x)\ s)) \wedge (\forall x{::}real.\ IN\ x\ s \longrightarrow (0{::}real) \leq f''\ x))$

thm Counting_spheres.GOTCJAH_convex_sum:

$\forall (n{::}nat)\ (t{::}real)\ (bet{::}nat \Rightarrow real)\ u{::}real.\ (0{::}nat) < n \wedge u \leq real\_of\_nat\ n * pi \wedge (0{::}real) \leq u \wedge (0{::}real) < t \wedge t < (1{::}real) \wedge sum\ (dotdot\ (0{::}nat)\ (n - (1{::}nat)))\ bet = u \wedge (\forall i{<}n.\ (0{::}real) \leq bet\ i \wedge bet\ i \leq pi) \longrightarrow u - real\_of\_nat\ n * asn\ (sin\ (u\ /\ real\_of\_nat\ n) * t) \leq sum\ (dotdot\ (0{::}nat)\ (n - (1{::}nat)))\ (\lambda i{::}nat.\ bet\ i - asn\ (sin\ (bet\ i) * t))$

thm Counting_spheres.dih_dot:

$\forall (u{::}(real,\ 3)\ cart)\ (v{::}(real,\ 3)\ cart)\ w{::}(real,\ 3)\ cart.\ u \neq vec\ (0{::}nat) \wedge dot\ (vector\_sub\ w\ u)\ v = (0{::}real) \wedge dot\ (vector\_sub\ w\ u)\ u = (0{::}real) \longrightarrow dihV\ (vec\ (0{::}nat))\ u\ v\ w = pi\ /\ real\_of\_nat\ (2{::}nat)$

thm Counting_spheres.abs_1_prod:

$\forall (x::real)\ y::real.\ |x| \le (1::real) \wedge |y| \le (1::real) \longrightarrow |x * y| \le (1::real)$

thm Counting_spheres.sloc2_ortho:

$\forall (va::(real,\ 3)\ cart)\ (vb::(real,\ 3)\ cart)\ vc::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ va\ (INSERT\ vb\ (INSERT\ vc\ EMPTY)))) \wedge dihV\ (vec\ (0::nat))\ vc\ va\ vb = pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow LET\ (\lambda bet::real.\ LET\_END\ (LET\ (\lambda alp::real.\ LET\_END\ (LET\ (\lambda t::real.\ LET\_END\ (cos\ alp = sin\ bet * t))\ (cos\ (arcV\ (vec\ (0::nat))\ vb\ vc))))\ (dihV\ (vec\ (0::nat))\ va\ vb\ vc)))\ (dihV\ (vec\ (0::nat))\ vb\ vc\ va)$

thm Counting_spheres.vol_solid_triangle_ortho:

$\forall (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)))) \wedge dot\ (vector\_sub\ w\ u)\ v = (0::real) \wedge dot\ (vector\_sub\ w\ u)\ u = (0::real) \longrightarrow LET\ (\lambda bet::real.\ LET\_END\ (LET\ (\lambda t::real.\ LET\_END\ (real\_of\_nat\ (3::nat) * vol\_solid\_triangle\ (vec\ (0::nat))\ u\ v\ w\ (1::real) = bet - asn\ (sin\ bet * t)))\ (cos\ (arcV\ (vec\ (0::nat))\ v\ u))))\ (dihV\ (vec\ (0::nat))\ v\ u\ w)$

thm Counting_spheres.INJ_IMAGE:

$\forall (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ INJ\ (?f::?'b::type \Rightarrow ?'a::type)\ a\ b \longrightarrow SUBSET\ (IMAGE\ ?f\ a)\ b$

thm Counting_spheres.INJ_CARD:

$\forall (a::?'b::type \Rightarrow bool)\ (b::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ b \wedge INJ\ f\ a\ b \longrightarrow FINITE\ a \wedge CARD\ a \le CARD\ b$

thm Counting_spheres.card_packing_ball:

$\forall r{\ge}0::real.\ \exists n::nat.\ \forall S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge SUBSET\ S\ (ball\ (vec\ (0::nat),\ r)) \longrightarrow FINITE\ S \wedge CARD\ S \le n$

thm Counting_spheres.card_packing_annulus:

$\exists n::nat.\ \forall S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge SUBSET\ S\ ball\_annulus \longrightarrow FINITE\ S \wedge CARD\ S \le n$

thm Counting_spheres.FINITE_MAX_EXISTS:

$\forall s::nat \Rightarrow bool.\ s \ne EMPTY \wedge FINITE\ s \longrightarrow (\exists a::nat.\ s\ a \wedge (\forall b::nat.\ s\ b \longrightarrow b \le a))$

thm Counting_spheres.NOT_EMPTY_IMAGE:

$\forall (S::?'b::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ S \ne EMPTY \longrightarrow IMAGE\ f\ S \ne EMPTY$

thm Counting_spheres.PACKING_INSERT:

$\forall (v::(real,\ 3)\ cart)\ S::(real,\ 3)\ cart \Rightarrow bool.\ packing\ S \wedge \neg\ S\ v \wedge (\forall w::(real,\ 3)\ cart.\ S\ w \longrightarrow real\_of\_nat\ (2::nat) \le distance\ (v,\ w)) \longrightarrow packing\ (INSERT\ v\ S)$

thm Counting_spheres.weak_saturation:

$\forall$ (*W*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) (*S*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *r*::*real*. *real_of_nat* (*2*::*nat*) $\leq$ *r* $\wedge$ *r* $\leq$ *real_of_nat* (*2*::*nat*) $*$ *h0* $\wedge$ *SUBSET S W* $\wedge$ *packing W* $\wedge$ *SUBSET W ball_annulus* $\wedge$ ($\forall$ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *S v* $\wedge$ *W w* $\wedge$ *distance* (*v*, *w*) $<$ *r* $\longrightarrow$ *v* = *w*) $\longrightarrow$ ($\exists$ *V*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *SUBSET V ball_annulus* $\wedge$ *packing V* $\wedge$ *weakly_saturated V r* (*real_of_nat* (*2*::*nat*) $*$ *h0*) $\wedge$ *FINITE V* $\wedge$ *SUBSET W V* $\wedge$ ($\forall$ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *S v* $\wedge$ *V w* $\wedge$ *distance* (*v*, *w*) $<$ *r* $\longrightarrow$ *v* = *w*))

thm Counting_spheres.RADIAL_NORM_LINEAR_INVARIANT:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *r*::*real*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\wedge$ ($\forall$ *y*::(*real*, *?'a*::*type*) *cart*. $\exists$ *x*::(*real*, *?'b*::*type*) *cart*. *f x* = *y*) $\longrightarrow$ *radial r* (*vec* (*0*::*nat*)) (*IMAGE f s*) = *radial r* (*vec* (*0*::*nat*)) *s*

thm Counting_spheres.linear_inter_normball:

$\forall$ (*f*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ (*real*, *?'a*::*type*) *cart*) (*s*::(*real*, *?'b*::*type*) *cart* $\Rightarrow$ *bool*) *r*::*real*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *?'b*::*type*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\longrightarrow$ *HOL_Light_Import.INTER* (*IMAGE f s*) (*normball* (*vec* (*0*::*nat*)) *r*) = *IMAGE f* (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*))

thm Counting_spheres.solvec0_sold:

$\forall$ *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *solvec0 s* = (*if* $\exists$ *r*>*0*::*real*. *measurable* (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)) $\wedge$ *radial r* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)) *then sol* (*vec* (*0*::*nat*)) *s else* (*0*::*real*))

thm Counting_spheres.sol0_linear:

$\forall$ (*f*::(*real*, *3*) *cart* $\Rightarrow$ (*real*, *3*) *cart*) *s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *3*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\wedge$ ($\forall$ *y*::(*real*, *3*) *cart*. $\exists$ *x*::(*real*, *3*) *cart*. *f x* = *y*) $\longrightarrow$ ($\exists$ *r*>*0*::*real*. *measurable* (*HOL_Light_Import.INTER* (*IMAGE f s*) (*normball* (*vec* (*0*::*nat*)) *r*)) $\wedge$ *radial r* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER* (*IMAGE f s*) (*normball* (*vec* (*0*::*nat*)) *r*))) = ($\exists$ *r*>*0*::*real*. *measurable* (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)) $\wedge$ *radial r* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)))

thm Counting_spheres.sol0_linear_r:

$\forall$ (*f*::(*real*, *3*) *cart* $\Rightarrow$ (*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart* $\Rightarrow$ *bool*) *r*::*real*. *linear f* $\wedge$ ($\forall$ *x*::(*real*, *3*) *cart*. *vector_norm* (*f x*) = *vector_norm x*) $\wedge$ ($\forall$ *y*::(*real*, *3*) *cart*. $\exists$ *x*::(*real*, *3*) *cart*. *f x* = *y*) $\wedge$ (*0*::*real*) $<$ *r* $\longrightarrow$ (*measurable* (*HOL_Light_Import.INTER* (*IMAGE f s*) (*normball* (*vec* (*0*::*nat*)) *r*)) $\wedge$ *radial r* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER* (*IMAGE f s*) (*normball* (*vec* (*0*::*nat*)) *r*))) = (*measurable* (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)) $\wedge$ *radial r* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER s* (*normball* (*vec* (*0*::*nat*)) *r*)))

thm Counting_spheres.SOLVEC0_LINEAR_INVARIANT_3:

$\forall (f::(real, 3)\ cart \Rightarrow (real, 3)\ cart)\ s::(real, 3)\ cart \Rightarrow bool.\ linear\ f \wedge (\forall x::(real, 3)\ cart.\ vector\_norm\ (f\ x) = vector\_norm\ x) \wedge (\forall y::(real, 3)\ cart.\ \exists x::(real, 3)\ cart.\ f\ x = y) \wedge ((2::nat) \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow det\ (matrix\ f) = (1::real)) \longrightarrow solvec0\ (IMAGE\ f\ s) = solvec0\ s$

thm Counting_spheres.dropout_pad2d3d:

$\forall x::(real, 2)\ cart.\ dropout\ (3::nat)\ (pad2d3d\ x) = x$

thm Counting_spheres.pad2d3d_dropout:

$\forall x::(real, 3)\ cart.\ \$\ x\ (3::nat) = (0::real) \longrightarrow pad2d3d\ (dropout\ (3::nat)\ x) = x$

thm Counting_spheres.pad2d3d_dropout_lemma:

$\forall (A::?'a::type \Rightarrow bool)\ (P::?'a::type \Rightarrow bool)\ h::?'a::type \Rightarrow ?'a::type.\ (\forall x::?'a::type.\ IN\ x\ A \longrightarrow P\ x) \wedge (\forall x::?'a::type.\ P\ x \longrightarrow h\ x = x) \longrightarrow IMAGE\ h\ A = A$

thm Counting_spheres.pad_in:

$\forall (x::(real, 2)\ cart)\ A::(real, 3)\ cart \Rightarrow bool.\ (\forall u::(real, 3)\ cart.\ IN\ u\ A \longrightarrow \$\ u\ (3::nat) = (0::real)) \longrightarrow IN\ (pad2d3d\ x)\ A = IN\ x\ (IMAGE\ (dropout\ (3::nat))\ A)$

thm Counting_spheres.pad2d3d_facet:

$\forall (P::(real, 3)\ cart \Rightarrow bool)\ n::nat.\ polyhedron\ P \wedge (\forall u::(real, 3)\ cart.\ IN\ u\ P \longrightarrow \$\ u\ (3::nat) = (0::real)) \wedge HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%2056::(real, 3)\ cart \Rightarrow bool.\ \exists c::(real, 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2056\ (facet\_of\ c\ P)\ c))\ n \longrightarrow HAS\_SIZE\ (GSPEC\ (\lambda GEN\%PVAR\%2057::(real, 2)\ cart \Rightarrow bool.\ \exists d::(real, 2)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2057\ (facet\_of\ d\ (IMAGE\ (dropout\ (3::nat))\ P))\ d))\ n$

thm Counting_spheres.ARG_SCALE:

$\forall (u::(real, 2)\ cart)\ (w::(real, 2)\ cart)\ r::real.\ (0::real) < r \longrightarrow Arg\ (complex\_div\ (complex\_mul\ (Cx\ r)\ u)\ w) = Arg\ (complex\_div\ u\ w) \wedge Arg\ (complex\_div\ u\ (complex\_mul\ (Cx\ r)\ w)) = Arg\ (complex\_div\ u\ w)$

thm Counting_spheres.complex_frac_cancel:

$\forall (a::(real, 2)\ cart)\ (b::(real, 2)\ cart)\ c::(real, 2)\ cart.\ b \neq Cx\ (0::real) \longrightarrow complex\_div\ (complex\_div\ a\ b)\ (complex\_div\ c\ b) = complex\_div\ a\ c$

thm Counting_spheres.REAL_CX0:

$\forall z::(real, 2)\ cart.\ HOL\_Light\_Import.real\ z \wedge Re\ z = (0::real) \longrightarrow z = Cx\ (0::real)$

thm Counting_spheres.ARG_INV_ALT:

$\forall (u::(real, 2)\ cart)\ (x::(real, 2)\ cart)\ y::(real, 2)\ cart.\ u \neq Cx\ (0::real) \wedge x \neq Cx\ (0::real) \wedge y \neq Cx\ (0::real) \wedge Arg\ (complex\_div\ x\ u) \neq Arg\ (complex\_div\ y$

$u) \longrightarrow Arg\ (complex\_div\ x\ y) = real\_of\_nat\ (2::nat) * pi - Arg\ (complex\_div\ y\ x)$

thm Counting_spheres.ARG_ORDER:

$\forall (u::(real,\ 2)\ cart)\ (h::nat \Rightarrow (real,\ 2)\ cart)\ n::nat.\ u \neq Cx\ (0::real) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow h\ i \neq Cx\ (0::real)) \wedge (\forall (i::nat)\ j::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \wedge IN\ j\ (dotdot\ (1::nat)\ n) \wedge i < j \longrightarrow Arg\ (complex\_div\ (h\ i)\ u) < Arg\ (complex\_div\ (h\ j)\ u)) \wedge h\ (n + (1::nat)) = h\ (1::nat) \longrightarrow (\forall (i::nat)\ j::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \wedge IN\ j\ (dotdot\ (1::nat)\ n) \wedge i \neq j \longrightarrow Arg\ (complex\_div\ (h\ (i + (1::nat)))\ (h\ i)) \leq Arg\ (complex\_div\ (h\ j)\ (h\ i)))$

thm Counting_spheres.POLYSORT_BIJ2:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ (n::nat)\ (s::((real,\ 2)\ cart \Rightarrow bool) \Rightarrow bool)\ (r::real)\ u::(real,\ 2)\ cart.\ s = GSPEC\ (\lambda GEN\%PVAR\%2058::(real,\ 2)\ cart \Rightarrow bool.\ \exists c::(real,\ 2)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2058\ (facet\_of\ c\ P)\ c) \wedge bounded\ P \wedge polyhedron\ P \wedge (0::real) < r \wedge (\forall p::(real,\ 2)\ cart.\ vector\_norm\ p < r \longrightarrow P\ p) \wedge u \neq Cx\ (0::real) \wedge HAS\_SIZE\ s\ n \longrightarrow (\exists f::nat \Rightarrow (real,\ 2)\ cart \Rightarrow bool.\ s = IMAGE\ f\ (dotdot\ (1::nat)\ n) \wedge BIJ\ f\ (dotdot\ (1::nat)\ n)\ s \wedge (\forall (i::nat)\ k::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \wedge IN\ k\ (dotdot\ (1::nat)\ n) \wedge i \neq k \longrightarrow Arg\ (complex\_div\ (facet\_rep\_a\ P\ (f\ (i + (1::nat))))\ (facet\_rep\_a\ P\ (f\ i))) \leq Arg\ (complex\_div\ (facet\_rep\_a\ P\ (f\ k))\ (facet\_rep\_a\ P\ (f\ i)))) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow Arg\ (complex\_div\ (facet\_rep\_a\ P\ (f\ (i + (1::nat))))\ (facet\_rep\_a\ P\ (f\ i))) < pi) \wedge f\ (n + (1::nat)) = f\ (1::nat) \wedge (\forall (j::nat)\ k::nat.\ IN\ j\ (dotdot\ (1::nat)\ n) \wedge IN\ k\ (dotdot\ (1::nat)\ n) \wedge j < k \longrightarrow Arg\ (complex\_div\ (facet\_rep\_a\ P\ (f\ j))\ u) < Arg\ (complex\_div\ (facet\_rep\_a\ P\ (f\ k))\ u)))$

thm Counting_spheres.EUSOTYP_simple:

$\forall (P::(real,\ 2)\ cart \Rightarrow bool)\ (s::((real,\ 2)\ cart \Rightarrow bool) \Rightarrow bool)\ (r::real)\ (n::nat)\ u2::(real,\ 2)\ cart.\ polyhedron\ P \wedge bounded\ P \wedge s = GSPEC\ (\lambda GEN\%PVAR\%2063::(real,\ 2)\ cart \Rightarrow bool.\ \exists c::(real,\ 2)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2063\ (facet\_of\ c\ P)\ c) \wedge HAS\_SIZE\ s\ n \wedge (0::real) < r \wedge u2 \neq vec\ (0::nat) \wedge (\forall p2::(real,\ 2)\ cart.\ vector\_norm\ p2 < r \longrightarrow IN\ p2\ P) \longrightarrow (\exists (g::nat \Rightarrow (real,\ 2)\ cart)\ h::nat \Rightarrow (real,\ 2)\ cart.\ (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow IN\ (g\ i)\ P \wedge vector\_norm\ (g\ i) = r) \wedge g\ (n + (1::nat)) = g\ (1::nat) \wedge (\forall (j::nat)\ k::nat.\ IN\ j\ (dotdot\ (1::nat)\ n) \wedge IN\ k\ (dotdot\ (1::nat)\ n) \wedge j < k \longrightarrow Arg\ (complex\_div\ (g\ j)\ u2) < Arg\ (complex\_div\ (g\ k)\ u2)) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow IN\ (h\ i)\ P \wedge vector\_norm\ (h\ i) = r * inverse\_class.inverse\ (cos\ (Arg\ (complex\_div\ (g\ (i + (1::nat)))\ (g\ i)) / real\_of\_nat\ (2::nat)))) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow Arg\ (complex\_div\ (h\ i)\ (g\ i)) = Arg\ (complex\_div\ (g\ (i + (1::nat)))\ (g\ i)) / real\_of\_nat\ (2::nat) \wedge Arg\ (complex\_div\ (g\ (i + (1::nat)))\ (h\ i)) = Arg\ (complex\_div\ (g\ (i + (1::nat)))\ (g\ i)) / real\_of\_nat\ (2::nat)) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow dot\ (g\ i)\ (vector\_sub\ (h\ i)\ (g\ i)) = (0::real) \wedge dot\ (g\ (i + (1::nat)))\ (vector\_sub\ (h\ i)\ (g\ (i + (1::nat)))) = (0::real)) \wedge (1::nat) < n \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow g\ i \neq Cx\ (0::real)) \wedge (\forall i::nat.\ IN\ i\ (dotdot\ (1::nat)\ n) \longrightarrow h\ i \neq Cx\ (0::real)) \wedge$

(∀ i::nat. IN i (dotdot (1::nat) n) ⟶ Arg (complex_div (g (i + (1::nat))) (g i)) < pi))

thm Counting_spheres.EUSOTYP_general:

∀ (P::(real, 3) cart ⇒ bool) (A::(real, 3) cart ⇒ bool) (n::nat) (s::((real, 3) cart ⇒ bool) ⇒ bool) (r::real) (u0::(real, 3) cart) (u1::(real, 3) cart) u2::(real, 3) cart. polyhedron P ∧ bounded P ∧ SUBSET P A ∧ s = GSPEC (λGEN%PVAR%2066::(real, 3) cart ⇒ bool. ∃ c::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2066 (facet_of c P) c) ∧ HAS_SIZE s n ∧ (0::real) < r ∧ u2 ≠ u0 ∧ u1 ≠ u0 ∧ IN u0 P ∧ IN u2 A ∧ (∀ v::(real, 3) cart. IN v A = (dot (vector_sub v u0) (vector_sub u1 u0) = (0::real))) ∧ (∀ p::(real, 3) cart. distance (p, u0) < r ∧ IN p A ⟶ IN p P) ⟶ (∃ (g::nat ⇒ (real, 3) cart) h::nat ⇒ (real, 3) cart. (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ IN (g i) P ∧ distance (g i, u0) = r) ∧ g (n + (1::nat)) = g (1::nat) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ IN (h i) P ∧ vector_norm (vector_sub (h i) u0) = r ∗ inverse_class.inverse (cos (azim u0 u1 (g i) (g (i + (1::nat))) / real_of_nat (2::nat)))) ∧ (∀ (j::nat) k::nat. IN j (dotdot (1::nat) n) ∧ IN k (dotdot (1::nat) n) ∧ j < k ⟶ azim u0 u1 u2 (g j) < azim u0 u1 u2 (g k)) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ azim u0 u1 (g i) (h i) = azim u0 u1 (g i) (g (i + (1::nat))) / real_of_nat (2::nat) ∧ azim u0 u1 (h i) (g (i + (1::nat))) = azim u0 u1 (g i) (g (i + (1::nat))) / real_of_nat (2::nat)) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ dot (vector_sub (g i) u0) (vector_sub (h i) (g i)) = (0::real) ∧ dot (vector_sub (g (i + (1::nat))) u0) (vector_sub (h i) (g (i + (1::nat)))) = (0::real)) ∧ (1::nat) < n ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ g i ≠ u0) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ h i ≠ u0) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ azim u0 u1 (g i) (g (i + (1::nat))) < pi))

thm Counting_spheres.AZIM_SUM_LE:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (w1::(real, 3) cart) (w2::(real, 3) cart) w3::(real, 3) cart. ¬ collinear (INSERT x (INSERT y (INSERT z EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT w1 EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT w2 EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT w3 EMPTY))) ∧ azim x y z w1 ≤ azim x y z w2 ∧ azim x y z w2 ≤ azim x y z w3 ⟶ azim x y w1 w3 = azim x y w1 w2 + azim x y w2 w3

thm Counting_spheres.AZIM_NN:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) u::(real, 3) cart. (0::real) ≤ azim x y z u

thm Counting_spheres.AZIM_BASE_SHIFT_LT:

∀ (x::(real, 3) cart) (y::(real, 3) cart) (z::(real, 3) cart) (z′::(real, 3) cart) (w1::(real, 3) cart) (w2::(real, 3) cart) w3::(real, 3) cart. ¬ collinear (INSERT x (INSERT y (INSERT z EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT z′ EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT w1 EMPTY))) ∧ ¬ collinear (INSERT x (INSERT y (INSERT w2 EMPTY))) ∧

2017

$\neg$ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* *w3* *EMPTY*))) $\wedge$ *azim x y z w1* < *azim x y z w2* $\wedge$ *azim x y z w2* < *azim x y z w3* $\wedge$ *azim x y z′ w1* < *azim x y z′ w3* $\longrightarrow$ *azim x y z′ w1* < *azim x y z′ w2* $\wedge$ *azim x y z′ w2* < *azim x y z′ w3*

thm Counting_spheres.AZIM_COMP_LT:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. (*0*::*real*) < *azim x y z u* $\wedge$ *azim x y z u* < *azim x y z v* $\longrightarrow$ *azim x y v z* < *azim x y u z*

thm Counting_spheres.AZIM_COMP_LE:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. (*0*::*real*) < *azim x y z u* $\wedge$ *azim x y z u* $\leq$ *azim x y z v* $\longrightarrow$ *azim x y v z* $\leq$ *azim x y u z*

thm Counting_spheres.AZIM_COMP2_LE:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. (*0*::*real*) < *azim x y u z* $\wedge$ (*0*::*real*) < *azim x y v z* $\wedge$ *azim x y u z* $\leq$ *azim x y v z* $\longrightarrow$ *azim x y z v* $\leq$ *azim x y z u*

thm Counting_spheres.AZIM_COMP2_LT:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *v*::(*real*, *3*) *cart*. (*0*::*real*) < *azim x y u z* $\wedge$ (*0*::*real*) < *azim x y v z* $\wedge$ *azim x y u z* < *azim x y v z* $\longrightarrow$ *azim x y z v* < *azim x y z u*

thm Counting_spheres.WEDGE_ORDER_DISJOINT:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*n*::*nat*) *g*::*nat* $\Rightarrow$ (*real*, *3*) *cart*. $\neg$ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ ($\forall$ *i*::*nat*. *IN* *i* (*dotdot* (*1*::*nat*) *n*) $\longrightarrow$ $\neg$ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* (*g i*) *EMPTY*)))) $\wedge$ *g* (*n* + (*1*::*nat*)) = *g* (*1*::*nat*) $\wedge$ ($\forall$ (*j*::*nat*) *k*::*nat*. *IN* *j* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *IN* *k* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *j* < *k* $\longrightarrow$ *azim x y z* (*g j*) < *azim x y z* (*g k*)) $\longrightarrow$ ($\forall$ (*j*::*nat*) *k*::*nat*. *IN* *j* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *IN* *k* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *j* $\neq$ *k* $\longrightarrow$ *HOL_Light_Import.INTER* (*wedge x y* (*g j*) (*g* (*j* + (*1*::*nat*)))) (*wedge x y* (*g k*) (*g* (*k* + (*1*::*nat*)))) = *EMPTY*)

thm Counting_spheres.ORDER_AZIM_SUM2Pi:

$\forall$ (*x*::(*real*, *3*) *cart*) (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*n*::*nat*) *g*::*nat* $\Rightarrow$ (*real*, *3*) *cart*. $\neg$ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* *z* *EMPTY*))) $\wedge$ ($\forall$ *i*::*nat*. *IN* *i* (*dotdot* (*1*::*nat*) *n*) $\longrightarrow$ $\neg$ *collinear* (*INSERT* *x* (*INSERT* *y* (*INSERT* (*g i*) *EMPTY*)))) $\wedge$ *g* (*n* + (*1*::*nat*)) = *g* (*1*::*nat*) $\wedge$ (*1*::*nat*) < *n* $\wedge$ ($\forall$ (*j*::*nat*) *k*::*nat*. *IN* *j* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *IN* *k* (*dotdot* (*1*::*nat*) *n*) $\wedge$ *j* < *k* $\longrightarrow$ *azim x y z* (*g j*) < *azim x y z* (*g k*)) $\longrightarrow$ *sum* (*dotdot* (*1*::*nat*) *n*) (λ*i*::*nat*. *azim x y* (*g i*) (*g* (*i* + (*1*::*nat*)))) = *real_of_nat* (*2*::*nat*) $*$ *pi*

thm Counting_spheres.AFFINE_VEC0:

$\forall$ (*u*::(*real*, *?′a*::*type*) *cart*) *t*::*real*. *t* $\neq$ (*1*::*real*) $\longrightarrow$ *IN* (*vec* (*0*::*nat*)) (*hull* *affine* (*INSERT* *u* (*INSERT* (% *t u*) *EMPTY*)))

thm Counting_spheres.RELATIVE_INTERIOR_AFFINE_FACE:

$\forall (C::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (p::(real,\ ?'a::type)\ cart)\ f::(real,\ ?'a::type)\ cart \Rightarrow bool.\ convex\ C \wedge face\_of\ f\ C \wedge IN\ p\ (hull\ affine\ f) \wedge IN\ p\ (relative\_interior\ C) \longrightarrow f = C$

thm Counting_spheres.FCHANGED_AFFINE:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ f::(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow HOL\_Light\_Import.INTER\ (fchanged\ f)\ (hull\ affine\ f) = relative\_interior\ f$

thm Counting_spheres.RCONE_PREP:

$\forall (p::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u0::(real,\ 3)\ cart)\ b::real.\ (0::real) < b \wedge v \neq vec\ (0::nat) \wedge (0::real) < dot\ v\ v \wedge u0 = \%\ (b\ /\ dot\ v\ v)\ v \wedge (0::real) < (?t::real) \wedge ?t < (1::real) \wedge dot\ p\ v = b \longrightarrow dot\ u0\ u0 = b * b\ /\ dot\ v\ v \wedge dot\ p\ u0 = b * b\ /\ dot\ v\ v \wedge (distance\ (p,\ u0))^2 = dot\ p\ p - b * b\ /\ dot\ v\ v$

thm Counting_spheres.RCONE_DISK:

$\forall (p::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (u0::(real,\ 3)\ cart)\ (b::real)\ (r::real)\ t::real.\ (0::real) < b \wedge v \neq vec\ (0::nat) \wedge (0::real) < dot\ v\ v \wedge distance\ (p,\ u0) < r \wedge u0 = \%\ (b\ /\ dot\ v\ v)\ v \wedge (0::real) < t \wedge t < (1::real) \wedge dot\ p\ v = b \wedge r = b * (sqrt\ ((1::real) - t^2)\ /\ (t * vector\_norm\ v)) \longrightarrow IN\ p\ (rcone\_gt\ (vec\ (0::nat))\ v\ t)$

thm Counting_spheres.RDISK_R:

$\forall (v::(real,\ 3)\ cart)\ (u0::(real,\ 3)\ cart)\ (b::real)\ t::real.\ (0::real) < b \wedge v \neq vec\ (0::nat) \wedge (0::real) < dot\ v\ v \wedge (0::real) < t \wedge t < (1::real) \wedge u0 = \%\ (b\ /\ dot\ v\ v)\ v \longrightarrow (\exists\, r>0::real.\ (\forall\, p::(real,\ 3)\ cart.\ distance\ (p,\ u0) < r \wedge dot\ p\ v = b \longrightarrow IN\ p\ (rcone\_gt\ (vec\ (0::nat))\ v\ t)) \wedge (\forall\, w::(real,\ 3)\ cart.\ distance\ (w,\ u0) = r \wedge dot\ w\ v = b \longrightarrow cos\ (arcV\ (vec\ (0::nat))\ u0\ w) = t))$

thm Counting_spheres.FCHANGED_MEASURABLE:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (f::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow measurable\ (HOL\_Light\_Import.INTER\ (fchanged\ f)\ (normball\ (vec\ (0::nat))\ r))$

thm Counting_spheres.RADIAL_NORMBALL:

$\forall (p::(real,\ 3)\ cart)\ r::real.\ radial\ r\ p\ (normball\ p\ r)$

thm Counting_spheres.FCHANGED_RADIAL:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (f::(real,\ 3)\ cart \Rightarrow bool)\ r::real.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow radial\ r\ (vec\ (0::nat))\ (HOL\_Light\_Import.INTER\ (fchanged\ f)\ (normball\ (vec\ (0::nat))\ r))$

thm Counting_spheres.WEDGE_SPLIT:

$\forall (u0::(real,\ 3)\ cart)\ (u1::(real,\ 3)\ cart)\ (u2::(real,\ 3)\ cart)\ (u3::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ u0\ (INSERT\ u1\ (INSERT\ u2$

EMPTY))) ∧ ¬ collinear (INSERT u0 (INSERT u1 (INSERT u3 EMPTY))) ∧ IN w (wedge u0 u1 u2 u3) ⟶ ¬ collinear (INSERT u0 (INSERT u1 (INSERT w EMPTY))) ∧ HOL_Light_Import.INTER (wedge u0 u1 u2 w) (wedge u0 u1 w u3) = EMPTY ∧ SUBSET (wedge u0 u1 u2 w) (wedge u0 u1 u2 u3) ∧ SUBSET (wedge u0 u1 w u3) (wedge u0 u1 u2 u3)

thm Counting_spheres.cone0_subset_lune:

∀ (u0::(real, ?'a::type) cart) (u1::(real, ?'a::type) cart) (u2::(real, ?'a::type) cart) u3::(real, ?'a::type) cart. SUBSET (cone0 u0 (INSERT u1 (INSERT u2 (INSERT u3 EMPTY)))) (aff_gt (INSERT u0 (INSERT u1 EMPTY)) (INSERT u2 (INSERT u3 EMPTY)))

thm Counting_spheres.COLLINEAR_UNEQUAL:

∀ (u0::(real, ?'a::type) cart) (u1::(real, ?'a::type) cart) u2::(real, ?'a::type) cart. ¬ collinear (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) ⟶ ¬ IN u2 (INSERT u0 (INSERT u1 EMPTY)) ∧ ¬ IN u1 (INSERT u0 EMPTY)

thm Counting_spheres.HAS_SIZE_GE_2:

∀ s::?'a::type ⇒ bool. FINITE s ∧ (1::nat) < CARD s ⟶ (∀ x::?'a::type. IN x s ⟶ (∃ y::?'a::type. IN y s ∧ y ≠ x))

thm Counting_spheres.TWO_IMP_HAS_SIZE_GE_2:

∀ (s::?'a::type ⇒ bool) (x::?'a::type) y::?'a::type. IN x s ∧ IN y s ∧ x ≠ y ∧ FINITE s ⟶ (1::nat) < CARD s

thm Counting_spheres.AFF_GT_RELATIVE_INTERIOR:

∀ s::(real, ?'a::type) cart ⇒ bool. FINITE s ∧ (1::nat) < CARD s ⟶ SUBSET (aff_gt EMPTY s) (relative_interior (hull convex s))

thm Counting_spheres.NOT_COLLINEAR_AFF_DIM_2:

∀ (u0::(real, ?'a::type) cart) (u1::(real, ?'a::type) cart) u2::(real, ?'a::type) cart. ¬ collinear (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) ⟶ aff_dim (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) = int (2::nat)

thm Counting_spheres.FACET_AFF_DIM_2:

∀ (p::(real, 3) cart ⇒ bool) f::(real, 3) cart ⇒ bool. polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ⟶ aff_dim f = int (2::nat)

thm Counting_spheres.CONE0_RELATIVE_INTERIOR_FACET:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart ⇒ bool) (u0::(real, 3) cart) (u1::(real, 3) cart) u2::(real, 3) cart. polyhedron p ∧ bounded p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ∧ ¬ collinear (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) ∧ SUBSET (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) f ⟶ SUBSET (aff_gt EMPTY (INSERT u0 (INSERT u1 (INSERT u2 EMPTY)))) (relative_interior f)

thm Counting_spheres.CONE0_FCHANGED_AFF_GT:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *FINITE s* ∧ (*1*::*nat*) < *CARD s* ∧ ¬ *IN* (*vec* (*0*::*nat*)) *s* ⟶ *SUBSET* (*cone0* (*vec* (*0*::*nat*)) *s*) (*fchanged* (*hull convex s*))

thm Counting_spheres.CONE0_FCHANGED:

∀ (*p*::(*real*, *3*) *cart* ⇒ *bool*) (*f*::(*real*, *3*) *cart* ⇒ *bool*) (*u0*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *u2*::(*real*, *3*) *cart*. *polyhedron p* ∧ *bounded p* ∧ *IN* (*vec* (*0*::*nat*)) (*interior p*) ∧ *facet_of f p* ∧ ¬ *collinear* (*INSERT u0* (*INSERT u1* (*INSERT u2 EMPTY*))) ∧ *SUBSET* (*INSERT u0* (*INSERT u1* (*INSERT u2 EMPTY*))) *f* ⟶ *SUBSET* (*cone0* (*vec* (*0*::*nat*)) (*INSERT u0* (*INSERT u1* (*INSERT u2 EMPTY*)))) (*fchanged f*)

thm Counting_spheres.COLLINEAR_ORTHO_PLANE:

∀ (*p*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*u0*::(*real*, *?'a*::*type*) *cart*) (*b*::*real*) *u1*::(*real*, *?'a*::*type*) *cart*. *v* ≠ *vec* (*0*::*nat*) ∧ *u0* ≠ *u1* ∧ *dot u0 v* = *b* ∧ *dot p v* = *b* ∧ *u1* = *vector_add u0 v* ∧ *collinear* (*INSERT u0* (*INSERT u1* (*INSERT p EMPTY*))) ⟶ *p* = *u0*

thm Counting_spheres.collinear_translate_axis:

∀ (*t*::*real*) (*u1*::(*real*, *3*) *cart*) *u2*::(*real*, *3*) *cart*. *collinear* (*INSERT* (% *t u1*) (*INSERT u1* (*INSERT u2 EMPTY*))) = *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*vector_sub u1* (% *t u1*)) (*INSERT u2 EMPTY*)))

thm Counting_spheres.azim_axis:

∀ (*t*::*real*) (*u1*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* (% *t u1*) (*INSERT u1* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT* (% *t u1*) (*INSERT u1* (*INSERT w EMPTY*))) ⟶ *azim* (% *t u1*) *u1 u w* = *azim* (*vec* (*0*::*nat*)) (*vector_sub u1* (% *t u1*)) *u w*

thm Counting_spheres.EUSOTYP2_general:

∀ (*P*::(*real*, *3*) *cart* ⇒ *bool*) (*c3*::(*real*, *3*) *cart* ⇒ *bool*) (*A*::(*real*, *3*) *cart* ⇒ *bool*) (*n*::*nat*) (*s*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*t*::*real*) (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *b*::*real*. *polyhedron P* ∧ *bounded P* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ *facet_of c3 P* ∧ *SUBSET c3 A* ∧ *HOL_Light_Import.INTER P A* = *c3* ∧ *A* = *GSPEC* (λ*GEN%PVAR%2069*::(*real*, *3*) *cart*. ∃ *p*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2069* (*dot p v* = *b*) *p*) ∧ *s* = *GSPEC* (λ*GEN%PVAR%2070*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *c*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2070* (*facet_of c c3*) *c*) ∧ *HAS_SIZE s n* ∧ (*0*::*real*) < *b* ∧ (*0*::*real*) < *t* ∧ *t* < (*1*::*real*) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT u EMPTY*))) ∧ *SUBSET* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) (*fchanged c3*) ∧ *SUBSET* (*HOL_Light_Import.INTER* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) *A*) *c3* ⟶ (∃ (*g*::*nat* ⇒ (*real*, *3*) *cart*) *h*::*nat* ⇒ (*real*, *3*) *cart*. (∀ *i*::*nat*. *IN i* (*dotdot* (*1*::*nat*) *n*) ⟶ *IN* (*g i*) *c3* ∧ *cos* (*arcV* (*vec* (*0*::*nat*)) *v* (*g i*)) = *t*) ∧ *g* (*n* + (*1*::*nat*)) = *g* (*1*::*nat*) ∧ (∀ *i*::*nat*. *IN i* (*dotdot* (*1*::*nat*) *n*) ⟶ *IN* (*h i*) *c3*) ∧ (∀ (*j*::*nat*) *k*::*nat*. *IN j* (*dotdot* (*1*::*nat*) *n*) ∧ *IN k* (*dotdot* (*1*::*nat*) *n*) ∧ *j* < *k* ⟶ *azim* (*vec* (*0*::*nat*)) *v u* (*g j*) < *azim* (*vec* (*0*::*nat*)) *v u* (*g k*)) ∧ (∀ *i*::*nat*. *IN i* (*dotdot* (*1*::*nat*) *n*) ⟶ *azim* (*vec* (*0*::*nat*)) *v* (*g i*) (*h*

i) = azim (vec (0::nat)) v (g i) (g (i + (1::nat))) / real_of_nat (2::nat) ∧ azim (vec (0::nat)) v (h i) (g (i + (1::nat))) = azim (vec (0::nat)) v (g i) (g (i + (1::nat))) / real_of_nat (2::nat)) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ dot (vector_sub (h i) (g i)) v = (0::real) ∧ dot (vector_sub (h i) (g (i + (1::nat)))) v = (0::real) ∧ dot (vector_sub (h i) (g i)) (g i) = (0::real) ∧ dot (vector_sub (h i) (g (i + (1::nat)))) (g (i + (1::nat))) = (0::real)) ∧ (1::nat) < n ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT (g i) EMPTY)))) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT (h i) EMPTY)))) ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ azim (vec (0::nat)) v (g i) (g (i + (1::nat))) < pi))

thm Counting_spheres.CONE0_SUBSET_WEDGE:

∀ (v::(real, 3) cart) (u::(real, 3) cart) w::(real, 3) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT u EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT w EMPTY))) ∧ (0::real) < azim (vec (0::nat)) v u w ∧ azim (vec (0::nat)) v u w < pi ⟶ SUBSET (cone0 (vec (0::nat)) (INSERT v (INSERT u (INSERT w EMPTY)))) (wedge (vec (0::nat)) v u w)

thm Counting_spheres.FACET_INTER_DISJOINT:

∀ (p::(real, ?'a::type) cart ⇒ bool) f::(real, ?'a::type) cart ⇒ bool. polyhedron p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ⟶ ¬ IN (vec (0::nat)) f

thm Counting_spheres.CONE0_AFF_GT:

∀ (x::(real, ?'a::type) cart) U::(real, ?'a::type) cart ⇒ bool. cone0 x U = aff_gt (INSERT x EMPTY) U

thm Counting_spheres.DISJOINT0_SCALE:

∀ (t::real) (u0::(real, ?'a::type) cart) (u1::(real, ?'a::type) cart) u2::(real, ?'a::type) cart. DISJOINT (INSERT (vec (0::nat)) EMPTY) (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) ∧ t ≠ (0::real) ⟶ DISJOINT (INSERT (vec (0::nat)) EMPTY) (INSERT (% t u0) (INSERT u1 (INSERT u2 EMPTY)))

thm Counting_spheres.CONE0_SCALE:

∀ (t::real) (u0::(real, ?'a::type) cart) (u1::(real, ?'a::type) cart) u2::(real, ?'a::type) cart. DISJOINT (INSERT (vec (0::nat)) EMPTY) (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) ∧ (0::real) < t ⟶ cone0 (vec (0::nat)) (INSERT u0 (INSERT u1 (INSERT u2 EMPTY))) = cone0 (vec (0::nat)) (INSERT (% t u0) (INSERT u1 (INSERT u2 EMPTY)))

thm Counting_spheres.CONE0_FCHANGED_SCALE:

∀ (p::(real, 3) cart ⇒ bool) (f::(real, 3) cart ⇒ bool) (u0::(real, 3) cart) (u1::(real, 3) cart) (u2::(real, 3) cart) t::real. polyhedron p ∧ bounded p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of f p ∧ ¬ coplanar (INSERT (vec (0::nat)) (INSERT u0 (INSERT u1 (INSERT u2 EMPTY)))) ∧ SUBSET (INSERT (% t u0) (INSERT u1 (INSERT u2 EMPTY))) f ∧ (0::real) < t ⟶ SUBSET (cone0 (*vec* (*0*::*nat*)) (*INSERT u0* (*INSERT u1* (*INSERT u2 EMPTY*)))) (*fchanged f*)

thm Counting_spheres.gotcjah_sol_half:

∀ (*c3*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*b*::*real*) (*P*::(*real*, *3*) *cart* ⇒ *bool*) (*W*::(*real*, *3*) *cart* ⇒ *bool*) (*t*::*real*) (*rho*::*real*) (*bet*::*real*) (*w0*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) *s*::*real*. *polyhedron P* ∧ *bounded P* ∧ (*0*::*real*) < *b* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ *facet_of c3 P* ∧ *fchanged c3* = *W* ∧ ((*0*::*real*) < *t* ∧ *t* < (*1*::*real*)) ∧ (*0*::*real*) < *rho* ∧ (*0*::*real*) < *s* ∧ *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2074*::(*real*, *3*) *cart*. ∃ *p*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2074* (*dot p v* = *b*) *p*)) = *c3* ∧ *SUBSET* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) *W* ∧ *v* ≠ *vec* (*0*::*nat*) ∧ (*0*::*real*) < *dot v v* ∧ *cos* (*arcV* (*vec* (*0*::*nat*)) *v w0*) = *t* ∧ *IN* (*%* *s v*) *c3* ∧ *IN w0 c3* ∧ *IN w1 c3* ∧ ¬ *coplanar* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w0* (*INSERT w1 EMPTY*)))) ∧ *dihV* (*vec* (*0*::*nat*)) *v w0 w1* = *bet* ∧ *dot* (*vector_sub w1 w0*) *v* = (*0*::*real*) ∧ *dot* (*vector_sub w1 w0*) *w0* = (*0*::*real*) ⟶ (∃ *X*::(*real*, *3*) *cart* ⇒ *bool*. *X* = *cone0* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w0* (*INSERT w1 EMPTY*))) ∧ *SUBSET X* (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v EMPTY*)) (*INSERT w0* (*INSERT w1 EMPTY*))) *W*) ∧ *measurable* (*HOL_Light_Import.INTER X* (*normball* (*vec* (*0*::*nat*)) *rho*)) ∧ *radial_norm rho* (*vec* (*0*::*nat*)) (*HOL_Light_Import.INTER X* (*normball* (*vec* (*0*::*nat*)) *rho*)) ∧ *bet* − *asn* (*sin bet* ∗ *t*) = *sol* (*vec* (*0*::*nat*)) *X*)

thm Counting_spheres.AZIM_LE_PI_EQ_DIHV_ALT:

∀ (*a*::(*real*, *3*) *cart*) (*b*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT a* (*INSERT b* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT a* (*INSERT b* (*INSERT y EMPTY*))) ∧ *azim a b x y* ≤ *pi* ⟶ *dihV a b x y* = *azim a b x y*

thm Counting_spheres.gotcjah_sol_lemma:

∀ (*c3*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*b*::*real*) (*P*::(*real*, *3*) *cart* ⇒ *bool*) (*W*::(*real*, *3*) *cart* ⇒ *bool*) (*t*::*real*) (*rho*::*real*) (*bet*::*real*) (*w0*::(*real*, *3*) *cart*) (*w1*::(*real*, *3*) *cart*) (*w2*::(*real*, *3*) *cart*) *s*::*real*. *polyhedron P* ∧ *bounded P* ∧ (*0*::*real*) < *b* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ *facet_of c3 P* ∧ *fchanged c3* = *W* ∧ ((*0*::*real*) < *t* ∧ *t* < (*1*::*real*)) ∧ (*0*::*real*) < *rho* ∧ (*0*::*real*) < *s* ∧ *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2075*::(*real*, *3*) *cart*. ∃ *p*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2075* (*dot p v* = *b*) *p*)) = *c3* ∧ *SUBSET* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) *W* ∧ *v* ≠ *vec* (*0*::*nat*) ∧ (*0*::*real*) < *dot v v* ∧ *cos* (*arcV* (*vec* (*0*::*nat*)) *v w0*) = *t* ∧ *cos* (*arcV* (*vec* (*0*::*nat*)) *v w2*) = *t* ∧ *IN* (*%* *s v*) *c3* ∧ *IN w0 c3* ∧ *IN w1 c3* ∧ *IN w2 c3* ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w0 EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w1 EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w2 EMPTY*))) ∧ *IN* (*%* *s v*) *c3* ∧ *IN w0 c3* ∧ *IN w1 c3* ∧ *IN w2 c3* ∧ *azim* (*vec* (*0*::*nat*)) *v w0 w2* / *real_of_nat* (*2*::*nat*) = *bet* ∧ *azim* (*vec* (*0*::*nat*)) *v w0 w2* < *pi* ∧ *azim* (*vec* (*0*::*nat*)) *v w0 w1* = *bet* ∧ *azim* (*vec* (*0*::*nat*)) *v w1 w2* = *bet* ∧ *dot* (*vector_sub w1 w0*) *v* = (*0*::*real*) ∧

dot (vector_sub w1 w0) w0 = (0::real) ∧ dot (vector_sub w1 w2) v = (0::real) ∧ dot (vector_sub w1 w2) w2 = (0::real) ⟶ (∃ X::(real, 3) cart ⇒ bool. SUBSET X (HOL_Light_Import.INTER (wedge (vec (0::nat)) v w0 w2) W) ∧ measurable (HOL_Light_Import.INTER X (normball (vec (0::nat)) rho)) ∧ radial_norm rho (vec (0::nat)) (HOL_Light_Import.INTER X (normball (vec (0::nat)) rho)) ∧ real_of_nat (2::nat) ∗ (bet − asn (sin bet ∗ t)) = sol (vec (0::nat)) X)

thm Counting_spheres.c3_lemma:

∀ (c3::(real, 3) cart ⇒ bool) (v::(real, 3) cart) b::real. SUBSET c3 (GSPEC (λGEN%PVAR%2076::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2076 (dot p v = b) p)) ∧ (0::real) < b ⟶ SUBSET (HOL_Light_Import.INTER (GSPEC (λGEN%PVAR%2077::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2077 (dot p v = b) p)) (fchanged c3)) c3

thm Counting_spheres.NOT_COLLINEAR:

∀ v::(real, 3) cart. v ≠ vec (0::nat) ⟶ (∃ u::(real, 3) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT u EMPTY))))

thm Counting_spheres.gotcjah_prep:

∀ (c::(real, 3) cart ⇒ bool) (v::(real, 3) cart) (b::real) (P::(real, 3) cart ⇒ bool) (WF::(real, 3) cart ⇒ bool) (t::real) (n::nat) (u0::(real, 3) cart) A::(real, 3) cart ⇒ bool. polyhedron P ∧ bounded P ∧ (0::real) < b ∧ IN (vec (0::nat)) (interior P) ∧ facet_of c P ∧ GSPEC (λGEN%PVAR%2078::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2078 (dot p v = b) p) = A ∧ % (b / dot v v) v = u0 ∧ fchanged c = WF ∧ ((0::real) < t ∧ t < (1::real)) ∧ HOL_Light_Import.INTER P (GSPEC (λGEN%PVAR%2079::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2079 (dot p v = b) p)) = c ∧ SUBSET (rcone_gt (vec (0::nat)) v t) WF ∧ HAS_SIZE (GSPEC (λGEN%PVAR%2080::(real, 3) cart ⇒ bool. ∃ f::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2080 (facet_of f c) f)) n ⟶ SUBSET c A ∧ v ≠ vec (0::nat) ∧ (0::real) < dot v v ∧ IN u0 (rcone_gt (vec (0::nat)) v t) ∧ IN u0 c ∧ SUBSET (HOL_Light_Import.INTER (rcone_gt (vec (0::nat)) v t) A) c ∧ (∃ u::(real, 3) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT u EMPTY))))

thm Counting_spheres.azim_pos:

∀ (x::(real, 3) cart) (v::(real, 3) cart) (u::(real, 3) cart) (w1::(real, 3) cart) w2::(real, 3) cart. azim x v u w1 < azim x v u w2 ∧ ¬ collinear (INSERT x (INSERT v (INSERT w1 EMPTY))) ∧ ¬ collinear (INSERT x (INSERT v (INSERT w2 EMPTY))) ∧ ¬ collinear (INSERT x (INSERT v (INSERT u EMPTY))) ⟶ (0::real) < azim x v w1 w2

thm Counting_spheres.convex_sum_corollary:

∀ (n::nat) (t::real) bet::nat ⇒ real. (0::nat) < n ∧ (0::real) < t ∧ t < (1::real) ∧ sum (dotdot (1::nat) n) bet = pi ∧ (∀ i::nat. IN i (dotdot (1::nat) n) ⟶ (0::real) ≤ bet i ∧ bet i ≤ pi) ⟶ pi − real_of_nat n ∗ asn (sin (pi

/ *real_of_nat n*) ∗ *t*) ≤ *sum* (*dotdot* (*1*::*nat*) *n*) (λ*i*::*nat*. *bet i* − *asn* (*sin* (*bet i*) ∗ *t*))

thm Counting_spheres.SOL_SUBSET:

∀ (*x*::(*real*, *3*) *cart*) (*s*::(*real*, *3*) *cart* ⇒ *bool*) (*t*::(*real*, *3*) *cart* ⇒ *bool*) *r*::*real*. (*0*::*real*) < *r* ∧ *measurable* (*HOL_Light_Import.INTER s* (*normball x r*)) ∧ *measurable* (*HOL_Light_Import.INTER t* (*normball x r*)) ∧ *SUBSET s t* ∧ *radial_norm r x* (*HOL_Light_Import.INTER s* (*normball x r*)) ∧ *radial_norm r x* (*HOL_Light_Import.INTER t* (*normball x r*)) ⟶ *sol x s* ≤ *sol x t*

thm Counting_spheres.GOTCJAH:

∀ (*c*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*b*::*real*) (*P*::(*real*, *3*) *cart* ⇒ *bool*) (*WF*::(*real*, *3*) *cart* ⇒ *bool*) (*t*::*real*) *n*::*nat*. *polyhedron P* ∧ *bounded P* ∧ (*0*::*real*) < *b* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ *facet_of c P* ∧ *fchanged c* = *WF* ∧ ((*0*::*real*) < *t* ∧ *t* < (*1*::*real*)) ∧ (*c* = *HOL_Light_Import.INTER P* (*GSPEC* (λ*GEN%PVAR%2081*::(*real*, *3*) *cart*. ∃ *p*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2081* (*dot p v* = *b*) *p*)) ∧ *SUBSET* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) *WF*) ∧ *HAS_SIZE* (*GSPEC* (λ*GEN%PVAR%2082*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *u*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2082* (*facet_of u c*) *u*)) *n* ⟶ *real_of_nat* (*2*::*nat*) ∗ *pi* − *real_of_nat* (*2*::*nat*) ∗ (*real_of_nat n* ∗ *asn* (*t* ∗ *sin* (*pi* / *real_of_nat n*))) ≤ *sol* (*vec* (*0*::*nat*)) *WF*

thm Counting_spheres.rcone_def_alt:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) (*t*::*real*) *p*::(*real*, *?'a*::*type*) *cart*. *IN p* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) = (*vector_norm p* ∗ (*vector_norm v* ∗ *t*) < *dot p v*)

thm Counting_spheres.rcone_refl:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) *t*::*real*. *t* < (*1*::*real*) ∧ *v* ≠ *vec* (*0*::*nat*) ⟶ *IN v* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*)

thm Counting_spheres.rcone_nz:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) (*p*::(*real*, *?'a*::*type*) *cart*) *t*::*real*. (*0*::*real*) < *t* ∧ *IN p* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) ⟶ *p* ≠ *vec* (*0*::*nat*) ∧ *v* ≠ *vec* (*0*::*nat*)

thm Counting_spheres.rcone_dot_pos:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) (*t*::*real*) *p*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *t* ∧ *IN p* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) ⟶ (*0*::*real*) < *dot p v*

thm Counting_spheres.rcone_hyperplane:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) (*t*::*real*) (*b*::*real*) (*q*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart*. ((*0*::*real*) < *t* ∧ *t* < (*1*::*real*)) ∧ *IN p* (*rcone_gt* (*vec* (*0*::*nat*)) *v t*) ∧ % (*b* / *dot p v*) *p* = *q* ⟶ *dot q v* = *b*

thm Local_lemmas1.ARCV_BOUNDS_conjunct1:

arcV (*?x*::(*real*, *?'a*::*type*) *cart*) (*?y*::(*real*, *?'a*::*type*) *cart*) (*?z*::(*real*, *?'a*::*type*) *cart*) ≤ *pi*

thm Local_lemmas1.ARCV_BOUNDS_conjunct0:

$(0::real) \leq arcV\ (?x::(real,\ ?'a::type)\ cart)\ (?y::(real,\ ?'a::type)\ cart)\ (?z::(real,\ ?'a::type)\ cart)$

thm Counting_spheres.rcone_gt_arcV:

$\forall (v::(real,\ 3)\ cart)\ (g::real)\ p::(real,\ 3)\ cart.\ (0::real) < g \wedge g < pi\ /\ real\_of\_nat\ (2::nat) \wedge IN\ p\ (rcone\_gt\ (vec\ (0::nat))\ v\ (cos\ g)) \longrightarrow arcV\ (vec\ (0::nat))\ p\ v < g$

thm Counting_spheres.cos_bounds_0_Pi2:

$\forall x::real.\ (0::real) < x \wedge x < pi\ /\ real\_of\_nat\ (2::nat) \longrightarrow (0::real) < cos\ x \wedge cos\ x < (1::real)$

thm Counting_spheres.rcone_gt_arc_triangle:

$\forall (p::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (gv::real)\ gw::real.\ w \neq vec\ (0::nat) \wedge (0::real) < gv \wedge gv < pi\ /\ real\_of\_nat\ (2::nat) \wedge IN\ p\ (rcone\_gt\ (vec\ (0::nat))\ v\ (cos\ gv)) \wedge gv + gw \leq arcV\ (vec\ (0::nat))\ v\ w \longrightarrow gw < arcV\ (vec\ (0::nat))\ p\ w$

thm Counting_spheres.rcone_gt_facet:

$\forall (gv::real)\ (gw::real)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (q::(real,\ 3)\ cart)\ p::(real,\ 3)\ cart.\ ((0::real) < gv \wedge gv < pi\ /\ real\_of\_nat\ (2::nat)) \wedge ((0::real) < gw \wedge gw < pi\ /\ real\_of\_nat\ (2::nat)) \wedge w \neq vec\ (0::nat) \wedge IN\ p\ (rcone\_gt\ (vec\ (0::nat))\ v\ (cos\ gv)) \wedge q = \%\ (vector\_norm\ v * cos\ gv\ /\ dot\ p\ v)\ p \wedge gv + gw \leq arcV\ (vec\ (0::nat))\ v\ w \longrightarrow dot\ q\ w < vector\_norm\ w * cos\ gw$

thm Counting_spheres.edges_of_facet_of:

$\forall (P::(real,\ 3)\ cart \Rightarrow bool)\ f::(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ P \wedge bounded\ P \wedge IN\ (vec\ (0::nat))\ (interior\ P) \longrightarrow edge\_of\ f\ P = (\exists\, c::(real,\ 3)\ cart \Rightarrow bool.\ facet\_of\ f\ c \wedge facet\_of\ c\ P)$

thm Counting_spheres.BIJ_SYM:

$\forall (a::?'b::type \Rightarrow bool)\ b::?'a::type \Rightarrow bool.\ (\exists f::?'b::type \Rightarrow ?'a::type.\ BIJ\ f\ a\ b) \longrightarrow (\exists\, g::?'a::type \Rightarrow ?'b::type.\ BIJ\ g\ b\ a)$

thm Counting_spheres.BIJ_TRANS:

$\forall (B::?'c::type \Rightarrow bool)\ (A::?'b::type \Rightarrow bool)\ C::?'a::type \Rightarrow bool.\ (\exists\, pab::?'b::type \Rightarrow ?'c::type.\ BIJ\ pab\ A\ B) \wedge (\exists\, pbc::?'c::type \Rightarrow ?'a::type.\ BIJ\ pbc\ B\ C) \longrightarrow (\exists\, pab::?'b::type \Rightarrow ?'a::type.\ BIJ\ pab\ A\ C)$

thm Counting_spheres.SND_BIJ:

$\forall (a::?'b::type)\ B::?'a::type \Rightarrow bool.\ BIJ\ snd\ (GSPEC\ (\lambda GEN\%PVAR\%2087::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%2087\ (x = a \wedge B\ y)\ (x,\ y)))\ B$

thm Counting_spheres.FST_BIJ:

$\forall (A::?'b::type \Rightarrow bool)\ b::?'a::type.\ BIJ\ fst\ (GSPEC\ (\lambda GEN\%PVAR\%2088::?'b::type \times ?'a::type.\ \exists (x::?'b::type)\ y::?'a::type.\ SETSPEC\ GEN\%PVAR\%2088\ (A\ x \wedge y = b)\ (x,\ y)))\ A$

thm Counting_spheres.PREIMAGE_BIJ:

$\forall (A::?'c::type \Rightarrow bool)\ (B::?'b::type \Rightarrow bool)\ (C::?'a::type \Rightarrow bool)\ (f::?'c::type \Rightarrow ?'a::type)\ g::?'b::type \Rightarrow ?'a::type.\ (\forall a::?'c::type.\ IN\ a\ A \longrightarrow IN\ (f\ a)\ C) \wedge (\forall b::?'b::type.\ IN\ b\ B \longrightarrow IN\ (g\ b)\ C) \wedge (\forall c::?'a::type.\ IN\ c\ C \longrightarrow (\exists p::?'c::type \Rightarrow ?'b::type.\ BIJ\ p\ (preimage\ A\ f\ (INSERT\ c\ EMPTY))\ (preimage\ B\ g\ (INSERT\ c\ EMPTY)))) \longrightarrow (\exists q::?'c::type \Rightarrow ?'b::type.\ BIJ\ q\ A\ B)$

thm Counting_spheres.BIJ_FACET_HYPERFACE:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow (\exists b::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ BIJ\ b\ (GSPEC\ (\lambda GEN\%PVAR\%2090::(real,\ 3)\ cart \Rightarrow bool.\ \exists f::(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2090\ (facet\_of\ f\ p)\ f))\ (face\_set\ (hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))))$

thm Counting_spheres.POLYHEDRON_CONFORMING_FAN:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow conforming\_fan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)$

thm Counting_spheres.POLYHEDRON_D1_D:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow d\_fan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p) = d1\_fan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)$

thm Counting_spheres.POLYHEDRON_PLAIN:

$\forall p::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \longrightarrow plain\_hypermap\ (hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))$

thm Counting_spheres.POLYHEDRON_NODE_3:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ x\ (d\_fan\ (vec\ (0::nat),\ vertices\ p,\ edges\ p)) \longrightarrow (3::nat) \le CARD\ (node\ (hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))\ x)$

thm Counting_spheres.POLYHEDRON_TGJISOK:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ H::((real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ hypermap.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge H = hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p) \longrightarrow CARD\ (dart\ H) \le (6::nat) * number\_of\_faces\ H - (12::nat)$

thm Counting_spheres.EDGE_PAIR_pr23:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (d::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ d'::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ e\_fan\ x\ V\ E\ d = d' \longrightarrow pr2\ d = pr3\ d' \wedge pr3\ d = pr2\ d'$

thm Counting_spheres.EDGE_pr23:

$\forall (x::(real,\ 3)\ cart)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ y1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ FAN\ (x,\ V,\ E) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (1::nat) < CARD\ (set\_of\_edge\ v\ V\ E)) \wedge IN\ y\ (d1\_fan\ (x,\ V,\ E)) \wedge IN\ y1\ (d1\_fan\ (x,\ V,\ E)) \wedge INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY) = INSERT\ (pr2\ y1)\ (INSERT\ (pr3\ y1)\ EMPTY) \wedge y \neq y1 \longrightarrow y = edge\_map\ (hypermap1\_of\_fanx\ (x,\ V,\ E))\ y1$

thm Counting_spheres.SIMPLE_FACE_EDGE_INJ:

$\forall (H::?'a::type\ hypermap)\ (y::?'a::type)\ y1::?'a::type.\ simple\_hypermap\ H \wedge (1::nat) < CARD\ (node\ H\ (face\_map\ H\ y)) \wedge IN\ y\ (dart\ H) \wedge IN\ y\ (face\ H\ y1) \longrightarrow y \neq edge\_map\ H\ y1$

thm Counting_spheres.INJ_EDGES_FACE_pr23:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (y1::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart)\ y::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))) \wedge IN\ y\ f \wedge IN\ y1\ f \wedge INSERT\ (pr2\ y)\ (INSERT\ (pr3\ y)\ EMPTY) = INSERT\ (pr2\ y1)\ (INSERT\ (pr3\ y1)\ EMPTY) \longrightarrow y = y1$

thm Counting_spheres.BIJ_EDGES_DART_FACE:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (f::(real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ f1::(real,\ 3)\ cart \Rightarrow bool.\ bounded\ p \wedge polyhedron\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge IN\ f\ (face\_set\ (hypermap1\_of\_fanx\ (vec\ (0::nat),\ vertices\ p,\ edges\ p))) \wedge facet\_of\ f1\ p \wedge fchanged\ f1 = dartset\_leads\_into\_fan\ (vec\ (0::nat))\ (vertices\ p)\ (edges\ p)\ f \longrightarrow (\exists b::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart \times (real,\ 3)\ cart.\ BIJ\ b\ (edges\ f1)\ f)$

thm Counting_spheres.SEGMENT_EDGE_ONTO:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ e::(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge edge\_of\ e\ p \longrightarrow (\exists (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ e = closed\_segment\ [(v,\ w)])$

thm Counting_spheres.EDGE_OF_FACET_OF:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ (c::(real,\ 3)\ cart \Rightarrow bool)\ f::?'a::type.\ polyhedron\ p \wedge bounded\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ c\ p \longrightarrow edge\_of\ (?e::(real,\ 3)\ cart \Rightarrow bool)\ c = facet\_of\ ?e\ c$

thm Counting_spheres.EDGE_OF_FACET_EDGE:

∀ (p::(real, 3) cart ⇒ bool) (c::(real, 3) cart ⇒ bool) e::(real, 3) cart ⇒ bool. polyhedron p ∧ bounded p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of c p ∧ facet_of e c ⟶ edge_of e p

thm Counting_spheres.BIJ_FACET2_EDGE:

∀ (p::(real, 3) cart ⇒ bool) c::(real, 3) cart ⇒ bool. polyhedron p ∧ bounded p ∧ IN (vec (0::nat)) (interior p) ∧ facet_of c p ⟶ (∃ b::((real, 3) cart ⇒ bool) ⇒ (real, 3) cart ⇒ bool. BIJ b (GSPEC (λGEN%PVAR%2092::(real, 3) cart ⇒ bool. ∃ e::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2092 (IN e (edges c)) e)) (GSPEC (λGEN%PVAR%2093::(real, 3) cart ⇒ bool. ∃ u::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2093 (facet_of u c) u)))

thm Counting_spheres.HYPERFACE_EXISTS:

∀ (P::(real, 3) cart ⇒ bool) U::(real, 3) cart ⇒ bool. bounded P ∧ polyhedron P ∧ IN (vec (0::nat)) (interior P) ∧ topological_component_yfan (vec (0::nat), vertices P, edges P) U ⟶ (∃!f::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ bool. IN f (face_set (hypermap1_of_fanx (vec (0::nat), vertices P, edges P))) ∧ dartset_leads_into_fan (vec (0::nat)) (vertices P) (edges P) f = U)

thm Counting_spheres.BIJ_DART_POLYEDGE:

∀ P::(real, 3) cart ⇒ bool. bounded P ∧ polyhedron P ∧ IN (vec (0::nat)) (interior P) ⟶ (∃ b::(real, 3) cart × (real, 3) cart × (real, 3) cart × (real, 3) cart ⇒ ((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool). BIJ b (dart (hypermap1_of_fanx (vec (0::nat), vertices P, edges P))) (GSPEC (λGEN%PVAR%2094::((real, 3) cart ⇒ bool) × ((real, 3) cart ⇒ bool). ∃ (e::(real, 3) cart ⇒ bool) f1::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2094 (facet_of e f1 ∧ facet_of f1 P) (e, f1))))

thm Counting_spheres.FINITE_EDGE:

∀ P::(real, ?'a::type) cart ⇒ bool. polyhedron P ∧ bounded P ⟶ (∀ f::(real, ?'a::type) cart ⇒ bool. facet_of f P ⟶ FINITE (GSPEC (λGEN%PVAR%2099::(real, ?'a::type) cart ⇒ bool. ∃ e::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%2099 (facet_of e f) e))) ∧ FINITE (GSPEC (λGEN%PVAR%2100::(real, ?'a::type) cart ⇒ bool. ∃ f::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%2100 (facet_of f P) f)) ∧ FINITE (GSPEC (λGEN%PVAR%2101::((real, ?'a::type) cart ⇒ bool) × ((real, ?'a::type) cart ⇒ bool). ∃ (f::(real, ?'a::type) cart ⇒ bool) e::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%2101 (facet_of f P ∧ facet_of e f) (f, e)))

thm Counting_spheres.polyhedron_sum_sum_edge:

∀ P::(real, 3) cart ⇒ bool. bounded P ∧ polyhedron P ⟶ sum (GSPEC (λGEN%PVAR%2109::(real, 3) cart ⇒ bool. ∃ f::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2109 (facet_of f P) f)) (λf::(real, 3) cart ⇒ bool. real_of_nat (CARD (GSPEC (λGEN%PVAR%2110::(real, 3) cart ⇒ bool. ∃ e::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2110 (facet_of e f) e)))) = real_of_nat (CARD (GSPEC (*λGEN%PVAR%2111*::((*real*, *3*) *cart* ⇒ *bool*) × ((*real*, *3*) *cart* ⇒ *bool*). ∃ (*f*::(*real*, *3*) *cart* ⇒ *bool*) *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2111* (*facet_of f P* ∧ *facet_of e f*) (*f*, *e*))))

thm Counting_spheres.polyhedron_edge_sum:

∀ (*P*::(*real*, *3*) *cart* ⇒ *bool*) *n*::*nat*. *bounded P* ∧ *polyhedron P* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ *HAS_SIZE* (*GSPEC* (*λGEN%PVAR%2117*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *f*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2117* (*facet_of f P*) *f*)) *n* ∧ (*2*::*nat*) ≤ *n* ⟶ *sum* (*GSPEC* (*λGEN%PVAR%2118*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *f*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2118* (*facet_of f P*) *f*)) (*λf*::(*real*, *3*) *cart* ⇒ *bool*. *real_of_nat* (*CARD* (*GSPEC* (*λGEN%PVAR%2119*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *e*::(*real*, *3*) *cart* ⇒ *bool*. *SETSPEC GEN%PVAR%2119* (*facet_of e f*) *e*)))) ≤ *real_of_nat* (*6*::*nat*) ∗ *real_of_nat n* − *real_of_nat* (*12*::*nat*)

thm Counting_spheres.RELATIVE_INTERIOR_POLYHEDRON_EXPLICIT_ALT:

∀ (*V*::*?′b*::*type* ⇒ *bool*) (*P*::(*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*h*::*?′b*::*type* ⇒ (*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*a*::*?′b*::*type* ⇒ (*real*, *?′a*::*type*) *cart*) (*b*::*?′b*::*type* ⇒ *real*) (*p*::(*real*, *?′a*::*type*) *cart*) *v0*::*?′b*::*type*. *FINITE V* ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ *h v* = *GSPEC* (*λGEN%PVAR%2122*::(*real*, *?′a*::*type*) *cart*. ∃ *p*::(*real*, *?′a*::*type*) *cart*. *SETSPEC GEN%PVAR%2122* (*dot* (*a v*) *p* ≤ *b v*) *p*)) ∧ *P* = *INTERS* (*IMAGE h V*) ∧ *IN v0 V* ∧ *dot* (*a v0*) *p* = *b v0* ∧ (∀ *w*::*?′b*::*type*. *IN w V* ∧ *w* ≠ *v0* ⟶ *dot* (*a w*) *p* < *b w*) ⟶ *IN p* (*relative_interior* (*HOL_Light_Import.INTER P* (*GSPEC* (*λGEN%PVAR%2123*::(*real*, *?′a*::*type*) *cart*. ∃ *p*::(*real*, *?′a*::*type*) *cart*. *SETSPEC GEN%PVAR%2123* (*dot* (*a v0*) *p* = *b v0*) *p*))))

thm Counting_spheres.FACET_RELEVANT:

∀ (*V*::*?′b*::*type* ⇒ *bool*) (*a*::*?′b*::*type* ⇒ (*real*, *?′a*::*type*) *cart*) (*b*::*?′b*::*type* ⇒ *real*) (*p*::(*real*, *?′a*::*type*) *cart*) *v0*::*?′b*::*type*. *FINITE V* ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ (*0*::*real*) < *b v*) ∧ *dot* (*a v0*) *p* = *b v0* ∧ *IN v0 V* ∧ (∀ *w*::*?′b*::*type*. *IN w V* ∧ *v0* ≠ *w* ⟶ *dot* (*a w*) *p* < *b w*) ⟶ (∃ *t*::*real*. *b v0* < *dot* (*a v0*) (% *t p*) ∧ (∀ *w*::*?′b*::*type*. *IN w V* ∧ *v0* ≠ *w* ⟶ *dot* (*a w*) (% *t p*) < *b w*))

thm Counting_spheres.FACET_OF_POLYHEDRON_EXPLICIT_ALT:

∀ (*V*::*?′b*::*type* ⇒ *bool*) (*P*::(*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*h*::*?′b*::*type* ⇒ (*real*, *?′a*::*type*) *cart* ⇒ *bool*) (*a*::*?′b*::*type* ⇒ (*real*, *?′a*::*type*) *cart*) *b*::*?′b*::*type* ⇒ *real*. *FINITE V* ∧ *IN* (*vec* (*0*::*nat*)) (*interior P*) ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ *h v* = *GSPEC* (*λGEN%PVAR%2135*::(*real*, *?′a*::*type*) *cart*. ∃ *p*::(*real*, *?′a*::*type*) *cart*. *SETSPEC GEN%PVAR%2135* (*dot* (*a v*) *p* ≤ *b v*) *p*)) ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ (*0*::*real*) < *b v*) ∧ *INTERS* (*IMAGE h V*) = *P* ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ *a v* ≠ *vec* (*0*::*nat*)) ∧ (∀ *v*::*?′b*::*type*. *IN v V* ⟶ (∃ *p*::(*real*, *?′a*::*type*) *cart*. *dot* (*a v*) *p* = *b v* ∧ (∀ *w*::*?′b*::*type*. *IN w V* ∧ *v* ≠ *w* ⟶ *dot* (*a w*) *p* < *b w*))) ⟶ *BIJ* (*λv*::*?′b*::*type*. *HOL_Light_Import.INTER P* (*GSPEC* (*λGEN%PVAR%2136*::(*real*, *?′a*::*type*) *cart*. ∃ *p*::(*real*, *?′a*::*type*) *cart*. *SETSPEC GEN%PVAR%2136* (*dot* (*a v*) *p* = *b v*) *p*))) *V* (*GSPEC* (*λGEN%PVAR%2137::(real, ?'a::type) cart ⇒ bool. ∃ c::(real, ?'a::type) cart ⇒ bool. SETSPEC GEN%PVAR%2137 (facet_of c P) c)*)

thm Counting_spheres.EXISTS_M_POLYHEDRON:

∀ (V::(real, 3) cart ⇒ bool) (theta::(real, 3) cart ⇒ real) (r::real) n::nat. SUBSET V ball_annulus ∧ packing V ∧ weakly_saturated V r (real_of_nat (2::nat) ∗ h0) ∧ HAS_SIZE V n ∧ V ≠ EMPTY ∧ (real_of_nat (2::nat) ≤ r ∧ r ≤ real_of_nat (2::nat) ∗ h0) ∧ (∀ (v::(real, 3) cart) w::(real, 3) cart. IN v V ∧ IN w V ∧ v ≠ w ⟶ theta v + theta w ≤ arcV (vec (0::nat)) v w) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (0::real) < theta v ∧ theta v < pi / real_of_nat (2::nat)) ⟶ (∃ (b::(real, 3) cart ⇒ real) (f::(real, 3) cart ⇒ (real, 3) cart ⇒ bool) (h::(real, 3) cart ⇒ (real, 3) cart ⇒ bool) P::(real, 3) cart ⇒ bool. (∀ v::(real, 3) cart. IN v V ⟶ h v = GSPEC (λGEN%PVAR%2146::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2146 (dot v p ≤ b v) p)) ∧ INTERS (IMAGE h V) = P ∧ (∀ v::(real, 3) cart. IN v V ⟶ (0::real) < b v) ∧ polyhedron P ∧ bounded P ∧ IN (vec (0::nat)) (interior P) ∧ BIJ f V (GSPEC (λGEN%PVAR%2147::(real, 3) cart ⇒ bool. ∃ c::(real, 3) cart ⇒ bool. SETSPEC GEN%PVAR%2147 (facet_of c P) c)) ∧ (∀ v::(real, 3) cart. IN v V ⟶ f v = HOL_Light_Import.INTER P (GSPEC (λGEN%PVAR%2148::(real, 3) cart. ∃ p::(real, 3) cart. SETSPEC GEN%PVAR%2148 (dot v p = b v) p))) ∧ (∀ v::(real, 3) cart. IN v V ⟶ b v = vector_norm v ∗ cos (theta v)) ∧ (∀ v::(real, 3) cart. IN v V ⟶ SUBSET (rcone_gt (vec (0::nat)) v (cos (theta v))) (fchanged (f v))) ∧ (∀ v::(real, 3) cart. IN v V ⟶ (0::real) < cos (theta v) ∧ cos (theta v) < (1::real)))

thm Counting_spheres.LMFUN_LE_1:

∀ h≥1::real. lmfun h ≤ (1::real)

thm Counting_spheres.LMFUN_INEQ_CENTER_IMP_13:

∀ V::(real, 3) cart ⇒ bool. FINITE V ∧ SUBSET V ball_annulus ∧ ¬ lmfun_ineq_center V ⟶ (13::nat) ≤ CARD V

thm Counting_spheres.LMFUN_INEQ_CENTER_SUBSET:

∀ (V::(real, ?'a::type) cart ⇒ bool) W::(real, ?'a::type) cart ⇒ bool. FINITE V ∧ SUBSET W V ∧ lmfun_ineq_center V ⟶ lmfun_ineq_center W

thm Counting_spheres.SATURATE_BALL_ANNULUS:

∀ (W::(real, 3) cart ⇒ bool) (S::(real, 3) cart ⇒ bool) r::real. packing W ∧ SUBSET W ball_annulus ∧ ¬ lmfun_ineq_center W ∧ SUBSET S W ∧ real_of_nat (2::nat) ≤ r ∧ r ≤ real_of_nat (2::nat) ∗ h0 ∧ (∀ (v::(real, 3) cart) w::(real, 3) cart. S v ∧ W w ∧ distance (v, w) < r ⟶ v = w) ⟶ (∃ V::(real, 3) cart ⇒ bool. SUBSET V ball_annulus ∧ packing V ∧ weakly_saturated V r (real_of_nat (2::nat) ∗ h0) ∧ FINITE V ∧ SUBSET W V ∧ (∀ (v::(real, 3) cart) w::(real, 3) cart. S v ∧ V w ∧ distance (v, w) < r ⟶ v = w) ∧ ¬ lmfun_ineq_center V ∧ (13::nat) ≤ CARD V)

thm Counting_spheres.POLYHEDRON_FACET_SUM_4Pi:

$\forall P{::}(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ P \wedge bounded\ P \wedge IN\ (vec\ (0{::}nat))\ (interior\ P) \longrightarrow sum\ (GSPEC\ (\lambda GEN\%PVAR\%2154{::}(real,\ 3)\ cart \Rightarrow bool.\ \exists\, c{::}(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2154\ (facet\_of\ c\ P)\ c))\ (\lambda c{::}(real,\ 3)\ cart \Rightarrow bool.\ sol\ (vec\ (0{::}nat))\ (fchanged\ c)) = real\_of\_nat\ (4{::}nat) * pi$

thm Counting_spheres.COSG:

$\forall h{::}real.\ -\ real\_of\_nat\ (2{::}nat) \leq h \wedge h \leq real\_of\_nat\ (2{::}nat) \wedge (?g{::}real) = acs\ (h\ /\ real\_of\_nat\ (2{::}nat)) - pi\ /\ real\_of\_nat\ (6{::}nat) \longrightarrow cos\ ?g = h * (sqrt\ (real\_of\_nat\ (3{::}nat))\ /\ real\_of\_nat\ (4{::}nat)) + sqrt\ ((1{::}real) - (h\ /\ real\_of\_nat\ (2{::}nat))^2)\ /\ real\_of\_nat\ (2{::}nat)$

thm Counting_spheres.FACET_FINITE:

$\forall (p{::}(real,\ 3)\ cart \Rightarrow bool)\ f{::}(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge facet\_of\ f\ p \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2158{::}(real,\ 3)\ cart \Rightarrow bool.\ \exists\, e{::}(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2158\ (facet\_of\ e\ f)\ e))$

thm Counting_spheres.BIJ_SUM:

$\forall (A{::}?'b{::}type \Rightarrow bool)\ (B{::}?'a{::}type \Rightarrow bool)\ (f{::}?'a{::}type \Rightarrow real)\ ab{::}?'b{::}type \Rightarrow ?'a{::}type.\ BIJ\ ab\ A\ B \longrightarrow sum\ A\ (f \circ ab) = sum\ B\ f$

thm Counting_spheres.CARD_AT_LEAST3:

$\forall (x{::}?'a{::}type)\ (y{::}?'a{::}type)\ (z{::}?'a{::}type)\ A{::}?'a{::}type \Rightarrow bool.\ FINITE\ A \wedge IN\ x\ A \wedge IN\ y\ A \wedge IN\ z\ A \wedge x \neq y \wedge y \neq z \wedge x \neq z \longrightarrow (3{::}nat) \leq CARD\ A$

thm Counting_spheres.polyhedron_3_facets:

$\forall p{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge int\ (1{::}nat) < aff\_dim\ p \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2161{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ \exists\, c{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2161\ (facet\_of\ c\ p)\ c)) \wedge (3{::}nat) \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2162{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ \exists\, c{::}(real,\ ?'a{::}type)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2162\ (facet\_of\ c\ p)\ c))$

thm Counting_spheres.facet_3_facets:

$\forall (p{::}(real,\ 3)\ cart \Rightarrow bool)\ f{::}(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge IN\ (vec\ (0{::}nat))\ (interior\ p) \wedge facet\_of\ f\ p \longrightarrow FINITE\ (GSPEC\ (\lambda GEN\%PVAR\%2164{::}(real,\ 3)\ cart \Rightarrow bool.\ \exists\, e{::}(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2164\ (facet\_of\ e\ f)\ e)) \wedge (3{::}nat) \leq CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2165{::}(real,\ 3)\ cart \Rightarrow bool.\ \exists\, e{::}(real,\ 3)\ cart \Rightarrow bool.\ SETSPEC\ GEN\%PVAR\%2165\ (facet\_of\ e\ f)\ e))$

thm Counting_spheres.YSSKQOY_VECTOR:

$\forall (v{::}(real,\ 3)\ cart)\ (w{::}(real,\ 3)\ cart)\ theta{::}(real,\ 3)\ cart \Rightarrow real.\ IN\ v\ ball\_annulus \wedge IN\ w\ ball\_annulus \wedge v \neq w \wedge real\_of\_nat\ (2{::}nat) \leq distance\ (v,\ w) \wedge (\lambda v{::}(real,\ 3)\ cart.\ acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4{::}nat)) - pi\ /\ real\_of\_nat\ (6{::}nat)) = theta \longrightarrow theta\ v + theta\ w \leq arcV\ (vec\ (0{::}nat))\ v\ w$

thm Counting_spheres.PACK_INEQ_DEF_A_797:

$\forall (v::(real,\ 3)\ cart)\ v0::(real,\ 3)\ cart.\ pack\_ineq\_def\_a \wedge vector\_norm\ v0 = real\_of\_nat\ (2::nat) \wedge real\_of\_nat\ (2::nat) * h0 \leq distance\ (v,\ v0) \wedge real\_of\_nat\ (2::nat) \leq vector\_norm\ v \wedge vector\_norm\ v \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow DECIMAL\ (797::nat)\ (1000::nat) + (acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) < arclength\ (vector\_norm\ v)\ (real\_of\_nat\ (2::nat))\ (distance\ (v,\ v0))$

thm Counting_spheres.YSSKQOY_VECTOR2:

$\forall (v0::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ IN\ v0\ ball\_annulus \wedge IN\ w\ ball\_annulus \wedge w \neq v0 \wedge real\_of\_nat\ (2::nat) * h0 \leq distance\ (w,\ v0) \wedge pack\_ineq\_def\_a \wedge vector\_norm\ v0 = real\_of\_nat\ (2::nat) \longrightarrow DECIMAL\ (797::nat)\ (1000::nat) + (acs\ (vector\_norm\ w\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) \leq arcV\ (vec\ (0::nat))\ v0\ w$

thm Counting_spheres.YSSKQOY_VECTOR2_ALT:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ (w::(real,\ 3)\ cart)\ (v0::(real,\ 3)\ cart)\ theta::(real,\ 3)\ cart \Rightarrow real.\ SUBSET\ V\ ball\_annulus \wedge packing\ V \wedge IN\ v\ V \wedge IN\ w\ V \wedge IN\ v0\ V \wedge v \neq w \wedge (\forall w::(real,\ 3)\ cart.\ IN\ w\ V \wedge w \neq v0 \longrightarrow real\_of\_nat\ (2::nat) * h0 \leq distance\ (w,\ v0)) \wedge pack\_ineq\_def\_a \wedge vector\_norm\ v0 = real\_of\_nat\ (2::nat) \wedge (\lambda v::(real,\ 3)\ cart.\ if\ v = v0\ then\ DECIMAL\ (797::nat)\ (1000::nat)\ else\ acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) = theta \longrightarrow theta\ v + theta\ w \leq arcV\ (vec\ (0::nat))\ v\ w$

thm Counting_spheres.ACS_ROOT32:

$acs\ (sqrt\ (real\_of\_nat\ (3::nat))\ /\ real\_of\_nat\ (2::nat)) = pi\ /\ real\_of\_nat\ (6::nat)$

thm Counting_spheres.ASN_HALF:

$asn\ ((1::real)\ /\ real\_of\_nat\ (2::nat)) = pi\ /\ real\_of\_nat\ (6::nat)$

thm Counting_spheres.THETA_BOUNDS:

$\forall (v::(real,\ 3)\ cart)\ theta::(real,\ 3)\ cart \Rightarrow real.\ IN\ v\ ball\_annulus \wedge (\lambda v::(real,\ 3)\ cart.\ acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) = theta \longrightarrow (0::real) < theta\ v \wedge theta\ v < pi\ /\ real\_of\_nat\ (2::nat)$

thm Counting_spheres.INJ_FINITE_EXISTS:

$\forall (n::nat)\ (A::?'b::type \Rightarrow bool)\ B::?'a::type \Rightarrow bool.\ HAS\_SIZE\ A\ n \wedge FINITE\ B \wedge n \leq CARD\ B \longrightarrow (\exists j::?'b::type \Rightarrow ?'a::type.\ INJ\ j\ A\ B)$

thm Counting_spheres.INJ_EXTENSION:

$\forall (A::?'b::type \Rightarrow bool)\ (B::?'a::type \Rightarrow bool)\ (A'::?'b::type \Rightarrow bool)\ j'::?'b::type \Rightarrow ?'a::type.\ INJ\ j'\ A'\ B \wedge SUBSET\ A'\ A \wedge FINITE\ A \wedge FINITE\ B \wedge CARD\ A \leq CARD\ B \longrightarrow (\exists j::?'b::type \Rightarrow ?'a::type.\ INJ\ j\ A\ B \wedge (\forall a::?'b::type.\ IN\ a\ A' \longrightarrow j\ a = j'\ a))$

thm Counting_spheres.BIJ_EXTENDS_INJ:

$\forall (A::?'b::type \Rightarrow bool)\ (B::?'a::type \Rightarrow bool)\ (A'::?'b::type \Rightarrow bool)\ j'::?'b::type \Rightarrow ?'a::type.\ FINITE\ A \wedge FINITE\ B \wedge SUBSET\ A'\ A \wedge INJ\ j'\ A'\ B \wedge CARD\ A = CARD\ B \longrightarrow (\exists j::?'b::type \Rightarrow ?'a::type.\ BIJ\ j\ A\ B \wedge (\forall a::?'b::type.\ IN\ a\ A' \longrightarrow j'\ a = j\ a))$

thm Counting_spheres.DLWCHEM_VECTOR_sum:

$\forall (k::(real,\ 3)\ cart \Rightarrow nat)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (n::nat)\ theta::(real,\ 3)\ cart \Rightarrow real.\ pack\_ineq\_def\_a \wedge (\lambda v::(real,\ 3)\ cart.\ acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) = theta \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (3::nat) \leq k\ v) \wedge (12::nat) < n \wedge HAS\_SIZE\ V\ n \wedge SUBSET\ V\ ball\_annulus \wedge sum\ V\ (\lambda v::(real,\ 3)\ cart.\ real\_of\_nat\ (k\ v)) \leq real\_of\_nat\ (6::nat) * real\_of\_nat\ n - real\_of\_nat\ (12::nat) \wedge sum\ V\ (\lambda v::(real,\ 3)\ cart.\ max\ (0::real)\ (regular\_spherical\_polygon\_area\ (cos\ (theta\ v))\ (real\_of\_nat\ (k\ v)))) \leq real\_of\_nat\ (4::nat) * pi \wedge \neg\ lmfun\_ineq\_center\ V \longrightarrow n < (16::nat)$

thm Counting_spheres.XULJEPR_VECTOR_sum:

$\forall (k::(real,\ 3)\ cart \Rightarrow nat)\ (V::(real,\ 3)\ cart \Rightarrow bool)\ (n::nat)\ (theta::(real,\ 3)\ cart \Rightarrow real)\ v0::(real,\ 3)\ cart.\ pack\_ineq\_def\_a \wedge IN\ v0\ V \wedge (\lambda v::(real,\ 3)\ cart.\ if\ v = v0\ then\ DECIMAL\ (797::nat)\ (1000::nat)\ else\ acs\ (vector\_norm\ v\ /\ real\_of\_nat\ (4::nat)) - pi\ /\ real\_of\_nat\ (6::nat)) = theta \wedge (12::nat) < n \wedge vector\_norm\ v0 = real\_of\_nat\ (2::nat) \wedge (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow (3::nat) \leq k\ v) \wedge HAS\_SIZE\ V\ n \wedge SUBSET\ V\ ball\_annulus \wedge sum\ V\ (\lambda v::(real,\ 3)\ cart.\ real\_of\_nat\ (k\ v)) \leq real\_of\_nat\ (6::nat) * real\_of\_nat\ n - real\_of\_nat\ (12::nat) \wedge sum\ V\ (\lambda v::(real,\ 3)\ cart.\ max\ (0::real)\ (regular\_spherical\_polygon\_area\ (cos\ (theta\ v))\ (real\_of\_nat\ (k\ v)))) \leq real\_of\_nat\ (4::nat) * pi \wedge \neg\ lmfun\_ineq\_center\ V \longrightarrow False$

thm Counting_spheres.SOL_NN:

$\forall (x::(real,\ 3)\ cart)\ U::(real,\ 3)\ cart \Rightarrow bool.\ (\exists r>0::real.\ measurable\ (HOL\_Light\_Import.INTER\ U\ (normball\ x\ r)) \wedge radial\_norm\ r\ x\ (HOL\_Light\_Import.INTER\ U\ (normball\ x\ r))) \longrightarrow (0::real) \leq sol\ x\ U$

thm Counting_spheres.FACET_SOL_NN:

$\forall (p::(real,\ 3)\ cart \Rightarrow bool)\ c::(real,\ 3)\ cart \Rightarrow bool.\ polyhedron\ p \wedge bounded\ p \wedge IN\ (vec\ (0::nat))\ (interior\ p) \wedge facet\_of\ c\ p \longrightarrow (0::real) \leq sol\ (vec\ (0::nat))\ (fchanged\ c)$

thm Counting_spheres.DLWCHEM:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge pack\_ineq\_def\_a \wedge SUBSET\ V\ ball\_annulus \wedge \neg\ lmfun\_ineq\_center\ V \longrightarrow CARD\ V = (13::nat) \vee CARD\ V = (14::nat) \vee CARD\ V = (15::nat)$

thm Counting_spheres.XULJEPR:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \wedge pack\_ineq\_def\_a \wedge (\exists v::(real,\ 3)\ cart.\ IN\ v\ V \wedge vector\_norm\ v = real\_of\_nat\ (2::nat) \wedge$

$(\forall\, u{::}(real,\ 3)\ cart.\ IN\ u\ V\ \wedge\ u \neq v \longrightarrow real\_of\_nat\ (2{::}nat) * h0 \leq distance\ (u,\ v))) \longrightarrow lmfun\_ineq\_center\ V$

thm Dih2k_hypermap.cstab:

$cstab = DECIMAL\ (301{::}nat)\ (100{::}nat)$

thm Dih2k_hypermap.tau_fun:

$\forall\,(V{::}?'a{::}type)\ (E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ f{::}(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ tau\_fun\ V\ E\ f = sum\ f\ (\lambda e{::}(real,\ 3)\ cart \times (real,\ 3)\ cart.\ rho\_fun\ (vector\_norm\ (fst\ e)) * azim\_in\_fan\ e\ E) - (pi + sol0) * real\_of\_nat\ (CARD\ f - (2{::}nat))$

thm DEF_standard:

$standard = (\lambda(\_6788701{::}(real,\ ?'a{::}type)\ cart)\ \_6788702{::}(real,\ ?'a{::}type)\ cart.\ real\_of\_nat\ (2{::}nat) \leq vector\_norm\ (vector\_sub\ \_6788701\ \_6788702) \wedge vector\_norm\ (vector\_sub\ \_6788701\ \_6788702) \leq real\_of\_nat\ (2{::}nat) * h0)$

thm Dih2k_hypermap.standard:

$\forall\,(v{::}(real,\ ?'a{::}type)\ cart)\ w{::}(real,\ ?'a{::}type)\ cart.\ standard\ v\ w = (real\_of\_nat\ (2{::}nat) \leq vector\_norm\ (vector\_sub\ v\ w) \wedge vector\_norm\ (vector\_sub\ v\ w) \leq real\_of\_nat\ (2{::}nat) * h0)$

thm DEF_protracted:

$protracted = (\lambda(\_6788713{::}(real,\ ?'a{::}type)\ cart)\ \_6788714{::}(real,\ ?'a{::}type)\ cart.\ real\_of\_nat\ (2{::}nat) * h0 \leq vector\_norm\ (vector\_sub\ \_6788713\ \_6788714) \wedge vector\_norm\ (vector\_sub\ \_6788713\ \_6788714) \leq sqrt\ (real\_of\_nat\ (8{::}nat)))$

thm Dih2k_hypermap.protracted:

$\forall\,(v{::}(real,\ ?'a{::}type)\ cart)\ w{::}(real,\ ?'a{::}type)\ cart.\ protracted\ v\ w = (real\_of\_nat\ (2{::}nat) * h0 \leq vector\_norm\ (vector\_sub\ v\ w) \wedge vector\_norm\ (vector\_sub\ v\ w) \leq sqrt\ (real\_of\_nat\ (8{::}nat)))$

thm DEF_diagonal0:

$diagonal0 = (\lambda(\_6788725{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (\_6788726{::}(real,\ 3)\ cart)\ \_6788727{::}(real,\ 3)\ cart.\ \_6788726 \neq \_6788727 \wedge \neg\ IN\ (INSERT\ \_6788726\ (INSERT\ \_6788727\ EMPTY))\ \_6788725)$

thm Dih2k_hypermap.diagonal0:

$\forall\,(v{::}(real,\ 3)\ cart)\ (w{::}(real,\ 3)\ cart)\ E{::}((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ diagonal0\ E\ v\ w = (v \neq w \wedge \neg\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)$

thm DEF_diagonal1:

$diagonal1 = (\lambda\_6788746{::}((real,\ ?'a{::}type)\ cart \Rightarrow bool) \times (((real,\ ?'a{::}type)\ cart \Rightarrow bool) \Rightarrow bool).\ \forall\,(v{::}(real,\ ?'a{::}type)\ cart)\ w{::}(real,\ ?'a{::}type)\ cart.\ v \neq w \wedge IN\ v\ (fst\ \_6788746) \wedge IN\ w\ (fst\ \_6788746) \wedge \neg\ IN\ (INSERT\ v\ (INSERT$

$w\ EMPTY))\ (snd\ \_6788746) \longrightarrow real\_of\_nat\ (2::nat) * h0 \leq vector\_norm\ (vector\_sub\ v\ w))$

thm Dih2k_hypermap.diagonal1:

$\forall (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool.\ diagonal1\ (V,\ E) = (\forall (v::(real,\ ?'a::type)\ cart)\ w::(real,\ ?'a::type)\ cart.\ v \neq w \wedge IN\ v\ V \wedge IN\ w\ V \wedge \neg\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E \longrightarrow real\_of\_nat\ (2::nat) * h0 \leq vector\_norm\ (vector\_sub\ v\ w))$

thm DEF_main_estimate:

$main\_estimate = (\lambda\_6788755::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \times ((real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool).\ convex\_local\_fan\ (fst\ \_6788755,\ fst\ (snd\ \_6788755),\ snd\ (snd\ \_6788755)) \wedge packing\ (fst\ \_6788755) \wedge SUBSET\ (fst\ \_6788755)\ ball\_annulus \wedge diagonal1\ (fst\ \_6788755,\ fst\ (snd\ \_6788755)) \wedge CARD\ (fst\ (snd\ \_6788755)) = CARD\ (snd\ (snd\ \_6788755)) \wedge (3::nat) \leq CARD\ (fst\ (snd\ \_6788755)) \wedge CARD\ (fst\ (snd\ \_6788755)) \leq (6::nat))$

thm Dih2k_hypermap.main_estimate:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (f::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool.\ main\_estimate\ (V,\ E,\ f) = (convex\_local\_fan\ (V,\ E,\ f) \wedge packing\ V \wedge SUBSET\ V\ ball\_annulus \wedge diagonal1\ (V,\ E) \wedge CARD\ E = CARD\ f \wedge (3::nat) \leq CARD\ E \wedge CARD\ E \leq (6::nat))$

thm DEF_torsor:

$torsor = (\lambda(\_6788768::?'a::type \Rightarrow bool)\ (\_6788769::nat)\ \_6788770::?'a::type \Rightarrow ?'a::type.\ (\forall x::?'a::type.\ IN\ x\ \_6788768 \longrightarrow IN\ (\_6788770\ x)\ \_6788768) \wedge (\forall (x1::?'a::type)\ x2::?'a::type.\ IN\ x1\ \_6788768 \wedge IN\ x2\ \_6788768 \wedge \_6788770\ x1 = \_6788770\ x2 \longrightarrow x1 = x2) \wedge (\forall (i::nat)\ x::?'a::type.\ (0::nat) < i \wedge i < \_6788769 \wedge IN\ x\ \_6788768 \longrightarrow POWER\ \_6788770\ i\ x \neq x) \wedge (\forall x::?'a::type.\ IN\ x\ \_6788768 \longrightarrow POWER\ \_6788770\ \_6788769\ x = x) \wedge HAS\_SIZE\ \_6788768\ \_6788769)$

thm Dih2k_hypermap.torsor:

$\forall (f::?'a::type \Rightarrow ?'a::type)\ (s::?'a::type \Rightarrow bool)\ k::nat.\ torsor\ s\ k\ f = ((\forall x::?'a::type.\ IN\ x\ s \longrightarrow IN\ (f\ x)\ s) \wedge (\forall (x1::?'a::type)\ x2::?'a::type.\ IN\ x1\ s \wedge IN\ x2\ s \wedge f\ x1 = f\ x2 \longrightarrow x1 = x2) \wedge (\forall (i::nat)\ x::?'a::type.\ (0::nat) < i \wedge i < k \wedge IN\ x\ s \longrightarrow POWER\ f\ i\ x \neq x) \wedge (\forall x::?'a::type.\ IN\ x\ s \longrightarrow POWER\ f\ k\ x = x) \wedge HAS\_SIZE\ s\ k)$

thm DEF_constraint_system:

$constraint\_system = (\lambda(\_6788789::nat)\ (\_6788790::?'b::type)\ (\_6788791::?'a::type \Rightarrow bool)\ (\_6788792::?'a::type \times ?'a::type \Rightarrow real)\ (\_6788793::?'a::type \times ?'a::type \Rightarrow real)\ (\_6788794::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_6788795::?'a::type \Rightarrow ?'a::type.\ (3::nat) \leq \_6788789 \wedge \_6788789 \leq (6::nat) \wedge torsor\ \_6788791\ \_6788789\ \_6788795 \wedge (\forall (i::?'a::type)\ j::?'a::type.\ \_6788792\ (i,\ j) = \_6788792\ (j,\ i)$

$\wedge$ *_6788793* $(i, j) =$ *_6788793* $(j, i) \wedge$ *_6788792* $(i, j) \leq$ *_6788793* $(i, j)) \wedge$ $(\forall (i::?'a::type)\ j::?'a::type.$ *_6788792* $(i, j) =$ *_6788792* $(i,$ *POWER _6788795 _6788789* $j) \wedge$ *_6788793* $(i, j) =$ *_6788793* $(i,$ *POWER _6788795 _6788789* $j)) \wedge$ *SUBSET _6788794* (*GSPEC* (λ*GEN%PVAR%2176*::?'a::type $\Rightarrow$ *bool*. $\exists\, i$::?'a::type. *SETSPEC GEN%PVAR%2176* (*IN i _6788791*) (*INSERT i* (*INSERT* (*_6788795 i*) *EMPTY*)))) $\wedge$ *CARD _6788794* + *_6788789* $\leq$ (*6*::*nat*))

thm Dih2k_hypermap.constraint_system:

$\forall (d::?'b::type)\ (a::?'a::type \times ?'a::type \Rightarrow real)\ (b::?'a::type \times ?'a::type \Rightarrow real)\ (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ (J::(?'a::type \Rightarrow bool) \Rightarrow bool)\ k::nat.$ *constraint_system k d s a b J f* $= ((3::nat) \leq k \wedge k \leq (6::nat) \wedge$ *torsor s k f* $\wedge (\forall (i::?'a::type)\ j::?'a::type.\ a\ (i, j) = a\ (j, i) \wedge b\ (i, j) = b\ (j, i) \wedge a\ (i, j) \leq b\ (i, j)) \wedge (\forall (i::?'a::type)\ j::?'a::type.\ a\ (i, j) = a\ (i,$ *POWER f k j*$) \wedge b\ (i, j) = b\ (i,$ *POWER f k j*$)) \wedge$ *SUBSET J* (*GSPEC* (λ*GEN%PVAR%2176*::?'a::type $\Rightarrow$ *bool*. $\exists\, i$::?'a::type. *SETSPEC GEN%PVAR%2176* (*IN i s*) (*INSERT i* (*INSERT* (*f i*) *EMPTY*)))) $\wedge$ *CARD J* + $k \leq (6::nat))$

thm DEF_stable_system:

stable_system = (λ(*_6788866*::*nat*) (*_6788867*::?'b::type) (*_6788868*::?'a::type $\Rightarrow$ *bool*) (*_6788869*::?'a::type $\times$?'a::type $\Rightarrow$ *real*) (*_6788870*::?'a::type $\times$?'a::type $\Rightarrow$ *real*) (*_6788871*::(?'a::type $\Rightarrow$ *bool*) $\Rightarrow$ *bool*) *_6788872*::?'a::type $\Rightarrow$?'a::type. *constraint_system _6788866 _6788867 _6788868 _6788869 _6788870 _6788871 _6788872* $\wedge$ ($\forall$ (*i*::?'a::type) *j*::?'a::type. *IN i _6788868* $\wedge$ *IN j _6788868* $\wedge$ $i \neq j \longrightarrow$ *real_of_nat* (*2*::*nat*) $\leq$ *_6788869* $(i, j) \wedge$ *_6788869* $(i, j) \leq$ *cstab*) $\wedge$ ($\forall$ *i*::?'a::type. *IN i _6788868* $\longrightarrow$ *_6788869* $(i, i) = (0::real) \wedge$ *_6788870* (*i*, *_6788872 i*) $\leq$ *cstab*) $\wedge$ ($\forall$ (*i*::?'a::type) *j*::?'a::type. *IN* (*INSERT i* (*INSERT j EMPTY*)) *_6788871* $\longrightarrow$ *_6788869* $(i, j) =$ *sqrt* (*real_of_nat* (*8*::*nat*)) $\wedge$ *_6788870* $(i, j) =$ *cstab*))

thm Dih2k_hypermap.stable_system:

$\forall (k::nat)\ (d::?'b::type)\ (s::?'a::type \Rightarrow bool)\ (f::?'a::type \Rightarrow ?'a::type)\ (J::(?'a::type \Rightarrow bool) \Rightarrow bool)\ (a::?'a::type \times ?'a::type \Rightarrow real)\ b::?'a::type \times ?'a::type \Rightarrow real.$ *stable_system k d s a b J f* = (*constraint_system k d s a b J f* $\wedge (\forall (i::?'a::type)\ j::?'a::type.$ *IN i s* $\wedge$ *IN j s* $\wedge i \neq j \longrightarrow$ *real_of_nat* $(2::nat) \leq a\ (i, j) \wedge a\ (i, j) \leq$ *cstab*$) \wedge (\forall i::?'a::type.$ *IN i s* $\longrightarrow a\ (i, i) = (0::real) \wedge b\ (i, f\, i) \leq$ *cstab*$) \wedge (\forall (i::?'a::type)\ j::?'a::type.$ *IN* (*INSERT i* (*INSERT j EMPTY*)) $J \longrightarrow a\ (i, j) =$ *sqrt* (*real_of_nat* $(8::nat)) \wedge b\ (i, j) =$ *cstab*))

thm DEF_V_SY:

V_SY = *rows*

thm Dih2k_hypermap.V_SY:

$\forall v$::((*real*, ?'b::type) *cart*, ?'a::type) *cart*. *V_SY v* = *rows v*

thm DEF_E_SY:

$E\_SY = (\lambda\_6788948::((real, ?'b::type) cart, ?'a::type) cart. GSPEC (\lambda GEN\%PVAR\%2177::(real, ?'b::type) cart \Rightarrow bool. \exists i::nat. SETSPEC GEN\%PVAR\%2177 ((1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV) (INSERT (row i \_6788948) (INSERT (row (Suc (i mod dimindex HOL\_Light\_Import.UNIV)) \_6788948) EMPTY))))$

thm Dih2k_hypermap.E_SY:

$\forall v::((real, ?'b::type) cart, ?'a::type) cart. E\_SY v = GSPEC (\lambda GEN\%PVAR\%2177::(real, ?'b::type) cart \Rightarrow bool. \exists i::nat. SETSPEC GEN\%PVAR\%2177 ((1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV) (INSERT (row i v) (INSERT (row (Suc (i mod dimindex HOL\_Light\_Import.UNIV)) v) EMPTY)))$

thm DEF_F_SY:

$F\_SY = (\lambda\_6788953::((real, ?'b::type) cart, ?'a::type) cart. GSPEC (\lambda GEN\%PVAR\%2178::(real, ?'b::type) cart \times (real, ?'b::type) cart. \exists i::nat. SETSPEC GEN\%PVAR\%2178 ((1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV) (row i \_6788953, row (Suc (i mod dimindex HOL\_Light\_Import.UNIV)) \_6788953)))$

thm Dih2k_hypermap.F_SY:

$\forall v::((real, ?'b::type) cart, ?'a::type) cart. F\_SY v = GSPEC (\lambda GEN\%PVAR\%2178::(real, ?'b::type) cart \times (real, ?'b::type) cart. \exists i::nat. SETSPEC GEN\%PVAR\%2178 ((1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV) (row i v, row (Suc (i mod dimindex HOL\_Light\_Import.UNIV)) v))$

thm DEF_CONDITION1_SY:

$CONDITION1\_SY = (\lambda(\_6788958::nat \times nat \Rightarrow real) (\_6788959::nat \times nat \Rightarrow real) \_6788960::((real, ?'b::type) cart, ?'a::type) cart. \forall (i::nat) j::nat. (1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow \_6788958 (i, j) \leq vector\_norm (vector\_sub (row i \_6788960) (row j \_6788960)) \wedge vector\_norm (vector\_sub (row i \_6788960) (row j \_6788960)) \leq \_6788959 (i, j))$

thm Dih2k_hypermap.CONDITION1_SY:

$\forall (a::nat \times nat \Rightarrow real) (v::((real, ?'b::type) cart, ?'a::type) cart) b::nat \times nat \Rightarrow real. CONDITION1\_SY a b v = (\forall (i::nat) j::nat. (1::nat) \leq i \wedge i \leq dimindex HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex HOL\_Light\_Import.UNIV \longrightarrow a (i, j) \leq vector\_norm (vector\_sub (row i v) (row j v)) \wedge vector\_norm (vector\_sub (row i v) (row j v)) \leq b (i, j))$

thm DEF_CONDITION2_SY:

$CONDITION2\_SY = (\lambda\_6788979::((real, 3) cart, ?'a::type) cart. convex\_local\_fan (V\_SY \_6788979, E\_SY \_6788979, F\_SY \_6788979))$

thm Dih2k_hypermap.CONDITION2_SY:

$\forall v::((real, 3) cart, ?'a::type) cart. CONDITION2\_SY v = convex\_local\_fan (V\_SY v, E\_SY v, F\_SY v)$

thm TYDEF_finite_product:

*mk_finite_product (dest_finite_product (?a::(?'b::type, ?'a::type) finite_product)) = ?a ∧ IN (?r::nat) (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV * dimindex HOL_Light_Import.UNIV)) = (dest_finite_product (mk_finite_product ?r) = ?r)*

thm Dih2k_hypermap.finite_product_tybij_conjunct1:

∀ *r::nat. IN r (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV * dimindex HOL_Light_Import.UNIV)) = (dest_finite_product (mk_finite_product r) = r)*

thm Dih2k_hypermap.finite_product_tybij_conjunct0:

∀ *a::(?'b::type, ?'a::type) finite_product. mk_finite_product (dest_finite_product a) = a*

thm Dih2k_hypermap.finite_product_tybij:

*(∀ a::(?'b::type, ?'a::type) finite_product. mk_finite_product (dest_finite_product a) = a) ∧ (∀ r::nat. IN r (dotdot (1::nat) (dimindex HOL_Light_Import.UNIV * dimindex HOL_Light_Import.UNIV)) = (dest_finite_product (mk_finite_product r) = r))*

thm DEF_matvec:

matvec = (λ_6788984::((?'c::type, ?'b::type) cart, ?'a::type) cart. lambda (λi::nat. $ ($ _6788984 (if i mod dimindex HOL_Light_Import.UNIV = (0::nat) then i div dimindex HOL_Light_Import.UNIV else Suc (i div dimindex HOL_Light_Import.UNIV))) (if i mod dimindex HOL_Light_Import.UNIV = (0::nat) then dimindex HOL_Light_Import.UNIV else i mod dimindex HOL_Light_Import.UNIV)))

thm Dih2k_hypermap.matvec:

∀ *f::((?'c::type, ?'b::type) cart, ?'a::type) cart. matvec f = lambda (λi::nat. $ ($ f (if i mod dimindex HOL_Light_Import.UNIV = (0::nat) then i div dimindex HOL_Light_Import.UNIV else Suc (i div dimindex HOL_Light_Import.UNIV))) (if i mod dimindex HOL_Light_Import.UNIV = (0::nat) then dimindex HOL_Light_Import.UNIV else i mod dimindex HOL_Light_Import.UNIV))*

thm DEF_vecmat:

*vecmat = (λ(_6788989::nat) _6788990::(?'c::type, (?'b::type, ?'a::type) finite_product) cart. lambda (λi::nat. $ _6788990 (_6788989 * dimindex HOL_Light_Import.UNIV + i)))*

thm Dih2k_hypermap.vecmat:

∀ *(f::(?'c::type, (?'b::type, ?'a::type) finite_product) cart) j::nat. vecmat j f = lambda (λi::nat. $ f (j * dimindex HOL_Light_Import.UNIV + i))*

thm DEF_vecmats:

*vecmats = (λ_6789001::(?'c::type, (?'b::type, ?'a::type) finite_product) cart. lambda (λi::nat. lambda (λj::nat. $ _6789001 ((i − (1::nat)) * dimindex HOL_Light_Import.UNIV + j))))*

thm Dih2k_hypermap.vecmats:

$\forall f::(?'c::type, (?'b::type, ?'a::type)\ finite\_product)\ cart.\ vecmats\ f = lambda\ (\lambda i::nat.\ lambda\ (\lambda j::nat.\ \$\ f\ ((i - (1::nat)) * dimindex\ HOL\_Light\_Import.UNIV + j)))$

thm DEF_B_SY1:

$B\_SY1 = (\lambda(\_6789006::nat \times nat \Rightarrow real)\ \_6789007::nat \times nat \Rightarrow real.\ GSPEC\ (\lambda GEN\%PVAR\%2179::(real, (?'a::type, 3)\ finite\_product)\ cart.\ \exists v::((real, 3)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2179\ ((\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (row\ i\ v)\ ball\_annulus) \wedge CONDITION1\_SY\ \_6789006\ \_6789007\ v \wedge CONDITION2\_SY\ v)\ (matvec\ v)))$

thm Dih2k_hypermap.B_SY1:

$\forall (a::nat \times nat \Rightarrow real)\ b::nat \times nat \Rightarrow real.\ B\_SY1\ a\ b = GSPEC\ (\lambda GEN\%PVAR\%2179::(real, (?'a::type, 3)\ finite\_product)\ cart.\ \exists v::((real, 3)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2179\ ((\forall i::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (row\ i\ v)\ ball\_annulus) \wedge CONDITION1\_SY\ a\ b\ v \wedge CONDITION2\_SY\ v)\ (matvec\ v))$

thm Dih2k_hypermap.FINITE_PRODUCT_IMAGE:

$HOL\_Light\_Import.UNIV = IMAGE\ mk\_finite\_product\ (dotdot\ (1::nat)\ (dimindex\ HOL\_Light\_Import.UNIV * dimindex\ HOL\_Light\_Import.UNIV))$

thm Dih2k_hypermap.DIMINDEX_HAS_SIZE_FINITE_PRODUCT:

$HAS\_SIZE\ HOL\_Light\_Import.UNIV\ (dimindex\ HOL\_Light\_Import.UNIV * dimindex\ HOL\_Light\_Import.UNIV)$

thm Dih2k_hypermap.DIMINDEX_FINITE_PRODUCT:

$dimindex\ HOL\_Light\_Import.UNIV = dimindex\ HOL\_Light\_Import.UNIV * dimindex\ HOL\_Light\_Import.UNIV$

thm Dih2k_hypermap.INDEX_VECMAT:

$(?i::nat) < dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?i'::nat) \wedge ?i' \leq dimindex\ HOL\_Light\_Import.UNIV \longrightarrow (1::nat) \leq ?i * dimindex\ HOL\_Light\_Import.UNIV + ?i' \wedge ?i * dimindex\ HOL\_Light\_Import.UNIV + ?i' \leq dimindex\ HOL\_Light\_Import.UNIV$

thm Dih2k_hypermap.VECMAT_ROW:

$(?i::nat) < dimindex\ HOL\_Light\_Import.UNIV \longrightarrow vecmat\ ?i\ (matvec\ (?f::((real, ?'a::type)\ cart, ?'b::type)\ cart)) = row\ (Suc\ ?i)\ ?f$

thm Dih2k_hypermap.VECMATS_MATVEC_ID:

$vecmats\ (matvec\ (?A::((real, ?'b::type)\ cart, ?'a::type)\ cart)) = ?A$

thm Dih2k_hypermap.MATVEC_VECMATS_ID:

$matvec\ (vecmats\ (?A::(real, (?'b::type, ?'a::type)\ finite\_product)\ cart)) = ?A$

thm Dih2k_hypermap.LINEAR_VECMAT:

linear (vecmat (?i::nat))

thm Dih2k_hypermap.VECMAT_VEC:

∀ *(i::nat) n::nat. i* < *dimindex HOL_Light_Import.UNIV* ⟶ *vecmat i (vec n)* = *vec n*

thm Dih2k_hypermap.VECMAT_ADD:

∀ *(i::nat) (x::(real, (?'b::type, ?'a::type) finite_product) cart) y::(real, (?'b::type, ?'a::type) finite_product) cart. vecmat i (vector_add x y)* = *vector_add (vecmat i x) (vecmat i y)*

thm Dih2k_hypermap.VECMAT_CMUL:

∀ *(i::nat) (x::(real, (?'b::type, ?'a::type) finite_product) cart) c::real. vecmat i* (% *c x*) = % *c (vecmat i x)*

thm Dih2k_hypermap.VECMAT_NEG:

∀ *(i::nat) x::(real, (?'b::type, ?'a::type) finite_product) cart. vector_neg (vecmat i x)* = *vecmat i (vector_neg x)*

thm Dih2k_hypermap.VECMAT_SUB:

∀ *(i::nat) (x::(real, (?'b::type, ?'a::type) finite_product) cart) y::(real, (?'b::type, ?'a::type) finite_product) cart. vecmat i (vector_sub x y)* = *vector_sub (vecmat i x) (vecmat i y)*

thm Dih2k_hypermap.FSTCART_VSUM:

∀ *(k::?'c::type* ⇒ *bool) (x::?'c::type* ⇒ *(real, (?'b::type, ?'a::type) finite_product) cart) i::nat. FINITE k* ⟶ *i* < *dimindex HOL_Light_Import.UNIV* ⟶ *vecmat i (vsum k x)* = *vsum k (λi'::?'c::type. vecmat i (x i'))*

thm Dih2k_hypermap.MATVEC_ADD:

vector_add (matvec (?x::((real, ?'a::type) cart, ?'b::type) cart)) (matvec (?y::((real, ?'a::type) cart, ?'b::type) cart)) = *matvec (matrix_add ?x ?y)*

thm Dih2k_hypermap.MATVEC_SUB:

vector_sub (matvec (?x::((real, ?'a::type) cart, ?'b::type) cart)) (matvec (?y::((real, ?'a::type) cart, ?'b::type) cart)) = *matvec (matrix_sub ?x ?y)*

thm Dih2k_hypermap.NORM_VECMAT:

∀ *x::(real, (?'b::type, ?'a::type) finite_product) cart. (?i::nat)* < *dimindex HOL_Light_Import.UNIV* ⟶ *vector_norm (vecmat ?i x)* ≤ *vector_norm x*

thm Dih2k_hypermap.DIST_VECMAT:

∀ *(x::(real, (?'b::type, ?'a::type) finite_product) cart) y::(real, (?'b::type, ?'a::type) finite_product) cart. (?i::nat)* < *dimindex HOL_Light_Import.UNIV* ⟶ *distance (vecmat ?i x, vecmat ?i y)* ≤ *distance (x, y)*

thm Dih2k_hypermap.DOT_VECMAT:

$\forall (x::(real, (?'b::type, ?'a::type)\ finite\_product)\ cart)\ y::(real, (?'b::type, ?'a::type)\ finite\_product)\ cart.\ sum\ (dotdot\ (0::nat)\ (dimindex\ HOL\_Light\_Import.UNIV - (1::nat)))\ (\lambda i::nat.\ dot\ (vecmat\ i\ x)\ (vecmat\ i\ y)) = dot\ x\ y$

thm Dih2k_hypermap.NORM_VECMAT_SUM:

$\forall x::(real, (?'b::type, ?'a::type)\ finite\_product)\ cart.\ vector\_norm\ x \le sum\ (dotdot\ (0::nat)\ (dimindex\ HOL\_Light\_Import.UNIV - (1::nat)))\ (\lambda i::nat.\ vector\_norm\ (vecmat\ i\ x))$

thm Dih2k_hypermap.BOUNDED_MATVEC:

$\forall s::nat \Rightarrow (real, ?'b::type)\ cart \Rightarrow bool.\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow bounded\ (s\ i)) \longrightarrow bounded\ (GSPEC\ (\lambda GEN\%PVAR\%2180::(real, (?'a::type, ?'b::type)\ finite\_product)\ cart.\ \exists x::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2180\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (row\ i\ x)\ (s\ i))\ (matvec\ x)))$

thm Dih2k_hypermap.CLOSED_MATVEC:

$\forall s::nat \Rightarrow (real, ?'b::type)\ cart \Rightarrow bool.\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow HOL\_Light\_Import.closed\ (s\ i)) \longrightarrow HOL\_Light\_Import.closed\ (GSPEC\ (\lambda GEN\%PVAR\%2181::(real, (?'a::type, ?'b::type)\ finite\_product)\ cart.\ \exists x::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2181\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (row\ i\ x)\ (s\ i))\ (matvec\ x)))$

thm Dih2k_hypermap.COMPACT_MATVEC:

$\forall s::nat \Rightarrow (real, ?'b::type)\ cart \Rightarrow bool.\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow compact\ (s\ i)) \longrightarrow compact\ (GSPEC\ (\lambda GEN\%PVAR\%2182::(real, (?'a::type, ?'b::type)\ finite\_product)\ cart.\ \exists x::((real, ?'b::type)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2182\ (\forall i::nat.\ (1::nat) \le i \land i \le dimindex\ HOL\_Light\_Import.UNIV \longrightarrow IN\ (row\ i\ x)\ (s\ i))\ (matvec\ x)))$

thm Dih2k_hypermap.CLOSED_BALL_ANNULUS:

$HOL\_Light\_Import.closed\ ball\_annulus$

thm Dih2k_hypermap.BOUNDED_BALL_ANNULUS:

$bounded\ ball\_annulus$

thm Dih2k_hypermap.COMPACT_BALL_ANNULUS:

$compact\ ball\_annulus$

thm Dih2k_hypermap.COMPACT_BALL_ANNULUS_MATVEC:

$compact\ (GSPEC\ (\lambda GEN\%PVAR\%2183::(real, (?'a::type, 3)\ finite\_product)\ cart.\ \exists v::((real, 3)\ cart, ?'a::type)\ cart.\ SETSPEC\ GEN\%PVAR\%2183\ (\forall i::nat.$

(1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) (matvec v)))

thm Dih2k_hypermap.CLOSED_CONDITION1_SY:

HOL_Light_Import.closed (GSPEC (λGEN%PVAR%2184::(real, (?'b::type, ?'a::type) finite_product) cart. ∃ v::((real, ?'a::type) cart, ?'b::type) cart. SETSPEC GEN%PVAR%2184 (CONDITION1_SY (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) v) (matvec v)))

thm Dih2k_hypermap.SUC_NOT:

(1::nat) < (?k::nat) ⟶ (1::nat) ≤ Suc ((?i::nat) mod ?k) ∧ Suc (?i mod ?k) ≤ ?k ∧ ?i ≠ Suc (?i mod ?k)

thm Dih2k_hypermap.NONPARALLEL_BALL_ANNULUS:

real_of_nat (2::nat) ≤ vector_norm (vector_sub (?v::(real, 3) cart) (?w::(real, 3) cart)) ∧ vector_norm (vector_sub ?v ?w) ≤ cstab ∧ IN ?v ball_annulus ∧ IN ?w ball_annulus ⟶ ¬ collinear (HOL_Light_Import.UNION (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY)))

thm Dih2k_hypermap.VEC0_BALL_ANNULUS:

real_of_nat (2::nat) ≤ vector_norm (vector_sub (?v::(real, 3) cart) (?w::(real, 3) cart)) ∧ ?v ≠ vec (0::nat) ∧ ?w ≠ vec (0::nat) ∧ IN ?v ball_annulus ∧ IN ?w ball_annulus ∧ IN (?z::(real, 3) cart) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT ?v EMPTY)) ∧ IN ?z (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT ?w EMPTY)) ⟶ ?z = vec (0::nat)

thm Dih2k_hypermap.BALL_ANNULUS_3PONITS_ANGLE:

(?v::(real, 3) cart) ≠ vec (0::nat) ∧ (?w::(real, 3) cart) ≠ vec (0::nat) ∧ ?v ≠ ?w ∧ real_of_nat (2::nat) ≤ vector_norm (vector_sub ?v ?w) ∧ IN ?v ball_annulus ∧ IN ?w ball_annulus ⟶ (0::real) < cos (angle (?v, ?w, vec (0::nat)))

thm Dih2k_hypermap.BALL_ANNULUS_3PONITS_NORM_MIN:

(?v::(real, 3) cart) ≠ vec (0::nat) ∧ (?w::(real, 3) cart) ≠ vec (0::nat) ∧ ?v ≠ ?w ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?v (INSERT ?w EMPTY))) ∧ real_of_nat (2::nat) ≤ vector_norm (vector_sub ?v ?w) ∧ IN ?v ball_annulus ∧ IN ?w ball_annulus ∧ % (?t::real) ?w = (?z::(real, 3) cart) ∧ (0::real) < ?t ∧ (?a::real) = sqrt (real_of_nat (4::nat) − $h0^2$) ⟶ ?a ≤ vector_norm (vector_sub ?v ?z)

thm Dih2k_hypermap.BALL_ANNULUS_4PONITS_AFF_GT:

¬ collinear (INSERT (vec (0::nat)) (INSERT (?v::(real, 3) cart) (INSERT (?z::(real, 3) cart) EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT (?w::(real, 3) cart) (INSERT ?z EMPTY))) ∧ real_of_nat (2::nat) ≤ vector_norm (vector_sub ?v ?w) ∧ vector_norm (vector_sub ?v ?w) ≤ cstab ∧ real_of_nat (2::nat) ≤ vector_norm (vector_sub ?z ?v) ∧ real_of_nat (2::nat) ≤ vector_norm

(vector_sub ?z ?w) ∧ IN ?v ball_annulus ∧ IN ?w ball_annulus ∧ IN ?z ball_annulus ⟶ ¬ IN ?z (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT ?v (INSERT ?w EMPTY)))

thm Dih2k_hypermap.AFF_INTER_AFF_GT_EQ_EMPTY:

¬ collinear (INSERT (?x::(real, 3) cart) (INSERT (?y::(real, 3) cart) (INSERT (?z::(real, 3) cart) EMPTY))) ⟶ HOL_Light_Import.INTER (aff (INSERT ?x (INSERT ?y EMPTY))) (aff_gt (INSERT ?x EMPTY) (INSERT ?y (INSERT ?z EMPTY))) = EMPTY

thm Dih2k_hypermap.AFF_GE_INTER_AFF_GT_EQ_EMPTY:

¬ collinear (INSERT (?x::(real, 3) cart) (INSERT (?y::(real, 3) cart) (INSERT (?z::(real, 3) cart) EMPTY))) ∧ ?x ≠ (?u::(real, 3) cart) ∧ ¬ IN ?u (aff_gt (INSERT ?x EMPTY) (INSERT ?y (INSERT ?z EMPTY))) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT ?x EMPTY) (INSERT ?u EMPTY)) (aff_gt (INSERT ?x EMPTY) (INSERT ?y (INSERT ?z EMPTY))) = EMPTY

thm Dih2k_hypermap.CONTINUOUS_ON_LIFT_PRODUCT:

∀ k::nat. (∀ i::nat. IN i (dotdot (1::nat) k) ⟶ continuous_on (lift ∘ (?c::nat ⇒ (real, ?'a::type) cart ⇒ real) i) (?s::(real, ?'a::type) cart ⇒ bool)) ⟶ continuous_on (lift ∘ (λx::(real, ?'a::type) cart. product (dotdot (1::nat) k) (λi::nat. ?c i x))) ?s

thm Dih2k_hypermap.CONTINUOUS_ON_DET:

∀ s::(real, (?'a::type, ?'a::type) finite_product) cart ⇒ bool. continuous_on (lift ∘ (λy::(real, (?'a::type, ?'a::type) finite_product) cart. det (vecmats y))) s

thm Dih2k_hypermap.ROW_SUB:

(1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ⟶ row ?i (matrix_sub (?x::((real, ?'a::type) cart, ?'b::type) cart) (?y::((real, ?'a::type) cart, ?'b::type) cart)) = vector_sub (row ?i ?x) (row ?i ?y)

thm Dih2k_hypermap.LIM_MATVEC:

(∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ --> (λn::nat. row i ((?x::nat ⇒ ((real, ?'a::type) cart, ?'b::type) cart) n)) (row i (?l::((real, ?'a::type) cart, ?'b::type) cart)) sequentially) ⟶ --> (λn::nat. matvec (?x n)) (matvec ?l) sequentially

thm Dih2k_hypermap.LIM_VECMAT:

--> (λn::nat. matvec ((?x::nat ⇒ ((real, ?'a::type) cart, ?'b::type) cart) n)) (matvec (?l::((real, ?'a::type) cart, ?'b::type) cart)) sequentially ⟶ (∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ --> (λn::nat. row i (?x n)) (row i ?l) sequentially)

thm Dih2k_hypermap.CROSS_DOT_SEQUENTIALLY:

--> (?f::nat ⇒ (real, 3) cart) (?a::(real, 3) cart) sequentially ∧ --> (?g::nat ⇒ (real, 3) cart) (?b::(real, 3) cart) sequentially ∧ --> (?h::nat ⇒ (real, 3) *cart*) (*?c*::(*real*, *3*) *cart*) *sequentially* ⟶ *−−>* (*lift* ∘ (λ*n*::*nat*. *dot* (*cross* (*?f* *n*) (*?g n*)) (*?h n*))) (*lift* (*dot* (*cross ?a ?b*) *?c*)) *sequentially*

thm Dih2k_hypermap.ABS_LT_EPSI:

∀ (*a*::*real*) *b*::*real*. |*a* − *b*| < *b* / *real_of_nat* (*4*::*nat*) ∧ (*0*::*real*) < *b* ⟶ (*0*::*real*) < *a*

thm Dih2k_hypermap.LIM_SUBSEQUENCE1:

∀ (*s*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) *r*::*nat* ⇒ *nat*. (∀ *n*::*nat*. *n* ≤ *r n*) ∧ *−−> s* (*?l*::(*real*, *?'a*::*type*) *cart*) *sequentially* ⟶ *−−>* (*s* ∘ *r*) *?l sequentially*

thm Dih2k_hypermap.SEQUENTIALLY_EQ_2POINT:

∀ (*h*::*nat* ⇒ *?'a*::*type*) (*f*::*nat* ⇒ *?'a*::*type*) *g*::*nat* ⇒ *?'a*::*type*. (∀ *n*::*nat*. *IN* (*h* *n*) (*INSERT* (*f n*) (*INSERT* (*g n*) *EMPTY*))) ⟶ (∃ *r*::*nat* ⇒ *nat*. ∀ *n*::*nat*. *n* ≤ *r n* ∧ *h* (*r n*) = *f* (*r n*)) ∨ (∃ *r*::*nat* ⇒ *nat*. ∀ *n*::*nat*. *n* ≤ *r n* ∧ *h* (*r n*) = *g* (*r n*))

thm Dih2k_hypermap.LIM_IN_SET:

∀ (*f*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*g*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*h*::*nat* ⇒ (*real*, *?'a*::*type*) *cart*) (*a*::(*real*, *?'a*::*type*) *cart*) (*b*::(*real*, *?'a*::*type*) *cart*) *c*::(*real*, *?'a*::*type*) *cart*. *−−> f a sequentially* ∧ *−−> g b sequentially* ∧ *−−> h* *c sequentially* ∧ (∀ *n*::*nat*. *IN* (*h n*) (*INSERT* (*f n*) (*INSERT* (*g n*) *EMPTY*))) ⟶ *IN c* (*INSERT a* (*INSERT b EMPTY*))

thm Dih2k_hypermap.POINT_COM_AFF_GT_INTER:

∀ (*y*::(*real*, *3*) *cart*) (*z*::(*real*, *3*) *cart*) (*z1*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT y* (*INSERT z EMPTY*))) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT y* (*INSERT z1 EMPTY*))) ∧ *IN w* (*HOL_Light_Import.INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT y* (*INSERT z EMPTY*))) (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT y* (*INSERT z1 EMPTY*)))) ⟶ *IN z1* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT y EMPTY*)) (*INSERT z EMPTY*))

thm Dih2k_hypermap.DART_FAN_SY:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*row ?i* (*vecmats ?l*), *row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*)) ⟶ *IN ?x* (*dart* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))))

thm Dih2k_hypermap.DART_FAN_SY1:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) = (*row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row ?i* (*vecmats ?l*)) ⟶ *IN ?x* (*dart* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))))

thm Dih2k_hypermap.EDGE_IN_E_SY:

$\forall$ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. (*1*::*nat*) $\leq$ (*?i*::*nat*) $\wedge$ *?i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*?u*::(*real*, *3*) *cart*) = *row ?i* (*vecmats l*) $\wedge$ (*?v*::(*real*, *3*) *cart*) = *row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats l*) $\longrightarrow$ *IN* (*INSERT ?u* (*INSERT ?v EMPTY*)) (*E_SY* (*vecmats l*))

thm Dih2k_hypermap.EQ_EDGE_E_SY:

$\forall$ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *FAN* (*vec* (*0*::*nat*), *V_SY* (*vecmats l*), *E_SY* (*vecmats l*)) $\wedge$ (*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ (*?i*::*nat*) $\wedge$ *?i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row ?i* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) $\wedge$ *row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats l*) = (*?x*::(*real*, *3*) *cart*) $\wedge$ *INSERT ?v* (*INSERT ?x EMPTY*) = *INSERT ?v* (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*) $\wedge$ ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row i* (*vecmats l*) = *row j* (*vecmats l*) $\longrightarrow$ *i* = *j*) $\longrightarrow$ *?x* = *?w*

thm Dih2k_hypermap.EQ_EDGE_E_SY1:

$\forall$ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. (*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ (*?i*::*nat*) $\wedge$ *?i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row ?i* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) $\wedge$ *row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats l*) = (*?x*::(*real*, *3*) *cart*) $\wedge$ *INSERT ?v* (*INSERT ?x EMPTY*) = *INSERT* (*?w*::(*real*, *3*) *cart*) (*INSERT ?x EMPTY*) $\wedge$ ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row i* (*vecmats l*) = *row j* (*vecmats l*) $\longrightarrow$ *i* = *j*) $\longrightarrow$ *?v* = *?w*

thm Dih2k_hypermap.MOD_IMP_EQ:

$\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ (*?k*::*nat*) $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *?k* $\wedge$ *i mod ?k* = *j mod ?k* $\longrightarrow$ *i* = *j*

thm Dih2k_hypermap.SET_OF_EDGE_CARD_EQ2:

$\forall$ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. (*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ (*?i*::*nat*) $\wedge$ *?i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row ?i* (*vecmats l*) = (*?u*::(*real*, *3*) *cart*) $\wedge$ *row* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) $\wedge$ *row* (*Suc* (*Suc* (*?i mod dimindex HOL_Light_Import.UNIV*) *mod dimindex HOL_Light_Import.UNIV*)) (*vecmats l*) = (*?w*::(*real*, *3*) *cart*) $\wedge$ ($\forall$ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) $\leq$ *i* $\wedge$ *i* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ (*1*::*nat*) $\leq$ *j* $\wedge$ *j* $\leq$ *dimindex HOL_Light_Import.UNIV* $\wedge$ *row i* (*vecmats l*) = *row j* (*vecmats l*) $\longrightarrow$ *i* = *j*) $\longrightarrow$ *set_of_edge ?v* (*V_SY* (*vecmats l*)) (*E_SY* (*vecmats l*)) = *INSERT ?u* (*INSERT ?w EMPTY*)

thm Dih2k_hypermap.INV_AZIM_CYCLE_EQ:

$\forall$ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *FAN* (*vec* (*0*::*nat*), *V_SY* (*vecmats l*), *E_SY* (*vecmats l*)) $\wedge$ (*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* $\wedge$

$(1::nat) \leq (?i::nat) \wedge ?i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ ?i\ (vecmats\ l) = (?u::(real, 3)\ cart) \wedge row\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l) = (?v::(real, 3)\ cart) \wedge row\ (Suc\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV)\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l) = (?w::(real, 3)\ cart) \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ (vecmats\ l) = row\ j\ (vecmats\ l) \longrightarrow i = j) \longrightarrow ivs\_azim\_cycle\ (EE\ ?v\ (E\_SY\ (vecmats\ l)))\ (vec\ (0::nat))\ ?v\ ?u = ?w$

thm Dih2k_hypermap.INV_AZIM_CYCLE_EQ1:

$\forall l::(real, (?'a::type, 3)\ finite\_product)\ cart.\ FAN\ (vec\ (0::nat), V\_SY\ (vecmats\ l), E\_SY\ (vecmats\ l)) \wedge (1::nat) < dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?i::nat) \wedge ?i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ ?i\ (vecmats\ l) = (?u::(real, 3)\ cart) \wedge row\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l) = (?v::(real, 3)\ cart) \wedge row\ (Suc\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV)\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l) = (?w::(real, 3)\ cart) \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ (vecmats\ l) = row\ j\ (vecmats\ l) \longrightarrow i = j) \longrightarrow ivs\_azim\_cycle\ (EE\ ?v\ (E\_SY\ (vecmats\ l)))\ (vec\ (0::nat))\ ?v\ ?w = ?u$

thm Dih2k_hypermap.FF_OF_HYP_EQ:

$FAN\ (vec\ (0::nat), V\_SY\ (vecmats\ (?l::(real, (?'a::type, 3)\ finite\_product)\ cart)), E\_SY\ (vecmats\ ?l)) \wedge (1::nat) < dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq (?i::nat) \wedge ?i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ ?i\ (vecmats\ ?l) = (?u::(real, 3)\ cart) \wedge row\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ ?l) = (?v::(real, 3)\ cart) \wedge row\ (Suc\ (Suc\ (?i\ mod\ dimindex\ HOL\_Light\_Import.UNIV)\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ ?l) = (?w::(real, 3)\ cart) \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ (vecmats\ ?l) = row\ j\ (vecmats\ ?l) \longrightarrow i = j) \longrightarrow (?v, ?w) = ff\_of\_hyp\ (vec\ (0::nat), V\_SY\ (vecmats\ ?l), E\_SY\ (vecmats\ ?l))\ (?u, ?v)$

thm Dih2k_hypermap.POWER_FF_OF_HYP_EQ:

$\forall (i::nat)\ (u::(real, 3)\ cart)\ (v::(real, 3)\ cart)\ (l::(real, (?'a::type, 3)\ finite\_product)\ cart)\ x::(real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (vec\ (0::nat), V\_SY\ (vecmats\ l), E\_SY\ (vecmats\ l)) \wedge (1::nat) < dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ (vecmats\ l) = u \wedge row\ (Suc\ (i\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l) = v \wedge x = (row\ (1::nat)\ (vecmats\ l), row\ (Suc\ ((1::nat)\ mod\ dimindex\ HOL\_Light\_Import.UNIV))\ (vecmats\ l)) \wedge (\forall (i::nat)\ j::nat.\ (1::nat) \leq i \wedge i \leq dimindex\ HOL\_Light\_Import.UNIV \wedge (1::nat) \leq j \wedge j \leq dimindex\ HOL\_Light\_Import.UNIV \wedge row\ i\ (vecmats\ l) = row\ j\ (vecmats\ l) \longrightarrow i = j) \longrightarrow (u, v) = POWER\ (ff\_of\_hyp\ (vec\ (0::nat), V\_SY\ (vecmats\ l), E\_SY\ (vecmats\ l)))\ (i - (1::nat))\ x$

thm Dih2k_hypermap.POWER_FF_HYP_ID:

∀ (k::nat) (l::(real, (?'a::type, 3) finite_product) cart) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l)) ∧ (1::nat) < k ∧ dimindex HOL_Light_Import.UNIV = k ∧ (row (1::nat) (vecmats l), row (Suc ((1::nat) mod dimindex HOL_Light_Import.UNIV)) (vecmats l)) = x ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ x = POWER (ff_of_hyp (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l))) k x

thm Dih2k_hypermap.FACE_HYP_FAN_SY:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (row (1::nat) (vecmats ?l), row (Suc ((1::nat) mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l)) = (?x::(real, 3) cart × (real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ F_SY (vecmats ?l) = face (hypermap (HYP (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)))) ?x

thm Dih2k_hypermap.CARD_F_SY_EQ:

(1::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)) = row j (vecmats ?l) ⟶ i = j) ⟶ CARD (F_SY (vecmats ?l)) = dimindex HOL_Light_Import.UNIV

thm Dih2k_hypermap.AZIM_CYCLE_EQ1:

∀ l::(real, (?'a::type, 3) finite_product) cart. FAN (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ row ?i (vecmats l) = (?u::(real, 3) cart) ∧ row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats l) = (?v::(real, 3) cart) ∧ row (Suc (Suc (?i mod dimindex HOL_Light_Import.UNIV) mod dimindex HOL_Light_Import.UNIV)) (vecmats l) = (?w::(real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ azim_cycle (EE ?v (E_SY (vecmats l))) (vec (0::nat)) ?v ?w = ?u

thm Dih2k_hypermap.AZIM_CYCLE_EQ:

∀ l::(real, (?'a::type, 3) finite_product) cart. FAN (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ row ?i (vecmats l) = (?u::(real, 3) cart) ∧ row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats l) = (?v::(real, 3) cart) ∧ row (Suc (Suc (?i mod dimindex HOL_Light_Import.UNIV) mod dimindex HOL_Light_Import.UNIV)) (vecmats l) = (?w::(real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧

(1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ azim_cycle (EE ?v (E_SY (vecmats l))) (vec (0::nat)) ?v ?u = ?w

thm Dih2k_hypermap.NN_OF_HYP_EQ1:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ row ?i (vecmats ?l) = (?u::(real, 3) cart) ∧ row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?v::(real, 3) cart) ∧ row (Suc (Suc (?i mod dimindex HOL_Light_Import.UNIV) mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?w::(real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ (?v, ?u) = nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)) (?v, ?w)

thm Dih2k_hypermap.NN_OF_HYP_EQ:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ row ?i (vecmats ?l) = (?u::(real, 3) cart) ∧ row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?v::(real, 3) cart) ∧ row (Suc (Suc (?i mod dimindex HOL_Light_Import.UNIV) mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?w::(real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ (?v, ?w) = nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)) (?v, ?u)

thm Dih2k_hypermap.DART_OF_HYP_SY_EQ:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ darts_of_hyp (E_SY (vecmats ?l)) (V_SY (vecmats ?l)) = HOL_Light_Import.UNION (F_SY (vecmats ?l)) (IMAGE (nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l))) (F_SY (vecmats ?l)))

thm Dih2k_hypermap.IMAGE_NN_OF_HYP_F_SY:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ IMAGE (nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l))) (F_SY (vecmats ?l)) = GSPEC (λGEN%PVAR%2187::(real, 3) cart × (real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%2187

((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row* (*Suc* (*i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row i* (*vecmats ?l*)))

thm Dih2k_hypermap.CARD_IMAGE_F_SY_EQ:

(*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *CARD* (*GSPEC* (λ*GEN%PVAR%2190*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%2190* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row* (*Suc* (*i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row i* (*vecmats ?l*)))) = *dimindex HOL_Light_Import.UNIV*

thm Dih2k_hypermap.SUC_POWER2_NOT:

(*2*::*nat*) < (*?k*::*nat*) ∧ (*?i*::*nat*) ≤ *?k* ⟶ *?i* ≠ *Suc* (*Suc* (*?i mod ?k*) *mod ?k*)

thm Dih2k_hypermap.F_SY_INTER_IMAGE_NN_EMPTY:

(*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *HOL_Light_Import.INTER* (*F_SY* (*vecmats ?l*)) (*GSPEC* (λ*GEN%PVAR%2191*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%2191* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row* (*Suc* (*i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row i* (*vecmats ?l*)))) = *EMPTY*

thm Dih2k_hypermap.FINITE_F_SY:

FINITE (*F_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *?'b*::*type*) *finite_product*) *cart*)))

thm Dih2k_hypermap.FINITE_IMAGE_F_SY:

(*1*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *FINITE* (*GSPEC* (λ*GEN%PVAR%2196*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. ∃ *i*::*nat*. *SETSPEC GEN%PVAR%2196* ((*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row* (*Suc* (*i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row i* (*vecmats ?l*))))

thm Dih2k_hypermap.CARD_DART_OF_HYP:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *CARD* (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) = (*2*::*nat*) ∗ *dimindex HOL_Light_Import.UNIV*

thm Dih2k_hypermap.IMAGE_NN_OF_HYP_EQ_F_SY:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀(i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ IMAGE (nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l))) (GSPEC (λGEN%PVAR%2197::(real, 3) cart × (real, 3) cart. ∃i::nat. SETSPEC GEN%PVAR%2197 ((1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV) (row (Suc (i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l), row i (vecmats ?l)))) = F_SY (vecmats ?l)

thm Dih2k_hypermap.DART_OF_HYP_SY_EQ1:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀(i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ∧ GSPEC (λGEN%PVAR%2198::(real, 3) cart × (real, 3) cart. ∃i::nat. SETSPEC GEN%PVAR%2198 ((1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV) (row (Suc (i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l), row i (vecmats ?l))) = (?S::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ darts_of_hyp (E_SY (vecmats ?l)) (V_SY (vecmats ?l)) = HOL_Light_Import.UNION ?S (IMAGE (nn_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l))) ?S)

thm Dih2k_hypermap.F_SY_EQ_FACE:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ (row ?i (vecmats ?l), row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l)) = (?x::(real, 3) cart × (real, 3) cart) ∧ (∀(i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ F_SY (vecmats ?l) = face (hypermap (HYP (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)))) ?x

thm Dih2k_hypermap.FF_OF_HYP_EQ1:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ row ?i (vecmats ?l) = (?u::(real, 3) cart) ∧ row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?v::(real, 3) cart) ∧ row (Suc (Suc (?i mod dimindex HOL_Light_Import.UNIV) mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l) = (?w::(real, 3) cart) ∧ (∀(i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ (?v, ?u) = ff_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)) (?w, ?v)

thm Dih2k_hypermap.POWER_FF_OF_HYP_EQ1:

∀ (j::nat) (i::nat) (u::(real, 3) cart) (v::(real, 3) cart) (l::(real, (?'a::type, 3) finite_product) cart) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ j = dimindex HOL_Light_Import.UNIV − i + (1::nat) ∧ row i (vecmats l) = u ∧ row (Suc (i mod dimindex HOL_Light_Import.UNIV)) (vecmats l) = v ∧ x = (row (Suc ((1::nat) mod dimindex HOL_Light_Import.UNIV)) (vecmats l), row (1::nat) (vecmats l)) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ (v, u) = POWER (ff_of_hyp (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l))) j x

thm Dih2k_hypermap.POWER_FF_HYP_ID1:

∀ (k::nat) (l::(real, (?'a::type, 3) finite_product) cart) x::(real, 3) cart × (real, 3) cart. FAN (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l)) ∧ (1::nat) < k ∧ dimindex HOL_Light_Import.UNIV = k ∧ (row (Suc ((1::nat) mod dimindex HOL_Light_Import.UNIV)) (vecmats l), row (1::nat) (vecmats l)) = x ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ x = POWER (ff_of_hyp (vec (0::nat), V_SY (vecmats l), E_SY (vecmats l))) k x

thm Dih2k_hypermap.FACE_HYP_FAN_SY1:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (row (Suc ((1::nat) mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l), row (1::nat) (vecmats ?l)) = (?x::(real, 3) cart × (real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ∧ (?S::(real, 3) cart × (real, 3) cart ⇒ bool) = GSPEC (λGEN%PVAR%2199::(real, 3) cart × (real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%2199 ((1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV) (row (Suc (i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l), row i (vecmats ?l))) ⟶ ?S = face (hypermap (HYP (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)))) ?x

thm Dih2k_hypermap.F_SY_EQ_FACE1:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (1::nat) < dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ (row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats ?l), row ?i (vecmats ?l)) = (?x::(real, 3) cart × (real, 3) cart) ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ∧ (?S::(real, 3) cart × (real, 3) cart ⇒ bool) = GSPEC (λGEN%PVAR%2200::(real, 3) cart × (real, 3) cart. ∃ i::nat. SETSPEC GEN%PVAR%2200 ((1::nat)

≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV*) (*row* (*Suc* (*i mod dimindex HOL_Light_Import.UNIV*)) (*vecmats ?l*), *row i* (*vecmats ?l*))) ⟶ *?S* = *face* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x*

thm Dih2k_hypermap.DART_OF_HYP_EQ_FACE_SY:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀(*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ∧ (*?S*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x* ⟶ *darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*)) = *HOL_Light_Import.UNION ?S* (*IMAGE* (*nn_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?S*)

thm Dih2k_hypermap.ID_FF_OF_HYP_NOT_DARTS:

∀ *n*::*nat*. *FAN* (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀(*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ ¬ *IN* (*?v*::(*real*, *3*) *cart*, *?u*::(*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ⟶ *POWER* (*ff_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) *n* (*?v*, *?u*) = (*?v*, *?u*)

thm Dih2k_hypermap.CARD_FACE_SY:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀(*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ∧ (*?S*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) = *face* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x* ⟶ *CARD ?S* = *dimindex HOL_Light_Import.UNIV*

thm Dih2k_hypermap.FF_OF_HYP_POWER_EQ_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀(*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *POWER* (*ff_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) (*dimindex HOL_Light_Import.UNIV*) = *id*

thm Dih2k_hypermap.EXISTS_POINT_DART_OF_HYP:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV*

∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ (∃ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))))

thm Dih2k_hypermap.FF_OF_HYP_NOT_EQ_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ (∀ *i*::*nat*. (*0*::*nat*) < *i* ∧ *i* < *dimindex HOL_Light_Import.UNIV* ⟶ *POWER* (*ff_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) *i* ≠ *id*)

thm Dih2k_hypermap.FF_OF_HYP_HAS_ORDERS:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *has_orders* (*ff_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) (*dimindex HOL_Light_Import.UNIV*)

thm Dih2k_hypermap.CARD_NODE_SY:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ⟶ *CARD* (*node* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x*) = (*2*::*nat*)

thm Dih2k_hypermap.NODE_SY_POWER_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ⟶ *node_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) (*node_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x*) = *?x*

thm Dih2k_hypermap.ID_NN_OF_HYP_NOT_DARTS:

∀ *n*::*nat*. *FAN* (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV*

∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ ¬ *IN* (*?v*::(*real*, *3*) *cart*, *?u*::(*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ⟶ *POWER* (*nn_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) *n* (*?v*, *?u*) = (*?v*, *?u*)

thm Dih2k_hypermap.NN_OF_HYP_POWER_EQ_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *POWER* (*nn_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) (*2*::*nat*) = *id*

thm Dih2k_hypermap.NODE_SY_NOT_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ∧ *IN* (*?x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*) (*darts_of_hyp* (*E_SY* (*vecmats ?l*)) (*V_SY* (*vecmats ?l*))) ⟶ *node_map* (*hypermap* (*HYP* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*)))) *?x* ≠ *?x*

thm Dih2k_hypermap.NN_OF_HYP_NOT_EQ_ID:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ (∀ *i*::*nat*. (*0*::*nat*) < *i* ∧ *i* < (*2*::*nat*) ⟶ *POWER* (*nn_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) *i* ≠ *id*)

thm Dih2k_hypermap.NN_OF_HYP_HAS_ORDERS:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧ (*1*::*nat*) ≤ *j* ∧ *j* ≤ *dimindex HOL_Light_Import.UNIV* ∧ *row i* (*vecmats ?l*) = *row j* (*vecmats ?l*) ⟶ *i* = *j*) ⟶ *has_orders* (*nn_of_hyp* (*vec* (*0*::*nat*), *V_SY* (*vecmats ?l*), *E_SY* (*vecmats ?l*))) (*2*::*nat*)

thm Dih2k_hypermap.EE_OF_HYP_HAS_ORDERS:

FAN (*vec* (*0*::*nat*), *V_SY* (*vecmats* (*?l*::(*real*, (*?'b*::*type*, *3*) *finite_product*) *cart*)), *E_SY* (*vecmats ?l*)) ∧ (*2*::*nat*) < *dimindex HOL_Light_Import.UNIV* ∧ (∀ (*i*::*nat*) *j*::*nat*. (*1*::*nat*) ≤ *i* ∧ *i* ≤ *dimindex HOL_Light_Import.UNIV* ∧

(1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ has_orders (ee_of_hyp (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l))) (2::nat)

thm Dih2k_hypermap.DIH2K_FAN_HYP_SY:

FAN (vec (0::nat), V_SY (vecmats (?l::(real, (?'a::type, 3) finite_product) cart)), E_SY (vecmats ?l)) ∧ (2::nat) < dimindex HOL_Light_Import.UNIV ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats ?l) = row j (vecmats ?l) ⟶ i = j) ⟶ dih2k (hypermap (HYP (vec (0::nat), V_SY (vecmats ?l), E_SY (vecmats ?l)))) (CARD (F_SY (vecmats ?l)))

thm Wjscpro.POWER_MOD_FUN:

∀ n::nat. (1::nat) ≤ n ∧ (1::nat) < (?k::nat) ⟶ POWER (λi::nat. ((1::nat) + i) mod ?k) n (?i::nat) = (n + ?i) mod ?k

thm Wjscpro.CLOSED_SY:

stable_system (?k::nat) (?d::?'b::type) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (1::nat) < ?k ∧ (2::nat) < ?k ⟶ HOL_Light_Import.closed (GSPEC (λGEN%PVAR%2205::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2205 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v ∧ CONDITION2_SY v) (matvec v)))

thm Wjscpro.BOUNDED_SY:

stable_system (?k::nat) (?d::?'b::type) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (1::nat) < ?k ∧ (2::nat) < ?k ⟶ bounded (GSPEC (λGEN%PVAR%2207::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2207 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v ∧ CONDITION2_SY v) (matvec v)))

thm Wjscpro.WJSCPRO:

stable_system (?k::nat) (?d::real) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < ?k ⟶ compact (GSPEC (λGEN%PVAR%2208::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2208 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v ∧ CONDITION2_SY v) (matvec v)))

thm Tecoxbm.CROSS_DOT_POS_SY:

∀ *l*::(*real*, (*?'b*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*?'a*::*type*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* *?k*) ∧ *?k* = *dimindex* *HOL_Light_Import*.*UNIV* ∧ (*2*::*nat*) < *?k* ∧ *IN* (*?u*::(*real*, *3*) *cart*) (*V_SY* (*vecmats* *l*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *dimindex* *HOL_Light_Import*.*UNIV* ∧ *row* *?i* (*vecmats* *l*) = (*?y*::(*real*, *3*) *cart*) ∧ *row* (*Suc* (*?i* *mod* *dimindex* *HOL_Light_Import*.*UNIV*)) (*vecmats* *l*) = (*?z*::(*real*, *3*) *cart*) ∧ *IN* *l* (*B_SY1* *?a* *?b*) ⟶ (*0*::*real*) ≤ *dot* (*cross* *?y* *?z*) *?u*

thm Tecoxbm.IVS_RHO_NODE_IN_EDGE:

∀ *v*::(*real*, *3*) *cart*. *local_fan* (*?V*::(*real*, *3*) *cart* ⇒ *bool*, *?E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*, *?FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) ∧ *IN* *v* *?V* ⟶ *IN* (*INSERT* *v* (*INSERT* (*ivs_rho_node1* *?FF* *v*) *EMPTY*)) *?E*

thm Tecoxbm.PROPERTIES_OF_FAN_IN_B_SY:

∀ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*real*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* *?k*) ∧ *dimindex* *HOL_Light_Import*.*UNIV* = *?k* ∧ (*2*::*nat*) < *?k* ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *dimindex* *HOL_Light_Import*.*UNIV* ∧ (*1*::*nat*) ≤ (*?j*::*nat*) ∧ *?j* ≤ *dimindex* *HOL_Light_Import*.*UNIV* ∧ *?i* ≠ *?j* ∧ *row* *?i* (*vecmats* *l*) = (*?y*::(*real*, *3*) *cart*) ∧ *row* *?j* (*vecmats* *l*) = (*?z*::(*real*, *3*) *cart*) ∧ *IN* *l* (*B_SY1* *?a* *?b*) ⟶ *real_of_nat* (*2*::*nat*) ≤ *vector_norm* (*vector_sub* *?y* *?z*)

thm Tecoxbm.AFF_GT_INTER_AFF_SY:

∀ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*real*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* *?k*) ∧ *dimindex* *HOL_Light_Import*.*UNIV* = *?k* ∧ (*2*::*nat*) < *?k* ∧ *SUBSET* (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*)) (*V_SY* (*vecmats* *l*)) ∧ *vector_norm* (*vector_sub* *?u* *?w*) ≤ *cstab* ∧ *real_of_nat* (*2*::*nat*) ≤ *vector_norm* (*vector_sub* *?u* *?w*) ∧ ¬ *IN* (*INSERT* *?u* (*INSERT* *?w* *EMPTY*)) (*E_SY* (*vecmats* *l*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *dimindex* *HOL_Light_Import*.*UNIV* ∧ *row* *?i* (*vecmats* *l*) = (*?y*::(*real*, *3*) *cart*) ∧ *row* (*Suc* (*?i* *mod* *dimindex* *HOL_Light_Import*.*UNIV*)) (*vecmats* *l*) = (*?z*::(*real*, *3*) *cart*) ∧ *IN* *l* (*B_SY1* *?a* *?b*) ⟶ *HOL_Light_Import*.*INTER* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* *?u* (*INSERT* *?w* *EMPTY*))) (*aff* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *?y* (*INSERT* *?z* *EMPTY*)))) = *EMPTY*

thm Tecoxbm.TECOXBM1:

∀ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*real*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* *?k*) ∧ *?k* = *dimindex* *HOL_Light_Import*.*UNIV* ∧ (*2*::*nat*) < *?k* ∧ *SUBSET* (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*)) (*V_SY* (*vecmats* *l*)) ∧ *vector_norm* (*vector_sub* *?u* *?w*) ≤ *cstab* ∧ *real_of_nat* (*2*::*nat*) ≤ *vector_norm* (*vector_sub ?u ?w*) ∧ ¬ *IN* (*INSERT ?u* (*INSERT ?w EMPTY*)) (*E_SY* (*vecmats l*)) ∧ *IN l* (*B_SY1 ?a ?b*) ⟶ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*F_SY* (*vecmats l*)) ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?u* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x* (*E_SY* (*vecmats l*))))

thm Tecoxbm.TECOXBM2:

∀ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*real*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ *?k* = *dimindex HOL_Light_Import.UNIV* ∧ (*2*::*nat*) < *?k* ∧ *SUBSET* (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*)) (*V_SY* (*vecmats l*)) ∧ *vector_norm* (*vector_sub ?u ?w*) ≤ *cstab* ∧ *real_of_nat* (*2*::*nat*) ≤ *vector_norm* (*vector_sub ?u ?w*) ∧ ¬ *IN* (*INSERT ?u* (*INSERT ?w EMPTY*)) (*E_SY* (*vecmats l*)) ∧ *IN l* (*B_SY1 ?a ?b*) ⟶ ¬ *collinear* (*HOL_Light_Import.UNION* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?u* (*INSERT ?w EMPTY*)))

thm Tecoxbm.TECOXBM:

∀ *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *stable_system* (*?k*::*nat*) (*?d*::*real*) (*dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*))) (*?a*::*nat* × *nat* ⇒ *real*) (*?b*::*nat* × *nat* ⇒ *real*) (*?J*::(*nat* ⇒ *bool*) ⇒ *bool*) (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ *?k* = *dimindex HOL_Light_Import.UNIV* ∧ (*2*::*nat*) < *?k* ∧ *SUBSET* (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*)) (*V_SY* (*vecmats l*)) ∧ *vector_norm* (*vector_sub ?u ?w*) ≤ *cstab* ∧ *real_of_nat* (*2*::*nat*) ≤ *vector_norm* (*vector_sub ?u ?w*) ∧ ¬ *IN* (*INSERT ?u* (*INSERT ?w EMPTY*)) (*E_SY* (*vecmats l*)) ∧ *IN l* (*B_SY1 ?a ?b*) ⟶ ¬ *collinear* (*HOL_Light_Import.UNION* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?u* (*INSERT ?w EMPTY*))) ∧ (∀ *x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN x* (*F_SY* (*vecmats l*)) ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT ?u* (*INSERT ?w EMPTY*))) (*wedge_in_fan_gt x* (*E_SY* (*vecmats l*))))

thm Vpwshto.NORM_COS_ANGLE_LE:

∀ (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *vector_norm* (*vector_sub v w*) = *vector_norm* (*vector_sub v w1*) ∧ *v* ≠ *u* ∧ *v* ≠ *w* ∧ *v* ≠ *w1* ⟶ (*vector_norm* (*vector_sub u w*) ≤ *vector_norm* (*vector_sub u w1*)) = (*cos* (*angle* (*u*, *v*, *w1*)) ≤ *cos* (*angle* (*u*, *v*, *w*)))

thm Vpwshto.NORM_COS_ANGLE_LT:

∀ (*v*::(*real*, *3*) *cart*) (*v1*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *vector_norm* (*vector_sub v w*) = *vector_norm* (*vector_sub v1 w1*) ∧ *vector_norm* (*vector_sub v u*) = *vector_norm* (*vector_sub v1 u1*) ∧ *v* ≠ *u* ∧ *v* ≠ *w* ∧ *v1* ≠ *w1* ⟶ (*vector_norm* (*vector_sub u w*) < *vector_norm* (*vector_sub u1 w1*)) = (*angle* (*u*, *v*, *w*) < *angle* (*u1*, *v1*, *w1*))

thm Vpwshto.NORM_COS_ANGLE_4POINT:

∀ (*v*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. *vector_norm* (*vector_sub v w*) = *vector_norm* (*vector_sub u w1*) ∧ *v* ≠ *u* ∧ *v*

≠ w ∧ u ≠ w1 ⟶ (vector_norm (vector_sub u w) = vector_norm (vector_sub v w1)) = (cos (angle (v, u, w1)) = cos (angle (u, v, w)))

thm Vpwshto.MAX_COPLANAR_4POINT:

∀ (v::(real, 3) cart) (x::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) w1::(real, 3) cart. v ≠ x ∧ v ≠ u ∧ v ≠ w ∧ x ≠ u ∧ x ≠ w ∧ u ≠ w ∧ v ≠ w1 ∧ x ≠ w1 ∧ u ≠ w1 ∧ vector_norm (vector_sub v x) = (?t::real) ∧ vector_norm (vector_sub v u) = (?a::real) ∧ vector_norm (vector_sub x w) = ?a ∧ vector_norm (vector_sub u w) = ?a ∧ IN (?y::(real, 3) cart) (HOL_Light_Import.INTER (open_segment (v, w)) (open_segment (x, u))) ∧ vector_norm (vector_sub x w1) = ?a ∧ vector_norm (vector_sub u w1) = ?a ⟶ min (vector_norm (vector_sub v w1)) (vector_norm (vector_sub x u)) ≤ min (vector_norm (vector_sub v w)) (vector_norm (vector_sub x u))

thm Vpwshto.SUM_4ANGLE_4POINT_EQ_2PI:

∀ (v::(real, 3) cart) (x::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) y::(real, 3) cart. v ≠ x ∧ v ≠ u ∧ v ≠ w ∧ x ≠ u ∧ x ≠ w ∧ u ≠ w ∧ IN y (HOL_Light_Import.INTER (open_segment (v, w)) (open_segment (x, u))) ⟶ angle (x, v, u) + (angle (v, u, w) + (angle (u, w, x) + angle (w, x, v))) = real_of_nat (2::nat) * pi

thm Vpwshto.EQ_DIAGONAL_MIN:

∀ (v::(real, 3) cart) (v1::(real, 3) cart) (x::(real, 3) cart) (x1::(real, 3) cart) (u::(real, 3) cart) (w::(real, 3) cart) (u1::(real, 3) cart) w1::(real, 3) cart. v ≠ x ∧ v ≠ u ∧ v ≠ w ∧ x ≠ u ∧ x ≠ w ∧ u ≠ w ∧ v1 ≠ x1 ∧ v1 ≠ u1 ∧ v1 ≠ w1 ∧ x1 ≠ u1 ∧ x1 ≠ w1 ∧ u1 ≠ w1 ∧ vector_norm (vector_sub v x) = (?t::real) ∧ vector_norm (vector_sub v u) = (?a::real) ∧ vector_norm (vector_sub x w) = ?a ∧ vector_norm (vector_sub u w) = ?a ∧ IN (?y::(real, 3) cart) (HOL_Light_Import.INTER (open_segment (v, w)) (open_segment (x, u))) ∧ vector_norm (vector_sub v w) = vector_norm (vector_sub x u) ∧ vector_norm (vector_sub v1 x1) = ?t ∧ vector_norm (vector_sub v1 u1) = ?a ∧ vector_norm (vector_sub x1 w1) = ?a ∧ vector_norm (vector_sub u1 w1) = ?a ∧ IN (?y1.0::(real, 3) cart) (HOL_Light_Import.INTER (open_segment (v1, w1)) (open_segment (x1, u1))) ⟶ min (vector_norm (vector_sub v1 w1)) (vector_norm (vector_sub x1 u1)) ≤ vector_norm (vector_sub x u)

thm Vpwshto.TWO_DIAGONAL_AT_MOST:

real_of_nat (2::nat) ≤ (?t::real) ∧ ?t ≤ real_of_nat (4::nat) ⟶ (∃ (u::(real, 3) cart) (w::(real, 3) cart) (v::(real, 3) cart) (x::(real, 3) cart) y::(real, 3) cart. v ≠ x ∧ v ≠ u ∧ v ≠ w ∧ x ≠ u ∧ x ≠ w ∧ u ≠ w ∧ vector_norm (vector_sub v x) = ?t ∧ vector_norm (vector_sub v u) = real_of_nat (2::nat) ∧ vector_norm (vector_sub x w) = real_of_nat (2::nat) ∧ vector_norm (vector_sub u w) = real_of_nat (2::nat) ∧ IN y (HOL_Light_Import.INTER (open_segment (v, w)) (open_segment (x, u))) ∧ vector_norm (vector_sub v w) = vector_norm (vector_sub x u) ∧ (?t ≤ vector_norm (vector_sub v w) ⟶ vector_norm (vector_sub *v w*) ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*)) ∧ *?t* ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*))))

thm COLLINEAR_SEGMENT_conjunct1:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *collinear* (*open_segment* (*a*, *b*))

thm COLLINEAR_SEGMENT_conjunct0:

∀ (*a*::(*real*, *?'a*::*type*) *cart*) *b*::(*real*, *?'a*::*type*) *cart*. *collinear* (*closed_segment* [(*a*, *b*)])

thm Vpwshto.MAX_IF_COPLANAR:

∀ (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) *t*::*real*. *a* ≤ *t* ∧ *v* ≠ *x* ∧ *v* ≠ *u* ∧ *v* ≠ *w* ∧ *x* ≠ *u* ∧ *x* ≠ *w* ∧ *u* ≠ *w* ∧ ¬ *collinear* (*INSERT v* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT x* (*INSERT u EMPTY*))) ∧ *vector_norm* (*vector_sub v x*) = *t* ∧ *vector_norm* (*vector_sub v u*) = *a* ∧ *vector_norm* (*vector_sub x w*) = *a* ∧ *vector_norm* (*vector_sub u w*) = *a* ∧ *t* ≤ *vector_norm* (*vector_sub x u*) ⟶ (∃ (*w1*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *IN y* (*HOL_Light_Import.INTER* (*open_segment* (*v*, *w1*)) (*open_segment* (*x*, *u*))) ∧ *vector_norm* (*vector_sub x w1*) = *a* ∧ *vector_norm* (*vector_sub u w1*) = *a* ∧ *v* ≠ *w1*)

thm Vpwshto.MAX_IF_COPLANAR1:

∀ (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*a*::*real*) (*a2*::*real*) *t*::*real*. *a* ≤ *t* ∧ *v* ≠ *x* ∧ *v* ≠ *u* ∧ *v* ≠ *w* ∧ *x* ≠ *u* ∧ *x* ≠ *w* ∧ *u* ≠ *w* ∧ ¬ *collinear* (*INSERT v* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT x* (*INSERT u EMPTY*))) ∧ *vector_norm* (*vector_sub v x*) = *t* ∧ *vector_norm* (*vector_sub v u*) = *a* ∧ *vector_norm* (*vector_sub x w*) = *a2* ∧ *vector_norm* (*vector_sub u w*) = *a2* ∧ *a2* = *t* ∧ *t* ≤ *vector_norm* (*vector_sub x u*) ⟶ (∃ (*w1*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *IN y* (*HOL_Light_Import.INTER* (*open_segment* (*v*, *w1*)) (*open_segment* (*x*, *u*))) ∧ *vector_norm* (*vector_sub x w1*) = *a2* ∧ *vector_norm* (*vector_sub u w1*) = *a2* ∧ *v* ≠ *w1*)

thm Vpwshto.TWO_DIAGONAL_AT_MOST1:

(*?t*::*real*) ≤ *real_of_nat* (*2*::*nat*) ∧ (*0*::*real*) < *?t* ⟶ (∃ (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) *y*::(*real*, *3*) *cart*. *v* ≠ *x* ∧ *v* ≠ *u* ∧ *v* ≠ *w* ∧ *x* ≠ *u* ∧ *x* ≠ *w* ∧ *u* ≠ *w* ∧ *vector_norm* (*vector_sub v x*) = *?t* ∧ *vector_norm* (*vector_sub v u*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub x w*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub u w*) = *real_of_nat* (*2*::*nat*) ∧ *IN y* (*HOL_Light_Import.INTER* (*open_segment* (*v*, *w*)) (*open_segment* (*x*, *u*))) ∧ *vector_norm* (*vector_sub v w*) = *vector_norm* (*vector_sub x u*) ∧ (*?t* ≤ *vector_norm* (*vector_sub v w*) ⟶ *vector_norm* (*vector_sub v w*) ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*)) ∧ *?t* ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*))))

thm Vpwshto.VPWSHTO1:

∀ (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. (*?t*::*real*) ≤ *real_of_nat* (*4*::*nat*) ∧ ¬ *collinear* (*INSERT v* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT x* (*INSERT u EMPTY*))) ∧ *v* ≠ *w* ∧ *vector_norm* (*vector_sub v x*) = *?t* ∧ *vector_norm* (*vector_sub v u*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub x w*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub u w*) = *real_of_nat* (*2*::*nat*) ∧ *?t* ≤ *vector_norm* (*vector_sub x u*) ∧ *?t* ≤ *vector_norm* (*vector_sub v w*) ⟶ *min* (*vector_norm* (*vector_sub v w*)) (*vector_norm* (*vector_sub x u*)) ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*)) ∧ *?t* ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*))

thm Vpwshto.VPWSHTO2:

∀ (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT w* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT v* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT v* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT w* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT w* (*INSERT v EMPTY*))) ∧ *vector_norm* (*vector_sub v x*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub x u*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub u w*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub w w1*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub w1 v*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub v u*) ≤ *vector_norm* (*vector_sub v w*) ∧ *vector_norm* (*vector_sub v u*) ≤ *vector_norm* (*vector_sub u w1*) ∧ *vector_norm* (*vector_sub v u*) ≤ *vector_norm* (*vector_sub w1 x*) ∧ *vector_norm* (*vector_sub v u*) ≤ *vector_norm* (*vector_sub x w*) ⟶ (∃ (*v1*::(*real*, *3*) *cart*) (*u1*::(*real*, *3*) *cart*) *w2*::(*real*, *3*) *cart*. *IN u1* (*INSERT v* (*INSERT x* (*INSERT u* (*INSERT w* (*INSERT w1 EMPTY*))))) ∧ *IN u1* (*INSERT v* (*INSERT x* (*INSERT u* (*INSERT w* (*INSERT w1 EMPTY*))))) ∧ *IN w2* (*INSERT v* (*INSERT x* (*INSERT u* (*INSERT w* (*INSERT w1 EMPTY*))))) ∧ *v1* ≠ *u1* ∧ *u1* ≠ *w2* ∧ *v1* ≠ *w2* ∧ *vector_norm* (*vector_sub v1 u1*) ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*)) ∧ *vector_norm* (*vector_sub v1 w2*) ≤ (*1*::*real*) + *sqrt* (*real_of_nat* (*5*::*nat*)))

thm Vpwshto.VPWSHTO:

∀ (*v*::(*real*, *3*) *cart*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *w1*::(*real*, *3*) *cart*. ¬ *collinear* (*INSERT v* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT x* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT v* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT w* (*INSERT u EMPTY*))) ∧ ¬ *collinear* (*INSERT w* (*INSERT v* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT v* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT w1* (*INSERT w* (*INSERT x EMPTY*))) ∧ ¬ *collinear* (*INSERT* *w1 (INSERT w (INSERT v EMPTY))) ∧ vector_norm (vector_sub v x) = real_of_nat (2::nat) ∧ vector_norm (vector_sub x u) = real_of_nat (2::nat) ∧ vector_norm (vector_sub u w) = real_of_nat (2::nat) ∧ vector_norm (vector_sub w w1) = real_of_nat (2::nat) ∧ vector_norm (vector_sub w1 v) = real_of_nat (2::nat) ⟶ (∃ (v1::(real, 3) cart) (u1::(real, 3) cart) w2::(real, 3) cart. IN u1 (INSERT v (INSERT x (INSERT u (INSERT w (INSERT w1 EMPTY))))) ∧ IN u1 (INSERT v (INSERT x (INSERT u (INSERT w (INSERT w1 EMPTY))))) ∧ IN w2 (INSERT v (INSERT x (INSERT u (INSERT w (INSERT w1 EMPTY))))) ∧ v1 ≠ u1 ∧ u1 ≠ w2 ∧ v1 ≠ w2 ∧ vector_norm (vector_sub v1 u1) ≤ (1::real) + sqrt (real_of_nat (5::nat)) ∧ vector_norm (vector_sub v1 w2) ≤ (1::real) + sqrt (real_of_nat (5::nat)))*

thm Vpwshto.POINTS_IN_BALL_ANNULUS_NOT_COLLINEAR:

SUBSET (INSERT (?u::(real, 3) cart) (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY))) ball_annulus ∧ packing (INSERT ?u (INSERT ?v (INSERT ?w EMPTY))) ∧ ?u ≠ ?v ∧ ?u ≠ ?w ∧ ?v ≠ ?w ⟶ ¬ IN ?v (closed_segment [(?u, ?w)])

thm Vpwshto.POINTS_IN_BALL_ANNULUS_NOT_COLLINEAR2:

SUBSET (INSERT (?u::(real, 3) cart) (INSERT (?v::(real, 3) cart) (INSERT (?w::(real, 3) cart) EMPTY))) ball_annulus ∧ packing (INSERT ?u (INSERT ?v (INSERT ?w EMPTY))) ∧ ?v ≠ ?u ∧ ?v ≠ ?w ∧ ?u ≠ ?w ⟶ ¬ collinear (INSERT ?u (INSERT ?v (INSERT ?w EMPTY)))

thm Vpwshto.SUBSET_PACKING:

∀ (sub::(real, 3) cart ⇒ bool) s::(real, 3) cart ⇒ bool. packing s ∧ SUBSET sub s ⟶ packing sub

thm Vpwshto.VPWSHTO_PRIME:

SUBSET (INSERT (?v::(real, 3) cart) (INSERT (?x::(real, 3) cart) (INSERT (?u::(real, 3) cart) (INSERT (?w::(real, 3) cart) (INSERT (?w1.0::(real, 3) cart) EMPTY))))) ball_annulus ∧ packing (INSERT ?v (INSERT ?x (INSERT ?u (INSERT ?w (INSERT ?w1.0 EMPTY))))) ∧ ?v ≠ ?x ∧ ?v ≠ ?u ∧ ?v ≠ ?w ∧ ?v ≠ ?w1.0 ∧ ?x ≠ ?u ∧ ?x ≠ ?w ∧ ?x ≠ ?w1.0 ∧ ?u ≠ ?w ∧ ?u ≠ ?w1.0 ∧ ?w ≠ ?w1.0 ∧ vector_norm (vector_sub ?v ?x) = real_of_nat (2::nat) ∧ vector_norm (vector_sub ?x ?u) = real_of_nat (2::nat) ∧ vector_norm (vector_sub ?u ?w) = real_of_nat (2::nat) ∧ vector_norm (vector_sub ?w ?w1.0) = real_of_nat (2::nat) ∧ vector_norm (vector_sub ?w1.0 ?v) = real_of_nat (2::nat) ⟶ (∃ (v1::(real, 3) cart) (u1::(real, 3) cart) w2::(real, 3) cart. IN u1 (INSERT ?v (INSERT ?x (INSERT ?u (INSERT ?w (INSERT ?w1.0 EMPTY))))) ∧ IN u1 (INSERT ?v (INSERT ?x (INSERT ?u (INSERT ?w (INSERT ?w1.0 EMPTY))))) ∧ IN w2 (INSERT ?v (INSERT ?x (INSERT ?u (INSERT ?w (INSERT ?w1.0 EMPTY))))) ∧ v1 ≠ u1 ∧ u1 ≠ w2 ∧ v1 ≠ w2 ∧ vector_norm (vector_sub v1 u1) ≤ (1::real) + sqrt (real_of_nat (5::nat)) ∧ vector_norm (vector_sub v1 w2) ≤ (1::real) + sqrt (real_of_nat (5::nat)))

thm Lfjcixp.LFJCIXP:

(∀ (*y1*::*real*) (*y2*::*real*) (*y3*::*real*) (*y4*::*real*) (*y5*::*real*) *y6*::*real*. *real_of_nat* (*2*::*nat*) ≤ *y1* ∧ *y1* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *y2* ∧ *y2* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *y3* ∧ *y3* ≤ *DECIMAL* (*252*::*nat*) (*100*::*nat*) ∧ *real_of_nat* (*2*::*nat*) ≤ *y4* ∧ *y4* ≤ *DECIMAL* (*452*::*nat*) (*100*::*nat*) ∧ *y5* = *real_of_nat* (*2*::*nat*) ∧ *y6* = *real_of_nat* (*2*::*nat*) ⟶ *y4* ≤ *DECIMAL* (*3915*::*nat*) (*1000*::*nat*) ∨ *delta* ($y1^2$) ($y2^2$) ($y3^2$) ($y4^2$) ($y5^2$) ($y6^2$) < (*0*::*real*)) ∧ *SUBSET* (*INSERT* (*?v*::(*real*, *3*) *cart*) (*INSERT* (*?u*::(*real*, *3*) *cart*) (*INSERT* (*?w*::(*real*, *3*) *cart*) *EMPTY*))) *ball_annulus* ∧ *packing* (*INSERT* *?v* (*INSERT* *?u* (*INSERT* *?w* *EMPTY*))) ∧ *?u* ≠ *?w* ∧ *vector_norm* (*vector_sub* *?v* *?u*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub* *?v* *?w*) = *real_of_nat* (*2*::*nat*) ∧ *vector_norm* (*vector_sub* *?u* *?w*) ≤ *DECIMAL* (*452*::*nat*) (*100*::*nat*) ⟶ *vector_norm* (*vector_sub* *?u* *?w*) ≤ *DECIMAL* (*3915*::*nat*) (*1000*::*nat*)

thm Polar_fan.AFF_GT_1_1:

∀ (*x*::(*real*, *?'b*::*type*) *cart*) (*v*::(*real*, *?'b*::*type*) *cart*) *w*::*?'a*::*type*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*) ⟶ *aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* *EMPTY*) = *GSPEC* (λ*GEN%PVAR%2215*::(*real*, *?'b*::*type*) *cart*. ∃ *y*::(*real*, *?'b*::*type*) *cart*. *SETSPEC* *GEN%PVAR%2215* (∃ (*t1*::*real*) *t2*::*real*. (*0*::*real*) < *t2* ∧ *t1* + *t2* = (*1*::*real*) ∧ *y* = *vector_add* (% *t1* *x*) (% *t2* *v*)) *y*)

thm Polar_fan.AFF_GT_1_2:

∀ (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *DISJOINT* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *w* *EMPTY*)) ⟶ *aff_gt* (*INSERT* *x* *EMPTY*) (*INSERT* *v* (*INSERT* *w* *EMPTY*)) = *GSPEC* (λ*GEN%PVAR%2216*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC* *GEN%PVAR%2216* (∃ (*t1*::*real*) (*t2*::*real*) *t3*::*real*. (*0*::*real*) < *t2* ∧ (*0*::*real*) < *t3* ∧ *t1* + (*t2* + *t3*) = (*1*::*real*) ∧ *y* = *vector_add* (% *t1* *x*) (*vector_add* (% *t2* *v*) (% *t3* *w*))) *y*)

thm Polar_fan.AFFSIGN_MONO_SHUFFLE:

∀ (*sgn*::*real* ⇒ *bool*) (*s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*t*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) (*s'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*) *t'*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. *HOL_Light_Import.UNION* *s'* *t'* = *HOL_Light_Import.UNION* *s* *t* ∧ *SUBSET* *t'* *t* ⟶ *SUBSET* (*affsign* *sgn* *s* *t*) (*affsign* *sgn* *s'* *t'*)

thm Polar_fan.AZIM_CYCLE_BASIC_PROPERTIES:

∀ (*W*::(*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *p*::(*real*, *3*) *cart*. *FINITE* *W* ∧ *IN* *p* *W* ⟶ *IN* (*azim_cycle* *W* *v* *w* *p*) *W* ∧ (∀ *q*::(*real*, *3*) *cart*. *IN* *q* *W* ∧ *q* ≠ *p* ⟶ *azim* *v* *w* *p* (*azim_cycle* *W* *v* *w* *p*) ≤ *azim* *v* *w* *p* *q*)

thm Polar_fan.AZIM_CYCLE_TWO_POINT_SET_ALT:

∀ (*W*::(*real*, *3*) *cart* ⇒ *bool*) (*x*::(*real*, *3*) *cart*) (*u*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *W* = *INSERT* *v* (*INSERT* *w* *EMPTY*) ⟶ *azim_cycle* *W* *x* *u* *v* = *w*

thm Polar_fan.IVS_AZIM_CYCLE_TWO_POINT_SET:

$\forall (a::(real,\ 3)\ cart)\ b::(real,\ 3)\ cart.\ ivs\_azim\_cycle\ (INSERT\ a\ (INSERT\ b\ EMPTY))\ (?v::(real,\ 3)\ cart)\ (?w::(real,\ 3)\ cart)\ a = b$

thm Polar_fan.IVS_AZIM_CYCLE_TWO_POINT_SET_ALT:

$\forall (W::(real,\ 3)\ cart \Rightarrow bool)\ (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ W = INSERT\ v\ (INSERT\ w\ EMPTY) \longrightarrow ivs\_azim\_cycle\ W\ x\ u\ v = w$

thm Polar_fan.AZIM_CYCLE_SING:

$\forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ azim\_cycle\ (INSERT\ v\ EMPTY)\ x\ u\ v = v$

thm Polar_fan.IVS_AZIM_CYCLE_SING:

$\forall (x::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ ivs\_azim\_cycle\ (INSERT\ v\ EMPTY)\ x\ u\ v = v$

thm Polar_fan.RHO_NODE1_INJECTIVE:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ local\_fan\ (V,\ E,\ FF) \longrightarrow (\forall (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ IN\ v\ V \wedge IN\ w\ V \longrightarrow (rho\_node1\ FF\ v = rho\_node1\ FF\ w) = (v = w))$

thm Polar_fan.IVS_RHO_NODE1_IN_V:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ local\_fan\ (V,\ E,\ FF) \longrightarrow (\forall v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow IN\ (ivs\_rho\_node1\ FF\ v)\ V)$

thm Polar_fan.LOCAL_FAN_ITER_IVS_RHO_NODE_IN_V:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool.\ local\_fan\ (V,\ E,\ FF) \wedge IN\ (?v::(real,\ 3)\ cart)\ V \longrightarrow (\forall i::nat.\ IN\ (ITER\ i\ (ivs\_rho\_node1\ FF)\ ?v)\ V)$

thm Polar_fan.LOCAL_FAN_ORBIT_MAP_EXPLICIT:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ local\_fan\ (V,\ E,\ FF) \wedge IN\ v\ V \wedge IN\ w\ V \longrightarrow (\exists i<CARD\ V.\ w = ITER\ i\ (rho\_node1\ FF)\ v)$

thm Polar_fan.LOCAL_FAN_ORBIT_MAP_EXPLICIT_IVS:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ local\_fan\ (V,\ E,\ FF) \wedge IN\ v\ V \wedge IN\ w\ V \longrightarrow (\exists i<CARD\ V.\ w = ITER\ i\ (ivs\_rho\_node1\ FF)\ v)$

thm Polar_fan.ITER_IVS_RHO_IDD:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (v::(real, 3) cart) n::nat. local_fan (V, E, FF) ∧ IN v V ⟶ ITER n (ivs_rho_node1 FF) (ITER n (rho_node1 FF) v) = v

thm Polar_fan.ITER_RHO_IVS_IDD:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (v::(real, 3) cart) n::nat. local_fan (V, E, FF) ∧ IN v V ⟶ ITER n (rho_node1 FF) (ITER n (ivs_rho_node1 FF) v) = v

thm Polar_fan.LOFA_IMP_ITER_IVS_RHO_NODE_ID:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) FF′::?′a::type. local_fan (V, E, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ⟶ (∀ v::(real, 3) cart. IN v V ⟶ ITER (CARD V) (ivs_rho_node1 ?FF) v = v)

thm Polar_fan.GENERIC_LOCAL_FAN_AZIM_POS:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. convex_local_fan (V, E, FF) ∧ generic V E ∧ (∀ v::(real, 3) cart. IN v V ⟶ interior_angle1 (vec (0::nat)) FF v < pi) ∧ IN v V ∧ IN w V ∧ w ≠ v ∧ w ≠ rho_node1 FF v ⟶ (0::real) < sin (azim (vec (0::nat)) v (rho_node1 FF v) w)

thm Polar_fan.nn_of_hyp3:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. nn_of_hyp (x, V, E) = GABS (λf::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart. ∀ (v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (if IN (v, w) (darts_of_hyp E V) then (v, azim_cycle (EE v E) x v w) else (v, w)))

thm Polar_fan.ff_of_hyp3:

∀ (x::(real, 3) cart) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. ff_of_hyp (x, V, E) = GABS (λf::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart. ∀ (v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (if IN (v, w) (darts_of_hyp E V) then (w, ivs_azim_cycle (EE w E) x w v) else (v, w)))

thm Polar_fan.ee_of_hyp3:

∀ (x::?′a::type) (V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. ee_of_hyp (x, V, E) = GABS (λf::(real, 3) cart × (real, 3) cart ⇒ (real, 3) cart × (real, 3) cart. ∀ (v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (if IN (v, w) (darts_of_hyp E V) then (w, v) else (v, w)))

thm Polar_fan.GMLWKPK:

∀ (x::(real, ?′a::type) cart) (V::(real, ?′a::type) cart ⇒ bool) E::((real, ?′a::type) cart ⇒ bool) ⇒ bool. graph E ⟶ fan7 (x, V, E) = (∀ (e1::(real, ?′a::type) cart ⇒ bool) e2::(real, ?′a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2217::(real, ?′a::type) cart ⇒ bool. ∃ v::(real,

$?'a::type$) cart. SETSPEC GEN%PVAR%2217 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2218::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2218 (IN v V) (INSERT v EMPTY)))) ⟶ (HOL_Light_Import.INTER e1 e2 = EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = INSERT x EMPTY) ∧ (∀ v::(real, ?'a::type) cart. HOL_Light_Import.INTER e1 e2 = INSERT v EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (INSERT v EMPTY)))

thm Polar_fan.FAN_ECONOMIZED:

∀ (x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. FAN (x, V, E) = (SUBSET (UNIONS E) V ∧ graph E ∧ fan1 (x, V, E) ∧ fan2 (x, V, E) ∧ fan6 (x, V, E) ∧ (∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2219::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2219 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2220::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2220 (IN v V) (INSERT v EMPTY)))) ⟶ (HOL_Light_Import.INTER e1 e2 = EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = INSERT x EMPTY) ∧ (∀ v::(real, ?'a::type) cart. HOL_Light_Import.INTER e1 e2 = INSERT v EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (INSERT v EMPTY))))

thm Polar_fan.FAN7_AFF_GT_CONDITION:

∀ (x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. graph E ∧ ¬ IN x V ∧ (∀ e::(real, ?'a::type) cart ⇒ bool. IN e E ⟶ SUBSET e V ∧ ¬ IN x e) ∧ (∀ (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. IN v V ∧ IN w V ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) (INSERT v EMPTY)) (aff_ge (INSERT x EMPTY) (INSERT w EMPTY)) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER (INSERT v EMPTY) (INSERT w EMPTY))) ∧ (∀ (v::(real, ?'a::type) cart) e::(real, ?'a::type) cart ⇒ bool. IN v V ∧ IN e E ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v EMPTY)) (aff_gt (INSERT x EMPTY) e) = EMPTY) ∧ (∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 E ∧ IN e2 E ∧ e1 ≠ e2 ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) e1) (aff_gt (INSERT x EMPTY) e2) = EMPTY) ⟶ fan7 (x, V, E)

thm Polar_fan.FAN_AFF_GT_CONDITION:

∀ (x::(real, ?'a::type) cart) (V::(real, ?'a::type) cart ⇒ bool) E::((real, ?'a::type) cart ⇒ bool) ⇒ bool. SUBSET (UNIONS E) V ∧ graph E ∧ fan1 (x, V, E) ∧ fan2 (x, V, E) ∧ fan6 (x, V, E) ∧ (∀ (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. IN v V ∧ IN w V ⟶ HOL_Light_Import.INTER (aff_ge

(INSERT x EMPTY) (INSERT v EMPTY)) (aff_ge (INSERT x EMPTY) (INSERT w EMPTY)) = aff_ge (INSERT x EMPTY) (HOL_Light_Import.INTER (INSERT v EMPTY) (INSERT w EMPTY))) ∧ (∀ (v::(real, ?'a::type) cart) e::(real, ?'a::type) cart ⇒ bool. IN v V ∧ IN e E ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) (INSERT v EMPTY)) (aff_gt (INSERT x EMPTY) e) = EMPTY) ∧ (∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 E ∧ IN e2 E ∧ e1 ≠ e2 ⟶ HOL_Light_Import.INTER (aff_gt (INSERT x EMPTY) e1) (aff_gt (INSERT x EMPTY) e2) = EMPTY) ⟶ FAN (x, V, E)

thm DEF_polar_fan:

polar_fan = (λ_7157268::((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool) × ((real, 3) cart × (real, 3) cart ⇒ bool). LET (λr::(real, 3) cart ⇒ (real, 3) cart. LET_END (LET (λprime::(real, 3) cart ⇒ (real, 3) cart. LET_END (GSPEC (λGEN%PVAR%2223::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2223 (IN v (fst _7157268)) (prime v)), GSPEC (λGEN%PVAR%2224::(real, 3) cart ⇒ bool. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2224 (IN v (fst _7157268)) (INSERT (prime v) (INSERT (prime (r v)) EMPTY))), GSPEC (λGEN%PVAR%2225::(real, 3) cart × (real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2225 (IN v (fst _7157268)) (prime v, prime (r v))))) (λv::(real, 3) cart. cross v (r v)))) (rho_node1 (snd (snd _7157268))))

thm Polar_fan.JNVXCRC:

∀ (E::((real, 3) cart ⇒ bool) ⇒ bool) (V::(real, 3) cart ⇒ bool) FF::(real, 3) cart × (real, 3) cart ⇒ bool. polar_fan (V, E, FF) = LET (λr::(real, 3) cart ⇒ (real, 3) cart. LET_END (LET (λprime::(real, 3) cart ⇒ (real, 3) cart. LET_END (GSPEC (λGEN%PVAR%2223::(real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2223 (IN v V) (prime v)), GSPEC (λGEN%PVAR%2224::(real, 3) cart ⇒ bool. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2224 (IN v V) (INSERT (prime v) (INSERT (prime (r v)) EMPTY))), GSPEC (λGEN%PVAR%2225::(real, 3) cart × (real, 3) cart. ∃ v::(real, 3) cart. SETSPEC GEN%PVAR%2225 (IN v V) (prime v, prime (r v))))) (λv::(real, 3) cart. cross v (r v)))) (rho_node1 FF)

thm Polar_fan.BGMIFTE:

∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (V'::(real, 3) cart ⇒ bool) (E'::((real, 3) cart ⇒ bool) ⇒ bool) FF'::(real, 3) cart × (real, 3) cart ⇒ bool. convex_local_fan (V, E, FF) ∧ generic V E ∧ (∀ v::(real, 3) cart. IN v V ⟶ interior_angle1 (vec (0::nat)) FF v < pi) ∧ (V', E', FF') = polar_fan (V, E, FF) ⟶ convex_local_fan (V', E', FF') ∧ generic V' E' ∧ CARD V' = CARD V ∧ LET (λr::(real, 3) cart ⇒ (real, 3) cart. LET_END (LET (λprime::(real, 3) cart ⇒ (real, 3) cart. LET_END ((∀ v::(real, 3) cart. IN v V ⟶ arcV (vec (0::nat)) (prime v) (prime (r v)) = pi − interior_angle1 (vec (0::nat)) FF (r v) ∧ (0::real) < arcV (vec (0::nat)) (prime v) (prime (r v)) ∧ arcV

$(vec\ (0::nat))\ (prime\ v)\ (prime\ (r\ v)) < pi) \wedge (\forall\, v::(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow arcV\ (vec\ (0::nat))\ v\ (r\ v) = pi - interior\_angle1\ (vec\ (0::nat))\ FF'\ (prime\ v) \wedge (0::real) < arcV\ (vec\ (0::nat))\ v\ (r\ v) \wedge arcV\ (vec\ (0::nat))\ v\ (r\ v) < pi)))\ (\lambda v::(real,\ 3)\ cart.\ cross\ v\ (r\ v))))\ (rho\_node1\ FF)$

thm DEF_fan_perimeter:

$fan\_perimeter = (\lambda\_7355572::((real,\ 3)\ cart \Rightarrow bool) \times (((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool) \times ((real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool).\ LET\ (\lambda v::(real,\ 3)\ cart.\ LET\_END\ (sum\ (dotdot\ (0::nat)\ (CARD\ (snd\ (snd\ \_7355572)) - (1::nat)))\ (\lambda i::nat.\ arcV\ (vec\ (0::nat))\ (ITER\ i\ (rho\_node1\ (snd\ (snd\ \_7355572)))\ v)\ (ITER\ (i + (1::nat))\ (rho\_node1\ (snd\ (snd\ \_7355572)))\ v)))))\ (SOME\ v::(real,\ 3)\ cart.\ IN\ v\ (fst\ \_7355572)))$

thm Polar_fan.IQCPCGW:

$\forall\, (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ V::(real,\ 3)\ cart \Rightarrow bool.\ fan\_perimeter\ (V,\ E,\ FF) = LET\ (\lambda v::(real,\ 3)\ cart.\ LET\_END\ (sum\ (dotdot\ (0::nat)\ (CARD\ FF - (1::nat)))\ (\lambda i::nat.\ arcV\ (vec\ (0::nat))\ (ITER\ i\ (rho\_node1\ FF)\ v)\ (ITER\ (i + (1::nat))\ (rho\_node1\ FF)\ v))))\ (SOME\ v::(real,\ 3)\ cart.\ IN\ v\ V)$

thm Polar_fan.FAN_PERIMETER_INVARIANT:

$\forall\, (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF::(real,\ 3)\ cart \times (real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ local\_fan\ (V,\ E,\ FF) \wedge IN\ v\ V \longrightarrow fan\_perimeter\ (V,\ E,\ FF) = sum\ (dotdot\ (0::nat)\ (CARD\ FF - (1::nat)))\ (\lambda i::nat.\ arcV\ (vec\ (0::nat))\ (ITER\ i\ (rho\_node1\ FF)\ v)\ (ITER\ (i + (1::nat))\ (rho\_node1\ FF)\ v))$

thm DEF_a_ear0:

$a\_ear0 = (\lambda(\_7355947::(nat \Rightarrow bool) \Rightarrow bool)\ \_7355948::nat \times nat.\ if\ fst\ \_7355948\ mod\ (3::nat) = snd\ \_7355948\ mod\ (3::nat)\ then\ 0::real\ else\ if\ IN\ (INSERT\ (fst\ \_7355948\ mod\ (3::nat))\ (INSERT\ (snd\ \_7355948\ mod\ (3::nat))\ EMPTY))\ \_7355947\ then\ sqrt\ (real\_of\_nat\ (8::nat))\ else\ real\_of\_nat\ (2::nat))$

thm Hdplygy.a_ear0:

$\forall\, (i::nat)\ (j::nat)\ J::(nat \Rightarrow bool) \Rightarrow bool.\ a\_ear0\ J\ (i,\ j) = (if\ i\ mod\ (3::nat) = j\ mod\ (3::nat)\ then\ 0::real\ else\ if\ IN\ (INSERT\ (i\ mod\ (3::nat))\ (INSERT\ (j\ mod\ (3::nat))\ EMPTY))\ J\ then\ sqrt\ (real\_of\_nat\ (8::nat))\ else\ real\_of\_nat\ (2::nat))$

thm DEF_b_ear0:

$b\_ear0 = (\lambda(\_7355964::(nat \Rightarrow bool) \Rightarrow bool)\ \_7355965::nat \times nat.\ if\ fst\ \_7355965\ mod\ (3::nat) = snd\ \_7355965\ mod\ (3::nat)\ then\ 0::real\ else\ if\ IN\ (INSERT\ (fst\ \_7355965\ mod\ (3::nat))\ (INSERT\ (snd\ \_7355965\ mod\ (3::nat))\ EMPTY))\ \_7355964\ then\ cstab\ else\ real\_of\_nat\ (2::nat) * h0)$

thm Hdplygy.b_ear0:

$\forall (i::nat)\ (j::nat)\ J::(nat \Rightarrow bool) \Rightarrow bool.\ b\_ear0\ J\ (i,\ j) = (if\ i\ mod\ (3::nat) = j\ mod\ (3::nat)\ then\ 0::real\ else\ if\ IN\ (INSERT\ (i\ mod\ (3::nat))\ (INSERT\ (j\ mod\ (3::nat))\ EMPTY))\ J\ then\ cstab\ else\ real\_of\_nat\ (2::nat) * h0)$

thm Hdplygy.MOD_EQ_MOD1:

$\forall (x1::nat)\ (x2::nat)\ (y::nat)\ n::nat.\ n \neq (0::nat) \land (y + x1)\ mod\ n = (y + x2)\ mod\ n \land x2 \leq x1 \longrightarrow x1\ mod\ n = x2\ mod\ n$

thm Hdplygy.MOD_EQ_MOD:

$\forall (x1::nat)\ (x2::nat)\ (y::nat)\ n::nat.\ n \neq (0::nat) \land (y + x1)\ mod\ n = (y + x2)\ mod\ n \longrightarrow x1\ mod\ n = x2\ mod\ n$

thm Hdplygy.EAR_STABLE_SYSTEM:

$stable\_system\ (3::nat)\ (DECIMAL\ (11::nat)\ (100::nat))\ (dotdot\ (0::nat)\ (2::nat))\ (a\_ear0\ (INSERT\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ EMPTY))\ EMPTY))\ (b\_ear0\ (INSERT\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ EMPTY))\ EMPTY))\ (INSERT\ (INSERT\ (1::nat)\ (INSERT\ (2::nat)\ EMPTY))\ EMPTY)\ (\lambda i::nat.\ ((1::nat) + i)\ mod\ (3::nat))$

thm Hdplygy.exist_stable_system:

$\exists s::nat \times real \times (nat \Rightarrow bool) \times (nat \times nat \Rightarrow real) \times (nat \times nat \Rightarrow real) \times ((nat \Rightarrow bool) \Rightarrow bool) \times (nat \Rightarrow nat).\ stable\_system\ (fst\ s)\ (fst\ (snd\ s))\ (fst\ (snd\ (snd\ s)))\ (fst\ (snd\ (snd\ (snd\ s))))\ (fst\ (snd\ (snd\ (snd\ (snd\ s)))))\ (fst\ (snd\ (snd\ (snd\ (snd\ (snd\ s))))))\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ s))))))$

thm TYDEF_stable_sy:

$stable\_sy\ (tuple\_stable\_sy\ (?a::stable\_sy)) = ?a \land stable\_system\ (fst\ (?r::nat \times real \times (nat \Rightarrow bool) \times (nat \times nat \Rightarrow real) \times (nat \times nat \Rightarrow real) \times ((nat \Rightarrow bool) \Rightarrow bool) \times (nat \Rightarrow nat)))\ (fst\ (snd\ ?r))\ (fst\ (snd\ (snd\ ?r)))\ (fst\ (snd\ (snd\ (snd\ ?r))))\ (fst\ (snd\ (snd\ (snd\ (snd\ ?r)))))\ (fst\ (snd\ (snd\ (snd\ (snd\ (snd\ ?r))))))\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ ?r)))))) = (tuple\_stable\_sy\ (stable\_sy\ ?r) = ?r)$

thm Hdplygy.stable_sy_tybij_conjunct1:

$\forall r::nat \times real \times (nat \Rightarrow bool) \times (nat \times nat \Rightarrow real) \times (nat \times nat \Rightarrow real) \times ((nat \Rightarrow bool) \Rightarrow bool) \times (nat \Rightarrow nat).\ stable\_system\ (fst\ r)\ (fst\ (snd\ r))\ (fst\ (snd\ (snd\ r)))\ (fst\ (snd\ (snd\ (snd\ r))))\ (fst\ (snd\ (snd\ (snd\ (snd\ r)))))\ (fst\ (snd\ (snd\ (snd\ (snd\ (snd\ r))))))\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ r)))))) = (tuple\_stable\_sy\ (stable\_sy\ r) = r)$

thm Hdplygy.stable_sy_tybij_conjunct0:

$\forall a::stable\_sy.\ stable\_sy\ (tuple\_stable\_sy\ a) = a$

thm Hdplygy.stable_sy_tybij:

$(\forall a::stable\_sy.\ stable\_sy\ (tuple\_stable\_sy\ a) = a) \land (\forall r::nat \times real \times (nat \Rightarrow bool) \times (nat \times nat \Rightarrow real) \times (nat \times nat \Rightarrow real) \times ((nat \Rightarrow bool) \Rightarrow bool)$

$\times$ (*nat* $\Rightarrow$ *nat*). *stable_system* (*fst r*) (*fst* (*snd r*)) (*fst* (*snd* (*snd r*))) (*fst* (*snd* (*snd* (*snd r*)))) (*fst* (*snd* (*snd* (*snd* (*snd r*))))) (*fst* (*snd* (*snd* (*snd* (*snd* (*snd r*)))))) (*snd* (*snd* (*snd* (*snd* (*snd* (*snd r*)))))) = (*tuple_stable_sy* (*stable_sy r*) = *r*))

thm DEF_k_sy:

k_sy = (λ*_7357047*::*stable_sy*. *fst* (*tuple_stable_sy _7357047*))

thm Hdplygy.k_sy:

∀ *s*::*stable_sy*. *k_sy s* = *fst* (*tuple_stable_sy s*)

thm DEF_d_sy:

d_sy = (λ*_7357052*::*stable_sy*. *fst* (*snd* (*tuple_stable_sy _7357052*)))

thm Hdplygy.d_sy:

∀ *s*::*stable_sy*. *d_sy s* = *fst* (*snd* (*tuple_stable_sy s*))

thm DEF_I_SY:

I_SY = (λ*_7357057*::*stable_sy*. *fst* (*snd* (*snd* (*tuple_stable_sy _7357057*))))

thm Hdplygy.I_SY:

∀ *s*::*stable_sy*. *I_SY s* = *fst* (*snd* (*snd* (*tuple_stable_sy s*)))

thm DEF_a_sy:

a_sy = (λ*_7357062*::*stable_sy*. *fst* (*snd* (*snd* (*snd* (*tuple_stable_sy _7357062*)))))

thm Hdplygy.a_sy:

∀ *s*::*stable_sy*. *a_sy s* = *fst* (*snd* (*snd* (*snd* (*tuple_stable_sy s*))))

thm DEF_b_sy:

b_sy = (λ*_7357067*::*stable_sy*. *fst* (*snd* (*snd* (*snd* (*snd* (*tuple_stable_sy _7357067*))))))

thm Hdplygy.b_sy:

∀ *s*::*stable_sy*. *b_sy s* = *fst* (*snd* (*snd* (*snd* (*snd* (*tuple_stable_sy s*)))))

thm DEF_J_SY:

J_SY = (λ*_7357072*::*stable_sy*. *fst* (*snd* (*snd* (*snd* (*snd* (*snd* (*tuple_stable_sy _7357072*)))))))

thm Hdplygy.J_SY:

∀ *s*::*stable_sy*. *J_SY s* = *fst* (*snd* (*snd* (*snd* (*snd* (*snd* (*tuple_stable_sy s*))))))

thm DEF_f_sy:

f_sy = (λ*_7357077*::*stable_sy*. *snd* (*snd* (*snd* (*snd* (*snd* (*snd* (*tuple_stable_sy _7357077*)))))))

thm Hdplygy.f_sy:

$\forall s::stable\_sy.\ f\_sy\ s = snd\ (snd\ (snd\ (snd\ (snd\ (snd\ (tuple\_stable\_sy\ s))))))$

thm Hdplygy.stable_sy_lemma:

$\forall s::stable\_sy.\ stable\_system\ (k\_sy\ s)\ (d\_sy\ s)\ (I\_SY\ s)\ (a\_sy\ s)\ (b\_sy\ s)\ (J\_SY\ s)\ (f\_sy\ s)$

thm DEF_ear_sy:

$ear\_sy = (\lambda\_7357082::stable\_sy.\ CARD\ (I\_SY\ \_7357082) = (3::nat) \wedge d\_sy\ \_7357082 = DECIMAL\ (11::nat)\ (100::nat) \wedge CARD\ (J\_SY\ \_7357082) = (1::nat) \wedge a\_sy\ \_7357082 = a\_ear0\ (J\_SY\ \_7357082) \wedge b\_sy\ \_7357082 = b\_ear0\ (J\_SY\ \_7357082))$

thm Hdplygy.ear_sy:

$\forall s::stable\_sy.\ ear\_sy\ s = (CARD\ (I\_SY\ s) = (3::nat) \wedge d\_sy\ s = DECIMAL\ (11::nat)\ (100::nat) \wedge CARD\ (J\_SY\ s) = (1::nat) \wedge a\_sy\ s = a\_ear0\ (J\_SY\ s) \wedge b\_sy\ s = b\_ear0\ (J\_SY\ s))$

thm DEF_sigma_sy:

$sigma\_sy = (\lambda\_7357087::stable\_sy.\ if\ ear\_sy\ \_7357087\ then\ 1::real\ else - (1::real))$

thm Hdplygy.sigma_sy:

$\forall s::stable\_sy.\ sigma\_sy\ s = (if\ ear\_sy\ s\ then\ 1::real\ else - (1::real))$

thm DEF_J1_SY:

$J1\_SY = (\lambda\_7357092::stable\_sy.\ GSPEC\ (\lambda GEN\%PVAR\%2229::nat \times nat.\ \exists x::nat \times nat.\ SETSPEC\ GEN\%PVAR\%2229\ (\exists i::nat.\ IN\ (INSERT\ (i\ mod\ k\_sy\ \_7357092)\ (INSERT\ (f\_sy\ \_7357092\ (i\ mod\ k\_sy\ \_7357092))\ EMPTY))\ (J\_SY\ \_7357092) \wedge IN\ i\ (dotdot\ (1::nat)\ (k\_sy\ \_7357092)) \wedge x = (i,\ Suc\ (i\ mod\ k\_sy\ \_7357092)))\ x))$

thm Hdplygy.J1_SY:

$\forall s::stable\_sy.\ J1\_SY\ s = GSPEC\ (\lambda GEN\%PVAR\%2229::nat \times nat.\ \exists x::nat \times nat.\ SETSPEC\ GEN\%PVAR\%2229\ (\exists i::nat.\ IN\ (INSERT\ (i\ mod\ k\_sy\ s)\ (INSERT\ (f\_sy\ s\ (i\ mod\ k\_sy\ s))\ EMPTY))\ (J\_SY\ s) \wedge IN\ i\ (dotdot\ (1::nat)\ (k\_sy\ s)) \wedge x = (i,\ Suc\ (i\ mod\ k\_sy\ s)))\ x)$

thm DEF_d_fun:

$d\_fun = (\lambda\_7357097::stable\_sy \times (real,\ (?'a::type,\ 3)\ finite\_product)\ cart.\ d\_sy\ (fst\ \_7357097) + DECIMAL\ (1::nat)\ (10::nat) * (sigma\_sy\ (fst\ \_7357097) * sum\ (J1\_SY\ (fst\ \_7357097))\ (\lambda x::nat \times nat.\ cstab - vector\_norm\ (vector\_sub\ (row\ (fst\ x)\ (vecmats\ (snd\ \_7357097)))\ (row\ (snd\ x)\ (vecmats\ (snd\ \_7357097)))))))$

thm Hdplygy.d_fun:

$\forall (s::stable\_sy)\ l::(real,\ (?'a::type,\ 3)\ finite\_product)\ cart.\ d\_fun\ (s,\ l) = d\_sy\ s + DECIMAL\ (1::nat)\ (10::nat) * (sigma\_sy\ s * sum\ (J1\_SY\ s)\ (\lambda x::nat \times$

nat. cstab − vector_norm (vector_sub (row (fst x) (vecmats l)) (row (snd x) (vecmats l)))))

thm DEF_tau_star:

tau_star = (λ(_7357106::stable_sy) _7357107::(real, (?'a::type, 3) finite_product) cart. tau_fun (V_SY (vecmats _7357107)) (E_SY (vecmats _7357107)) (F_SY (vecmats _7357107)) − d_fun (_7357106, _7357107))

thm Hdplygy.tau_star:

∀ (s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. tau_star s l = tau_fun (V_SY (vecmats l)) (E_SY (vecmats l)) (F_SY (vecmats l)) − d_fun (s, l)

thm Hdplygy.FINITE_J_SY:

∀ s::stable_sy. FINITE (J_SY s)

thm Hdplygy.FINITE_J1_SY:

∀ s::stable_sy. FINITE (J1_SY s)

thm Hdplygy.CONTINUOUS_ON_ROW:

∀ (i::nat) s::(real, (?'b::type, ?'a::type) finite_product) cart ⇒ bool. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ continuous_on (λx::(real, (?'b::type, ?'a::type) finite_product) cart. row i (vecmats x)) s

thm Hdplygy.INDEX_J1_SY:

∀ s::stable_sy. IN (?x::nat × nat) (J1_SY s) ⟶ (1::nat) ≤ fst ?x ∧ fst ?x ≤ k_sy s

thm Hdplygy.CONTINUOUS_ON_D_FUN:

∀ s::stable_sy. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ⟶ continuous_on (lift ∘ (λl::(real, (?'a::type, 3) finite_product) cart. d_fun (s, l))) (B_SY1 (a_sy s) (b_sy s))

thm Hdplygy.INJ_B_SY:

∀ (s::stable_sy) x::(real, (?'a::type, 3) finite_product) cart. IN x (B_SY1 (a_sy s) (b_sy s)) ∧ k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ⟶ (∀ (i::nat) j::nat. IN i (dotdot (1::nat) ?k) ∧ IN j (dotdot (1::nat) ?k) ∧ (row i (vecmats x), row (Suc (i mod ?k)) (vecmats x)) = (row j (vecmats x), row (Suc (j mod ?k)) (vecmats x)) ⟶ i = j)

thm Hdplygy.INJ_ROW_B_SY:

∀ (s::stable_sy) x::(real, (?'a::type, 3) finite_product) cart. IN x (B_SY1 (a_sy s) (b_sy s)) ∧ k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) +

i) mod ?k) ∧ (2::nat) < ?k ⟶ (∀ (i::nat) j::nat. IN i (dotdot (1::nat) ?k) ∧ IN j (dotdot (1::nat) ?k) ∧ row i (vecmats x) = row j (vecmats x) ⟶ i = j)

thm Hdplygy.CHANGE_SUM_TAU_FUN:

∀ (s::stable_sy) x::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ IN x (B_SY1 (a_sy s) (b_sy s)) ⟶ sum (F_SY (vecmats x)) (λe::(real, 3) cart × (real, 3) cart. rho_fun (vector_norm (fst e)) ∗ azim_in_fan e (E_SY (vecmats x))) = sum (dotdot (1::nat) (k_sy s)) (λi::nat. rho_fun (vector_norm (row i (vecmats x))) ∗ azim_in_fan (row i (vecmats x), row (Suc (i mod k_sy s)) (vecmats x)) (E_SY (vecmats x)))

thm Hdplygy.CONTINUOUS_ON_SAME_DOMAIN:

∀ (f::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) (g::(real, ?'b::type) cart ⇒ (real, ?'a::type) cart) s::(real, ?'b::type) cart ⇒ bool. (∀ x::(real, ?'b::type) cart. IN x s ⟶ f x = g x) ∧ continuous_on f s ⟶ continuous_on g s

thm Hdplygy.EDGE_IN_F_SY:

∀ l::(real, (?'a::type, 3) finite_product) cart. (1::nat) ≤ (?i::nat) ∧ ?i ≤ dimindex HOL_Light_Import.UNIV ∧ (?u::(real, 3) cart) = row ?i (vecmats l) ∧ (?v::(real, 3) cart) = row (Suc (?i mod dimindex HOL_Light_Import.UNIV)) (vecmats l) ⟶ IN (?u, ?v) (F_SY (vecmats l))

thm Hdplygy.JBDNJJB3:

∀ (u::(real, 3) cart) (v::(real, 3) cart) w::(real, 3) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT u (INSERT v EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT u (INSERT w EMPTY))) ⟶ sin (azim (vec (0::nat)) u v w) = inverse_class.inverse (sqrt ((vector_norm w)^2 − (inverse_class.inverse (vector_norm u) ∗ dot u w)^2) ∗ vector_norm (cross u v)) ∗ dot (cross u v) w

thm Hdplygy.SEQUENTIALLY_DIVH:

∀ (f::nat ⇒ (real, 3) cart) (g::nat ⇒ (real, 3) cart) h::nat ⇒ (real, 3) cart. −−> f (?a::(real, 3) cart) sequentially ∧ −−> g (?b::(real, 3) cart) sequentially ∧ −−> h (?c::(real, 3) cart) sequentially ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?a (INSERT ?b EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT ?a (INSERT ?c EMPTY))) ∧ (∀ n::nat. ¬ collinear (INSERT (vec (0::nat)) (INSERT (f n) (INSERT (g n) EMPTY))) ∧ ¬ collinear (INSERT (vec (0::nat)) (INSERT (f n) (INSERT (h n) EMPTY)))) ⟶ −−> (lift ∘ (λn::nat. dihV (vec (0::nat)) (f n) (g n) (h n))) (lift (dihV (vec (0::nat)) ?a ?b ?c)) sequentially

thm Hdplygy.COLLINEAR_B_SY:

∀ (s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ IN *l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *?k* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*row ?i* (*vecmats l*)) (*INSERT* (*row* (*Suc* (*?i mod k_sy s*)) (*vecmats l*)) *EMPTY*)))

thm Hdplygy.COLLINEAR_AZIM_CYCLE_B_SY:

∀ (*s*::*stable_sy*) *l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ (*2*::*nat*) < *?k* ∧ *IN l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ∧ (*1*::*nat*) ≤ (*?i*::*nat*) ∧ *?i* ≤ *?k* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* (*row ?i* (*vecmats l*)) (*INSERT* (*azim_cycle* (*EE* (*row ?i* (*vecmats l*)) (*E_SY* (*vecmats l*))) (*vec* (*0*::*nat*)) (*row ?i* (*vecmats l*)) (*row* (*Suc* (*?i mod k_sy s*)) (*vecmats l*))) *EMPTY*)))

thm Hdplygy.AZIM_EQ_DIHV_IN_B_SY:

∀ (*s*::*stable_sy*) (*l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*) *i*::*nat*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ (*2*::*nat*) < *?k* ∧ (*1*::*nat*) ≤ *i* ∧ *i* ≤ *?k* ∧ *row i* (*vecmats l*) = (*?u*::(*real*, *3*) *cart*) ∧ *row* (*Suc* (*i mod k_sy s*)) (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *azim_cycle* (*EE ?u* (*E_SY* (*vecmats l*))) (*vec* (*0*::*nat*)) *?u ?v* = (*?w*::(*real*, *3*) *cart*) ∧ *IN l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ⟶ *azim* (*vec* (*0*::*nat*)) *?u ?v ?w* = *dihV* (*vec* (*0*::*nat*)) *?u ?v ?w*

thm Hdplygy.CONTINUOUS_ON_RHO_FUN_AND_AZIM:

∀ *s*::*stable_sy*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ (*2*::*nat*) < *?k* ⟶ *continuous_on* (*lift* ∘ (λ*x*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *sum* (*dotdot* (*1*::*nat*) (*k_sy s*)) (λ*i*::*nat*. *rho_fun* (*vector_norm* (*row i* (*vecmats x*))) ∗ *azim_in_fan* (*row i* (*vecmats x*), *row* (*Suc* (*i mod k_sy s*)) (*vecmats x*)) (*E_SY* (*vecmats x*))))) (*B_SY1* (*a_sy s*) (*b_sy s*))

thm Hdplygy.CONTINUOUS_ON_TAU_FUN:

∀ *s*::*stable_sy*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ (*2*::*nat*) < *?k* ⟶ *continuous_on* (*lift* ∘ (λ*l*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *tau_fun* (*V_SY* (*vecmats l*)) (*E_SY* (*vecmats l*)) (*F_SY* (*vecmats l*)))) (*B_SY1* (*a_sy s*) (*b_sy s*))

thm Hdplygy.CONTINUOUS_ON_TAU_STAR:

∀ *s*::*stable_sy*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ (*2*::*nat*) < *?k* ⟶ *continuous_on* (*lift* ∘ *tau_star s*) (*B_SY1* (*a_sy s*) (*b_sy s*))

thm Hdplygy.MINIMUM_IN_B_SY:

∀ *s*::*stable_sy*. *k_sy s* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*)

+ i) mod ?k) ∧ (2::nat) < ?k ∧ B_SY1 (a_sy s) (b_sy s) ≠ EMPTY ⟶ (∃ x::(real, (?'a::type, 3) finite_product) cart. IN x (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ y::(real, (?'a::type, 3) finite_product) cart. IN y (B_SY1 (a_sy s) (b_sy s)) ⟶ tau_star s x ≤ tau_star s y))

thm Hdplygy.HDPLYGY:

∀ s::stable_sy. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ B_SY1 (a_sy s) (b_sy s) ≠ EMPTY ⟶ continuous_on (lift ∘ tau_star s) (B_SY1 (a_sy s) (b_sy s)) ∧ (∃ x::(real, (?'a::type, 3) finite_product) cart. IN x (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ y::(real, (?'a::type, 3) finite_product) cart. IN y (B_SY1 (a_sy s) (b_sy s)) ⟶ tau_star s x ≤ tau_star s y))

thm Gbycpxs.CARD_F_SY_IN_B_SY:

∀ (s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ CARD (F_SY (vecmats l)) = ?k

thm Gbycpxs.SIGMA_SY_LE1:

∀ s::stable_sy. sigma_sy s ≤ (1::real)

thm Gbycpxs.B_SY_LE_CSTAB:

∀ (s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ ?k ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ vector_norm (vector_sub (row ?i (vecmats l)) (row (Suc (?i mod ?k)) (vecmats l))) ≤ cstab

thm Gbycpxs.PROPERTIES_EAR_SY:

∀ s::stable_sy. ear_sy s ⟶ (∃ i::nat. J_SY s = INSERT (INSERT i (INSERT (f_sy s i) EMPTY)) EMPTY ∧ IN i (I_SY s))

thm Gbycpxs.SING_J1_SY:

∀ s::stable_sy. ear_sy s ∧ I_SY s = dotdot (0::nat) (k_sy s − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod (?k::nat)) ∧ k_sy s = ?k ∧ (2::nat) < ?k ⟶ (∃ i::nat. J1_SY s = INSERT (i, Suc (i mod k_sy s)) EMPTY ∧ J_SY s = INSERT (INSERT (i mod ?k) (INSERT (f_sy s i) EMPTY)) EMPTY ∧ i ≤ ?k ∧ (1::nat) ≤ i)

thm Gbycpxs.D_FUN_LE:

∀ (s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ d_sy s ≤ DECIMAL (9::nat) (10::nat) ∧ pi ≤ sol_local (E_SY (vecmats l)) (F_SY (vecmats l)) ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ d_fun (s, l) ≤ DECIMAL (92::nat) (100::nat)

thm Gbycpxs.TAU_FUN_LE:

∀(s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ pi ≤ sol_local (E_SY (vecmats l)) (F_SY (vecmats l)) ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ DECIMAL (92::nat) (100::nat) < tau_fun (V_SY (vecmats l)) (E_SY (vecmats l)) (F_SY (vecmats l))

thm Gbycpxs.TAU_STAR_POS:

∀(s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ d_sy s ≤ DECIMAL (9::nat) (10::nat) ∧ pi ≤ sol_local (E_SY (vecmats l)) (F_SY (vecmats l)) ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ (0::real) < tau_star s l

thm Gbycpxs.CIRCULAR_SOL_EQ_2PI:

convex_local_fan (?V::(real, 3) cart ⇒ bool, ?E::((real, 3) cart ⇒ bool) ⇒ bool, ?FF::(real, 3) cart × (real, 3) cart ⇒ bool) ∧ circular ?V ?E ⟶ sol_local ?E ?FF = real_of_nat (2::nat) ∗ pi

thm Gbycpxs.NOT_CIRCULAR_SY:

∀(s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ d_sy s ≤ DECIMAL (9::nat) (10::nat) ∧ tau_star s l ≤ (0::real) ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ ¬ circular (V_SY (vecmats l)) (E_SY (vecmats l))

thm Gbycpxs.GBYCPXS:

∀(s::stable_sy) l::(real, (?'a::type, 3) finite_product) cart. k_sy s = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ (2::nat) < ?k ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ (d_sy s ≤ DECIMAL (9::nat) (10::nat) ∧ pi ≤ sol_local (E_SY (vecmats l)) (F_SY (vecmats l)) ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ⟶ (0::real) < tau_star s l) ∧ (d_sy s ≤ DECIMAL (9::nat) (10::nat) ∧ tau_star s l ≤ (0::real) ⟶ ¬ circular (V_SY (vecmats l)) (E_SY (vecmats l)))

thm DEF_IS_SY:

IS_SY = (λ(_7405238::stable_sy) (_7405239::nat) _7405240::nat. GSPEC (λGEN%PVAR%2236::nat. ∃x::nat. SETSPEC GEN%PVAR%2236 (∃(n::nat) m::nat. n < m ∧ m < k_sy _7405238 ∧ POWER (f_sy _7405238) n x = _7405240 ∧ POWER (f_sy _7405238) m x = _7405239) x))

thm Mtuwlun.IS_SY:

$\forall (q::nat)\ (s::stable\_sy)\ p::nat.\ IS\_SY\ s\ p\ q = GSPEC\ (\lambda GEN\%PVAR\%2236::nat.\ \exists\, x::nat.\ SETSPEC\ GEN\%PVAR\%2236\ (\exists (n::nat)\ m::nat.\ n < m \wedge m < k\_sy\ s \wedge POWER\ (f\_sy\ s)\ n\ x = q \wedge POWER\ (f\_sy\ s)\ m\ x = p)\ x)$

thm DEF_kl_sy:

$kl\_sy = (\lambda(\_7405259::stable\_sy)\ (\_7405260::nat)\ \_7405261::nat.\ CARD\ (IS\_SY\ \_7405259\ \_7405260\ \_7405261))$

thm Mtuwlun.kl_sy:

$\forall (s::stable\_sy)\ (p::nat)\ q::nat.\ kl\_sy\ s\ p\ q = CARD\ (IS\_SY\ s\ p\ q)$

thm DEF_fl_sy:

$fl\_sy = (\lambda(\_7405280::stable\_sy)\ (\_7405281::nat)\ (\_7405282::nat)\ \_7405283::nat.\ if\ \_7405283 \neq \_7405282\ then\ f\_sy\ \_7405280\ \_7405283\ else\ \_7405281)$

thm Mtuwlun.fl_sy:

$\forall (q::nat)\ (s::stable\_sy)\ (i::nat)\ p::nat.\ fl\_sy\ s\ p\ q\ i = (if\ i \neq q\ then\ f\_sy\ s\ i\ else\ p)$

thm DEF_COVER1_SY:

$COVER1\_SY = (\lambda(\_7405312::nat)\ (\_7405313::nat)\ (\_7405314::stable\_sy)\ (\_7405315::stable\_sy)\ \_7405316::stable\_sy.\ I\_SY\ \_7405315 = IS\_SY\ \_7405314\ \_7405312\ \_7405313 \wedge I\_SY\ \_7405316 = IS\_SY\ \_7405314\ \_7405313\ \_7405312 \wedge f\_sy\ \_7405315 = fl\_sy\ \_7405314\ \_7405312\ \_7405313 \wedge f\_sy\ \_7405316 = fl\_sy\ \_7405314\ \_7405313\ \_7405312)$

thm Mtuwlun.COVER1_SY:

$\forall (s1::stable\_sy)\ (s2::stable\_sy)\ (s::stable\_sy)\ (q::nat)\ p::nat.\ COVER1\_SY\ p\ q\ s\ s1\ s2 = (I\_SY\ s1 = IS\_SY\ s\ p\ q \wedge I\_SY\ s2 = IS\_SY\ s\ q\ p \wedge f\_sy\ s1 = fl\_sy\ s\ p\ q \wedge f\_sy\ s2 = fl\_sy\ s\ q\ p)$

thm DEF_COVER2_SY:

$COVER2\_SY = (\lambda(\_7405357::stable\_sy)\ (\_7405358::stable\_sy)\ \_7405359::stable\_sy.\ d\_sy\ \_7405357 \leq d\_sy\ \_7405358 + d\_sy\ \_7405359)$

thm Mtuwlun.COVER2_SY:

$\forall (s::stable\_sy)\ (s1::stable\_sy)\ s2::stable\_sy.\ COVER2\_SY\ s\ s1\ s2 = (d\_sy\ s \leq d\_sy\ s1 + d\_sy\ s2)$

thm DEF_COVER3_SY:

$COVER3\_SY = (\lambda(\_7405378::nat)\ (\_7405379::nat)\ (\_7405380::stable\_sy)\ (\_7405381::stable\_sy)\ \_7405382::stable\_sy.\ SUBSET\ (J\_SY\ \_7405381)\ (HOL\_Light\_Import.UNION\ (J\_SY\ \_7405380)\ (INSERT\ (INSERT\ \_7405378\ (INSERT\ \_7405379\ EMPTY))\ EMPTY)) \wedge SUBSET\ (J\_SY\ \_7405382)\ (HOL\_Light\_Import.UNION\ (J\_SY\ \_7405380)\ (INSERT\ (INSERT\ \_7405378\ (INSERT\ \_7405379\ EMPTY))\ EMPTY)))$

thm Mtuwlun.COVER3_SY:

$\forall (s1::stable\_sy)\ (s2::stable\_sy)\ (s::stable\_sy)\ (p::nat)\ q::nat.\ COVER3\_SY\ p\ q\ s\ s1\ s2 = (SUBSET\ (J\_SY\ s1)\ (HOL\_Light\_Import.UNION\ (J\_SY\ s)\ (INSERT\ (INSERT\ p\ (INSERT\ q\ EMPTY))\ EMPTY)) \wedge SUBSET\ (J\_SY\ s2)\ (HOL\_Light\_Import.UNION\ (J\_SY\ s)\ (INSERT\ (INSERT\ p\ (INSERT\ q\ EMPTY))\ EMPTY)))$

thm DEF_COVER4_SY:

$COVER4\_SY = (\lambda(\_7405423::nat)\ (\_7405424::nat)\ (\_7405425::stable\_sy)\ (\_7405426::stable\_sy)\ \_7405427::stable\_sy.\ (\forall (i::nat)\ j::nat.\ INSERT\ i\ (INSERT\ j\ EMPTY) \neq INSERT\ \_7405423\ (INSERT\ \_7405424\ EMPTY) \wedge IN\ i\ (I\_SY\ \_7405426) \wedge IN\ j\ (I\_SY\ \_7405426) \longrightarrow a\_sy\ \_7405426\ (i,\ j) = a\_sy\ \_7405425\ (i,\ j) \wedge b\_sy\ \_7405426\ (i,\ j) = b\_sy\ \_7405425\ (i,\ j)) \wedge (\forall (i::nat)\ j::nat.\ INSERT\ i\ (INSERT\ j\ EMPTY) \neq INSERT\ \_7405423\ (INSERT\ \_7405424\ EMPTY) \wedge IN\ i\ (I\_SY\ \_7405427) \wedge IN\ j\ (I\_SY\ \_7405427) \longrightarrow a\_sy\ \_7405427\ (i,\ j) = a\_sy\ \_7405425\ (i,\ j) \wedge b\_sy\ \_7405427\ (i,\ j) = b\_sy\ \_7405425\ (i,\ j)))$

thm Mtuwlun.COVER4_SY:

$\forall (s1::stable\_sy)\ (p::nat)\ (q::nat)\ (s2::stable\_sy)\ s::stable\_sy.\ COVER4\_SY\ p\ q\ s\ s1\ s2 = ((\forall (i::nat)\ j::nat.\ INSERT\ i\ (INSERT\ j\ EMPTY) \neq INSERT\ p\ (INSERT\ q\ EMPTY) \wedge IN\ i\ (I\_SY\ s1) \wedge IN\ j\ (I\_SY\ s1) \longrightarrow a\_sy\ s1\ (i,\ j) = a\_sy\ s\ (i,\ j) \wedge b\_sy\ s1\ (i,\ j) = b\_sy\ s\ (i,\ j)) \wedge (\forall (i::nat)\ j::nat.\ INSERT\ i\ (INSERT\ j\ EMPTY) \neq INSERT\ p\ (INSERT\ q\ EMPTY) \wedge IN\ i\ (I\_SY\ s2) \wedge IN\ j\ (I\_SY\ s2) \longrightarrow a\_sy\ s2\ (i,\ j) = a\_sy\ s\ (i,\ j) \wedge b\_sy\ s2\ (i,\ j) = b\_sy\ s\ (i,\ j)))$

thm DEF_COVER5_SY:

$COVER5\_SY = (\lambda(\_7405468::nat)\ (\_7405469::nat)\ (\_7405470::stable\_sy)\ (\_7405471::stable\_sy)\ \_7405472::stable\_sy.\ a\_sy\ \_7405471\ (\_7405468,\ \_7405469) \leq a\_sy\ \_7405470\ (\_7405468,\ \_7405469) \wedge a\_sy\ \_7405472\ (\_7405468,\ \_7405469) \leq a\_sy\ \_7405470\ (\_7405468,\ \_7405469) \wedge a\_sy\ \_7405470\ (\_7405468,\ \_7405469) \leq b\_sy\ \_7405471\ (\_7405468,\ \_7405469) \wedge a\_sy\ \_7405470\ (\_7405468,\ \_7405469) \leq b\_sy\ \_7405472\ (\_7405468,\ \_7405469))$

thm Mtuwlun.COVER5_SY:

$\forall (s1::stable\_sy)\ (s::stable\_sy)\ (s2::stable\_sy)\ (p::nat)\ q::nat.\ COVER5\_SY\ p\ q\ s\ s1\ s2 = (a\_sy\ s1\ (p,\ q) \leq a\_sy\ s\ (p,\ q) \wedge a\_sy\ s2\ (p,\ q) \leq a\_sy\ s\ (p,\ q) \wedge a\_sy\ s\ (p,\ q) \leq b\_sy\ s1\ (p,\ q) \wedge a\_sy\ s\ (p,\ q) \leq b\_sy\ s2\ (p,\ q))$

thm DEF_COVER6_SY:

$COVER6\_SY = (\lambda(\_7405513::nat)\ (\_7405514::nat)\ (\_7405515::stable\_sy)\ (\_7405516::stable\_sy)\ \_7405517::stable\_sy.\ IN\ (INSERT\ \_7405513\ (INSERT\ \_7405514\ EMPTY))\ (J\_SY\ \_7405516) = IN\ (INSERT\ \_7405513\ (INSERT\ \_7405514\ EMPTY))\ (J\_SY\ \_7405517) \wedge IN\ (INSERT\ \_7405513\ (INSERT\ \_7405514\ EMPTY))\ (J\_SY\ \_7405516) = (ear\_sy\ \_7405516 \vee ear\_sy\ \_7405517))$

thm Mtuwlun.COVER6_SY:

$\forall (s::stable\_sy)\ (p::nat)\ (q::nat)\ (s1::stable\_sy)\ s2::stable\_sy.\ COVER6\_SY\ p\ q\ s\ s1\ s2 = (IN\ (INSERT\ p\ (INSERT\ q\ EMPTY))\ (J\_SY\ s1) = IN\ (INSERT$

p (*INSERT q EMPTY*)) (*J_SY s2*) ∧ *IN* (*INSERT p* (*INSERT q EMPTY*)) (*J_SY s1*) = (*ear_sy s1* ∨ *ear_sy s2*))

thm DEF_COVER_SY:

COVER_SY = (λ(*_7405558*::*nat*) (*_7405559*::*nat*) (*_7405560*::*stable_sy*) (*_7405561*::*stable_sy*) *_7405562*::*stable_sy*. *COVER1_SY _7405558 _7405559 _7405560 _7405561 _7405562* ∧ *COVER2_SY _7405560 _7405561 _7405562* ∧ *COVER3_SY _7405558 _7405559 _7405560 _7405561 _7405562* ∧ *COVER4_SY _7405558 _7405559 _7405560 _7405561 _7405562* ∧ *COVER5_SY _7405558 _7405559 _7405560 _7405561 _7405562* ∧ *COVER6_SY _7405558 _7405559 _7405560 _7405561 _7405562*)

thm Mtuwlun.COVER_SY:

∀ (*p*::*nat*) (*q*::*nat*) (*s*::*stable_sy*) (*s1*::*stable_sy*) *s2*::*stable_sy*. *COVER_SY p q s s1 s2* = (*COVER1_SY p q s s1 s2* ∧ *COVER2_SY s s1 s2* ∧ *COVER3_SY p q s s1 s2* ∧ *COVER4_SY p q s s1 s2* ∧ *COVER5_SY p q s s1 s2* ∧ *COVER6_SY p q s s1 s2*)

thm DEF_pmat1:

pmat1 = (λ*_7405603*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if i* < *dimindex HOL_Light_Import.UNIV then* $ ($ *_7405603 i*) *j else* $ ($ *_7405603* (*dimindex HOL_Light_Import.UNIV*)) *j*)))

thm Mtuwlun.pmat1:

∀ *A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *pmat1 A* = *lambda* (λ*i*::*nat*. *lambda* (λ*j*::*nat*. *if i* < *dimindex HOL_Light_Import.UNIV then* $ ($ *A i*) *j else* $ ($ *A* (*dimindex HOL_Light_Import.UNIV*)) *j*))

thm DEF_pmat2:

pmat2 = (λ*_7405608*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *lambda* (λ*i*::*nat*. *lambda* ($ ($ *_7405608* (*dimindex HOL_Light_Import.UNIV* − *dimindex HOL_Light_Import.UNIV* + *i*)))))

thm Mtuwlun.pmat2:

∀ *A*::((*real*, *?'c*::*type*) *cart*, *?'b*::*type*) *cart*. *pmat2 A* = *lambda* (λ*i*::*nat*. *lambda* ($ ($ *A* (*dimindex HOL_Light_Import.UNIV* − *dimindex HOL_Light_Import.UNIV* + *i*))))

thm DEF_DIA_SY:

DIA_SY = (λ(*_7405613*::*nat*) (*_7405614*::*nat*) *_7405615*::*stable_sy*. *_7405613* ≠ *_7405614* ∧ *IN _7405613* (*I_SY _7405615*) ∧ *IN _7405614* (*I_SY _7405615*) ∧ ¬ (∃ *i*::*nat*. *IN i* (*I_SY _7405615*) ∧ *INSERT _7405613* (*INSERT _7405614 EMPTY*) = *INSERT i* (*INSERT* (*f_sy _7405615 i*) *EMPTY*)))

thm Mtuwlun.DIA_SY:

∀ (*p*::*nat*) (*q*::*nat*) *s*::*stable_sy*. *DIA_SY p q s* = (*p* ≠ *q* ∧ *IN p* (*I_SY s*) ∧ *IN q* (*I_SY s*) ∧ ¬ (∃ *i*::*nat*. *IN i* (*I_SY s*) ∧ *INSERT p* (*INSERT q EMPTY*) = *INSERT i* (*INSERT* (*f_sy s i*) *EMPTY*)))

thm Mtuwlun.CARD_I_SY_LT_3:

$DIA\_SY\ (?p::nat)\ (?q::nat)\ (?s::stable\_sy) \longrightarrow (3::nat) < CARD\ (I\_SY\ ?s)$

thm Mtuwlun.COVER_NOT_EAR_SY:

$\forall s::stable\_sy.\ DIA\_SY\ (?p::nat)\ (?q::nat)\ s \longrightarrow \neg\ ear\_sy\ s$

thm Mtuwlun.DIAGONAL_SY:

$DIA\_SY\ (?p::nat)\ (?q::nat)\ (?s::stable\_sy) \longrightarrow \neg\ IN\ (INSERT\ ?p\ (INSERT\ ?q\ EMPTY))\ (J\_SY\ ?s)$

thm DEF_SCHANGE:

$SCHANGE = (\lambda(\_7406620::nat \Rightarrow nat)\ (\_7406621::stable\_sy)\ \_7406622::stable\_sy.\ k\_sy\ \_7406621 = k\_sy\ \_7406622 \wedge d\_sy\ \_7406621 = d\_sy\ \_7406622 \wedge I\_SY\ \_7406621 = IMAGE\ \_7406620\ (I\_SY\ \_7406622) \wedge (\forall (p::nat)\ q::nat.\ IN\ (INSERT\ (\_7406620\ p)\ (INSERT\ (\_7406620\ q)\ EMPTY))\ (J\_SY\ \_7406621) = IN\ (INSERT\ p\ (INSERT\ q\ EMPTY))\ (J\_SY\ \_7406622)) \wedge (\forall (p::nat)\ q::nat.\ a\_sy\ \_7406621\ (\_7406620\ p,\ \_7406620\ q) = a\_sy\ \_7406622\ (p,\ q)) \wedge (\forall (p::nat)\ q::nat.\ b\_sy\ \_7406621\ (\_7406620\ p,\ \_7406620\ q) = b\_sy\ \_7406622\ (p,\ q)) \wedge f\_sy\ \_7406622 = f\_sy\ \_7406621 \circ \_7406620)$

thm Mtuwlun.SCHANGE:

$\forall (s1::stable\_sy)\ (s::stable\_sy)\ f::nat \Rightarrow nat.\ SCHANGE\ f\ s\ s1 = (k\_sy\ s = k\_sy\ s1 \wedge d\_sy\ s = d\_sy\ s1 \wedge I\_SY\ s = IMAGE\ f\ (I\_SY\ s1) \wedge (\forall (p::nat)\ q::nat.\ IN\ (INSERT\ (f\ p)\ (INSERT\ (f\ q)\ EMPTY))\ (J\_SY\ s) = IN\ (INSERT\ p\ (INSERT\ q\ EMPTY))\ (J\_SY\ s1)) \wedge (\forall (p::nat)\ q::nat.\ a\_sy\ s\ (f\ p,\ f\ q) = a\_sy\ s1\ (p,\ q)) \wedge (\forall (p::nat)\ q::nat.\ b\_sy\ s\ (f\ p,\ f\ q) = b\_sy\ s1\ (p,\ q)) \wedge f\_sy\ s1 = f\_sy\ s \circ f)$

thm Mtuwlun.IK_SY:

$(1::nat) \leq (?p::nat) \wedge I\_SY\ (?s::stable\_sy) = dotdot\ (0::nat)\ (?p - (1::nat)) \longrightarrow k\_sy\ ?s = ?p$

thm Mtuwlun.K_SY_LE2:

$(2::nat) < k\_sy\ (?s::stable\_sy)$

thm Mtuwlun.IN_J_IMP_IN_J1_SY:

$k\_sy\ (?s1.0::stable\_sy) = (?p::nat) \wedge I\_SY\ ?s1.0 = dotdot\ (0::nat)\ (?p - (1::nat)) \wedge f\_sy\ ?s1.0 = (\lambda i::nat.\ ((1::nat) + i)\ mod\ ?p) \wedge IN\ (INSERT\ (0::nat)\ (INSERT\ (?p - (1::nat))\ EMPTY))\ (J\_SY\ ?s1.0) \longrightarrow IN\ (?p - (1::nat),\ ?p)\ (J1\_SY\ ?s1.0)$

thm Mtuwlun.MTUWLUN1:

$\forall (s::stable\_sy)\ (s1::stable\_sy)\ (s2::stable\_sy)\ (s3::stable\_sy)\ (l::(real,\ (?'c::type,\ 3)\ finite\_product)\ cart)\ (l1::(real,\ (?'b::type,\ 3)\ finite\_product)\ cart)\ l2::(real,\ (?'a::type,\ 3)\ finite\_product)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV = (?k::nat)$

∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ⟶ d_fun (s, l) ≤ d_fun (s1, l1) + d_fun (s3, l2)

thm Mtuwlun.F_SY_COVER_EQ:

∀ (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ (2::nat) < ?p ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ HOL_Light_Import.UNION (DELETE (F_SY (vecmats l1)) (row (?p − (1::nat)) (vecmats l1), row ?p (vecmats l1))) (DELETE (F_SY (vecmats l2)) (row (?k − ?p + (2::nat)) (vecmats l2), row (1::nat) (vecmats l2))) = F_SY (vecmats l)

thm Mtuwlun.F_SY_COVER_INTER:

∀ (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ (2::nat) < ?p ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ (∀ (i::nat) j::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ (1::nat) ≤ j ∧ j ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = row j (vecmats l) ⟶ i = j) ⟶ HOL_Light_Import.INTER (DELETE (F_SY (vecmats l1)) (row (?p − (1::nat)) (vecmats l1), row ?p (vecmats l1))) (DELETE (F_SY (vecmats l2)) (row (?k − ?p + (2::nat)) (vecmats l2), row (1::nat) (vecmats l2))) = EMPTY

thm Mtuwlun.ROW_IN_F_SY:

∀ (l::(real, (?'b::type, 3) finite_product) cart) l1::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?p::nat) ∧ (2::nat) < ?p ∧ matvec (pmat1 (vecmats l)) = l1 ⟶ IN (row (?p − (1::nat)) (vecmats l1), row ?p (vecmats l1)) (F_SY (vecmats l1))

thm Mtuwlun.PMAT2_EQ_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ ?k − ?p + (2::nat) ⟶ row ?i (vecmats l2) = row (?p − (2::nat) + ?i) (vecmats l)

thm Mtuwlun.PMAT1_EQ_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ (1::nat) ≤ (?i::nat) ∧ ?i ≤ ?p − (1::nat) ⟶ row ?i (vecmats l1) = row ?i (vecmats l)

thm Mtuwlun.PMAT1_EQ_SY_P:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot

(0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ⟶ row ?p (vecmats l1) = row ?k (vecmats l)

thm Mtuwlun.POWER_RHO_NODE_SY:

∀ (i::nat) (u::(real, 3) cart) l::(real, (?'a::type, 3) finite_product) cart. local_fan (V_SY (vecmats l), E_SY (vecmats l), F_SY (vecmats l)) ∧ (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ∧ row i (vecmats l) = u ∧ row (1::nat) (vecmats l) = (?v::(real, 3) cart) ⟶ ITER (i − (1::nat)) (rho_node1 (F_SY (vecmats l))) ?v = u

thm Mtuwlun.RHO_NODE_K_SY:

∀ l::(real, (?'a::type, 3) finite_product) cart. local_fan (V_SY (vecmats l), E_SY (vecmats l), F_SY (vecmats l)) ∧ dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ row ?k (vecmats l) = (?u::(real, 3) cart) ∧ row (1::nat) (vecmats l) = (?v::(real, 3) cart) ⟶ rho_node1 (F_SY (vecmats l)) ?u = ?v

thm Mtuwlun.RHO_NODE_SY:

∀ (i::nat) l::(real, (?'a::type, 3) finite_product) cart. local_fan (V_SY (vecmats l), E_SY (vecmats l), F_SY (vecmats l)) ∧ (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ rho_node1 (F_SY (vecmats l)) (row i (vecmats l)) = row (Suc (i mod dimindex HOL_Light_Import.UNIV)) (vecmats l)

thm Mtuwlun.ORDER_COVER1_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY

s) (*INSERT* (*INSERT* (*0*::*nat*) (*INSERT* (*?p* − (*1*::*nat*)) *EMPTY*)) *EMPTY*)) ⟶ *vector_norm* (*vector_sub* (*row i* (*vecmats l*)) (*row j* (*vecmats l*))) ≤ *cstab*) ∧ *matvec* (*pmat1* (*vecmats l*)) = *l1* ∧ *matvec* (*pmat2* (*vecmats l*)) = *l2* ∧ *row ?k* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *row* (*?p* − (*1*::*nat*)) (*vecmats l*) = (*?w*::(*real*, *3*) *cart*) ⟶ *order* (*rho_node1* (*F_SY* (*vecmats l*))) *?v ?w* = *?p* − (*1*::*nat*)

thm Mtuwlun.SLICEV_EQ_V_SY:

∀ (*s*::*stable_sy*) (*s1*::*stable_sy*) (*s2*::*stable_sy*) (*s3*::*stable_sy*) (*l*::(*real*, (*?'c*::*type*, *3*) *finite_product*) *cart*) (*l1*::(*real*, (*?'b*::*type*, *3*) *finite_product*) *cart*) *l2*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *dimindex HOL_Light_Import.UNIV* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* − (*?p*::*nat*) + (*2*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?p* ∧ (*1*::*nat*) ≤ *?k* − *?p* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ *I_SY s1* = *dotdot* (*0*::*nat*) (*?p* − (*1*::*nat*)) ∧ *f_sy s1* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?p*) ∧ *I_SY s3* = *dotdot* (*0*::*nat*) (*?k* − *?p* + (*1*::*nat*)) ∧ *f_sy s3* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* (*?k* − *?p* + (*2*::*nat*))) ∧ *ear_sy s2* = *ear_sy s3* ∧ *DIA_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s* ∧ *SCHANGE* (λ*x*::*nat*. *if x* = (*0*::*nat*) *then 0*::*nat else* (*?p* − (*2*::*nat*) + *x*) *mod ?k*) *s2 s3* ∧ *COVER_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s s1 s2* ∧ *IN l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ∧ (∀ (*i*::*nat*) *j*::*nat*. *IN* (*INSERT* (*i mod ?k*) (*INSERT* (*j mod ?k*) *EMPTY*)) (*HOL_Light_Import.UNION* (*J_SY s*) (*INSERT* (*INSERT* (*0*::*nat*) (*INSERT* (*?p* − (*1*::*nat*)) *EMPTY*)) *EMPTY*)) ⟶ *vector_norm* (*vector_sub* (*row i* (*vecmats l*)) (*row j* (*vecmats l*))) ≤ *cstab*) ∧ *matvec* (*pmat1* (*vecmats l*)) = *l1* ∧ *matvec* (*pmat2* (*vecmats l*)) = *l2* ∧ *row ?k* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *row* (*?p* − (*1*::*nat*)) (*vecmats l*) = (*?w*::(*real*, *3*) *cart*) ⟶ *slicev* (*E_SY* (*vecmats l*)) (*F_SY* (*vecmats l*)) *?v ?w* = *V_SY* (*vecmats l1*)

thm Mtuwlun.SLICEE_EQ_E_SY:

∀ (*s*::*stable_sy*) (*s1*::*stable_sy*) (*s2*::*stable_sy*) (*s3*::*stable_sy*) (*l*::(*real*, (*?'c*::*type*, *3*) *finite_product*) *cart*) (*l1*::(*real*, (*?'b*::*type*, *3*) *finite_product*) *cart*) *l2*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *dimindex HOL_Light_Import.UNIV* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* − (*?p*::*nat*) + (*2*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?p* ∧ (*1*::*nat*) ≤ *?k* − *?p* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ *I_SY s1* = *dotdot* (*0*::*nat*) (*?p* − (*1*::*nat*)) ∧ *f_sy s1* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?p*) ∧ *I_SY s3* = *dotdot* (*0*::*nat*) (*?k* − *?p* + (*1*::*nat*)) ∧ *f_sy s3* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* (*?k* − *?p* + (*2*::*nat*))) ∧ *ear_sy s2* = *ear_sy s3* ∧ *DIA_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s* ∧ *SCHANGE* (λ*x*::*nat*. *if x* = (*0*::*nat*) *then 0*::*nat else* (*?p* − (*2*::*nat*) + *x*) *mod ?k*) *s2 s3* ∧ *COVER_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s s1 s2* ∧ *IN l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ∧ (∀ (*i*::*nat*) *j*::*nat*. *IN* (*INSERT* (*i mod ?k*) (*INSERT* (*j mod ?k*) *EMPTY*)) (*HOL_Light_Import.UNION* (*J_SY s*) (*INSERT* (*INSERT* (*0*::*nat*) (*INSERT* (*?p* − (*1*::*nat*)) *EMPTY*)) *EMPTY*)) ⟶ *vector_norm* (*vector_sub* (*row i* (*vecmats l*)) (*row j* (*vecmats l*))) ≤ *cstab*) ∧ *matvec* (*pmat1* (*vecmats l*)) = *l1* ∧ *matvec* (*pmat2* (*vecmats l*)) = *l2* ∧ *row ?k* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *row* (*?p* − (*1*::*nat*)) (*vecmats l*) =

(?w::(real, 3) cart) ⟶ slicee (E_SY (vecmats l)) (F_SY (vecmats l)) ?v ?w = E_SY (vecmats l1)

thm Mtuwlun.SLICEF_EQ_F_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ row ?k (vecmats l) = (?v::(real, 3) cart) ∧ row (?p − (1::nat)) (vecmats l) = (?w::(real, 3) cart) ⟶ slicef (E_SY (vecmats l)) (F_SY (vecmats l)) ?v ?w = F_SY (vecmats l1)

thm Mtuwlun.ORDER_COVER2_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ row ?k (vecmats l) = (?v::(real, 3) cart) ∧ row (?p − (1::nat)) (vecmats l) = (?w::(real, 3) cart) ⟶ order (rho_node1 (F_SY (vecmats l))) ?w ?v = ?k − ?p + (1::nat)

thm Mtuwlun.SLICEV_EQ_V2_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ row ?k (vecmats l) = (?v::(real, 3) cart) ∧ row (?p − (1::nat)) (vecmats l) = (?w::(real, 3) cart) ⟶ slicev (E_SY (vecmats l)) (F_SY (vecmats l)) ?w ?v = V_SY (vecmats l2)

thm Mtuwlun.SLICEE_EQ_E2_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ row ?k (vecmats l) = (?v::(real, 3) cart) ∧ row (?p − (1::nat)) (vecmats l) = (?w::(real, 3) cart) ⟶ slicee (E_SY (vecmats l)) (F_SY (vecmats l)) ?w ?v = E_SY (vecmats l2)

thm Mtuwlun.SLICEF_EQ_F2_SY:

∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimin-

dex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY)) EMPTY)) ⟶ vector_norm (vector_sub (row i (vecmats l)) (row j (vecmats l))) ≤ cstab) ∧ matvec (pmat1 (vecmats l)) = l1 ∧ matvec (pmat2 (vecmats l)) = l2 ∧ row ?k (vecmats l) = (?v::(real, 3) cart) ∧ row (?p − (1::nat)) (vecmats l) = (?w::(real, 3) cart) ⟶ slicef (E_SY (vecmats l)) (F_SY (vecmats l)) ?w ?v = F_SY (vecmats l2)

thm Mtuwlun.TAU_STAR_COVER:

(∀ (V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (FF::(real, 3) cart × (real, 3) cart ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. convex_local_fan (V, E, FF) ∧ IN v V ∧ IN w V ∧ (∀ (u::(real, 3) cart) u1::(real, 3) cart. IN u (INSERT v (INSERT w EMPTY)) ∧ IN u1 V ∧ u ≠ u1 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT u (INSERT u1 EMPTY)))) ∧ (∀ e::(real, 3) cart × (real, 3) cart. IN e FF ⟶ SUBSET (aff_gt (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT w EMPTY))) (wedge_in_fan_gt e E)) ⟶ convex_local_fan (slicev E FF v w, slicee E FF v w, slicef E FF v w) ∧ convex_local_fan (slicev E FF w v, slicee E FF w v, slicef E FF w v) ∧ tau_fun (slicev E FF v w) (slicee E FF v w) (slicef E FF v w) + tau_fun (slicev E FF w v) (slicee E FF w v) (slicef E FF w v) ≤ tau_fun V E FF ∧ sol_local E FF = sol_local (slicee E FF v w) (slicef E FF v w) + sol_local (slicee E FF w v) (slicef E FF w v) ∧ CARD (slicev E FF v w) < CARD V ∧ CARD (slicev E FF w v) < CARD V ∧ (generic V E ⟶ generic (slicev E FF v w) (slicee E FF v w) ∧ generic (slicev E FF w v) (slicee E FF w v))) ⟶ (∀ (s::stable_sy) (s1::stable_sy) (s2::stable_sy) (s3::stable_sy) (l::(real, (?'c::type, 3) finite_product) cart) (l1::(real, (?'b::type, 3) finite_product) cart) l2::(real, (?'a::type, 3) finite_product) cart. dimindex HOL_Light_Import.UNIV = (?k::nat) ∧ dimindex HOL_Light_Import.UNIV = ?k − (?p::nat) + (2::nat) ∧ dimindex HOL_Light_Import.UNIV = ?p ∧ (1::nat) ≤ ?k − ?p ∧ I_SY s = dotdot (0::nat) (?k − (1::nat)) ∧ f_sy s = (λi::nat. ((1::nat) + i) mod ?k) ∧ I_SY s1 = dotdot (0::nat) (?p − (1::nat)) ∧ f_sy s1 = (λi::nat. ((1::nat) + i) mod ?p) ∧ I_SY s3 = dotdot (0::nat) (?k − ?p + (1::nat)) ∧ f_sy s3 = (λi::nat. ((1::nat) + i) mod (?k − ?p + (2::nat))) ∧ ear_sy s2 = ear_sy s3 ∧ DIA_SY (0::nat) (?p − (1::nat)) s ∧ SCHANGE (λx::nat. if x = (0::nat) then 0::nat else (?p − (2::nat) + x) mod ?k) s2 s3 ∧ COVER_SY (0::nat) (?p − (1::nat)) s s1 s2 ∧ IN l (B_SY1 (a_sy s) (b_sy s)) ∧ (∀ (i::nat) j::nat. IN (INSERT (i mod ?k) (INSERT (j mod ?k) EMPTY)) (HOL_Light_Import.UNION (J_SY s) (INSERT (INSERT (0::nat) (INSERT (?p − (1::nat)) EMPTY))

EMPTY)) ⟶ *vector_norm* (*vector_sub* (*row i* (*vecmats l*)) (*row j* (*vecmats l*))) ≤ *cstab*) ∧ *matvec* (*pmat1* (*vecmats l*)) = *l1* ∧ *matvec* (*pmat2* (*vecmats l*)) = *l2* ∧ *row ?k* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *row* (*?p* − (*1*::*nat*)) (*vecmats l*) = (*?w*::(*real*, *3*) *cart*) ∧ (∀ (*u*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *IN u* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN u1* (*V_SY* (*vecmats l*)) ∧ *u* ≠ *u1* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT u* (*INSERT u1 EMPTY*)))) ⟶ *tau_star s1 l1* + *tau_star s3 l2* ≤ *tau_star s l*)

thm Mtuwlun.IN_B_SY_COVER:

(∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) (*FF*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *convex_local_fan* (*V*, *E*, *FF*) ∧ *IN v V* ∧ *IN w V* ∧ (∀ (*u*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *IN u* (*INSERT v* (*INSERT w EMPTY*)) ∧ *IN u1 V* ∧ *u* ≠ *u1* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT u* (*INSERT u1 EMPTY*)))) ∧ (∀ *e*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *IN e FF* ⟶ *SUBSET* (*aff_gt* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*wedge_in_fan_gt e E*)) ⟶ *convex_local_fan* (*slicev E FF v w*, *slicee E FF v w*, *slicef E FF v w*) ∧ *convex_local_fan* (*slicev E FF w v*, *slicee E FF w v*, *slicef E FF w v*) ∧ *tau_fun* (*slicev E FF v w*) (*slicee E FF v w*) (*slicef E FF v w*) + *tau_fun* (*slicev E FF w v*) (*slicee E FF w v*) (*slicef E FF w v*) ≤ *tau_fun V E FF* ∧ *sol_local E FF* = *sol_local* (*slicee E FF v w*) (*slicef E FF v w*) + *sol_local* (*slicee E FF w v*) (*slicef E FF w v*) ∧ *CARD* (*slicev E FF v w*) < *CARD V* ∧ *CARD* (*slicev E FF w v*) < *CARD V* ∧ (*generic V E* ⟶ *generic* (*slicev E FF v w*) (*slicee E FF v w*) ∧ *generic* (*slicev E FF w v*) (*slicee E FF w v*))) ⟶ (∀ (*s*::*stable_sy*) (*s1*::*stable_sy*) (*s2*::*stable_sy*) (*s3*::*stable_sy*) (*l*::(*real*, (*?'c*::*type*, *3*) *finite_product*) *cart*) (*l1*::(*real*, (*?'b*::*type*, *3*) *finite_product*) *cart*) *l2*::(*real*, (*?'a*::*type*, *3*) *finite_product*) *cart*. *dimindex HOL_Light_Import.UNIV* = (*?k*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?k* − (*?p*::*nat*) + (*2*::*nat*) ∧ *dimindex HOL_Light_Import.UNIV* = *?p* ∧ (*1*::*nat*) ≤ *?k* − *?p* ∧ *I_SY s* = *dotdot* (*0*::*nat*) (*?k* − (*1*::*nat*)) ∧ *f_sy s* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?k*) ∧ *I_SY s1* = *dotdot* (*0*::*nat*) (*?p* − (*1*::*nat*)) ∧ *f_sy s1* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod ?p*) ∧ *I_SY s3* = *dotdot* (*0*::*nat*) (*?k* − *?p* + (*1*::*nat*)) ∧ *f_sy s3* = (λ*i*::*nat*. ((*1*::*nat*) + *i*) *mod* (*?k* − *?p* + (*2*::*nat*))) ∧ *ear_sy s2* = *ear_sy s3* ∧ *DIA_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s* ∧ *SCHANGE* (λ*x*::*nat*. *if x* = (*0*::*nat*) *then 0*::*nat else* (*?p* − (*2*::*nat*) + *x*) *mod ?k*) *s2 s3* ∧ *COVER_SY* (*0*::*nat*) (*?p* − (*1*::*nat*)) *s s1 s2* ∧ *IN l* (*B_SY1* (*a_sy s*) (*b_sy s*)) ∧ (∀ (*i*::*nat*) *j*::*nat*. *IN* (*INSERT* (*i mod ?k*) (*INSERT* (*j mod ?k*) *EMPTY*)) (*HOL_Light_Import.UNION* (*J_SY s*) (*INSERT* (*INSERT* (*0*::*nat*) (*INSERT* (*?p* − (*1*::*nat*)) *EMPTY*)) *EMPTY*)) ⟶ *vector_norm* (*vector_sub* (*row i* (*vecmats l*)) (*row j* (*vecmats l*))) ≤ *cstab*) ∧ *matvec* (*pmat1* (*vecmats l*)) = *l1* ∧ *matvec* (*pmat2* (*vecmats l*)) = *l2* ∧ *row ?k* (*vecmats l*) = (*?v*::(*real*, *3*) *cart*) ∧ *row* (*?p* − (*1*::*nat*)) (*vecmats l*) = (*?w*::(*real*, *3*) *cart*) ∧ (∀ (*u*::(*real*, *3*) *cart*) *u1*::(*real*, *3*) *cart*. *IN u* (*INSERT ?v* (*INSERT ?w EMPTY*)) ∧ *IN u1* (*V_SY* (*vecmats l*)) ∧ *u* ≠ *u1* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT u* (*INSERT u1 EMPTY*)))) ∧ *vector_norm* (*vector_sub ?v ?w*) ≤ *min* (*b_sy s1* (*?p* − (*1*::*nat*), *0*::*nat*)) (*b_sy s2* (*?p* − (*1*::*nat*), *0*::*nat*)) ⟶ *IN l1* (*B_SY1* (*a_sy s1*) (*b_sy s1*)))

thm Mtuwlun.IN_B_SY2_COVER:

$(\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (FF::(real, 3)\ cart \times (real, 3)\ cart \Rightarrow bool)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ convex\_local\_fan\ (V, E, FF) \wedge IN\ v\ V \wedge IN\ w\ V \wedge (\forall (u::(real, 3)\ cart)\ u1::(real, 3)\ cart.\ IN\ u\ (INSERT\ v\ (INSERT\ w\ EMPTY)) \wedge IN\ u1\ V \wedge u \neq u1 \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ u\ (INSERT\ u1\ EMPTY)))) \wedge (\forall e::(real, 3)\ cart \times (real, 3)\ cart.\ IN\ e\ FF \longrightarrow SUBSET\ (aff\_gt\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))\ (wedge\_in\_fan\_gt\ e\ E)) \longrightarrow convex\_local\_fan\ (slicev\ E\ FF\ v\ w, slicee\ E\ FF\ v\ w, slicef\ E\ FF\ v\ w) \wedge convex\_local\_fan\ (slicev\ E\ FF\ w\ v, slicee\ E\ FF\ w\ v, slicef\ E\ FF\ w\ v) \wedge tau\_fun\ (slicev\ E\ FF\ v\ w)\ (slicee\ E\ FF\ v\ w)\ (slicef\ E\ FF\ v\ w) + tau\_fun\ (slicev\ E\ FF\ w\ v)\ (slicee\ E\ FF\ w\ v)\ (slicef\ E\ FF\ w\ v) \leq tau\_fun\ V\ E\ FF \wedge sol\_local\ E\ FF = sol\_local\ (slicee\ E\ FF\ v\ w)\ (slicef\ E\ FF\ v\ w) + sol\_local\ (slicee\ E\ FF\ w\ v)\ (slicef\ E\ FF\ w\ v) \wedge CARD\ (slicev\ E\ FF\ v\ w) < CARD\ V \wedge CARD\ (slicev\ E\ FF\ w\ v) < CARD\ V \wedge (generic\ V\ E \longrightarrow generic\ (slicev\ E\ FF\ v\ w)\ (slicee\ E\ FF\ v\ w) \wedge generic\ (slicev\ E\ FF\ w\ v)\ (slicee\ E\ FF\ w\ v))) \longrightarrow (\forall (s::stable\_sy)\ (s1::stable\_sy)\ (s2::stable\_sy)\ (s3::stable\_sy)\ (l::(real, (?'c::type, 3)\ finite\_product)\ cart)\ (l1::(real, (?'b::type, 3)\ finite\_product)\ cart)\ l2::(real, (?'a::type, 3)\ finite\_product)\ cart.\ dimindex\ HOL\_Light\_Import.UNIV = (?k::nat) \wedge dimindex\ HOL\_Light\_Import.UNIV = ?k - (?p::nat) + (2::nat) \wedge dimindex\ HOL\_Light\_Import.UNIV = ?p \wedge (1::nat) \leq ?k - ?p \wedge I\_SY\ s = dotdot\ (0::nat)\ (?k - (1::nat)) \wedge f\_sy\ s = (\lambda i::nat.\ ((1::nat) + i)\ mod\ ?k) \wedge I\_SY\ s1 = dotdot\ (0::nat)\ (?p - (1::nat)) \wedge f\_sy\ s1 = (\lambda i::nat.\ ((1::nat) + i)\ mod\ ?p) \wedge I\_SY\ s3 = dotdot\ (0::nat)\ (?k - ?p + (1::nat)) \wedge f\_sy\ s3 = (\lambda i::nat.\ ((1::nat) + i)\ mod\ (?k - ?p + (2::nat))) \wedge ear\_sy\ s2 = ear\_sy\ s3 \wedge DIA\_SY\ (0::nat)\ (?p - (1::nat))\ s \wedge SCHANGE\ (\lambda x::nat.\ if\ x = (0::nat)\ then\ 0::nat\ else\ (?p - (2::nat) + x)\ mod\ ?k)\ s2\ s3 \wedge COVER\_SY\ (0::nat)\ (?p - (1::nat))\ s\ s1\ s2 \wedge IN\ l\ (B\_SY1\ (a\_sy\ s)\ (b\_sy\ s)) \wedge (\forall (i::nat)\ j::nat.\ IN\ (INSERT\ (i\ mod\ ?k)\ (INSERT\ (j\ mod\ ?k)\ EMPTY))\ (HOL\_Light\_Import.UNION\ (J\_SY\ s)\ (INSERT\ (INSERT\ (0::nat)\ (INSERT\ (?p - (1::nat))\ EMPTY))\ EMPTY)) \longrightarrow vector\_norm\ (vector\_sub\ (row\ i\ (vecmats\ l))\ (row\ j\ (vecmats\ l))) \leq cstab) \wedge matvec\ (pmat1\ (vecmats\ l)) = l1 \wedge matvec\ (pmat2\ (vecmats\ l)) = l2 \wedge row\ ?k\ (vecmats\ l) = (?v::(real, 3)\ cart) \wedge row\ (?p - (1::nat))\ (vecmats\ l) = (?w::(real, 3)\ cart) \wedge (\forall (u::(real, 3)\ cart)\ u1::(real, 3)\ cart.\ IN\ u\ (INSERT\ ?v\ (INSERT\ ?w\ EMPTY)) \wedge IN\ u1\ (V\_SY\ (vecmats\ l)) \wedge u \neq u1 \longrightarrow \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ u\ (INSERT\ u1\ EMPTY)))) \wedge vector\_norm\ (vector\_sub\ ?v\ ?w) \leq min\ (b\_sy\ s1\ (?p - (1::nat), 0::nat))\ (b\_sy\ s2\ (?p - (1::nat), 0::nat)) \longrightarrow IN\ l2\ (B\_SY1\ (a\_sy\ s3)\ (b\_sy\ s3)))$

thm DEF_tri_stable:

$tri\_stable = (\lambda(\_7470343::nat)\ (\_7470344::?'b::type)\ (\_7470345::?'a::type \Rightarrow bool)\ (\_7470346::?'a::type \times ?'a::type \Rightarrow real)\ (\_7470347::?'a::type \times ?'a::type \Rightarrow real)\ (\_7470348::(?'a::type \Rightarrow bool) \Rightarrow bool)\ \_7470349::?'a::type \Rightarrow ?'a::type.\ constraint\_system\ \_7470343\ \_7470344\ \_7470345\ \_7470346\ \_7470347\ \_7470348\ \_7470349 \wedge \_7470343 = (3::nat) \wedge (\forall (i::?'a::type)\ j::?'a::type.\ IN\ i\ \_7470345$

$\wedge$ *IN j _7470345* $\wedge$ *i* $\neq$ *j* $\longrightarrow$ *real_of_nat (2::nat)* $\leq$ *_7470346 (i, j)* $\wedge$ *_7470346 (i, j)* $\leq$ *cstab)* $\wedge$ ($\forall$ *i::?'a::type. IN i _7470345* $\longrightarrow$ *_7470346 (i, i)* = *(0::real)* $\wedge$ *_7470347 (i, _7470349 i)* < *real_of_nat (4::nat))* $\wedge$ ($\forall$ *(i::?'a::type) j::?'a::type. IN (INSERT i (INSERT j EMPTY)) _7470348* $\longrightarrow$ *_7470346 (i, j)* = *sqrt (real_of_nat (8::nat))* $\wedge$ *_7470347 (i, j)* = *cstab))*

thm Pcrttid.tri_stable:

$\forall$ *(d::?'b::type) (k::nat) (s::?'a::type* $\Rightarrow$ *bool) (f::?'a::type* $\Rightarrow$ *?'a::type) (J::(?'a::type* $\Rightarrow$ *bool)* $\Rightarrow$ *bool) (a::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *real) b::?'a::type* $\times$ *?'a::type* $\Rightarrow$ *real. tri_stable k d s a b J f* = *(constraint_system k d s a b J f* $\wedge$ *k* = *(3::nat)* $\wedge$ ($\forall$ *(i::?'a::type) j::?'a::type. IN i s* $\wedge$ *IN j s* $\wedge$ *i* $\neq$ *j* $\longrightarrow$ *real_of_nat (2::nat)* $\leq$ *a (i, j)* $\wedge$ *a (i, j)* $\leq$ *cstab)* $\wedge$ ($\forall$ *i::?'a::type. IN i s* $\longrightarrow$ *a (i, i)* = *(0::real)* $\wedge$ *b (i, f i)* < *real_of_nat (4::nat))* $\wedge$ ($\forall$ *(i::?'a::type) j::?'a::type. IN (INSERT i (INSERT j EMPTY)) J* $\longrightarrow$ *a (i, j)* = *sqrt (real_of_nat (8::nat))* $\wedge$ *b (i, j)* = *cstab))*

thm Pcrttid.EAR_TRI_STABLE_SYSTEM:

tri_stable (3::nat) (DECIMAL (11::nat) (100::nat)) (dotdot (0::nat) (2::nat)) (a_ear0 (INSERT (INSERT (1::nat) (INSERT (2::nat) EMPTY)) EMPTY)) (b_ear0 (INSERT (INSERT (1::nat) (INSERT (2::nat) EMPTY)) EMPTY)) (INSERT (INSERT (1::nat) (INSERT (2::nat) EMPTY)) EMPTY) (λ*i::nat. ((1::nat) + i) mod (3::nat))*

thm Pcrttid.exist_tri_stable:

$\exists$ *s::nat* $\times$ *real* $\times$ *(nat* $\Rightarrow$ *bool)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *((nat* $\Rightarrow$ *bool)* $\Rightarrow$ *bool)* $\times$ *(nat* $\Rightarrow$ *nat). tri_stable (fst s) (fst (snd s)) (fst (snd (snd s))) (fst (snd (snd (snd s)))) (fst (snd (snd (snd (snd s))))) (fst (snd (snd (snd (snd (snd s)))))) (snd (snd (snd (snd (snd (snd s))))))*

thm TYDEF_tri_sy:

tri_sy (tuple_tri_sy (?a::tri_sy)) = *?a* $\wedge$ *tri_stable (fst (?r::nat* $\times$ *real* $\times$ *(nat* $\Rightarrow$ *bool)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *((nat* $\Rightarrow$ *bool)* $\Rightarrow$ *bool)* $\times$ *(nat* $\Rightarrow$ *nat))) (fst (snd ?r)) (fst (snd (snd ?r))) (fst (snd (snd (snd ?r)))) (fst (snd (snd (snd (snd ?r))))) (fst (snd (snd (snd (snd (snd ?r)))))) (snd (snd (snd (snd (snd (snd ?r))))))* = *(tuple_tri_sy (tri_sy ?r)* = *?r)*

thm Pcrttid.tri_sy_tybij_conjunct1:

$\forall$ *r::nat* $\times$ *real* $\times$ *(nat* $\Rightarrow$ *bool)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *(nat* $\times$ *nat* $\Rightarrow$ *real)* $\times$ *((nat* $\Rightarrow$ *bool)* $\Rightarrow$ *bool)* $\times$ *(nat* $\Rightarrow$ *nat). tri_stable (fst r) (fst (snd r)) (fst (snd (snd r))) (fst (snd (snd (snd r)))) (fst (snd (snd (snd (snd r))))) (fst (snd (snd (snd (snd (snd r)))))) (snd (snd (snd (snd (snd (snd r))))))* = *(tuple_tri_sy (tri_sy r)* = *r)*

thm Pcrttid.tri_sy_tybij_conjunct0:

$\forall$ *a::tri_sy. tri_sy (tuple_tri_sy a)* = *a*

thm Pcrttid.tri_sy_tybij:

$(\forall\, a{::}tri\_sy.\ tri\_sy\ (tuple\_tri\_sy\ a) = a) \wedge (\forall\, r{::}nat \times real \times (nat \Rightarrow bool) \times (nat \times nat \Rightarrow real) \times (nat \times nat \Rightarrow real) \times ((nat \Rightarrow bool) \Rightarrow bool) \times (nat \Rightarrow nat).\ tri\_stable\ (fst\ r)\ (fst\ (snd\ r))\ (fst\ (snd\ (snd\ r)))\ (fst\ (snd\ (snd\ (snd\ r))))\ (fst\ (snd\ (snd\ (snd\ (snd\ r)))))\ (fst\ (snd\ (snd\ (snd\ (snd\ (snd\ r))))))\ (snd\ (snd\ (snd\ (snd\ (snd\ (snd\ r)))))) = (tuple\_tri\_sy\ (tri\_sy\ r) = r))$

thm DEF_k_ts:

$k\_ts = (\lambda\_7471398{::}tri\_sy.\ fst\ (tuple\_tri\_sy\ \_7471398))$

thm Pcrttid.k_ts:

$\forall\, s{::}tri\_sy.\ k\_ts\ s = fst\ (tuple\_tri\_sy\ s)$

thm DEF_d_ts:

$d\_ts = (\lambda\_7471403{::}tri\_sy.\ fst\ (snd\ (tuple\_tri\_sy\ \_7471403)))$

thm Pcrttid.d_ts:

$\forall\, s{::}tri\_sy.\ d\_ts\ s = fst\ (snd\ (tuple\_tri\_sy\ s))$

thm DEF_I_TS:

$I\_TS = (\lambda\_7471408{::}tri\_sy.\ fst\ (snd\ (snd\ (tuple\_tri\_sy\ \_7471408))))$

thm Pcrttid.I_TS:

$\forall\, s{::}tri\_sy.\ I\_TS\ s = fst\ (snd\ (snd\ (tuple\_tri\_sy\ s)))$

thm DEF_a_ts:

$a\_ts = (\lambda\_7471413{::}tri\_sy.\ fst\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ \_7471413)))))$

thm Pcrttid.a_ts:

$\forall\, s{::}tri\_sy.\ a\_ts\ s = fst\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ s))))$

thm DEF_b_ts:

$b\_ts = (\lambda\_7471418{::}tri\_sy.\ fst\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ \_7471418))))))$

thm Pcrttid.b_ts:

$\forall\, s{::}tri\_sy.\ b\_ts\ s = fst\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ s)))))$

thm DEF_J_TS:

$J\_TS = (\lambda\_7471423{::}tri\_sy.\ fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ \_7471423)))))))$

thm Pcrttid.J_TS:

$\forall\, s{::}tri\_sy.\ J\_TS\ s = fst\ (snd\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ s))))))$

thm DEF_f_ts:

$f\_ts = (\lambda\_7471428{::}tri\_sy.\ snd\ (snd\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ \_7471428)))))))$

thm Pcrttid.f_ts:

$\forall s::tri\_sy.\ f\_ts\ s = snd\ (snd\ (snd\ (snd\ (snd\ (snd\ (tuple\_tri\_sy\ s))))))$

thm Pcrttid.tri_sy_lemma:

$\forall s::tri\_sy.\ tri\_stable\ (k\_ts\ s)\ (d\_ts\ s)\ (I\_TS\ s)\ (a\_ts\ s)\ (b\_ts\ s)\ (J\_TS\ s)\ (f\_ts\ s)$

thm DEF_augmented_constraint_system1:

$augmented\_constraint\_system1 = (\lambda(\_7471433::stable\_sy)\ (\_7471434::?'b::type)\ (\_7471435::?'a::type)\ (\_7471436::nat \times nat \Rightarrow real)\ (\_7471437::nat \times nat \Rightarrow real)\ \_7471438::nat.\ d\_sy\ \_7471433 \leq DECIMAL\ (9::nat)\ (10::nat) \wedge \_7471438 = CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2239::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%2239\ (IN\ i\ (I\_SY\ \_7471433) \wedge (\exists j::nat.\ IN\ j\ (I\_SY\ \_7471433) \wedge (real\_of\_nat\ (2::nat) < a\_sy\ \_7471433\ (i,\ j) \vee real\_of\_nat\ (2::nat) * h0 < b\_sy\ \_7471433\ (i,\ j))))\ i))\ div\ (2::nat) \wedge \_7471438 + k\_sy\ \_7471433 \leq (6::nat) \wedge (\forall (i::nat)\ j::nat.\ IN\ i\ (I\_SY\ \_7471433) \wedge IN\ j\ (I\_SY\ \_7471433) \longrightarrow a\_sy\ \_7471433\ (i,\ j) \leq \_7471436\ (i,\ j) \wedge \_7471436\ (i,\ j) \leq \_7471437\ (i,\ j) \wedge \_7471437\ (i,\ j) \leq b\_sy\ \_7471433\ (i,\ j)))$

thm Pcrttid.augmented_constraint_system1:

$\forall (I\_lo::?'b::type)\ (I\_str::?'a::type)\ (m::nat)\ (a::nat \times nat \Rightarrow real)\ (b::nat \times nat \Rightarrow real)\ s::stable\_sy.\ augmented\_constraint\_system1\ s\ I\_lo\ I\_str\ a\ b\ m = (d\_sy\ s \leq DECIMAL\ (9::nat)\ (10::nat) \wedge m = CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2239::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%2239\ (IN\ i\ (I\_SY\ s) \wedge (\exists j::nat.\ IN\ j\ (I\_SY\ s) \wedge (real\_of\_nat\ (2::nat) < a\_sy\ s\ (i,\ j) \vee real\_of\_nat\ (2::nat) * h0 < b\_sy\ s\ (i,\ j))))\ i))\ div\ (2::nat) \wedge m + k\_sy\ s \leq (6::nat) \wedge (\forall (i::nat)\ j::nat.\ IN\ i\ (I\_SY\ s) \wedge IN\ j\ (I\_SY\ s) \longrightarrow a\_sy\ s\ (i,\ j) \leq a\ (i,\ j) \wedge a\ (i,\ j) \leq b\ (i,\ j) \wedge b\ (i,\ j) \leq b\_sy\ s\ (i,\ j)))$

thm DEF_augmented_constraint_system3:

$augmented\_constraint\_system3 = (\lambda(\_7471493::tri\_sy)\ (\_7471494::?'b::type)\ (\_7471495::?'a::type)\ (\_7471496::nat \times nat \Rightarrow real)\ (\_7471497::nat \times nat \Rightarrow real)\ \_7471498::nat.\ d\_ts\ \_7471493 \leq DECIMAL\ (9::nat)\ (10::nat) \wedge \_7471498 = CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2240::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%2240\ (IN\ i\ (I\_TS\ \_7471493) \wedge (\exists j::nat.\ IN\ j\ (I\_TS\ \_7471493) \wedge (real\_of\_nat\ (2::nat) < a\_ts\ \_7471493\ (i,\ j) \vee real\_of\_nat\ (2::nat) * h0 < b\_ts\ \_7471493\ (i,\ j))))\ i))\ div\ (2::nat) \wedge \_7471498 + k\_ts\ \_7471493 \leq (6::nat) \wedge (\forall (i::nat)\ j::nat.\ IN\ i\ (I\_TS\ \_7471493) \wedge IN\ j\ (I\_TS\ \_7471493) \longrightarrow a\_ts\ \_7471493\ (i,\ j) \leq \_7471496\ (i,\ j) \wedge \_7471496\ (i,\ j) \leq \_7471497\ (i,\ j) \wedge \_7471497\ (i,\ j) \leq b\_ts\ \_7471493\ (i,\ j)))$

thm Pcrttid.augmented_constraint_system2:

$\forall (I\_lo::?'b::type)\ (I\_str::?'a::type)\ (m::nat)\ (a::nat \times nat \Rightarrow real)\ (b::nat \times nat \Rightarrow real)\ s::tri\_sy.\ augmented\_constraint\_system3\ s\ I\_lo\ I\_str\ a\ b\ m = (d\_ts\ s \leq DECIMAL\ (9::nat)\ (10::nat) \wedge m = CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2240::nat.\ \exists i::nat.\ SETSPEC\ GEN\%PVAR\%2240\ (IN\ i\ (I\_TS\ s) \wedge (\exists j::nat.\ IN\ j\ (I\_TS\ s) \wedge (real\_of\_nat\ (2::nat) < a\_ts\ s\ (i,\ j) \vee real\_of\_nat\ (2::nat) * h0 < b\_ts$

s (i, j)))) i)) div (2::nat) ∧ m + k_ts s ≤ (6::nat) ∧ (∀ (i::nat) j::nat. IN i (I_TS s) ∧ IN j (I_TS s) ⟶ a_ts s (i, j) ≤ a (i, j) ∧ a (i, j) ≤ b (i, j) ∧ b (i, j) ≤ b_ts s (i, j)))

thm Pcrttid.CLOSED_TRI_SY:

tri_stable (?k::nat) (?d::?'b::type) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < ?k ⟶ HOL_Light_Import.closed (GSPEC (λGEN%PVAR%2241::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2241 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v) (matvec v)))

thm Pcrttid.BOUNDED_TRI_SY:

tri_stable (?k::nat) (?d::?'b::type) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < ?k ⟶ bounded (GSPEC (λGEN%PVAR%2243::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2243 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v) (matvec v)))

thm Pcrttid.COMPACT_TRI_STABLE:

tri_stable (?k::nat) (?d::real) (dotdot (0::nat) (?k − (1::nat))) (?a::nat × nat ⇒ real) (?b::nat × nat ⇒ real) (?J::(nat ⇒ bool) ⇒ bool) (λi::nat. ((1::nat) + i) mod ?k) ∧ ?k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < ?k ⟶ compact (GSPEC (λGEN%PVAR%2244::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2244 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY ?a ?b v) (matvec v)))

thm Pcrttid.PCRTTID_conjunct1:

∀ (d::real) (J::(nat ⇒ bool) ⇒ bool) (k::nat) (a::nat × nat ⇒ real) b::nat × nat ⇒ real. stable_system k d (dotdot (0::nat) (k − (1::nat))) a b J (λi::nat. ((1::nat) + i) mod k) ∧ k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < k ⟶ compact (GSPEC (λGEN%PVAR%2246::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2246 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY a b v ∧ CONDITION2_SY v) (matvec v)))

thm Pcrttid.PCRTTID_conjunct0:

∀ (d::real) (J::(nat ⇒ bool) ⇒ bool) (k::nat) (a::nat × nat ⇒ real) b::nat × nat ⇒ real. tri_stable k d (dotdot (0::nat) (k − (1::nat))) a b J (λi::nat. ((1::nat) + i) mod k) ∧ k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < k ⟶ compact (GSPEC (λGEN%PVAR%2245::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2245 ((∀ i::nat.

(1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY a b v) (matvec v)))

thm Pcrttid.PCRTTID:

(∀ (d::real) (J::(nat ⇒ bool) ⇒ bool) (k::nat) (a::nat × nat ⇒ real) b::nat × nat ⇒ real. tri_stable k d (dotdot (0::nat) (k − (1::nat))) a b J (λi::nat. ((1::nat) + i) mod k) ∧ k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < k ⟶ compact (GSPEC (λGEN%PVAR%2245::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2245 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY a b v) (matvec v)))) ∧ (∀ (d::real) (J::(nat ⇒ bool) ⇒ bool) (k::nat) (a::nat × nat ⇒ real) b::nat × nat ⇒ real. stable_system k d (dotdot (0::nat) (k − (1::nat))) a b J (λi::nat. ((1::nat) + i) mod k) ∧ k = dimindex HOL_Light_Import.UNIV ∧ (2::nat) < k ⟶ compact (GSPEC (λGEN%PVAR%2246::(real, (?'a::type, 3) finite_product) cart. ∃ v::((real, 3) cart, ?'a::type) cart. SETSPEC GEN%PVAR%2246 ((∀ i::nat. (1::nat) ≤ i ∧ i ≤ dimindex HOL_Light_Import.UNIV ⟶ IN (row i v) ball_annulus) ∧ CONDITION1_SY a b v ∧ CONDITION2_SY v) (matvec v))))

thm DEF_exceptional_face:

exceptional_face = (λ(_7472056::?'a::type hypermap) _7472057::?'a::type. (5::nat) ≤ CARD (face _7472056 _7472057))

thm Tame_defs.exceptional_face:

∀ (H::?'a::type hypermap) x::?'a::type. exceptional_face H x = ((5::nat) ≤ CARD (face H x))

thm DEF_set_of_triangles_meeting_node:

set_of_triangles_meeting_node = (λ(_7472068::?'a::type hypermap) _7472069::?'a::type. GSPEC (λGEN%PVAR%2247::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2247 (IN y (dart _7472068) ∧ CARD (face _7472068 y) = (3::nat) ∧ IN y (node _7472068 _7472069)) (face _7472068 y)))

thm Tame_defs.set_of_triangles_meeting_node:

∀ (x::?'a::type) H::?'a::type hypermap. set_of_triangles_meeting_node H x = GSPEC (λGEN%PVAR%2247::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2247 (IN y (dart H) ∧ CARD (face H y) = (3::nat) ∧ IN y (node H x)) (face H y))

thm DEF_set_of_quadrilaterals_meeting_node:

set_of_quadrilaterals_meeting_node = (λ(_7472080::?'a::type hypermap) _7472081::?'a::type. GSPEC (λGEN%PVAR%2248::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2248 (IN y (dart _7472080) ∧ CARD (face _7472080 y) = (4::nat) ∧ IN y (node _7472080 _7472081)) (face _7472080 y)))

thm Tame_defs.set_of_quadrilaterals_meeting_node:

∀ (x::?'a::type) H::?'a::type hypermap. set_of_quadrilaterals_meeting_node H x = GSPEC (λGEN%PVAR%2248::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC

GEN%PVAR%2248 (IN y (dart H) ∧ CARD (face H y) = (4::nat) ∧ IN y (node H x)) (face H y))

thm DEF_set_of_exceptional_meeting_node:

set_of_exceptional_meeting_node = (λ(_7472092::?'a::type hypermap) _7472093::?'a::type. GSPEC (λGEN%PVAR%2249::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2249 (IN y (dart _7472092) ∧ (5::nat) ≤ CARD (face _7472092 y) ∧ IN y (node _7472092 _7472093)) (face _7472092 y)))

thm Tame_defs.set_of_exceptional_meeting_node:

∀ (x::?'a::type) H::?'a::type hypermap. set_of_exceptional_meeting_node H x = GSPEC (λGEN%PVAR%2249::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2249 (IN y (dart H) ∧ (5::nat) ≤ CARD (face H y) ∧ IN y (node H x)) (face H y))

thm DEF_set_of_face_meeting_node:

set_of_face_meeting_node = (λ(_7472104::?'a::type hypermap) _7472105::?'a::type. GSPEC (λGEN%PVAR%2250::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2250 (IN y (dart _7472104) ∧ IN y (node _7472104 _7472105)) (face _7472104 y)))

thm Tame_defs.set_of_face_meeting_node:

∀ (x::?'a::type) H::?'a::type hypermap. set_of_face_meeting_node H x = GSPEC (λGEN%PVAR%2250::?'a::type ⇒ bool. ∃ y::?'a::type. SETSPEC GEN%PVAR%2250 (IN y (dart H) ∧ IN y (node H x)) (face H y))

thm DEF_type_of_node:

type_of_node = (λ(_7472116::?'a::type hypermap) _7472117::?'a::type. (CARD (set_of_triangles_meeting_node _7472116 _7472117), CARD (set_of_quadrilaterals_meeting_node _7472116 _7472117), CARD (set_of_exceptional_meeting_node _7472116 _7472117)))

thm Tame_defs.type_of_node:

∀ (H::?'a::type hypermap) x::?'a::type. type_of_node H x = (CARD (set_of_triangles_meeting_node H x), CARD (set_of_quadrilaterals_meeting_node H x), CARD (set_of_exceptional_meeting_node H x))

thm DEF_node_type_exceptional_face:

node_type_exceptional_face = (λ(_7472128::?'a::type hypermap) _7472129::?'a::type. exceptional_face _7472128 _7472129 ∧ CARD (node _7472128 _7472129) = (6::nat) ⟶ type_of_node _7472128 _7472129 = (5::nat, 0::nat, 1::nat))

thm Tame_defs.node_type_exceptional_face:

∀ (H::?'a::type hypermap) x::?'a::type. node_type_exceptional_face H x = (exceptional_face H x ∧ CARD (node H x) = (6::nat) ⟶ type_of_node H x = (5::nat, 0::nat, 1::nat))

thm DEF_node_exceptional_face:

node_exceptional_face = (λ(*_7472140*::*?'a*::*type hypermap*) *_7472141*::*?'a*::*type*. *exceptional_face _7472140 _7472141* ⟶ *CARD* (*node _7472140 _7472141*) ≤ (*6*::*nat*))

thm Tame_defs.node_exceptional_face:

∀ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *node_exceptional_face H x* = (*exceptional_face H x* ⟶ *CARD* (*node H x*) ≤ (*6*::*nat*))

thm Dont_repeat_yourself.squander:

tgt = *DECIMAL* (*1541*::*nat*) (*1000*::*nat*)

thm DEF_b_tame:

b_tame = (λ(*_7472152*::*nat*) *_7472153*::*nat*. *if* (*_7472152*, *_7472153*) = (*0*::*nat*, *3*::*nat*) *then DECIMAL* (*618*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*0*::*nat*, *4*::*nat*) *then DECIMAL* (*97*::*nat*) (*100*::*nat*) *else if* (*_7472152*, *_7472153*) = (*1*::*nat*, *2*::*nat*) *then DECIMAL* (*656*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*1*::*nat*, *3*::*nat*) *then DECIMAL* (*618*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*2*::*nat*, *1*::*nat*) *then DECIMAL* (*797*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*2*::*nat*, *2*::*nat*) *then DECIMAL* (*412*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*2*::*nat*, *3*::*nat*) *then DECIMAL* (*12851*::*nat*) (*10000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*3*::*nat*, *1*::*nat*) *then DECIMAL* (*311*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*3*::*nat*, *2*::*nat*) *then DECIMAL* (*817*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*4*::*nat*, *0*::*nat*) *then DECIMAL* (*347*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*4*::*nat*, *1*::*nat*) *then DECIMAL* (*366*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*5*::*nat*, *0*::*nat*) *then DECIMAL* (*4*::*nat*) (*100*::*nat*) *else if* (*_7472152*, *_7472153*) = (*5*::*nat*, *1*::*nat*) *then DECIMAL* (*1136*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*6*::*nat*, *0*::*nat*) *then DECIMAL* (*686*::*nat*) (*1000*::*nat*) *else if* (*_7472152*, *_7472153*) = (*7*::*nat*, *0*::*nat*) *then DECIMAL* (*1450*::*nat*) (*1000*::*nat*) *else tgt*)

thm Tame_defs.b_tame:

∀ (*p*::*nat*) *q*::*nat*. *b_tame p q* = (*if* (*p*, *q*) = (*0*::*nat*, *3*::*nat*) *then DECIMAL* (*618*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*0*::*nat*, *4*::*nat*) *then DECIMAL* (*97*::*nat*) (*100*::*nat*) *else if* (*p*, *q*) = (*1*::*nat*, *2*::*nat*) *then DECIMAL* (*656*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*1*::*nat*, *3*::*nat*) *then DECIMAL* (*618*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*2*::*nat*, *1*::*nat*) *then DECIMAL* (*797*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*2*::*nat*, *2*::*nat*) *then DECIMAL* (*412*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*2*::*nat*, *3*::*nat*) *then DECIMAL* (*12851*::*nat*) (*10000*::*nat*) *else if* (*p*, *q*) = (*3*::*nat*, *1*::*nat*) *then DECIMAL* (*311*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*3*::*nat*, *2*::*nat*) *then DECIMAL* (*817*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*4*::*nat*, *0*::*nat*) *then DECIMAL* (*347*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*4*::*nat*, *1*::*nat*) *then DECIMAL* (*366*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*5*::*nat*, *0*::*nat*) *then DECIMAL* (*4*::*nat*) (*100*::*nat*) *else if* (*p*, *q*) = (*5*::*nat*, *1*::*nat*) *then DECIMAL* (*1136*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*6*::*nat*, *0*::*nat*) *then DECIMAL* (*686*::*nat*) (*1000*::*nat*) *else if* (*p*, *q*) = (*7*::*nat*, *0*::*nat*) *then DECIMAL* (*1450*::*nat*) (*1000*::*nat*) *else tgt*)

thm DEF_d_tame:

d_tame = (λ_7472164::nat. if _7472164 = (3::nat) then 0::real else if _7472164 = (4::nat) then DECIMAL (206::nat) (1000::nat) else if _7472164 = (5::nat) then DECIMAL (4819::nat) (10000::nat) else if _7472164 = (6::nat) then DECIMAL (712::nat) (1000::nat) else tgt)

thm Tame_defs.d_tame:

∀ n::nat. d_tame n = (if n = (3::nat) then 0::real else if n = (4::nat) then DECIMAL (206::nat) (1000::nat) else if n = (5::nat) then DECIMAL (4819::nat) (10000::nat) else if n = (6::nat) then DECIMAL (712::nat) (1000::nat) else tgt)

thm Tame_defs.a_tame:

a_tame = DECIMAL (63::nat) (100::nat)

thm DEF_total_weight:

total_weight = (λ_7472169::?'a::type hypermap. sum (face_set _7472169))

thm Tame_defs.total_weight:

∀ (H::?'a::type hypermap) w::(?'a::type ⇒ bool) ⇒ real. total_weight H w = sum (face_set H) w

thm DEF_adm_1:

adm_1 = (λ(_7472181::?'a::type hypermap) _7472182::(?'a::type ⇒ bool) ⇒ real. ∀ x::?'a::type. d_tame (CARD (face _7472181 x)) ≤ _7472182 (face _7472181 x))

thm Tame_defs.adm_1:

∀ (w::(?'a::type ⇒ bool) ⇒ real) H::?'a::type hypermap. adm_1 H w = (∀ x::?'a::type. d_tame (CARD (face H x)) ≤ w (face H x))

thm DEF_adm_2:

adm_2 = (λ(_7472193::?'a::type hypermap) _7472194::(?'a::type ⇒ bool) ⇒ real. ∀ x::?'a::type. CARD (set_of_exceptional_meeting_node _7472193 x) = (0::nat) ⟶ b_tame (CARD (set_of_triangles_meeting_node _7472193 x)) (CARD (set_of_quadrilaterals_meeting_node _7472193 x)) ≤ sum (set_of_face_meeting_node _7472193 x) _7472194)

thm Tame_defs.adm_2:

∀ (w::(?'a::type ⇒ bool) ⇒ real) H::?'a::type hypermap. adm_2 H w = (∀ x::?'a::type. CARD (set_of_exceptional_meeting_node H x) = (0::nat) ⟶ b_tame (CARD (set_of_triangles_meeting_node H x)) (CARD (set_of_quadrilaterals_meeting_node H x)) ≤ sum (set_of_face_meeting_node H x) w)

thm DEF_adm_3:

adm_3 = (λ(*_7472205*::*?'a*::*type hypermap*) *_7472206*::(*?'a*::*type* ⇒ *bool*) ⇒ *real*. ∀ *x*::*?'a*::*type*. *type_of_node _7472205 x* = (*5*::*nat*, *0*::*nat*, *1*::*nat*) ⟶ *a_tame* ≤ *sum* (*set_of_triangles_meeting_node _7472205 x*) *_7472206*)

thm Tame_defs.adm_3:

∀ (*H*::*?'a*::*type hypermap*) *w*::(*?'a*::*type* ⇒ *bool*) ⇒ *real*. *adm_3 H w* = (∀ *x*::*?'a*::*type*. *type_of_node H x* = (*5*::*nat*, *0*::*nat*, *1*::*nat*) ⟶ *a_tame* ≤ *sum* (*set_of_triangles_meeting_node H x*) *w*)

thm DEF_admissible_weight:

admissible_weight = (λ(*_7472217*::*?'a*::*type hypermap*) *_7472218*::(*?'a*::*type* ⇒ *bool*) ⇒ *real*. *adm_1 _7472217 _7472218* ∧ *adm_2 _7472217 _7472218* ∧ *adm_3 _7472217 _7472218*)

thm Tame_defs.admissible_weight:

∀ (*H*::*?'a*::*type hypermap*) *w*::(*?'a*::*type* ⇒ *bool*) ⇒ *real*. *admissible_weight H w* = (*adm_1 H w* ∧ *adm_2 H w* ∧ *adm_3 H w*)

thm DEF_tame_1:

tame_1 = (λ*_7472229*::*?'a*::*type hypermap*. *plain_hypermap _7472229* ∧ *planar_hypermap _7472229*)

thm Tame_defs.tame_1:

∀ *H*::*?'a*::*type hypermap*. *tame_1 H* = (*plain_hypermap H* ∧ *planar_hypermap H*)

thm DEF_tame_2:

tame_2 = (λ*_7472234*::*?'a*::*type hypermap*. *connected_hypermap _7472234* ∧ *simple_hypermap _7472234*)

thm Tame_defs.tame_2:

∀ *H*::*?'a*::*type hypermap*. *tame_2 H* = (*connected_hypermap H* ∧ *simple_hypermap H*)

thm DEF_tame_3:

tame_3 = *is_edge_nondegenerate*

thm Tame_defs.tame_3:

∀ *H*::*?'a*::*type hypermap*. *tame_3 H* = *is_edge_nondegenerate H*

thm DEF_tame_4:

tame_4 = *no_loops*

thm Tame_defs.tame_4:

∀ *H*::*?'a*::*type hypermap*. *tame_4 H* = *no_loops H*

thm DEF_tame_5a:

$tame\_5a = is\_no\_double\_joints$

thm Tame_defs.tame_5a:

$\forall H::?'a::type\ hypermap.\ tame\_5a\ H = is\_no\_double\_joints\ H$

thm DEF_tame_8:

$tame\_8 = (\lambda\_7472254::?'a::type\ hypermap.\ (3::nat) \leq number\_of\_faces\ \_7472254)$

thm Tame_defs.tame_8:

$\forall H::?'a::type\ hypermap.\ tame\_8\ H = ((3::nat) \leq number\_of\_faces\ H)$

thm DEF_tame_9a:

$tame\_9a = (\lambda\_7472259::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_7472259) \longrightarrow (3::nat) \leq CARD\ (face\ \_7472259\ x) \wedge CARD\ (face\ \_7472259\ x) \leq (6::nat))$

thm Tame_defs.tame_9a:

$\forall H::?'a::type\ hypermap.\ tame\_9a\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow (3::nat) \leq CARD\ (face\ H\ x) \wedge CARD\ (face\ H\ x) \leq (6::nat))$

thm DEF_tame_10:

$tame\_10 = (\lambda\_7472264::?'a::type\ hypermap.\ IN\ (number\_of\_nodes\ \_7472264)\ (INSERT\ (13::nat)\ (INSERT\ (14::nat)\ (INSERT\ (15::nat)\ EMPTY))))$

thm Tame_defs.tame_10:

$\forall H::?'a::type\ hypermap.\ tame\_10\ H = IN\ (number\_of\_nodes\ H)\ (INSERT\ (13::nat)\ (INSERT\ (14::nat)\ (INSERT\ (15::nat)\ EMPTY)))$

thm DEF_tame_11a:

$tame\_11a = (\lambda\_7472269::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_7472269) \longrightarrow (3::nat) \leq CARD\ (node\ \_7472269\ x))$

thm Tame_defs.tame_11a:

$\forall H::?'a::type\ hypermap.\ tame\_11a\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow (3::nat) \leq CARD\ (node\ H\ x))$

thm DEF_tame_11b:

$tame\_11b = (\lambda\_7472274::?'a::type\ hypermap.\ \forall x::?'a::type.\ IN\ x\ (dart\ \_7472274) \longrightarrow CARD\ (node\ \_7472274\ x) \leq (7::nat))$

thm Tame_defs.tame_11b:

$\forall H::?'a::type\ hypermap.\ tame\_11b\ H = (\forall x::?'a::type.\ IN\ x\ (dart\ H) \longrightarrow CARD\ (node\ H\ x) \leq (7::nat))$

thm DEF_tame_12o:

$tame\_12o = (\lambda\_7472279::?'a::type\ hypermap.\ \forall x::?'a::type.\ node\_type\_exceptional\_face\ \_7472279\ x \wedge node\_exceptional\_face\ \_7472279\ x)$

thm Tame_defs.tame_12o:

$\forall H::?'a::type\ hypermap.\ tame\_12o\ H = (\forall x::?'a::type.\ node\_type\_exceptional\_face\ H\ x \wedge node\_exceptional\_face\ H\ x)$

thm DEF_tame_13a:

$tame\_13a = (\lambda\_7472284::?'a::type\ hypermap.\ \exists w::(?'a::type \Rightarrow bool) \Rightarrow real.\ admissible\_weight\ \_7472284\ w \wedge total\_weight\ \_7472284\ w < tgt)$

thm Tame_defs.tame_13a:

$\forall H::?'a::type\ hypermap.\ tame\_13a\ H = (\exists w::(?'a::type \Rightarrow bool) \Rightarrow real.\ admissible\_weight\ H\ w \wedge total\_weight\ H\ w < tgt)$

thm DEF_tame_hypermap:

$tame\_hypermap = (\lambda\_7472289::?'a::type\ hypermap.\ tame\_1\ \_7472289 \wedge tame\_2\ \_7472289 \wedge tame\_3\ \_7472289 \wedge tame\_4\ \_7472289 \wedge tame\_5a\ \_7472289 \wedge tame\_8\ \_7472289 \wedge tame\_9a\ \_7472289 \wedge tame\_10\ \_7472289 \wedge tame\_11a\ \_7472289 \wedge tame\_11b\ \_7472289 \wedge tame\_12o\ \_7472289 \wedge tame\_13a\ \_7472289)$

thm Tame_defs.tame_hypermap:

$\forall H::?'a::type\ hypermap.\ tame\_hypermap\ H = (tame\_1\ H \wedge tame\_2\ H \wedge tame\_3\ H \wedge tame\_4\ H \wedge tame\_5a\ H \wedge tame\_8\ H \wedge tame\_9a\ H \wedge tame\_10\ H \wedge tame\_11a\ H \wedge tame\_11b\ H \wedge tame\_12o\ H \wedge tame\_13a\ H)$

thm DEF_opposite_hypermap:

$opposite\_hypermap = (\lambda\_7472294::?'a::type\ hypermap.\ hypermap\ (dart\ \_7472294,\ face\_map\ \_7472294 \circ node\_map\ \_7472294,\ HOL\_Light\_Import.inverse\ (node\_map\ \_7472294),\ HOL\_Light\_Import.inverse\ (face\_map\ \_7472294)))$

thm Tame_defs.opposite_hypermap:

$\forall H::?'a::type\ hypermap.\ opposite\_hypermap\ H = hypermap\ (dart\ H,\ face\_map\ H \circ node\_map\ H,\ HOL\_Light\_Import.inverse\ (node\_map\ H),\ HOL\_Light\_Import.inverse\ (face\_map\ H))$

thm DEF_ESTD:

$ESTD = (\lambda\_7472299::(real,\ 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2251::(real,\ 3)\ cart \Rightarrow bool.\ \exists (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2251\ (IN\ v\ \_7472299 \wedge IN\ w\ \_7472299 \wedge v \neq w \wedge distance\ (v,\ w) \leq real\_of\_nat\ (2::nat) * h0)\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm Tame_defs.ESTD:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ ESTD\ V = GSPEC\ (\lambda GEN\%PVAR\%2251::(real,\ 3)\ cart \Rightarrow bool.\ \exists (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2251\ (IN\ v\ V \wedge IN\ w\ V \wedge v \neq w \wedge distance\ (v,\ w) \leq real\_of\_nat\ (2::nat) * h0)\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm DEF_ECTC:

$ECTC = (\lambda\_7472304::(real, 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2252::(real, 3)\ cart \Rightarrow bool.\ \exists (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2252\ (IN\ v\ \_7472304 \wedge IN\ w\ \_7472304 \wedge v \neq w \wedge distance\ (v, w) = real\_of\_nat\ (2::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY))))$

thm Tame_defs.ECTC:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ ECTC\ V = GSPEC\ (\lambda GEN\%PVAR\%2252::(real, 3)\ cart \Rightarrow bool.\ \exists (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2252\ (IN\ v\ V \wedge IN\ w\ V \wedge v \neq w \wedge distance\ (v, w) = real\_of\_nat\ (2::nat))\ (INSERT\ v\ (INSERT\ w\ EMPTY)))$

thm Tame_defs.dart1_of_fan:

$\forall (V::?'a::type \Rightarrow bool)\ E::(?'a::type \Rightarrow bool) \Rightarrow bool.\ dart1\_of\_fan\ (V, E) = GSPEC\ (\lambda GEN\%PVAR\%2253::?'a::type \times ?'a::type.\ \exists (v::?'a::type)\ w::?'a::type.\ SETSPEC\ GEN\%PVAR\%2253\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)\ (v, w))$

thm Tame_defs.dart_of_fan:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ dart\_of\_fan\ (V, E) = HOL\_Light\_Import.UNION\ (GSPEC\ (\lambda GEN\%PVAR\%2254::(real, 3)\ cart \times (real, 3)\ cart.\ \exists v::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2254\ (IN\ v\ V \wedge set\_of\_edge\ v\ V\ E = EMPTY)\ (v, v)))\ (GSPEC\ (\lambda GEN\%PVAR\%2255::(real, 3)\ cart \times (real, 3)\ cart.\ \exists (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2255\ (IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ E)\ (v, w)))$

thm Tame_defs.e_fan_pair:

$\forall (V::?'d::type)\ (E::?'c::type)\ (w::?'b::type)\ v::?'a::type.\ e\_fan\_pair\ (V, E)\ (v, w) = (w, v)$

thm DEF_face_set_of_fan:

$face\_set\_of\_fan = (\lambda\_7472309::((real, 3)\ cart \Rightarrow bool) \times (((real, 3)\ cart \Rightarrow bool) \Rightarrow bool).\ face\_set\ (hypermap\_of\_fan\ (fst\ \_7472309, snd\ \_7472309)))$

thm Tame_defs.face_set_of_fan:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool.\ face\_set\_of\_fan\ (V, E) = face\_set\ (hypermap\_of\_fan\ (V, E))$

thm DEF_scriptL:

$scriptL = (\lambda\_7472318::(real, 3)\ cart \Rightarrow bool.\ sum\ \_7472318\ (\lambda v::(real, 3)\ cart.\ lmfun\ (vector\_norm\ v\ /\ real\_of\_nat\ (2::nat))))$

thm Tame_defs.scriptL:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ scriptL\ V = sum\ V\ (\lambda v::(real, 3)\ cart.\ lmfun\ (vector\_norm\ v\ /\ real\_of\_nat\ (2::nat)))$

thm DEF_contravening:

contravening = (λ*_7472323*::(*real*, *3*) *cart* ⇒ *bool*. *packing _7472323* ∧ *SUBSET _7472323 ball_annulus* ∧ *real_of_nat* (*12*::*nat*) < *scriptL _7472323* ∧ (∀ *W*::(*real*, *3*) *cart* ⇒ *bool*. *packing W* ∧ *SUBSET W ball_annulus* ⟶ *scriptL W* ≤ *scriptL _7472323*) ∧ (*CARD _7472323* = (*13*::*nat*) ∨ *CARD _7472323* = (*14*::*nat*) ∨ *CARD _7472323* = (*15*::*nat*)) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v _7472323* ⟶ *surrounded_node* (*_7472323*, *ESTD _7472323*) *v*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v _7472323* ⟶ *surrounded_node* (*_7472323*, *ECTC _7472323*) *v* ∨ *vector_norm v* = *real_of_nat* (*2*::*nat*)))

thm Tame_defs.contravening:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening V* = (*packing V* ∧ *SUBSET V ball_annulus* ∧ *real_of_nat* (*12*::*nat*) < *scriptL V* ∧ (∀ *W*::(*real*, *3*) *cart* ⇒ *bool*. *packing W* ∧ *SUBSET W ball_annulus* ⟶ *scriptL W* ≤ *scriptL V*) ∧ (*CARD V* = (*13*::*nat*) ∨ *CARD V* = (*14*::*nat*) ∨ *CARD V* = (*15*::*nat*)) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ *surrounded_node* (*V*, *ESTD V*) *v*) ∧ (∀ *v*::(*real*, *3*) *cart*. *IN v V* ⟶ *surrounded_node* (*V*, *ECTC V*) *v* ∨ *vector_norm v* = *real_of_nat* (*2*::*nat*)))

thm Tame_defs.topological_component_yfan:

∀ (*x*::(*real*, *3*) *cart*) (*V*::(*real*, *3*) *cart* ⇒ *bool*) *E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*. *topological_component_yfan* (*x*, *V*, *E*) = *GSPEC* (λ*GEN%PVAR%2256*::(*real*, *3*) *cart* ⇒ *bool*. ∃ *y*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2256* (*IN y* (*yfan* (*x*, *V*, *E*))) (*connected_component* (*yfan* (*x*, *V*, *E*)) *y*))

thm DEF_h_dart:

h_dart = (λ*_7472328*::(*real*, *3*) *cart* × *?'a*::*type*. *vector_norm* (*fst _7472328*) / *real_of_nat* (*2*::*nat*))

thm Tame_defs.h_dart:

∀ *x*::(*real*, *3*) *cart* × *?'a*::*type*. *h_dart x* = *vector_norm* (*fst x*) / *real_of_nat* (*2*::*nat*)

thm DEF_tauVEF:

tauVEF = (λ*_7472333*::((*real*, *3*) *cart* ⇒ *bool*) × (((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) × ((*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*). *sum* (*snd* (*snd _7472333*)) (λ*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_dart* (*fst _7472333*, *fst* (*snd _7472333*)) *x* ∗ ((*1*::*real*) + *sol0* / *pi* ∗ ((*1*::*real*) − *lmfun* (*h_dart x*)))) + (*pi* + *sol0*) ∗ (*real_of_nat* (*2*::*nat*) − *real_of_nat* (*CARD* (*snd* (*snd _7472333*)))))

thm Tame_defs.tauVEF:

∀ (*V*::(*real*, *3*) *cart* ⇒ *bool*) (*E*::((*real*, *3*) *cart* ⇒ *bool*) ⇒ *bool*) *f*::(*real*, *3*) *cart* × (*real*, *3*) *cart* ⇒ *bool*. *tauVEF* (*V*, *E*, *f*) = *sum f* (λ*x*::(*real*, *3*) *cart* × (*real*, *3*) *cart*. *azim_dart* (*V*, *E*) *x* ∗ ((*1*::*real*) + *sol0* / *pi* ∗ ((*1*::*real*) − *lmfun* (*h_dart x*)))) + (*pi* + *sol0*) ∗ (*real_of_nat* (*2*::*nat*) − *real_of_nat* (*CARD f*))

thm DEF_restricted_hypermap:

restricted_hypermap = (λ_7472346::?'a::type hypermap. is_no_double_joints _7472346 ∧ dart _7472346 ≠ EMPTY ∧ planar_hypermap _7472346 ∧ connected_hypermap _7472346 ∧ plain_hypermap _7472346 ∧ simple_hypermap _7472346 ∧ is_edge_nondegenerate _7472346 ∧ is_node_nondegenerate _7472346 ∧ (∀f::?'a::type ⇒ bool. IN f (face_set _7472346) ⟶ (3::nat) ≤ CARD f))

thm Tame_defs.restricted_hypermap:

∀H::?'a::type hypermap. restricted_hypermap H = (is_no_double_joints H ∧ dart H ≠ EMPTY ∧ planar_hypermap H ∧ connected_hypermap H ∧ plain_hypermap H ∧ simple_hypermap H ∧ is_edge_nondegenerate H ∧ is_node_nondegenerate H ∧ (∀f::?'a::type ⇒ bool. IN f (face_set H) ⟶ (3::nat) ≤ CARD f))

thm DEF_rho_node:

rho_node = (λ(_7472351::?'d::type × ?'c::type × (?'b::type × ?'a::type ⇒ bool)) _7472352::?'b::type. SOME w::?'a::type. IN (_7472352, w) (snd (snd _7472351)))

thm Tame_defs.rho_node:

∀(V::?'d::type) (E::?'c::type) (v::?'b::type) f::?'b::type × ?'a::type ⇒ bool. rho_node (V, E, f) v = (SOME w::?'a::type. IN (v, w) f)

thm DEF_per:

per = (λ(_7472373::?'c::type × ?'b::type × ((real, ?'a::type) cart × (real, ?'a::type) cart ⇒ bool)) (_7472374::(real, ?'a::type) cart) _7472375::nat. sum (dotdot (0::nat) (_7472375 − (1::nat))) (λi::nat. arcV (vec (0::nat)) (POWER (rho_node (fst _7472373, fst (snd _7472373), snd (snd _7472373))) i _7472374) (POWER (rho_node (fst _7472373, fst (snd _7472373), snd (snd _7472373))) (i + (1::nat)) _7472374)))

thm Tame_defs.per:

∀(k::nat) (V::?'c::type) (E::?'b::type) (f::(real, ?'a::type) cart × (real, ?'a::type) cart ⇒ bool) v::(real, ?'a::type) cart. per (V, E, f) v k = sum (dotdot (0::nat) (k − (1::nat))) (λi::nat. arcV (vec (0::nat)) (POWER (rho_node (V, E, f)) i v) (POWER (rho_node (V, E, f)) (i + (1::nat)) v))

thm DEF_perimeterbound:

perimeterbound = (λ_7472406::((real, 3) cart ⇒ bool) × (((real, 3) cart ⇒ bool) ⇒ bool). ∀f::(real, 3) cart × (real, 3) cart ⇒ bool. IN f (face_set_of_fan (fst _7472406, snd _7472406)) ⟶ sum f (GABS (λf::(real, 3) cart × (real, 3) cart ⇒ real. ∀(v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (arcV (vec (0::nat)) v w))) ≤ real_of_nat (2::nat) ∗ pi)

thm Tame_defs.perimeterbound:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. perimeterbound (V, E) = (∀f::(real, 3) cart × (real, 3) cart ⇒ bool. IN f (face_set_of_fan (V, E)) ⟶ sum f (GABS (λf::(real, 3) cart × (real, 3) cart ⇒ real. ∀(v::(real, *3) cart) w::(real, 3) cart. GEQ (f (v, w)) (arcV (vec (0::nat)) v w))) ≤ real_of_nat (2::nat) * pi)*

thm Tame_general.INEQ_ALT:

$\forall$ *(A::bool) bounds::(real × real × real) list. ineq bounds A = (list_all (GABS (λf::real × real × real ⇒ bool.* $\forall$ *(a::real) (x::real) b::real. GEQ (f (a, x, b)) (a ≤ x ∧ x ≤ b))) bounds ⟶ A)*

thm Tame_inequalities.lemma:

$\forall$ *(a::real) (b::real) (c::real) (x::real) (x0::real) x1::real. a < (0::real) ∧ x0 ≤ x ∧ x ≤ x1 ⟶ a * (x0 * x0) + (b * x0 + c) ≤ a * (x * x) + (b * x + c) ∨ a * (x1 * x1) + (b * x1 + c) ≤ a * (x * x) + (b * x + c)*

thm Tame_inequalities.lemma′:

$\forall$ *(f::real ⇒ real) (a::real) (b::real) (c::real) (x::real) (x0::real) x1::real. f = (λx::real. a * (x * x) + (b * x + c)) ∧ x0 ≤ x ∧ x ≤ x1 ∧ a < (0::real) ⟶ f x0 ≤ f x ∨ f x1 ≤ f x*

thm Tame_inequalities.lemma4:

$\forall$ *(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. DECIMAL (40::nat) (10::nat) ≤ x4 ∧ x4 ≤ DECIMAL (63504::nat) (10000::nat) ∧ DECIMAL (40::nat) (10::nat) ≤ x1 ⟶ delta_x x1 x2 x3 (DECIMAL (40::nat) (10::nat)) x5 x6 ≤ delta_x x1 x2 x3 x4 x5 x6 ∨ delta_x x1 x2 x3 (DECIMAL (63504::nat) (10000::nat)) x5 x6 ≤ delta_x x1 x2 x3 x4 x5 x6*

thm Tame_inequalities.lemma5:

$\forall$ *(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. DECIMAL (40::nat) (10::nat) ≤ x5 ∧ x5 ≤ DECIMAL (63504::nat) (10000::nat) ∧ DECIMAL (40::nat) (10::nat) ≤ x2 ⟶ delta_x x1 x2 x3 x4 (DECIMAL (40::nat) (10::nat)) x6 ≤ delta_x x1 x2 x3 x4 x5 x6 ∨ delta_x x1 x2 x3 x4 (DECIMAL (63504::nat) (10000::nat)) x6 ≤ delta_x x1 x2 x3 x4 x5 x6*

thm Tame_inequalities.main_lemma:

$\forall$ *(f::real ⇒ real) (x::real) (x0::real) (x1::real) m::real. (f x0 ≤ f x ∨ f x1 ≤ f x) ∧ m ≤ f x0 ∧ m ≤ f x1 ⟶ m ≤ f x*

thm Tame_inequalities.delta_x_pos:

$\forall$ *(x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. ineq [(DECIMAL (40::nat) (10::nat), x1, DECIMAL (63504::nat) (10000::nat)), (DECIMAL (40::nat) (10::nat), x2, DECIMAL (63504::nat) (10000::nat)), (DECIMAL (40::nat) (10::nat), x3, DECIMAL (63504::nat) (10000::nat)), (DECIMAL (40::nat) (10::nat), x4, DECIMAL (63504::nat) (10000::nat)), (DECIMAL (40::nat) (10::nat), x5, DECIMAL (63504::nat) (10000::nat)), (DECIMAL (40::nat) (10::nat), x6, DECIMAL (63504::nat) (10000::nat))] (DECIMAL (1280::nat) (10::nat) ≤ delta_x x1 x2 x3 x4 x5 x6)*

thm Tame_inequalities.lemma6:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) (*x4*::*real*) (*x5*::*real*) *x6*::*real*. *DECIMAL* (*64*::*nat*) (*100*::*nat*) ≤ *x6* ∧ *x6* ≤ *DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ∧ *DECIMAL* (*40*::*nat*) (*10*::*nat*) ≤ *x3* ⟶ *delta_x x1 x2 x3 x4 x5* (*DECIMAL* (*64*::*nat*) (*100*::*nat*)) ≤ *delta_x x1 x2 x3 x4 x5 x6* ∨ *delta_x x1 x2 x3 x4 x5* (*DECIMAL* (*63504*::*nat*) (*10000*::*nat*)) ≤ *delta_x x1 x2 x3 x4 x5 x6*

thm Tame_inequalities.delta_x_3_points:

∀ (*x1*::*real*) (*x2*::*real*) (*x3*::*real*) *x6*::*real*. *ineq* [(*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x1*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)), (*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x2*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)), (*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x3*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)), (*DECIMAL* (*64*::*nat*) (*100*::*nat*), *x6*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*))] (*DECIMAL* (*130*::*nat*) (*10*::*nat*) ≤ *delta_x x1 x2 x3* (*DECIMAL* (*40*::*nat*) (*10*::*nat*)) (*DECIMAL* (*40*::*nat*) (*10*::*nat*)) *x6*)

thm Tame_inequalities.lemma1:

∀ (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *DECIMAL* (*40*::*nat*) (*10*::*nat*) ≤ *x1* ∧ *x1* ≤ *DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ⟶ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x* (*DECIMAL* (*40*::*nat*) (*10*::*nat*)) *x2 x3* − *DECIMAL* (*40*::*nat*) (*10*::*nat*) ∗ (*x2* ∗ *x3*) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*) ∨ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x* (*DECIMAL* (*63504*::*nat*) (*10000*::*nat*)) *x2 x3* − *DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ∗ (*x2* ∗ *x3*) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*)

thm Tame_inequalities.lemma2:

∀ (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *DECIMAL* (*40*::*nat*) (*10*::*nat*) ≤ *x2* ∧ *x2* ≤ *DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ⟶ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1* (*DECIMAL* (*40*::*nat*) (*10*::*nat*)) *x3* − *x1* ∗ (*DECIMAL* (*40*::*nat*) (*10*::*nat*) ∗ *x3*) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*) ∨ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1* (*DECIMAL* (*63504*::*nat*) (*10000*::*nat*)) *x3* − *x1* ∗ (*DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ∗ *x3*) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*)

thm Tame_inequalities.lemma3:

∀ (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *DECIMAL* (*40*::*nat*) (*10*::*nat*) ≤ *x3* ∧ *x3* ≤ *DECIMAL* (*63504*::*nat*) (*10000*::*nat*) ⟶ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2* (*DECIMAL* (*40*::*nat*) (*10*::*nat*)) − *x1* ∗ (*x2* ∗ *DECIMAL* (*40*::*nat*) (*10*::*nat*)) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*) ∨ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2* (*DECIMAL* (*63504*::*nat*) (*10000*::*nat*)) − *x1* ∗ (*x2* ∗ *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)) ≤ *DECIMAL* (*22*::*nat*) (*10*::*nat*) ∗ *ups_x x1 x2 x3* − *x1* ∗ (*x2* ∗ *x3*)

thm Tame_inequalities.eta_x_ineq_lemma:

∀ (*x1*::*real*) (*x2*::*real*) *x3*::*real*. *ineq* [(*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x1*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)), (*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x2*, *DECIMAL* (*63504*::*nat*) (*10000*::*nat*)), (*DECIMAL* (*40*::*nat*) (*10*::*nat*), *x3*, *DEC-*

IMAL (63504::nat) (10000::nat))] ((0::real) ≤ DECIMAL (22::nat) (10::nat) ∗ ups_x x1 x2 x3 − x1 ∗ (x2 ∗ x3))

thm Tame_inequalities.ETA_Y_4_POINTS_INEQ:

∀ (y1::real) (y2::real) y3::real. ineq [(real_of_nat (2::nat), y1, real_of_nat (2::nat) ∗ h0), (real_of_nat (2::nat), y2, real_of_nat (2::nat) ∗ h0), (real_of_nat (2::nat), y3, real_of_nat (2::nat) ∗ h0)] ((eta_y y1 y2 y3)^2 *≤ DECIMAL (22::nat) (10::nat))*

thm Tame_inequalities.DELTA_X4_MONO_LT_4:

∀ (x1::real) (x2::real) (x3::real) (a::real) (x5::real) (x6::real) b::real. a < b ∧ (0::real) < x1 ⟶ delta_x4 x1 x2 x3 b x5 x6 < delta_x4 x1 x2 x3 a x5 x6

thm Tame_inequalities.DELTA_X4_MONO_LE_4:

∀ (x1::real) (x2::real) (x3::real) (a::real) (x5::real) (x6::real) b::real. a < b ∧ (0::real) ≤ x1 ⟶ delta_x4 x1 x2 x3 b x5 x6 ≤ delta_x4 x1 x2 x3 a x5 x6

thm Tame_inequalities.REAL_LT_SQUARE_POS:

∀ (x::real) y::real. (0::real) < x ∧ x < y ⟶ $x^2 < y^2$

thm Tame_inequalities.ATN2_ACS_LEMMA:

∀ (a::real) b::real. b ∗ b < a ⟶ pi / real_of_nat (2::nat) + atn2 (sqrt (a − b ∗ b), − b) = acs (b / sqrt a)

thm Tame_inequalities.DELTA_X_AND_DELTA_X4:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. LET (λd4::real. LET_END (LET (λd::real. LET_END (LET (λv1::real. LET_END (LET (λv2::real. LET_END (real_of_nat (4::nat) ∗ (x1 ∗ d) = v1 ∗ v2 − d4 ∗ d4)) (ups_x x1 x3 x5))) (ups_x x1 x2 x6))) (delta_x x1 x2 x3 x4 x5 x6))) (delta_x4 x1 x2 x3 x4 x5 x6)

thm Tame_inequalities.DELTA_EQ_DELTA_X:

∀ (x1::real) (x2::real) (x3::real) (x4::real) (x5::real) x6::real. delta x1 x2 x3 x6 x5 x4 = delta_x x1 x2 x3 x4 x5 x6

thm Tame_inequalities.DIH_X_MONO_LT_4:

∀ (x1::real) (x2::real) (x3::real) (a::real) (x5::real) (x6::real) b::real. a < b ∧ (0::real) < x1 ∧ (0::real) < delta_x x1 x2 x3 a x5 x6 ∧ (0::real) < delta_x x1 x2 x3 b x5 x6 ⟶ dih_x x1 x2 x3 a x5 x6 < dih_x x1 x2 x3 b x5 x6

thm Tame_inequalities.DIH_Y_INEQ:

(∀ (y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. ineq [(DECIMAL (20::nat) (10::nat), y1, real_of_nat (2::nat) ∗ h0), (DECIMAL (20::nat) (10::nat), y2, real_of_nat (2::nat) ∗ h0), (DECIMAL (20::nat) (10::nat), y3, real_of_nat (2::nat) ∗ h0), (real_of_nat (2::nat) ∗ h0, y4, real_of_nat (2::nat) ∗ h0), (DECIMAL (20::nat) (10::nat), y5, real_of_nat (2::nat) ∗ h0), (DECIMAL

(*20*::*nat*) (*10*::*nat*), *y6*, *real_of_nat* (*2*::*nat*) ∗ *h0*)] (*DECIMAL* (*115*::*nat*) (*100*::*nat*) < *dih_y y1 y2 y3 y4 y5 y6*)) ⟶ *ineq* [(*DECIMAL* (*20*::*nat*) (*10*::*nat*), *?y1.0*::*real*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *?y2.0*::*real*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *?y3.0*::*real*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*252*::*nat*) (*100*::*nat*), *?y4.0*::*real*, *DECIMAL* (*504*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *?y5.0*::*real*, *DECIMAL* (*252*::*nat*) (*100*::*nat*)), (*DECIMAL* (*20*::*nat*) (*10*::*nat*), *?y6.0*::*real*, *DECIMAL* (*252*::*nat*) (*100*::*nat*))] (*DECIMAL* (*115*::*nat*) (*100*::*nat*) < *dih_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* ∨ *delta_y ?y1.0 ?y2.0 ?y3.0 ?y4.0 ?y5.0 ?y6.0* ≤ (*0*::*real*))

thm Ckqowsa_3_points.coplanar_eq_coplanar_alt:

∀ *s*::(*real*, *?'a*::*type*) *cart* ⇒ *bool*. (*2*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ⟶ *coplanar s* = *coplanar_alt s*

thm Ckqowsa_3_points.projection_lemma:

∀ (*v*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *n* ≠ *vec* (*0*::*nat*) ⟶ (∃ (*a*::*real*) *x*::(*real*, *?'a*::*type*) *cart*. *dot v n* / *dot n n* = *a* ∧ *dot x n* = (*0*::*real*) ∧ *v* = *vector_add x* (% *a n*))

thm Ckqowsa_3_points.dist_lower_bound:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *n* ≠ *vec* (*0*::*nat*) ⟶ (*dot* (*vector_sub v1 v2*) *n*)^2 / *dot n n* ≤ (*distance* (*v1*, *v2*))^2

thm Ckqowsa_3_points.non_collinear_lemma:

∀ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v ball_annulus* ∧ *IN w ball_annulus* ∧ *real_of_nat* (*2*::*nat*) ≤ *distance* (*v*, *w*) ∧ (*0*::*real*) < *dot v w* ⟶ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w EMPTY*)))

thm Ckqowsa_3_points.aff_ge_0_2:

∀ (*u*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *u* ≠ *vec* (*0*::*nat*) ∧ *v* ≠ *vec* (*0*::*nat*) ⟶ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT u* (*INSERT v EMPTY*)) = *GSPEC* (λ*GEN%PVAR%2258*::(*real*, *?'a*::*type*) *cart*. ∃ *y*::(*real*, *?'a*::*type*) *cart*. *SETSPEC GEN%PVAR%2258* (∃ (*t1*::*real*) *t2*::*real*. (*0*::*real*) ≤ *t1* ∧ (*0*::*real*) ≤ *t2* ∧ *y* = *vector_add* (% *t1 u*) (% *t2 v*)) *y*)

thm Ckqowsa_3_points.in_aff_ge_0_2:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v1 ball_annulus* ∧ *IN v2 ball_annulus* ∧ *IN w ball_annulus* ∧ *real_of_nat* (*2*::*nat*) ≤ *distance* (*v1*, *w*) ∧ *real_of_nat* (*2*::*nat*) ≤ *distance* (*v2*, *w*) ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ⟶ (∃ (*t1*::*real*) *t2*::*real*. (*0*::*real*) < *t1* ∧ (*0*::*real*) < *t2* ∧ *w* = *vector_add* (% *t1 v1*) (% *t2 v2*))

thm Ckqowsa_3_points.in_aff_ge_0_2_imp_dot_pos:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *dot v1 v2* ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* *v1* (*INSERT v2 EMPTY*))) ∧ *w* ≠ *vec* (*0*::*nat*) ⟶ (*0*::*real*) < *dot v1 w* ∧ (*0*::*real*) < *dot v2 w*

thm Ckqowsa_3_points.aff_ge_eq_lemma:

∀ (*a*::*real*) (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. (*0*::*real*) < *a* ∧ *u* = % *a v2* ∧ *v1* ≠ *vec* (*0*::*nat*) ∧ *v2* ≠ *vec* (*0*::*nat*) ⟶ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*)) = *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT u EMPTY*))

thm Ckqowsa_3_points.triangle_height_lemma:

∀ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *w* ≠ *vec* (*0*::*nat*) ⟶ (∃ (*a*::*real*) *n*::(*real*, *3*) *cart*. *v* = *vector_add* (% *a w*) *n* ∧ *dot n w* = (*0*::*real*) ∧ *vector_norm n* = *vector_norm* (*cross v w*) / *vector_norm w*)

thm Ckqowsa_3_points.in_aff_ge_dist_lemma:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*n1*::(*real*, *?'a*::*type*) *cart*) (*n2*::(*real*, *?'a*::*type*) *cart*) (*a*::*real*) (*b*::*real*) *w*::(*real*, *?'a*::*type*) *cart*. *w* ≠ *vec* (*0*::*nat*) ∧ *v1* ≠ *vec* (*0*::*nat*) ∧ *v2* ≠ *vec* (*0*::*nat*) ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ *v1* = *vector_add* (% *a w*) *n1* ∧ *v2* = *vector_add* (% *b w*) *n2* ∧ *dot n1 w* = (*0*::*real*) ∧ *dot n2 w* = (*0*::*real*) ⟶ *vector_norm* (*vector_sub n1 n2*) = *vector_norm n1* + *vector_norm n2*

thm Ckqowsa_3_points.in_aff_ge_dist_lower_bound:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *v1* ≠ *vec* (*0*::*nat*) ∧ *v2* ≠ *vec* (*0*::*nat*) ∧ *w* ≠ *vec* (*0*::*nat*) ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ⟶ (*vector_norm* (*cross v1 w*) + *vector_norm* (*cross v2 w*)) / *vector_norm w* ≤ *distance* (*v1*, *v2*)

thm Ckqowsa_3_points.triangle_area_lower_bound:

∀ (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v ball_annulus* ∧ *IN w ball_annulus* ∧ *distance* (*v*, *w*) = *real_of_nat* (*2*::*nat*) ⟶ *DECIMAL* (*148*::*nat*) (*100*::*nat*) ∗ *sqrt* (*real_of_nat* (*3*::*nat*)) ≤ *vector_norm* (*cross v w*)

thm Ckqowsa_3_points.DIST_LOWER_BOUND_lemma:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. *IN v1 ball_annulus* ∧ *IN v2 ball_annulus* ∧ *IN w ball_annulus* ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ *distance* (*v1*, *w*) = *real_of_nat* (*2*::*nat*) ∧ *distance* (*v2*, *w*) = *real_of_nat* (*2*::*nat*) ⟶ (*1*::*real*) ≤ *distance* (*v1*, *v2*)

thm Ckqowsa_3_points.aff_ge_trans:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v3*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *IN v3* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v3 EMPTY*))) ∧ *v* ≠ *vec* (*0*::*nat*) ∧ *v1* ≠ *vec*

(0::nat) ∧ v2 ≠ vec (0::nat) ∧ v3 ≠ vec (0::nat) ⟶ IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT v2 EMPTY)))

thm Ckqowsa_3_points.rotation_dist_decrease:

∀(v::(real, ?'a::type) cart) (w::(real, ?'a::type) cart) u::(real, ?'a::type) cart. vector_norm u = vector_norm v ∧ (0::real) ≤ dot v w ∧ w ≠ vec (0::nat) ∧ IN u (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT w EMPTY))) ⟶ distance (u, w) ≤ distance (v, w)

thm Ckqowsa_3_points.pos_vector_angle_bounds:

∀(v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. (0::real) < dot v u ⟶ (0::real) ≤ vector_angle v u ∧ vector_angle v u ≤ pi / real_of_nat (2::nat)

thm Ckqowsa_3_points.rotation_lemma:

∀(v::(real, ?'a::type) cart) u::(real, ?'a::type) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT u EMPTY))) ∧ vector_norm v = vector_norm u ⟶ (∃f::(real, unit) cart ⇒ (real, ?'a::type) cart. f (lift (0::real)) = v ∧ f (lift (1::real)) = u ∧ (∀t::(real, unit) cart. ∃(a::real) b::real. f t = vector_add (% a v) (% b u)) ∧ (∀t::(real, unit) cart. vector_norm (f t) = vector_norm v) ∧ (∀t::(real, unit) cart. IN t (closed_interval [(lift (0::real), lift (1::real))]) ⟶ IN (f t) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT u EMPTY)))) ∧ continuous_on f HOL_Light_Import.UNIV)

thm Ckqowsa_3_points.dot_pos_lemma:

∀(v::(real, 3) cart) w::(real, 3) cart. IN v ball_annulus ∧ IN w ball_annulus ∧ distance (v, w) ≤ real_of_nat (2::nat) ∗ h0 ⟶ (0::real) < dot v w

thm Ckqowsa_3_points.dist_decreasing_ivt_lemma:

∀(f::(real, unit) cart ⇒ (real, ?'a::type) cart) (v::(real, ?'a::type) cart) (t1::real) d::real. (0::real) ≤ t1 ∧ continuous_on f HOL_Light_Import.UNIV ∧ distance (f (lift t1), v) ≤ d ∧ d ≤ distance (f (lift (0::real)), v) ⟶ (∃t::(real, unit) cart. IN t (closed_interval [(lift (0::real), lift t1)]) ∧ distance (f t, v) = d)

thm Ckqowsa_3_points.lemma_3_points:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v1 (INSERT v2 EMPTY))) ∧ IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ distance (v1, v2) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v3) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ⟶ (∃v::(real, 3) cart. IN v ball_annulus ∧ IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v (INSERT v2 EMPTY))) ∧ distance (v, v2) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v, v3) = real_of_nat (2::nat))

thm Ckqowsa_3_points.LEMMA_3_POINTS:

∀(v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v1 (INSERT v2 EMPTY))) ∧ IN v1 ball_annulus

∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ distance (v1, v2) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v3) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ⟶ (∃ (v1′::(real, 3) cart) v2′::(real, 3) cart. IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v1′ (INSERT v2′ EMPTY))) ∧ IN v1′ ball_annulus ∧ IN v2′ ball_annulus ∧ IN v3 ball_annulus ∧ distance (v1′, v2′) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1′, v3) = real_of_nat (2::nat) ∧ distance (v2′, v3) = real_of_nat (2::nat))

thm Ckqowsa_3_points.LEMMA_3_POINTS_FINAL:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) v3::(real, 3) cart. IN v3 (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v1 (INSERT v2 EMPTY))) ∧ IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ distance (v1, v2) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v3) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ⟶ False

thm Ckqowsa_4_points.IN_INTERVAL_1:

∀ (a::real) (b::real) c::real. IN (lift c) (closed_interval [(lift a, lift b)]) = (a ≤ c ∧ c ≤ b)

thm Ckqowsa_4_points.DIST_SQR:

∀ (v::(real, ?′a::type) cart) (w::(real, ?′a::type) cart) d::real. (distance (v, w) = d) = ((distance (v, w))^2 = d^2 ∧ (0::real) ≤ d)

thm Ckqowsa_4_points.estd_non_collinear_lemma:

∀ (v::(real, 3) cart) w::(real, 3) cart. IN v ball_annulus ∧ IN w ball_annulus ∧ real_of_nat (2::nat) ≤ distance (v, w) ∧ distance (v, w) ≤ real_of_nat (2::nat) ∗ h0 ⟶ ¬ collinear (INSERT (vec (0::nat)) (INSERT v (INSERT w EMPTY)))

thm Ckqowsa_4_points.zero_not_between:

∀ (v::(real, ?′a::type) cart) w::(real, ?′a::type) cart. ¬ between (vec (0::nat)) (v, w) ⟶ v ≠ vec (0::nat) ∧ w ≠ vec (0::nat) ∧ (∀ (a::real) b::real. a < (0::real) ∧ (0::real) < b ⟶ % a w ≠ % b v)

thm Ckqowsa_4_points.zero_not_between_estd:

∀ (v::(real, 3) cart) w::(real, 3) cart. IN v ball_annulus ∧ IN w ball_annulus ∧ distance (v, w) ≤ real_of_nat (2::nat) ∗ h0 ⟶ ¬ between (vec (0::nat)) (v, w)

thm Ckqowsa_4_points.VECTOR_PROJECTION:

∀ (d::(real, ?′a::type) cart) v::(real, ?′a::type) cart. v = vector_add (projection d v) (% (dot v d / dot d d) d)

thm Ckqowsa_4_points.PROJECTION_0:

∀ d::(real, ?′a::type) cart. projection d (vec (0::nat)) = vec (0::nat)

thm Ckqowsa_4_points.PROJECTION_ZERO:

$\forall\, d$::(*real*, *?'a*::*type*) *cart*. $d \neq$ *vec* (*0*::*nat*) $\longrightarrow$ *projection d d* = *vec* (*0*::*nat*)

thm Ckqowsa_4_points.PROJECTION_LINEAR:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *a*::*real*. *projection d* (*vector_add v w*) = *vector_add* (*projection d v*) (*projection d w*) $\wedge$ *projection d* (% *a v*) = % *a* (*projection d v*)

thm Ckqowsa_4_points.PROJECTION_NEG:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. *projection d* (*vector_neg v*) = *vector_neg* (*projection d v*)

thm Ckqowsa_4_points.PROJECTION_SUB:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *projection d* (*vector_sub v w*) = *vector_sub* (*projection d v*) (*projection d w*)

thm Ckqowsa_4_points.PROJECTION_DIST2:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. $d \neq$ *vec* (*0*::*nat*) $\longrightarrow$ (*distance* (*projection d v*, *projection d w*))^2 = (*distance* (*v*, *w*))^2 $-$ (*dot* (*vector_sub v w*) *d*)^2 / (*vector_norm d*)^2

thm Ckqowsa_4_points.PROJECTION_SUM_DIST:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*y*::(*real*, *?'a*::*type*) *cart*) (*a*::*real*) *b*::*real*. *dot x d* = (*0*::*real*) $\wedge$ *dot y d* = (*0*::*real*) $\longrightarrow$ (*distance* (*vector_add x* (% *a d*), *vector_add y* (% *b d*)))^2 = (*distance* (*x*, *y*))^2 + $(a - b)^2 *$ *dot d d*

thm Ckqowsa_4_points.PROJECTION_DIST_SPECIAL_EQ:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. $d \neq$ *vec* (*0*::*nat*) $\wedge$ *dot x d* = (*0*::*real*) $\longrightarrow$ (*distance* (*vector_add x* (*vector_sub v* (*projection d v*)), *w*))^2 = (*distance* (*x*, *projection d w*))^2 $-$ (*distance* (*projection d v*, *projection d w*))^2 + (*distance* (*v*, *w*))^2

thm Ckqowsa_4_points.PROJECTION_DIST_SPECIAL_LE:

$\forall$ (*d*::(*real*, *?'a*::*type*) *cart*) (*x*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. $d \neq$ *vec* (*0*::*nat*) $\wedge$ *dot x d* = (*0*::*real*) $\wedge$ *distance* (*x*, *projection d w*) $\leq$ *distance* (*projection d v*, *projection d w*) $\longrightarrow$ *distance* (*vector_add x* (*vector_sub v* (*projection d v*)), *w*) $\leq$ *distance* (*v*, *w*)

thm Ckqowsa_4_points.aff_ge_0_2_eq_segment:

$\forall$ *v*::(*real*, *?'a*::*type*) *cart*. *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v EMPTY*)) = *closed_segment* [(*vec* (*0*::*nat*), *v*)] $\wedge$ *aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*)) = *closed_segment* [(*vec* (*0*::*nat*), *v*)]

thm Ckqowsa_4_points.in_segment_imp_in_aff_ge_0_2:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. *IN v* (*closed_segment* [(*v1*, *v2*)]) $\longrightarrow$ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Ckqowsa_4_points.points_in_aff_ge_0_2:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) *v2*::(*real*, *?'a*::*type*) *cart*. *IN* (*vec* (*0*::*nat*)) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *IN v1* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *IN v2* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Ckqowsa_4_points.aff_ge_0_2_SUBSET:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. *v1* $\neq$ *vec* (*0*::*nat*) $\wedge$ *v2* $\neq$ *vec* (*0*::*nat*) $\wedge$ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) $\longrightarrow$ *SUBSET* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Ckqowsa_4_points.segment_inter_aff_ge_ends:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *v1* $\neq$ *vec* (*0*::*nat*) $\wedge$ *v2* $\neq$ *vec* (*0*::*nat*) $\wedge$ *dot v1 n* = (*0*::*real*) $\wedge$ *dot v2 n* = (*0*::*real*) $\wedge$ *dot v n* = (*0*::*real*) $\wedge$ *dot w n* $\neq$ (*0*::*real*) $\wedge$ *HOL_Light_Import.INTER* (*closed_segment* [(*v*, *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) $\neq$ *EMPTY* $\longrightarrow$ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*)))

thm Ckqowsa_4_points.in_affine_hull_lemma:

$\forall$ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) *v*::(*real*, *?'a*::*type*) *cart*. $\neg$ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *IN v* (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*)))) $\longrightarrow$ ($\exists$ (*t1*::*real*) *t2*::*real*. *v* = *vector_add* (% *t1 v1*) (% *t2 v2*) $\wedge$ ($\forall$ (*t1'*::*real*) *t2'*::*real*. *v* = *vector_add* (% *t1' v1*) (% *t2' v2*) $\longrightarrow$ *t1'* = *t1* $\wedge$ *t2'* = *t2*))

thm Ckqowsa_4_points.affine_hull_3_plane:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) *n*::(*real*, *3*) *cart*. $\neg$ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *n* $\neq$ *vec* (*0*::*nat*) $\wedge$ *dot v1 n* = (*0*::*real*) $\wedge$ *dot v2 n* = (*0*::*real*) $\longrightarrow$ *hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) = *GSPEC* (λ*GEN%PVAR%2259*::(*real*, *3*) *cart*. $\exists$ *v*::(*real*, *3*) *cart*. *SETSPEC GEN%PVAR%2259* (*dot v n* = (*0*::*real*)) *v*)

thm Ckqowsa_4_points.intersection_point_exists:

$\forall$ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*n*::(*real*, *3*) *cart*) (*v*::(*real*, *3*) *cart*) *w*::(*real*, *3*) *cart*. $\neg$ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) $\wedge$ *n* $\neq$ *vec* (*0*::*nat*) $\wedge$ *dot v1 n* = (*0*::*real*) $\wedge$ *dot v2 n* = (*0*::*real*) $\wedge$ *dot v n* $\leq$ (*0*::*real*) $\wedge$ (*0*::*real*) $\leq$ *dot w n* $\longrightarrow$ ($\exists$ *p*::(*real*, *3*) *cart*. *IN p*

(*HOL_Light_Import.INTER* (*closed_segment* [(*v*, *w*)]) (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))))))

thm Ckqowsa_4_points.aux_ineq:

$\forall$ (*a*::*real*) (*b*::*real*) *x*::*real*. (*0*::*real*) < *a* $\wedge$ (*0*::*real*) < *b* $\wedge$ *x* = ((*1*::*real*) − *a* ∗ *a* − *b* ∗ *b*) / (*real_of_nat* (*2*::*nat*) ∗ (*a* ∗ *b*)) $\wedge$ − (*1*::*real*) ≤ *x* $\wedge$ *x* ≤ (*1*::*real*) ⟶ *x* ∗ ((*1*::*real*) − *a*) ≤ *b*

thm Ckqowsa_4_points.rotation_dist_decrease_lemma:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *u*::(*real*, *?'a*::*type*) *cart*. *IN u* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) $\wedge$ *vector_norm u* = *vector_norm v* ⟶ *distance* (*u*, *w*) ≤ *distance* (*v*, *w*)

thm Ckqowsa_4_points.rotation_dist_decrease_special_case:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *a*::*real*. (*0*::*real*) ≤ *a* $\wedge$ *w* = % (− *a*) *v* $\wedge$ *vector_norm u* = *vector_norm v* ⟶ *distance* (*u*, *w*) ≤ *distance* (*v*, *w*)

thm Ckqowsa_4_points.continuous_lemma_inc:

$\forall$ (*f*::*real* ⇒ *real*) (*c*::*real*) *t1*::*real*. (*0*::*real*) ≤ *t1* $\wedge$ *real_continuous_on f* (*closed_real_interval* [(*0*::*real*, *t1*)]) $\wedge$ *f* (*0*::*real*) ≤ *c* $\wedge$ *c* ≤ *f t1* ⟶ ($\exists$ *x*≥*0*::*real*. *x* ≤ *t1* $\wedge$ *f x* = *c* $\wedge$ ($\forall$ *t*::*real*. (*0*::*real*) ≤ *t* $\wedge$ *t* < *x* ⟶ *f t* < *c*))

thm Ckqowsa_4_points.continuous_lemma_dec:

$\forall$ (*f*::*real* ⇒ *real*) (*c*::*real*) *t1*::*real*. (*0*::*real*) ≤ *t1* $\wedge$ *real_continuous_on f* (*closed_real_interval* [(*0*::*real*, *t1*)]) $\wedge$ *c* ≤ *f* (*0*::*real*) $\wedge$ *f t1* ≤ *c* ⟶ ($\exists$ *x*≥*0*::*real*. *x* ≤ *t1* $\wedge$ *f x* = *c* $\wedge$ ($\forall$ *t*::*real*. (*0*::*real*) ≤ *t* $\wedge$ *t* < *x* ⟶ *c* < *f t*))

thm Ckqowsa_4_points.rotation_lemma_special:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) *n*::(*real*, *?'a*::*type*) *cart*. *v* ≠ *vec* (*0*::*nat*) $\wedge$ *w* ≠ *vec* (*0*::*nat*) $\wedge$ *dot v n* = (*0*::*real*) $\wedge$ *dot w n* = (*0*::*real*) $\wedge$ (*3*::*nat*) ≤ *dimindex HOL_Light_Import.UNIV* ⟶ ($\exists$ *f*::(*real*, *unit*) *cart* ⇒ (*real*, *?'a*::*type*) *cart*. *continuous_on f HOL_Light_Import.UNIV* $\wedge$ *f* (*lift* (*0*::*real*)) = *v* $\wedge$ *f* (*lift* (*1*::*real*)) = % (*vector_norm v* / *vector_norm w*) *w* $\wedge$ ($\forall$ *t*::(*real*, *unit*) *cart*. *vector_norm* (*f t*) = *vector_norm v* $\wedge$ *dot* (*f t*) *n* = (*0*::*real*)) $\wedge$ ($\forall$ *t*::*real*. (*0*::*real*) ≤ *t* $\wedge$ *t* ≤ (*1*::*real*) ⟶ *distance* (*f* (*lift t*), *w*) ≤ *distance* (*v*, *w*)) $\wedge$ (¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT w EMPTY*))) ⟶ ($\forall$ *t*::*real*. (*0*::*real*) ≤ *t* $\wedge$ *t* ≤ (*1*::*real*) ⟶ *IN* (*f* (*lift t*)) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))))))

thm Ckqowsa_4_points.aff_ge_inter_segments:

$\forall$ (*v*::(*real*, *?'a*::*type*) *cart*) (*w*::(*real*, *?'a*::*type*) *cart*) (*u*::(*real*, *?'a*::*type*) *cart*) *p*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v* (*INSERT u EMPTY*))) $\wedge$ *IN p* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT u EMPTY*))) $\wedge$ *HOL_Light_Import.INTER* (*closed_segment* [(*v*, *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT u EMPTY*)) ≠ *EMPTY*

⟶ HOL_Light_Import.INTER (closed_segment [(p, w)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT u EMPTY)) ≠ EMPTY

thm Ckqowsa_4_points.rotation_lemma_segments:

∀ (v::(real, ?'a::type) cart) (w::(real, ?'a::type) cart) (u::(real, ?'a::type) cart) n::(real, ?'a::type) cart. (3::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ v ≠ vec (0::nat) ∧ u ≠ vec (0::nat) ∧ dot v n = (0::real) ∧ dot w n = (0::real) ∧ dot u n = (0::real) ∧ HOL_Light_Import.INTER (closed_segment [(vec (0::nat), u)]) (closed_segment [(v, w)]) ≠ EMPTY ⟶ (∃ f::(real, unit) cart ⇒ (real, ?'a::type) cart. continuous_on f HOL_Light_Import.UNIV ∧ f (lift (0::real)) = v ∧ f (lift (1::real)) = % (vector_norm v / vector_norm u) u ∧ (∀ t::(real, unit) cart. vector_norm (f t) = vector_norm v ∧ dot (f t) n = (0::real)) ∧ (∀ t::real. (0::real) ≤ t ∧ t ≤ (1::real) ⟶ distance (f (lift t), u) ≤ distance (v, u) ∧ distance (f (lift t), w) ≤ distance (v, w)) ∧ (∀ t::real. (0::real) ≤ t ∧ t ≤ (1::real) ⟶ HOL_Light_Import.INTER (closed_segment [(f (lift t), w)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT u EMPTY)) ≠ EMPTY))

thm Ckqowsa_4_points.rotation_about_axis:

∀ (d::(real, ?'a::type) cart) (v::(real, ?'a::type) cart) w::(real, ?'a::type) cart. (3::nat) ≤ dimindex HOL_Light_Import.UNIV ∧ d ≠ vec (0::nat) ∧ projection d w ≠ vec (0::nat) ∧ projection d v ≠ vec (0::nat) ⟶ (∃ (f::(real, unit) cart ⇒ (real, ?'a::type) cart) (a::real) b::real. (0::real) < a ∧ continuous_on f HOL_Light_Import.UNIV ∧ f (lift (0::real)) = v ∧ f (lift (1::real)) = vector_add (% a w) (% b d) ∧ (∀ (t::(real, unit) cart) c::real. distance (f t, % c d) = distance (v, % c d)) ∧ (∀ t::real. (0::real) ≤ t ∧ t ≤ (1::real) ⟶ distance (f (lift t), w) ≤ distance (v, w)))

thm Ckqowsa_4_points.continuous_solution_aux:

∀ (a11::real) (a12::real) (a21::real) (a22::real) (f::real ⇒ real) (g::real ⇒ real) (t1::real) t2::real. real_continuous_on f (closed_real_interval [(t1, t2)]) ∧ real_continuous_on g (closed_real_interval [(t1, t2)]) ∧ a11 * a22 − a12 * a21 ≠ (0::real) ⟶ (∃ (x1::real ⇒ real) x2::real ⇒ real. real_continuous_on x1 (closed_real_interval [(t1, t2)]) ∧ real_continuous_on x2 (closed_real_interval [(t1, t2)]) ∧ (∀ t::real. a11 * x1 t + a12 * x2 t = f t ∧ a21 * x1 t + a22 * x2 t = g t ∧ (∀ (y1::real) y2::real. a11 * y1 + a12 * y2 = f t ∧ a21 * y1 + a22 * y2 = g t ⟶ y1 = x1 t ∧ y2 = x2 t)))

thm Ckqowsa_4_points.continuous_intersection_point:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (n::(real, 3) cart) (f::(real, unit) cart ⇒ (real, 3) cart) (w::(real, 3) cart) t1::real. ¬ collinear (INSERT (vec (0::nat)) (INSERT v1 (INSERT v2 EMPTY))) ∧ continuous_on f (closed_interval [(lift (0::real), lift t1)]) ∧ (0::real) ≤ t1 ∧ dot v1 n = (0::real) ∧ dot v2 n = (0::real) ∧ (0::real) < dot w n ∧ (∀ t::real. (0::real) ≤ t ∧ t ≤ t1 ⟶ dot (f (lift t)) n ≤ (0::real)) ⟶ (∃ (a::real ⇒ real) b::real ⇒ real. real_continuous_on a (closed_real_interval [(0::real, t1)]) ∧ real_continuous_on b (closed_real_interval [(0::real, t1)]) ∧ (∀ t::real. (0::real) ≤ t ∧ t ≤ t1 ⟶ HOL_Light_Import.INTER (*closed_segment* [(*f* (*lift t*), *w*)]) (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT* *v1* (*INSERT v2 EMPTY*)))) = *INSERT* (*vector_add* (% (*a t*) *v1*) (% (*b t*) *v2*)) *EMPTY*))

thm Ckqowsa_4_points.in_aff_ge_cases_lemma:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *between* (*vec* (*0*::*nat*)) (*v*, *w*) ∧ *IN v* (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*)))) ∧ *IN w* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ⟶ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∨ *IN v1* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) ∨ *IN v2* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))

thm Ckqowsa_4_points.segment_intersects_aff_ge_lemma:

∀ (*v1*::(*real*, *?'a*::*type*) *cart*) (*v2*::(*real*, *?'a*::*type*) *cart*) (*v*::(*real*, *?'a*::*type*) *cart*) *w*::(*real*, *?'a*::*type*) *cart*. ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) ∧ ¬ *between* (*vec* (*0*::*nat*)) (*v*, *w*) ∧ *IN v* (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*)))) ∧ *IN w* (*hull affine* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*)))) ∧ *HOL_Light_Import.INTER* (*closed_segment* [(*v*, *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ≠ *EMPTY* ⟶ *IN v* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∨ *IN v1* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*))) ∨ *IN v2* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*)))

thm Ckqowsa_4_points.continuous_lemma_aff_ge:

∀ (*v1*::(*real*, *3*) *cart*) (*v2*::(*real*, *3*) *cart*) (*f*::(*real*, *unit*) *cart* ⇒ (*real*, *3*) *cart*) (*w*::(*real*, *3*) *cart*) *h*::*real*. (*0*::*real*) ≤ *h* ∧ *continuous_on f* (*closed_interval* [(*lift* (*0*::*real*), *lift h*)]) ∧ *HOL_Light_Import.INTER* (*closed_segment* [(*f* (*lift* (*0*::*real*)), *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ≠ *EMPTY* ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*closed_interval* [(*lift* (*0*::*real*), *lift h*)]) ⟶ ¬ *between* (*vec* (*0*::*nat*)) (*f t*, *w*)) ∧ ¬ *collinear* (*INSERT* (*vec* (*0*::*nat*)) (*INSERT v1* (*INSERT v2 EMPTY*))) ⟶ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*closed_interval* [(*lift* (*0*::*real*), *lift h*)]) ⟶ *HOL_Light_Import.INTER* (*closed_segment* [(*f t*, *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ≠ *EMPTY*) ∨ (∃ *x*≥*0*::*real*. *x* ≤ *h* ∧ (∀ *t*::(*real*, *unit*) *cart*. *IN t* (*closed_interval* [(*lift* (*0*::*real*), *lift x*)]) ⟶ *HOL_Light_Import.INTER* (*closed_segment* [(*f t*, *w*)]) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ≠ *EMPTY*) ∧ (*IN* (*f* (*lift x*)) (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT v1* (*INSERT v2 EMPTY*))) ∨ *IN v1* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*f* (*lift x*)) (*INSERT w EMPTY*))) ∨ *IN v2* (*aff_ge* (*INSERT* (*vec* (*0*::*nat*)) *EMPTY*) (*INSERT* (*f* (*lift x*)) (*INSERT w EMPTY*)))))

thm Ckqowsa_4_points.separation_plane_4_points:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ⟶ (∃ n::(real, 3) cart. n ≠ vec (0::nat) ∧ dot v2 n = (0::real) ∧ dot v4 n = (0::real) ∧ dot v1 n < (0::real) ∧ (0::real) < dot v3 n)

thm Ckqowsa_4_points.lemma_4_points_rotation1:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ⟶ (∃ v1′::(real, 3) cart. IN v1′ ball_annulus ∧ vector_norm v1′ = vector_norm v1 ∧ distance (v1′, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1′, v2) = distance (v1, v2) ∧ distance (v1′, v4) = real_of_nat (2::nat) ∧ HOL_Light_Import.INTER (closed_segment [(v1′, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY)

thm Ckqowsa_4_points.lemma_4_points_rotation1_full:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ⟶ (∃ (v1′::(real, 3) cart) v3′::(real, 3) cart. IN v1′ ball_annulus ∧ IN v3′ ball_annulus ∧ vector_norm v1′ = vector_norm v1 ∧ vector_norm v3′ = vector_norm v3 ∧ distance (v1′, v3′) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1′, v2) = real_of_nat (2::nat) ∧ distance (v1′, v4) = real_of_nat (2::nat) ∧ distance (v2, v3′) = real_of_nat (2::nat) ∧ distance (v3′, v4) = real_of_nat (2::nat) ∧ HOL_Light_Import.INTER (closed_segment [(v1′, v3′)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY)

thm Ckqowsa_4_points.segment_inter_conv:

∀ (v1::(real, ?′a::type) cart) (v2::(real, ?′a::type) cart) (v3::(real, ?′a::type) cart) (v4::(real, ?′a::type) cart) n::(real, ?′a::type) cart. v2 ≠ vec (0::nat) ∧

v4 ≠ vec (0::nat) ∧ v2 ≠ v4 ∧ projection (vector_sub v4 v2) (vector_neg v2) ≠ vec (0::nat) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ∧ dot v1 n < (0::real) ∧ (0::real) < dot v3 n ∧ dot v2 n = (0::real) ∧ dot v4 n = (0::real) ∧ HOL_Light_Import.INTER (closed_segment [(projection (vector_sub v4 v2) (vector_sub v1 v2), projection (vector_sub v4 v2) (vector_sub v3 v2))]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT (projection (vector_sub v4 v2) (vector_neg v2)) EMPTY)) ≠ EMPTY ⟶ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (hull convex (INSERT (vec (0::nat)) (INSERT v2 (INSERT v4 EMPTY)))) ≠ EMPTY

thm Ckqowsa_4_points.lemma_4_points_rotation2:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (hull convex (INSERT (vec (0::nat)) (INSERT v2 (INSERT v4 EMPTY)))) ≠ EMPTY ⟶ (∃ v1′::(real, 3) cart. IN v1′ ball_annulus ∧ vector_norm v1′ = real_of_nat (2::nat) ∧ distance (v1′, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1′, v2) = distance (v1, v2) ∧ distance (v1′, v4) = distance (v1, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1′, v3)]) (hull convex (INSERT (vec (0::nat)) (INSERT v2 (INSERT v4 EMPTY)))) ≠ EMPTY)

thm Ckqowsa_4_points.lemma_4_points_rotation2_full:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (hull convex (INSERT (vec (0::nat)) (INSERT v2 (INSERT v4 EMPTY)))) ≠ EMPTY ⟶ (∃ (v1′::(real, 3) cart) v3′::(real, 3) cart. IN v1′ ball_annulus ∧ IN v3′ ball_annulus ∧ vector_norm v1′ = real_of_nat (2::nat) ∧ vector_norm v3′ = real_of_nat (2::nat) ∧ distance (v1′, v3′) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v1′, v2) = distance (v1, v2) ∧ distance (v1′, v4) = distance (v1, v4) ∧ distance (v2, v3′) = distance (v2, v3) ∧ distance (v3′, v4) = distance (v3, v4) ∧ HOL_Light_Import.INTER (closed_segment [(v1′, v3′)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY)

thm Ckqowsa_4_points.lemma_4_points_circumcenter:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. ¬ collinear (INSERT (vec (0::nat)) (INSERT v2 (INSERT v4 EMPTY))) ∧ IN v2 ball_annulus ∧ IN v4 ball_annulus ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ∧ vector_norm v1 = real_of_nat (2::nat) ∧ vector_norm v3 = real_of_nat (2::nat) ∧ distance (v1, v2) = real_of_nat (2::nat) ∧ distance (v1, v4) = real_of_nat (2::nat) ∧ distance (v3, v2) = real_of_nat (2::nat) ∧ distance (v3, v4) = real_of_nat (2::nat) ⟶ vector_norm (vector_add v1 v3) = real_of_nat (2::nat) ∗ eta_y (vector_norm v2) (vector_norm v4) (distance (v2, v4))

thm Ckqowsa_4_points.PARALLELOGRAM_LAW:

∀ (v1::(real, ?'a::type) cart) v2::(real, ?'a::type) cart. real_of_nat (2::nat) ∗ ((vector_norm v1)^2 + (vector_norm v2)^2) = (vector_norm (vector_add v1 v2))^2 + (vector_norm (vector_sub v1 v2))^2

thm Ckqowsa_4_points.lemma_4_points_contradiction:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v2 ball_annulus ∧ IN v4 ball_annulus ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ HOL_Light_Import.INTER (closed_segment [(v1, v3)]) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) ≠ EMPTY ∧ vector_norm v1 = real_of_nat (2::nat) ∧ vector_norm v3 = real_of_nat (2::nat) ∧ distance (v1, v2) = real_of_nat (2::nat) ∧ distance (v1, v4) = real_of_nat (2::nat) ∧ distance (v3, v2) = real_of_nat (2::nat) ∧ distance (v3, v4) = real_of_nat (2::nat) ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ⟶ False

thm Ckqowsa_4_points.LEMMA_4_POINTS_FINAL:

∀ (v1::(real, 3) cart) (v2::(real, 3) cart) (v3::(real, 3) cart) v4::(real, 3) cart. IN v1 ball_annulus ∧ IN v2 ball_annulus ∧ IN v3 ball_annulus ∧ IN v4 ball_annulus ∧ distance (v1, v3) ≤ real_of_nat (2::nat) ∗ h0 ∧ distance (v2, v4) ≤ real_of_nat (2::nat) ∗ h0 ∧ real_of_nat (2::nat) ≤ distance (v1, v2) ∧ real_of_nat (2::nat) ≤ distance (v1, v4) ∧ real_of_nat (2::nat) ≤ distance (v2, v3) ∧ real_of_nat (2::nat) ≤ distance (v3, v4) ∧ real_of_nat (2::nat) ≤ distance (v1, v3) ∧ real_of_nat (2::nat) ≤ distance (v2, v4) ⟶ HOL_Light_Import.INTER (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v1 (INSERT v3 EMPTY))) (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT v2 (INSERT v4 EMPTY))) = INSERT (vec (0::nat)) EMPTY

thm Ckqowsa.packing:

∀ V::(real, 3) cart ⇒ bool. packing V = (∀ (v::(real, 3) cart) w::(real, 3) cart. IN v V ∧ IN w V ∧ v ≠ w ⟶ real_of_nat (2::nat) ≤ distance (v, w))

thm Ckqowsa.ESTD_fan0:

∀ V::(real, 3) cart ⇒ bool. SUBSET (UNIONS (ESTD V)) V ∧ graph (ESTD V)

thm Ckqowsa.ESTD_fan1:

$\forall V{::}(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \land V \neq EMPTY \longrightarrow fan1\ (vec\ (0{::}nat),\ V,\ ESTD\ V)$

thm Ckqowsa.ESTD_fan2:

$\forall V{::}(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \longrightarrow fan2\ (vec\ (0{::}nat),\ V,\ ESTD\ V)$

thm Ckqowsa.ESTD_fan6:

$\forall V{::}(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \longrightarrow fan6\ (vec\ (0{::}nat),\ V,\ ESTD\ V)$

thm Ckqowsa.fan7_2:

$\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (v{::}(real,\ 3)\ cart)\ w{::}(real,\ 3)\ cart.\ SUBSET\ V\ ball\_annulus \land packing\ V \land IN\ v\ V \land IN\ w\ V \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ v\ EMPTY))\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ w\ EMPTY)) = aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ (INSERT\ v\ EMPTY)\ (INSERT\ w\ EMPTY))$

thm Ckqowsa.fan7_3:

$\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (e{::}(real,\ 3)\ cart \Rightarrow bool)\ u{::}(real,\ 3)\ cart.\ SUBSET\ V\ ball\_annulus \land packing\ V \land IN\ u\ V \land IN\ e\ (ESTD\ V) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ u\ EMPTY))\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ e) = aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ (INSERT\ u\ EMPTY)\ e)$

thm Ckqowsa.fan7_4_0:

$\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (e1{::}(real,\ 3)\ cart \Rightarrow bool)\ e2{::}(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \land IN\ e1\ (ESTD\ V) \land IN\ e2\ (ESTD\ V) \land HAS\_SIZE\ (HOL\_Light\_Import.INTER\ e1\ e2)\ (0{::}nat) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ e1)\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ e2) = aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ e1\ e2)$

thm Ckqowsa.fan7_4_1_one_case:

$\forall (V{::}(real,\ 3)\ cart \Rightarrow bool)\ (v1{::}(real,\ 3)\ cart)\ (v2{::}(real,\ 3)\ cart)\ (v3{::}(real,\ 3)\ cart)\ v4{::}(real,\ 3)\ cart.\ SUBSET\ V\ ball\_annulus \land packing\ V \land IN\ v1\ V \land IN\ v2\ V \land IN\ v3\ V \land IN\ v4\ V \land v1 \neq v3 \land v2 \neq v4 \land v1 = v2 \land v3 \neq v4 \land distance\ (v1,\ v3) \leq real\_of\_nat\ (2{::}nat) * h0 \land distance\ (v2,\ v4) \leq real\_of\_nat\ (2{::}nat) * h0 \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ v1\ (INSERT\ v3\ EMPTY)))\ (aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (INSERT\ v2\ (INSERT\ v4\ EMPTY))) = aff\_ge\ (INSERT\ (vec\ (0{::}nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))\ (INSERT\ v2\ (INSERT\ v4\ EMPTY)))$

thm Ckqowsa.fan7_4_1_cases:

$\forall (v{::}?'a{::}type)\ (w{::}?'a{::}type)\ (v'{::}?'a{::}type)\ w'{::}?'a{::}type.\ v \neq w \land v' \neq w' \land HAS\_SIZE\ (HOL\_Light\_Import.INTER\ (INSERT\ v\ (INSERT\ w\ EMPTY))$

$(INSERT\ v'\ (INSERT\ w'\ EMPTY)))\ (1::nat) \longrightarrow v = v' \land w \neq w' \lor v = w' \land w \neq v' \lor w = v' \land v \neq w' \lor w = w' \land v \neq v'$

thm Ckqowsa.fan7_4_1:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (e1::(real,\ 3)\ cart \Rightarrow bool)\ e2::(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \land IN\ e1\ (ESTD\ V) \land IN\ e2\ (ESTD\ V) \land HAS\_SIZE\ (HOL\_Light\_Import.INTER\ e1\ e2)\ (1::nat) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ e1)\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ e2) = aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ e1\ e2)$

thm Ckqowsa.fan7_4_2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (e1::(real,\ 3)\ cart \Rightarrow bool)\ e2::(real,\ 3)\ cart \Rightarrow bool.\ IN\ e1\ (ESTD\ V) \land IN\ e2\ (ESTD\ V) \land HAS\_SIZE\ (HOL\_Light\_Import.INTER\ e1\ e2)\ (2::nat) \longrightarrow HOL\_Light\_Import.INTER\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ e1)\ (aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ e2) = aff\_ge\ (INSERT\ (vec\ (0::nat))\ EMPTY)\ (HOL\_Light\_Import.INTER\ e1\ e2)$

thm Ckqowsa.ESTD_fan7:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \longrightarrow fan7\ (vec\ (0::nat),\ V,\ ESTD\ V)$

thm Ckqowsa.CKQOWSA:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ SUBSET\ V\ ball\_annulus \land packing\ V \land V \neq EMPTY \longrightarrow FAN\ (vec\ (0::nat),\ V,\ ESTD\ V)$

thm Tame_general.RUNOQPQ:

$\forall H::?'a::type\ hypermap.\ tame\_hypermap\ H \longrightarrow restricted\_hypermap\ H$

thm Tame_general.UBHDEUU2:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \land SUBSET\ V\ ball\_annulus \land V \neq EMPTY \longrightarrow FAN\ (vec\ (0::nat),\ V,\ ECTC\ V)$

thm Tame_general.ly_EQ_lmfun:

$\forall x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ vector\_norm\ (fst\ x) \leq real\_of\_nat\ (2::nat) * h0 \longrightarrow lmfun\ (h\_dart\ x) = ly\ (vector\_norm\ (fst\ x))$

thm Tame_general.DIHV_EQ_DIH_Y:

$\forall (v0::(real,\ 3)\ cart)\ (v1::(real,\ 3)\ cart)\ (v2::(real,\ 3)\ cart)\ v3::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v2\ EMPTY))) \land \neg\ collinear\ (INSERT\ v0\ (INSERT\ v1\ (INSERT\ v3\ EMPTY))) \longrightarrow LET\ (\lambda v01::real.\ LET\_END\ (LET\ (\lambda v02::real.\ LET\_END\ (LET\ (\lambda v03::real.\ LET\_END\ (LET\ (\lambda v12::real.\ LET\_END\ (LET\ (\lambda v13::real.\ LET\_END\ (LET\ (\lambda v23::real.\ LET\_END\ (dihV\ v0\ v1\ v2\ v3 = dih\_y\ v01\ v02\ v03\ v23\ v13\ v12))\ (distance\ (v2,\ v3))))\ (distance\ (v1,\ v3))))\ (distance\ (v1,\ v2))))\ (distance\ (v0,\ v3))))\ (distance\ (v0,\ v2))))\ (distance\ (v0,\ v1))$

thm Tame_general.DIFF_LEMMA:

$\forall (A::?'a::type \Rightarrow bool)\ B::?'a::type \Rightarrow bool.\ SUBSET\ A\ B \longrightarrow A = DIFF\ B\ (DIFF\ B\ A)$

thm Tame_general.CONTRAVENING_FAN:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ contravening\ V \longrightarrow FAN\ (vec\ (0::nat),\ V,\ ESTD\ V)$

thm Tame_general.CONTRAVENING_IMP_SURROUNDED:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ contravening\ V \wedge IN\ v\ V \longrightarrow surrounded\_node\ (V,\ ESTD\ V)\ v$

thm Tame_general.CONTRAVENING_IMP_FULLY_SURROUNDED:

$\forall V::(real,\ 3)\ cart \Rightarrow bool.\ FAN\ (vec\ (0::nat),\ V,\ ESTD\ V) \wedge contravening\ V \longrightarrow fully\_surrounded\ (V,\ ESTD\ V)$

thm Tame_general.CONTRAVENING_IMP_IN_DART1_OF_FAN:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ contravening\ V \wedge IN\ x\ (dart\_of\_fan\ (V,\ ESTD\ V)) \longrightarrow IN\ x\ (dart1\_of\_fan\ (V,\ ESTD\ V))$

thm Tame_general.CONTRAVENING_IMP_DART_FST_NEQ_SND:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ contravening\ V \wedge IN\ x\ (dart\_of\_fan\ (V,\ ESTD\ V)) \longrightarrow fst\ x \neq snd\ x$

thm Tame_general.CONTRAVENING_DIST:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ contravening\ V \wedge IN\ v\ V \longrightarrow DECIMAL\ (20::nat)\ (10::nat) \leq distance\ (vec\ (0::nat),\ v) \wedge distance\ (vec\ (0::nat),\ v) \leq DECIMAL\ (252::nat)\ (100::nat) \wedge (\forall w::(real,\ 3)\ cart.\ IN\ w\ V \wedge v \neq w \longrightarrow DECIMAL\ (20::nat)\ (10::nat) \leq distance\ (v,\ w))$

thm Tame_general.IN_ESTD:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ (ESTD\ V) = (IN\ v\ V \wedge IN\ w\ V \wedge v \neq w \wedge distance\ (v,\ w) \leq DECIMAL\ (252::nat)\ (100::nat))$

thm Tame_general.CONTRAVENING_ESTD_DIST:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (v::(real,\ 3)\ cart)\ w::(real,\ 3)\ cart.\ contravening\ V \wedge IN\ v\ V \wedge IN\ w\ V \wedge IN\ (INSERT\ v\ (INSERT\ w\ EMPTY))\ (ESTD\ V) \longrightarrow DECIMAL\ (20::nat)\ (10::nat) \leq distance\ (v,\ w) \wedge distance\ (v,\ w) \leq DECIMAL\ (252::nat)\ (100::nat)$

thm Tame_general.CONTRAVENING_DART_DIST:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ x::(real,\ 3)\ cart \times (real,\ 3)\ cart.\ contravening\ V \wedge IN\ x\ (dart\_of\_fan\ (V,\ ESTD\ V)) \longrightarrow DECIMAL\ (20::nat)\ (10::nat) \leq distance\ x \wedge distance\ x \leq DECIMAL\ (252::nat)\ (100::nat)$

thm Tame_general.CONTRAVENING_LMFUN_BOUND:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ v::(real, 3)\ cart.\ contravening\ V \wedge IN\ v\ V \longrightarrow lmfun\ (vector\_norm\ v\ /\ real\_of\_nat\ (2::nat)) \leq (1::real)$

thm Tame_general.CONTRAVENING_IMP_AZIM_DART_EQ_AZIM:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ contravening\ V \wedge IN\ (v, w)\ (dart\_of\_fan\ (V, ESTD\ V)) \longrightarrow azim\_dart\ (V, ESTD\ V)\ (v, w) = azim\ (vec\ (0::nat))\ v\ w\ (sigma\_fan\ (vec\ (0::nat))\ V\ (ESTD\ V)\ v\ w)$

thm Tame_general.CONTRAVENING_IMP_NOT_COPLANAR:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ contravening\ V \wedge IN\ (v, w)\ (dart\_of\_fan\ (V, ESTD\ V)) \longrightarrow \neg\ coplanar\ (INSERT\ (vec\ (0::nat))\ (INSERT\ v\ (INSERT\ w\ (INSERT\ (sigma\_fan\ (vec\ (0::nat))\ V\ (ESTD\ V)\ v\ w)\ EMPTY))))$

thm Tame_general.CONTRAVENING_AZIM_DART_EQ_DIH_Y:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (v::(real, 3)\ cart)\ w::(real, 3)\ cart.\ contravening\ V \wedge IN\ (v, w)\ (dart\_of\_fan\ (V, ESTD\ V)) \longrightarrow LET\ (\lambda w'::(real, 3)\ cart.\ LET\_END\ (LET\ (\lambda y1::real.\ LET\_END\ (LET\ (\lambda y2::real.\ LET\_END\ (LET\ (\lambda y3::real.\ LET\_END\ (LET\ (\lambda y4::real.\ LET\_END\ (LET\ (\lambda y5::real.\ LET\_END\ (LET\ (\lambda y6::real.\ LET\_END\ (azim\_dart\ (V, ESTD\ V)\ (v, w) = dih\_y\ y1\ y2\ y3\ y4\ y5\ y6))\ (distance\ (v, w))))\ (distance\ (v, w'))))\ (distance\ (w, w'))))\ (vector\_norm\ w')))\ (vector\_norm\ w)))\ (vector\_norm\ v)))\ (sigma\_fan\ (vec\ (0::nat))\ V\ (ESTD\ V)\ v\ w)$

thm Tame_general.CONTRAVENING_IMP_CARD_FACE_GE_3:

$\forall V::(real, 3)\ cart \Rightarrow bool.\ contravening\ V \longrightarrow (\forall x::(real, 3)\ cart \times (real, 3)\ cart.\ IN\ x\ (dart\_of\_fan\ (V, ESTD\ V)) \longrightarrow (3::nat) \leq CARD\ (face\ (hypermap\_of\_fan\ (V, ESTD\ V))\ x))$

thm Tame_general.NODE_TYPE_lemma:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ simple\_hypermap\ H \wedge IN\ x\ (dart\ H) \longrightarrow type\_of\_node\ H\ x = (CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2262::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%2262\ (IN\ y\ (node\ H\ x) \wedge CARD\ (face\ H\ y) = (3::nat))\ y)),\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2263::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%2263\ (IN\ y\ (node\ H\ x) \wedge CARD\ (face\ H\ y) = (4::nat))\ y)),\ CARD\ (GSPEC\ (\lambda GEN\%PVAR\%2264::?'a::type.\ \exists y::?'a::type.\ SETSPEC\ GEN\%PVAR\%2264\ (IN\ y\ (node\ H\ x) \wedge (5::nat) \leq CARD\ (face\ H\ y))\ y)))$

thm Tame_general.FULLY_SURROUNDED_IMP_CARD_NODE_EQ_SUM_NODE_TYPE:

$\forall (V::(real, 3)\ cart \Rightarrow bool)\ (E::((real, 3)\ cart \Rightarrow bool) \Rightarrow bool)\ x::(real, 3)\ cart \times (real, 3)\ cart.\ FAN\ (vec\ (0::nat), V, E) \wedge simple\_hypermap\ (hypermap\_of\_fan\ (V, E)) \wedge fully\_surrounded\ (V, E) \wedge IN\ x\ (dart\_of\_fan\ (V, E)) \longrightarrow LET\ (GABS\ (\lambda f::nat \times nat \times nat \Rightarrow bool.\ \forall (p::nat)\ (q::nat)\ r::nat.\ GEQ\ (f\ (p,$

q, r)) (LET_END (CARD (node (hypermap_of_fan (V, E)) x) = p + (q + r))))) (type_of_node (hypermap_of_fan (V, E)) x)

thm Tame_general.tauVEF_EQ_taum:

∀(V::(real, 3) cart ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. contravening V ∧ IN f (face_set_of_fan (V, ESTD V)) ∧ CARD f = (3::nat) ⟶ (∀(v::(real, 3) cart) w::(real, 3) cart. IN (v, w) f ⟶ LET (λw′::(real, 3) cart. LET_END (LET (λy1::real. LET_END (LET (λy2::real. LET_END (LET (λy3::real. LET_END (LET (λy4::real. LET_END (LET (λy5::real. LET_END (LET (λy6::real. LET_END (tauVEF (V, ESTD V, f) = taum y1 y2 y3 y4 y5 y6)) (distance (v, w)))) (distance (v, w′)))) (distance (w, w′)))) (vector_norm w′))) (vector_norm w))) (vector_norm v))) (sigma_fan (vec (0::nat)) V (ESTD V) v w))

thm Tame_general.CONTRAVENING_TAUVEF_EQ_TAUM:

∀(V::(real, 3) cart ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. contravening V ∧ IN (v, w) (dart_of_fan (V, ESTD V)) ∧ CARD (face (hypermap_of_fan (V, ESTD V)) (v, w)) = (3::nat) ⟶ LET (λw′::(real, 3) cart. LET_END (LET (λy1::real. LET_END (LET (λy2::real. LET_END (LET (λy3::real. LET_END (LET (λy4::real. LET_END (LET (λy5::real. LET_END (LET (λy6::real. LET_END (tauVEF (V, ESTD V, face (hypermap_of_fan (V, ESTD V)) (v, w)) = taum y1 y2 y3 y4 y5 y6)) (distance (v, w)))) (distance (v, w′)))) (distance (w, w′)))) (vector_norm w′))) (vector_norm w))) (vector_norm v))) (sigma_fan (vec (0::nat)) V (ESTD V) v w)

thm Tame_general.CONTRAVENING_TRIANGULAR_FACE_DIST:

∀(V::(real, 3) cart ⇒ bool) (v::(real, 3) cart) w::(real, 3) cart. contravening V ∧ IN (v, w) (dart_of_fan (V, ESTD V)) ∧ CARD (face (hypermap_of_fan (V, ESTD V)) (v, w)) = (3::nat) ⟶ LET (λw′::(real, 3) cart. LET_END (LET (λy1::real. LET_END (LET (λy2::real. LET_END (LET (λy3::real. LET_END (LET (λy4::real. LET_END (LET (λy5::real. LET_END (LET (λy6::real. LET_END ((real_of_nat (2::nat) ≤ y1 ∧ y1 ≤ DECIMAL (252::nat) (100::nat)) ∧ (real_of_nat (2::nat) ≤ y2 ∧ y2 ≤ DECIMAL (252::nat) (100::nat)) ∧ (real_of_nat (2::nat) ≤ y3 ∧ y3 ≤ DECIMAL (252::nat) (100::nat)) ∧ (real_of_nat (2::nat) ≤ y4 ∧ y4 ≤ DECIMAL (252::nat) (100::nat)) ∧ (real_of_nat (2::nat) ≤ y5 ∧ y5 ≤ DECIMAL (252::nat) (100::nat)) ∧ real_of_nat (2::nat) ≤ y6 ∧ y6 ≤ DECIMAL (252::nat) (100::nat))) (distance (v, w)))) (distance (v, w′)))) (distance (w, w′)))) (vector_norm w′))) (vector_norm w))) (vector_norm v))) (sigma_fan (vec (0::nat)) V (ESTD V) v w)

thm Tame_general.TRIANGULAR_FACE_AZIM_DART_BOUNDS:

(∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. ineq (dart_std3 y1 y2 y3 y4 y5 y6) (DECIMAL (852::nat) (1000::nat) < dih_y y1 y2 y3 y4 y5 y6)) ∧ (∀(y1::real) (y2::real) (y3::real) (y4::real) (y5::real) y6::real. ineq (dart_std3 y1 y2 y3 y4 y5 y6) (dih_y y1 y2 y3 y4 y5 y6 < DECIMAL (1893::nat) (1000::nat))) ⟶ (∀(V::(real, 3) cart ⇒ bool) x::(real, 3) cart ×

(*real*, *3*) *cart*. *contravening* *V* ∧ *IN* *x* (*dart_of_fan* (*V*, *ESTD* *V*)) ∧ *CARD* (*face* (*hypermap_of_fan* (*V*, *ESTD* *V*)) *x*) = (*3*::*nat*) ⟶ *DECIMAL* (*852*::*nat*) (*1000*::*nat*) < *azim_dart* (*V*, *ESTD* *V*) *x* ∧ *azim_dart* (*V*, *ESTD* *V*) *x* < *DECIMAL* (*1893*::*nat*) (*1000*::*nat*))

thm Tame_general.JGTDEBU2:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *plain_hypermap* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.JGTDEBU5:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *is_edge_nondegenerate* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.JGTDEBU6:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *no_loops* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.JGTDEBU7:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *is_no_double_joints* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.CONTRAVENING_HAS_SIZE_lemma:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ (∃ *n*>*0*::*nat*. *HAS_SIZE* *V* *n*)

thm Tame_general.JGTDEBU8:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *simple_hypermap* (*hypermap_of_fan* (*V*, *ESTD* *V*)) ⟶ *contravening* *V* ⟶ (*3*::*nat*) ≤ *number_of_faces* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.JGTDEBU10:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *tame_10* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_general.JGTDEBU11:

∀ *V*::(*real*, *3*) *cart* ⇒ *bool*. *contravening* *V* ⟶ *tame_11a* (*hypermap_of_fan* (*V*, *ESTD* *V*))

thm Tame_opposite.tuple_opposite_hypermap:

∀ *H*::*?'a*::*type* *hypermap*. *tuple_hypermap* (*opposite_hypermap* *H*) = (*dart* *H*, *face_map* *H* ∘ *node_map* *H*, *HOL_Light_Import*.*inverse* (*node_map* *H*), *HOL_Light_Import*.*inverse* (*face_map* *H*))

thm Tame_opposite.opposite_hypermap_plain:

∀ *H*::*?'a*::*type* *hypermap*. *plain_hypermap* *H* ⟶ *plain_hypermap* (*opposite_hypermap* *H*)

thm Tame_opposite.opposite_components:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *dart* (*opposite_hypermap H*) = *dart H* $\wedge$ *node* (*opposite_hypermap H*) *x* = *node H x* $\wedge$ *face* (*opposite_hypermap H*) *x* = *face H x*

thm Tame_opposite.opposite_hypermap_simple:

$\forall$ *H*::*?'a*::*type hypermap*. *simple_hypermap H* $\longrightarrow$ *simple_hypermap* (*opposite_hypermap H*)

thm Tame_opposite.hypermap_eq_lemma:

$\forall$ *H*::*?'a*::*type hypermap*. *tuple_hypermap H* = (*dart H*, *edge_map H*, *node_map H*, *face_map H*)

thm Tame_opposite.opposite_opposite_hypermap_eq_hypermap:

$\forall$ *H*::*?'a*::*type hypermap*. *opposite_hypermap* (*opposite_hypermap H*) = *H*

thm Tame_opposite.truncated_path_lemma:

$\forall$ (*H*::*?'a*::*type hypermap*) (*p*::*nat* $\Rightarrow$ *?'a*::*type*) (*q*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. *is_path H p n* $\wedge$ ($\forall$ *i*$\leq$*n*. *q i* = *p i*) $\longrightarrow$ *is_path H q n*

thm Tame_opposite.opposite_path:

$\forall$ (*H*::*?'a*::*type hypermap*) (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. *is_path H p n* $\longrightarrow$ ($\exists$ (*q*::*nat* $\Rightarrow$ *?'a*::*type*) *m*::*nat*. *is_path* (*opposite_hypermap H*) *q m* $\wedge$ *q* (*0*::*nat*) = *p* (*0*::*nat*) $\wedge$ *q m* = *p n*)

thm Tame_opposite.opposite_path_alt:

$\forall$ (*H*::*?'a*::*type hypermap*) (*q*::*nat* $\Rightarrow$ *?'a*::*type*) *m*::*nat*. *is_path* (*opposite_hypermap H*) *q m* $\longrightarrow$ ($\exists$ (*p*::*nat* $\Rightarrow$ *?'a*::*type*) *n*::*nat*. *is_path H p n* $\wedge$ *p* (*0*::*nat*) = *q* (*0*::*nat*) $\wedge$ *p n* = *q m*)

thm Tame_opposite.opposite_sets_of_components:

$\forall$ *H*::*?'a*::*type hypermap*. *node_set* (*opposite_hypermap H*) = *node_set H* $\wedge$ *face_set* (*opposite_hypermap H*) = *face_set H* $\wedge$ *set_of_components* (*opposite_hypermap H*) = *set_of_components H*

thm Tame_opposite.opposite_hypermap_connected:

$\forall$ *H*::*?'a*::*type hypermap*. *connected_hypermap H* $\longrightarrow$ *connected_hypermap* (*opposite_hypermap H*)

thm Tame_opposite.opposite_nondegenerate:

$\forall$ *H*::*?'a*::*type hypermap*. *plain_hypermap H* $\wedge$ *is_edge_nondegenerate H* $\longrightarrow$ *is_edge_nondegenerate* (*opposite_hypermap H*)

thm Tame_opposite.opposite_no_double_joints:

$\forall$ *H*::*?'a*::*type hypermap*. *is_no_double_joints H* $\wedge$ *plain_hypermap H* $\longrightarrow$ *is_no_double_joints* (*opposite_hypermap H*)

thm Tame_opposite.plain_hypermap_edge:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *plain_hypermap H* $\longrightarrow$ *edge H x* = *INSERT x* (*INSERT* (*edge_map H x*) *EMPTY*)

thm Tame_opposite.opposite_no_loops:

$\forall$ *H*::*?'a*::*type hypermap*. *no_loops H* $\wedge$ *plain_hypermap H* $\longrightarrow$ *no_loops* (*opposite_hypermap H*)

thm Tame_opposite.edge_CARD_dart:

$\forall$ *H*::*?'a*::*type hypermap*. *plain_hypermap H* $\wedge$ *is_edge_nondegenerate H* $\longrightarrow$ *CARD* (*dart H*) = (*2*::*nat*) $*$ *number_of_edges H*

thm Tame_opposite.edge_CARD:

$\forall$ *H*::*?'a*::*type hypermap*. *plain_hypermap H* $\wedge$ *is_edge_nondegenerate H* $\longrightarrow$ *number_of_edges H* = *CARD* (*dart H*) *div* (*2*::*nat*)

thm Tame_opposite.opposite_planar:

$\forall$ *H*::*?'a*::*type hypermap*. *planar_hypermap H* $\wedge$ *is_edge_nondegenerate H* $\wedge$ *plain_hypermap H* $\longrightarrow$ *planar_hypermap* (*opposite_hypermap H*)

thm Tame_opposite.CARD_faces1:

$\forall$ *H*::*?'a*::*type hypermap*. *tame_8 H* $\longrightarrow$ *tame_8* (*opposite_hypermap H*)

thm Tame_opposite.CARD_in_face:

$\forall$ *H*::*?'a*::*type hypermap*. *tame_9a H* $\longrightarrow$ *tame_9a* (*opposite_hypermap H*)

thm Tame_opposite.CARD_nodes:

$\forall$ *H*::*?'a*::*type hypermap*. *tame_10 H* $\longrightarrow$ *tame_10* (*opposite_hypermap H*)

thm Tame_opposite.CARD_in_node:

$\forall$ *H*::*?'a*::*type hypermap*. *tame_11a H* $\wedge$ *tame_11b H* $\longrightarrow$ *tame_11a* (*opposite_hypermap H*) $\wedge$ *tame_11b* (*opposite_hypermap H*)

thm Tame_opposite.the_SAME_orbit_triangles:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *set_of_triangles_meeting_node H x* = *set_of_triangles_meeting_node* (*opposite_hypermap H*) *x*

thm Tame_opposite.the_SAME_orbit_quadrilaterals:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *set_of_quadrilaterals_meeting_node H x* = *set_of_quadrilaterals_meeting_node* (*opposite_hypermap H*) *x*

thm Tame_opposite.the_SAME_orbit_exceptional:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *set_of_exceptional_meeting_node H x* = *set_of_exceptional_meeting_node* (*opposite_hypermap H*) *x*

thm Tame_opposite.the_SAME_orbit_face:

$\forall$ (*H*::*?'a*::*type hypermap*) *x*::*?'a*::*type*. *set_of_face_meeting_node H x* = *set_of_face_meeting_node* (*opposite_hypermap H*) *x*

thm Tame_opposite.the_SAME_type:

$\forall (H::?'a::type\ hypermap)\ x::?'a::type.\ type\_of\_node\ H\ x = type\_of\_node\ (opposite\_hypermap\ H)\ x$

thm Tame_opposite.opposite_tame_12o:

$\forall H::?'a::type\ hypermap.\ tame\_12o\ H \longrightarrow tame\_12o\ (opposite\_hypermap\ H)$

thm Tame_opposite.opposite_tame_13a:

$\forall H::?'a::type\ hypermap.\ tame\_13a\ H \longrightarrow tame\_13a\ (opposite\_hypermap\ H)$

thm Tame_opposite.tame_opposite_hypermap:

$\forall H::?'a::type\ hypermap.\ tame\_hypermap\ H \longrightarrow tame\_hypermap\ (opposite\_hypermap\ H)$

thm Tame_opposite.PPHEUFG:

$\forall H::?'a::type\ hypermap.\ tame\_hypermap\ H = tame\_hypermap\ (opposite\_hypermap\ H)$

thm Fatugpd.UBHDEUU2_hypothesis:

$UBHDEUU2\_hypothesis = (\forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \longrightarrow FAN\ (vec\ (0::nat),\ V,\ ECTC\ V))$

thm Fatugpd.UBHDEUU2_quasi:

$UBHDEUU2\_hypothesis \Longrightarrow \forall V::(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \longrightarrow FAN\ (vec\ (0::nat),\ V,\ ECTC\ V)$

thm Fatugpd.finite_num_func_attain_max:

$\forall (S::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow nat.\ FINITE\ S \wedge S \neq EMPTY \longrightarrow (\exists x::?'a::type.\ IN\ x\ S \wedge (\forall y::?'a::type.\ IN\ y\ S \longrightarrow f\ y \leq f\ x))$

thm Fatugpd.sup_property1:

$\forall S::real \Rightarrow bool.\ S \neq EMPTY \wedge (\exists b::real.\ \forall x::real.\ IN\ x\ S \longrightarrow x \leq b) \longrightarrow (\forall epsilon>0::real.\ \exists x::real.\ IN\ x\ S \wedge HOL\_Light\_Import.sup\ S - epsilon < x)$

thm Fatugpd.bdd_num_func_attain_max:

$\forall (S::?'a::type \Rightarrow bool)\ f::?'a::type \Rightarrow nat.\ S \neq EMPTY \wedge (\exists m::nat.\ \forall x::?'a::type.\ IN\ x\ S \longrightarrow f\ x \leq m) \longrightarrow (\exists x::?'a::type.\ IN\ x\ S \wedge (\forall y::?'a::type.\ IN\ y\ S \longrightarrow f\ y \leq f\ x))$

thm Fatugpd.BIJ_CARD_EQ:

$\forall (V::?'b::type \Rightarrow bool)\ (U::?'a::type \Rightarrow bool)\ f::?'b::type \Rightarrow ?'a::type.\ FINITE\ V \wedge BIJ\ f\ V\ U \longrightarrow CARD\ U = CARD\ V$

thm Fatugpd.SUP_lt:

$\forall (S::real \Rightarrow bool)\ b::real.\ S \neq EMPTY \wedge FINITE\ S \wedge (\forall x::real.\ IN\ x\ S \longrightarrow x < b) \longrightarrow HOL\_Light\_Import.sup\ S < b$

thm Fatugpd.epsilon_lemma:

$\forall (a::real)\ b::real.\ a < b \longrightarrow (\exists epsilon{>}0::real.\ a + epsilon < b)$

thm Fatugpd.norm_normalize:

$\forall v::(real,\ 3)\ cart.\ v \neq vec\ (0::nat) \longrightarrow vector\_norm\ (normalize\ v) = (1::real)$

thm Fatugpd.normalize_vec_0:

$normalize\ (vec\ (0::nat)) = vec\ (0::nat)$

thm Fatugpd.norm_mul_normalize:

$\forall v::(real,\ 3)\ cart.\ \%\ (vector\_norm\ v)\ (normalize\ v) = v$

thm Fatugpd.dot_normalize:

$\forall v::(real,\ 3)\ cart.\ dot\ v\ (normalize\ v) = vector\_norm\ v$

thm Fatugpd.fourier:

$\forall (v::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow v = vector\_add\ (\%\ (dot\ v\ e1)\ e1)\ (vector\_add\ (\%\ (dot\ v\ e2)\ e2)\ (\%\ (dot\ v\ e3)\ e3))$

thm Fatugpd.norm_lemma1:

$\forall (v::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow vector\_norm\ v = sqrt\ ((dot\ v\ e1)^2 + ((dot\ v\ e2)^2 + (dot\ v\ e3)^2))$

thm Fatugpd.coordinates_lemma:

$\forall (v::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ (e3::(real,\ 3)\ cart)\ (x::real)\ (y::real)\ z::real.\ orthonormal\ e1\ e2\ e3 \wedge v = vector\_add\ (\%\ x\ e1)\ (vector\_add\ (\%\ y\ e2)\ (\%\ z\ e3)) \longrightarrow x = dot\ v\ e1 \wedge y = dot\ v\ e2 \wedge z = dot\ v\ e3$

thm Fatugpd.dot_coordinates:

$\forall (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ e3::(real,\ 3)\ cart.\ orthonormal\ e1\ e2\ e3 \longrightarrow dot\ v\ u = dot\ v\ e1 * dot\ u\ e1 + (dot\ v\ e2 * dot\ u\ e2 + dot\ v\ e3 * dot\ u\ e3)$

thm Fatugpd.dot_coordinates_2:

$\forall (v::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ (e3::(real,\ 3)\ cart)\ (x::real)\ (y::real)\ (z::real)\ (a::real)\ (b::real)\ c::real.\ orthonormal\ e1\ e2\ e3 \wedge v = vector\_add\ (\%\ x\ e1)\ (vector\_add\ (\%\ y\ e2)\ (\%\ z\ e3)) \wedge u = vector\_add\ (\%\ a\ e1)\ (vector\_add\ (\%\ b\ e2)\ (\%\ c\ e3)) \longrightarrow dot\ v\ u = x * a + (y * b + z * c)$

thm Fatugpd.norm_lemma2:

$\forall (v::(real,\ 3)\ cart)\ (e1::(real,\ 3)\ cart)\ (e2::(real,\ 3)\ cart)\ (e3::(real,\ 3)\ cart)\ (x::real)\ (y::real)\ z::real.\ orthonormal\ e1\ e2\ e3 \wedge v = vector\_add\ (\%\ x\ e1)\ (vector\_add\ (\%\ y\ e2)\ (\%\ z\ e3)) \longrightarrow vector\_norm\ v = sqrt\ (x^2 + (y^2 + z^2))$

thm Fatugpd.dot_gt_0:

$\forall (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ ((0::real) < dot\ v\ u) = ((0::real) < dot\ v\ (normalize\ u))$

thm Fatugpd.dot_eq_0:

$\forall (v::(real,\ 3)\ cart)\ u::(real,\ 3)\ cart.\ (dot\ v\ u = (0::real)) = (dot\ (normalize\ v)\ u = (0::real))$

thm Fatugpd.azim_ge_azim_dart:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ (E::((real,\ 3)\ cart \Rightarrow bool) \Rightarrow bool)\ (w::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ FAN\ (vec\ (0::nat),\ V,\ E) \wedge IN\ w\ V \wedge IN\ u\ (set\_of\_edge\ w\ V\ E) \wedge IN\ v\ (set\_of\_edge\ w\ V\ E) \wedge w \neq u \wedge v \neq u \longrightarrow azim\_dart\ (V,\ E)\ (w,\ u) \leq azim\ (vec\ (0::nat))\ w\ u\ v$

thm DEF_set_of_iso:

$set\_of\_iso = (\lambda\_7476875::(real,\ 3)\ cart \Rightarrow bool.\ GSPEC\ (\lambda GEN\%PVAR\%2271::(real,\ 3)\ cart.\ \exists w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2271\ (IN\ w\ \_7476875 \wedge set\_of\_edge\ w\ \_7476875\ (ECTC\ \_7476875) = EMPTY)\ w))$

thm Fatugpd.set_of_iso:

$\forall W::(real,\ 3)\ cart \Rightarrow bool.\ set\_of\_iso\ W = GSPEC\ (\lambda GEN\%PVAR\%2271::(real,\ 3)\ cart.\ \exists w::(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2271\ (IN\ w\ W \wedge set\_of\_edge\ w\ W\ (ECTC\ W) = EMPTY)\ w)$

thm Fatugpd.lemma2:

$\forall (V::(real,\ 3)\ cart \Rightarrow bool)\ v::(real,\ 3)\ cart.\ packing\ V \wedge FINITE\ V \wedge IN\ v\ V \longrightarrow (\exists epsilon>0::real.\ \forall w::(real,\ 3)\ cart.\ IN\ w\ V \wedge w \neq v \wedge \neg\ IN\ w\ (set\_of\_edge\ v\ V\ (ECTC\ V)) \longrightarrow real\_of\_nat\ (2::nat) + epsilon < distance\ (v,\ w))$

thm Fatugpd.lemma3:

$\forall (w::(real,\ 3)\ cart)\ W::(real,\ 3)\ cart \Rightarrow bool.\ packing\ W \wedge IN\ w\ W \wedge set\_of\_edge\ w\ W\ (ECTC\ W) \neq EMPTY \wedge \neg\ surrounded\_node\ (W,\ ECTC\ W)\ w \longrightarrow (\exists u::(real,\ 3)\ cart.\ IN\ u\ (set\_of\_edge\ w\ W\ (ECTC\ W)) \wedge pi \leq azim\_dart\ (W,\ ECTC\ W)\ (w,\ u))$

thm Fatugpd.lemma4:

$\forall (w::(real,\ 3)\ cart)\ (u::(real,\ 3)\ cart)\ v::(real,\ 3)\ cart.\ \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ w\ (INSERT\ u\ EMPTY))) \wedge \neg\ collinear\ (INSERT\ (vec\ (0::nat))\ (INSERT\ w\ (INSERT\ v\ EMPTY))) \wedge pi \leq azim\ (vec\ (0::nat))\ w\ u\ v \longrightarrow dot\ (cross\ w\ u)\ v \leq (0::real)$

thm Fatugpd.lemma5:

$\forall\,(V{::}(real,\ 3)\ cart \Rightarrow bool)\ v{::}(real,\ 3)\ cart.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \wedge IN\ v\ V \longrightarrow (\forall\,u{::}(real,\ 3)\ cart.\ IN\ u\ (set\_of\_edge\ v\ V\ (ECTC\ V)) \longrightarrow (0{::}real) < dot\ v\ u)$

thm Fatugpd.lemma6:

$\forall\,(v{::}(real,\ 3)\ cart)\ (u{::}(real,\ 3)\ cart)\ (phi{::}real)\ (e1{::}(real,\ 3)\ cart)\ (e2{::}(real,\ 3)\ cart)\ e3{::}(real,\ 3)\ cart.\ e1 = normalize\ v \wedge orthonormal\ e1\ e2\ e3 \wedge u = vector\_add\ (\%\ (cos\ phi * vector\_norm\ v)\ e1)\ (\%\ (sin\ phi * vector\_norm\ v)\ e2) \longrightarrow vector\_norm\ u = vector\_norm\ v$

thm Fatugpd.lemma7:

$\forall\,(v{::}(real,\ 3)\ cart)\ (u{::}(real,\ 3)\ cart)\ (phi{::}real)\ (e1{::}(real,\ 3)\ cart)\ (e2{::}(real,\ 3)\ cart)\ e3{::}(real,\ 3)\ cart.\ e1 = normalize\ v \wedge orthonormal\ e1\ e2\ e3 \wedge (0{::}real) < phi \wedge phi < pi \wedge (0{::}real) < vector\_norm\ v \wedge u = vector\_add\ (\%\ (cos\ phi * vector\_norm\ v)\ e1)\ (\%\ (sin\ phi * vector\_norm\ v)\ e2) \longrightarrow (\forall\,w{::}(real,\ 3)\ cart.\ (0{::}real) < dot\ w\ e1 \wedge dot\ w\ e2 \le (0{::}real) \longrightarrow distance\ (v,\ w) < distance\ (u,\ w))$

thm Fatugpd.lemma8:

$\forall\,(w{::}(real,\ 3)\ cart)\ W{::}(real,\ 3)\ cart \Rightarrow bool.\ IN\ w\ W \longrightarrow set\_of\_edge\ w\ W\ (ECTC\ W) = GSPEC\ (\lambda GEN\%PVAR\%2273{::}(real,\ 3)\ cart.\ \exists\,v{::}(real,\ 3)\ cart.\ SETSPEC\ GEN\%PVAR\%2273\ (IN\ v\ W \wedge distance\ (w,\ v) = real\_of\_nat\ (2{::}nat))\ v)$

thm PI2_BOUNDS_conjunct1:

$pi\ /\ real\_of\_nat\ (2{::}nat) < real\_of\_nat\ (2{::}nat)$

thm PI2_BOUNDS_conjunct0:

$(0{::}real) < pi\ /\ real\_of\_nat\ (2{::}nat)$

thm Fatugpd.FATUGPD_quasi:

$UBHDEUU2\_hypothesis \longrightarrow (\forall\,V{::}(real,\ 3)\ cart \Rightarrow bool.\ packing\ V \wedge SUBSET\ V\ ball\_annulus \longrightarrow (\exists\,(W{::}(real,\ 3)\ cart \Rightarrow bool)\ phi{::}(real,\ 3)\ cart \Rightarrow (real,\ 3)\ cart.\ BIJ\ phi\ V\ W \wedge (\forall\,v{::}(real,\ 3)\ cart.\ IN\ v\ V \longrightarrow vector\_norm\ v = vector\_norm\ (phi\ v)) \wedge (\forall\,w{::}(real,\ 3)\ cart.\ IN\ w\ W \longrightarrow set\_of\_edge\ w\ W\ (ECTC\ W) = EMPTY \vee surrounded\_node\ (W,\ ECTC\ W)\ w)))$

thm Crttxat_tame.ORBIT_MAP_PAIR_SUM_lemma:

$\forall\,(P{::}?'a{::}type \Rightarrow real)\ (s{::}?'a{::}type \times ?'a{::}type \Rightarrow bool)\ (f{::}?'a{::}type \times ?'a{::}type \Rightarrow ?'a{::}type \times ?'a{::}type)\ (g{::}?'a{::}type \times ?'a{::}type \Rightarrow ?'a{::}type)\ x{::}?'a{::}type \times ?'a{::}type.\ FINITE\ s \wedge permutes\ f\ s \wedge (\forall\,y{::}?'a{::}type \times ?'a{::}type.\ IN\ y\ (orbit\_map\ f\ x) \longrightarrow f\ y = (snd\ y,\ g\ y)) \longrightarrow sum\ (orbit\_map\ f\ x)\ (\lambda x{::}?'a{::}type \times ?'a{::}type.\ P\ (fst\ x)) = sum\ (orbit\_map\ f\ x)\ (\lambda x{::}?'a{::}type \times ?'a{::}type.\ P\ (snd\ x))$

thm Crttxat_tame.FACE_SUM_lemma:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) (x::(real, 3) cart × (real, 3) cart) P::(real, 3) cart ⇒ real. FAN (vec (0::nat), V, E) ∧ IN x (dart1_of_fan (V, E)) ⟶ sum (face (hypermap_of_fan (V, E)) x) (λx::(real, 3) cart × (real, 3) cart. P (fst x)) = sum (face (hypermap_of_fan (V, E)) x) (λx::(real, 3) cart × (real, 3) cart. P (snd x))

thm Crttxat_tame.CRTTXAT_lemma1:

∀(V::(real, 3) cart ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. simple_hypermap (hypermap_of_fan (V, ESTD V)) ⟶ contravening V ∧ IN f (face_set (hypermap_of_fan (V, ESTD V))) ⟶ real_of_nat (12::nat) + (real_of_nat (CARD f) − real_of_nat (CARD V)) ≤ sum f (λx::(real, 3) cart × (real, 3) cart. lmfun (h_dart x))

thm Crttxat_tame.CRTTXAT_lemma2:

∀(V::(real, 3) cart ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. FAN (vec (0::nat), V, ESTD V) ∧ IN f (face_set (hypermap_of_fan (V, ESTD V))) ⟶ sum f (GABS (λf::(real, 3) cart × (real, 3) cart ⇒ real. ∀(v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (arcV (vec (0::nat)) v w))) = sum f (GABS (λf::(real, 3) cart × (real, 3) cart ⇒ real. ∀(v::(real, 3) cart) w::(real, 3) cart. GEQ (f (v, w)) (arclength (vector_norm v) (vector_norm w) (distance (v, w)))))

thm Crttxat_tame.CRTTXAT_lemma1′:

∀(V::(real, 3) cart ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. simple_hypermap (hypermap_of_fan (V, ESTD V)) ⟶ contravening V ∧ IN f (face_set (hypermap_of_fan (V, ESTD V))) ⟶ real_of_nat (12::nat) + (real_of_nat (CARD f) − real_of_nat (CARD V)) ≤ sum f (λx::(real, 3) cart × (real, 3) cart. lmfun (vector_norm (snd x) / real_of_nat (2::nat)))

thm Crttxat_tame.SUM_RMUL_BOUND:

∀(s::?′a::type ⇒ bool) (f::?′a::type ⇒ real) (g::?′a::type ⇒ real) c::real. FINITE s ∧ (∀x::?′a::type. IN x s ⟶ c ≤ g x ∧ (0::real) ≤ f x) ⟶ sum s f ∗ c ≤ sum s (λx::?′a::type. f x ∗ g x)

thm Crttxat_tame.CRTTXAT:

(∀V::(real, 3) cart ⇒ bool. contravening V ⟶ simple_hypermap (hypermap_of_fan (V, ESTD V))) ⟶ (∀V::(real, 3) cart ⇒ bool. contravening V ∧ perimeterbound (V, ESTD V) ⟶ tame_9a (hypermap_of_fan (V, ESTD V)))

thm Hrxefdm_tame.tauVEF_alt1:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. conforming (V, E) ∧ IN f (face_set_of_fan (V, E)) ⟶ tauVEF (V, E, f) = sol (vec (0::nat)) (dartset_leads_into (vec (0::nat), V, E) f) + ((real_of_nat (2::nat) − real_of_nat (CARD f)) ∗ sol0 − sol0 / pi ∗ sum f (λx::(real, 3) cart × (real, 3) cart. azim_dart (V, E) x ∗ (lmfun (h_dart x) − (1::real))))

thm Hrxefdm_tame.tauVEF_alt2:

∀(V::(real, 3) cart ⇒ bool) (E::((real, 3) cart ⇒ bool) ⇒ bool) f::(real, 3) cart × (real, 3) cart ⇒ bool. conforming (V, E) ∧ IN f (face_set_of_fan (V, E)) ⟶ tauVEF (V, E, f) = sol (vec (0::nat)) (dartset_leads_into (vec (0::nat), V, E) f) ∗ ((1::real) + sol0 / pi) − sol0 / pi ∗ sum f (λx::(real, 3) cart × (real, 3) cart. azim_dart (V, E) x ∗ lmfun (h_dart x))

thm Hrxefdm_tame.CHOICE_CONST_LEMMA:

∀(f::?'b::type ⇒ ?'a::type) s::?'b::type ⇒ bool. (∀(x::?'b::type) y::?'b::type. IN x s ∧ IN y s ⟶ f x = f y) ⟶ (∀x::?'b::type. IN x s ⟶ f x = f (CHOICE s))

thm Hrxefdm_tame.scriptL_lemma:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ scriptL V = sum (node_set (hypermap_of_fan (V, E))) (λn::(real, 3) cart × (real, 3) cart ⇒ bool. lmfun (h_dart (CHOICE n)))

thm Hrxefdm_tame.HRXEFDM_lemma1:

∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. FAN (vec (0::nat), V, E) ⟶ sum (face_set_of_fan (V, E)) (λf::(real, 3) cart × (real, 3) cart ⇒ bool. sum f (λx::(real, 3) cart × (real, 3) cart. azim_dart (V, E) x ∗ lmfun (h_dart x))) = real_of_nat (2::nat) ∗ (pi ∗ scriptL V)

thm Hrxefdm_tame.HRXEFDM:

(∀V::(real, 3) cart ⇒ bool. contravening V ⟶ conforming (V, ESTD V)) ∧ (∀(V::(real, 3) cart ⇒ bool) E::((real, 3) cart ⇒ bool) ⇒ bool. sum (face_set_of_fan (V, E)) (λf::(real, 3) cart × (real, 3) cart ⇒ bool. sol (vec (0::nat)) (dartset_leads_into (vec (0::nat), V, E) f)) = real_of_nat (4::nat) ∗ pi) ⟶ (∀V::(real, 3) cart ⇒ bool. contravening V ⟶ sum (face_set_of_fan (V, ESTD V)) (λf::(real, 3) cart × (real, 3) cart ⇒ bool. tauVEF (V, ESTD V, f)) < real_of_nat (4::nat) ∗ pi − real_of_nat (20::nat) ∗ sol0)

thm Pishort.TAYLOR_SIN:

∀(n::nat) x::real. |sin x − sum (dotdot (0::nat) n) (λi::nat. (if i mod (4::nat) = (0::nat) then 0::real else if i mod (4::nat) = (1::nat) then 1::real else if i mod (4::nat) = (2::nat) then 0::real else − (1::real)) ∗ (x^{i} / real_of_nat (fact i)))| ≤ $|x|^{n + (1::nat)}$ / real_of_nat (fact (n + (1::nat)))

thm Pishort.PI_LOWERBOUND_WEAK:

real_of_nat (627::nat) / real_of_nat (256::nat) ≤ pi

thm Pishort.SIN_PI6_STRADDLE:

∀(a::real) b::real. (0::real) ≤ a ∧ a ≤ b ∧ b ≤ real_of_nat (7::nat) ∧ sin (a / real_of_nat (6::nat)) ≤ (1::real) / real_of_nat (2::nat) ∧ (1::real) / real_of_nat (2::nat) ≤ sin (b / real_of_nat (6::nat)) ⟶ a ≤ pi ∧ pi ≤ b

thm Pishort.bound_for_pi:

$\forall$ *n::nat. real_of_nat n* $*$ (*real_of_nat* (*852::nat*) / *real_of_nat* (*1000::nat*)) $\leq$ *real_of_nat* (*2::nat*) $*$ *pi* $\longrightarrow$ *n* $\leq$ (*7::nat*)

thm Deformation.COMPACT_SPHERE_0:

$\forall$ *a::real. compact* (*GSPEC* (λ*GEN%PVAR%2278::*(*real, ?'a::type*) *cart.* $\exists$ *x::*(*real, ?'a::type*) *cart. SETSPEC GEN%PVAR%2278* (*vector_norm x* = *a*) *x*))

thm Deformation.AFF_GE_1_2_0:

$\forall$ (*v::*(*real, ?'a::type*) *cart*) *w::*(*real, ?'a::type*) *cart. v* $\neq$ *vec* (*0::nat*) $\wedge$ *w* $\neq$ *vec* (*0::nat*) $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0::nat*)) *EMPTY*) (*INSERT v* (*INSERT w EMPTY*)) = *GSPEC* (λ*GEN%PVAR%2280::*(*real, ?'a::type*) *cart.* $\exists$ (*a::real*) *b::real. SETSPEC GEN%PVAR%2280* ((*0::real*) $\leq$ *a* $\wedge$ (*0::real*) $\leq$ *b*) (*vector_add* (*% a v*) (*% b w*)))

thm Deformation.AFF_GE_1_1_0:

$\forall$ *v::*(*real, ?'a::type*) *cart. v* $\neq$ *vec* (*0::nat*) $\longrightarrow$ *aff_ge* (*INSERT* (*vec* (*0::nat*)) *EMPTY*) (*INSERT v EMPTY*) = *GSPEC* (λ*GEN%PVAR%2281::*(*real, ?'a::type*) *cart.* $\exists$ *a::real. SETSPEC GEN%PVAR%2281* ((*0::real*) $\leq$ *a*) (*% a v*))

thm Deformation.GMLWKPK:

$\forall$ (*x::*(*real, ?'a::type*) *cart*) (*V::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) *E::*((*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool. graph E* $\longrightarrow$ *fan7* (*x, V, E*) = ($\forall$ (*e1::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) *e2::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool. IN e1* (*HOL_Light_Import.UNION E* (*GSPEC* (λ*GEN%PVAR%2282::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool.* $\exists$ *v::*(*real, ?'a::type*) *cart. SETSPEC GEN%PVAR%2282* (*IN v V*) (*INSERT v EMPTY*)))) $\wedge$ *IN e2* (*HOL_Light_Import.UNION E* (*GSPEC* (λ*GEN%PVAR%2283::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool.* $\exists$ *v::*(*real, ?'a::type*) *cart. SETSPEC GEN%PVAR%2283* (*IN v V*) (*INSERT v EMPTY*)))) $\longrightarrow$ (*HOL_Light_Import.INTER e1 e2* = *EMPTY* $\longrightarrow$ *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) *e1*) (*aff_ge* (*INSERT x EMPTY*) *e2*) = *INSERT x EMPTY*) $\wedge$ ($\forall$ *v::*(*real, ?'a::type*) *cart. HOL_Light_Import.INTER e1 e2* = *INSERT v EMPTY* $\longrightarrow$ *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) *e1*) (*aff_ge* (*INSERT x EMPTY*) *e2*) = *aff_ge* (*INSERT x EMPTY*) (*INSERT v EMPTY*)))

thm Deformation.GMLWKPK_ALT:

$\forall$ (*x::*(*real, ?'a::type*) *cart*) (*V::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) *E::*((*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) $\Rightarrow$ *bool. graph E* $\wedge$ ($\forall$ *e::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool. IN e E* $\longrightarrow$ $\neg$ *IN x e*) $\longrightarrow$ *fan7* (*x, V, E*) = (($\forall$ (*e1::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) *e2::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool. IN e1* (*HOL_Light_Import.UNION E* (*GSPEC* (λ*GEN%PVAR%2284::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool.* $\exists$ *v::*(*real, ?'a::type*) *cart. SETSPEC GEN%PVAR%2284* (*IN v V*) (*INSERT v EMPTY*)))) $\wedge$ *IN e2* (*HOL_Light_Import.UNION E* (*GSPEC* (λ*GEN%PVAR%2285::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool.* $\exists$ *v::*(*real, ?'a::type*) *cart. SETSPEC GEN%PVAR%2285* (*IN v V*) (*INSERT v EMPTY*)))) $\wedge$ *HOL_Light_Import.INTER e1 e2* = *EMPTY* $\longrightarrow$ *HOL_Light_Import.INTER* (*aff_ge* (*INSERT x EMPTY*) *e1*) (*aff_ge* (*INSERT x EMPTY*) *e2*) = *INSERT x EMPTY*) $\wedge$ ($\forall$ (*e1::*(*real, ?'a::type*) *cart* $\Rightarrow$ *bool*) (e2::(real, ?'a::type) cart ⇒ bool) v::(real, ?'a::type) cart. IN e1 E ∧ IN e2 E ∧ HOL_Light_Import.INTER e1 e2 = INSERT v EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = aff_ge (INSERT x EMPTY) (INSERT v EMPTY)))

thm Deformation.GMLWKPK_SIMPLE:

∀ (E::((real, ?'a::type) cart ⇒ bool) ⇒ bool) (V::(real, ?'a::type) cart ⇒ bool) x::(real, ?'a::type) cart. SUBSET (UNIONS E) V ∧ graph E ∧ fan6 (x, V, E) ∧ (∀ e::(real, ?'a::type) cart ⇒ bool. IN e E ⟶ ¬ IN x e) ⟶ fan7 (x, V, E) = (∀ (e1::(real, ?'a::type) cart ⇒ bool) e2::(real, ?'a::type) cart ⇒ bool. IN e1 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2286::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2286 (IN v V) (INSERT v EMPTY)))) ∧ IN e2 (HOL_Light_Import.UNION E (GSPEC (λGEN%PVAR%2287::(real, ?'a::type) cart ⇒ bool. ∃ v::(real, ?'a::type) cart. SETSPEC GEN%PVAR%2287 (IN v V) (INSERT v EMPTY)))) ∧ HOL_Light_Import.INTER e1 e2 = EMPTY ⟶ HOL_Light_Import.INTER (aff_ge (INSERT x EMPTY) e1) (aff_ge (INSERT x EMPTY) e2) = INSERT x EMPTY)

thm Deformation.lemma_1:

∀ (x::(real, ?'a::type) cart) e::real. x ≠ vec (0::nat) ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ x'::(real, ?'a::type) cart. distance (x, x') < d ⟶ (∀ z'::(real, ?'a::type) cart. IN z' (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT x' EMPTY)) ⟶ (∃ z::(real, ?'a::type) cart. IN z (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT x EMPTY)) ∧ vector_norm (vector_sub z' z) ≤ e * vector_norm z)))

thm Deformation.lemma_2:

∀ (x::(real, ?'a::type) cart) (y::(real, ?'a::type) cart) e::real. ¬ collinear (INSERT (vec (0::nat)) (INSERT x (INSERT y EMPTY))) ∧ (0::real) < e ⟶ (∃ d>0::real. ∀ (x'::(real, ?'a::type) cart) y'::(real, ?'a::type) cart. distance (x, x') < d ∧ distance (y, y') < d ⟶ (∀ z'::(real, ?'a::type) cart. IN z' (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT x' (INSERT y' EMPTY))) ⟶ (∃ z::(real, ?'a::type) cart. IN z (aff_ge (INSERT (vec (0::nat)) EMPTY) (INSERT x (INSERT y EMPTY))) ∧ vector_norm (vector_sub z' z) ≤ e * vector_norm z)))

thm Deformation.MINIMIZE_OVER_MEMBERS:

∀ (P::?'a::type ⇒ real ⇒ bool) s::?'a::type ⇒ bool. FINITE s ∧ (∀ x::?'a::type. IN x s ⟶ (∃ e>0::real. ∀ t::real. |t| < e ⟶ P x t)) ⟶ (∃ e>0::real. ∀ t::real. |t| < e ⟶ (∀ x::?'a::type. IN x s ⟶ P x t))

thm Deformation.MINIMIZE_OVER_2:

∀ (P::real ⇒ bool) Q::real ⇒ bool. (∃ e>0::real. ∀ t::real. |t| < e ⟶ P t) ∧ (∃ e>0::real. ∀ t::real. |t| < e ⟶ Q t) ⟶ (∃ e>0::real. ∀ t::real. |t| < e ⟶ P t ∧ Q t)

thm Deformation.MINIMIZE_OVER_STRONGER:

$\forall (P::real \Rightarrow bool)\ Q::real \Rightarrow bool.\ (\forall t::real.\ P\ t \longrightarrow Q\ t) \wedge (\exists e{>}0::real.\ \forall t::real.\ |t| < e \longrightarrow P\ t) \longrightarrow (\exists e{>}0::real.\ \forall t::real.\ |t| < e \longrightarrow Q\ t)$

thm Deformation.deformation:

$\forall (a::real)\ (b::real)\ (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ ff::(real,\ ?'a::type)\ cart \Rightarrow real \Rightarrow (real,\ ?'a::type)\ cart.\ deformation\ ff\ V\ (a,\ b) = (IN\ (0::real)\ (open\_real\_interval\ (a,\ b)) \wedge (\forall (v::(real,\ ?'a::type)\ cart)\ r::real.\ IN\ v\ V \wedge IN\ r\ (open\_real\_interval\ (a,\ b)) \longrightarrow continuous\ (ff\ v)\ (atreal\ r)) \wedge (\forall v::(real,\ ?'a::type)\ cart.\ IN\ v\ V \longrightarrow ff\ v\ (0::real) = v))$

thm Deformation.FAN7_SMALL_DEFORMATION:

$\forall (V::(real,\ ?'a::type)\ cart \Rightarrow bool)\ (E::((real,\ ?'a::type)\ cart \Rightarrow bool) \Rightarrow bool)\ (a::real)\ (b::real)\ phii::(real,\ ?'a::type)\ cart \Rightarrow real \Rightarrow (real,\ ?'a::type)\ cart.\ deformation\ phii\ V\ (a,\ b) \wedge FAN\ (vec\ (0::nat),\ V,\ E) \longrightarrow (\exists e{>}0::real.\ \forall t::real.\ -\ e < t \wedge t < e \longrightarrow fan7\ (vec\ (0::nat),\ IMAGE\ (\lambda v::(real,\ ?'a::type)\ cart.\ phii\ v\ t)\ V,\ IMAGE\ (IMAGE\ (\lambda v::(real,\ ?'a::type)\ cart.\ phii\ v\ t))\ E))$

thm Dont_repeat_yourself.table_multiplier:

$table\_multiplier = real\_of\_nat\ (10000::nat)$

thm Dont_repeat_yourself.a_bn_eq:

$a\_tame * table\_multiplier = real\_of\_nat\ bn\_excessTCount$

thm Dont_repeat_yourself.COND_MUL:

$\forall (a::bool)\ (b::real)\ (c::real)\ d::real.\ (if\ a\ then\ b * d\ else\ c * d) = (if\ a\ then\ b\ else\ c) * d$

thm Dont_repeat_yourself.b_bn_eq:

$\forall (p::nat)\ q::nat.\ b\_tame\ p\ q * table\_multiplier = real\_of\_nat\ (bn\_squanderVertex\ p\ q)$

thm Dont_repeat_yourself.d_bn_eq:

$\forall n::nat.\ d\_tame\ n * table\_multiplier = real\_of\_nat\ (bn\_squanderFace\ n)$