Computational Logic

Vincent van Oostrom
Course/slides by Aart Middeldorp

Department of Computer Science
University of Innsbruck

SS 2020

Outline

- Overviews
- First-Order Logic
- Herbrand Models
- Uniform Notation
- Hintikka's Lemma
- Exercises
- Further Reading

Overview of previous lecture

The previous lecture was concerned with two things, namely (meta-theoretic) consequences of Model Existence and the presentation of Hilbert systems.

- The rough idea of Model Existence is based on maximally deriving consequences of a given set formulas, without arriving at a contradiction. Maximality is captured by the notion of a Hintikka set, which one can think of as expressing that the set is closed under taking consequences, i.e. closed under the tableau expansion rules, and is consistent in that it does not contain a formula and its negation, no contradiction. Hintikka's lemma expresses that such sets are satisfiable (think of satisying a branch of a tableau where we have exhousted all possible consequence without it being closed), and Model Existence that every set in a Propositional Consistency Property, a collection of sets of formulas having certain closure properties, is satisfiable because it can be extended to a Hintikka set (in the collection). From this completeness of tableaux follows, since if never a closed tableau is obtained (from the negation of a formula), a Propositional Consistency Property is obtained, so the negated formula is satisfiable, hence the original formula a tautology.

Overview of previous lecture (ctd)

- Compactness will be commented on below, for first-order logic.
- Interpolation is a meta-theoretic result having uses in model checking and database theory. One can think of it as expressing a restriction on the converse of transitivity: Whereas, transitivity expresses that given X implies Y and Y implies Z, it follows that X implies Z, Craig's interpolation theorem expresses that given X implies Z, one can find an interpolant Y such that X implies Y and Y implies Z, and that for finding Y we may restrict ourselves to formulas speaking about the variables X and Z; i.e. an interpolant/way-point Y can always be found using only variables that are in the common language of X and Z. Versions of interpolation hold e.g. for first-order logic (we will see) and equational logic. There are also logics for which interpolation does not hold, e.g. rewriting logic.

Overview of previous lecture (ctd)

- There is usually an exchange in inference systems, between having many axioms and few inference rules, and having few axioms and many inference rules. Hilbert systems are the extreme case of the former, just the inference rule of modus ponens, whereas natural deduction is an instance of the latter, having introduction and elimination rules for each connective. The deduction theorem connects both by showing that Y can be inferred from X iff there is a Hilbert proof of $X \supset Y$. The proof is constructive, so that to prove the latter, one can first prove the former, and then transform that into a Hilbert proof using the deduction theorem. The Hilbert system presented has only two axioms, but was shown complete using Model Existence, showing the versatility of the latter result.

Overview of this lecture

This lecture is concerned with generalising our set-up for to first-order logic, the default logic in automated reasoning, by allowing to express properties of and relations between individuals (predicates instead of just properties), and all ($\forall$) or some ($\exists$) of these. The items below, on this page, should be known.

- syntax: terms to represent individuals (e.g. natural numbers) and operations on them (e.g. addition), predicates to describe properties of individuals (unary predicates) and relations between them (binary, ternary, . . . predicates), and quantifiers to express properties holding for some/all individuals.
- semantics: meaning of terms is given by means of a domain from which individuals are taken and over which quantifiers range; operations are interpreted as functions on the domain, and predicates as relation on it. The meaning of a concrete formula depends on an assignment giving meaning to the variables in a term, as elements of the domain. This dependence is there to allow interpreting formulas having quantifiers by recursion on the formula. E.g. $(\forall x)\Phi$ is true if for all assignments to x, the subformula Φ is true. The semantics is then used to define the generalised notions of validity and satisfiability.

Overview of this lecture (ctd)

- Herbrand models are models where terms are interpreted as themselves. That is, Herbrand models are a kind of syntactic models; instead of taking as domain say the natural numbers, or people or . . . we take the terms themselves as individuals. For the Herbrand model, assignments are substitutions for the variables. Herbrand models are counterintuitive (aren't interpretations meant to give meaning/semantics to the language/syntax? how can syntax fulfill the role of semantics?) but (cf. free groups) they work: Herbrand models are typically used for meta-theoretic results connecting semantics to syntax, with the reasoning going roughly as follows: if a formula is valid, then it is true in all models, so in particular in the Herbrand model; thus (Herbrand's theorm), if the formula is existential it suffices to find appropriate termsfor the existentially quantified variables. This enables automation. Proof search by enumerating terms is the basis for Prolog (how would one check all interpretations in all domains?).
- Uniform notation is generalised by noting that $\forall$ is a generalised conjunction and $\exists$ a generalised disjunction. For example, one can think of $(\exists n)n \geq 5$ in the natural numbers as $(0 \geq 5) \vee (1 \geq 5) \vee (2 \geq 5) \ldots$.

Overview of this lecture (ctd)

- The meta-theoretic results follow by generalising the propositional case, or rather the cpnverse: Hintikka and Model Existence were set-up for the propositional case such that they would allow easy generalisation to the first-order case. In the propositional case one could often do with finite sets of propositions (e.g. for showing completeness results), but that would not do for the first-order case. An intuition for this insufficiency is provided by the above correspondence between quantified formulas and infinite con/disjunctions.
- Compactness is a meta-theoretic result in that it can be used to show the limitations of first-order logic. In particular, it implies that there is no first-order formula that can express that the domain is finite. From the Löwenheim–Skolem theorem follows a similar limitation of first-order logic: whatever first-order axiomatisation one gives of the real numbers, there will always be a countable model. This means that first-order formulas cannot capture the uncountability of the real numbers.

Part I: Propositional Logic

compactness, completeness, Hilbert systems, Hintikka's lemma, interpolation, logical consequence, model existence theorem, propositional semantic tableaux, soundness

Part II: First-Order Logic

compactness, completeness, Craig's interpolation theorem, cut elimination, first-order semantic tableaux, Herbrand models, Herbrand's theorem, Hilbert systems, Hintikka's lemma, Löwenheim-Skolem, logical consequence, model existence theorem, prenex form, skolemization, soundness

Part III: Limitations and Extensions of First-Order Logic

Curry-Howard isomorphism, intuitionistic logic, Kripke models, second-order logic, simply-typed λ-calculus

Part I: Propositional Logic

compactness, completeness, Hilbert systems, Hintikka's lemma, interpolation, logical consequence, model existence theorem, propositional semantic tableaux, soundness

Part II: First-Order Logic

compactness, completeness, Craig's interpolation theorem, cut elimination, first-order semantic tableaux, Herbrand models, Herbrand's theorem, Hilbert systems, Hintikka's lemma, Löwenheim-Skolem, logical consequence, model existence theorem, prenex form, skolemization, soundness

Part III: Limitations and Extensions of First-Order Logic

Curry-Howard isomorphism, intuitionistic logic, Kripke models, second-order logic, simply-typed λ-calculus

Outline

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$
- variables: $v_1, v_2, \ldots$

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$
- variables: v_1, v_2, $\ldots$

Definition

first-order language is determined by specifying

1. countable set **R** of relation or predicate symbols, each of which has positive integer associated with it

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$
- variables: $v_1, v_2, \ldots$

Definition

first-order language is determined by specifying

1. countable set **R** of relation or predicate symbols, each of which has positive integer associated with it
2. countable set **F** of function symbols, each of which has positive integer associated with it

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$
- variables: $v_1, v_2, \ldots$

Definition

first-order language is determined by specifying

1. countable set **R** of relation or predicate symbols, each of which has positive integer associated with it
2. countable set **F** of function symbols, each of which has positive integer associated with it
3. countable set **C** of constant symbols

First-Order Languages – Common Items

- propositional connectives: primary connectives are basic, secondary connectives are defined, propositional constants $\top$ and $\bot$
- quantifiers: $\forall$ and $\exists$
- variables: $v_1, v_2, \ldots$

Definition

first-order language is determined by specifying

1. countable set **R** of relation or predicate symbols, each of which has positive integer associated with it
2. countable set **F** of function symbols, each of which has positive integer associated with it
3. countable set **C** of constant symbols

notation: $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ (or simply L)

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \ldots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \ldots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

term is closed if it contains no variables

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \ldots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

term is closed if it contains no variables

Example

one-place function symbol f, two-place function symbol g, constants a and b, variables x and y

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \dots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

term is closed if it contains no variables

Example

one-place function symbol f, two-place function symbol g, constants a and b, variables x and y

terms: $f(g(a, x))$

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \dots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

term is closed if it contains no variables

Example

one-place function symbol f, two-place function symbol g, constants a and b, variables x and y

terms: $f(g(a, x))$ $g(f(x), g(x, y))$

Definition

family of terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any variable is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. any constant symbol (member of $\mathbf{C}$) is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
3. if f is n-place function symbol (member of $\mathbf{F}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ then $f(t_1, \dots, t_n)$ is term of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

term is closed if it contains no variables

Example

one-place function symbol f, two-place function symbol g, constants a and b, variables x and y

terms: $f(g(a, x))$ $g(f(x), g(x, y))$ $g(a, g(a, g(a, b)))$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \ldots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \ldots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

also $\top$ and $\bot$ are atomic formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \dots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

also $\top$ and $\bot$ are atomic formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \ldots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \ldots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

also $\top$ and $\bot$ are atomic formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. if A is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ so is $\neg A$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \dots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

also $\top$ and $\bot$ are atomic formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. if A is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ so is $\neg A$
3. if A and B are formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and $\circ$ is binary connective then $A \circ B$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is any string of form $R(t_1, \dots, t_n)$ where R is n-place relation symbol (member of $\mathbf{R}$) and $t_1, \dots, t_n$ are terms of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

also $\top$ and $\bot$ are atomic formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Definition

family of formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is smallest set such that

1. any atomic formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
2. if A is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ so is $\neg A$
3. if A and B are formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and $\circ$ is binary connective then $A \circ B$ is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$
4. if A is formula of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and x is variable then $(\forall x)A$ and $(\exists x)A$ are formulas of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

Example

two-place relation symbol R, two-place function symbol f

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Definition

free-variable occurrences in formula are defined as follows:

1. free-variable occurrences in atomic formula are all variable occurrences in that formula

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Definition

free-variable occurrences in formula are defined as follows:

1. free-variable occurrences in atomic formula are all variable occurrences in that formula
2. free-variable occurrences in $\neg A$ are free-variable occurrences in A

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Definition

free-variable occurrences in formula are defined as follows:

1. free-variable occurrences in atomic formula are all variable occurrences in that formula
2. free-variable occurrences in $\neg A$ are free-variable occurrences in A
3. free-variable occurrences in $(A \circ B)$ are free-variable occurrences in A together with free-variable occurrences in B

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Definition

free-variable occurrences in formula are defined as follows:

1. free-variable occurrences in atomic formula are all variable occurrences in that formula
2. free-variable occurrences in $\neg A$ are free-variable occurrences in A
3. free-variable occurrences in $(A \circ B)$ are free-variable occurrences in A together with free-variable occurrences in B
4. free-variable occurrences in $(\forall x)A$ and $(\exists x)A$ are free-variable occurrences in A, except for occurrences of x

Example

two-place relation symbol R, two-place function symbol f

formulas: $(\forall x)(\forall y)(R(x,y) \supset (\exists z)(R(x,z) \wedge R(z,y)))$ $(\forall x)(\exists y)R(f(x,y),z)$

Definition

free-variable occurrences in formula are defined as follows:

1. free-variable occurrences in atomic formula are all variable occurrences in that formula
2. free-variable occurrences in $\neg A$ are free-variable occurrences in A
3. free-variable occurrences in $(A \circ B)$ are free-variable occurrences in A together with free-variable occurrences in B
4. free-variable occurrences in $(\forall x)A$ and $(\exists x)A$ are free-variable occurrences in A, except for occurrences of x

variable occurrence is **bound** if it is not free

Example

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$

Example

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$ free-variable occurences

Example

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$ free-variable occurences

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$ bound-variable occurences

Example

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$ free-variable occurences

$(\forall x)[(\exists y)R(f(x,y),c) \supset (\exists z)S(y,z)]$ bound-variable occurences

Definition

sentence (or closed formula) of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ is formula of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ with no free-variable occurrences

Outline

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

given substitution σ

1. $c\sigma = c$ for constant symbol c

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

given substitution σ

1. $c\sigma = c$ for constant symbol c
2. $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ for n-place function symbol f

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

given substitution σ

1. $c\sigma = c$ for constant symbol c
2. $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ for n-place function symbol f

Example

$x\sigma = f(x, y)$, $y\sigma = h(a)$, $z\sigma = g(c, h(x))$

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

given substitution σ

1. $c\sigma = c$ for constant symbol c
2. $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ for n-place function symbol f

Example

$x\sigma = f(x, y)$, $y\sigma = h(a)$, $z\sigma = g(c, h(x))$

$j(k(x), y)\sigma = j(k(f(x, y)), h(a))$

Definition

substitution is mapping $\sigma\colon \mathbf{V} \to \mathbf{T}$ from set of variables $\mathbf{V}$ to set of terms $\mathbf{T}$

Definition

given substitution σ

1. $c\sigma = c$ for constant symbol c
2. $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$ for n-place function symbol f

Example

$x\sigma = f(x, y)$, $y\sigma = h(a)$, $z\sigma = g(c, h(x))$
$j(k(x), y)\sigma = j(k(f(x, y)), h(a))$

Definition

composition of substitutions σ and τ is substitution $\sigma\tau$ such that $x(\sigma\tau) = (x\sigma)\tau$ for each variable x

Lemma

$t(\sigma\tau) = (t\sigma)\tau$ *for every term* t

Lemma

$t(\sigma\tau) = (t\sigma)\tau$ *for every term* t

Lemma

composition of substitutions is associative

Lemma

$t(\sigma\tau) = (t\sigma)\tau$ *for every term* t

Lemma

composition of substitutions is associative

Definition

- support of substitution σ is set of variables x for which $x\sigma \neq x$

Lemma

$t(\sigma\tau) = (t\sigma)\tau$ for every term t

Lemma

composition of substitutions is associative

Definition

- support of substitution σ is set of variables x for which $x\sigma \neq x$
- substitution has finite support if support set is finite

Lemma

$t(\sigma\tau) = (t\sigma)\tau$ for every term t

Lemma

composition of substitutions is associative

Definition

- support of substitution σ is set of variables x for which $x\sigma \neq x$
- substitution has finite support if support set is finite

Lemma

composition of substitutions having finite support is substitution having finite support

Notation

$\{x_1/t_1, \ldots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \ldots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Notation

$\{x_1/t_1, \dots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \dots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Notation

$\{x_1/t_1, \dots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \dots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Definition

1. $[A(t_1, \dots, t_n)]\sigma = A(t_1\sigma, \dots, t_n\sigma)$, $\top\sigma = \top$, $\bot\sigma = \bot$

Notation

$\{x_1/t_1, \ldots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \ldots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Definition

1. $[A(t_1, \ldots, t_n)]\sigma = A(t_1\sigma, \ldots, t_n\sigma)$, $\top\sigma = \top$, $\bot\sigma = \bot$
2. $[\neg X]\sigma = \neg[X\sigma]$

Notation

$\{x_1/t_1, \ldots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \ldots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Definition

1. $[A(t_1, \ldots, t_n)]\sigma = A(t_1\sigma, \ldots, t_n\sigma)$, $\top\sigma = \top$, $\bot\sigma = \bot$
2. $[\neg X]\sigma = \neg[X\sigma]$
3. $(X \circ Y)\sigma = (X\sigma \circ Y\sigma)$

Notation

$\{x_1/t_1,\ldots,x_n/t_n\}$ for substitution σ having finite support $\{x_1,\ldots,x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Definition

1. $[A(t_1,\ldots,t_n)]\sigma = A(t_1\sigma,\ldots,t_n\sigma)$, $\top\sigma = \top$, $\bot\sigma = \bot$
2. $[\neg X]\sigma = \neg[X\sigma]$
3. $(X \circ Y)\sigma = (X\sigma \circ Y\sigma)$
4. $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

Notation

$\{x_1/t_1, \dots, x_n/t_n\}$ for substitution σ having finite support $\{x_1, \dots, x_n\}$ and $x_i\sigma = t_i$ for $1 \leqslant i \leqslant n$

Definition

given substitution σ and variable x, substitution σ_x is defined as follows:

$y\sigma_x = x$ if $y = x$ and $y\sigma_x = y\sigma$ if $y \neq x$

Definition

1. $[A(t_1, \dots, t_n)]\sigma = A(t_1\sigma, \dots, t_n\sigma)$, $\top\sigma = \top$, $\bot\sigma = \bot$
2. $[\neg X]\sigma = \neg[X\sigma]$
3. $(X \circ Y)\sigma = (X\sigma \circ Y\sigma)$
4. $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$
5. $[(\exists x)\Phi]\sigma = (\exists x)[\Phi\sigma_x]$

Example

if $\sigma = \{x/a, y/b\}$ then

$$[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma$$

Example

if $\sigma = \{x/a, y/b\}$ then

$$[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma \;=\; [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma$$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned}[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y\end{aligned}$$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned}
[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\
&= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\
&= (\forall x)R(x,b) \supset (\exists y)R(a,y)
\end{aligned}$$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned} [(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\ &= (\forall x)R(x,b) \supset (\exists y)R(a,y) \end{aligned}$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned} [(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\ &= (\forall x)R(x,b) \supset (\exists y)R(a,y) \end{aligned}$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

$\Phi = (\forall y)R(x,y)$, $\sigma = \{x/y\}$, $\tau = \{y/c\}$

Example

if $\sigma = \{x/a, y/b\}$ then

$$
\begin{aligned}
[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\
&= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\
&= (\forall x)R(x,b) \supset (\exists y)R(a,y)
\end{aligned}
$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

$\Phi = (\forall y)R(x,y)$, $\sigma = \{x/y\}$, $\tau = \{y/c\}$

- $(\Phi\sigma)\tau = (\forall y)R(y,y)\tau$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned} [(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\ &= (\forall x)R(x,b) \supset (\exists y)R(a,y) \end{aligned}$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

$\Phi = (\forall y)R(x,y)$, $\sigma = \{x/y\}$, $\tau = \{y/c\}$

- $(\Phi\sigma)\tau = (\forall y)R(y,y)\tau = (\forall y)R(y,y)$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned}[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\ &= (\forall x)R(x,b) \supset (\exists y)R(a,y)\end{aligned}$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

$\Phi = (\forall y)R(x,y)$, $\sigma = \{x/y\}$, $\tau = \{y/c\}$

- $(\Phi\sigma)\tau = (\forall y)R(y,y)\tau = (\forall y)R(y,y)$
- $\sigma\tau = \{x/c, y/c\}$

Example

if $\sigma = \{x/a, y/b\}$ then

$$\begin{aligned}[(\forall x)R(x,y) \supset (\exists y)R(x,y)]\sigma &= [(\forall x)R(x,y)]\sigma \supset [(\exists y)R(x,y)]\sigma \\ &= (\forall x)[R(x,y)]\sigma_x \supset (\exists y)[R(x,y)]\sigma_y \\ &= (\forall x)R(x,b) \supset (\exists y)R(a,y)\end{aligned}$$

Remark

$(\Phi\sigma)\tau = \Phi(\sigma\tau)$ need not hold

Example

$\Phi = (\forall y)R(x,y)$, $\sigma = \{x/y\}$, $\tau = \{y/c\}$

- $(\Phi\sigma)\tau = (\forall y)R(y,y)\tau = (\forall y)R(y,y)$
- $\sigma\tau = \{x/c, y/c\}$
- $\Phi(\sigma\tau) = (\forall y)R(c,y)$

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A
2. σ is free for $\neg X$ if σ is free for X

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A
2. σ is free for $\neg X$ if σ is free for X
3. σ is free for $(X \circ Y)$ if σ is free for X and σ is free for Y

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A
2. σ is free for $\neg X$ if σ is free for X
3. σ is free for $(X \circ Y)$ if σ is free for X and σ is free for Y
4. σ is free for $(\forall x)\Phi$ and $(\exists x)\Phi$ provided
   - σ_x is free for Φ

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A
2. σ is free for $\neg X$ if σ is free for X
3. σ is free for $(X \circ Y)$ if σ is free for X and σ is free for Y
4. σ is free for $(\forall x)\Phi$ and $(\exists x)\Phi$ provided
   - σ_x is free for Φ
   - if y is free variable of Φ other than x, $y\sigma$ does not contain x

Definition

substitution σ being free for formula is defined as follows:

1. if A is atomic then σ free for A
2. σ is free for $\neg X$ if σ is free for X
3. σ is free for $(X \circ Y)$ if σ is free for X and σ is free for Y
4. σ is free for $(\forall x)\Phi$ and $(\exists x)\Phi$ provided
   - σ_x is free for Φ
   - if y is free variable of Φ other than x, $y\sigma$ does not contain x

Theorem

if substitution σ is free for formula X and substitution τ is free for $X\sigma$ then $(X\sigma)\tau = X(\sigma\tau)$

Proof

structural induction on X

- atomic case is obvious

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ σ is free for Y $X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ σ is free for Y $X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad$ σ is free for Y $\quad$ $X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau))$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ σ is free for Y $X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z)$ $\quad \sigma$ is free for Y and Z

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z) \quad \sigma$ is free for Y and $Z \quad X\sigma = (Y\sigma \circ Z\sigma)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z)$ $\quad \sigma$ is free for Y and Z $\quad X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z) \quad \sigma$ is free for Y and $Z \quad X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ and $(Z\sigma)\tau = Z(\sigma\tau)$ follow from induction hypothesis

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ $\quad \sigma$ is free for Y $\quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z)$ $\quad \sigma$ is free for Y and Z $\quad X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ and $(Z\sigma)\tau = Z(\sigma\tau)$ follow from induction hypothesis

  $(X\sigma)\tau = ((Y\sigma)\tau \circ (Z\sigma)\tau)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y$ σ is free for Y $X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z)$ σ is free for Y and Z $X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ and $(Z\sigma)\tau = Z(\sigma\tau)$ follow from induction hypothesis

  $(X\sigma)\tau = ((Y\sigma)\tau \circ (Z\sigma)\tau) = (Y(\sigma\tau) \circ Z(\sigma\tau))$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z) \quad \sigma$ is free for Y and $Z \quad X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ and $(Z\sigma)\tau = Z(\sigma\tau)$ follow from induction hypothesis

  $(X\sigma)\tau = ((Y\sigma)\tau \circ (Z\sigma)\tau) = (Y(\sigma\tau) \circ Z(\sigma\tau)) = (Y \circ Z)(\sigma\tau)$

Proof

structural induction on X

- atomic case is obvious
- $X = \neg Y \quad \sigma$ is free for $Y \quad X\sigma = \neg(Y\sigma)$

  τ is free for $Y\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ follows from induction hypothesis

  $(X\sigma)\tau = [\neg(Y\sigma)]\tau = \neg((Y\sigma)\tau) = \neg(Y(\sigma\tau)) = (\neg Y)(\sigma\tau) = X(\sigma\tau)$
- $X = (Y \circ Z) \quad \sigma$ is free for Y and $Z \quad X\sigma = (Y\sigma \circ Z\sigma)$

  τ is free for $Y\sigma$ and $Z\sigma$

  $(Y\sigma)\tau = Y(\sigma\tau)$ and $(Z\sigma)\tau = Z(\sigma\tau)$ follow from induction hypothesis

  $(X\sigma)\tau = ((Y\sigma)\tau \circ (Z\sigma)\tau) = (Y(\sigma\tau) \circ Z(\sigma\tau)) = (Y \circ Z)(\sigma\tau) = X(\sigma\tau)$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for Φ

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$
  - if $y \neq x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x$
  - if $y \neq x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x$
  - if $y \neq x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x$
  - if $y \neq x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau)$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$(X\sigma)\tau = ((\forall x)[\Phi\sigma_x])\tau$$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$(X\sigma)\tau = ((\forall x)[\Phi\sigma_x])\tau = (\forall x)[(\Phi\sigma_x)\tau_x]$$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$(X\sigma)\tau = ((\forall x)[\Phi\sigma_x])\tau = (\forall x)[(\Phi\sigma_x)\tau_x] = (\forall x)[\Phi(\sigma_x\tau_x)]$$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$\begin{aligned}(X\sigma)\tau &= ((\forall x)[\Phi\sigma_x])\tau = (\forall x)[(\Phi\sigma_x)\tau_x] = (\forall x)[\Phi(\sigma_x\tau_x)] \\ &= (\forall x)[\Phi(\sigma\tau)_x]\end{aligned}$$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi \quad \sigma_x$ is free for $\Phi \quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$\begin{aligned}(X\sigma)\tau &= ((\forall x)[\Phi\sigma_x])\tau = (\forall x)[(\Phi\sigma_x)\tau_x] = (\forall x)[\Phi(\sigma_x\tau_x)] \\ &= (\forall x)[\Phi(\sigma\tau)_x] = X(\sigma\tau)\end{aligned}$$

Proof (cont'd)

structural induction on X

- $X = (\forall x)\Phi$ $\quad \sigma_x$ is free for Φ $\quad \tau$ is free for $[(\forall x)\Phi]\sigma = (\forall x)[\Phi\sigma_x]$

  τ_x is free for $\Phi\sigma_x$

  $(\Phi\sigma_x)\tau_x = \Phi(\sigma_x\tau_x)$ follows from induction hypothesis

  claim: $\Phi(\sigma_x\tau_x) = \Phi(\sigma\tau)_x$

  let y be free variable of Φ

  - if $y = x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = x\tau_x = x = y(\sigma\tau)_x$
  - if $y \neq x$ then $y(\sigma_x\tau_x) = (y\sigma_x)\tau_x = (y\sigma)\tau_x = (y\sigma)\tau = y(\sigma\tau) = y(\sigma\tau)_x$

$$\begin{aligned}(X\sigma)\tau = ((\forall x)[\Phi\sigma_x])\tau = (\forall x)[(\Phi\sigma_x)\tau_x] = (\forall x)[\Phi(\sigma_x\tau_x)]\\ = (\forall x)[\Phi(\sigma\tau)_x] = X(\sigma\tau)\end{aligned}$$

- $X = (\exists x)\Phi$ $\quad$ similar

Outline

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation, that associates
  - to every $c \in \mathbf{C}$ some member $c^{\mathbf{I}} \in \mathbf{D}$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation, that associates
  - to every $c \in \mathbf{C}$ some member $c^{\mathbf{I}} \in \mathbf{D}$
  - to every n-place $f \in \mathbf{F}$ some n-ary function $f^{\mathbf{I}} \colon \mathbf{D}^n \to \mathbf{D}$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation, that associates
  - to every $c \in \mathbf{C}$ some member $c^{\mathbf{I}} \in \mathbf{D}$
  - to every n-place $f \in \mathbf{F}$ some n-ary function $f^{\mathbf{I}} \colon \mathbf{D}^n \to \mathbf{D}$
  - to every n-place $P \in \mathbf{R}$ some n-ary relation $P^{\mathbf{I}} \subseteq \mathbf{D}^n$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation, that associates
  - to every $c \in \mathbf{C}$ some member $c^{\mathbf{I}} \in \mathbf{D}$
  - to every n-place $f \in \mathbf{F}$ some n-ary function $f^{\mathbf{I}} \colon \mathbf{D}^n \to \mathbf{D}$
  - to every n-place $P \in \mathbf{R}$ some n-ary relation $P^{\mathbf{I}} \subseteq \mathbf{D}^n$

Definition

assignment in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is mapping $\mathbf{A}$ from set of variables to set $\mathbf{D}$

Definition

model for first-order language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is pair $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ where

- $\mathbf{D}$ is nonempty set, called domain of $\mathbf{M}$
- $\mathbf{I}$ is mapping, called interpretation, that associates
  - to every $c \in \mathbf{C}$ some member $c^{\mathbf{I}} \in \mathbf{D}$
  - to every n-place $f \in \mathbf{F}$ some n-ary function $f^{\mathbf{I}} \colon \mathbf{D}^n \to \mathbf{D}$
  - to every n-place $P \in \mathbf{R}$ some n-ary relation $P^{\mathbf{I}} \subseteq \mathbf{D}^n$

Definition

assignment in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is mapping $\mathbf{A}$ from set of variables to set $\mathbf{D}$

image of variable v under assignment $\mathbf{A}$ is denoted by $v^{\mathbf{A}}$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_1 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $0^{\mathbf{I}} = 0$, $s^{\mathbf{I}}(n) = n + 1$, $+^{\mathbf{I}}(m, n) = m + n$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_1 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $0^{\mathbf{I}} = 0$, $s^{\mathbf{I}}(n) = n + 1$, $+^{\mathbf{I}}(m, n) = m + n$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = 3$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_1 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $0^{\mathbf{I}} = 0$, $s^{\mathbf{I}}(n) = n + 1$, $+^{\mathbf{I}}(m, n) = m + n$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = 3$

$t_1^{\mathbf{I},\mathbf{A}} = 6$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_1 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $0^{\mathbf{I}} = 0$, $s^{\mathbf{I}}(n) = n + 1$, $+^{\mathbf{I}}(m, n) = m + n$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = 3$

$t_1^{\mathbf{I},\mathbf{A}} = 6$ and $t_2^{\mathbf{I},\mathbf{A}} =$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_1 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $0^{\mathbf{I}} = 0$, $s^{\mathbf{I}}(n) = n + 1$, $+^{\mathbf{I}}(m, n) = m + n$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = 3$

$t_1^{\mathbf{I},\mathbf{A}} = 6$ and $t_2^{\mathbf{I},\mathbf{A}} = 9$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_2 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \{a, b\}^*$, $0^{\mathbf{I}} = a$, $s^{\mathbf{I}}(w) = wa$, $+^{\mathbf{I}}(v, w) = vw$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_2 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \{a, b\}^*$, $0^{\mathbf{I}} = a$, $s^{\mathbf{I}}(w) = wa$, $+^{\mathbf{I}}(v, w) = vw$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = aba$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_2 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \{a, b\}^*$, $0^{\mathbf{I}} = a$, $s^{\mathbf{I}}(w) = wa$, $+^{\mathbf{I}}(v, w) = vw$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = aba$

$t_1^{\mathbf{I},\mathbf{A}} =$ and $t_2^{\mathbf{I},\mathbf{A}} =$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_2 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \{a, b\}^*$, $0^{\mathbf{I}} = a$, $s^{\mathbf{I}}(w) = wa$, $+^{\mathbf{I}}(v, w) = vw$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = aba$

$t_1^{\mathbf{I},\mathbf{A}} = aaabaaa$ and $t_2^{\mathbf{I},\mathbf{A}} =$

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment $\mathbf{A}$ in $\mathbf{M}$, value $t^{\mathbf{I},\mathbf{A}}$ in $\mathbf{D}$ is defined inductively:

1. $c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$ for constant symbol c
2. $v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$ for variable v
3. $[f(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$ for n-place function symbol f

Example

L with constant 0, one-place function symbol s, two-place function symbol $+$

terms $t_1 = s(s(0) + s(x))$ and $t_2 = s(x + s(x + s(0)))$

model $\mathbf{M}_2 = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \{a, b\}^*$, $0^{\mathbf{I}} = a$, $s^{\mathbf{I}}(w) = wa$, $+^{\mathbf{I}}(v, w) = vw$

assignment $\mathbf{A}$ with $x^{\mathbf{A}} = aba$

$t_1^{\mathbf{I},\mathbf{A}} = aaabaaa$ and $t_2^{\mathbf{I},\mathbf{A}} = abaabaaaaa$

Definition

assignment **B** in model **M** is *x-variant* of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

assignment **B** in model **M** is *x*-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly *x*

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$

Definition

assignment **B** in model **M** is x-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$ $\quad \top^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ $\quad \bot^{\mathbf{I},\mathbf{A}} = \mathrm{f}$

Definition

assignment **B** in model **M** is x-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$ $\quad \top^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ $\quad \bot^{\mathbf{I},\mathbf{A}} = \mathrm{f}$
2. $[\neg X]^{\mathbf{I},\mathbf{A}} = \neg[X^{\mathbf{I},\mathbf{A}}]$

Definition

assignment **B** in model **M** is x-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$ $\quad \top^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ $\quad \bot^{\mathbf{I},\mathbf{A}} = \mathrm{f}$
2. $[\neg X]^{\mathbf{I},\mathbf{A}} = \neg[X^{\mathbf{I},\mathbf{A}}]$
3. $[X \circ Y]^{\mathbf{I},\mathbf{A}} = X^{\mathbf{I},\mathbf{A}} \circ Y^{\mathbf{I},\mathbf{A}}$

Definition

assignment **B** in model **M** is x-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$ $\quad \top^{\mathbf{I},\mathbf{A}} = \mathrm{t} \quad \bot^{\mathbf{I},\mathbf{A}} = \mathrm{f}$
2. $[\neg X]^{\mathbf{I},\mathbf{A}} = \neg[X^{\mathbf{I},\mathbf{A}}]$
3. $[X \circ Y]^{\mathbf{I},\mathbf{A}} = X^{\mathbf{I},\mathbf{A}} \circ Y^{\mathbf{I},\mathbf{A}}$
4. $[(\forall x)\Phi]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \Phi^{\mathbf{I},\mathbf{B}} = \mathrm{t}$ for every assignment **B** in **M** that is x-variant of **A**

Definition

assignment **B** in model **M** is x-variant of assignment **A** provided **A** and **B** assign same values to every variable except possibly x

Definition

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ and assignment **A** in **M**, truth value $\Phi^{\mathbf{I},\mathbf{A}}$ for formula Φ of $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ is defined inductively:

1. $[P(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle \in P^{\mathbf{I}}$ $\quad \top^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ $\quad \bot^{\mathbf{I},\mathbf{A}} = \mathrm{f}$
2. $[\neg X]^{\mathbf{I},\mathbf{A}} = \neg[X^{\mathbf{I},\mathbf{A}}]$
3. $[X \circ Y]^{\mathbf{I},\mathbf{A}} = X^{\mathbf{I},\mathbf{A}} \circ Y^{\mathbf{I},\mathbf{A}}$
4. $[(\forall x)\Phi]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \Phi^{\mathbf{I},\mathbf{B}} = \mathrm{t}$ for every assignment **B** in **M** that is x-variant of **A**
5. $[(\exists x)\Phi]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \iff \Phi^{\mathbf{I},\mathbf{B}} = \mathrm{t}$ for some assignment **B** in **M** that is x-variant of **A**

Notation

$\Phi^{\mathbf{I}}$ instead of $\Phi^{\mathbf{I},\mathbf{A}}$ for formulas Φ without free variables

Definitions

- formula Φ of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ is true in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for $L(\mathbf{R},\mathbf{F},\mathbf{C})$ provided $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all assignments $\mathbf{A}$

Notation

$\Phi^{\mathbf{I}}$ instead of $\Phi^{\mathbf{I},\mathbf{A}}$ for formulas Φ without free variables

Definitions

- formula Φ of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ is true in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for $L(\mathbf{R},\mathbf{F},\mathbf{C})$ provided $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all assignments $\mathbf{A}$
- formula Φ is valid if Φ is true in all models for $L(\mathbf{R},\mathbf{F},\mathbf{C})$

Notation

$\Phi^{\mathbf{I}}$ instead of $\Phi^{\mathbf{I},\mathbf{A}}$ for formulas Φ without free variables

Definitions

- formula Φ of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ is true in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for $L(\mathbf{R},\mathbf{F},\mathbf{C})$ provided $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all assignments $\mathbf{A}$
- formula Φ is valid if Φ is true in all models for $L(\mathbf{R},\mathbf{F},\mathbf{C})$
- set S of formulas is satisfiable in $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ provided there exists assignment $\mathbf{A}$ (called satisfying assignment) such that $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all $\Phi \in S$

Notation

$\Phi^{\mathbf{I}}$ instead of $\Phi^{\mathbf{I},\mathbf{A}}$ for formulas Φ without free variables

Definitions

- formula Φ of $L(\mathbf{R},\mathbf{F},\mathbf{C})$ is true in model $\mathbf{M} = \langle \mathbf{D},\mathbf{I} \rangle$ for $L(\mathbf{R},\mathbf{F},\mathbf{C})$ provided $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all assignments $\mathbf{A}$
- formula Φ is valid if Φ is true in all models for $L(\mathbf{R},\mathbf{F},\mathbf{C})$
- set S of formulas is satisfiable in $\mathbf{M} = \langle \mathbf{D},\mathbf{I} \rangle$ provided there exists assignment $\mathbf{A}$ (called satisfying assignment) such that $\Phi^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ for all $\Phi \in S$
- set S of formulas is satisfiable if S is satisfiable in some model

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y) R(x, y \oplus y)$

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y) R(x, y \oplus y)$

  $(\exists y) R(x, y \oplus y)^{\mathbf{I},\mathbf{A}}$ is true if and only if $x^{\mathbf{A}}$ is even number

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y)R(x, y \oplus y)$

  $(\exists y)R(x, y \oplus y)^{\mathbf{I},\mathbf{A}}$ is true if and only if $x^{\mathbf{A}}$ is even number

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}_{+}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is greater-than relation

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y)R(x, y \oplus y)$

  $(\exists y)R(x, y \oplus y)^{\mathbf{I},\mathbf{A}}$ is true if and only if $x^{\mathbf{A}}$ is even number

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}_+$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is greater-than relation

  sentence $(\forall x)(\forall y)(\exists z)R(x \oplus y, z)$

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y)R(x, y \oplus y)$

  $(\exists y)R(x, y \oplus y)^{\mathbf{I},\mathbf{A}}$ is true if and only if $x^{\mathbf{A}}$ is even number

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}_+$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is greater-than relation

  sentence $(\forall x)(\forall y)(\exists z)R(x \oplus y, z)$ is true in $\mathbf{M}$

Example

L with two-place function symbol $\oplus$ and two-place relation symbol R

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is equality relation

  formula $(\exists y)R(x, y \oplus y)$

  $(\exists y)R(x, y \oplus y)^{\mathbf{I},\mathbf{A}}$ is true if and only if $x^{\mathbf{A}}$ is even number

- model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ with $\mathbf{D} = \mathbb{N}_+$, $\oplus^{\mathbf{I}}(x, y) = x + y$, $R^{\mathbf{I}}$ is greater-than relation

  sentence $(\forall x)(\forall y)(\exists z)R(x \oplus y, z)$ is true in $\mathbf{M}$

Lemma

given closed term t, formula Φ of first-order language L, model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for L

if x is variable and $\mathbf{A}$ assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$ then $[\Phi\{x/t\}]^{\mathbf{I},\mathbf{B}} = \Phi^{\mathbf{I},\mathbf{A}}$ for any x-variant $\mathbf{B}$ of $\mathbf{A}$

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L, *formula* Φ *in* L, *assignment* $\mathbf{A}$ *in* $\mathbf{M}$, *substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$ suppose $(\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$ suppose $(\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$

  $(\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathrm{t}$ for some x-variant $\mathbf{A}'$ of $\mathbf{A}$

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$ suppose $(\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$

  $(\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathrm{t}$ for some x-variant $\mathbf{A}'$ of $\mathbf{A}$

  define assignment $\mathbf{B}'$ by $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$ suppose $(\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$

  $(\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$ for some x-variant $\mathbf{A}'$ of $\mathbf{A}$

  define assignment $\mathbf{B}'$ by $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  σ_x is free for φ

Lemma

given model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *for language* L*, formula* Φ *in* L*, assignment* $\mathbf{A}$ *in* $\mathbf{M}$*, substitution* σ *that is free for* Φ

if assignment $\mathbf{B}$ *is defined by* $v^{\mathbf{B}} = (v\sigma)^{\mathbf{I},\mathbf{A}}$ *for each variable* v *then* $\Phi^{\mathbf{I},\mathbf{B}} = (\Phi\sigma)^{\mathbf{I},\mathbf{A}}$

Proof

structural induction on Φ

- atomic and propositional cases are straightforward

  $t^{\mathbf{I},\mathbf{B}} = (t\sigma)^{\mathbf{I},\mathbf{A}}$ for all terms t is obtained by induction on t

- $\Phi = (\exists x)\varphi$ suppose $(\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$

  $(\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathrm{t}$ for some x-variant $\mathbf{A}'$ of $\mathbf{A}$

  define assignment $\mathbf{B}'$ by $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  σ_x is free for φ

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathrm{t}$ by induction hypothesis

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathrm{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}} = v^{\mathbf{B}}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}} = v^{\mathbf{B}}$

  $\Phi^{\mathbf{I},\mathbf{B}} = [(\exists x)\varphi]^{\mathbf{I},\mathbf{B}}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}} = v^{\mathbf{B}}$

  $\Phi^{\mathbf{I},\mathbf{B}} = [(\exists x)\varphi]^{\mathbf{I},\mathbf{B}} = \varphi^{\mathbf{I},\mathbf{B}'} = \mathsf{t}$

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}} = v^{\mathbf{B}}$

  $\Phi^{\mathbf{I},\mathbf{B}} = [(\exists x)\varphi]^{\mathbf{I},\mathbf{B}} = \varphi^{\mathbf{I},\mathbf{B}'} = \mathsf{t}$

  proof of converse direction is similar

Proof (cont'd)

- $\Phi = (\exists x)\varphi \quad (\Phi\sigma)^{\mathbf{I},\mathbf{A}} = [(\exists x)(\varphi\sigma_x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t} \quad \sigma$ is free for Φ

  assignment $\mathbf{B}'$ with $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'}$ for each variable v

  $\varphi^{\mathbf{I},\mathbf{B}'} = (\varphi\sigma_x)^{\mathbf{I},\mathbf{A}'} = \mathsf{t}$

  claim: $\mathbf{B}'$ is x-variant $\mathbf{B}$

  if $v \neq x$ then $v^{\mathbf{B}'} = (v\sigma_x)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}'} = (v\sigma)^{\mathbf{I},\mathbf{A}} = v^{\mathbf{B}}$

  $\Phi^{\mathbf{I},\mathbf{B}} = [(\exists x)\varphi]^{\mathbf{I},\mathbf{B}} = \varphi^{\mathbf{I},\mathbf{B}'} = \mathsf{t}$

  proof of converse direction is similar

- $\Phi = (\forall x)\varphi \quad$ similar

Outline

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is Herbrand model for L then $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ for each term t of L

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is Herbrand model for L then $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ for each term t of L

Proof

structural induction on t

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is Herbrand model for L then $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ for each term t of L

Proof

structural induction on t

- if t is variable v then $t^{\mathbf{I},\mathbf{A}} = v^{\mathbf{I},\mathbf{A}}$

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is Herbrand model for L then $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ for each term t of L

Proof

structural induction on t

- if t is variable v then $t^{\mathbf{I},\mathbf{A}} = v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}}$

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then* $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ *for each term* t *of* L

Proof

structural induction on t

- if t is variable v then $t^{\mathbf{I},\mathbf{A}} = v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}} = v\mathbf{A}$

Definition

model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ for language L is Herbrand model if

1. $\mathbf{D}$ is set of closed terms of L (which is assumed to be nonempty)
2. $t^{\mathbf{I}} = t$ for each closed term t

Remark

assigments in Herbrand model are substitutions

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ is Herbrand model for L then $t^{\mathbf{I},\mathbf{A}} = (t\mathbf{A})^{\mathbf{I}}$ for each term t of L

Proof

structural induction on t

- if t is variable v then $t^{\mathbf{I},\mathbf{A}} = v^{\mathbf{I},\mathbf{A}} = v^{\mathbf{A}} = v\mathbf{A} = (v\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}}$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}}$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \dots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}})$$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \ldots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}}) = f^{\mathbf{I}}((t_1\mathbf{A})^{\mathbf{I}}, \ldots, (t_n\mathbf{A})^{\mathbf{I}})$$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \ldots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}}) = f^{\mathbf{I}}((t_1\mathbf{A})^{\mathbf{I}}, \ldots, (t_n\mathbf{A})^{\mathbf{I}}) = [f(t_1\mathbf{A}, \ldots, t_n\mathbf{A})]^{\mathbf{I}}$$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \ldots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}}) = f^{\mathbf{I}}((t_1\mathbf{A})^{\mathbf{I}}, \ldots, (t_n\mathbf{A})^{\mathbf{I}}) = [f(t_1\mathbf{A}, \ldots, t_n\mathbf{A})]^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$$

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \ldots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}}) = f^{\mathbf{I}}((t_1\mathbf{A})^{\mathbf{I}}, \ldots, (t_n\mathbf{A})^{\mathbf{I}}) = [f(t_1\mathbf{A}, \ldots, t_n\mathbf{A})]^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$$

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then* $\Phi^{\mathbf{I},\mathbf{A}} = (\Phi\mathbf{A})^{\mathbf{I}}$ *for each formula* Φ *of* L

Proof (cont'd)

structural induction on t

- if t is constant symbol c of L then $t^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I},\mathbf{A}} = c^{\mathbf{I}} = (c\mathbf{A})^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$
- if $t = f(t_1, \dots, t_n)$ then

$$t^{\mathbf{I},\mathbf{A}} = f^{\mathbf{I}}(t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}}) = f^{\mathbf{I}}((t_1\mathbf{A})^{\mathbf{I}}, \dots, (t_n\mathbf{A})^{\mathbf{I}}) = [f(t_1\mathbf{A}, \dots, t_n\mathbf{A})]^{\mathbf{I}} = (t\mathbf{A})^{\mathbf{I}}$$

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then* $\Phi^{\mathbf{I},\mathbf{A}} = (\Phi\mathbf{A})^{\mathbf{I}}$ *for each formula* Φ *of* L

Proof

. . . exercise . . .

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then*

1. $(\forall x)\Phi$ *is true in* $\mathbf{M} \iff \Phi\{x/d\}$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then*

1. $(\forall x)\Phi$ *is true in* $\mathbf{M} \iff \Phi\{x/d\}$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$
2. $(\exists x)\Phi$ *is true in* $\mathbf{M} \iff \Phi\{x/d\}$ *is true in* $\mathbf{M}$ *for some* $d \in \mathbf{D}$

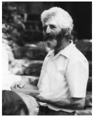
Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then*

1. $(\forall x)\Phi$ *is true in* $\mathbf{M} \iff \Phi\{x/d\}$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$
2. $(\exists x)\Phi$ *is true in* $\mathbf{M} \iff \Phi\{x/d\}$ *is true in* $\mathbf{M}$ *for some* $d \in \mathbf{D}$

Proof

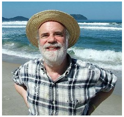
... exercise ...

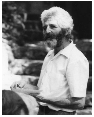


William Craig
(1918 – 2016)

David Hilbert
(1862 – 1943)

Jaakko Hintikka
(1929 – 2015)



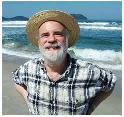


William Craig
(1918 – 2016)

Jacques Herbrand
(1908 – 1931)

David Hilbert
(1862 – 1943)

Jaakko Hintikka
(1929 – 2015)



Outline

Definition

universal		existential	
γ		δ	
$(\forall x)\Phi$		$(\exists x)\Phi$	
$\neg(\exists x)\Phi$		$\neg(\forall x)\Phi$	

Definition

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$(\forall x)\Phi$	$\Phi\{x/t\}$	$(\exists x)\Phi$	$\Phi\{x/t\}$
$\neg(\exists x)\Phi$	$\neg\Phi\{x/t\}$	$\neg(\forall x)\Phi$	$\neg\Phi\{x/t\}$

Definition

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$(\forall x)\Phi$	$\Phi\{x/t\}$	$(\exists x)\Phi$	$\Phi\{x/t\}$
$\neg(\exists x)\Phi$	$\neg\Phi\{x/t\}$	$\neg(\forall x)\Phi$	$\neg\Phi\{x/t\}$

Lemma

$\gamma \equiv (\forall y)\gamma(y)$ *and* $\delta \equiv (\exists y)\delta(y)$ *are valid, provided* y *is variable new to* γ *and* δ

Definition

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$(\forall x)\Phi$	$\Phi\{x/t\}$	$(\exists x)\Phi$	$\Phi\{x/t\}$
$\neg(\exists x)\Phi$	$\neg\Phi\{x/t\}$	$\neg(\forall x)\Phi$	$\neg\Phi\{x/t\}$

Lemma

$\gamma \equiv (\forall y)\gamma(y)$ *and* $\delta \equiv (\exists y)\delta(y)$ *are valid, provided* y *is variable new to* γ *and* δ

Lemma

set S *of sentences, sentences* γ *and* δ

1. *if* $S \cup \{\gamma\}$ *is satisfiable then* $S \cup \{\gamma, \gamma(t)\}$ *is satisfiable for any closed term* t

Definition

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$(\forall x)\Phi$	$\Phi\{x/t\}$	$(\exists x)\Phi$	$\Phi\{x/t\}$
$\neg(\exists x)\Phi$	$\neg\Phi\{x/t\}$	$\neg(\forall x)\Phi$	$\neg\Phi\{x/t\}$

Lemma

$\gamma \equiv (\forall y)\gamma(y)$ *and* $\delta \equiv (\exists y)\delta(y)$ *are valid, provided* y *is variable new to* γ *and* δ

Lemma

set S *of sentences, sentences* γ *and* δ

1. *if* $S \cup \{\gamma\}$ *is satisfiable then* $S \cup \{\gamma, \gamma(t)\}$ *is satisfiable for any closed term* t
2. *if* $S \cup \{\delta\}$ *is satisfiable then* $S \cup \{\delta, \delta(p)\}$ *is satisfiable for any constant symbol* p *that is new to* S *and* δ

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}}$ (using Lemma on slide 43)

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

   construct new model $\mathbf{M}^* = \langle \mathbf{D}, \mathbf{J} \rangle$ with $\mathbf{J}$ identical to $\mathbf{I}$ except $p^{\mathbf{J}} = x^{\mathbf{A}}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

   construct new model $\mathbf{M}^* = \langle \mathbf{D}, \mathbf{J} \rangle$ with $\mathbf{J}$ identical to $\mathbf{I}$ except $p^{\mathbf{J}} = x^{\mathbf{A}}$

   $S \cup \{\delta\}$ is satisfiable in $\mathbf{M}^*$ and $[\delta(x)]^{\mathbf{J},\mathbf{A}}$ is true

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

   construct new model $\mathbf{M}^* = \langle \mathbf{D}, \mathbf{J} \rangle$ with $\mathbf{J}$ identical to $\mathbf{I}$ except $p^{\mathbf{J}} = x^{\mathbf{A}}$

   $S \cup \{\delta\}$ is satisfiable in $\mathbf{M}^*$ and $[\delta(x)]^{\mathbf{J},\mathbf{A}}$ is true

   $[\delta(p)]^{\mathbf{J},\mathbf{A}} = [\delta\{x/p\}]^{\mathbf{J},\mathbf{A}}$

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

   construct new model $\mathbf{M}^* = \langle \mathbf{D}, \mathbf{J} \rangle$ with $\mathbf{J}$ identical to $\mathbf{I}$ except $p^{\mathbf{J}} = x^{\mathbf{A}}$

   $S \cup \{\delta\}$ is satisfiable in $\mathbf{M}^*$ and $[\delta(x)]^{\mathbf{J},\mathbf{A}}$ is true

   $[\delta(p)]^{\mathbf{J},\mathbf{A}} = [\delta\{x/p\}]^{\mathbf{J},\mathbf{A}} = [\delta(x)]^{\mathbf{J},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

Proof

1. suppose $S \cup \{\gamma\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\forall x)\gamma(x)$ is true in $\mathbf{M}$ (with x new to γ)

   $[\gamma(x)]^{\mathbf{I},\mathbf{A}}$ is true for every assignment $\mathbf{A}$

   let $\mathbf{A}$ be any assignment such that $x^{\mathbf{A}} = t^{\mathbf{I}}$

   $[\gamma(t)]^{\mathbf{I},\mathbf{A}} = [\gamma\{x/t\}]^{\mathbf{I},\mathbf{A}} = [\gamma(x)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

2. suppose $S \cup \{\delta\}$ is satisfiable in model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

   $(\exists x)\delta(x)$ is true in $\mathbf{M}$ (with x new to δ)

   $[\delta(x)]^{\mathbf{I},\mathbf{A}}$ is true for some assignment $\mathbf{A}$

   construct new model $\mathbf{M}^* = \langle \mathbf{D}, \mathbf{J} \rangle$ with $\mathbf{J}$ identical to $\mathbf{I}$ except $p^{\mathbf{J}} = x^{\mathbf{A}}$

   $S \cup \{\delta\}$ is satisfiable in $\mathbf{M}^*$ and $[\delta(x)]^{\mathbf{J},\mathbf{A}}$ is true

   $[\delta(p)]^{\mathbf{J},\mathbf{A}} = [\delta\{x/p\}]^{\mathbf{J},\mathbf{A}} = [\delta(x)]^{\mathbf{J},\mathbf{A}} = \mathrm{t}$ (using Lemma on slide 43)

   $S \cup \{\delta, \delta(p)\}$ is satisfiable (in $\mathbf{M}^*$)

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X has property Q then $\neg\neg X$ has property Q

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X has property Q then $\neg\neg X$ has property Q

  if α_1 and α_2 have property Q then α has property Q

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X has property Q then $\neg\neg X$ has property Q

  if α_1 and α_2 have property Q then α has property Q

  if β_1 and β_2 have property Q then β has property Q

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X has property Q then $\neg\neg X$ has property Q

  if α_1 and α_2 have property Q then α has property Q

  if β_1 and β_2 have property Q then β has property Q

  if $\gamma(t)$ has property Q for each term t then γ has property Q

Definition

rank $r(X)$ of first-order formula: $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$

$$r(\neg\top) = r(\neg\bot) = 1 \qquad r(\neg\neg Z) = r(Z) + 1 \qquad r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$$

$$r(\beta) = r(\beta_1) + r(\beta_2) + 1 \qquad r(\gamma) = r(\gamma(x)) + 1 \qquad r(\delta) = r(\delta(x)) + 1$$

Theorem (First-Order Structural Induction)

every formula of first-order language L has property Q provided

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X has property Q then $\neg\neg X$ has property Q

  if α_1 and α_2 have property Q then α has property Q

  if β_1 and β_2 have property Q then β has property Q

  if $\gamma(t)$ has property Q for each term t then γ has property Q

  if $\delta(t)$ has property Q for each term t then δ has property Q

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then*

- *formula* γ *of* L *is true in* $\mathbf{M} \iff \gamma(d)$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for L then*

- *formula* γ *of L is true in* $\mathbf{M} \iff \gamma(d)$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$
- *formula* δ *of L is true in* $\mathbf{M} \iff \gamma(d)$ *is true in* $\mathbf{M}$ *for some* $d \in \mathbf{D}$

Lemma

if $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$ *is Herbrand model for* L *then*

- *formula* γ *of* L *is true in* $\mathbf{M} \iff \gamma(d)$ *is true in* $\mathbf{M}$ *for every* $d \in \mathbf{D}$
- *formula* δ *of* L *is true in* $\mathbf{M} \iff \gamma(d)$ *is true in* $\mathbf{M}$ *for some* $d \in \mathbf{D}$

Proof

. . . exercise . . .

Outline

Definition

set **H** of sentences of first-order language L is first-order Hintikka set, provided

1. for any propositional letter A, not both $A \in \mathbf{H}$ and $\neg A \in \mathbf{H}$
2. $\bot \notin \mathbf{H}$, $\neg\top \notin \mathbf{H}$
3. if $\neg\neg Z \in \mathbf{H}$ then $Z \in \mathbf{H}$
4. if $\alpha \in \mathbf{H}$ then $\alpha_1 \in \mathbf{H}$ and $\alpha_2 \in \mathbf{H}$
5. if $\beta \in \mathbf{H}$ then $\beta_1 \in \mathbf{H}$ or $\beta_2 \in \mathbf{H}$

Definition

set **H** of sentences of first-order language L is first-order Hintikka set, provided

1. for any propositional letter A, not both $A \in \mathbf{H}$ and $\neg A \in \mathbf{H}$
2. $\bot \notin \mathbf{H}$, $\neg\top \notin \mathbf{H}$
3. if $\neg\neg Z \in \mathbf{H}$ then $Z \in \mathbf{H}$
4. if $\alpha \in \mathbf{H}$ then $\alpha_1 \in \mathbf{H}$ and $\alpha_2 \in \mathbf{H}$
5. if $\beta \in \mathbf{H}$ then $\beta_1 \in \mathbf{H}$ or $\beta_2 \in \mathbf{H}$
6. if $\gamma \in \mathbf{H}$ then $\gamma(t) \in \mathbf{H}$ for every closed term t of L

Definition

set **H** of sentences of first-order language L is first-order Hintikka set, provided

1. for any propositional letter A, not both $A \in \mathbf{H}$ and $\neg A \in \mathbf{H}$
2. $\bot \notin \mathbf{H}$, $\neg\top \notin \mathbf{H}$
3. if $\neg\neg Z \in \mathbf{H}$ then $Z \in \mathbf{H}$
4. if $\alpha \in \mathbf{H}$ then $\alpha_1 \in \mathbf{H}$ and $\alpha_2 \in \mathbf{H}$
5. if $\beta \in \mathbf{H}$ then $\beta_1 \in \mathbf{H}$ or $\beta_2 \in \mathbf{H}$
6. if $\gamma \in \mathbf{H}$ then $\gamma(t) \in \mathbf{H}$ for every closed term t of L
7. if $\delta \in \mathbf{H}$ then $\delta(t) \in \mathbf{H}$ for some closed term t of L

Definition

set $\mathbf{H}$ of sentences of first-order language L is first-order Hintikka set, provided

1. for any propositional letter A, not both $A \in \mathbf{H}$ and $\neg A \in \mathbf{H}$
2. $\bot \notin \mathbf{H}$, $\neg\top \notin \mathbf{H}$
3. if $\neg\neg Z \in \mathbf{H}$ then $Z \in \mathbf{H}$
4. if $\alpha \in \mathbf{H}$ then $\alpha_1 \in \mathbf{H}$ and $\alpha_2 \in \mathbf{H}$
5. if $\beta \in \mathbf{H}$ then $\beta_1 \in \mathbf{H}$ or $\beta_2 \in \mathbf{H}$
6. if $\gamma \in \mathbf{H}$ then $\gamma(t) \in \mathbf{H}$ for every closed term t of L
7. if $\delta \in \mathbf{H}$ then $\delta(t) \in \mathbf{H}$ for some closed term t of L

Lemma (Hintikka's Lemma)

if $\mathbf{H}$ *is first-order Hintikka set with respect to language* L *with nonempty set of closed terms then* $\mathbf{H}$ *is satisfiable in Herbrand model*

Proof

- **H** is first-order Hintikka set with respect to L

Proof

- $\mathbf{H}$ is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$

Proof

- **H** is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - **D** is collection of closed terms of L

Proof

- $\mathbf{H}$ is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - $\mathbf{D}$ is collection of closed terms of L
  - $c^{\mathbf{I}} = c$ for constant symbols c of L

Proof

- $\mathbf{H}$ is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - $\mathbf{D}$ is collection of closed terms of L
  - $c^{\mathbf{I}} = c$ for constant symbols c of L
  - $f^{\mathbf{I}}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$ for n-place function symbols f of L and $t_1, \dots, t_n \in D$

Proof

- **H** is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - **D** is collection of closed terms of L
  - $c^{\mathbf{I}} = c$ for constant symbols c of L
  - $f^{\mathbf{I}}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$ for n-place function symbols f of L and $t_1, \dots, t_n \in D$
- $t^{\mathbf{I}} = t$ for each closed term t

Proof

- $\mathbf{H}$ is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - $\mathbf{D}$ is collection of closed terms of L
  - $c^{\mathbf{I}} = c$ for constant symbols c of L
  - $f^{\mathbf{I}}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$ for n-place function symbols f of L and $t_1, \dots, t_n \in D$
  - $\langle t_1, \dots, t_n \rangle$ belongs to $R^{\mathbf{I}}$ for n-place relation symbols R of L if $R(t_1, \dots, t_n) \in \mathbf{H}$
- $t^{\mathbf{I}} = t$ for each closed term t

Proof

- $\mathbf{H}$ is first-order Hintikka set with respect to L
- construct model $\mathbf{M} = \langle \mathbf{D}, \mathbf{I} \rangle$
  - $\mathbf{D}$ is collection of closed terms of L
  - $c^{\mathbf{I}} = c$ for constant symbols c of L
  - $f^{\mathbf{I}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ for n-place function symbols f of L and $t_1, \ldots, t_n \in D$
  - $\langle t_1, \ldots, t_n \rangle$ belongs to $R^{\mathbf{I}}$ for n-place relation symbols R of L if $R(t_1, \ldots, t_n) \in \mathbf{H}$
- $t^{\mathbf{I}} = t$ for each closed term t
- claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for every sentence X of L

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \ldots, t_n) \in \mathbf{H}$

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \dots, t_n) \in \mathbf{H}$

  $[R(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \dots, t_n \rangle \in R^{\mathbf{I}}$

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \ldots, t_n) \in \mathbf{H}$

  $[R(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \ldots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \ldots, t_n) \in \mathbf{H}$

  $[R(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \ldots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \ldots, t_n) \in \mathbf{H}$

  $[R(t_1, \ldots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \ldots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \ldots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case
- suppose $\gamma \in \mathbf{H}$

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \dots, t_n) \in \mathbf{H}$

  $[R(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \dots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case
- suppose $\gamma \in \mathbf{H}$

  $\gamma(t) \in \mathbf{H}$ for every closed term t

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \dots, t_n) \in \mathbf{H}$

  $[R(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \dots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case
- suppose $\gamma \in \mathbf{H}$

  $\gamma(t) \in \mathbf{H}$ for every closed term t

  $\gamma(t)$ is true in $\mathbf{M}$ for every $t \in \mathbf{D}$ according to induction hypothesis

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \dots, t_n) \in \mathbf{H}$

  $[R(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathsf{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \dots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case
- suppose $\gamma \in \mathbf{H}$

  $\gamma(t) \in \mathbf{H}$ for every closed term t

  $\gamma(t)$ is true in $\mathbf{M}$ for every $t \in \mathbf{D}$ according to induction hypothesis

  γ is true in $\mathbf{M}$ using Lemma on slide 55

Proof (cont'd)

claim: if $X \in \mathbf{H}$ then X is true in $\mathbf{M}$, for each sentence X of L

induction on X

- $\top$, $\bot$: trivial
- suppose $R(t_1, \dots, t_n) \in \mathbf{H}$

  $[R(t_1, \dots, t_n)]^{\mathbf{I},\mathbf{A}} = \mathrm{t}$ because $\langle t_1^{\mathbf{I},\mathbf{A}}, \dots, t_n^{\mathbf{I},\mathbf{A}} \rangle = \langle t_1, \dots, t_n \rangle \in R^{\mathbf{I}}$
- negation of atomic formula is straightforward
- α and β are treated as in propositional case
- suppose $\gamma \in \mathbf{H}$

  $\gamma(t) \in \mathbf{H}$ for every closed term t

  $\gamma(t)$ is true in $\mathbf{M}$ for every $t \in \mathbf{D}$ according to induction hypothesis

  γ is true in $\mathbf{M}$ using Lemma on slide 55
- δ ... exercise ...

Outline

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from $\mathbf{C}$

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from $\mathbf{C}$
- elements of **par** are called parameters

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from $\mathbf{C}$
- elements of **par** are called parameters
- $L^{\mathbf{par}} = L(\mathbf{R}, \mathbf{F}, \mathbf{C} \cup \mathbf{par})$

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from **C**
- elements of **par** are called parameters
- $L^{\mathbf{par}} = L(\mathbf{R}, \mathbf{F}, \mathbf{C} \cup \mathbf{par})$

Definition

collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ is first-order consistency property if $\mathcal{C}$ is propositional consistency property

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from **C**
- elements of **par** are called parameters
- $L^{\mathbf{par}} = L(\mathbf{R}, \mathbf{F}, \mathbf{C} \cup \mathbf{par})$

Definition

collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ is first-order consistency property if $\mathcal{C}$ is propositional consistency property and for each $S \in \mathcal{C}$:

6. if $\gamma \in S$ then $S \cup \{\gamma(t)\} \in \mathcal{C}$ for every closed term t of $L^{\mathbf{par}}$

Definition

given first-order language $L = L(\mathbf{R}, \mathbf{F}, \mathbf{C})$

- **par** is countably infinite set of constant symbols disjoint from **C**
- elements of **par** are called parameters
- $L^{\mathbf{par}} = L(\mathbf{R}, \mathbf{F}, \mathbf{C} \cup \mathbf{par})$

Definition

collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ is first-order consistency property if $\mathcal{C}$ is propositional consistency property and for each $S \in \mathcal{C}$:

6. if $\gamma \in S$ then $S \cup \{\gamma(t)\} \in \mathcal{C}$ for every closed term t of $L^{\mathbf{par}}$
7. if $\delta \in S$ then $S \cup \{\delta(p)\} \in \mathcal{C}$ for some parameter p of $L^{\mathbf{par}}$

Theorem (First-Order Model Existence)

if $\mathcal{C}$ is first-order consistency property with respect to L and $S \in \mathcal{C}$ is set of sentences over L then S is satisfiable

Theorem (First-Order Model Existence)

if $\mathcal{C}$ is first-order consistency property with respect to L and $S \in \mathcal{C}$ is set of sentences over L then S is satisfiable in Herbrand model with respect to $L^{\mathbf{par}}$

Theorem (First-Order Model Existence)

if $\mathcal{C}$ is first-order consistency property with respect to L and $S \in \mathcal{C}$ is set of sentences over L then S is satisfiable in Herbrand model with respect to $L^{\mathbf{par}}$

Lemma

every first-order consistency property can be extended to one that is subset closed

Theorem (First-Order Model Existence)

if $\mathcal{C}$ is first-order consistency property with respect to L and $S \in \mathcal{C}$ is set of sentences over L then S is satisfiable in Herbrand model with respect to $L^{\mathbf{par}}$

Lemma

every first-order consistency property can be extended to one that is subset closed

Definition

alternate first-order consistency property is collection **C** meeting conditions for first-order consistency property, except that condition 7 is replaced by

7' if $\delta \in S$ then $S \cup \{\delta(p)\} \in \mathcal{C}$ for every parameter p that is new to S

Definition

- parameter substitution is mapping π from set of parameters to itself

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of $L^{\mathbf{par}}$ in the obvious way

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of L^{par} in the obvious way

Lemma

let $\mathcal{C}$ be first-order consistency property that is closed under subsets and let $\mathcal{C}^+ = \{S \mid S\pi \in \mathcal{C}$ for some parameter substitution $\pi\}$

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of L^{par} in the obvious way

Lemma

let $\mathcal{C}$ be first-order consistency property that is closed under subsets and let $\mathcal{C}^+ = \{S \mid S\pi \in \mathcal{C}$ for some parameter substitution $\pi\}$

- *$\mathcal{C}^+$ extends $\mathcal{C}$*

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of L^{par} in the obvious way

Lemma

let $\mathcal{C}$ be first-order consistency property that is closed under subsets and let $\mathcal{C}^+ = \{S \mid S\pi \in \mathcal{C}$ for some parameter substitution $\pi\}$

- *$\mathcal{C}^+$ extends $\mathcal{C}$*
- *$\mathcal{C}^+$ is closed under subsets*

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of $L^{\textbf{par}}$ in the obvious way

Lemma

let $\mathcal{C}$ be first-order consistency property that is closed under subsets and let $\mathcal{C}^+ = \{S \mid S\pi \in \mathcal{C}$ for some parameter substitution $\pi\}$

- *$\mathcal{C}^+$ extends $\mathcal{C}$*
- *$\mathcal{C}^+$ is closed under subsets*
- *$\mathcal{C}^+$ is alternate first-order consistency property*

Definition

- parameter substitution is mapping π from set of parameters to itself
- π is extended to (sets of) formulas of $L^{\mathbf{par}}$ in the obvious way

Lemma

let $\mathcal{C}$ be first-order consistency property that is closed under subsets and let $\mathcal{C}^+ = \{S \mid S\pi \in \mathcal{C}$ for some parameter substitution $\pi\}$

- *$\mathcal{C}^+$ extends $\mathcal{C}$*
- *$\mathcal{C}^+$ is closed under subsets*
- *$\mathcal{C}^+$ is alternate first-order consistency property*

Lemma

every subset closed alternate first-order consistency property can be extended to one of finite character

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\mathbf{par}}$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\textbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S$$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\mathbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ & \\ & \end{cases}$$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of L^{par}
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ & \end{cases}$$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\textbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ & \end{cases}$$

with fixed parameter p which is new to $S_n \cup \{X_n\}$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\textbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

with fixed parameter p which is new to $S_n \cup \{X_n\}$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of L^{par}
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\mathbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$

- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of L^{par}
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S
- $\mathbf{H} \in \mathcal{C}^*$ (because $\mathcal{C}^*$ is of finite character)

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\textbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S
- $\mathbf{H} \in \mathcal{C}^*$ (because $\mathcal{C}^*$ is of finite character) and $\mathbf{H}$ is maximal in $\mathcal{C}^*$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \dots$ be enumeration of all sentences of $L^{\mathbf{par}}$
- define sequence $S_1, S_2, S_3, \dots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S
- $\mathbf{H} \in \mathcal{C}^*$ (because $\mathcal{C}^*$ is of finite character) and $\mathbf{H}$ is maximal in $\mathcal{C}^*$
- $\mathbf{H}$ is first-order Hintikka set with respect to $L^{\mathbf{par}}$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\mathbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S
- $\mathbf{H} \in \mathcal{C}^*$ (because $\mathcal{C}^*$ is of finite character) and $\mathbf{H}$ is maximal in $\mathcal{C}^*$
- $\mathbf{H}$ is first-order Hintikka set with respect to $L^{\mathbf{par}}$
- $\mathbf{H}$ is satisfiable by Hintikka's Lemma in Herbrand model with respect to $L^{\mathbf{par}}$

Proof of First-Order Model Existence

- extend $\mathcal{C}$ to alternate first-order consistency property $\mathcal{C}^*$ of finite character
- let $X_1, X_2, X_3, \ldots$ be enumeration of all sentences of $L^{\mathbf{par}}$
- define sequence $S_1, S_2, S_3, \ldots$ of members of $\mathcal{C}^*$:

$$S_1 = S \qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n \neq \delta \\ S_n \cup \{X_n, \delta(p)\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C}^* \text{ and } X_n = \delta \\ S_n & \text{otherwise} \end{cases}$$

  with fixed parameter p which is new to $S_n \cup \{X_n\}$
- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ extends S
- $\mathbf{H} \in \mathcal{C}^*$ (because $\mathcal{C}^*$ is of finite character) and $\mathbf{H}$ is maximal in $\mathcal{C}^*$
- $\mathbf{H}$ is first-order Hintikka set with respect to $L^{\mathbf{par}}$
- $\mathbf{H}$ is satisfiable by Hintikka's Lemma in Herbrand model with respect to $L^{\mathbf{par}}$
- $S \subseteq \mathbf{H}$ is satisfiable in Herbrand model with respect to $L^{\mathbf{par}}$

Outline

- Overviews
- First-Order Logic
- Herbrand Models
- Uniform Notation
- Hintikka's Lemma
  - Model Existence Theorem
  - Compactness
  - Löwenheim-Skolem
  - Logical Consequence
- Exercises
- Further Reading

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Proof

- define collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Proof

- define collection $\mathcal{C}$ of sets of sentences of L^{par} as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. every finite subset of W is satisfiable

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Proof

- define collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. every finite subset of W is satisfiable
- $S \in \mathcal{C}$

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Proof

- define collection $\mathcal{C}$ of sets of sentences of $L^{\textsf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. every finite subset of W is satisfiable
- $S \in \mathcal{C}$
- $\mathcal{C}$ is first-order consistency property

Theorem (First-Order Compactness)

if every finite subset of a set S of sentences of first-order language L is satisfiable then S is satisfiable

Proof

- define collection $\mathcal{C}$ of sets of sentences of L^{par} as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. every finite subset of W is satisfiable
- $S \in \mathcal{C}$
- $\mathcal{C}$ is first-order consistency property
- S is satisfiable according to First-Order Model Existence Theorem

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \ldots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \dots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements
- consider $S^* = S \cup \{A_2, A_3, \dots\}$

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \ldots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements
- consider $S^* = S \cup \{A_2, A_3, \ldots\}$
- every finite subset of S^* is satisfiable

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \dots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements
- consider $S^* = S \cup \{A_2, A_3, \dots\}$
- every finite subset of S^* is satisfiable
- S^* is satisfiable by First-Order Compactness Theorem

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \dots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements
- consider $S^* = S \cup \{A_2, A_3, \dots\}$
- every finite subset of S^* is satisfiable
- S^* is satisfiable by First-Order Compactness Theorem
- any model in which S^* is satisfiable must be infinite

Corollary

any set S of sentences of first-order language L that is satisfiable in arbitrarily large finite models is satisfiable in some infinite model

Proof

- suppose S is satisfiable in arbitrary large finite models
- let R be two-place relation symbol not in L and let L' be L extended with R
- there exist sentences $A_2, A_3, \ldots$ involving R such that A_i is not true in any model with less than i elements but can be made true in any domain with at least i elements
- consider $S^* = S \cup \{A_2, A_3, \ldots\}$
- every finite subset of S^* is satisfiable
- S^* is satisfiable by First-Order Compactness Theorem
- any model in which S^* is satisfiable must be infinite

Example

sentence

$$A_n \;=\; (\exists x_1)(\exists x_2)\cdots(\exists x_n)\Big[\;\bigwedge_{i=1}^{n} R(x_i,x_i) \;\wedge \bigwedge_{1\leqslant i<j\leqslant n} \neg R(x_i,x_j)\;\Big]$$

Example

sentence

$$A_n = (\exists x_1)(\exists x_2)\cdots(\exists x_n)\Big[\bigwedge_{i=1}^{n} R(x_i,x_i) \wedge \bigwedge_{1\leqslant i<j\leqslant n} \neg R(x_i,x_j)\Big]$$

- is not true in any model with less than n elements

Example

sentence

$$A_n \;=\; (\exists x_1)(\exists x_2)\cdots(\exists x_n)\Big[\,\bigwedge_{i=1}^{n} R(x_i,x_i) \;\wedge \bigwedge_{1\leqslant i<j\leqslant n} \neg R(x_i,x_j)\,\Big]$$

- is not true in any model with less than n elements
- can be made true in any domain with at least n elements

Example

sentence

$$A_n \;=\; (\exists x_1)(\exists x_2)\cdots(\exists x_n)\Big[\ \bigwedge_{i=1}^{n} R(x_i,x_i)\ \wedge \bigwedge_{1\leqslant i<j\leqslant n} \neg R(x_i,x_j)\ \Big]$$

- is not true in any model with less than n elements
- can be made true in any domain with at least n elements

Remark

notion of finiteness cannot be captured using machinery of classical first-order logic

Outline

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of $L^{\mathbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. W is satisfiable

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of $L^{\textbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. W is satisfiable
- $S \in \mathcal{C}$

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of L^{par} as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. W is satisfiable
- $S \in \mathcal{C}$
- $\mathcal{C}$ is first-order consistency property

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

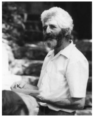
- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of $L^{\textbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. W is satisfiable
- $S \in \mathcal{C}$
- $\mathcal{C}$ is first-order consistency property
- Model Existence Theorem:
  S is satisfiable in Herbrand model with respect to $L^{\textbf{par}}$

Theorem (Löwenheim-Skolem)

if set S of sentences of first-order language L is satisfiable then S is satisfiable in countable model

Proof

- suppose S is satisfiable
- define collection $\mathcal{C}$ of sets of sentences of $L^{\textbf{par}}$ as follows: $W \in \mathcal{C}$ if
  1. infinitely many parameters are new to W
  2. W is satisfiable
- $S \in \mathcal{C}$
- $\mathcal{C}$ is first-order consistency property
- Model Existence Theorem:
  S is satisfiable in Herbrand model with respect to $L^{\textbf{par}}$
- $L^{\textbf{par}}$ has countable alphabet and hence countably many closed terms

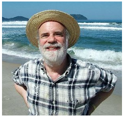


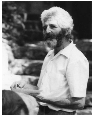
William Craig
(1918 – 2016)

Jacques Herbrand
(1908 – 1931)

David Hilbert
(1862 – 1943)

Jaakko Hintikka
(1929 – 2015)



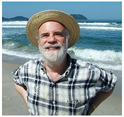


William Craig
(1918 – 2016)

Jacques Herbrand
(1908 – 1931)

David Hilbert
(1862 – 1943)

Jaakko Hintikka
(1929 – 2015)



Leopold Löwenheim
(1878 – 1957)

Thoralf Skolem
(1887 – 1963)

Outline

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Theorem

$S \vDash_f X$ if and only if $S_0 \vDash_f X$ for some finite subset S_0 of S

Definition

sentence X is logical consequence of set S of sentences, $S \models_f X$, if X is true in every model in which all members of S are true

Theorem

$S \models_f X$ if and only if $S_0 \models_f X$ for some finite subset S_0 of S

Proof

$\Rightarrow$ if $S \models_f X$ then $S \cup \{\neg X\}$ is not satisfiable

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Theorem

$S \vDash_f X$ if and only if $S_0 \vDash_f X$ for some finite subset S_0 of S

Proof

$\Rightarrow$ if $S \vDash_f X$ then $S \cup \{\neg X\}$ is not satisfiable

some finite subset S' of $S \cup \{\neg X\}$ is not satisfiable by compactness

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Theorem

$S \vDash_f X$ if and only if $S_0 \vDash_f X$ for some finite subset S_0 of S

Proof

$\Rightarrow$ if $S \vDash_f X$ then $S \cup \{\neg X\}$ is not satisfiable

some finite subset S' of $S \cup \{\neg X\}$ is not satisfiable by compactness

we may assume that $S' = S_0 \cup \{\neg X\}$ for finite subset S_0 of S

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Theorem

$S \vDash_f X$ if and only if $S_0 \vDash_f X$ for some finite subset S_0 of S

Proof

$\Rightarrow$ if $S \vDash_f X$ then $S \cup \{\neg X\}$ is not satisfiable

some finite subset S' of $S \cup \{\neg X\}$ is not satisfiable by compactness

we may assume that $S' = S_0 \cup \{\neg X\}$ for finite subset S_0 of S

$S_0 \vDash_f X$

Definition

sentence X is logical consequence of set S of sentences, $S \vDash_f X$, if X is true in every model in which all members of S are true

Theorem

$S \vDash_f X$ if and only if $S_0 \vDash_f X$ for some finite subset S_0 of S

Proof

$\Rightarrow$ if $S \vDash_f X$ then $S \cup \{\neg X\}$ is not satisfiable

some finite subset S' of $S \cup \{\neg X\}$ is not satisfiable by compactness

we may assume that $S' = S_0 \cup \{\neg X\}$ for finite subset S_0 of S

$S_0 \vDash_f X$

$\Leftarrow$ easy

Outline

Fitting

- Give a translation from propositional logic into first-order logic that is natural in the sense that properties carry over, e.g. that a formula is satisfiable/valid/a contradiction iff so is its translation.
- Exercise 5.2.2
- Bonus Exercise 5.2.4
- Exercise 5.3.2
- Exercise 5.3.9 !
- Exercise 5.4.1 or Exercise 5.4.2
- Bonus Exercise 5.5.2 or Exercise 5.6.3
- Exercise 5.9.2
- Bonus Exercise 5.9.3
- Exercise 5.10.1 or 5.10.3(1,2)

Outline

Fitting

- Section 5.7
- Section 5.8 !
- Section 5.9 !
- Section 5.10
- Section 6.1 !
- Section 6.3 !
- Section 6.4 !
- Section 6.5 !
- Section 8.2