Automated Theorem Proving

Georg Moser

Institute of Computer Science @ UIBK

Winter 2015

Reconciling Computational Logic and Automated Theorem Proving

Theorem (Fitting)

if $\mathcal{C}$ is first-order consistency property with respect to $\mathcal{L}$ and $S \in \mathcal{C}$ is set of sentences over $\mathcal{L}$ then S is satisfiable in Herbrand model with respect to $\mathcal{L}^{par}$

Theorem

1. *if S^* is a set of formula sets of $\mathcal{L}^+$ having the satisfaction properties, then $\forall$ formula sets $\mathcal{G} \in S^*$ of $\mathcal{L}$, $\exists\ \mathcal{M}$, $\mathcal{M} \models \mathcal{G}$*
2. *$\forall$ elements m of $\mathcal{M}$: m denotes term in $\mathcal{L}^+$*

Fact

same result!

Reconciling (cont'd)

Definition (Optimised Skolemisation)

- let A be a sentence in NNF and $B = \exists x_1 \cdots \exists x_k (E \wedge F)$ a subformula of A with $\mathcal{FV}\mathrm{ar}(\exists \vec{x}(E \wedge F)) = \{y_1, \ldots, y_n\}$
- suppose $A = C[B]$
- suppose $A \to \forall y_1, \ldots, \forall y_n \exists x_1 \cdots \exists x_k E$ is valid
- we define an optimised Skolemisation step as follows

  $$\mathrm{opt\_step}(A) = \forall \vec{y} E\{\ldots, x_i \mapsto f_i(\vec{y}), \ldots\} \wedge C[F\{\ldots, x_i \mapsto f_i(\vec{y}), \ldots\}]$$

  where $f_1, \ldots, f_k$ are new Skolem function symbols

Theorem (Skolemization)

...

Summary Last Lecture

Definition

subsumption and resolution can be combined in the following ways

1. forward subsumption
   newly derived clauses subsumed by existing clauses are deleted
2. backward subsumption
   existing clauses C subsumed by newly derived clauses D become inactive
   inactive clauses are reactivated, if D is no ancestor of current clause
3. replacement
   the set of all clauses (derived and intital) are frequently reduced under subsumption

Theorem

(ordered) resolution is complete under forward subsumption and tautology elimination

Outline of the Lecture

Early Approaches in Automated Reasoning

Herbrand's theorem for dummies, Gilmore's prover, method of Davis and Putnam

Starting Points

resolution, tableau provers, Skolemisation, ordered resolution, redundancy and deletion

Automated Reasoning with Equality

paramodulation, ordered completion and proof orders, superposition

Applications of Automated Reasoning

Neuman-Stubblebinde Key Exchange Protocol, Robbins problem

Definition (Satisfaction Properties)

let $\mathcal{L}^+$ be an extension of $\mathcal{L}$ with infinitely many individual constants (= parameters); let S be a set of sets of formulas over $\mathcal{L}^+$ such that

1. if $\mathcal{G}_0 \subseteq \mathcal{G}$, then $\mathcal{G}_0 \in S$
2. no formula F and $\neg F$ in $\mathcal{G}$
3. if $\neg\neg F \in \mathcal{G}$, then $\mathcal{G} \cup \{F\} \in S$
4. if $(E \vee F) \in \mathcal{G}$, then $\mathcal{G} \cup \{E\} \in S$ or $\mathcal{G} \cup \{F\} \in S$
5. if $\neg(E \vee F) \in \mathcal{G}$, then $\mathcal{G} \cup \{\neg E\} \in S$ and $\mathcal{G} \cup \{\neg F\} \in S$
6. if $\exists x F(x) \in \mathcal{G}$, the constant c doesn't occur in $\mathcal{G}$, then $\mathcal{G} \cup \{F(c)\} \in S$
7. if $\neg\exists x F(x) \in \mathcal{G}$, then $\forall$ terms t, $\mathcal{G} \cup \{\neg F(t)\} \in S$
8. for any term t, $\mathcal{G} \cup \{t = t\} \in S$
9. if $\{F(s), s = t\} \subseteq \mathcal{G}$, then $\mathcal{G} \cup \{F(t)\} \in S$

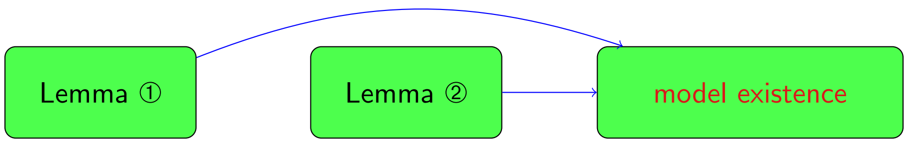
then S has the satisfaction properties (= is first-order consistency property)

First-Order Model Existence with Equality

$\mathcal{L}$ base language; $\mathcal{L}^+ \supseteq \mathcal{L}$ infinitely many new individual constants

Theorem (Model Existence Theorem (with Equality))

1. *if S^* is a set of formula sets of $\mathcal{L}^+$ having the satisfaction properties, then $\forall$ formula sets $\mathcal{G} \in S^*$ of $\mathcal{L}$, $\exists$ $\mathcal{M}$, $\mathcal{M} \models \mathcal{G}$*
2. *$\forall$ elements m of $\mathcal{M}$: m denotes term in $\mathcal{L}^+$*



Closure Properties (= Hintikka set)

Lemma

the set $\mathcal{G}$ of formulas that are true in $\mathcal{M}$ admit

1. *no formula F and $\neg F$ in $\mathcal{G}$*
2. *if $\neg\neg F \in \mathcal{G}$, then $F \in \mathcal{G}$*
3. *if $(E \vee F) \in \mathcal{G}$, then $E \in \mathcal{G}$ or $F \in \mathcal{G}$*
4. *if $\neg(E \vee F) \in \mathcal{G}$, then $\neg E \in \mathcal{G}$ and $\neg F \in \mathcal{G}$*
5. *if $\exists x F(x) \in \mathcal{G}$, then $\exists$ term t (of $\mathcal{L}^+$), $F(t) \in \mathcal{G}$*
6. *if $\neg\exists x F(x) \in \mathcal{G}$, then $\forall$ term t (of $\mathcal{L}^+$), $\neg F(t) \in \mathcal{G}$*
7. *$\forall$ term t (of $\mathcal{L}^+$), $t = t \in \mathcal{G}$*
8. *if $F(s) \in \mathcal{G}$, $s = t \in \mathcal{G}$, then $F(t) \in \mathcal{G}$*

Definition

we call the properties of $\mathcal{G}$ closure properties (= Hintikka set)

Lemma ①

1. let $\mathcal{G}$ be a formula set admitting the closure properties
2. then $\exists$ interpretation $\mathcal{M}$ in which every element of the domain is the denotation of some term
3. $\mathcal{M} \models \mathcal{G}$

Lemma ②

1. let $\mathcal{L}$ be a language; $\mathcal{L}^+$ extension of $\mathcal{L}$ with infinitely many individual constants
2. let S^* be a set of formula sets (of $\mathcal{L}^+$), let S^* admit the satisfaction properties
3. $\forall$ formula set $\mathcal{G} \in S^*$ (of $\mathcal{L}$), $\exists$ $\mathcal{G}^* \supseteq \mathcal{G}$ (of $\mathcal{L}^+$), such that $\mathcal{G}^*$ fulfils the closure properties

Proof of Model Existence

by Lemma ② and Lemma ① ■

Proof of Lemma ①

(no identity, no function symbols)

- let $\mathcal{G}$ be a formula set admitting the closure properties
- then $\exists$ interpretation $\mathcal{M}$ in which every element of the domain is the denotation of some term
- $\mathcal{M} \models \mathcal{G}$

Proof

1. the domain of $\mathcal{M}$ is the set of terms (of $\mathcal{L}^+$)
2. $\forall$ constants c
$$c^{\mathcal{M}} := c$$
3. $\forall$ predicate constant P, $\forall$ terms $t_1, \dots, t_n$:
$$(t_1, \dots, t_n) \in P^{\mathcal{M}} \iff P(t_1, \dots, t_n) \in \mathcal{G}$$
4. $\forall$ variables x: $\ell(x) := x$

Proof (cont'd)

5. definition of $\mathcal{M}$ takes care of the demand that every element of its domain is the denotation of a term
6. we claim $\forall$ formulas F: $F \in \mathcal{G} \Rightarrow \mathcal{M} \models F$

Claim: $F \in \mathcal{G} \Rightarrow \mathcal{M} \models F$

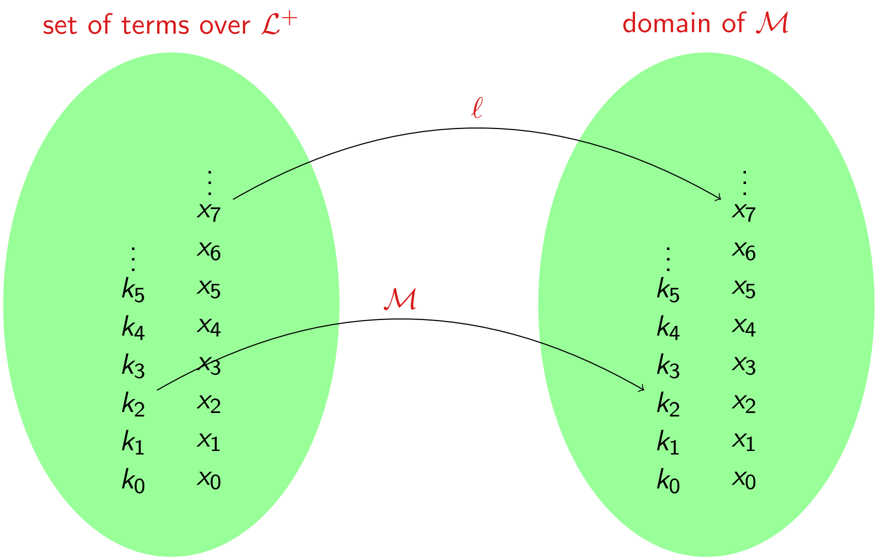
we show the claim by induction on F:

- for the base case, let $F = P(t_1, \dots, t_n)$, if $F \in \mathcal{G}$, then by definition $(t_1, \dots, t_n) \in P^{\mathcal{M}}$; hence $\mathcal{M} \models F$
- for the step case, we assume $F = \exists x G(x)$ and $F \in \mathcal{G}$; the other cases are similar

  by assumption $\mathcal{G}$ fulfils the closure properties, hence there exists a term t such that $G(t) \in \mathcal{G}$

  by induction hypothesis: $\mathcal{M} \models G(t)$ and thus $\mathcal{M} \models \exists x G(x)$ ■

Model Construction in a Picture



Proof of Lemma ②

(no identity, no function symbols)

- let $\mathcal{L}$ be a language; $\mathcal{L}^+$ extension of $\mathcal{L}$ with infinitely many individual constants
- let S^* be a set of formula sets (of $\mathcal{L}^+$), let S^* admit the satisfaction properties
- $\forall$ formula set $\mathcal{G} \in S^*$ (of $\mathcal{L}$), $\exists\, \mathcal{G}^* \supseteq \mathcal{G}$ (of $\mathcal{L}^+$), such that $\mathcal{G}^*$ fulfils the closure properties

Proof

- construct sequence of sets belonging to S^*

$$\mathcal{G} = \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots \qquad \mathcal{G}_n \subseteq \mathcal{G}_{n+1}$$

- $\mathcal{G}_n$ is constructed in step n
- set $\mathcal{G}^* = \bigcup_{n \geqslant 0} \mathcal{G}_n$
- closure properties induce (infinitely many) demands

Proof (cont'd)

Demands

1. no formula F and $\neg F$ in $\mathcal{G}_n$ for all $n \geqslant 0$
2. if $\neg\neg F \in \mathcal{G}_n$, then $\exists k \geqslant n$, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{F\}$
3. if $(E \vee F) \in \mathcal{G}_n$, then $\exists k \geqslant n$, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{E\}$ or $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{F\}$
4. if $\neg(E \vee F) \in \mathcal{G}_n$, then $\exists k_1, k_2 \geqslant n$, $\mathcal{G}_{k_1+1} = \mathcal{G}_{k_1} \cup \{\neg E\}$ and $\mathcal{G}_{k_2+1} = \mathcal{G}_{k_2} \cup \{\neg F\}$
5. if $\exists x F(x) \in \mathcal{G}_n$, then $\exists$ term t, $\exists k \geqslant n$, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{F(t)\}$
6. if $\neg\exists x F(x) \in \mathcal{G}_n$, then $\forall$ term t, $\exists k \geqslant n$, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{\neg F(t)\}$
7. $\forall$ terms t, $\exists k \geqslant n$ such that $t = t \in \mathcal{G}_k$
8. if $F(s) \in \mathcal{G}_n$, and $s = t \in \mathcal{G}_n$, $\exists k \geqslant n$ $F(t) \in \mathcal{G}_k$

Claim

all demands can be granted, in particular the satisfaction properties guarantee that any demand can be met

Proof (cont'd)

- consider Demand 5:
  if $\exists x F(x) \in \mathcal{G}_n$, then $\exists$ term t, $\exists k \geqslant n$, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{F(t)\}$
- we use that S^* fulfils the satisfaction properties (c is fresh):

$$\exists x F(x) \in \mathcal{G}_n \in S^* \Rightarrow \forall k \geqslant n\ \mathcal{G}_k \cup \{F(c)\} \in S^*$$

- we fulfil demand by setting (at step k)

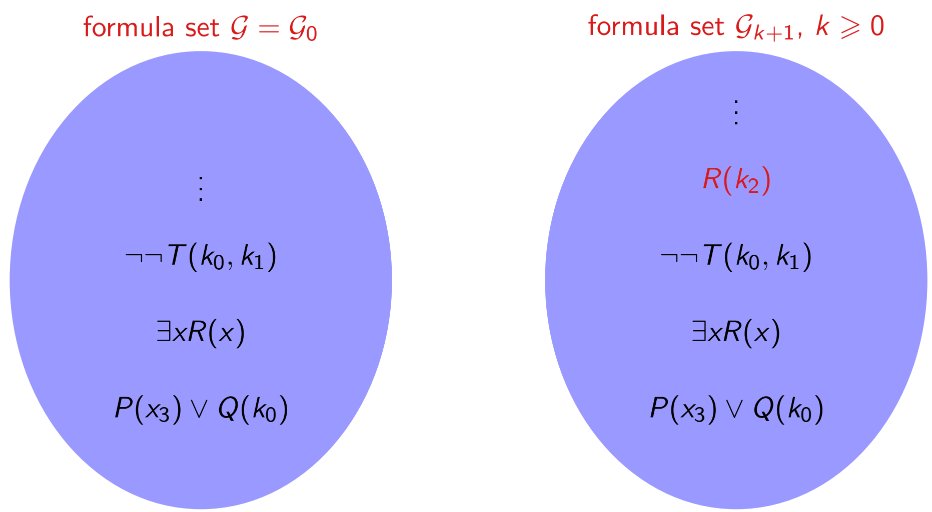
$$\mathcal{G}_{k+1} := \mathcal{G}_k \cup \{F(c)\} \qquad \text{for fresh } c$$

- similar for the Demands 2–8

Claim: $\exists$ fair strategy

- assign a pair (i, n) to each demand except Demand 6
  assign triple $(i, n, \ulcorner t \urcorner)$ to Demand 6, i is the number of the demand raised at step n, $\ulcorner t \urcorner$ Gödel number of t
- enumerate all pairs or triples and encode them as number k
- in step k we grant the demand raised at step n ■

Saturation of $\mathcal{G}$ in a Picture



$\exists x F(x) \in \mathcal{G}_n$, then $\exists k \geqslant n$, $\exists$ term t, $\mathcal{G}_{k+1} = \mathcal{G}_k \cup \{F(t)\}$

Generalisation I: Function Constants

Lemma ① (revisited)

1. let $\mathcal{G}$ be a formula set admitting the closure properties
2. suppose that $\mathcal{L}$ is free of the equality symbol
3. then $\exists$ interpretation $\mathcal{M}$ in which every element of the domain is the denotation of some term
4. $\mathcal{M} \models \mathcal{G}$

Proof.

1. $t_1, \dots, t_n$ elements of $\mathcal{M}$ and f an n-ary function symbol in $\mathcal{L}$
2. define: $f^{\mathcal{M}}(t_1, \dots, t_n) := f(t_1, \dots, t_n)$
3. following the earlier proof, we verify $\mathcal{M} \models \mathcal{G}$

this extends model existence to first-order logic (without $=$)

Generalisation II: Equality

Lemma ① (revisited again)

1. let $\mathcal{G}$ be a formula set admitting the closure properties
2. then $\exists$ interpretation $\mathcal{M}$ in which every element of the domain is the denotation of some term
3. $\mathcal{M} \models \mathcal{G}$

Proof.

1. suppose $(s = t) \in \mathcal{G}$, where s and t are syntactically different
2. for $\mathcal{M}$ according to the original construction, we have $\mathcal{M} \not\models s = t$
3. define a variant of the model $\mathcal{M}$, denoted as $\mathcal{M}'$
4. consider the set $\mathcal{E}$ of all equations induced by $\mathcal{G}$:

$$\mathcal{E} = \{s = t \mid \mathcal{G} \models s = t\}$$

Proof (cont'd).

5. $\mathcal{E}$ gives rise to an equivalence relation $\sim$
6. domain of $\mathcal{M}'$ is set of equivalent classes of terms of $\mathcal{L}^+$
7. $[t]_\sim$ denotes the equivalence class of t
8. definition of the structure underlying $\mathcal{M}'$:

$$f^{\mathcal{M}}([t_1]_\sim, \dots, [t_n]_\sim) = [f(t_1, \dots, t_n)]_\sim \qquad f \text{ is } n\text{-ary function}$$
$$P^{\mathcal{M}}([t_1]_\sim, \dots, [t_n]_\sim) \iff P(t_1, \dots, t_n) \in \mathcal{G} \qquad P \text{ is } n\text{-ary predicate}$$

9. from this $\mathcal{M}' \models \mathcal{G}$ ∎

this extends model existence to full first-order logic

Paramodulation Calculus

Definition

- let $\Box$ be a fresh constant; let $\mathcal{L}$ be our basic language
- terms of $\mathcal{L} \cup \{\Box\}$ such that $\Box$ occurs exactly once, are called contexts
- empty context is denoted as $\Box$
- for context $C[\Box]$ and a term t we write $C[t]$ for the replacement of $\Box$ by t

Example

- let $\mathcal{L} = \{\mathsf{c}, \mathsf{f}, \mathsf{P}\}$
- $\mathsf{P}(\mathsf{f}(\Box)) =: C[\Box]$ is a context
- $C[\mathsf{f}(\mathsf{c})] = \mathsf{P}(\mathsf{f}(\mathsf{f}(\mathsf{c})))$

Definition

$$\frac{C \lor A \quad D \lor \neg B}{(C \lor D)\sigma_1} \qquad \frac{C \lor A \lor B}{(C \lor A)\sigma_1}$$

$$\frac{C \lor s \neq s'}{C\sigma_2} \qquad \frac{C \lor s = t \quad D \lor L[s']}{(C \lor D \lor L[t])\sigma_2}$$

- σ_1 is a mgu of A and B (A, B atomic)
- σ_2 is a mgu of s and s'

Example

consider $\mathcal{C} = \{\mathsf{c} \neq \mathsf{d}, \mathsf{b} = \mathsf{d}, \mathsf{a} \neq \mathsf{d} \lor \mathsf{a} = \mathsf{c}, \mathsf{a} = \mathsf{b} \lor \mathsf{a} = \mathsf{d}\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{b} = \mathsf{d} \quad \mathsf{a} = \mathsf{b} \lor \mathsf{a} = \mathsf{d}}{\mathsf{a} = \mathsf{d} \lor \mathsf{a} = \mathsf{d}}}{\mathsf{a} = \mathsf{d}} \quad \mathsf{c} \neq \mathsf{d}}{\mathsf{a} \neq \mathsf{c}} \quad \dfrac{\dfrac{\mathsf{a} = \mathsf{d} \quad \mathsf{a} \neq \mathsf{d} \lor \mathsf{a} = \mathsf{c}}{\mathsf{d} \neq \mathsf{d} \lor \mathsf{a} = \mathsf{c}}}{\mathsf{a} = \mathsf{c}}}{\square}$$

Definition

- define the paramodulation operator $\mathsf{Res}_\mathsf{P}(\mathcal{C})$ as follows:

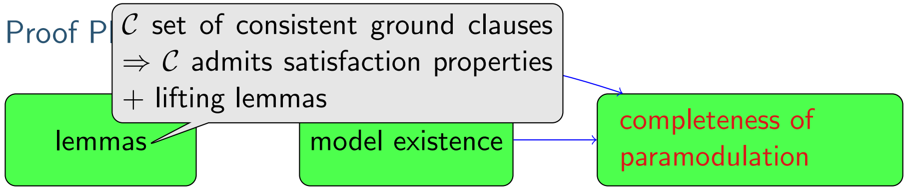
  $\mathsf{Res}_\mathsf{P}(\mathcal{C}) = \{D \mid D$ is paramodulant, etc. with premises in $\mathcal{C}\}$

- n^{th} (unrestricted) iteration $\mathsf{Res}^n_\mathsf{P}$ ($\mathsf{Res}^*_\mathsf{P}$) of the operator Res_P is defined as before

Theorem

paramodulation is sound and complete: if F is a sentence and $\mathcal{C}$ its clause form, then F is unsatisfiable iff $\square \in \mathsf{Res}^*_\mathsf{P}(\mathcal{C})$

Proof Pl



A Problem with Lifting

Claim

- let τ_1 and τ_2 be a ground and consider

  $$\frac{C\tau_1 \lor (s = t)\tau_1 \quad D\tau_2 \lor L\tau_2[s'\tau_2]}{C\tau_1 \lor D\tau_2 \lor L\tau_2[t\tau_2]}$$

  where $s\tau_1 = s'\tau_2$
- $\exists$ mgu σ of s and s', such that σ is more general then τ_1 and τ_2 and the following paramodulation step is valid

  $$\frac{C \lor s = t \quad D \lor L[s']}{(C \lor D \lor L[t])\sigma}$$

Fact

observe that paramodulation into variables is allowed

Example

- consider the following unit clauses

  $$\mathsf{a} = \mathsf{b} \qquad \mathsf{f}(x) = \mathsf{c}$$

  consider the paramodulation inference is $\mathsf{f}(\mathsf{b}) = \mathsf{c}$
- consider the following ground step:

  $$\frac{\mathsf{a} = \mathsf{b} \quad \mathsf{f}(\mathsf{f}(\mathsf{a})) = \mathsf{c}}{\mathsf{f}(\mathsf{f}(\mathsf{b})) = \mathsf{c}}$$

  then no lifting is possible: oops ☹...
- we add the functional reflexivity equation $\mathsf{f}(x) = \mathsf{f}(x)$ from which we get $\mathsf{f}(\mathsf{a}) = \mathsf{f}(\mathsf{b})$ by paramodulation into a variable
- then lifting becomes possible (using two steps)

  $$\dfrac{\dfrac{\mathsf{a} = \mathsf{b} \quad \mathsf{f}(x) = \mathsf{f}(x)}{\mathsf{f}(\mathsf{a}) = \mathsf{f}(\mathsf{b})} \quad \mathsf{f}(x) = \mathsf{c}}{\mathsf{f}(\mathsf{f}(\mathsf{b})) = \mathsf{c}}$$

Definition

$f(x_1, \dots, x_n) = f(x_1, \dots, x_n)$ is called functional reflexivity equation

Lemma

- *let τ_1 and τ_2 be a ground and consider*

$$\frac{C\tau_1 \vee (s = t)\tau_1 \quad D\tau_2 \vee L\tau_2[x\tau_2]}{C\tau_1 \vee D\tau_2 \vee L\tau_2[f(t\tau_1)]}$$

  where $x\tau_2 = f(s'\tau_3)$ and $s\tau_1 = s'\tau_3$

- *then the following paramodulation step is valid, trivially more general than the ground step*

$$\frac{\dfrac{C \vee s = t \quad f(x) = f(x)}{C \vee f(s) = f(t)} \qquad D \vee L[x]}{C \vee D \vee L[f(t)]}$$

Theorem

paramodulation is sound and complete: if F is a sentence and $\mathcal{C}$ its clause form (containing all functional reflexive equations), then F is unsatisfiable iff $\square \in \mathrm{Res}^*_{\mathrm{P}}(\mathcal{C})$

Proof.

in proof, we follow the standard procedure of combining model existence + (updated) lifting lemma ∎

Discussion

- alternative completenesss proof employs an adaption of the semantic tree argument
- paramodulation is inefficient
- one idea to reduce the search space is to combine ordered resolution with paramodulation: ordered paramodulation