Propositional and Predicate Logic - X

Petr Gregor

KTIML MFF UK

WS 2015/2016

Equisatisfiability

We will see that the problem of satisfiability can be reduced to open theories.

- Theories T, T' are *equisatisfiable* if T has a model $\Leftrightarrow$ T' has a model.
- A formula φ is in the *prenex (normal) form (PNF)* if it is written as

$$(Q_1 x_1) \ldots (Q_n x_n)\varphi',$$

  where Q_i denotes $\forall$ or $\exists$, variables $x_1, \ldots, x_n$ are all distinct and φ' is an open formula, called the *matrix*. $(Q_1 x_1) \ldots (Q_n x_n)$ is called the *prefix*.
- In particular, if all quantifiers are $\forall$, then φ is a *universal* formula.

To find an open theory equisatisfiable with T we proceed as follows.

(1) We replace axioms of T by equivalent formulas in the prenex form.

(2) We transform them, using new function symbols, to equisatisfiable universal formulas, so called Skolem variants.

(3) We take their matrices as axioms of a new theory.

Conversion rules for quantifiers

Let Q denote $\forall$ or $\exists$ and let $\overline{Q}$ denote the complementary quantifier.
For every formulas φ, ψ such that x is not free in the formula ψ,

$$\begin{array}{lrcl}
\models & \neg(Qx)\varphi & \leftrightarrow & (\overline{Q}x)\neg\varphi \\
\models & ((Qx)\varphi \wedge \psi) & \leftrightarrow & (Qx)(\varphi \wedge \psi) \\
\models & ((Qx)\varphi \vee \psi) & \leftrightarrow & (Qx)(\varphi \vee \psi) \\
\models & ((Qx)\varphi \rightarrow \psi) & \leftrightarrow & (\overline{Q}x)(\varphi \rightarrow \psi) \\
\models & (\psi \rightarrow (Qx)\varphi) & \leftrightarrow & (Qx)(\psi \rightarrow \varphi)
\end{array}$$

The above equivalences can be verified semantically or proved by the tableau method (*by taking the universal closure if it is not a sentence*).

Remark *The assumption that x is not free in ψ is necessary in each rule above (except the first one) for some quantifier Q. For example,*

$$\not\models \; ((\exists x)P(x) \wedge P(x)) \;\leftrightarrow\; (\exists x)(P(x) \wedge P(x))$$

Conversion to the prenex normal form

Proposition *Let φ' be the formula obtained from φ by replacing some occurrences of a subformula ψ with ψ'. If $T \models \psi \leftrightarrow \psi'$, then $T \models \varphi \leftrightarrow \varphi'$.*

Proof Easily by induction on the structure of the formula φ. □

Proposition *For every formula φ there is an equivalent formula φ' in the prenex normal form, i.e. $\models \varphi \leftrightarrow \varphi'$.*

Proof By induction on the structure of φ applying the conversion rules for quantifiers, replacing subformulas with their variants if needed, and applying the above proposition on equivalent transformations. □

For example,

$$
\begin{aligned}
((\forall z)P(x,z) \wedge P(y,z)) &\rightarrow \neg(\exists x)P(x,y)\\
((\forall u)P(x,u) \wedge P(y,z)) &\rightarrow (\forall x)\neg P(x,y)\\
(\forall u)(P(x,u) \wedge P(y,z)) &\rightarrow (\forall v)\neg P(v,y)\\
(\exists u)((P(x,u) \wedge P(y,z)) &\rightarrow (\forall v)\neg P(v,y))\\
(\exists u)(\forall v)((P(x,u) \wedge P(y,z)) &\rightarrow \neg P(v,y))
\end{aligned}
$$

Skolem variants

Let φ be a sentence of a language L in the prenex normal form, let $y_1, \dots, y_n$ be the existentially quantified variables in φ (in this order), and for every $i \leq n$ let $x_1, \dots, x_{n_i}$ be the variables that are universally quantified in φ before y_i. Let L' be an extension of L with new n_i-ary function symbols f_i for all $i \leq n$.

Let φ_S denote the formula of L' obtained from φ by removing all $(\exists y_i)$'s from the prefix and by replacing each occurrence of y_i with the term $f_i(x_1, \dots, x_{n_i})$. Then φ_S is called a *Skolem variant* of φ.

For example, for the formula φ

$$(\exists y_1)(\forall x_1)(\forall x_2)(\exists y_2)(\forall x_3)R(y_1, x_1, x_2, y_2, x_3)$$

the following formula φ_S is a Skolem variant of φ

$$(\forall x_1)(\forall x_2)(\forall x_3)R(f_1, x_1, x_2, f_2(x_1, x_2), x_3),$$

where f_1 is a new constant symbol and f_2 is a new binary function symbol.

Properties of Skolem variants

Lemma *Let φ be a sentence $(\forall x_1)\dots(\forall x_n)(\exists y)\psi$ of L and φ' be a sentence $(\forall x_1)\dots(\forall x_n)\psi(y/f(x_1,\dots,x_n))$ where f is a new function symbol. Then*

(1) *the reduct $\mathcal{A}$ of every model $\mathcal{A}'$ of φ' to the language L is a model of φ,*

(2) *every model $\mathcal{A}$ of φ can be expanded into a model $\mathcal{A}'$ of φ'.*

Remark *Compared to extensions by definition of a function symbol, the expansion in (2) does not need to be unique now.*

Proof (1) Let $\mathcal{A}' \models \varphi'$ and $\mathcal{A}$ be the reduct of $\mathcal{A}'$ to L. Since $\mathcal{A} \models \psi[e(y/a)]$ for every assignment e where $a = (f(x_1,\dots,x_n))^{A'}[e]$, we have also $\mathcal{A} \models \varphi$.
(2) Let $\mathcal{A} \models \varphi$. There exists a function $f^A \colon A^n \to A$ such that for every assignment e it holds $\mathcal{A} \models \psi[e(y/a)]$ where $a = f^A(e(x_1),\dots,e(x_n))$, and thus the expansion $\mathcal{A}'$ of $\mathcal{A}$ by the function f^A is a model of φ'. $\square$

Corollary *If φ' is a Skolem variant of φ, then both statements (1) and (2) hold for φ, φ' as well. Hence φ, φ' are equisatisfiable.*

Skolem's theorem

Theorem *Every theory T has an open conservative extension T^*.*

Proof We may assume that T is in a closed form. Let L be its language.

- By replacing each axiom of T with an equivalent formula in the prenex normal form we obtain an equivalent theory T°.
- By replacing each axiom of T° with its Skolem variant we obtain a theory T' in an extended language $L' \supseteq L$.
- Since the reduct of every model of T' to the language L is a model of T, the theory T' is an extension of T.
- Furthermore, since every model of T can be expanded to a model of T', it is a conservative extension.
- Since every axiom of T' is a universal sentence, by replacing them with their matrices we obtain an open theory T^* equivalent to T'. $\square$

Corollary *For every theory there is an equisatisfiable open theory.*

Reduction of unsatisfiability to propositional logic

If an open theory is unsatisfiable, we can demonstrate it "via ground terms".

For example, in the language $L = \langle P, R, f, c\rangle$ the theory

$$T = \{P(x,y) \vee R(x,y),\ \neg P(c,y),\ \neg R(x,f(x))\}$$

is unsatisfiable, and this can be demonstrated by an unsatisfiable conjunction of finitely many instances of (some) axioms of T in ground terms

$$(P(c,f(c)) \vee R(c,f(c)))\ \wedge\ \neg P(c,f(c))\ \wedge\ \neg R(c,f(c)),$$

which may be seen as an unsatisfiable propositional formula

$$(p \vee r)\ \wedge\ \neg p\ \wedge\ \neg r.$$

An instance $\varphi(x_1/t_1, \ldots, x_n/t_n)$ of an open formula φ in free variables $x_1, \ldots, x_n$ is a *ground instance* if all terms $t_1, \ldots, t_n$ are ground terms (i.e. terms without variables).

Herbrand model

Let $L = \langle \mathcal{R}, \mathcal{F} \rangle$ be a language with at least one constant symbol. *(If needed, we add a new constant symbol to L.)*

- The *Herbrand universe* for L is the set of all ground terms of L.
  For example, for $L = \langle P, f, c \rangle$ with f binary function sym., c constant sym.
  $$A = \{c, f(c,c), f(f(c,c),c), f(c,f(c,c)), f(f(c,c),f(c,c)), \ldots\}$$
- An L-structure $\mathcal{A}$ is a *Herbrand structure* if its domain A is the Herbrand universe for L and for each n-ary function symbol $f \in \mathcal{F}$, $t_1, \ldots, t_n \in A$,
  $$f^A(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$
  (including $n = 0$, i.e. $c^A = c$ for every constant symbol c).
  Remark *Compared to a canonical model, the relations are not specified.*
  E.g. $\mathcal{A} = \langle A, P^A, f^A, c^A \rangle$ *with* $P^A = \emptyset$, $c^A = c$, $f^A(c,c) = f(c,c)$, $\ldots$.
- A *Herbrand model* of a theory T is a Herbrand structure that models T.

Herbrand's theorem

Theorem *Let T be an open theory of a language L without equality and with at least one constant symbol. Then*

(a) *either T has a Herbrand model, or*

(b) *there are finitely many ground instances of axioms of T whose conjunction is unsatisfiable, and thus T has no model.*

Proof Let T' be the set of all ground instances of axioms of T. Consider a finished (e.g. systematic) tableau τ from T' in the language L (without adding new constant symbols) with the root entry $F\bot$.

- If the tableau τ contains a noncontradictory branch V, the canonical model from V is a Herbrand model of T.
- Else, τ is contradictory, i.e. $T' \vdash \bot$. Moreover, τ is finite, so $\bot$ is provable from finitely many formulas of T', i.e. their conjunction is unsatisfiable. $\square$

Remark *If the language L is with equality, we extend T to T^* by axioms of equality for L and if T^* has a Herbrand model $\mathcal{A}$, we take its quotient by $=^A$.*

Corollaries of Herbrand's theorem

Let L be a language containing at least one constant symbol.

Corollary *For every open $\varphi(x_1, \ldots, x_n)$ of L, the formula $(\exists x_1) \ldots (\exists x_n)\varphi$ is valid if and only if there exist mn ground terms t_{ij} of L for some m such that*

$$\varphi(x_1/t_{11}, \ldots, x_n/t_{1n}) \vee \ldots \vee \varphi(x_1/t_{m1}, \ldots, x_n/t_{mn})$$

is a (propositional) tautology.

Proof $(\exists x_1) \ldots (\exists x_n)\varphi$ is valid $\Leftrightarrow$ $(\forall x_1) \ldots (\forall x_n)\neg\varphi$ is unsatisfiable $\Leftrightarrow$ $\neg\varphi$ is unsatisfiable. The rest follows from Herbrand's theorem for $\{\neg\varphi\}$. $\square$

Corollary *An open theory T of L is satisfiable if and only if the theory T' of all ground instances of axioms of T is satisfiable.*

Proof If T has a model $\mathcal{A}$, every instance of each axiom of T is valid in $\mathcal{A}$, thus $\mathcal{A}$ is a model of T'. If T is unsatisfiable, by H. theorem there are (finitely) formulas of T' whose conjunction is unsatisfiable, thus T' is unsatisfiable. $\square$

Resolution method in predicate logic - introduction

- A refutation procedure - its aim is to show that a given formula (or theory) is unsatisfiable.
- It assumes open formulas in CNF (and in clausal form).

  A *literal* is *(now)* an atomic formula or its negation.

  A *clause* is a finite set of literals, $\square$ denotes the empty clause.

  A *formula (in clausal form)* is a (possibly infinite) set of clauses.

  Remark *Every formula (theory) can be converted to an equisatisfiable open formula (theory) in CNF, and then to a formula in clausal form.*
- The resolution rule is more general - it allows to resolve through literals that are unifiable.
- Resolution in predicate logic is based on resolution in propositional logic and unification.

Local scope of variables

Variables can be renamed locally within clauses.

Let φ be an *(input)* open formula in CNF.

- φ is satisfiable if and only if its universal closure φ' is satisfiable.
- For every two formulas ψ, χ and a variable x

  $$\models \quad (\forall x)(\psi \wedge \chi) \leftrightarrow (\forall x)\psi \wedge (\forall x)\chi$$

  (also in the case that x is free both in ψ and χ).
- Every clause in φ can thus be replaced by its universal closure.
- We can then take any variants of clauses (to rename variables apart).

For example, by renaming variables in the second clause of (1) *we obtain an equisatisfiable formula* (2).

(1) $\{\{P(x), Q(x, y)\}, \{\neg P(x), \neg Q(y, x)\}\}$

(2) $\{\{P(x), Q(x, y)\}, \{\neg P(v), \neg Q(u, v)\}\}$

Reduction to propositional level (grounding)

Herbrand's theorem gives us the following (inefficient) method.

- Let S be the *(input)* formula in clausal form.
- We can assume that the language contains at least one constant symbol.
- Let S' be the set of all ground instances of all clauses from S.
- By introducing propositional letters representing atomic sentences we may view S' as a (possibly infinite) propositional formula in clausal form.
- We may verify that it is unsatisfiable by resolution on propositional level.

For example, for $S = \{\{P(x,y), R(x,y)\}, \{\neg P(c,y)\}, \{\neg R(x,f(x))\}\}$ *the set*

$$S' = \{\{P(c,c), R(c,c)\}, \{P(c,f(c)), R(c,f(c))\}, \{P(f(c),f(c)), R(f(c),f(c))\} \ldots, \\ \{\neg P(c,c)\}, \{\neg P(c,f(c))\}, \ldots, \{\neg R(c,f(c))\}, \{\neg R(f(c),f(f(c)))\}, \ldots\}$$

is unsatisfiable since on propositional level

$$S' \supseteq \{\{P(c,f(c)), R(c,f(c))\}, \{\neg P(c,f(c))\}, \{\neg R(c,f(c))\}\} \vdash_R \square.$$

Substitutions - examples

It is more efficient to use suitable substitutions. For example, in

$a)$ $\{P(x), Q(x,a)\}$, $\{\neg P(y), \neg Q(b,y)\}$ substituting x/b, y/a gives $\{P(b), Q(b,a)\}$, $\{\neg P(a), \neg Q(b,a)\}$, which resolves to $\{P(b), \neg P(a)\}$.

Or, substituting x/y and resolving through $P(y)$ gives $\{Q(y,a), \neg Q(b,y)\}$.

$b)$ $\{P(x), Q(x,a), Q(b,y)\}$, $\{\neg P(v), \neg Q(u,v)\}$ substituting x/b, y/a, u/b, v/a gives $\{P(b), Q(b,a)\}$, $\{\neg P(a), \neg Q(b,a)\}$, resolving to $\{P(b), \neg P(a)\}$.

$c)$ $\{P(x), Q(x,z)\}$, $\{\neg P(y), \neg Q(f(y),y)\}$ substituting $x/f(z)$, y/z gives $\{P(f(z)), Q(f(z),z)\}$, $\{\neg P(z), \neg Q(f(z),z)\}$, resolving to $\{P(f(z)), \neg P(z)\}$.

Alternatively, substituting $x/f(a)$, y/a, z/a gives $\{P(f(a)), Q(f(a),a)\}$, $\{\neg P(a), \neg Q(f(a),a)\}$, which resolves to $\{P(f(a)), \neg P(a)\}$. But the previous substitution is more general.

Substitutions

- A *substitution* is a (finite) set $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$, where x_i's are distinct variables, t_i's are terms, and the term t_i is not x_i.
- If all t_i's are ground terms, then σ is a *ground substitution*.
- If all t_i's are distinct variables, then σ is a *renaming of variables*.
- An *expression* is a literal or a term.
- An *instance* of an expression E *by substitution* $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ is the expression $E\sigma$ obtained from E by simultaneous replacing all occurrences of all x_i's for t_i's, respectively.
- For a set S of expressions, let $S\sigma = \{E\sigma \mid E \in S\}$.

Remark *Since we substitute for all variables simultaneously, a possible occurrence of x_i in t_j does not lead to a chain of substitutions.*

For example, for $S = \{P(x), R(y, z)\}$ and $\sigma = \{x/f(y, z), y/x, z/c\}$ we have

$$S\sigma = \{P(f(y, z)), R(x, c)\}.$$

Composing substitutions

For substitutions $\sigma = \{x_1/t_1, \ldots, x_n/t_n\}$ and $\tau = \{y_1/s_1, \ldots, y_n/s_n\}$ we define

$$\sigma\tau = \{x_i/t_i\tau \mid x_i \in X,\ t_i\tau \text{ is not } x_i\} \cup \{y_j/s_j \mid y_j \in Y \setminus X\}$$

to be the *composition* of σ and τ, where $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$.

For example, for $\sigma = \{x/f(y), w/v\}$, $\tau = \{x/a, y/g(x), v/w, u/c\}$ *we have* $\sigma\tau = \{x/f(g(x)), y/g(x), v/w, u/c\}$.

Proposition (without proof) *For every expression* E *and substitutions* σ, τ, ϱ,

(*i*) $(E\sigma)\tau = E(\sigma\tau)$,

(*ii*) $(\sigma\tau)\varrho = \sigma(\tau\varrho)$.

Remark *Composition of substitutions is not commutative, for the above* σ, τ,

$$\tau\sigma = \{x/a, y/g(f(y)), u/c, w/v\} \neq \sigma\tau.$$

Unification

Let $S = \{E_1, \ldots, E_n\}$ be a (finite) set of expressions.

- A *unification* of S is a substitution σ such that $E_1\sigma = E_2\sigma = \cdots = E_n\sigma$, i.e. $S\sigma$ is a singleton.
- S is *unifiable* if it has a unification.
- A unification σ of S is a *most general unification (mgu)* if for every unification τ of S there is a substitution λ such that $\tau = \sigma\lambda$.

For example, $S = \{P(f(x), y), P(f(a), w)\}$ *is unifiable by a most general* unification $\sigma = \{x/a, y/w\}$. A unification $\tau = \{x/a, y/b, w/b\}$ is obtained as $\sigma\lambda$ for $\lambda = \{w/b\}$. τ is not mgu, it cannot give us $\varrho = \{x/a, y/c, w/c\}$.

Observation *If σ, τ are two most general unifications of S, they differ only in renaming of variables.*

Unification algorithm

Let S be a (finite) nonempty set of expressions and p be the leftmost position in which some expressions of S differ. Then *the difference* in S is the set $D(S)$ of subexpressions of all expressions from S starting at the position p.

For example, $S = \{P(x,y), P(f(x),z), P(z,f(x))\}$ *has* $D(S) = \{x, f(x), z\}$.

Input Nonempty (finite) set of expressions S.

Output A most general unification σ of S or *"S is not unifiable"*.

(0) Let $S_0 := S$, $\sigma_0 := \emptyset$, $k := 0$. *(initialization)*

(1) If S_k is a singleton, output the substitution $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$. *(mgu of S)*

(2) Find if $D(S_k)$ contains a variable x and a term t with no occurrence of x.

(3) If not, output *"S is not unifiable"*.

(4) Otherwise, let $\sigma_{k+1} := \{x/t\}$, $S_{k+1} := S_k\sigma_{k+1}$, $k := k+1$ and go to (1).

Remark *The occurrence check of x in t in step (2) can be "expensive".*

Unification algorithm - an example

$$S = \{P(f(y, g(z)), h(b)),\ P(f(h(w), g(a)), t),\ P(f(h(b), g(z)), y)\}$$

1) $S_0 = S$ is not a singleton and $D(S_0) = \{y, h(w), h(b)\}$ has a term $h(w)$ and a variable y not occurring in $h(w)$. Then $\sigma_1 = \{y/h(w)\}$, $S_1 = S_0\sigma_1$, i.e.

$$S_1 = \{P(f(h(w), g(z)), h(b)),\ P(f(h(w), g(a)), t),\ P(f(h(b), g(z)), h(w))\}.$$

2) $D(S_1) = \{w, b\}$, $\sigma_2 = \{w/b\}$, $S_2 = S_1\sigma_2$, i.e.

$$S_2 = \{P(f(h(b), g(z)), h(b)),\ P(f(h(b), g(a)), t)\}.$$

3) $D(S_2) = \{z, a\}$, $\sigma_3 = \{z/a\}$, $S_3 = S_2\sigma_3$, i.e.

$$S_3 = \{P(f(h(b), g(a)), h(b)),\ P(f(h(b), g(a)), t)\}.$$

4) $D(S_3) = \{h(b), t\}$, $\sigma_4 = \{t/h(b)\}$, $S_4 = S_3\sigma_4$, i.e.

$$S_4 = \{P(f(h(b), g(a)), h(b))\}.$$

5) S_4 is a singleton and a most general unification of S is

$$\sigma = \{y/h(w)\}\{w/b\}\{z/a\}\{t/h(b)\} = \{y/h(b), w/b, z/a, t/h(b)\}.$$

Unification algorithm - correctness

Proposition *The unification algorithm outputs a correct answer in finite time for any input S, i.e. a most general unification σ of S or it detects that S is not unifiable.* $(*)$ *Moreover, for every unification τ of S it holds that $\tau = \sigma\tau$.*

Proof It eliminates one variable in each round, so it ends in finite time.

- If it ends negatively after k rounds, $D(S_k)$ is not unifiable, thus also S.
- If it outputs $\sigma = \sigma_0\sigma_1\cdots\sigma_k$, clearly σ is a unification of S.
- If we show the property $(*)$ for σ, then σ is a most general unification of S.

(1) Let τ be a unification of S. We show that $\tau = \sigma_0\sigma_1\cdots\sigma_i\tau$ for all $i \le k$.

(2) For $i = 0$ it holds. Let $\sigma_{i+1} = \{x/t\}$ and assume that $\tau = \sigma_0\sigma_1\cdots\sigma_i\tau$.

(3) It suffices to show that $v\sigma_{i+1}\tau = v\tau$ for every variable v.

(4) If $v \neq x$, $v\sigma_{i+1} = v$, so (3) holds. Otherwise $v = x$ and $v\sigma_{i+1} = x\sigma_{i+1} = t$.

(5) Since τ unifies $S_i = S\sigma_0\sigma_1\cdots\sigma_i$ and both the variable x and the term t are in $D(S_i)$, τ has to unify x and t, i.e. $t\tau = x\tau$, as required for (3). □

The general resolution rule

Let C_1, C_2 be clauses with distinct variables such that

$$C_1 = C_1' \sqcup \{A_1, \ldots, A_n\}, \quad C_2 = C_2' \sqcup \{\neg B_1, \ldots, \neg B_m\},$$

where $S = \{A_1, \ldots, A_n, B_1, \ldots, B_m\}$ is unifiable and $n, m \geq 1$. Then the clause

$$C = C_1'\sigma \cup C_2'\sigma,$$

where σ is a most general unification of S, is the *resolvent* of C_1 and C_2.

For example, in clauses $\{P(x), Q(x,z)\}$ *and* $\{\neg P(y), \neg Q(f(y), y)\}$ *we can* unify $S = \{Q(x,z), Q(f(y), y)\}$ applying a most general unification $\sigma = \{x/f(y), z/y\}$, and then resolve to a clause $\{P(f(y)), \neg P(y)\}$.

Remark *The condition on distinct variables can be satisfied by renaming variables apart. This is sometimes necessary, e.g. from* $\{\{P(x)\}, \{\neg P(f(x))\}\}$ *after renaming we can get* $\square$*, but* $\{P(x), P(f(x))\}$ *is not unifiable.*

Resolution proof

We have the same notions as in propositional logic, up to renaming variables.

- *Resolution proof (deduction)* of a clause C from a formula S is a finite sequence $C_0, \ldots, C_n = C$ such that for every $i \leq n$, we have $C_i = C_i'\sigma$ for some $C_i' \in S$ and a renaming of variables σ, or C_i is a resolvent of some previous clauses.
- A clause C is (resolution) *provable* from S, denoted by $S \vdash_R C$, if it has a resolution proof from S.
- A (resolution) *refutation* of a formula S is a resolution proof of $\square$ from S.
- S is (resolution) *refutable* if $S \vdash_R \square$.

Remark *Elimination of several literals at once is sometimes necessary, e.g.* $S = \{\{P(x), P(y)\}, \{\neg P(x), \neg P(y)\}\}$ *is resolution refutable, but it has no refutation that eliminates only a single literal in each resolution step.*

Resolution in predicate logic - an example

Consider $T = \{\neg P(x,x),\ P(x,y) \rightarrow P(y,x),\ P(x,y) \wedge P(y,z) \rightarrow P(x,z)\}$.

Is $T \models (\exists x)\neg P(x, f(x))$? Equivalently, is the following T' unsatisfiable?

$T' = \{\{\neg P(x,x)\}, \{\neg P(x,y), P(y,x)\}, \{\neg P(x,y), \neg P(y,z), P(x,z)\}, \{P(x, f(x))\}\}$

$T' \vdash_R \square$