Decision Procedures for Verification

First-Order Logic (2)

21.11.2022

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Conventions

In what follows we will use the following conventions:

constants (0-ary function symbols) are denoted with $a, b, c, d, \ldots$

function symbols with arity ≥ 1 are denoted

- $f, g, h, \ldots$ if the formulae are interpreted into arbitrary algebras
- $+, -, s, \ldots$ if the intended interpretation is into numerical domains

predicate symbols with arity 0 are denoted $P, Q, R, S, \ldots$

predicate symbols with arity ≥ 1 are denoted

- $p, q, r, \ldots$ if the formulae are interpreted into arbitrary algebras
- $\leq, \geq, <, >$ if the intended interpretation is into numerical domains

variables are denoted $x, y, z, \ldots$

Until now:

Syntax (one-sorted signatures vs. many-sorted signatures)

Semantics (one-sorted signatures vs. many-sorted signatures)

Validity, Satisfiability, Entailment and Equivalence

2.4 Algorithmic Problems

Validity(F): $\models F$?

Satisfiability(F): F satisfiable?

Entailment(F,G): does F entail G?

Model($\mathcal{A}$,F): $\mathcal{A} \models F$?

Solve($\mathcal{A}$,F): find an assignment β such that $\mathcal{A}, \beta \models F$

Solve(F): find a substitution σ such that $\models F\sigma$

Abduce(F): find G with "certain properties" such that G entails F

Decidability/Undecidability

In 1931, Gödel published his incompleteness theorems in

"Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme"

(in English "On Formally Undecidable Propositions of Principia Mathematica and Related Systems").

He proved for any computable axiomatic system that is powerful enough to describe the arithmetic of the natural numbers (e.g. the Peano axioms or Zermelo-Fraenkel set theory with the axiom of choice), that:

- If the system is consistent, it cannot be complete.
- The consistency of the axioms cannot be proven within the system.

Decidability/Undecidability

These theorems ended a half-century of attempts, beginning with the work of Frege and culminating in Principia Mathematica and Hilbert's formalism, to find a set of axioms sufficient for all mathematics.

The incompleteness theorems also imply that not all mathematical questions are computable.

Consequences of Gödel's Famous Theorems

1. For most signatures Σ, validity is undecidable for Σ-formulas. (One can easily encode Turing machines in most signatures.)
2. For each signature Σ, the set of valid Σ-formulas is recursively enumerable.
   (We will prove this by giving complete deduction systems.)
3. For $\Sigma = \Sigma_{PA}$ and $\mathbb{N}_* = (\mathbb{N}, 0, s, +, *)$, the theory $Th(\mathbb{N}_*)$ is not recursively enumerable.

These undecidability results motivate the study of subclasses of formulas (fragments) of first-order logic

Q: Can you think of any fragments of first-order logic for which validity is decidable?

Some Decidable Fragments/Problems

Validity/Satisfiability/Entailment: Some decidable fragments:

- Variable-free formulas without equality: satisfiability is NP-complete. (why?)
- Variable-free Horn clauses (clauses with at most one positive atom): entailment is decidable in linear time.
- Monadic class: no function symbols, all predicates unary; validity is NEXPTIME-complete.
- Q: Other decidable fragments of FOL (with variables)? Which methods for proving decidability?

Decidable problems.

Finite model checking is decidable in time polynomial in the size of the structure and the formula.

Goals

Identify:

- decidable fragments of first-order logic
- fragments of FOL for which satisfiability checking is easy

Methods:

- Theoretical methods (automata theory, finite model property)
- Adjust automated reasoning techniques
  (e.g. to obtaining efficient decision procedures)

  Extend methods for automated reasoning in propositional logic?

    Instantiation/reduction to propositional logic

    Extend the resolution calculus for first-order logic

Goals

Extend methods for automated reasoning in propositional logic?

Instantiation/reduction to propositional logic

Extend the resolution calculus for first-order logic

Ingredients:

- Give a method for translating formulae to clause form
- Regard formulae with variables as a set of all their instances (where variables are instantiated with ground terms)
  - Show that only certain instances are needed
    $\mapsto$ reduction to propositional logic
  - Finite encoding of infinitely many inferences
    $\mapsto$ resolution for first-order logic

Goals

Identify:

- decidable fragments of first-order logic
- fragments of FOL for which satisfiability checking is easy

- Other decidable logical theories and theory fragments

Today

- Logical theories - Definition, Examples
- Normal forms $\mapsto$ Resolution for first-order logic

Today

- Logical theories - Definition, Examples
- Normal forms $\mapsto$ Resolution for first-order logic

Theory of a Structure

Let $\mathcal{A} \in \Sigma$-alg. The (first-order) theory of $\mathcal{A}$ is defined as

$$Th(\mathcal{A}) = \{G \in \mathsf{F}_\Sigma(X) \mid \mathcal{A} \models G\}$$

Problem of axiomatizability:

For which structures $\mathcal{A}$ can one axiomatize $Th(\mathcal{A})$, that is, can one write down a formula F (or a recursively enumerable set F of formulas) such that

$$Th(\mathcal{A}) = \{G \mid F \models G\}?$$

Analogously for sets of structures.

Two Interesting Theories

Let $\Sigma_{Pres} = (\{0/0, s/1, +/2\},\ \emptyset)$ and $\mathbb{Z}_+ = (\mathbb{Z}, 0, s, +)$ its standard interpretation on the integers.
$Th(\mathbb{Z}_+)$ is called Presburger arithmetic (M. Presburger, 1929).
(There is no essential difference when one, instead of $\mathbb{Z}$, considers the natural numbers $\mathbb{N}$ as standard interpretation.)

Presburger arithmetic is decidable in 3EXPTIME (D. Oppen, JCSS, 16(3):323–332, 1978), and in 2EXPSPACE, using automata-theoretic methods (and there is a constant $c \geq 0$ such that $Th(\mathbb{Z}_+) \notin \text{NTIME}(2^{2^{cn}})$).

Two Interesting Theories

However, $\mathbb{N}_* = (\mathbb{N}, 0, s, +, *)$, the standard interpretation of $\Sigma_{PA} = (\{0/0, s/1, +/2, */2\}, \emptyset)$, has as theory the so-called Peano arithmetic which is undecidable, not even recursively enumerable.

Note: The choice of signature can make a big difference with regard to the computational complexity of theories.

Logical theories

Syntactic view

first-order theory: given by a set $\mathcal{F}$ of (closed) first-order Σ-formulae.

the models of $\mathcal{F}$: $\quad \mathsf{Mod}(\mathcal{F}) = \{\mathcal{A} \in \Sigma\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{F}\}$

Semantic view

given a class $\mathcal{M}$ of Σ-algebras

the first-order theory of $\mathcal{M}$: $\mathsf{Th}(\mathcal{M}) = \{G \in F_\Sigma(X) \text{ closed} \mid \mathcal{M} \models G\}$

Theories

$\mathcal{F}$ set of (closed) first-order formulae

$$\text{Mod}(\mathcal{F}) = \{A \in \Sigma\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{F}\}$$

$\mathcal{M}$ class of Σ-algebras

$$\text{Th}(\mathcal{M}) = \{G \in F_\Sigma(X) \text{ closed} \mid \mathcal{M} \models G\}$$

$\text{Th}(\text{Mod}(\mathcal{F}))$ the set of formulae true in all models of $\mathcal{F}$
represents exactly the set of consequences of $\mathcal{F}$

Theories

$\mathcal{F}$ set of (closed) first-order formulae

$$\mathrm{Mod}(\mathcal{F}) = \{A \in \Sigma\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{F}\}$$

$\mathcal{M}$ class of Σ-algebras

$$\mathrm{Th}(\mathcal{M}) = \{G \in F_\Sigma(X) \text{ closed} \mid \mathcal{M} \models G\}$$

$\mathrm{Th}(\mathrm{Mod}(\mathcal{F}))$ the set of formulae true in all models of $\mathcal{F}$
represents exactly the set of consequences of $\mathcal{F}$

Note: $\mathcal{F} \subseteq \mathrm{Th}(\mathrm{Mod}(\mathcal{F}))$ (typically strict)

$\mathcal{M} \subseteq \mathrm{Mod}(\mathrm{Th}(\mathcal{M}))$ (typically strict)

Examples

1. Groups

Let $\Sigma = (\{e/0, */2, i/1\}, \emptyset)$

Let $\mathcal{F}$ consist of all (universally quantified) group axioms:

$$\begin{array}{ll} \forall x, y, z & x * (y * z) \approx (x * y) * z \\ \forall x & x * i(x) \approx e \quad \wedge \quad i(x) * x \approx e \\ \forall x & x * e \approx x \quad \wedge \quad e * x \approx x \end{array}$$

Every group $\mathcal{G} = (G, e_G, *_G, i_G)$ is a model of $\mathcal{F}$

$\text{Mod}(\mathcal{F})$ is the class of all groups

$\mathcal{F} \subset \text{Th}(\text{Mod}(\mathcal{F}))$

Examples

2. Linear (positive)integer arithmetic

Let $\Sigma = (\{0/0, s/1, +/2\}, \{\leq /2\})$

Let $\mathbb{Z}_+ = (\mathbb{Z}, 0, s, +, \leq)$ the standard interpretation of integers.

$\{\mathbb{Z}_+\} \subset \mathrm{Mod}(\mathrm{Th}(\mathbb{Z}_+))$

3. Uninterpreted function symbols

Let $\Sigma = (\Omega, \Pi)$ be arbitrary

Let $\mathcal{M} = \Sigma$-alg be the class of all Σ-structures

The theory of uninterpreted function symbols is Th(Σ-alg) the family of all first-order formulae which are true in all Σ-algebras.

Examples

4. Lists

Let $\Sigma = (\{\text{car}/1, \text{cdr}/1, \text{cons}/2\}, \emptyset)$

Let $\mathcal{F}$ be the following set of list axioms:

$$\begin{aligned} \text{car}(\text{cons}(x, y)) &\approx x \\ \text{cdr}(\text{cons}(x, y)) &\approx y \\ \text{cons}(\text{car}(x), \text{cdr}(x)) &\approx x \end{aligned}$$

$\text{Mod}(\mathcal{F})$ class of all models of $\mathcal{F}$

$\text{Th}_{\text{Lists}} = \text{Th}(\text{Mod}(\mathcal{F}))$ theory of lists (axiomatized by $\mathcal{F}$)

Goals

Identify:

- decidable theories; decidable fragments of first-order logic
- fragments of FOL for which satisfiability checking is easy

Methods:

- Adjust automated reasoning techniques
  (e.g. to obtaining efficient decision procedures)

  Extend methods for automated reasoning in propositional logic?

  Instantiation/reduction to propositional logic

  Extend the resolution calculus for first-order logic

Today

- Logical theories - Definition, Examples
- Normal forms $\mapsto$ Resolution for first-order logic

2.5 Normal Forms and Skolemization

Study of normal forms motivated by

- reduction of logical concepts,
- efficient data structures for theorem proving.

The main problem in first-order logic is the treatment of quantifiers. The subsequent normal form transformations are intended to eliminate many of them.

Prenex Normal Form

Prenex formulas have the form

$$Q_1 x_1 \ldots Q_n x_n \; F,$$

where F is quantifier-free and $Q_i \in \{\forall, \exists\}$;
we call $Q_1 x_1 \ldots Q_n x_n$ the quantifier prefix and F the matrix of the formula.

Prenex Normal Form

Computing prenex normal form by the rewrite relation $\Rightarrow_P$:

$$
\begin{array}{rcll}
(F \leftrightarrow G) & \Rightarrow_P & (F \rightarrow G) \wedge (G \rightarrow F) & \\
\neg QxF & \Rightarrow_P & \overline{Q}x\neg F & (\neg Q) \\
(QxF \ \rho \ G) & \Rightarrow_P & Qy(F[y/x] \ \rho \ G),\ y \text{ fresh},\ \rho \in \{\wedge, \vee\} & \\
(QxF \rightarrow G) & \Rightarrow_P & \overline{Q}y(F[y/x] \rightarrow G),\ y \text{ fresh} & \\
(F \ \rho \ QxG) & \Rightarrow_P & Qy(F \ \rho \ G[y/x]),\ y \text{ fresh},\ \rho \in \{\wedge, \vee, \rightarrow\} &
\end{array}
$$

Here $\overline{Q}$ denotes the quantifier dual to Q, i.e., $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$.

Example

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \to ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

Example

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x' \, ((p(x') \vee q(x',y)) \wedge \exists z\, r(x',y,z)) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

Example

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'((p(x') \vee q(x',y)) \wedge \exists z\, r(x',y,z)) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'(\exists z'((p(x') \vee q(x',y)) \wedge r(x',y,z'))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y$$

Example

$$F := (\forall x((p(x) \vee q(x, y)) \wedge \exists z\, r(x, y, z))) \rightarrow ((p(z) \wedge q(x, z)) \wedge \forall z\, r(z, x, y))$$

$$\Rightarrow_P \quad \exists x'((p(x') \vee q(x', y)) \wedge \exists z\, r(x', y, z))) \rightarrow ((p(z) \wedge q(x, z)) \wedge \forall z\, r(z, x, y))$$

$$\Rightarrow_P \quad \exists x'(\exists z'((p(x') \vee q(x', y)) \wedge r(x', y, z'))) \rightarrow ((p(z) \wedge q(x, z)) \wedge \forall z\, r(z, x, y$$

$$\Rightarrow_P \quad \exists x'\forall z'\,(((p(x') \vee q(x', y)) \wedge r(x', y, z'))) \rightarrow ((p(z) \wedge q(x, z)) \wedge \forall z\, r(z, x, y$$

Example

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'((p(x') \vee q(x',y)) \wedge \exists z\, r(x',y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'(\exists z'((p(x') \vee q(x',y)) \wedge r(x',y,z'))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y$$

$$\Rightarrow_P \;\; \exists x' \forall z' \, ((p(x') \vee q(x',y)) \wedge r(x',y,z')) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x' \forall z' \, ((p(x') \vee q(x',y)) \wedge r(x',y,z')) \rightarrow \forall z''((p(z) \wedge q(x,z)) \wedge r(z'',x,$$

Example

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'((p(x') \vee q(x',y)) \wedge \exists z\, r(x',y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x'(\exists z'((p(x') \vee q(x',y)) \wedge r(x',y,z'))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y$$

$$\Rightarrow_P \;\; \exists x' \forall z'\, ((p(x') \vee q(x',y)) \wedge r(x',y,z')) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

$$\Rightarrow_P \;\; \exists x' \forall z'\, ((p(x') \vee q(x',y)) \wedge r(x',y,z')) \rightarrow \forall z''((p(z) \wedge q(x,z)) \wedge r(z'',x,$$

$$\Rightarrow_P \;\; \exists x' \forall z' \forall z''(((p(x') \vee q(x',y)) \wedge r(x',y,z')) \rightarrow ((p(z) \wedge q(x,z)) \wedge r(z'',x,$$

Skolemization

Intuition: remove $\exists y$.

For this:

- we introduce a concrete choice function sk_y computing y from all the arguments y depends on (i.e. from all variables $x_1, \ldots, x_n$ which occur universally quantified before y in the quantifier prefix)
- We replace y with $\mathrm{sk}_y(x_1, \ldots, x_n)$ everywhere in the scope of $\exists y$.

Transformation $\Rightarrow_S$ (to be applied outermost, *not* in subformulas):

$$\forall x_1, \ldots, x_n \exists y F \quad \Rightarrow_S \quad \forall x_1, \ldots, x_n F[\mathrm{sk}_y(x_1, \ldots, x_n)/y]$$

where sk_y/n is a new function symbol (Skolem function).

Skolemization

Together: $F \overset{*}{\Rightarrow}_P \underbrace{G}_{\text{prenex}} \overset{*}{\Rightarrow}_S \underbrace{H}_{\text{prenex, no } \exists}$

Theorem 2.9:

Let F, G, and H as defined above and closed. Then

(i) F and G are equivalent.

(ii) $H \models G$ but the converse is not true in general.

(iii) G satisfiable (wrt. Σ-alg) $\Leftrightarrow$ H satisfiable (wrt. Σ'-Alg) where $\Sigma' = (\Omega \cup SKF, \Pi)$, if $\Sigma = (\Omega, \Pi)$.

Clausal Normal Form (Conjunctive Normal Form)

$$
\begin{array}{rcl}
(F \leftrightarrow G) & \Rightarrow_K & (F \rightarrow G) \wedge (G \rightarrow F) \\
(F \rightarrow G) & \Rightarrow_K & (\neg F \vee G) \\
\neg(F \vee G) & \Rightarrow_K & (\neg F \wedge \neg G) \\
\neg(F \wedge G) & \Rightarrow_K & (\neg F \vee \neg G) \\
\neg\neg F & \Rightarrow_K & F \\
(F \wedge G) \vee H & \Rightarrow_K & (F \vee H) \wedge (G \vee H) \\
(F \wedge \top) & \Rightarrow_K & F \\
(F \wedge \bot) & \Rightarrow_K & \bot \\
(F \vee \top) & \Rightarrow_K & \top \\
(F \vee \bot) & \Rightarrow_K & F
\end{array}
$$

These rules are to be applied modulo associativity and commutativity of $\wedge$ and $\vee$. The first five rules, plus the rule $(\neg Q)$, compute the negation normal form (NNF) of a formula.

The Complete Picture

$$
\begin{array}{llll}
F & \stackrel{*}{\Rightarrow}_P & Q_1 y_1 \ldots Q_n y_n \; G & (G \text{ quantifier-free}) \\
 & \stackrel{*}{\Rightarrow}_S & \forall x_1, \ldots, x_m \; H & (m \leq n,\ H \text{ quantifier-free}) \\
 & \stackrel{*}{\Rightarrow}_K & \underbrace{\underbrace{\forall x_1, \ldots, x_m}_{\text{leave out}} \; \bigwedge_{i=1}^{k} \; \underbrace{\bigvee_{j=1}^{n_i} L_{ij}}_{\text{clauses } C_i}}_{F'} &
\end{array}
$$

$N = \{C_1, \ldots, C_k\}$ is called the clausal (normal) form (CNF) of F.

Note: the variables in the clauses are implicitly universally quantified.

Theorem 2.10:

Let F be closed. Then $F' \models F$. (The converse is not true in general.)

Theorem 2.11:

Let F be closed. Then F is satisfiable iff F' is satisfiable iff N is satisfiable

Example

Given: $\exists u \forall w (\exists x (p(w, x, u) \vee \forall y (q(w, x, y) \wedge \exists z\, r(y, z))))$

Example

Given: $\exists u \forall w (\exists x (p(w,x,u) \vee \forall y\, (q(w,x,y) \wedge \exists z\, r(y,z))))$

Prenex Normal Form:

$$\stackrel{*}{\Rightarrow}_P \; \exists u \forall w \exists x \forall y \exists z ((p(w,x,u) \vee (q(w,x,y) \wedge r(y,z))))$$

Example

Given: $\exists u \forall w (\exists x (p(w, x, u) \vee \forall y\, (q(w, x, y) \wedge \exists z\, r(y, z))))$

Prenex Normal Form:

$$\stackrel{*}{\Rightarrow}_P \ \exists u \forall w \exists x \forall y \exists z ((p(w, x, u) \vee (q(w, x, y) \wedge r(y, z))))$$

Skolemisation:

$$\stackrel{*}{\Rightarrow}_S \ \forall w \forall y ((p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee (q(w, \mathrm{sk}_x(w), y) \wedge r(y, \mathrm{sk}_z(w, y)))))$$

Example

Given: $\exists u \forall w (\exists x (p(w, x, u) \vee \forall y\, (q(w, x, y) \wedge \exists z\, r(y, z))))$

Prenex Normal Form:

$$\overset{*}{\Rightarrow}_P \; \exists u \forall w \exists x \forall y \exists z ((p(w, x, u) \vee (q(w, x, y) \wedge r(y, z))))$$

Skolemisation:

$$\overset{*}{\Rightarrow}_S \; \forall w \forall y ((p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee (q(w, \mathrm{sk}_x(w), y) \wedge r(y, \mathrm{sk}_z(w, y)))))$$

Clause normal form:

$$\overset{*}{\Rightarrow}_K \; \forall w \forall y [(p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee q(w, \mathrm{sk}_x(w), y)) \wedge$$
$$(p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee r(y, \mathrm{sk}_y(w, y)))]$$

Set of clauses:

$$\{p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee q(w, \mathrm{sk}_x(w), y), \quad p(w, \mathrm{sk}_x(w), \mathrm{sk}_u) \vee r(y, \mathrm{sk}_y(w, y))\}$$

Optimization

Here is lots of room for optimization since we only can preserve satisfiability anyway:

- size of the CNF exponential when done naively;
- want to preserve the original formula structure;
- want small arity of Skolem functions.

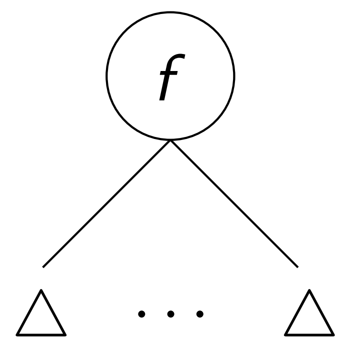
2.6 Herbrand Interpretations

From now an we shall consider PL without equality. Ω shall contains at least one constant symbol.

A Herbrand interpretation (over Σ) is a Σ-algebra $\mathcal{A}$ such that

- $U_{\mathcal{A}} = \mathsf{T}_{\Sigma}$ (= the set of ground terms over Σ)
- $f_{\mathcal{A}} : (s_1, \ldots, s_n) \mapsto f(s_1, \ldots, s_n), \; f/n \in \Omega$

$f_{\mathcal{A}}(\triangle, \ldots, \triangle) =$



Herbrand Interpretations

In other words, *values are fixed* to be ground terms and *functions are fixed* to be the term constructors. Only predicate symbols $p/m \in \Pi$ may be freely interpreted as relations $p_{\mathcal{A}} \subseteq \mathsf{T}_{\Sigma}^{m}$.

Proposition 2.12

Every set of ground atoms I uniquely determines a Herbrand interpretation $\mathcal{A}$ via

$$(s_1, \ldots, s_n) \in p_{\mathcal{A}} \quad :\Leftrightarrow \quad p(s_1, \ldots, s_n) \in I$$

Thus we shall identify Herbrand interpretations (over Σ) with sets of Σ-ground atoms.

Herbrand Interpretations

Example: $\Sigma_{Pres} = (\{0/0, s/1, +/2\},\ \{</2, \leq/2\})$

$\mathbb{N}$ as Herbrand interpretation over Σ_{Pres}:

$$
\begin{aligned}
I = \{\quad & 0 \leq 0,\ 0 \leq s(0),\ 0 \leq s(s(0)),\ \ldots, \\
& 0 + 0 \leq 0,\ 0 + 0 \leq s(0),\ \ldots, \\
& \ldots,\ (s(0) + 0) + s(0) \leq s(0) + (s(0) + s(0)) \\
& \ldots \\
& s(0) + 0 < s(0) + 0 + 0 + s(0) \\
& \ldots\}
\end{aligned}
$$

Existence of Herbrand Models

A Herbrand interpretation I is called a Herbrand model of F, if $I \models F$.

Theorem 2.13

Let N be a set of Σ-clauses.

$$\begin{aligned} N \text{ satisfiable} \quad &\Leftrightarrow \quad N \text{ has a Herbrand model (over } \Sigma) \\ &\Leftrightarrow \quad G_\Sigma(N) \text{ has a Herbrand model (over } \Sigma) \end{aligned}$$

where $G_\Sigma(N) = \{C\sigma \text{ ground clause} \mid C \in N,\ \sigma : X \to \mathsf{T}_\Sigma\}$ is the set of ground instances of N.

(Proof – completeness proof of resolution for first-order logic.)

Example of a G_Σ

For Σ_{Pres} one obtains for

$$C = (x < y) \vee (y \leq s(x))$$

the following ground instances:

$(0 < 0) \vee (0 \leq s(0))$

$(s(0) < 0) \vee (0 \leq s(s(0)))$

$\ldots$

$(s(0) + s(0) < s(0) + 0) \vee (s(0) + 0 \leq s(s(0) + s(0)))$

$\ldots$

Consequences of Herbrans's theorem

Decidability results.

- Formulae without function symbols and without equality

  The Bernays-Schönfinkel Class $\exists^*\forall^*$

The Bernays-Schönfinkel Class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Bernays-Schönfinkel class consists only of sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m F(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

The Bernays-Schönfinkel Class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Bernays-Schönfinkel class consists only of sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m F(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Idea: CNF translation:

$$\begin{aligned}
&\exists \overline{x}_1 \forall \overline{y}_1 F_1 \wedge \ldots \exists \overline{x}_n \forall \overline{y}_n F_n \\
&\qquad \Rightarrow_P \exists \overline{x}_1 \ldots \exists \overline{x}_n \forall \overline{y}_1 \ldots \forall \overline{y}_n F(\overline{x}_1, \ldots, \overline{x}_n, \overline{y}_1, \ldots, \overline{y}_n) \\
&\qquad \Rightarrow_S \forall \overline{y}_1 \ldots \forall \overline{y}_m F(\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n) \\
&\qquad \Rightarrow_K \forall \overline{y}_1 \ldots \forall \overline{y}_m \bigwedge \bigvee L_i((\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n)
\end{aligned}$$

$\overline{c}_1, \ldots, \overline{c}_n$ are tuples of Skolem constants

The Bernays-Schönfinkel Class

$\Sigma = (\Omega, \Pi)$, Ω is a finite set of constants

The Bernays-Schönfinkel class consists only of sentences of the form

$$\exists x_1 \ldots \exists x_n \forall y_1 \ldots \forall y_m F(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

Idea: CNF translation:

$$\begin{aligned}&\exists \overline{x}_1 \forall \overline{y}_1 F_1 \wedge \ldots \exists \overline{x}_n \forall \overline{y}_n F_n \\ &\qquad \Rightarrow^*_K \forall \overline{y}_1 \ldots \forall \overline{y}_m \bigwedge \bigvee L_i((\overline{c}_1, \ldots, \overline{c}_n, \overline{y}_1, \ldots, \overline{y}_n)\end{aligned}$$

$\overline{c}_1, \ldots, \overline{c}_n$ are tuples of Skolem constants

The Herbrand Universe is finite $\mapsto$ decidability