Decision Procedures in Verification

First-Order Logic (3)

26.11.2012

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Until now:

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

Semantics

Models, Validity, and Satisfiability

Entailment and Equivalence

Logical theories

Syntactic view: axioms $\mathcal{F}$ of (closed) first-order Σ-formulae.
$\mathrm{Mod}(\mathcal{F}) = \{\mathcal{A} \in \Sigma\text{-alg} \mid \mathcal{A} \models G, \text{ for all } G \text{ in } \mathcal{F}\}$

Semantic view: class $\mathcal{M}$ of Σ-algebras
$\mathrm{Th}(\mathcal{M}) = \{G \in F_\Sigma(X) \text{ closed} \mid \mathcal{M} \models G\}$

Algorithmic Problems; Decidability, Undecidability

Today:

Proving (un-)satisfiability of first-order formulas.

- Normal Forms
- Unification General Resolution
- Theorems of Herbrand and Löwenheim/Skolem
- Ordered Resolution with Selection

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.

Format often required:

$$\forall x_1 \ldots \forall x_n \; (L_{11} \vee \ldots \vee L_{1k}) \wedge \ldots \wedge (L_{n1} \vee \ldots \vee L_{nl})$$

Prenex Normal Form

Prenex formulas have the form

$$Q_1 x_1 \dots Q_n x_n \ F,$$

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

Prenex Normal Form

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

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

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

Example

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

Example

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

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

Example

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

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

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

Example

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

$$\begin{aligned}
&\Rightarrow_P \;\; \exists x'((p(x') \vee q(x',y)) \wedge \exists z\, r(x',y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) \\
&\Rightarrow_P \;\; \exists x'(\exists z'((p(x') \vee q(x',y)) \wedge r(x',y,z'))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) \\
&\Rightarrow_P \;\; \exists x' \forall z'\, (((p(x') \vee q(x',y)) \wedge r(x',y,z'))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))
\end{aligned}$$

Example

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

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

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

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

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

Example

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

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

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

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

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

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

Skolemization

Intuition: replacement of $\exists y$ by a concrete choice function computing y from all the arguments y depends on.

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

$$\forall x_1, \ldots, x_n \exists y F \quad \Rightarrow_S \quad \forall x_1, \ldots, x_n F[f(x_1, \ldots, x_n)/y]$$

where f/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

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

$N = \{C_1, \ldots, C_k\}$ 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:

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

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:

$$\Rightarrow_P^* \; \exists u \forall w \exists x \forall y \exists z ((p(w,x,u) \vee (q(w,x,y) \wedge r(y,z))))$$

Skolemisation:

$$\Rightarrow_S^* \; \forall w \forall y ((p(w, sk_x(w), sk_u) \vee (q(w, sk_x(w), y) \wedge r(y, g(w,y)))))$$

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:

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

Skolemisation:

$$\Rightarrow^*_S \; \forall w \forall y ((p(w, sk_x(w), sk_u) \vee (q(w, sk_x(w), y) \wedge r(y, g(w,y)))))$$

Clause normal form:

$$\Rightarrow^*_K \; \forall w \forall y [(p(w, sk_x(w), sk_u) \vee q(w, sk_x(w), y)) \wedge (p(w, sk_x(w), sk_u) \vee r(y, g(w,y)))]$$

Set of clauses:

$$\{p(w, sk_x(w), sk_u) \vee q(w, sk_x(w), y), \quad p(w, sk_x(w), sk_u) \vee r(y, g(w,y))\}$$

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:

$$\forall x \exists y . p(x) \vee q(y) \longleftarrow (\forall x\ p(x)) \vee (\exists y\ q(y)) \longrightarrow \exists y \forall x\ p(x) \vee q(y)$$

2.6 General Resolution

Propositional resolution:

refutationally complete,

clearly inferior to the DPLL procedure
(even with various improvements).

But: in contrast to the DPLL procedure, resolution can be easily extended to non-ground clauses.

Propositional resolution

Resolution inference rule:

$$\frac{C \vee A \qquad \neg A \vee D}{C \vee D}$$

Terminology: $C \vee D$: resolvent; A: resolved atom

(Positive) factorisation inference rule:

$$\frac{C \vee A \vee A}{C \vee A}$$

Resolution for ground clauses

- Exactly the same as for propositional clauses

  Ground atoms $\mapsto$ propositional variables

Theorem

Res is sound and refutationally complete (for all sets of ground clauses)

Sample Refutation

1.	$\neg P(f(a)) \vee \neg P(f(a)) \vee Q(b)$	(given)
2.	$P(f(a)) \vee Q(b)$	(given)
3.	$\neg P(g(b, a)) \vee \neg Q(b)$	(given)
4.	$P(g(b, a))$	(given)
5.	$\neg P(f(a)) \vee Q(b) \vee Q(b)$	(Res. 2. into 1.)
6.	$\neg P(f(a)) \vee Q(b)$	(Fact. 5.)
7.	$Q(b) \vee Q(b)$	(Res. 2. into 6.)
8.	$Q(b)$	(Fact. 7.)
9.	$\neg P(g(b, a))$	(Res. 8. into 3.)
10.	$\perp$	(Res. 4. into 9.)

Resolution for ground clauses

- Refinements with orderings and selection functions:

  Need: - well-founded ordering on ground atomic formulae/literals

  - selection function (for negative literals)

    $S : C \quad \mapsto \quad$ set of occurrences of *negative* literals in C

  Example of selection with selected literals indicated as $\boxed{X}$:

  $\boxed{\neg A} \vee \neg A \vee B$

  $\boxed{\neg B_0} \vee \boxed{\neg B_1} \vee A$

Resolution Calculus $Res_S^{\succ}$

Ordered resolution with selection

$$\frac{C \lor A \qquad \neg A \lor D}{C \lor D}$$

if

1. $A \succ C$;
2. nothing is selected in C by S;
3. $\neg A$ is selected in $D \lor \neg A$,
   or else nothing is selected in $D \lor \neg A$ and $\neg A \succeq \max(D)$.

Note: For positive literals, $A \succ C$ is the same as $A \succ \max(C)$.

Ordered factoring

$$\frac{C \lor A \lor A}{(C \lor A)}$$

if A is maximal in C and nothing is selected in C.

Resolution for ground clauses

Let $\succ$ be a total and well-founded ordering on ground atoms, and S a selection function.

Theorem. $\text{Res}_S^{\succ}$ is sound and refutationally complete for all sets of ground clauses.

Soundness: sufficient to show that

(1) $C \vee A, \neg A \vee D \models C \vee D$

(2) $C \vee A \vee A \models C \vee A$

Completeness: Let $\succ$ be a clause ordering, let N be saturated wrt. $Res_S^{\succ}$, and suppose that $\bot \notin N$. Then $I_N^{\succ} \models N$, where $I_N^{\succ}$ is incrementally constructed as follows:

Construction of Candidate Models Formally

Let $N, \succ$ be given.

- Order N increasing w.r.t. the extension of $\succ$ to clauses.
- Define sets I_C and Δ_C for all ground clauses C over the given signature inductively over $\succ$:

$$I_C := \bigcup_{C \succ D} \Delta_D$$

$$\Delta_C := \begin{cases} \{A\}, & \text{if } C \in N,\ C = C' \vee A,\ A \succ C',\ I_C \not\models C \\ & \text{and nothing is selected in } C \\ \emptyset, & \text{otherwise} \end{cases}$$

We say that C produces A, if $\Delta_C = \{A\}$.

The candidate model for N (wrt. $\succ$) is given as $I_N^{\succ} := \bigcup_C \Delta_C$.

(We write I_N for $I_N^{\succ}$ if $\succ$ is irrelevant or known from the context.)

Completeness

Theorem. Let $\succ$ be a clause ordering, let N be saturated wrt. $Res_S^{\succ}$, and suppose that $\bot \notin N$. Then $I_N^{\succ} \models N$.

Proof: Suppose $\bot \notin N$, but $I_N^{\succ} \not\models N$. Let $C \in N$ minimal (in $\succ$) such that $I_N^{\succ} \not\models C$. Since C is false in I_N, C is not productive. As $C \neq \bot$ there exists a maximal atom A in C.

1. $C = \neg A \vee C'$ (maximal atom occurs negatively) $\Rightarrow I_N \models A,\ I_N \not\models C'$
   Then some $D = D' \vee A \in N$ produces A. As $\frac{D' \vee A \qquad \neg A \vee C'}{D' \vee C'}$, we infer that $D' \vee C' \in N$, and $C \succ D' \vee C'$ and $I_N \not\models D' \vee C'$
   $\Rightarrow$ contradicts minimality of C.
2. $C = \boxed{\neg A} \vee C'$ ($\neg A$ is selected) $\Rightarrow I_N \models A,\ I_N \not\models C'$
   The argument in 1. applies also in this case.
3. $C = C' \vee A \vee A$. Then $\frac{C' \vee A \vee A}{C' \vee A}$ yields a smaller counterexample $C' \vee A \in N$. $\Rightarrow$ contradicts minimality of C.

General Resolution through Instantiation

Idea: instantiate clauses appropriately:

$$
\begin{tikzcd}
P(z',z') \vee \neg Q(z) \arrow[d, "{[a/z', f(a,b)/z]}"'] & & & \neg P(a,y) \arrow[dl, "{[a/y]}"'] \arrow[dr, "{[b/y]}"] & & & P(x',b) \vee Q(f(x',x)) \arrow[d, "{[a/x', b/x]}"] \\
P(a,a) \vee \neg Q(f(a,b)) \arrow[dr] & & \neg P(a,a) \arrow[dl] & & \neg P(a,b) \arrow[dr] & & P(a,b) \vee Q(f(a,b)) \arrow[dl] \\
& \neg Q(f(a,b)) \arrow[drr] & & & & Q(f(a,b)) \arrow[dll] & \\
& & & \perp & & &
\end{tikzcd}
$$

General Resolution through Instantiation

Problems:

More than one instance of a clause can participate in a proof.

Even worse: There are infinitely many possible instances.

Observation:

Instantiation must produce complementary literals
(so that inferences become possible).

Idea:

Do not instantiate more than necessary to get complementary literals.

General Resolution through Instantiation

Idea: do not instantiate more than necessary:

$$
\begin{tikzcd}
P(z',z') \vee \neg Q(z) \arrow[d, "{[a/z']}"'] & & & \neg P(a,y) \arrow[dl, "{[a/y]}"'] \arrow[dr, "{[b/y]}"] & & & P(x',b) \vee Q(f(x',x)) \arrow[d, "{[a/x']}"] \\
P(a,a) \vee \neg Q(z) \arrow[dr] & & \neg P(a,a) \arrow[dl] & & \neg P(a,b) \arrow[dr] & & P(a,b) \vee Q(f(a,x)) \arrow[dl] \\
& \neg Q(z) \arrow[d, "{[f(a,x)/z]}"'] & & & & Q(f(a,x)) \arrow[d] & \\
& \neg Q(f(a,x)) \arrow[drr] & & & & Q(f(a,x)) \arrow[dll] & \\
& & & \bot & & &
\end{tikzcd}
$$

Lifting Principle

Problem: Make saturation of infinite sets of clauses as they arise from taking the (ground) instances of finitely many general clauses (with variables) effective and efficient.

Idea (Robinson 65):

- Resolution for general clauses:
- *Equality* of ground atoms is generalized to *unifiability* of general atoms;
- Only compute *most general* (minimal) unifiers.

Lifting Principle

Significance: The advantage of the method in (Robinson 65) compared with (Gilmore 60) is that unification enumerates only those instances of clauses that participate in an inference.

Moreover, clauses are not right away instantiated into ground clauses. Rather they are instantiated only as far as required for an inference.

Inferences with non-ground clauses in general represent infinite sets of ground inferences which are computed simultaneously in a single step.

Resolution for General Clauses

General binary resolution *Res***:**

$$\frac{C \vee A \qquad \neg B \vee D}{(C \vee D)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \qquad \text{[resolution]}$$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \quad \text{[factorization]}$$

For inferences with more than one premise, we assume that the variables in the premises are (bijectively) renamed such that they become different to any variable in the other premises.
We do not formalize this. Which names one uses for variables is otherwise irrelevant.

Unification

Let $E = \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\}$ (s_i, t_i terms or atoms) a multi-set of equality problems. A substitution σ is called a unifier of E if $s_i\sigma = t_i\sigma$ for all $1 \leq i \leq n$.

If a unifier of E exists, then E is called unifiable.

Unification after Martelli/Montanari

$$
\begin{array}{llcl}
(1) & t \doteq t, E & \Rightarrow_{MM} & E \\
(2) & f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E & \Rightarrow_{MM} & s_1 \doteq t_1, \ldots, s_n \doteq t_n, E \\
(3) & f(\ldots) \doteq g(\ldots), E & \Rightarrow_{MM} & \bot \\
(4) & x \doteq t, E & \Rightarrow_{MM} & x \doteq t, E[t/x] \\
 & & & \quad \text{if } x \in var(E), x \notin var(t) \\
(5) & x \doteq t, E & \Rightarrow_{MM} & \bot \\
 & & & \quad \text{if } x \neq t, x \in var(t) \\
(6) & t \doteq x, E & \Rightarrow_{MM} & x \doteq t, E \\
 & & & \quad \text{if } t \notin X
\end{array}
$$

Examples

Example 1:

$$
\begin{array}{ll}
 & \{x \doteq f(a), \underline{g(x,x) \doteq g(x,y)}\} \\
\Rightarrow_4 & \{x \doteq f(a), \underline{g(f(a),f(a)) \doteq g(f(a),y)}\} \\
\Rightarrow_2 & \{x \doteq f(a), \underline{f(a) \doteq f(a)}, f(a) \doteq y\} \\
\Rightarrow_1 & \{x \doteq f(a), \underline{f(a) \doteq y}\} \\
\Rightarrow_6 & \{x \doteq f(a), y \doteq f(a)\}
\end{array}
$$

Example 2:

$$\{x \doteq f(a), \underline{g(x,x) \doteq h(x,y)}\} \quad \Rightarrow_3 \quad \bot$$

Example 3:

$$
\begin{array}{ll}
 & \{\underline{f(x,x) \doteq f(y,g(y))}\} \\
\Rightarrow_2 & \{\underline{x \doteq y}, x \doteq g(y)\} \\
\Rightarrow_4 & \{x \doteq y, \underline{y \doteq g(y)}\} \qquad \Rightarrow_5 \quad \bot
\end{array}
$$

Martelli/Montanari: Main Properties

If $E = x_1 \doteq u_1, \ldots, x_k \doteq u_k$, with x_i pairwise distinct, $x_i \notin var(u_j)$, then E is called an (equational problem in) solved form representing the solution $\sigma_E = [u_1/x_1, \ldots, u_k/x_k]$.

Proposition 2.28:
If E is a solved form then σ_E is am mgu of E.

Martelli/Montanari: Main Properties

If $E = x_1 \doteq u_1, \ldots, x_k \doteq u_k$, with x_i pairwise distinct, $x_i \notin var(u_j)$, then E is called an (equational problem in) solved form representing the solution $\sigma_E = [u_1/x_1, \ldots, u_k/x_k]$.

Theorem 2.29:

(1) If $E \Rightarrow_{MM} E'$ then σ is a unifier of E iff σ is a unifier of E'

(2) If $E \Rightarrow^*_{MM} \bot$ then E is not unifiable.

(3) If $E \Rightarrow^*_{MM} E'$ with E' in solved form, then $\sigma_{E'}$ is an mgu of E.

Proof:

(1) We have to show this for each of the rules. Let's treat the case for the 4th rule here. Suppose σ is a unifier of $x \doteq t$, that is, $x\sigma = t\sigma$. Thus, $\sigma \circ [t/x] = \sigma[x \mapsto t\sigma] = \sigma[x \mapsto x\sigma] = \sigma$. Therefore, for any equation $u \doteq v$ in E: $u\sigma = v\sigma$, iff $u[t/x]\sigma = v[t/x]\sigma$. (2) and (3) follow by induction from (1) using Proposition 2.28.

Main Unification Theorem

Theorem 2.30:
E is unifiable if and only if there is a most general unifier σ of E, such that σ is idempotent and $dom(\sigma) \cup codom(\sigma) \subseteq var(E)$.

Proof: See e.g. Baader & Nipkow: Term rewriting and all that.

Problem: *exponential growth* of terms possible

Example:

$E = \{x_1 \approx f(x_0, x_0), x_2 \approx f(x_1, x_1), \ldots, x_n \approx f(x_{n-1}, x_{n-1})\}$

m.g.u. $[x_1 \mapsto f(x_0, x_0), x_2 \mapsto f(f(x_0, x_0), f(x_0, x_0)), ...]$

$x_i \mapsto$ complete binary tree of heigth i

Solution: Use acyclic term graphs; union/find algorithms

Lifting Lemma

Lemma 2.31

Let C and D be variable-disjoint clauses. If

$$\dfrac{\begin{array}{c}C\\ \big\downarrow{\scriptstyle\sigma}\\ C\sigma\end{array}\qquad\qquad\begin{array}{c}D\\ \big\downarrow{\scriptstyle\rho}\\ D\rho\end{array}}{C'}\qquad \text{[propositional resolution]}$$

then there exists a substitution τ such that

$$\begin{array}{c}\dfrac{C\qquad\qquad D}{C''}\\ \big\downarrow{\scriptstyle\rho}\\ C' = C''\tau\end{array}\qquad\qquad \text{[general resolution]}$$

An analogous lifting lemma holds for factorization.

Saturation of Sets of General Clauses

Corollary 2.32:
Let N be a set of general clauses saturated under Res, i.e., $\mathit{Res}(N) \subseteq N$. Then also the set $G_\Sigma(N)$ of ground instances of N is saturated, that is,

$$\mathit{Res}(G_\Sigma(N)) \subseteq G_\Sigma(N).$$

Proof: W.l.o.g. we assume that clauses in N are pairwise variable-disjoint. (If not, make them disjoint; renaming changes neither $\mathit{Res}(N)$ nor $G_\Sigma(N)$.)

Let $C' \in \mathit{Res}(G_\Sigma(N))$, meaning (i) $C' \in G_\Sigma(N)$, (ii) there exist resolvable ground instances $C\sigma$ and $D\rho$ of N with resolvent C', or else (iii) C' is a factor of a ground instance $C\sigma$ of C.

Case (ii): By the Lifting Lemma, C and D are resolvable with a resolvent C'' with $C''\tau = C'$, for a suitable substitution τ. As $C'' \in N$ by assumption, we obtain that $C' \in G_\Sigma(N)$.

Case (iii): Similar.

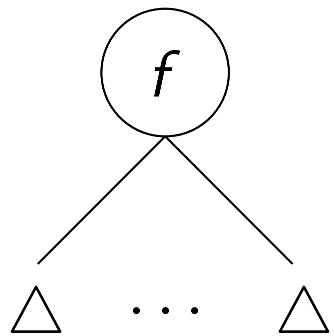
2.7 Herbrand Interpretations

From now on we shall consider PL without equality. Ω shall contain 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 $G_\Sigma(N) = \{C\sigma \text{ ground clause} \mid C \in N,\ \sigma : X \rightarrow \mathsf{T}_\Sigma\}$ 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:

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

Herbrand's Theorem

Lemma 2.33: Let N be a set of Σ-clauses, let $\mathcal{A}$ be an interpretation. Then $\mathcal{A} \models N$ implies $\mathcal{A} \models G_\Sigma(N)$.

Lemma 2.34: Let N be a set of Σ-clauses, let $\mathcal{A}$ be a *Herbrand* interpretation. Then $\mathcal{A} \models G_\Sigma(N)$ implies $\mathcal{A} \models N$.

Theorem 2.35 (Herbrand):
A set N of Σ-clauses is satisfiable iff it has a Herbrand model over Σ.

Proof:
The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let $N \not\models \bot$.

$$
\begin{aligned}
N \not\models \bot &\Rightarrow \bot \notin \mathit{Res}^*(N) && \text{(resolution is sound)}\\
&\Rightarrow \bot \notin G_\Sigma(\mathit{Res}^*(N))\\
&\Rightarrow I_{G_\Sigma(\mathit{Res}^*(N))} \models G_\Sigma(\mathit{Res}^*(N)) && \text{(Thm. 2.23; Cor. 2.32)}\\
&\Rightarrow I_{G_\Sigma(\mathit{Res}^*(N))} \models \mathit{Res}^*(N) && \text{(Lemma 2.34)}\\
&\Rightarrow I_{G_\Sigma(\mathit{Res}^*(N))} \models N && (N \subseteq \mathit{Res}^*(N))
\end{aligned}
$$

The Theorem of Löwenheim-Skolem

Theorem 2.36 (Löwenheim–Skolem):
Let Σ be a countable signature and let S be a set of closed Σ-formulas. Then S is satisfiable iff S has a model over a countable universe.

Proof:
If both X and Σ are countable, then S can be at most countably infinite. Now generate, maintaining satisfiability, a set N of clauses from S. This extends Σ by at most countably many new Skolem functions to Σ'. As Σ' is countable, so is $T_{\Sigma'}$, the universe of Herbrand-interpretations over Σ'. Now apply Theorem 2.35.

Refutational Completeness of General Resolution

Theorem 2.37:

Let N be a set of general clauses where $Res(N) \subseteq N$. Then

$$N \models \bot \Leftrightarrow \bot \in N.$$

Proof:

Let $Res(N) \subseteq N$. By Corollary 2.32: $Res(G_\Sigma(N)) \subseteq G_\Sigma(N)$

$$\begin{aligned} N \models \bot &\Leftrightarrow G_\Sigma(N) \models \bot && \text{(Lemma 2.33/2.34; Theorem 2.35)} \\ &\Leftrightarrow \bot \in G_\Sigma(N) && \text{(propositional resolution sound and complete)} \\ &\Leftrightarrow \bot \in N \end{aligned}$$

Compactness of Predicate Logic

Theorem 2.38 (Compactness Theorem for First-Order Logic):
Let Φ be a set of first-order formulas.
Φ is unsatisfiable $\Leftrightarrow$ some finite subset $\Psi \subseteq \Phi$ is unsatisfiable.

Proof:
The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let Φ be unsatisfiable and let N be the set of clauses obtained by Skolemization and CNF transformation of the formulas in Φ. Clearly $Res^*(N)$ is unsatisfiable. By Theorem 2.37, $\bot \in Res^*(N)$, and therefore $\bot \in Res^n(N)$ for some $n \in \mathbb{N}$. Consequently, $\bot$ has a finite resolution proof B of depth $\leq n$. Choose Ψ as the subset of formulas in Φ such that the corresponding clauses contain the assumptions (leaves) of B.