3.8 Hierarchic Superposition

The superposition calculus is a powerful tool to deal with formulas in *uninterpreted* first-order logic.

What can we do if some symbols have a *fixed interpretation*?

Can we combine superposition with decision procedures, e. g., for linear rational arithmetic? Can we integrate the decision procedure as a "black box"?

Sorted Logic

It is useful to treat this problem in sorted logic (cf. Sect. 1.11, page 31).

A many-sorted signature $\Sigma = (\Xi, \Omega, \Pi)$ fixes an alphabet of non-logical symbols, where

- Ξ is a set of sort symbols,
- Ω is a sets of function symbols,
- Π is a set of predicate symbols.

Each function symbol $f \in \Omega$ has a unique declaration $f : \xi_1 \times \cdots \times \xi_n \to \xi_0$; each predicate symbol $P \in \Pi$ has a unique declaration $P : \xi_1 \times \cdots \times \xi_n$ with $\xi_i \in \Xi$.

In addition, each variable x has a unique declaration $x : \xi$.

We assume that all terms, atoms, substitutions are well-sorted.

A many-sorted algebra $\mathcal{A}$ consists of

- a non-empty set $\xi_{\mathcal{A}}$ for each $\xi \in \Xi$,
- a function $f_{\mathcal{A}} : \xi_{1,\mathcal{A}} \times \cdots \times \xi_{n,\mathcal{A}} \to \xi_{0,\mathcal{A}}$ for each $f : \xi_1 \times \cdots \times \xi_n \to \xi_0 \in \Omega$,
- a subset $P_{\mathcal{A}} \subseteq \xi_{1,\mathcal{A}} \times \cdots \times \xi_{n,\mathcal{A}}$ for each $P : \xi_1 \times \cdots \times \xi_n \in \Pi$.

Hierarchic Specifications

A *specification* $SP = (\Sigma, \mathcal{C})$ consists of

- a signature $\Sigma = (\Xi, \Omega, \Pi)$,
- a class of term-generated Σ-algebras $\mathcal{C}$ closed under isomorphisms.

If $\mathcal{C}$ consists of *all* term-generated Σ-algebras satisfying the set of Σ-formulas N, we write $SP = (\Sigma, N)$.

A *hierarchic specification* $HSP = (SP, SP')$ consists of

- a base specification $SP = (\Sigma, \mathcal{C})$,
- an extension $SP' = (\Sigma', N')$,

where $\Sigma = (\Xi, \Omega, \Pi)$, $\Sigma' = (\Xi', \Omega', \Pi')$, $\Xi \subseteq \Xi'$, $\Omega \subseteq \Omega'$, and $\Pi \subseteq \Pi'$.

A Σ'-algebra $\mathcal{A}$ is called a model of $HSP = (SP, SP')$, if $\mathcal{A}$ is a model of N' and $\mathcal{A}|_\Sigma \in \mathcal{C}$, where the reduct $\mathcal{A}|_\Sigma$ is defined as $((\xi_\mathcal{A})_{\xi\in\Xi}, (f_\mathcal{A})_{f\in\Omega}, (P_\mathcal{A})_{P\in\Pi})$.

Note:

- no confusion: models of HSP may not identify elements that are different in the base models.
- no junk: models of HSP may not add new elements to the interpretations of base sorts.

Example:

Base specification: $((\Xi, \Omega, \Pi), \mathcal{C})$, where

$$\Xi = \{\mathit{int}\}$$

$$\begin{aligned}\Omega = \{\, & 0, 1, -1, 2, -2, \ldots : \to \mathit{int}, \\ & - : \mathit{int} \to \mathit{int}, \\ & + : \mathit{int} \times \mathit{int} \to \mathit{int}\,\}\end{aligned}$$

$$\begin{aligned}\Pi = \{\, & \geq : \mathit{int} \times \mathit{int}, \\ & > : \mathit{int} \times \mathit{int}\,\}\end{aligned}$$

$\mathcal{C}$ = isomorphy class of $\mathbb{Z}$

Extension: $((\Xi', \Omega', \Pi'), N')$, where

$$\Xi' = \Xi \cup \{\mathit{list}\}$$

$$\begin{aligned}\Omega' = \Omega \cup \{\, & \mathit{cons} : \mathit{int} \times \mathit{list} \to \mathit{list}, \\ & \mathit{length} : \mathit{list} \to \mathit{int}, \\ & \mathit{empty} : \to \mathit{list}, \\ & a : \to \mathit{list}\,\}\end{aligned}$$

$$\Pi' = \Pi$$

$$\begin{aligned}N' = \{\, & \mathit{length}(a) \geq 1, \\ & \mathit{length}(\mathit{cons}(x, y)) \approx \mathit{length}(y) + 1\,\}\end{aligned}$$

Goal:

Check whether N' has a model in which the sort *int* is interpreted by $\mathbb{Z}$ and the symbols from Ω and Π accordingly.

Hierarchic Superposition

In order to use a prover for the base theory, we must preprocess the clauses:

A term that consists only of base symbols and variables of base sort is called a base term (analogously for atoms, literals, clauses).

A clause C is called *weakly abstracted*, if every base term that occurs in C as a subterm of a non-base term (or non-base non-equational literal) is a variable.

Every clause can be transformed into an equivalent weakly abstracted clause. We assume that all input clauses are weakly abstracted.

A substitution is called simple, if it maps every variable of a base sort to a base term.

The inference rules of the hierarchic superposition calculus correspond to the rules of of the standard superposition calculus with the following modifications:

- The term ordering $\succ$ must have the property that every base ground term (or non-equational literal) is smaller than every non-base ground term (or non-equational literal).
- We consider only simple substitutions as unifiers.
- We perform only inferences on non-base terms (or non-base non-equational literals).
- If the conclusion of an inference is not weakly abstracted, we transform it into an equivalent weakly abstracted clause.

While clauses that contain non-base literals are manipulated using superposition rules, base clauses have to be passed to the base prover.

This yields one more inference rule:

Constraint Refutation: $\dfrac{M}{\bot}$

where M is a set of base clauses
that is inconsistent w. r. t. $\mathcal{C}$.

Problems

There are two potential problems that are harmful to refutational completeness:

- We can only apply the constraint refutation rule to finite sets M. If $\mathcal{C}$ is not compact, this is not sufficient.
- Since we only consider simple substitutions, we will only obtain a model of all *simple ground instances*.

  To show that we have a model of *all* instances, we need an additional condition called *sufficient completeness w. r. t. simple instances*.

A set N of clauses is called *sufficiently complete with respect to simple instances*, if for every model $\mathcal{A}'$ of the set of simple ground instances of N and every ground non-base term t of a base sort there exists a ground base term t such that $t' \approx t$ is true in $\mathcal{A}'$.

Note: Sufficient completeness w. r. t. simple instances ensures the absence of junk.

If the base signature contains Skolem constants, we can sometimes enforce sufficient completeness by equating ground extension terms with a base sort to Skolem constants.

Skolem constants may harmful to compactness, though.

Completeness of Hierarchic Superposition

If the base theory is compact, the hierarchic superposition calculus is refutationally complete for sets of clauses that are sufficiently complete with respect to simple instances (Bachmair, Ganzinger, Waldmann, 1994; Baumgartner, Waldmann 2013).

Main proof idea:

If the set of base clauses in N has some base model, represent this model by a set E of convergent ground equations and a set D of ground disequations.

Then show: If N is saturated w. r. t. hierarchic superposition, then $E \cup D \cup \tilde{N}$ is saturated w. r. t. standard superposition, where $\tilde{N}$ is the set of simple ground instances of clauses in N that are reduced w. r. t. E.

A Refinement

In practice, a base signature often contains *domain elements*, that is, constant symbols that are

- guaranteed to be different from each other in every base model, and
- minimal w. r. t. $\succ$ in their equivalent class.

Typical example for domain elements: number constants $0, 1, -1, 2, -2, \ldots$

If the base signature contains *domain elements*, then weak abstraction can be redefined as follows:

A clause C is called *weakly abstracted*, if every base term that occurs in C as a subterm of a non-base term (or non-base non-equational literal) is a variable *or a domain element*.

Why does that work?

Literature

Leo Bachmair, Harald Ganzinger. Uwe Waldmann: Refutational Theorem Proving for Hierarchic First-Order Theories. Applicable Algebra in Engineering, Communication and Computing, 5(3/4):193–212, 1994.

Peter Baumgartner, Uwe Waldmann: Hierarchic Superposition With Weak Abstraction. Automated Deduction, CADE-24, LNAI 7898, pp. 39–57, Springer, 2013.

3.9 Integrating Theories I: E-Unification

Dealing with mathematical theories naively in a superposition prover is difficult:

Some axioms (e. g., commutativity) cannot be oriented w. r. t. a reduction ordering.
⇒ Provers compute many equivalent copies of a formula.

Some axiom sets (e. g., torsion-freeness, divisibility) are infinite.
⇒ Can we tell which axioms are really needed?

Hierarchic ("black-box") superposition is easy to implement, but conditions like compactness and sufficient completeness are rather restrictive.

Can we integrate theories directly into theorem proving calculi ("white-box" integration)?

Idea:

In order to avoid enumerating entire congruence classes w. r. t. an equational theory E, treat formulas as *representatives* of their congruence classes.

Compute an inference between formula C and D if an inference between some clause represented by C and some clause represented by D would be possible.

Consequence: We have to check whether there are substitutions that make terms s and t equal w. r. t. E.
⇒ Unification is replaced by E-unification.

E-Unification

E-unification (unification modulo an equational theory E):

For a set of equality problems $\{s_1 \approx t_1, \dots, s_n \approx t_n\}$, an E-unifier is a substitution σ such that for all $i \in \{1, \dots, n\}$: $s_i\sigma \approx_E t_i\sigma$.

Recall: $s_i\sigma \approx_E t_i\sigma$ means $E \models s_i\sigma \approx t_i\sigma$.

In general, there are infinitely many (E-)unifiers.
What about most general unifiers?

Frequent cases: $E = \emptyset$, $E = \mathrm{AC}$, $E = \mathrm{ACU}$:

$$
\begin{aligned}
x + (y + z) &\approx (x + y) + z && \text{(associativity = A)}\\
x + y &\approx y + x && \text{(commutativity = C)}\\
x + 0 &\approx x && \text{(identity (unit) = U)}
\end{aligned}
$$

The identity axiom is also abbreviated by "1", in particular, if the binary operation is denoted by $*$. (ACU = AC1).

Example:

$x + y$ and c are ACU-unifiable with $\{x \mapsto c,\ y \mapsto 0\}$ and $\{x \mapsto 0,\ y \mapsto c\}$.

$x + y$ and $x' + y'$ are ACU-unifiable with $\{x \mapsto z_1 + z_2,\ y \mapsto z_3 + z_4,\ x' \mapsto z_1 + z_3,\ y' \mapsto z_2 + z_4\}$ (among others).

More general substitutions:

Let X be a set of variables.
A substitution σ is more general modulo E than a substitution σ' on X, if there exists a substitution ρ such that $x\sigma\rho \approx_E x\sigma'$ for all $x \in X$.

Notation: $\sigma \lesssim_E^X \sigma'$.

(Why X? Because we cannot restrict to idempotent substitutions.)

Complete sets of unifiers:

Let S be an E-unification problem, let $X = Var(S)$.
A set C of E-unifiers of S is called complete (CSU),
if for every E-unifier σ' of S there exists a $\sigma \in C$
with $\sigma \lesssim_E^X \sigma'$.

A complete set of E-unifiers C is called minimal (μCSU),
if for all $\sigma, \sigma' \in C$, $\sigma \lesssim_E^X \sigma'$ implies $\sigma = \sigma'$.

Note: every E-unification problem has a CSU. (Why?)

The set of equations E is of unification type

unitary, if every E-unification problem has a μCSU with cardinality ≤ 1 (e. g.: $E = \emptyset$);

finitary, if every E-unification problem has a finite μCSU (e. g.: $E =$ ACU, $E =$ AC, $E =$ C);

infinitary, if every E-unification problem has a μCSU and some E-unification problem has an infinite μCSU (e. g.: $E =$ A);

zero (or nullary), if some E-unification problem does not have a μCSU (e. g.: $E = \mathrm{A} \cup \{x + x \approx x\}$).

Unification modulo ACU

Let us first consider elementary ACU-unification:
the terms to be unified contain only variables and the function symbols from $\Sigma = (\{+/2,\ 0/0\}, \emptyset)$.

Since parentheses and the order of summands don't matter, every term over $X_n = \{x_1, \dots, x_n\}$ can be written as a sum $\sum_{i=1}^{n} a_i\, x_i$.

The ACU-equivalence class of a term $t = \sum_{i=1}^{n} a_i\, x_i \in T_\Sigma(X_n)$ is uniquely determined by the vector $\vec{v}_n(t) = (a_1, \dots, a_n)$.

Analogously, a substitution $\sigma = \{\, x_i \to \sum_{j=1}^{m} b_{ij}\, x_j \mid 1 \le i \le n \,\}$ is uniquely determined by the matrix

$$M_{n,m}(\sigma) = \begin{pmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nm} \end{pmatrix}$$

Let $t = \sum_{i=1}^{n} a_i\, x_i$ and $\sigma = \{\, x_i \to \sum_{j=1}^{m} b_{ij}\, x_j \mid 1 \le i \le n \,\}$.

Then $t\sigma = \sum_{i=1}^{n} a_i\, (\sum_{j=1}^{m} b_{ij}\, x_j)$
$\quad = \sum_{i=1}^{n} \sum_{j=1}^{m} a_i\, b_{ij}\, x_j$
$\quad = \sum_{j=1}^{m} \sum_{i=1}^{n} a_i\, b_{ij}\, x_j$
$\quad = \sum_{j=1}^{m} (\sum_{i=1}^{n} a_i\, b_{ij})\, x_j.$

Consequence:

$\vec{v}_m(t\sigma) = \vec{v}_n(t) \cdot M_{n,m}(\sigma).$

Let $S = \{s_1 \approx t_1, \dots, s_k \approx t_k\}$ be a set of equality problems over $T_\Sigma(X_n)$.

Then the following properties are equivalent:

(a) σ is an ACU-unifier of S from $X_n \to T_\Sigma(X_m)$.

(b) $\vec{v}_m(s_i\sigma) = \vec{v}_m(t_i\sigma)$ for all $i \in \{1, \dots, k\}$.

(c) $\vec{v}_n(s_i) \cdot M_{n,m}(\sigma) = \vec{v}_n(t_i) \cdot M_{n,m}(\sigma)$ for all $i \in \{1, \dots, k\}$.

(d) $(\vec{v}_n(s_i) - \vec{v}_n(t_i)) \cdot M_{n,m}(\sigma) = \vec{0}_m$ for all $i \in \{1, \dots, k\}$.

(e) $M_{k,n}(S) \cdot M_{n,m}(\sigma) = \vec{0}_{k,m}$.
where $M_{k,n}(S)$ is the $k \times n$ matrix whose rows are the vectors $\vec{v}_n(s_i) - \vec{v}_n(t_i)$.

(f) The columns of $M_{n,m}(\sigma)$ are non-negative integer solutions of the system of homogeneous linear diophantine equations $DE(S)$:
$$M_{k,n}(S) \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$