Hierarchic Superposition: Completeness without Compactness

Peter Baumgartner
NICTA and ANU, Canberra

Uwe Waldmann
MPI für Informatik, Saarbrücken

Hierarchic Reasoning

Question:

We have a decision procedure for some kind of arithmetic.

How can we use it to solve problems that involve more than arithmetic?

Hierarchic Reasoning

The decision procedure implements a background (BG) specification:

sorts, e.g., $\{\mathit{int}\}$

operators, e.g., $\{0, 1, -1, 2, -2, \ldots, -, +, >, \geq, \alpha, \beta, \ldots\}$

models, e.g., linear integer arithmetic (LIA), where the *parameters* $\alpha, \beta, \ldots$ can be interpreted by arbitrary elements of the universe.

Example:

$$\forall x (x \leq 0 \vee x \geq \alpha) \wedge \alpha > 0 \quad \rightarrow \quad \text{sat (choose } \alpha = 1)$$

$$\forall x (x < 0 \vee x > \alpha) \wedge \alpha > 0 \quad \rightarrow \quad \text{unsat}$$

Hierarchic Reasoning

A foreground (FG) specification extends the BG specification by

new sorts, e.g., $\{\mathit{list}\}$

new operators, e.g., $\{\mathit{cons} : \mathit{int} \times \mathit{list} \to \mathit{list},$
$\mathit{length} : \mathit{list} \to \mathit{int},$
$\mathit{empty} : \mathit{list},$
$a : \mathit{list}\}$

first-order clauses, e.g., $\{\mathit{length}(a) \geq 1,$
$\mathit{length}(\mathit{cons}(x, y)) \approx \mathit{length}(y) + 1\}.$

Hierarchic Reasoning

Goal:

Check whether the FG specification has models or not, using the BG decision procedure as a subroutine.

Note: We are only interested in models that leave the interpretation of BG sorts and operators unchanged, i. e., in *conservative extensions*.

Hierarchic Reasoning

Calculi for hierarchic reasoning:

If the FG clauses are ground:

DPLL(T) + Nelson–Oppen
$\Rightarrow$ decision procedure for the hierarchic combination.

Otherwise:

Hierarchic superposition
$\Rightarrow$ refutationally complete *under certain conditions.*

Hierarchic Superposition

Hierarchic superposition calculus:

Saturation-based calculus
(like resolution or standard superposition).

Input: a finite set N of FG clauses.

Output: a possibly infinite set N_0 of BG clauses
(to be passed to the BG prover).

If N_0 is unsatisfiable w. r. t. the BG specification,
then N is unsatisfiable w. r. t. the hierarchic specification.
(Reverse direction needs additional conditions.)

Condition 1

Fundamental problem 1:

The BG prover can detect an inconsistency only if it is expressed in the language of the BG prover.

⇒ Condition 1: *Sufficient completeness*

In every model of the FG clauses, every ground FG term that has a BG sort must be equivalent to some BG term.

– Very restrictive in practice.

– Undecidable.

– But can be established automatically by introducing new parameters if all BG-sorted FG terms are ground.

Condition 2

Fundamental problem 2:

We can only pass *finite* sets of BG clauses to the BG prover.

$\Rightarrow$ Condition 2: *Compactness*

Every unsatisfiable set of BG clauses must have a finite unsatisfiable subset.

– Holds for the first-order theory of LIA.

– Does not hold for the standard model $\mathbb{Z}$ of LIA (in the presence of parameters).

Condition 2

Example:

Input: $\{\, p(0),$

$\neg p(x) \;\vee\; x < \alpha,$

$\neg p(x) \;\vee\; x+1 < y \;\vee\; p(y) \,\}$

Output: $\{\, 0 < \alpha,$

$0+1 < y_1 \;\vee\; y_1 < \alpha,$

$0+1 < y_1 \;\vee\; y_1+1 < y_2 \;\vee\; y_2 < \alpha,$

$0+1 < y_1 \;\vee\; y_1+1 < y_2 \;\vee\; y_2+1 < y_3 \;\vee\; y_3 < \alpha,$

$\ldots \quad \}$

Condition 2

Example:

Input: $\{\, p(0),$

$\neg p(x) \;\vee\; x < \alpha,$

$\neg p(x) \;\vee\; x+1 < y \;\vee\; p(y)\,\}$

Output: $\{\, 0 < \alpha,$

$1 < \alpha,$

$2 < \alpha,$

$3 < \alpha,$

$\ldots \quad \}$

Completeness without Compactness

Question:

Are there classes of FG-clause sets for which we can guarantee that the first-order theory of LIA and the standard model of LIA behave in the same way?

(This would imply refutational completeness even w. r. t. the standard model of LIA.)

Completeness without Compactness

Answer:

Yes, it works, provided that every BG-sorted term is either

- a variable,
- or ground,
- or a sum $x + k$ of a variable x and a number $k \geq 0$ that occurs on the right-hand side of a positive literal $s < x + k$.

Note: The counterexample above had $x + 1$ on the *left-hand side* of the literal $x + 1 < y$.

Proof

Key observation:

After the initial introduction of parameters to ensure sufficient completeness, hierarchic superposition does not introduce any new BG-sorted ground terms.

Consequence:

The possibly infinite set of BG-clauses that is generated is built over a *finite* set of ground terms T (and an infinite set X of variables).

We can show that is it equivalent to some *finite* set of BG-clauses.

Proof

Step 1:

Let N_0 be a set of BG clauses with the restrictions above;
let T be the finite set of ground terms occurring in N_0.

Eliminate $>$ and $\geq$;
replace $\neg\, s < t$ by $t \leq s$ and $\neg\, s \leq t$ by $t < s$.

Result: All literals have the form $s \approx t$, $s \not\approx t$, $s < t$, $s \leq t$,
or $s < x + k$, where $s, t \in X \cup T$ and $k \in \mathbb{N}$.

Proof

Step 2:

Introduce new relation symbols $<_k$ defined by
$a <_k b \;\Leftrightarrow\; a < b + k$.

Replace $s < t$ by $s <_0 t$,

$s \leq t$ by $s <_1 t$,

$s < x + k$ by $s <_k x$.

Observe that $s <_k t$ entails $s <_n t$ whenever $k \leq n$.

Proof

Step 3:

Eliminate variables:

$$N \cup \{\, C \vee x \not\approx x \,\} \quad \rightarrow \quad N \cup \{\, C \,\}$$

$$N \cup \{\, C \vee x \not\approx t \,\} \quad \rightarrow \quad N \cup \{\, C[x \mapsto t] \,\}$$

$$N \cup \{\, C \vee x \approx x \,\} \quad \rightarrow \quad N$$

$$N \cup \{\, C \vee x \approx t \,\} \quad \rightarrow \quad N \cup \{\, C \vee x <_1 t,\ C \vee t <_1 x \,\}$$

$$N \cup \{\, C \vee \bigvee_{i \in I} x <_{k_i} s_i \vee \bigvee_{j \in J} t_j <_{n_j} x \,\}$$

$$\rightarrow \quad N \cup \{\, C \vee \bigvee_{i \in I} \bigvee_{j \in J} t_j <_{k_i + n_j} s_i \,\}$$

Proof

Step 4:

Ensure that any pair of terms s, t from T is related by at most one literal in any clause, e. g.:

$$N \cup \{\, C \vee s <_k t \vee s \approx t \,\} \;\rightarrow\; N \cup \{\, C \vee s <_k t \,\} \quad \text{if } k \geq 1$$

$$N \cup \{\, C \vee s <_0 t \vee s \approx t \,\} \;\rightarrow\; N \cup \{\, C \vee s <_1 t \,\}$$

$$N \cup \{\, C \vee s <_k t \vee s <_n t \,\} \;\rightarrow\; N \cup \{\, C \vee s <_n t \,\} \quad \text{if } k \leq n$$

$$N \cup \{\, C \vee s <_k t \vee t <_n s \,\} \;\rightarrow\; N \quad \text{if } k + n \geq 1$$

$$N \cup \{\, C \vee s <_0 t \vee t <_0 s \,\} \;\rightarrow\; N \cup \{\, C \vee s \not\approx t \,\}$$

...

Proof

Result:

All literals are ground.

Any pair of terms $s, t \in T$ is related by at most one literal per clause.

$\Rightarrow$ At most $\frac{1}{2}m(m+1)$ literals per clause, where $m = |T|$.

But the indices of $<_k$ are unbounded, so the number of clauses can still be infinite.

Proof

Step 5:

Introduce an equivalence relation $\sim$ on clauses:

$C \sim C'$, if for all $s, t \in T$

- $s \approx t \in C$ iff $s \approx t \in C'$,
- $s \not\approx t \in C$ iff $s \not\approx t \in C'$,
- $s <_k t \in C$ for some k iff $s <_n t \in C'$ for some n.

$\Rightarrow$ Finitely many equivalence classes.

Proof

Step 6:

Clauses C, C' in one equivalence class differ at most in the indices of the ordering literals.

C entails C' if the tuple of indices in C is pointwise smaller than the tuple of indices in C'.

Dickson's lemma: For every set of tuples in $\mathbb{N}^n$ the subset of all minimal tuples is finite.

The clauses that correspond to these minimal tuples entail all other clauses.

So N_0 is equivalent to a finite set of clauses. $\square$

Linear Rational Arithmetic

An analogous result for linear rational arithmetic can be proved in essentially the same way.

Thanks for your attention.