Hierarchic Superposition: Completeness without Compactness

Peter Baumgartner[1] and Uwe Waldmann[2]

[1] NICTA* and Australian National University, Canberra, Australia
Peter.Baumgartner@nicta.com.au
[2] MPI für Informatik, Saarbrücken, Germany
uwe@mpi-inf.mpg.de

Abstract. Many applications of automated deduction and verification require reasoning in combinations of theories, such as, on the one hand (some fragment of) first-order logic, and on, the other hand a background theory, such as some form of arithmetic. Unfortunately, due to the high expressivity of the full logic, complete reasoning is impossible in general. It is a realistic goal, however, to devise theorem provers that are "reasonably complete" in practice, and the hierarchic superposition calculus has been designed as a theoretical basis for that. In a recent paper we introduced an extension of hierarchic superposition and proved its completeness for the fragment where every term of the background sort is ground. In this paper, we extend this result and obtain completeness for a larger fragment that admits variables in certain places.

1 Hierarchic Superposition

Many applications of automated deduction and verification require reasoning in *combinations of theories*, such as, on the one hand (some fragment of) first-order logic and on the other hand some form of arithmetic. In hierarchic superposition [2, 3] we consider the following scenario:

We assume that we have a background ("BG") prover that accepts as input a set of clauses over a *BG signature* $\Sigma_{\mathrm{B}} = (\Xi_{\mathrm{B}}, \Omega_{\mathrm{B}})$, where Ξ_{B} is a set of *BG sorts* and Ω_{B} is a set of *BG operators*. Terms/clauses over Σ_{B} and BG-sorted variables are called *BG terms/clauses*. For instance, Ξ_{B} might be $\{\mathit{int}, \mathit{bool}_{\mathrm{B}}\}$ and Ω_{B} might contain the integer numbers, $+, -, <, \leq, \mathit{true}_{<}, \mathit{true}_{\leq}$, and additional parameters $\alpha, \beta, \ldots$ that may be interpreted freely over the *int*-domain. The BG prover decides the satisfiability of Σ_{B}-clause sets w. r. t. a *BG specification*, say linear integer arithmetic (LIA).

For technical reasons, we assume that equality is the only predicate symbol in our language and that any non-equational atom $p(t_1, \ldots, t_n)$ is encoded as an equation $p(t_1, \ldots, t_n) \approx \mathit{true}_p$. We refer to the terms that result from this encoding of atoms as *atom terms*; all other terms are called *proper terms*. When we simply write, say, $x \leq y$, this should always be taken as a shorthand for an equation as above.

* NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

The foreground ("FG") theorem prover accepts as inputs clauses over a signature $\Sigma = (\Xi, \Omega)$, where $\Xi_B \subseteq \Xi$ and $\Omega_B \subseteq \Omega$. The sorts in $\Xi_F = \Xi \setminus \Xi_B$ and the operator symbols in $\Omega_F = \Omega \setminus \Omega_B$ are called *FG sorts* and *FG operators*. For instance, Ξ_F might be $\{list, bool_F\}$ and Ω_F might contain operators $cons : int \times list \to list$, $length : list \to int$, $isempty : list \to bool_F$, and $true_{isempty} : \to bool_F$, among others. Σ-terms that are not BG terms are called FG terms. Notice that FG terms such as $length(x)$ can have BG sorts.

After abstracting out certain BG terms that occur as subterms of FG terms,[3] the FG prover saturates the set of Σ-clauses using the inference rules of hierarchic superposition, such as, e. g.,

$$\text{Negative superposition}\ \frac{l \approx r \vee C \qquad s[u] \not\approx t \vee D}{\text{abstr}((s[r] \not\approx t \vee C \vee D)\sigma)}$$

if (i) neither l nor u is a BG term, (ii) u is not a variable, (iii) σ is a simple mgu of l and u, (iv) $r\sigma \not\succeq l\sigma$, (v) $(l \approx r)\sigma$ is strictly maximal in $(l \approx r \vee C)\sigma$, (vi) the first premise does not have selected literals, (vii) $t\sigma \not\succeq s\sigma$, and (viii) if the second premise has selected literals, then $s \not\approx t$ is selected in the second premise, otherwise $(s \not\approx t)\sigma$ is maximal in $(s \not\approx t \vee D)\sigma$.

These differ from the standard superposition inference rules [1] mainly in that only the FG parts of clauses are overlapped and that any BG clauses derived during the saturation are instead passed to the BG prover. The BG prover implements an inference rule

$$\text{Close}\ \frac{C_1 \quad \cdots \quad C_n}{\square}$$

if $C_1, \ldots, C_n$ are BG clauses and $\{C_1, \ldots, C_n\}$ is unsatisfiable w. r. t. the BG specification.

As soon as one of the two provers detects a contradiction, the input clause set has been shown to be inconsistent w. r. t. *conservative extensions of the BG specification*.

2 Refutational Completeness

There are two requirements for the refutational completeness of hierarchic superposition. The first one is a variant of *sufficient completeness*: We must be able to prove that every ground BG-sorted FG term is equal to some BG term. Sufficient completeness of a set of Σ-clauses is a property that is not even recursively enumerable. For certain classes of Σ-clause sets, however, it is possible to establish sufficient completeness automatically [5, 3]: If all BG-sorted FG terms are ground, it suffices to add a *definition* $\alpha_t \approx t$ for every BG-sorted FG term t occurring in a clause $C[t]$, where α_t is a new parameter (BG constant); afterwards $C[t]$ can be replaced by $C[\alpha_t]$.

[3] *Abstracting out* a term t that occurs in a clause $C[t]$ means replacing $C[t]$ by $x \not\approx t \vee C[x]$ for a new variable x. The reverse operation is called *unabstraction*.

Since we can only pass *finite* clause sets to a BG prover, there is a second requirement for refutational completeness, namely the *compactness* of the BG specification. A specification is called compact, if every set of formulas that is unsatisfiable w. r. t. the specification has a finite unsatisfiable subset.

It is well-known that first-order logic is compact. So, if we assume that the BG prover checks satisfiability w. r. t., say, the first-order theory of LIA[4] the compactness requirement is automatically satisfied. Unfortunately, as soon as the BG signature contains parameters, satisfiability w. r. t. the first-order theory of LIA differs from satisfiability w. r. t. LIA over $\mathbb{Z}$. Consider the following example: Suppose that the BG signature contains the parameter α in addition to the integer numbers and the operator symbols of LIA, and that we have a unary FG predicate symbol p and the Σ-clauses $p(0)$, $\neg p(x) \vee x < \alpha$, and $\neg p(x) \vee x + 1 < y \vee p(y)$. Starting with these clauses, hierarchic superposition produces a set N_1 of BG clauses

$$
\begin{aligned}
&0 < \alpha,\\
&0 + 1 < y_1 \vee y_1 < \alpha,\\
&0 + 1 < y_1 \vee y_1 + 1 < y_2 \vee y_2 < \alpha,\\
&0 + 1 < y_1 \vee y_1 + 1 < y_2 \vee y_2 + 1 < y_3 \vee y_3 < \alpha,\\
&\ldots
\end{aligned}
$$

which, after removing the universally quantified variables by quantifier elimination, turns out to be equivalent to $\{0 < \alpha,\ 1 < \alpha,\ 2 < \alpha,\ 3 < \alpha,\ \ldots\}$. Each finite subset of N_1 is satisfiable in $\mathbb{Z}$, and hence in the first-order theory of LIA. By compactness of first-order logic, N_1 itself is also satisfiable in the first-order theory of LIA, for instance in the non-standard model $\mathbb{Q} \times \mathbb{Z}$ with $0 := (0,0)$, $1 := (0,1)$, $\alpha := (1,0)$, $(x,y) + (x',y') := (x + x', y + y')$, and a lexicographic ordering. On the other hand, the set N_1 is clearly unsatisfiable in $\mathbb{Z}$. This leaves us two undesirable choices: If we assume that the BG specification is given by LIA over $\mathbb{Z}$, hierarchic superposition is not refutationally complete – there is a contradiction, but we will never detect it. If we assume that the BG specification is the first-order theory of LIA, hierarchic superposition is refutationally complete, but we get non-standard models, that we would prefer to exclude in most applications.

3 Completeness without Compactness

Are there classes of Σ-clause sets for which we can guarantee that hierarchic superposition is refutationally complete even if we restrict ourselves to the standard models of linear integer or rational arithmetic? A first answer in this direction was given in [3]: If all BG-sorted terms in a clause set are ground, clauses are appropriately preprocessed, and some reasonable restrictions on simplifications are observed, then the hierarchic superposition calculus can produce only finitely many different BG clauses (up to unabstraction and duplication of literals). Refutational completeness follows immediately.

In the current paper, we extend this result significantly by permitting also BG-sorted variables and, in certain positions, even variables with offsets.

[4] That is, the set of all first-order BG sentences that hold in LIA.

Theorem 1. *Let N be a set of clauses over the signature of linear integer arithmetic (with parameters α, β, etc.), such that every proper term in these clauses is either (i) ground, or (ii) a variable, or (iii) 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 $s < x + k$. If the set of ground terms occurring in N is finite, then N is satisfiable in LIA over $\mathbb{Z}$ if and only if N is satisfiable w. r. t. the first-order theory of LIA.*

Proof. Let N be a set of clauses with the required properties, and let T be the finite set of ground terms occurring in N. We will show that N is equivalent to some *finite* set of clauses over the signature of linear integer arithmetic, which implies that it is satisfiable in the integer numbers if and only if it is satisfiable in the first-order theory of LIA.

In a first step, we replace every negative ordering literal $\neg s < t$ or $\neg s \leq t$ by the equivalent positive ordering literal $t \leq s$ or $t < s$. All literals of clauses in the resulting set N_0 have the form $s \approx t$, $s \not\approx t$, $s < t$, $s \leq t$, or $s < x + k$, where s and t are either variables or elements of T and $k \in \mathbb{N}$. Note that the number of variables in clauses in N_0 may be unbounded.

In order to handle the various inequality literals in a more uniform way, we introduce new binary relation symbols $<_k$ (for $k \in \mathbb{N}$) that are defined by $a <_k b$ if and only if $a < b + k$. Observe that $s <_k t$ entails $s <_n t$ whenever $k \leq n$. Obviously, we may replace every literal $s < t$ by $s <_0 t$, every literal $s \leq t$ by $s <_1 t$, and every literal $s < x + k$ by $s <_k x$. Let N_1 be the resulting clause set.

We will now transform N_1 into an equivalent set N_2 of ground clauses. We start by eliminating all equality literals that contain variables by exhaustively applying the following transformation rules:

$$
\begin{array}{lcll}
N \cup \{C \vee x \not\approx x\} & \to & N \cup \{C\} & \\
N \cup \{C \vee x \not\approx t\} & \to & N \cup \{C[x \mapsto t]\} & \text{if } t \neq x \\
N \cup \{C \vee x \approx x\} & \to & N & \\
N \cup \{C \vee x \approx t\} & \to & N \cup \{C \vee x <_1 t,\ C \vee t <_1 x\} & \text{if } t \neq x
\end{array}
$$

All variables in inequality literals are then eliminated in a Fourier-Motzkin-like manner by exhaustively applying the transformation rule

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

where x does not occur in C and one of the index sets I and J may be empty.

The clauses in N_2 are constructed over the finite set T of proper ground terms, but the length of the clauses in N_2 is potentially unbounded. In the next step, we will transform the clauses in such a way that any pair of terms s, t from T is related by at most one literal in any clause: We apply one of the following transformation rules as long as two terms s and t occur in more than one literal:

$$
\begin{array}{lcll}
N \cup \{C \vee s <_k t \vee s \approx t\} & \to & N \cup \{C \vee s <_k t\} & \text{if } k \geq 1 \\
N \cup \{C \vee s <_0 t \vee s \approx t\} & \to & N \cup \{C \vee s <_1 t\} & \\
N \cup \{C \vee s <_k t \vee s \not\approx t\} & \to & N & \text{if } k \geq 1 \\
N \cup \{C \vee s <_0 t \vee s \not\approx t\} & \to & N \cup \{C \vee s \not\approx t\} &
\end{array}
$$

$$
\begin{array}{lcll}
N \cup \{C \vee s <_k t \vee s <_n t\} & \to & N \cup \{C \vee s <_n t\} & \text{if } k \leq n \\
N \cup \{C \vee s <_k t \vee t <_n s\} & \to & N & \text{if } k + n \geq 1 \\
N \cup \{C \vee s <_0 t \vee t <_0 s\} & \to & N \cup \{C \vee s \not\approx t\} & \\
N \cup \{C \vee L \vee L\} & \to & N \cup \{C \vee L\} & \text{for any literal } L \\
N \cup \{C \vee s \approx t \vee s \not\approx t\} & \to & N &
\end{array}
$$

The length of the clauses in the resulting set N_3 is now bounded by $\frac{1}{2}m(m+1)$, where m is the cardinality of T. Still, due to the indices of the relation symbols $<_k$, N_3 may be infinite. We introduce an equivalence relation $\sim$ on clauses in N_3 as follows: Define $C \sim C'$ if for all $s, t \in T$ (i) $s \approx t \in C$ if and only if $s \approx t \in C'$, (ii) $s \not\approx t \in C$ if and only if $s \not\approx t \in C'$, and (iii) $s <_k t \in C$ for some k if and only if $s <_n t \in C'$ for some n. This relation splits N_3 into at most $(\frac{1}{2}m(m+1))^5$ equivalence classes.[5]

We will now show that each equivalence class is logically equivalent to a finite subset of itself. Let M be some equivalence class. Since any two clauses from M differ at most in the indices of their $<_k$-literals, we can write every clause $C_i \in M$ in the form

$$C_i \;=\; C \vee \bigvee_{1 \leq l \leq n} s_l <_{k_{il}} t_l$$

where C and the s_l and t_l are the same for all clauses in M. As we have mentioned above, $s_l <_{k_{il}} t_l$ entails $s_l <_{k_{jl}} t_l$ whenever $k_{il} \leq k_{jl}$; so a clause $C_i \in M$ entails $C_j \in M$ whenever the n-tuple $(k_{i1}, \ldots, k_{in})$ is pointwise smaller or equal to the n-tuple $(k_{j1}, \ldots, k_{jn})$ (that is, $k_{il} \leq k_{jl}$ for all $1 \leq l \leq n$).

Let Q be the set of n-tuples of natural numbers corresponding to the clauses in M. By Dickson's lemma [4], for every set of tuples in $\mathbb{N}^n$ the subset of minimal tuples (w. r. t. the pointwise extension of $\leq$ to tuples) is finite. Let Q' be the subset of minimal tuples in Q, and let M' be the set of clauses in M that correspond to the tuples in Q'. Since for every tuple in $Q \setminus Q'$ there is a smaller tuple in Q', we know that every clause in $M \setminus M'$ is entailed by some clause in M'. So the equivalence class M is logically equivalent to its finite subset M'. Since the number of equivalence classes is also finite and all transformation rules are sound, this proves our claim. □

Corollary 2. *The hierarchic superposition calculus is refutationally complete w. r. t. LIA over $\mathbb{Z}$ for finite sets of Σ-clauses in which every proper BG-sorted term is either (i) ground, or (ii) a variable, or (iii) 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 $s < x + k$.*[6]

Proof. Let N be a finite set of Σ-clauses with the required properties. By introducing definitions $\alpha_t \approx t$ as described above and weak abstraction we obtain a sufficiently complete finite set N_0 of abstracted clauses.

Now we run the hierarchic superposition calculus on N_0 (with the same restrictions on simplifications as in [3]). Let N_1 be the (possibly infinite) set of BG clauses generated during the run. By unabstracting these clauses, we obtain an equivalent set N_2 of

[5] Any pair of terms s, t is related in all clauses of an equivalence class by either a literal $s \approx t$, or $s \not\approx t$, or $s <_n t$ for some n, or $t <_n s$ for some n, or no literal at all, so there are five possibilities per unordered pair of terms.

[6] Note that in the counterexample above $x+1$ occurs on the *left-hand* side of the literal $x+1 < y$.

clauses that satisfy the conditions of Thm. 1, so N_2 is satisfiable in LIA over $\mathbb{Z}$ if and only if N is satisfiable w. r. t. the first-order theory of LIA. Since the hierarchic superposition calculus is refutationally complete w. r. t. the first-order theory of LIA, the result follows. □

Analogous results hold for linear rational arithmetic. Let n be the least common divisor of all numerical constants in the original clause set; then we define $a <_{2i} b$ by $a < b + \frac{i}{n}$ and $a <_{2i+1} b$ by $a \leq b + \frac{i}{n}$ for $i \in \mathbb{N}$ and express every inequation literal in terms of $<_k$. The Fourier-Motzkin transformation rule is replaced by

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

where x does not occur in C, one of the index sets I and J may be empty, and $k \bullet n$ is defined as $k + n - 1$ if both k and n are odd, and $k + n$ otherwise. The rest of the proof proceeds in the same way as before.

References

1. L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.
2. L. Bachmair, H. Ganzinger, and U. Waldmann. Refutational theorem proving for hierarchic first-order theories. *Appl. Algebra Eng. Commun. Comput*, 5:193–212, 1994.
3. P. Baumgartner and U. Waldmann. Hierarchic superposition with weak abstraction. In M. P. Bonacina, ed., *24nd Int. Conf. on Automated Deduction*, 2013, LNAI 7898, pp. 39–57. Full version: Research Report MPI-I-2013-RG1-002, Max-Planck-Institut für Informatik, Saarbrücken, Germany, June 2013, `http://domino.mpi-inf.mpg.de/internet/reports.nsf/NumberView/2013-RG1-002`.
4. L. E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *Amer. J. Math.*, 35(4):413–422, 1913.
5. E. Kruglov and C. Weidenbach. Superposition decides the first-order logic fragment over ground theories. *Math. in Comp. Sci.*, pp. 1–30, 2012.