Automated Theorem Proving

Prof. Dr. Jasmin Blanchette, Lydia Kondylidou,
Yiming Xu, PhD, and Tanguy Bozec
based on exercises by Dr. Uwe Waldmann

Winter Term 2024/25

Exercises 13: Superposition Continued

Exercise 13.1: Using the kbo with $f \succ b \succ c \succ d \succ e$ and weight 1 for all symbols and variables as the term ordering, compute the rewrite systems R_C and R_∞ for the set of ground clauses N:

$$f(b) \approx e \lor f(b) \not\approx f(b) \qquad (1)$$
$$b \not\approx e \lor f(c) \approx f(e) \qquad (2)$$
$$f(d) \approx f(e) \qquad (3)$$
$$f(e) \approx e \lor f(e) \approx c \qquad (4)$$
$$b \approx c \qquad (5)$$
$$d \approx e \qquad (6)$$

Which is the smallest clause $C \in N$ such that C is neither productive nor true in R_C? Use it to show that N is not saturated up to redundancy.

Proposed solution.

The following table summarizes the candidate interpretation construction:

Iter.	Clause C	R_C	E_C
0	$d \approx e$	$\emptyset$	$\{d \to e\}$
1	$b \approx c$	$\{d \to e\}$	$\{b \to c\}$
2	$f(e) \approx e \lor f(e) \approx c$	$\{d \to e, b \to c\}$	$\{f(e) \to c\}$
3	$f(d) \approx f(e)$	$\{d \to e, b \to c, f(e) \to c\}$	$\emptyset$
4	$b \not\approx e \lor f(c) \approx f(e)$	$\{d \to e, b \to c, f(e) \to c\}$	$\{f(c) \to f(e)\}$
5	$f(b) \approx e \lor f(b) \not\approx f(b)$	$\{d \to e, b \to c, f(e) \to c, f(c) \to f(e)\}$	$\emptyset$

The smallest clause that is neither productive nor true in R_C is $f(b) \approx e \lor f(b) \not\approx f(b)$. By Thm. 5.4.8 ("Model Construction"), the existence of such a clause means that the set N is not saturated up to redundancy.

Exercise 13.2: Compute R_∞ for the clause set $\{f(x) \approx b\}$ and the signature $\Sigma = (\{f/1,\ g/1,\ b/0\}, \emptyset)$. Use the kbo with $g \succ f \succ b$ and weights 1 for all symbols and variables.

Proposed solution. The following table summarizes the first 15 iterations of the candidate interpretation construction:

Iter.	Clause C	R_C	E_C
0	$f(b) \approx b$	$\emptyset$	$\{f(b) \to b\}$
1	$f(f(b)) \approx b$	$\{f(b) \to b\}$	$\emptyset$
2	$f(g(b)) \approx b$	$\{f(b) \to b\}$	$\{f(g(b)) \to b\}$
3	$f(f(f(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
4	$f(f(g(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
5	$f(g(f(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
6	$f(g(g(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\{f(g(g(b))) \to b\}$
7	$f(f(f(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
8	$f(f(f(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
9	$f(f(g(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
10	$f(f(g(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
11	$f(g(f(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
12	$f(g(f(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
13	$f(g(g(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
14	$f(g(g(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\{f(g(g(g(b)))) \to b\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

From this, we infer that ground clauses of the form $f(g^i(b)) \approx b$, for $i \in \mathbb{N}$, are productive, and only those. Thus $R_\infty = \{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b, \ldots\}$.

Exercise 13.3: Compute R_∞ for the clause set $\{f(x) \approx b\}$ and the signature $\Sigma = (\{f/1,\ g/1,\ b/0\}, \emptyset)$. This time, use the lpo with the precedence $g \succ f \succ b$.

Proposed solution. The following table summarizes the candidate interpretation construction:

Iter.	Clause C	R_C	E_C
0	$f(b) \approx b$	$\emptyset$	$\{f(b) \to b\}$
1	$f(f(b)) \approx b$	$\{f(b) \to b\}$	$\emptyset$
2	$f(f(f(b))) \approx b$	$\{f(b) \to b\}$	$\emptyset$
3	$f(f(f(f(b)))) \approx b$	$\{f(b) \to b\}$	$\emptyset$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
ω	$f(g(b)) \approx b$	$\{f(b) \to b\}$	$\{f(g(b)) \to b\}$
$\omega + 1$	$f(f(g(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
$\omega + 2$	$f(g(f(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
$\omega + 3$	$f(f(f(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
$\omega + 4$	$f(f(g(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
$\omega + 5$	$f(g(f(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\emptyset$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
2ω	$f(g(g(b))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b\}$	$\{f(g(g(b))) \to b\}$
$2\omega + 1$	$f(f(g(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
$2\omega + 2$	$f(g(f(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
$2\omega + 3$	$f(g(g(f(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\emptyset$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
3ω	$f(g(g(g(b)))) \approx b$	$\{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b\}$	$\{f(g(g(g(b)))) \to b\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

From this, we infer that ground clauses of the form $f(g^i(b)) \approx b$, for $i \in \mathbb{N}$, are productive, and only those. Thus $R_\infty = \{f(b) \to b,\ f(g(b)) \to b,\ f(g(g(b))) \to b, \ldots\}$.

Exercise 13.4: Let N be a set of equational clauses such that $\bot \notin N$. In Thm. 5.4.8, we have shown that whenever N is saturated up to redundancy, then every ground instance $C\theta \in G_\Sigma(N)$ is either productive or true in $R_{C\theta}$. The converse does not hold, not even for ground unit clauses: Give a small set of ground unit clauses N such that $\bot \notin N$ and every $C \in N$ is either productive or true in R_C, but N is not saturated up to redundancy.

Proposed solution. We use the kbo with $f \succ c \succ b$ and weight 1 for all symbols and variables as the term ordering. Consider the clause set $N = \{c \approx b,\ f(c) \not\approx b\}$.

The following table summarizes the candidate interpretation construction:

Iter.	Clause C	R_C	E_C
0	$c \approx b$	$\emptyset$	$\{c \to b\}$
1	$f(c) \not\approx b$	$\{c \to b\}$	$\emptyset$

We get $R_\infty = \{c \to b\}$ as a model for both clauses. Yet a "Negative Superposition" inference from $c \approx b$ and $f(c) \not\approx b$, with nonredundant conclusion $f(b) \not\approx b$, is possible.

Exercise 13.5: A clause is called *Horn* if it contains at most one positive literal. Prove that every inference of the superposition calculus from Horn premises generates a Horn conclusion.

Proposed solution. For "Positive Superposition" and "Negative Superposition," C' and D' consist of only negative literals; hence the conclusion is Horn. For "Equality Resolution," if the premise is Horn, then C' is Horn, and so is the conclusion. Finally, the case of "Equality Factoring" is impossible: The premise of that rule is never Horn.

Exercise 13.6: We call an equational clause *happy* if it contains at least one positive literal.

(a) Prove that every inference of the superposition calculus from happy premises generates a happy conclusion.

(b) Using part (a) and the refutational completeness of superposition, prove that all sets N of happy clauses are satisfiable.

(c) Re-prove the result of part (b) using basic model theory.

Proposed solution. (a) We inspect the rules of the superposition calculus. For "Positive Superposition" and "Equality Factoring," the conclusion always contains a positive literal. For "Negative Superposition," if the right premise is happy, then C' must contain a positive literal and is part of the conclusion. For "Equality Resolution," if the premise is happy, then C' must contain a positive literal and is part of the conclusion.

(b) Consider the set M defined inductively as the smallest set that includes N and that is closed under the application of rules of the superposition calculus. Clearly, M is saturated up to redundancy. Moreover, because inferences preserve happiness, every clause in M is happy. As a result, the unhappy empty clause is not in M. By refutational completeness of superposition, M is satisfiable, and hence $N \subseteq M$ is satisfiable.

Alternative proof: By contradiction. Suppose that N consists of happy clauses but is unsatisfiable. By refutational completeness of superposition, there exists a derivation tree of the empty clause from clauses in N. Each inner node in that tree represents the application of an inference rule of the superposition calculus. Since the leaf nodes are happy, the inner nodes are transitively all happy, including the empty clause at the root of the tree. Contradiction.

(c) Consider the interpretation $\mathcal{A}$ with a domain of cardinality 1 that equates all terms. Every positive equality is true in $\mathcal{A}$, and hence every happy clause is true in $\mathcal{A}$.

Exercise 13.7 (∗)**:** Find an unsatisfiable clause set N consisting of two unit clauses $s \approx t$ and $u \not\approx v$ and a term ordering $\succ$ such that the only nonredundant inference that does not violate the ordering restrictions of the superposition calculus is a "Positive Superposition" inference in which the left-hand side of $s \approx t$ is unified with the left-hand side of a renamed copy of $s \approx t$.

Proposed solution. We use the kbo with $d \succ c \succ b$ and weight 1 for all symbols and variables as the term ordering. We take $s := d$, $t := x$, $u := c$, and $v := b$. Then the only nonredundant inference is

$$\frac{d \approx x \quad d \approx y}{x \approx y}$$

Once this inference is performed, the rest of the derivation is straightforward: A "Negative Superposition" inference from $x \approx y$ and $c \not\approx b$ yields $y \not\approx b$. Then from $y \not\approx b$, an "Equality Resolution" inference generates the empty clause. By soundness of the superposition calculus, this means that the initial clause set is unsatisfiable.

Exercise 13.8 (∗)**:** Prove the lifting lemma (Lemma 5.4.6) for the "Equality Resolution" inference rule.

Proposed solution. Let ι be a ground "Equality Resolution" inference from $C\theta$. For ι to be possible, C must be of the form $C' \lor s \not\approx s'$, and θ must satisfy $s\theta = s'\theta$. Thus,

$$\iota = \frac{C'\theta \lor s\theta \not\approx s'\theta}{C'\theta}$$

Since $s\theta = s'\theta$, the terms s and s' are unifiable. Let σ be an mgu of s and s' such that $\theta = \sigma \circ \tau$. Without loss of generality, σ is idempotent, hence $\sigma \circ \theta = \sigma \circ \sigma \circ \tau = \sigma \circ \tau = \theta$.

For ι to be a ground inference, $s\theta \not\approx s'\theta$ must be a maximal literal in $C\theta$, which implies that $s\sigma \not\approx s'\sigma$ must be a maximal literal in $C\sigma$. (If it were not maximal, then $L\sigma \succ s\sigma \not\approx s'\sigma$ for some other literal $L\sigma$ in $C\sigma$; hence $L\theta \succ s\theta \not\approx s'\theta$ for a literal $L\theta$ in $C\theta$, contradicting the maximality of $s\theta \not\approx s'\theta$.) Thus, from $s\theta \succ s'\theta$ we conclude $s\sigma \not\preceq s'\sigma$ (since $s\sigma \preceq s'\sigma$ would imply $s\theta \preceq s'\theta$). Therefore

$$\iota' = \frac{C' \lor s \not\approx s'}{C'\sigma}$$

is an "Equality Resolution" inference from C. Moreover, by idempotence of σ, $C'\sigma\theta = C'\theta$, so ι is in the grounding of ι'.