The Superposition Calculus

Hans de Nivelle and Marc Bezem.

Our goal is to extend resolution to equality.

We first discuss the problems with explicit axiomatization of equality, and after that introduce the superposition calculus.

We sketch its completeness proof.

Equality

Definition: We introduce atoms of form $t_1 \approx t_2$, denoting t_1 equals t_2.

We assume that $t_1 \approx t_2$ and $t_2 \approx t_1$ are the same atom.

Explicit Axiomatization

A possible approach to equality is explicit axiomatization:

EQREFL $[X \approx X]$,

EQTRANS $[X \not\approx Y,\ Y \not\approx Z,\ X \approx Z]$,

EQFUNC For each function symbol, put

$$[X_1 \not\approx Y_1, \ldots, X_n \not\approx Y_n,\ f(X_1, \ldots, X_n) \approx f(Y_1, \ldots, Y_n)],$$

EQPRED

$$[X_1 \not\approx Y_1, \ldots, X_n \not\approx Y_n,\ \neg p(X_1, \ldots, X_n),\ p(Y_1, \ldots, Y_n)].$$

Superposition/ Paramodulation

In order to deal with equality directly, resolution is replaced by superposition.

Analogeous to resolution without equality, superposition is defined on the ground level, and after that lifted to the variable level.

Informally, ground superposition is the following rule:

From $[\ A[t_1]\ ] \cup R_1$ and $[\ t_1 \approx t_2\ ] \cup R_2$ derive $[\ A[t_2]\ ] \cup R_1 \cup R_2$.

An order $\succ$ on terms and equalities is used to restrict the possible inferences, (similar to L-orders)

This order decides which equalities can be used, and also in which direction the equality $t_1 \approx t_2$ can be used.

Reduction Orders

Definition: A reduction order is a total order on terms that satisfies the following additional conditions:

WF In every non-empty set of terms T over a finite signature there is at least one minimal term t. This is a term $t \in T$, such that there is no $t' \in T$ with $t \succ t'$.

CONT Relation $\succ$ is preserved in contexts: If $t_1 \succ t_2$, then $t[t_1] \succ t[t_2]$.

T The truth constant **true** is minimal: If $t \neq \mathbf{true}$, then $t \succ \mathbf{true}$.

Condition WF ensures that $\succ$ can be used for transfinite induction. In propositional logic, signatures were finite, so that WF is implicit.

The reason for condition T will come later.

Obtaining reduction orders is much harder than obtaining L-orders, mainly due to condition WF.

Simple lexicographic comparison on prefix representations is not WF.

If one puts $a \succ s$, then $a \succ s(a) \succ s(s(a)) \succ \cdots$, so that the set $\{a, s(a), s(s(a)), s^3(a), s^4(a), \ldots\}$ has no minimal element.

Knuth-Bendix orders can be obtained by first comparing size, then using alphabetic, lexicographic comparison on the prefix representations.

Elimination of Non-Equality

Before we give the superposition calculus, we eliminate non-equality atoms.

Positive non-equality literals of form $p(t_1, \ldots, t_n)$ can be replaced by $p(t_1, \ldots, t_n) \approx \text{true}$.

Negative non-equality literals of form $\neg p(t_1, \ldots, t_n)$ can be replaced by $p(t_1, \ldots, t_n) \not\approx \text{true}$.

We first give the ground calculus, after that the non-ground calculus.

POSSUPERPOS Let $[t_1 \approx t_2] \cup R_1$ and $[u_1[t_1] \approx u_2] \cup R_2$ be clauses, s.t.

$(t_1 \approx t_2) \succ R_1$,
$(u_1[t_1] \approx u_2) \succ R_2$,
$t_1 \succ t_2$.

Then the clause $[u_1[t_2] \approx u_2] \cup R_1 \cup R_2$ is obtained by positive superposition.

NEGSUPERPOS Let $[t_1 \approx t_2] \cup R_1$ and $[u_1[t_1] \not\approx u_2] \cup R_2$ be clauses, s.t.

$(t_1 \approx t_2) \succ R_1$,
$(u_1[t_1] \not\approx u_2) \succ R_2$,
$t_1 \succ t_2$.

Then the clause $[u_1[t_2] \not\approx u_2] \cup R_1 \cup R_2$ is obtained by negative superposition.

EQREFL Let $[t \not\approx t] \cup R$ be a clause s.t. $(t \not\approx t) \succeq R$. Then R is obtained by equality reflexivity from $[t \not\approx t] \cup R$.

EQFACT Let $[t_1 \approx t_2,\ t_1 \approx t_3\ ] \cup R$ be a clause, s.t. $t_1 \succ t_2 \succeq t_3$. Then the clause

$$[\ t_1 \approx t_2,\ t_2 \not\approx t_3\ ] \cup R$$

is an equality factor of c.

POSSUPERPOS Let $[t_1 \approx t_2] \cup R_1$ and $[u_1[t_1'] \approx u_2] \cup R_2$ be clauses, s.t. t_1 and t_1' are unifiable with unifier Θ, and the constraints

$$(t_1\Theta \approx t_2\Theta) \succ R_1\Theta, \quad (u_1[t_1']\Theta \approx u_2\Theta) \succ R_2\Theta, \quad t_1\Theta \succ t_2\Theta$$

are solvable.

Then the clause $[u_1[t_2]\Theta \approx u_2\Theta] \cup R_1\Theta \cup R_2\Theta$ is obtained by positive superposition.

NEGSUPERPOS Let $[t_1 \approx t_2] \cup R_1$ and $[u_1[t_1'] \not\approx u_2] \cup R_2$ be clauses, s.t. t_1 and t_1' are unifiable with unifier Θ, and the constraints

$$(t_1\Theta \approx t_2\Theta) \succ R_1\Theta, \quad (u_1[t_1']\Theta \not\approx u_2\Theta) \succ R_2\Theta, \quad t_1\Theta \succ t_2\Theta$$

are solvable.

Then the clause $[u_1[t_2]\Theta \not\approx u_2\Theta] \cup R_1\Theta \cup R_2\Theta$ is obtained by negative superposition.

EQREFL Let $[t_1 \not\approx t_2] \cup R$ be a clause, s.t. t_1 and t_2 are unifiable with mgu Θ, and the constraint

$$(t_1\Theta \not\approx t_2\Theta) \succeq R\Theta$$

is solvable. Then $R\Theta$ is obtained by equality reflexivity from $[t_1 \not\approx t_2] \cup R$.

EQFACT Let $[t_1 \approx t_2,\ t_1' \approx t_3\ ] \cup R$ be a clause, s.t. t_1 is unifiable with t_1' with mgu Θ, and the constraint

$$t_1\Theta \succ t_2\Theta \succeq t_3\Theta$$

is solvable. Then the clause

$$[\ t_1\Theta \approx t_2\Theta,\ t_2\Theta \not\approx t_3\Theta\ ] \cup R\Theta$$

is an equality factor of c.

From $[p(t_1,\ldots,t_n) \approx \text{true}] \cup R_1$ and $[p(t_1,\ldots,t_n) \not\approx \text{true}] \cup R_2$, one can obtain $[\text{true} \not\approx \text{true}] \cup R_1 \cup R_2$ by negative superposition.

This can be simplified into $R_1 \cup R_2$, so that resolution is still present.

From $[p(t_1,\ldots,t_n) \approx \text{true},\ p(t_1,\ldots,t_n) \approx \text{true}] \cup R$ one can derive $[p(t_1,\ldots,t_n) \approx \text{true},\ \text{true} \not\approx \text{true}] \cup R$ by equality factoring.

This can be simplified into

$$[\ p(t_1,\ldots,t_n) \approx \mathbf{true}\ ] \cup R,$$

so that factoring is still present.

Completeness

The completeness proof has the structure as the completeness proof for propositional, L-ordered resolution, but only infinitely more complicated.

Let S be a saturated set. As with propositional resolution, we order the set, using the multiset order. (Here WF is used) After that, we pass through the clauses, and construct an increasing sequence of interpretations $I_0, I_1, I_2, \ldots$.

Instead of literals, it is rewrite rules of form $t_1 \Rightarrow t_2$ that are inserted into the I_i.

Definition An interpretation I is a set of rewrite rules, s.t.

- For each rewrite rule $(t_1 \Rightarrow t_2) \in I, \quad t_1 \succ t_2$.
- For each pair of rewrite rules $(t_1 \Rightarrow t_2), (u_1 \Rightarrow u_2) \in I$, if t_1 is a subterm of u_1, then $t_1 = u_1$ and $t_2 = u_2$.

Given a term t, the normal form of t under I, written as $I(t)$ is the term that one obtains if one applies rewrite rules from I on t as long as possible.

It can be shown that normal forms always exist, and are unique.

Given an interpretation I, a clause C is true in I, if either

- it contains an equality $t_1 \approx t_2$, s.t. $I(t_1) = I(t_2)$, or
- it contains a disequality $t_1 \not\approx t_2$, s.t. $I(t_1) \neq I(t_2)$.

Model Construction

Let S be a saturated set. The construction starts with $I_0 = \{\ \}$.

At each rank i, let $C_i \in S$ be the clause with rank i.

If C_i is false in I_i, and the maximal literal in C_i is a positive equality of form $t_1 \approx t_2$, with $t_1 \neq t_2$, then we may assume without loss of generality that $t_1 \succ t_2$.

If there no equality $t_1 \approx t_3$ in C, such that $I_i(t_2) = I_i(t_3)$, and $I_i(t_1) = t_1$, then put $I_{i+1} = I_i \cup \{t_1 \Rightarrow t_2\}$.

Otherwise, put $I_{i+1} = I_i$.

Redundancy.

The definition of redundancy was:

Clauses $C_1, \ldots, C_n$ make D redundant if $C_1, \ldots, C_n \vdash D$ and $C_1, \ldots, C_n$ come before D in the clause ranking.

In the presence of equality, redundancy becomes much richer:

- $[a \approx b]$ makes $[s(a) \approx s(b)]$ redundant.
- $[a \approx b]$ and $[b \approx c]$ make $[s(a) \approx s(c)]$ redundant.
- $[a \approx b]$ and $[p(b) \approx \mathbf{true}]$ make $[p(a) \approx \mathbf{true}]$ redundant, if $a \succ b$.

The last example makes it possible to use an equality $t_1 \approx t_2$ as simplifier, in case $t_1 \succ t_2$.

Lifting

As for logic without equality, the superposition calculus for ground equations can be lifted to predicate logic.

For EQREFL, and EQFACT, lifting is analogeous to lifting for resolution.

With superposition, there is a problem:

Failure of Lifting for Superposition

$$[a \approx b] \text{ and } [s(a) \approx s(c),\ p(s(a)) \approx \mathbf{true}]$$

$$\Rightarrow [s(b) \approx s(c),\ p(s(a)) \approx \mathbf{true}].$$

Now look at the variable clauses:

$$[a \approx b] \text{ and } [s(X) \approx s(c),\ p(s(X)) \approx \mathbf{true}].$$

$$[a \approx b] \text{ and } [X \approx s(c),\ p(X) \approx \mathbf{true}].$$

Solution: Change the substitition!

If equality replacement in the ground clause takes place at a position which is still present in the variable clause, one can construct a unifier, and there is no problem.

Otherwise, it takes place on a position that is 'inside' a variable X.

Let $C_1 = [t_1 \approx t_2] \cup R_1$ be a clause.

Let $C_2 = [\ u_1[X] \approx u_2[X]\ ] \cup R_2(X)$ be a clause with instance

$$[\ u_1[\ t[t_1]\ ] \approx u_2[\ t[t_1]\ ]\ ] \cup R_2(\ t[t_1]\ )$$

using substitution $X := t[t_1]$.

Chanching the substitution into $X := t[t_2]$ results in the ground clause

$$[\ u_1[\ t[t_2]\ ] \approx u_2[\ t[t_2]\ ]\ ] \cup R_2(\ t[t_2]\ ).$$

This clause, together with C_1 makes the result of positive superposition redundant:

$$[\ u_1[\ t[t_2]\ ] \approx u_2[\ t[t_1]\ ]\ ] \cup R_1 \cup R_2(\ t[t_1]\ )$$

redundant.