Automated Deduction

Laura Kovács

TU Wien

Outline

First-order logic with equality

- Equality predicate: $=$.
- Equality: $l = r$.

The order of literals in equalities does not matter, that is, we consider an equality $l = r$ as a multiset consisting of two terms l, r, and so consider $l = r$ and $r = l$ equal.

Equality. An Axiomatisation (Recap)

- reflexivity axiom: $x = x$;
- symmetry axiom: $x = y \rightarrow y = x$;
- transitivity axiom: $x = y \wedge y = z \rightarrow x = z$;
- function substitution (congruence) axioms:
  $x_1 = y_1 \wedge \ldots \wedge x_n = y_n \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$, for every function symbol f;
- predicate substitution (congruence) axioms:
  $x_1 = y_1 \wedge \ldots \wedge x_n = y_n \wedge P(x_1, \ldots, x_n) \rightarrow P(y_1, \ldots, y_n)$ for every predicate symbol P.

Inference systems for logic with equality

We will define a resolution and superposition inference system. This system is complete. One can eliminate redundancy.

We will first define it only for ground clauses. On the theoretical side,

- Completeness is first proved for ground clauses only.
- It is then "lifted" to arbitrary first-order clauses using a technique called lifting.
- Moreover, this way some notions (ordering, selection function) can first be defined for ground clauses only and then it is relatively easy to see how to generalise them for non-ground clauses.

Simple Ground Superposition Inference System

Superposition: (right and left)

$$\frac{l = r \vee C \quad s[l] = t \vee D}{s[r] = t \vee C \vee D}\ (\text{Sup}), \quad \frac{l = r \vee C \quad s[l] \neq t \vee D}{s[r] \neq t \vee C \vee D}\ (\text{Sup}),$$

Equality Resolution:

$$\frac{s \neq s \vee C}{C}\ (\text{ER}),$$

Equality Factoring:

$$\frac{s = t \vee s = t' \vee C}{s = t \vee t \neq t' \vee C}\ (\text{EF}),$$

Example

$$f(a) = a \vee g(a) = a$$
$$f(f(a)) = a \vee g(g(a)) \neq a$$
$$f(f(a)) \neq a$$

Can this system be used for efficient theorem proving?

Not really. It has too many inferences. For example, from the clause $f(a) = a$ we can derive any clause of the form

$$f^m(a) = f^n(a)$$

where $m, n \geq 0$.
Worst of all, the derived clauses can be much larger than the original clause $f(a) = a$.
The recipe is to use the previously introduced ingredients:

1. Ordering;
2. Literal selection;
3. Redundancy elimination.

Atom and literal orderings on equalities

Equality atom comparison treats an equality $s = t$ as the multiset $\dot{\{}s, t\dot{\}}$.

- $(s' = t') \succ_{lit} (s = t)$ if $\dot{\{}s', t'\dot{\}} \succ \dot{\{}s, t\dot{\}}$
- $(s' \neq t') \succ_{lit} (s \neq t)$ if $\dot{\{}s', t'\dot{\}} \succ \dot{\{}s, t\dot{\}}$

with $\succ_{lit}$ being an induced ordering on literals.

Ground Superposition Inference System $\mathbb{S}\mathrm{up}_{\succ,\sigma}$

Let σ be a well-behaved literal selection function.

Superposition: (right and left)

$$\frac{l = r \vee C \quad s[l] = t \vee D}{s[r] = t \vee C \vee D}\ (\mathrm{Sup}), \qquad \frac{l = r \vee C \quad s[l] \neq t \vee D}{s[r] \neq t \vee C \vee D}\ (\mathrm{Sup}),$$

where (i) $l \succ r$, (ii) $s[l] \succ t$, (iii) $l = r$ is strictly greater than any literal in C, (iv) (only for the superposition-right rule) $s[l] = t$ is greater than or equal to any literal in D.

Equality Resolution:

$$\frac{s \neq s \vee C}{C}\ (\mathrm{ER}),$$

Equality Factoring:

$$\frac{s = t \vee s = t' \vee C}{s = t \vee t \neq t' \vee C}\ (\mathrm{EF}),$$

where (i) $s \succ t \succeq t'$; (ii) $s = t$ is greater than or equal to any literal in C.

Extension to arbitrary (non-equality) literals

- Consider a two-sorted logic in which equality is the only predicate symbol.
- Interpret terms as terms of the first sort and non-equality atoms as terms of the second sort.
- Add a constant $\top$ of the second sort.
- Replace non-equality atoms $p(t_1, \ldots, t_n)$ by equalities of the second sort $p(t_1, \ldots, t_n) = \top$.

For example, the clause

$$p(a, b) \vee \neg q(a) \vee a \neq b$$

becomes

$$p(a, b) = \top \vee q(a) \neq \top \vee a \neq b.$$

Binary resolution inferences can be represented by inferences in the superposition system

We ignore selection functions.

$$\frac{A \vee C_1 \quad \neg A \vee C_2}{C_1 \vee C_2}\ (\mathrm{BR})$$

$$\frac{\dfrac{A = \top \vee C_1 \quad A \neq \top \vee C_2}{\top \neq \top \vee C_1 \vee C_2}\ (\mathrm{Sup})}{C_1 \vee C_2}\ (\mathrm{ER})$$

Exercise

Positive factoring can also be represented by inferences in the superposition system.

Outline

Simplification Ordering

When we deal with equality, we need to work with term orderings.
Consider a strict ordering $\succ$ on signature symbols, such that $\succ$ is well-founded.
The ordering $\succ$ on terms is called a simplification ordering if

1. $\succ$ is well-founded;
2. $\succ$ is monotonic: if $l \succ r$, then $s[l] \succ s[r]$;
3. $\succ$ is stable under substitutions: if $l \succ r$, then $l\theta \succ r\theta$.

One can combine the last two properties into one:

2a. If $l \succ r$, then $s[l\theta] \succ s[r\theta]$.

A General Property of Term Orderings

If $\succ$ is a simplification ordering, then for every term $t[s]$ and its proper subterm s we have $s \not\succ t[s]$. Why?

Consider an example.

$$f(a) = a$$
$$f(f(a)) = a$$
$$f(f(f(a))) = a$$

Then both $f(f(a)) = a$ and $f(f(f(a))) = a$ are redundant. The clause $f(a) = a$ is a logical consequence of $\{f(f(a)) = a, f(f(f(a))) = a\}$ but is not redundant.

Exercise: Show that $\{f(a) = a, f(f(f(a))) \neq a\}$ is unsatisfiable, by using superposition with redundancy elimination.

How to "come up" with simplification orderings?

Term Algebra

Term algebra $TA(\Sigma)$ of signature Σ:

- Domain: the set of all ground terms of Σ.
- Interpretation of any function symbol f or constant c is defined as follows::

$$f_{TA(\Sigma)}(t_1, \ldots, t_n) \stackrel{\text{def}}{\Leftrightarrow} f(t_1, \ldots, t_n);$$
$$c_{TA(\Sigma)} \stackrel{\text{def}}{\Leftrightarrow} c.$$

Knuth-Bendix Ordering (KBO), Ground Case

Let us fix

- Signature Σ, it induces the term algebra $TA(\Sigma)$.
- Total ordering $\gg$ on Σ, called precedence relation;
- Weight function $w : \Sigma \to \mathbb{N}$.

Weight of a ground term t is

$$|g(t_1, \ldots, t_n)| = w(g) + \sum_{i=1}^{n} |t_i|.$$

$g(t_1, \ldots, t_n) \succ_{KB} h(s_1, \ldots, s_m)$ if

1. $|g(t_1, \ldots, t_n)| > |h(s_1, \ldots, s_m)|$ (by weight) or
2. $|g(t_1, \ldots, t_n)| = |h(s_1, \ldots, s_m)|$ and one of the following holds:
   - 2.1 $g \gg h$ (by precedence) or
   - 2.2 $g = h$ and for some $1 \leq i \leq n$ we have $t_1 = s_1, \ldots, t_{i-1} = s_{i-1}$ and $t_i \succ_{KB} s_i$ (lexicographically).

Example

$$\begin{aligned} w(a) &= 1 \\ w(b) &= 2 \\ w(f) &= 3 \\ w(g) &= 0 \end{aligned}$$

$$|f(g(a), f(a, b))| = |3(0(1), 3(1, 2))| = 3 + 0 + 1 + 3 + 1 + 2 = 10.$$

The Knuth-Bendix ordering is the main ordering used in Vampire and all other resolution and superposition theorem provers.

Same Property as for $\mathbb{BR}_\sigma$

The conclusion is strictly smaller than the rightmost premise:

$$\frac{\underline{l = r} \vee C \quad s[\underline{l}] = t \vee D}{s[r] = t \vee C \vee D}\ \text{(Sup)}, \qquad \frac{\underline{l = r} \vee C \quad s[\underline{l}] \neq t \vee D}{s[r] \neq t \vee C \vee D}\ \text{(Sup)},$$

where (i) $l \succ r$, (ii) $s[l] \succ t$, (iii) $l = r$ is strictly greater than any literal in C, (iv) $s[l] = t$ is greater than or equal to any literal in D.

New redundancy

Consider a superposition with a unit left premise:

$$\frac{\underline{l = r} \quad \underline{s[l] = t \vee D}}{s[r] = t \vee D} \text{ (Sup)},$$

Note that we have

$$l = r, s[r] = t \vee D \models s[l] = t \vee D$$

and we have

$$s[l] = t \vee D \succ s[r] = t \vee D.$$

If we also have $s[l] = t \vee D \succ l = r$, then the second premise is redundant and can be removed.

This rule (superposition plus deletion) is sometimes called demodulation (also rewriting by unit equalities).

Exercise

Consider the KBO ordering $\succ$ generated by:

– the precedence $P \gg Q \gg f \gg a$;

and

– the weight function w with $w(P) = w(Q) = 2$, $w(f) = w(a) = 1$.

Consider the set of clauses S to be:

$$\begin{array}{l} Q(a), \\ \neg Q(a) \vee f(a) = a, \\ \neg P(a), \\ P(f(a))\}. \end{array}$$

Apply saturation on S by using an inferece process with redundancy based on the (ground) superposition calculus $\mathbb{S}\mathrm{up}_{\succ,\sigma}$.