Model Existence Theorem

Theorem 3.24 (Bachmair & Ganzinger 1990) *Let $\succ$ be a clause ordering, let N be saturated w. r. t. Res, and suppose that $\bot \notin N$. Then $I_N^{\succ} \models N$.*

Corollary 3.25 *Let N be saturated w. r. t. Res. Then $N \models \bot \Leftrightarrow \bot \in N$.*

Proof of Theorem 3.24. Suppose $\bot \notin N$, but $I_N^{\succ} \not\models N$. Let $C \in N$ minimal (in $\succ$) such that $I_N^{\succ} \not\models C$. Since C is false in I_N, C is not productive. As $C \neq \bot$ there exists a maximal atom A in C.

Case 1: $C = \neg A \vee C'$ (i. e., the maximal atom occurs negatively)
$\Rightarrow I_N \models A$ and $I_N \not\models C'$
$\Rightarrow$ some $D = D' \vee A \in N$ produces A. As $\frac{D' \vee A \qquad \neg A \vee C'}{D' \vee C'}$, we infer that $D' \vee C' \in N$, and $C \succ D' \vee C'$ and $I_N \not\models D' \vee C'$
$\Rightarrow$ contradicts minimality of C.

Case 2: $C = C' \vee A \vee A$. Then $\frac{C' \vee A \vee A}{C' \vee A}$ yields a smaller counterexample $C' \vee A \in N$. $\Rightarrow$ contradicts minimality of C. □

Compactness of Propositional Logic

Theorem 3.26 (Compactness) *Let N be a set of propositional formulas. Then N is unsatisfiable, if and only if some finite subset $M \subseteq N$ is unsatisfiable.*

Proof. "$\Leftarrow$": trivial.

"$\Rightarrow$": Let N be unsatisfiable.
$\Rightarrow$ $Res^*(N)$ unsatisfiable
$\Rightarrow$ $\bot \in Res^*(N)$ by refutational completeness of resolution
$\Rightarrow$ $\exists n \geq 0 : \bot \in Res^n(N)$
$\Rightarrow$ $\bot$ has a finite resolution proof P;
choose M as the set of assumptions in P. □

3.12 General Resolution

Propositional resolution:

refutationally complete,

in its most naive version: not guaranteed to terminate for satisfiable sets of clauses, (improved versions do terminate, however)

in practice clearly inferior to the DPLL procedure (even with various improvements).

But: in contrast to the DPLL procedure, resolution can be easily extended to non-ground clauses.

General Resolution through Instantiation

Idea: instantiate clauses appropriately:

$$
\begin{tikzcd}
P(z',z') \lor \neg Q(z) \arrow[d, "{[a/z', f(a,b)/z]}"'] & \neg P(a,y) \arrow[d, "{[a/y]}"'] \arrow[dr, "{[b/y]}"] & & P(x',b) \lor Q(f(x',x)) \arrow[d, "{[a/x', b/x]}"] \\
P(a,a) \lor \neg Q(f(a,b)) \arrow[dr] & \neg P(a,a) \arrow[d] & \neg P(a,b) \arrow[d] & P(a,b) \lor Q(f(a,b)) \arrow[dl] \\
& \neg Q(f(a,b)) \arrow[d] & Q(f(a,b)) \arrow[dl] & \\
& \bot & &
\end{tikzcd}
$$

Problems:

More than one instance of a clause can participate in a proof.

Even worse: There are infinitely many possible instances.

Observation:

Instantiation must produce complementary literals (so that inferences become possible).

Idea:

Do not instantiate more than necessary to get complementary literals.

Idea: do not instantiate more than necessary:

$$
\begin{tikzcd}
P(z',z') \vee \neg Q(z) \arrow[d, "{[a/z']}"'] & & & \neg P(a,y) \arrow[dl, "{[a/y]}"'] \arrow[dr, "{[b/y]}"] & & & P(x',b) \vee Q(f(x',x)) \arrow[d, "{[a/x']}"] \\
P(a,a) \vee \neg Q(z) \arrow[dr] & & \neg P(a,a) \arrow[dl] & & \neg P(a,b) \arrow[dr] & & P(a,b) \vee Q(f(a,x)) \arrow[dl] \\
& \neg Q(z) \arrow[d, "{[f(a,x)/z]}"'] & & & & Q(f(a,x)) \arrow[d] & \\
& \neg Q(f(a,x)) \arrow[drr] & & & & Q(f(a,x)) \arrow[dll] & \\
& & & \bot & & &
\end{tikzcd}
$$

Lifting Principle

Problem: Make saturation of infinite sets of clauses as they arise from taking the (ground) instances of finitely many *general* clauses (with variables) effective and efficient.

Idea (Robinson 1965):

- Resolution for general clauses:
- *Equality* of ground atoms is generalized to *unifiability* of general atoms;
- Only compute *most general* (minimal) unifiers.

Significance: The advantage of the method in (Robinson 1965) compared with (Gilmore 1960) is that unification enumerates only those instances of clauses that participate in an inference. Moreover, clauses are not right away instantiated into ground clauses. Rather they are instantiated only as far as required for an inference. Inferences with non-ground clauses in general represent infinite sets of ground inferences which are computed simultaneously in a single step.

Resolution for General Clauses

General binary resolution *Res*:

$$\frac{D \vee B \qquad C \vee \neg A}{(D \vee C)\sigma} \quad \text{if } \sigma = \mathrm{mgu}(A, B) \qquad \text{[resolution]}$$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \quad \text{if } \sigma = \mathrm{mgu}(A, B) \quad \text{[factorization]}$$

General resolution RIF with implicit factorization:

$$\frac{D \vee B_1 \vee \ldots \vee B_n \qquad C \vee \neg A}{(D \vee C)\sigma} \quad \text{if } \sigma = \mathrm{mgu}(A, B_1, \ldots, B_n)$$

[RIF]

For inferences with more than one premise, we assume that the variables in the premises are (bijectively) renamed such that they become different to any variable in the other premises. We do not formalize this. Which names one uses for variables is otherwise irrelevant.

Unification

Let $E = \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\}$ (s_i, t_i terms or atoms) a multi-set of *equality problems*. A substitution σ is called a *unifier* of E if $s_i\sigma = t_i\sigma$ for all $1 \leq i \leq n$.

If a unifier of E exists, then E is called *unifiable*.

A substitution σ is called *more general* than a substitution τ, denoted by $\sigma \leq \tau$, if there exists a substitution ρ such that $\rho \circ \sigma = \tau$, where $(\rho \circ \sigma)(x) := (x\sigma)\rho$ is the composition of σ and ρ as mappings. (Note that $\rho \circ \sigma$ has a finite domain as required for a substitution.)

If a unifier of E is more general than any other unifier of E, then we speak of a *most general unifier* of E, denoted by $\mathrm{mgu}(E)$.

Proposition 3.27

(i) $\leq$ *is a quasi-ordering on substitutions, and* $\circ$ *is associative.*

(ii) If $\sigma \leq \tau$ *and* $\tau \leq \sigma$ *(we write* $\sigma \sim \tau$ *in this case), then* $x\sigma$ *and* $x\tau$ *are equal up to (bijective) variable renaming, for any* x *in* X.

A substitution σ is called *idempotent*, if $\sigma \circ \sigma = \sigma$.

Proposition 3.28 σ *is idempotent iff* $dom(\sigma) \cap codom(\sigma) = \emptyset$.

Rule Based Naive Standard Unification

$$
\begin{array}{rll}
t \doteq t, E & \Rightarrow_{SU} & E \\
f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E & \Rightarrow_{SU} & s_1 \doteq t_1, \ldots, s_n \doteq t_n, E \\
f(\ldots) \doteq g(\ldots), E & \Rightarrow_{SU} & \bot \\
x \doteq t, E & \Rightarrow_{SU} & x \doteq t, E[t/x] \\
 & \text{if } x \in var(E), x \notin var(t) & \\
x \doteq t, E & \Rightarrow_{SU} & \bot \\
 & \text{if } x \neq t, x \in var(t) & \\
t \doteq x, E & \Rightarrow_{SU} & x \doteq t, E \\
 & \text{if } t \notin X &
\end{array}
$$

SU: Main Properties

If $E = x_1 \doteq u_1, \ldots, x_k \doteq u_k$, with x_i pairwise distinct, $x_i \notin var(u_j)$, then E is called an (equational problem in) *solved form* representing the solution $\sigma_E = [u_1/x_1, \ldots, u_k/x_k]$.

Proposition 3.29 *If E is a solved form then σ_E is an mgu of E.*

Theorem 3.30

1. *If $E \Rightarrow_{SU} E'$ then σ is a unifier of E iff σ is a unifier of E'*
2. *If $E \Rightarrow^*_{SU} \bot$ then E is not unifiable.*
3. *If $E \Rightarrow^*_{SU} E'$ with E' in solved form, then $\sigma_{E'}$ is an mgu of E.*

Proof. (1) We have to show this for each of the rules. Let's treat the case for the 4th rule here. Suppose σ is a unifier of $x \doteq t$, that is, $x\sigma = t\sigma$. Thus, $\sigma \circ [t/x] = \sigma[x \mapsto t\sigma] = \sigma[x \mapsto x\sigma] = \sigma$. Therefore, for any equation $u \doteq v$ in E: $u\sigma = v\sigma$, iff $u[t/x]\sigma = v[t/x]\sigma$. (2) and (3) follow by induction from (1) using Proposition 3.29. $\square$

Main Unification Theorem

Theorem 3.31 *E is unifiable if and only if there is a most general unifier σ of E, such that σ is idempotent and $dom(\sigma) \cup codom(\sigma) \subseteq var(E)$.*

Problem: *exponential growth* of terms possible

Proof of Theorem 3.31.

- $\Rightarrow_{SU}$ is Noetherian. A suitable lexicographic ordering on the multisets E (with $\bot$ minimal) shows this. Compare in this order:

1. the number of defined variables (d.h. variables x in equations $x \doteq t$ with $x \notin var(t)$), which also occur outside their definition elsewhere in E;

2. the multi-set ordering induced by (i) the size (number of symbols) in an equation; (ii) if sizes are equal consider $x \doteq t$ smaller than $t \doteq x$, if $t \notin X$. $\square$

- A system E that is irreducible w. r. t. $\Rightarrow_{SU}$ is either $\bot$ or a solved form.
- Therefore, reducing any E by SU will end (no matter what reduction strategy we apply) in an irreducible E' having the same unifiers as E, and we can read off the mgu (or non-unifiability) of E from E' (Theorem 3.30, Proposition 3.29).
- σ is idempotent because of the substitution in rule 4. $dom(\sigma) \cup codom(\sigma) \subseteq var(E)$, as no new variables are generated.

Rule Based Polynomial Unification

$$
\begin{array}{rcl}
t \doteq t, E & \Rightarrow_{PU} & E \\
f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E & \Rightarrow_{PU} & s_1 \doteq t_1, \ldots, s_n \doteq t_n, E \\
f(\ldots) \doteq g(\ldots), E & \Rightarrow_{PU} & \bot \\
x \doteq y, E & \Rightarrow_{PU} & x \doteq y, E[y/x] \\
& & \text{if } x \in var(E), x \neq y \\
x_1 \doteq t_1, \ldots, x_n \doteq t_n, E & \Rightarrow_{PU} & \bot
\end{array}
$$

if there are positions p_i with $t_i/p_i = x_{i+1}, t_n/p_n = x_1$ and some $p_i \neq \epsilon$

$$
\begin{array}{rcl}
x \doteq t, E & \Rightarrow_{PU} & \bot \\
& & \text{if } x \neq t, x \in var(t) \\
t \doteq x, E & \Rightarrow_{PU} & x \doteq t, E \\
& & \text{if } t \notin X \\
x \doteq t, x \doteq s, E & \Rightarrow_{PU} & x \doteq t, t \doteq s, E \\
& & \text{if } t, s \notin X \text{ and } |t| \leq |s|
\end{array}
$$

Properties of PU

Theorem 3.32

1. *If $E \Rightarrow_{PU} E'$ then σ is a unifier of E iff σ is a unifier of E'*
2. *If $E \Rightarrow^*_{PU} \bot$ then E is not unifiable.*
3. *If $E \Rightarrow^*_{PU} E'$ with E' in solved form, then $\sigma_{E'}$ is an mgu of E.*