Automated Theorem Proving

Lecture 6: General Resolution

Prof. Dr. Jasmin Blanchette
based on slides by Dr. Uwe Waldmann

Winter Term 2024/25

3.10 General Resolution

Propositional (ground) resolution:

refutationally complete,

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

inferior to the CDCL procedure.

But in contrast to the CDCL procedure, resolution can be easily extended to nonground clauses.

Observation

If $\mathcal{A}$ is a model of an (implicitly universally quantified) clause C, then by Lemma 3.3.8 it is also a model of all (implicitly universally quantified) instances $C\sigma$ of C.

Consequently, if we show that some instances of clauses in a set N are unsatisfiable, then we have also shown that N itself is unsatisfiable.

General Resolution through Instantiation

Idea: instantiate clauses appropriately:

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

General Resolution through Instantiation

Early approaches (Gilmore 1960, Davis and Putnam 1960):

Generate ground instances of clauses.

Try to refute the set of ground instances by resolution.

If no contradiction is found, generate more ground instances.

Problems:

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

Even worse: There are infinitely many possible instances.

General Resolution through Instantiation

Observation:

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

General Resolution through Instantiation

Idea (Robinson 1965):

Do not instantiate more than necessary to get complementary literals
$\Rightarrow$ most general unifiers (mgu).

Calculus works with nonground clauses;
inferences with nonground clauses represent infinite sets
of ground inferences which are computed simultaneously
$\Rightarrow$ lifting principle.

Computation of instances becomes a by-product of boolean reasoning.

General Resolution through Instantiation

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

Unification

Let $E = \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\}$ (s_i, t_i terms or atoms) be a multiset 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.

Unification

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 $\mathrm{mgu}(E)$.

Unification

Proposition 3.10.1:

(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 3.10.2:
σ is idempotent if and only if $\mathrm{dom}(\sigma) \cap \mathrm{codom}(\sigma) = \emptyset$.

Rule-Based Naive Standard Unification

$$
\begin{array}{rll}
t \doteq t, E & \Rightarrow_{SU} & E \\[1em]
f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E & \Rightarrow_{SU} & s_1 \doteq t_1, \ldots, s_n \doteq t_n, E \\[1em]
f(\ldots) \doteq g(\ldots), E & \Rightarrow_{SU} & \perp \\
& & \text{if } f \neq g \\[1em]
x \doteq t, E & \Rightarrow_{SU} & x \doteq t, E\{x \mapsto t\} \\
& & \text{if } x \in \mathrm{var}(E), x \notin \mathrm{var}(t) \\[1em]
x \doteq t, E & \Rightarrow_{SU} & \perp \\
& & \text{if } x \neq t, x \in \mathrm{var}(t) \\[1em]
t \doteq x, E & \Rightarrow_{SU} & x \doteq t, E \\
& & \text{if } t \notin X
\end{array}
$$

Properties of SU

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

Proposition 3.10.3:
If E is a solved form then σ_E is an mgu of E.

Properties of SU

Theorem 3.10.4:

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

Main Unification Theorem

Theorem 3.10.5:
E is unifiable if and only if there is a most general unifier σ of E such that σ is idempotent and $\mathrm{dom}(\sigma) \cup \mathrm{codom}(\sigma) \subseteq \mathrm{var}(E)$.

Example of SU

Example 3.10.6:

We unify $g(x, f(x))$ and $g(b, y)$ using standard unification:

$$
\begin{aligned}
& g(x, f(x)) \doteq g(b, y) \\
\Rightarrow_{SU} \; & x \doteq b,\ f(x) \doteq y \\
\Rightarrow_{SU} \; & x \doteq b,\ f(b) \doteq y \\
\Rightarrow_{SU} \; & x \doteq b,\ y \doteq f(b) \\
\Rightarrow_{SU} \; & x \doteq b,\ y \doteq f(b)
\end{aligned}
$$

Resulting substitution: $\{x \mapsto b,\ y \mapsto f(b)\}$.

Exponential Growth of SU

Problem: Using $\Rightarrow_{SU}$, an *exponential growth* of terms is possible.

Example 3.10.7:
We unify $g(x, y, z)$ and $g(f(y, y), f(z, z), f(a, a))$ using SU:

$$
\begin{aligned}
& g(x, y, z) \doteq g(f(y, y), f(z, z), f(a, a)) \\
\Rightarrow_{SU} \; & x \doteq f(y, y),\ y \doteq f(z, z),\ z \doteq f(a, a) \\
\Rightarrow_{SU} \; & x \doteq f(f(z, z), f(z, z)),\ y \doteq f(z, z),\ z \doteq f(a, a) \\
\Rightarrow_{SU} \; & x \doteq f(f(f(a, a), f(a, a)), f(f(a, a), f(a, a))),\ y \doteq f(f(a, a), f(a, a)), \\
& z \doteq f(a, a)
\end{aligned}
$$

Resulting substitution: $\{x \mapsto f(f(f(a, a), f(a, a)), f(f(a, a), f(a, a))),\ y \mapsto f(f(a, a), f(a, a)),\ z \mapsto f(a, a)\}$.

Rule-Based Polynomial Unification

The following unification algorithm avoids the exponential growth problem, at least if the final solved form is represented as a DAG.

Rule-Based Polynomial Unification

$$
\begin{array}{rcl}
t \doteq t, E & \Rightarrow_{PU} & E \\
f(s_1, \dots, s_n) \doteq f(t_1, \dots, t_n), E & \Rightarrow_{PU} & s_1 \doteq t_1, \dots, s_n \doteq t_n, E \\
f(\dots) \doteq g(\dots), E & \Rightarrow_{PU} & \bot \\
& & \text{if } f \neq g \\
x \doteq y, E & \Rightarrow_{PU} & x \doteq y, E\{x \mapsto y\} \\
& & \text{if } x \in \mathrm{var}(E), x \neq y \\
x_1 \doteq t_1, \dots, x_n \doteq t_n, E & \Rightarrow_{PU} & \bot \\
& & \text{if there are positions } p_i \text{ with} \\
& & t_i|_{p_i} = x_{i+1}, t_n|_{p_n} = x_1 \\
& & \text{and some } p_i \neq \varepsilon
\end{array}
$$

Rule-Based Polynomial Unification

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

Properties of PU

Theorem 3.10.8:

1. If $E \Rightarrow_{PU} E'$ then σ is a unifier of E if and only if 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.

The solved form of $\Rightarrow_{PU}$ is different from the solved form obtained from $\Rightarrow_{SU}$. To obtain the unifier $\sigma_{E'}$, we have to sort the list of equality problems $x_i \doteq t_i$ in such a way that x_i does not occur in t_j for $j < i$, and then we have to compose the substitutions $\{x_1 \mapsto t_1\} \circ \cdots \circ \{x_k \mapsto t_k\}$.

Example of PU

Example 3.10.9:
We unify $g(x, f(x))$ and $g(b, y)$ using polynomial unification:

$$\begin{aligned} & g(x, f(x)) \doteq g(b, y) \\ \Rightarrow_{PU} \ & x \doteq b,\ f(x) \doteq y \\ \Rightarrow_{PU} \ & x \doteq b,\ y \doteq f(x) \end{aligned}$$

Resulting substitution: $\{x \mapsto b\} \circ \{y \mapsto f(x)\} = \{x \mapsto b,\ y \mapsto f(b)\}$.

Polynomial Growth of PU

Example 3.10.10:
We unify $g(x, y, z)$ and $g(f(y, y), f(z, z), f(a, a))$ using PU:

$$\begin{aligned} & g(x, y, z) \doteq g(f(y, y), f(z, z), f(a, a)) \\ \Rightarrow_{PU} \; & x \doteq f(y, y),\ y \doteq f(z, z),\ z \doteq f(a, a) \\ = \; & z \doteq f(a, a),\ y \doteq f(z, z),\ x \doteq f(y, y) \end{aligned}$$

Resulting substitution: $\{z \mapsto f(a, a)\} \circ \{y \mapsto f(z, z)\} \circ \{x \mapsto f(y, y)\}$.

Resolution for General Clauses

We obtain the resolution inference rules for nonground clauses from the inference rules for ground clauses by replacing equality by unifiability:

General resolution *Res***:**

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

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

Resolution for General Clauses

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.

Resolution for General Clauses

Example 3.10.11:

Consider the clauses

$$P(z', z') \lor \neg Q(z) \qquad (1)$$

$$\neg P(a, y) \qquad (2)$$

$$P(x', b) \lor Q(f(x', x)) \qquad (3)$$

From (1) and (2), using "Resolution" we obtain $\neg Q(z)$ (4).

From (3) and (2), using "Resolution" we obtain $Q(f(a, x))$ (5).

From (5) and (4), using "Resolution" we obtain the empty clause.

Lifting Lemma

Lemma 3.10.12:

Let C and D be variable-disjoint clauses. If

$$\begin{array}{ccc} D & & C \\ \big\downarrow \theta_1 & & \big\downarrow \theta_2 \\ \end{array}$$

$$\frac{D\theta_1 \qquad C\theta_2}{C'} \qquad \text{[ground resolution]}$$

then there exists a substitution ρ such that

$$\frac{D \qquad C}{C''} \qquad \text{[general resolution]}$$

$$\begin{array}{c} \big\downarrow \rho \\ C' = C''\rho \end{array}$$

Lifting Lemma

An analogous lifting lemma holds for factorization.

Saturation of Sets of General Clauses

Corollary 3.10.13:
Let N be a set of general clauses saturated under Res, i.e., $Res(N) \subseteq N$.
Then also $G_\Sigma(N)$ is saturated, that is,

$$Res(G_\Sigma(N)) \subseteq G_\Sigma(N).$$

Soundness for General Clauses

Proposition 3.10.14:

The general resolution calculus is sound.

Herbrand's Theorem

Lemma 3.10.15:

Let N be a set of Σ-clauses, let $\mathcal{A}$ be an interpretation.

Then $\mathcal{A} \models N$ implies $\mathcal{A} \models G_\Sigma(N)$.

Lemma 3.10.16:

Let N be a set of Σ-clauses, let $\mathcal{A}$ be an *Herbrand* interpretation.

Then $\mathcal{A} \models G_\Sigma(N)$ implies $\mathcal{A} \models N$.

Herbrand's Theorem

Theorem 3.10.17 (Herbrand):
A set N of Σ-clauses is satisfiable if and only if it has an Herbrand model over Σ.

Corollary 3.10.18:
A set N of Σ-clauses is satisfiable if and only if its set of ground instances $G_\Sigma(N)$ is satisfiable.

Refutational Completeness of General Resolution

Theorem 3.10.19:

Let N be a set of general clauses that is saturated w.r.t. *Res*.

Then $N \models \bot$ if and only if $\bot \in N$.

3.11 Theoretical Consequences

We get some classical results on properties of first-order logic as easy corollaries.

The Theorem of Löwenheim-Skolem

Theorem 3.11.1 (Löwenheim–Skolem):
Let Σ be a countable signature and let S be a set of closed Σ-formulas. Then S is satisfiable if and only if S has a model over a countable universe.

There exist more refined versions of this theorem. For instance, one can show that if S has some infinite model, then S has a model with a universe of cardinality κ for every κ that is larger than or equal to the cardinalty of the signature Σ.

Compactness of Predicate Logic

Theorem 3.11.2 (Compactness Theorem for First-Order Logic):

Let S be a set of closed first-order formulas.

S is unsatisfiable $\Leftrightarrow$ some finite subset $S' \subseteq S$ is unsatisfiable.