COMP6012: Automated Reasoning, Part II
Advanced Topics

Optional Material

Renate Schmidt

School of Computer Science
University of Manchester

schmidt@cs.man.ac.uk

http://www.cs.man.ac.uk/~schmidt/COMP60121/

Overview ...

These slides cover additional topics which couldn't be covered in lectures. In particular:

- Soundness and refutational completeness of *Res* for first-order clause logic
- Lexicographic orderings, reduction orderings

COMP6012: Automated Reasoning II

Optional material (unassessed)

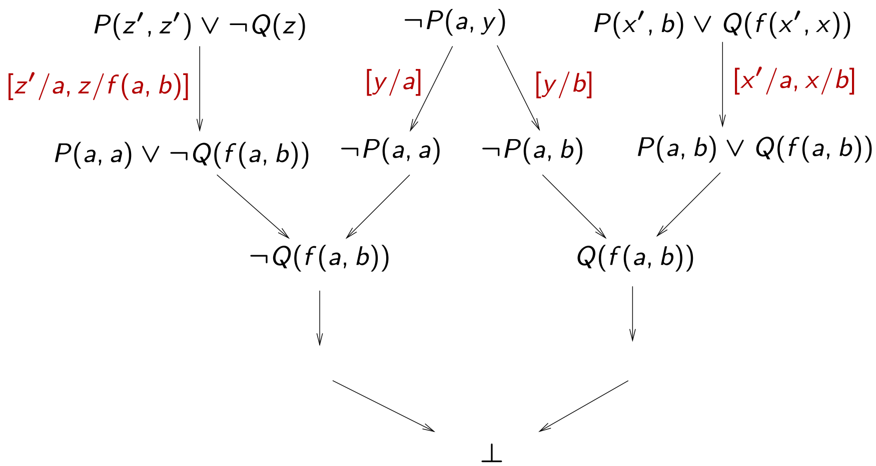
Generalising Resolution to Non-Ground Clauses

- Propositional/ground resolution:
  - refutationally complete,
  - in its most naive version: not guaranteed to terminate for satisfiable sets of clauses, (improved versions do terminate, however)
  - 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:

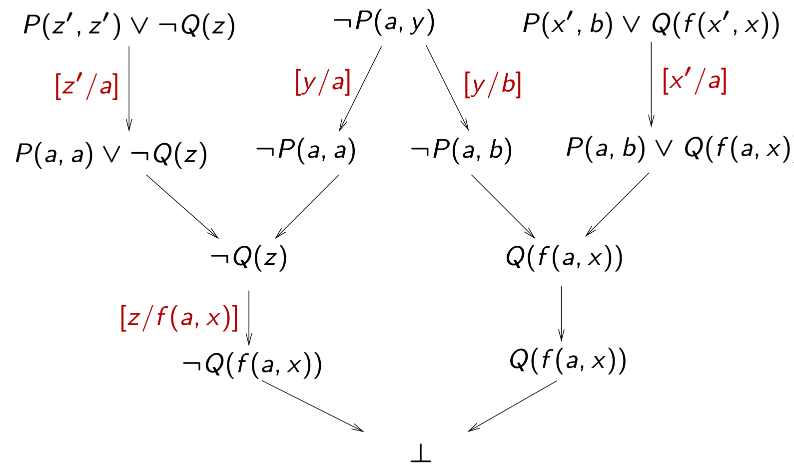


General Resolution through Instantiation: Problems

- 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.

General Resolution through Lazy Instantiation

Idea: do not instantiate more than necessary:



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 (matching) is generalised to unifiability of general atoms;
  - Only compute most general (minimal) unifiers.

Lifting Principle (cont'd)

- **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 calculus *Res*:**

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

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \quad \text{if } \sigma = \mathrm{mgu}(A, B) \quad (\text{positive factoring})$$

- **General resolution calculus *RIF* with implicit factoring:**

$$\frac{C \vee A_1 \vee \ldots \vee A_n \qquad D \vee \neg B}{(C \vee D)\sigma} \qquad (\text{RIF})$$

if $\sigma = \mathrm{mgu}(A_1, \ldots, A_n, B)$

Resolution for General Clauses (cont'd)

- 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.

Lifting Lemma

Lemma 1

Let C and D be variable-disjoint clauses. If

$$\begin{array}{ccc} C & & D \\ \downarrow \sigma & & \downarrow \rho \\ \hline \end{array}$$

$$\frac{C\sigma \qquad D\rho}{C'} \qquad (\text{propositional resolution})$$

then there exist C'' and a substitution τ such that

$$\frac{C \qquad D}{C''} \qquad (\text{general resolution})$$

$$\downarrow \tau$$

$$C' = C''\tau$$

Lifting Lemma (cont'd)

- An analogous lifting lemma holds for factoring.

Saturation of Sets of General Clauses

Recall that $G_\Sigma(N)$ denotes the set of ground instances of N over the signature Σ.

Corollary 2

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,

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

Saturation of Sets of General Clauses (cont'd)

Proof:

W.l.o.g. we may assume that clauses in N are pairwise variable-disjoint. (Otherwise make them disjoint, and this renaming process changes neither $Res(N)$ nor $G_\Sigma(N)$.)

Let $C' \in Res(G_\Sigma(N))$, meaning

(i) there exist resolvable ground instances $C\boldsymbol{\sigma}$ and $D\boldsymbol{\rho}$ of C and D belonging to N and C' is their resolvent, or else

(ii) C' is a factor of a ground instance $C\boldsymbol{\sigma}$ of $C \in N$.

Case (i): By the Lifting Lemma, C and D are resolvable with a resolvent C'' with $C''\boldsymbol{\tau} = C'$, for a suitable ground substitution $\boldsymbol{\tau}$. As $C'' \in N$ by assumption, we obtain that $C' \in G_\Sigma(N)$.

Case (ii): Similar (exercise).

Herbrand's Theorem

Lemma 3

Let N be a set of Σ-clauses, let $\mathcal{M}$ be an interpretation. Then $\mathcal{M} \models N$ implies $\mathcal{M} \models G_\Sigma(N)$.

Lemma 4

Let N be a set of Σ-clauses, let I be a Herbrand interpretation. Then $I \models G_\Sigma(N)$ implies $I \models N$.

Herbrand's Theorem (cont'd)

Property 5 (Herbrand Theorem)

A set N of Σ-clauses is satisfiable iff it has a Herbrand model over Σ.

Proof:

The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let $N \not\models \bot$.

$$
\begin{aligned}
N \not\models \bot \;&\Rightarrow\; \bot \notin Res^*(N) \qquad \text{(resolution is sound)}\\
&\Rightarrow\; \bot \notin G_\Sigma(Res^*(N))\\
&\Rightarrow\; I_{G_\Sigma(Res^*(N))} \models G_\Sigma(Res^*(N)) \qquad \text{(Prt. 12 (BG90); Cor. 2)}\\
&\Rightarrow\; I_{G_\Sigma(Res^*(N))} \models Res^*(N) \qquad \text{(Lemma 4)}\\
&\Rightarrow\; I_{G_\Sigma(Res^*(N))} \models N \qquad (N \subseteq Res^*(N))
\end{aligned}
$$

The Theorem of Löwenheim-Skolem

Property 6 (Löwenheim-Skolem Theorem)

Let Σ be a countable signature and let S be a set of closed Σ-formulae. Then S is satisfiable iff S has a model over a countable universe.

Proof:

If both X and Σ are countable, then S can be at most countably infinite. Now generate, maintaining satisfiability, a set N of clauses from S. This extends Σ by at most countably many new Skolem functions to Σ'. As Σ' is countable, so is $T_{\Sigma'}$, the universe of Herbrand-interpretations over Σ'. Now apply Herbrand's Theorem (Property 5).

Refutational Completeness of General Resolution

Property 7

Let N be a set of general clauses where $Res(N) \subseteq N$. Then

$$N \models \bot \;\text{ iff }\; \bot \in N.$$

Proof:

Let $Res(N) \subseteq N$.

By Corollary 2: $Res(G_\Sigma(N)) \subseteq G_\Sigma(N)$

$$
\begin{aligned}
N \models \bot \;&\Leftrightarrow\; G_\Sigma(N) \models \bot \qquad \text{(Lemmas 3 \& 4; Property 5)}\\
&\Leftrightarrow\; \bot \in G_\Sigma(N) \qquad \text{(prop. resol. is sound and complete)}\\
&\Leftrightarrow\; \bot \in N
\end{aligned}
$$

Compactness of First-Order Logic

Property 8 (Compactness Theorem for First-Order Logic)

Let Φ be a set of first-order formulae.

Φ is unsatisfiable iff some finite subset $\Psi \subseteq \Phi$ is unsatisfiable.

Proof:

The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let Φ be unsatisfiable and let N be the set of clauses obtained by Skolemisation and CNF transformation of the formulae in Φ. Clearly $Res^*(N)$ is unsatisfiable. By Property 7, $\bot \in Res^*(N)$, and therefore $\bot \in Res^n(N)$ for some $n \in \mathbb{N}$. Consequently, $\bot$ has a finite resolution proof Π of depth $\leq n$. Choose Ψ as the subset of formulae in Φ such that the corresponding clauses contain the assumptions (leaves) of Π.

Lifting Lemma for $Res_S^{\succ}$

Lemma 9

Let C and D be variable-disjoint clauses. If

$$\begin{array}{ccc} C & & D \\ \downarrow \sigma & & \downarrow \rho \\ \end{array} \qquad \frac{C\sigma \qquad D\rho}{C'} \quad \text{(propositional inference in } Res_S^{\succ}\text{)}$$

and if $S(C\sigma) \simeq S(C)$, $S(D\rho) \simeq S(D)$ (that is, "corresponding" literals are selected), then there exist C'' and a substitution τ s.t.

$$\frac{C \qquad D}{C''} \quad \text{(inference in } Res_S^{\succ}\text{)}$$

$$\downarrow \tau$$

$$C' = C''\tau$$

Lifting Lemma for $Res_S^{\succ}$ (cont'd)

- An analogous lifting lemma holds for factoring.

Saturation of General Clause Sets

Corollary 10

Let N be a set of general clauses saturated under $Res_S^{\succ}$, i.e. $Res_S^{\succ}(N) \subseteq N$. Then there exists a selection function S' such that $S|_N = S'|_N$ and $G_\Sigma(N)$ is also saturated, i.e.,

$$Res_{S'}^{\succ}(G_\Sigma(N)) \subseteq G_\Sigma(N).$$

Proof:

We first define the selection function S' such that $S'(C) = S(C)$ for all clauses $C \in G_\Sigma(N) \cap N$. For $C \in G_\Sigma(N) \setminus N$ we choose a fixed but arbitrary clause $D \in N$ with $C \in G_\Sigma(D)$ and define $S'(C)$ to be those occurrences of literals that are ground instances of the occurrences selected by S in D. Then proceed as in the proof of Corollary 2 using the above lifting lemma.

Soundness and Refutational Completeness

Property 11

Let $\succ$ be an atom ordering and S a selection function such that $Res_S^{\succ}(N) \subseteq N$. Then

$$N \models \bot \text{ iff } \bot \in N$$

Proof:

The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part consider the propositional level: Construct a candidate model $I_N^{\succ}$ as for unrestricted resolution, except that clauses C in N that have selected literals are not productive, even when they are false in I_C and when their maximal atom occurs only once and positively. The result for general clauses follows using Corollary 10.

Summary

- Resolution for general, first-order clauses
- Refutational completeness, consequence of:
  1. refutational completeness of ground case
  2. lifting lemmas
  3. Herbrand's Theorem
- Löwenheim–Skolem Theorem
- Compactness of FOL
- lifting lemmas for ordered resolution with selection

COMP6012: Automated Reasoning II

Optional material (unassessed)

Lexicographic Orderings

- **Lexicographic orderings:** Let $(X_1, \succ_1)$, $(X_2, \succ_2)$ be well-founded orderings. Define their lexicographic combination

$$\succ \; = (\succ_1, \succ_2)_{\text{lex}}$$

as an ordering on $X_1 \times X_2$ such that

$$(x_1, x_2) \succ (y_1, y_2) \quad \text{iff} \quad \text{(i) } x_1 \succ_1 y_1, \text{ or else} \quad \text{(ii) } x_1 = y_1 \text{ and } x_2 \succ_2 y_2$$

(Analogously for more than two orderings.)

This again yields a well-founded ordering (proof below).

- **Notation:** $\succ_{\text{lex}}$ for the lexicographic combination of $(X, \succ)$ twice (in general n times). I.e. $\succ_{\text{lex}} = (\succ, \succ)_{\text{lex}}$.

Lexicographic Orderings: Examples

- **Length-based ordering on words:** For alphabets Σ with a well-founded ordering $>_\Sigma$, the relation $\succ$, defined as

$$w \succ w' \quad \text{iff} \quad \text{(i) } |w| > |w'| \text{ or} \quad \text{(ii) } |w| = |w'| \text{ and } w >_{\Sigma,\text{lex}} w',$$

is a well-founded ordering on Σ^*.

- **Notation:** $>_{\Sigma,\text{lex}} = (>_\Sigma)_{\text{lex}}$.

Lexicographic Combinations of Well-Founded Orderings

Lemma 12
$(X_i, \succ_i)$ is well-founded for $i \in \{1,2\}$ iff $(X_1 \times X_2, \succ)$ with
$\succ = (\succ_1, \succ_2)_{\text{lex}}$ is well-founded.

Proof:
(i) "$\Rightarrow$": Suppose $(X_1 \times X_2, \succ)$ is not well-founded. Then there is an infinite sequence $(a_0, b_0) \succ (a_1, b_1) \succ (a_2, b_2) \succ \ldots$.
Let $A = \{a_i \mid i \geq 0\} \subseteq X_1$. Since $(X_1, \succ_1)$ is well-founded, A has a minimal element a_n. But then $B = \{b_i \mid i \geq n\} \subseteq X_2$ cannot have a minimal element, contradicting the well-foundedness of $(X_2, \succ_2)$.
(ii) "$\Leftarrow$": obvious (exercise).

Reduction orderings

- A strict ordering $\succ$ is a reduction ordering iff
  - (i) $\succ$ is well-founded
  - (ii) $\succ$ is stable under substitutions, i.e.
    $s \succ t$ implies $s\sigma \succ t\sigma$
    for all terms s, t and substitutions σ
  - (iii) $\succ$ is compatible with contexts, i.e.
    $s \succ t$ implies $u[s] \succ u[t]$
    for all terms s, t and contexts u
- Examples:
  - For (ii): $f(x) \succ g(x)$ implies $f(a) \succ g(a)$.
  - For (iii): $a \succ b$ implies $f(a) \succ f(a)$.

Lexicographic Path Orderings

- Let Σ be a finite signature and let X be a countably infinite set of variables.
- Let $\succ$ be a strict ordering (precedence) on the set of predicate and functions symbols in Σ.
- The lexicographic path ordering $\succ_{\text{lpo}}$ on the set of terms (and atoms) over Σ and X is an ordering induced by $\succ$, satisfying:
  $s \succ_{\text{lpo}} t$ iff
  1. $t \in \text{var}(s)$ and $t \neq s$, or
  2. $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and
     - (a) $s_i \succeq_{\text{lpo}} t$ for some i, or
     - (b) $f \succ g$ and $s \succ_{\text{lpo}} t_j$ for all j, or
     - (c) $f = g$, $s \succ_{\text{lpo}} t_j$ for all j, and
       $(s_1, \ldots, s_m)\ (\succ_{\text{lpo}})_{\text{lex}}\ (t_1, \ldots, t_n)$.

Lexicographic Path Orderings (cont'd)

- Definition: $s \succeq_{\text{lpo}} t$ iff $s \succ_{\text{lpo}} t$ or $s = t$
- Examples:
  - $f(x) \succ_{\text{lpo}} x$
  - if t is a subterm of s then $s \succ_{\text{lpo}} t$
  - $f(a, b, g(c), a) \succ_{\text{lpo}} f(a, b, c, g(b))$
  - If t can be homomorphically embedded into s and $s \neq t$ then $s \succ_{\text{lpo}} t$
    E.g.
    $$h(f(g(a), f(b, y))) \succ_{\text{lpo}} f(g(a), y)$$

Properties of LPOs

Lemma 13

$s \succ_{\mathrm{lpo}} t$ implies $\mathrm{var}(s) \supseteq \mathrm{var}(t)$.

Proof:

By induction on $|s| + |t|$ and case analysis.

Properties of LPOs (cont'd)

Property 14

$\succ_{\mathrm{lpo}}$ is a reduction ordering on the set of terms (and atoms) over Σ and X.

Proof:

Show transitivity, stability under substitutions, compatibility contexts, and irreflexivity, usually by induction on the sum of the term sizes and case analysis.

Details: Baader and Nipkow, page 119–120.

Properties of LPOs (cont'd)

Property 15

If the precedence $\succ$ is total, then the lexicographic path ordering $\succ_{\mathrm{lpo}}$ is total on ground terms (and ground atoms), i.e. for all ground terms (or atoms) s, t of the following is true: $s \succ_{\mathrm{lpo}} t$ or $t \succ_{\mathrm{lpo}} s$ or $s = t$.

Proof:

By induction on $|s| + |t|$ and case analysis.

Variations of the LPO

There are several possibilities to compare subterms in 2.(c):

- compare list of subterms lexicographically left-to-right ("lexicographic path ordering (lpo)", Kamin and Lvy)
- compare list of subterms lexicographically right-to-left (or according to some permutation π)
- compare multiset of subterms using the multiset extension ("multiset path ordering (mpo)", Dershowitz)
- to each function symbol f/n associate a status $\in \{mul\} \cup \{\, lex_\pi \mid \pi : \{1, \ldots, n\} \to \{1, \ldots, n\}\,\}$ and compare according to that status ("recursive path ordering (rpo) with status")