3.12 Ordered Resolution with Selection

Motivation: Search space for *Res* *very* large.

Ideas for improvement:

1. In the completeness proof (Model Existence Theorem 3.19) one only needs to resolve and factor maximal atoms
   $\Rightarrow$ if the calculus is restricted to inferences involving maximal atoms, the proof remains correct
   $\Rightarrow$ *ordering restrictions*
2. In the proof, it does not really matter with which negative literal an inference is performed
   $\Rightarrow$ choose a negative literal don't-care-nondeterministically
   $\Rightarrow$ *selection*

Selection Functions

A *selection function* is a mapping

$$S : C \quad \mapsto \quad \text{set of occurrences of } \textit{negative} \text{ literals in } C$$

Example of selection with selected literals indicated as $\boxed{X}$:

$$\boxed{\neg A} \vee \neg A \vee B$$

$$\boxed{\neg B_0} \vee \boxed{\neg B_1} \vee A$$

Intuition:

- If a clause has at least one selected literal, compute only inferences that involve a selected literal.
- If a clause has no selected literals, compute only inferences that involve a maximal literal.

Resolution Calculus $Res_S^{\succ}$

The resolution calculus $Res_S^{\succ}$ is parameterized by

- a selection function S
- and a total and well-founded atom ordering $\succ$.

In the completeness proof, we talk about (strictly) maximal literals of *ground* clauses.

In the non-ground calculus, we have to consider those literals that correspond to (strictly) maximal literals of ground instances:

A literal L is called *[strictly] maximal* in a clause C if and only if there exists a ground substitution σ such that $L\sigma$ is [strictly] maximal in $C\sigma$ (i.e., if for no other L' in C: $L\sigma \prec L'\sigma$ $[L\sigma \preceq L'\sigma]$).

$$\frac{D \vee B \qquad C \vee \neg A}{(D \vee C)\sigma} \qquad [\textit{ordered resolution with selection}]$$

if the following conditions are satisfied:

(i) $\sigma = \mathrm{mgu}(A, B)$;

(ii) $B\sigma$ strictly maximal in $D\sigma \vee B\sigma$;

(iii) nothing is selected in $D \vee B$ by S;

(iv) either $\neg A$ is selected, or else nothing is selected in $C \vee \neg A$ and $\neg A\sigma$ is maximal in $C\sigma \vee \neg A\sigma$.

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \qquad [\textit{ordered factorization}]$$

if the following conditions are satisfied:

(i) $\sigma = \mathrm{mgu}(A, B)$;

(ii) $A\sigma$ is maximal in $C\sigma \vee A\sigma \vee B\sigma$;

(iii) nothing is selected in $C \vee A \vee B$ by S.

Special Case: Propositional Logic

For ground clauses the resolution inference rule simplifies to

$$\frac{D \vee A \qquad C \vee \neg A}{D \vee C}$$

if the following conditions are satisfied:

(i) $A \succ D$;

(ii) nothing is selected in $D \vee A$ by S;

(iii) $\neg A$ is selected in $C \vee \neg A$, or else nothing is selected in $C \vee \neg A$ and $\neg A \succeq \max(C)$.

Note: For positive literals, $A \succ D$ is the same as $A \succ \max(D)$.

Analogously, the factorization rule simplifies to

$$\frac{C \vee A \vee A}{C \vee A}$$

if the following conditions are satisfied:

(i) A is the largest literal in $C \vee A \vee A$;

(ii) nothing is selected in $C \vee A \vee A$ by S.

Search Spaces Become Smaller

1	$A \vee B$	
2	$A \vee \boxed{\neg B}$	
3	$\neg A \vee B$	
4	$\neg A \vee \boxed{\neg B}$	
5	$B \vee B$	Res 1, 3
6	B	Fact 5
7	$\neg A$	Res 6, 4
8	A	Res 6, 2
9	$\bot$	Res 8, 7

we assume $A \succ B$ and S as indicated by $\boxed{X}$. The maximal literal in a clause is depicted in red.

With this ordering and selection function the refutation proceeds strictly deterministically in this example. Generally, proof search will still be non-deterministic but the search space will be much smaller than with unrestricted resolution.

Avoiding Rotation Redundancy

From

$$\frac{\dfrac{C_1 \vee A \quad C_2 \vee \neg A \vee B}{C_1 \vee C_2 \vee B} \qquad C_3 \vee \neg B}{C_1 \vee C_2 \vee C_3}$$

we can obtain by *rotation*

$$\frac{C_1 \vee A \qquad \dfrac{C_2 \vee \neg A \vee B \quad C_3 \vee \neg B}{C_2 \vee \neg A \vee C_3}}{C_1 \vee C_2 \vee C_3}$$

another proof of the same clause. In large proofs many rotations are possible. However, if $A \succ B$, then the second proof does not fulfill the orderings restrictions.

Conclusion: In the presence of orderings restrictions (however one chooses $\succ$) no rotations are possible. In other words, orderings identify exactly one representant in any class of rotation-equivalent proofs.

Lifting Lemma for $Res^{\succ}_S$

Lemma 3.36 *Let D and C be variable-disjoint clauses. If*

$$\begin{array}{c} \begin{array}{cc} D & C \\ \downarrow \sigma & \downarrow \rho \end{array} \\ \dfrac{D\sigma \qquad C\rho}{C'} \end{array} \qquad \textit{[propositional inference in } Res^{\succ}_S\textit{]}$$

and if $S(D\sigma) \simeq S(D)$, $S(C\rho) \simeq S(C)$ (that is, "corresponding" literals are selected), then there exists a substitution τ such that

$$\begin{array}{c} \dfrac{D \qquad C}{C''} \\ \downarrow \tau \\ C' = C''\tau \end{array} \qquad \textit{[inference in } Res^{\succ}_S\textit{]}$$

An analogous lifting lemma holds for factorization.

Saturation of General Clause Sets

Corollary 3.37 *Let N be a set of general clauses saturated under $Res^{\succ}_S$, i. e., $Res^{\succ}_S(N) \subseteq N$. Then there exists a selection function S' such that $S|_N = S'|_N$ and $G_\Sigma(N)$ is also saturated, i. e.,*

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

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 Cor. 3.29 using the above lifting lemma. □

Soundness and Refutational Completeness

Theorem 3.38 *Let $\succ$ be an atom ordering and S a selection function such that $Res^{\succ}_S(N) \subseteq N$. Then*

$$N \models \bot \Leftrightarrow \bot \in N$$

Proof. The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part consider first the propositional level: Construct a candidate interpretation I_N 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 3.37. □

Craig-Interpolation

A theoretical application of ordered resolution is Craig-Interpolation:

Theorem 3.39 (Craig 1957) *Let F and G be two propositional formulas such that $F \models G$. Then there exists a formula H (called the* interpolant *for $F \models G$), such that H contains only prop. variables occurring both in F and in G, and such that $F \models H$ and $H \models G$.*

Proof. Translate F and $\neg G$ into CNF. let N and M, resp., denote the resulting clause set. Choose an atom ordering $\succ$ for which the prop. variables that occur in F but not in G are maximal. Saturate N into N^* w.r.t. $Res_S^\succ$ with an empty selection function S . Then saturate $N^* \cup M$ w.r.t. $Res_S^\succ$ to derive $\bot$. As N^* is already saturated, due to the ordering restrictions only inferences need to be considered where premises, if they are from N^*, only contain symbols that also occur in G. The conjunction of these premises is an interpolant H. The theorem also holds for first-order formulas. For universal formulas the above proof can be easily extended. In the general case, a proof based on resolution technology is more complicated because of Skolemization. $\square$

Redundancy

So far: local restrictions of the resolution inference rules using orderings and selection functions.

Is it also possible to delete clauses altogether? Under which circumstances are clauses unnecessary? (Conjecture: e. g., if they are tautologies or if they are subsumed by other clauses.)

Intuition: If a clause is guaranteed to be neither a minimal counterexample nor productive, then we do not need it.

A Formal Notion of Redundancy

Let N be a set of ground clauses and C a ground clause (not necessarily in N). C is called *redundant* w.r.t. N, if there exist $C_1, \dots, C_n \in N$, $n \geq 0$, such that $C_i \prec C$ and $C_1, \dots, C_n \models C$.

Redundancy for general clauses: C is called *redundant* w.r.t. N, if all ground instances $C\sigma$ of C are redundant w.r.t. $G_\Sigma(N)$.

Intuition: Redundant clauses are neither minimal counterexamples nor productive.

Note: The same ordering $\prec$ is used for ordering restrictions and for redundancy (and for the completeness proof).

Examples of Redundancy

Proposition 3.40 *Some redundancy criteria:*

- C *tautology (i. e.,* $\models C$*)* $\Rightarrow$ C *redundant w. r. t. any set* N.
- $C\sigma \subset D$ $\Rightarrow$ D *redundant w. r. t.* $N \cup \{C\}$.
- $C\sigma \subseteq D$ $\Rightarrow$ $D \vee \overline{L}\sigma$ *redundant w. r. t.* $N \cup \{C \vee L,\ D\}$.

(Under certain conditions one may also use non-strict subsumption, but this requires a slightly more complicated definition of redundancy.)

Saturation up to Redundancy

N is called *saturated up to redundancy* (w. r. t. $Res_S^{\succ}$)

$$:\Leftrightarrow Res_S^{\succ}(N \setminus Red(N)) \subseteq N \cup Red(N)$$

Theorem 3.41 *Let* N *be saturated up to redundancy. Then*

$$N \models \bot \Leftrightarrow \bot \in N$$

Proof (Sketch). (i) Ground case:

- consider the construction of the candidate interpretation $I_N^{\succ}$ for $Res_S^{\succ}$
- redundant clauses are not productive
- redundant clauses in N are not minimal counterexamples for $I_N^{\succ}$

The premises of "essential" inferences are either minimal counterexamples or productive.

(ii) Lifting: no additional problems over the proof of Theorem 3.38. □

Monotonicity Properties of Redundancy

Theorem 3.42

(i) $N \subseteq M \Rightarrow Red(N) \subseteq Red(M)$

(ii) $M \subseteq Red(N) \Rightarrow Red(N) \subseteq Red(N \setminus M)$

We conclude that redundancy is preserved when, during a theorem proving process, one adds (derives) new clauses or deletes redundant clauses. Recall that $Red(N)$ may include clauses that are not in N.