Avoiding Rotation Redundancy

From

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

we can obtain by *rotation*

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

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 $Res_S^{\succ}$

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

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

$$\dfrac{D\sigma \qquad C\rho}{C'} \qquad \text{[propositional inference in } Res_S^{\succ}\text{]}$$

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

$$\dfrac{D \qquad C}{C''} \qquad \text{[inference in } Res_S^{\succ}\text{]}$$

$$\downarrow \tau$$

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

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 Corollary 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 "⇐" part is trivial. For the "⇒" 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. $Res^{\succ}_{S}$ with an empty selection function S . Then saturate $N^* \cup M$ w. r. t. $Res^{\succ}_{S}$ 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. □

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.

A Resolution Prover

So far: static view on completeness of resolution:

Saturated sets are inconsistent if and only if they contain $\bot$.

We will now consider a dynamic view:

How can we get saturated sets in practice?

The theorems 3.41 and 3.42 are the basis for the completeness proof of our prover RP.

Rules for Simplifications and Deletion

We want to employ the following rules for simplification of prover states N:

- *Deletion of tautologies*

$$N \cup \{C \vee A \vee \neg A\} \;\Rightarrow\; N$$

- *Deletion of subsumed clauses*

$$N \cup \{C, D\} \;\Rightarrow\; N \cup \{C\}$$

  if $C\sigma \subseteq D$ (C *subsumes* D).

- *Reduction* (also called *subsumption resolution*)

$$N \cup \{C \vee L,\;\; D \vee C\sigma \vee \overline{L}\sigma\} \;\Rightarrow\; N \cup \{C \vee L, D \vee C\sigma\}$$

Resolution Prover RP

3 clause sets: N(ew) containing new resolvents
P(rocessed) containing simplified resolvents
clauses get into O(ld) once their inferences have been computed

Strategy: Inferences will only be computed when there are no possibilities for simplification

Transition Rules for RP (I)

Tautology elimination
$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O}$
if C is a tautology

Forward subsumption
$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O}$
if some $D \in \boldsymbol{P} \cup \boldsymbol{O}$ subsumes C

Backward subsumption
$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \cup \{D\} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$
$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{D\} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$
if C strictly subsumes D

Transition Rules for RP (II)

Forward reduction

$$\boldsymbol{N} \cup \{C \vee L\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$$

if there exists $D \vee L' \in \boldsymbol{P} \cup \boldsymbol{O}$ such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$

Backward reduction

$$\boldsymbol{N} \mid \boldsymbol{P} \cup \{C \vee L\} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O}$$

$$\boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{C \vee L\} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O}$$

if there exists $D \vee L' \in \boldsymbol{N}$ such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$

Transition Rules for RP (III)

Clause processing

$$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O}$$

Inference computation

$$\emptyset \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O} \quad \Rightarrow_{RP} \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{C\},$$

with $\boldsymbol{N} = Res_S^{\succ}(\boldsymbol{O} \cup \{C\})$

Soundness and Completeness

Theorem 3.43

$$N \models \bot \quad \Leftrightarrow \quad N \mid \emptyset \mid \emptyset \quad \overset{*}{\Rightarrow}_{RP} \quad N' \cup \{\bot\} \mid \_ \mid \_$$

Proof in L. Bachmair, H. Ganzinger: Resolution Theorem Proving appeared in the Handbook of Automated Reasoning, 2001

Fairness

Problem:

If N is inconsistent, then $N \mid \emptyset \mid \emptyset \overset{*}{\Rightarrow}_{RP} N' \cup \{\bot\} \mid \_ \mid \_$.

Does this imply that *every* derivation starting from an inconsistent set N eventually produces $\bot$?

No: a clause could be kept in $\boldsymbol{P}$ without ever being used for an inference.

We need in addition a *fairness condition:*

If an inference is possible forever (that is, none of its premises is ever deleted), then it must be computed eventually.

One possible way to guarantee fairness: Implement $\boldsymbol{P}$ as a queue (there are other techniques to guarantee fairness).

With this additional requirement, we get a stronger result: If N is inconsistent, then every *fair* derivation will eventually produce $\bot$.

Hyperresolution

There are *many* variants of resolution. (We refer to [Bachmair, Ganzinger: Resolution Theorem Proving] for further reading.)

One well-known example is hyperresolution (Robinson 1965):

Assume that several negative literals are selected in a clause C. If we perform an inference with C, then one of the selected literals is eliminated.

Suppose that the remaining selected literals of C are again selected in the conclusion. Then we must eliminate the remaining selected literals one by one by further resolution steps.

Hyperresolution replaces these successive steps by a single inference. As for $Res_S^{\succ}$, the calculus is parameterized by an atom ordering $\succ$ and a selection function S.

$$\frac{D_1 \vee B_1 \quad \ldots \quad D_n \vee B_n \qquad C \vee \neg A_1 \vee \ldots \vee \neg A_n}{(D_1 \vee \ldots \vee D_n \vee C)\sigma}$$

with $\sigma = \mathrm{mgu}(A_1 \doteq B_1, \ldots, A_n \doteq B_n)$, if

(i) $B_i\sigma$ strictly maximal in $D_i\sigma$, $1 \leq i \leq n$;

(ii) nothing is selected in D_i;

(iii) the indicated occurrences of the $\neg A_i$ are exactly the ones selected by S, or else nothing is selected in the right premise and $n = 1$ and $\neg A_1\sigma$ is maximal in $C\sigma$.

Similarly to resolution, hyperresolution has to be complemented by a factoring inference.

As we have seen, hyperresolution can be simulated by iterated binary resolution.

However this yields intermediate clauses which HR might not derive, and many of them might not be extendable into a full HR inference.

3.13 Summary: Resolution Theorem Proving

- Resolution is a machine calculus.
- Subtle interleaving of enumerating ground instances and proving inconsistency through the use of unification.
- Parameters: atom ordering $\succ$ and selection function S. On the non-ground level, ordering constraints can (only) be solved approximatively.
- Completeness proof by constructing candidate interpretations from productive clauses $C \vee A$, $A \succ C$; inferences with those reduce counterexamples.
- *Local* restrictions of inferences via $\succ$ and S
  $\Rightarrow$ fewer proof variants.
- *Global* restrictions of the search space via elimination of redundancy
  $\Rightarrow$ computing with "smaller" clause sets;
  $\Rightarrow$ termination on many decidable fragments.
- However: not good enough for dealing with orderings, equality and more specific algebraic theories (lattices, abelian groups, rings, fields)
  $\Rightarrow$ further specialization of inference systems required.

3.14 Other Inference Systems

Instantiation-based methods for FOL:

- (Semantic) Tableau;
- Resolution-based instance generation;
- Disconnection calculus.

Further (mainly propositional) proof systems:

- Hilbert calculus;
- Sequent calculus;
- Natural deduction.

Instantiation-Based Methods for FOL

Idea:

Overlaps of complementary literals produce instantiations (as in resolution);

However, contrary to resolution, clauses are not recombined.

Instead: treat remaining variables as constant and use efficient propositional proof methods, such as DPLL.

There are both saturation-based variants, such as partial instantiation [Hooker et al.] or resolution-based instance generation (Inst-Gen) [Ganzinger and Korovin], and tableau-style variants, such as the disconnection calculus [Billon; Letz and Stenz].

Hilbert Calculus

Hilbert calculus:

Direct proof method (proves a theorem from axioms, rather than refuting its negation)

Axiom schemes, e. g.,

$$F \to (G \to F)$$
$$(F \to (G \to H)) \to ((F \to G) \to (F \to H))$$

plus Modus ponens:

$$\frac{F \qquad F \to G}{G}$$

Unsuitable for both humans and machines.

Natural Deduction

Natural deduction (Prawitz):

Models the concept of proofs from assumptions as humans do it (cf. Fitting or Huth/Ryan).

Sequent Calculus

Sequent calculus (Gentzen):

Assumptions internalized into the data structure of sequents

$$F_1, \dots, F_m \to G_1, \dots, G_k$$

meaning

$$F_1 \wedge \dots \wedge F_m \to G_1 \vee \dots \vee G_k$$

A kind of mixture between natural deduction and semantic tableaux.

Perfect symmetry between the handling of assumptions and their consequences.

Can be used both backwards and forwards.