3.11 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 DPLL procedure.

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

Two Lemmas

Lemma 3.21 *Let $\mathcal{A}$ be a Σ-algebra and let F be a Σ-formula with free variables $x_1, \ldots, x_n$. Then*

$$\mathcal{A} \models \forall x_1, \ldots, x_n F \quad \text{if and only if} \quad \mathcal{A} \models F$$

Lemma 3.22 *Let F be a Σ-formula with free variables $x_1, \ldots, x_n$, let σ be a substitution, and let $y_1, \ldots, y_m$ be the free variables of $F\sigma$. Then*

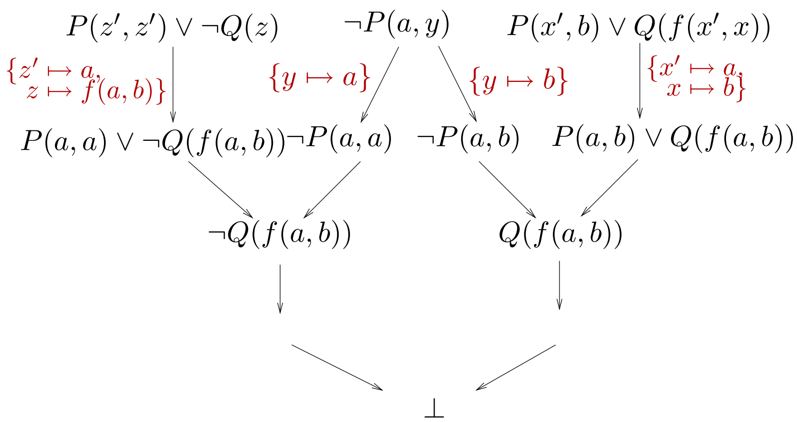
$$\mathcal{A} \models \forall x_1, \ldots, x_n F \quad \text{implies} \quad \mathcal{A} \models \forall y_1, \ldots, y_m F\sigma$$

In particular, if $\mathcal{A}$ is a model of an (implicitly universally quantified) clause C, then 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:



Problems:

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

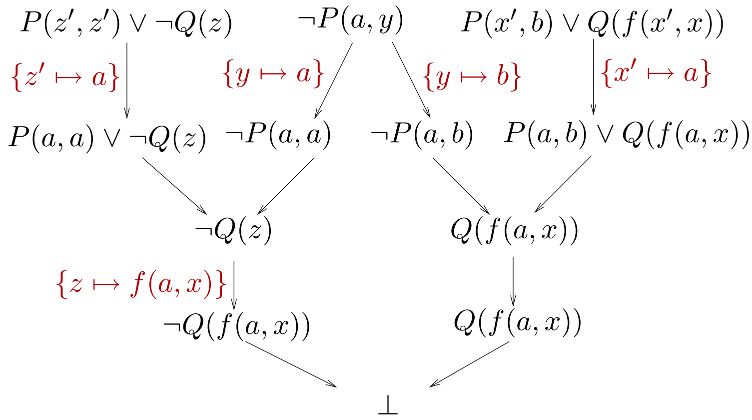
Even worse: There are infinitely many possible instances.

Observation:

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

Idea:

Do not instantiate more than necessary to get complementary literals.



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 (mgu).

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*:

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

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

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 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.*

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 $\text{mgu}(E)$.

Proposition 3.23

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

Rule-Based Naive Standard Unification

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

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 $\sigma_E = \{x_1 \mapsto u_1, \ldots, x_k \mapsto u_k\}$.

Proposition 3.25 *If E is a solved form then σ_E is an mgu of E.*

Theorem 3.26

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, $\sigma \circ \{x \mapsto t\} = \sigma[x \mapsto t\sigma] = \sigma[x \mapsto x\sigma] = \sigma$. Therefore, for any equation $u \doteq v$ in E: $u\sigma = v\sigma$, iff $u\{x \mapsto t\}\sigma = v\{x \mapsto t\}\sigma$. (2) and (3) follow by induction from (1) using Proposition 3.25. □

Main Unification Theorem

Theorem 3.27 *E is unifiable if and only if there is a most general unifier σ of E, such that σ is idempotent and $dom(\sigma) \cup codom(\sigma) \subseteq var(E)$.*

Proof.

- $\Rightarrow_{SU}$ is Noetherian. A suitable lexicographic ordering on the multisets E (with $\bot$ minimal) shows this. Compare in this order:
  (1) the number of variables that occur in E below a function or predicate symbol, or on the right-hand side of an equation, or at least twice;
  (2) the multiset of the sizes (numbers of symbols) of all equations in E;
  (3) the number of non-variable left-hand sides of equations in E.
- 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 3.26, Proposition 3.25).
- 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

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

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

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

Properties of PU

Theorem 3.28

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.*

Note: The solved form of $\Rightarrow_{PU}$ is different form the solved form obtained from $\Rightarrow_{SU}$. In order 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 $\{x_1 \mapsto t_1\} \circ \cdots \circ \{x_k \mapsto t_k\}$.

Lifting Lemma

Lemma 3.29 *Let C and D be variable-disjoint clauses. If*

$$\begin{array}{cc} D & C \\ \big\downarrow \sigma & \big\downarrow \rho \\ \multicolumn{2}{c}{\dfrac{D\sigma \qquad C\rho}{C'}} \end{array} \qquad \textit{[propositional resolution]}$$

then there exists a substitution τ such that

$$\begin{array}{c} \dfrac{D \qquad C}{C''} \\ \big\downarrow \tau \\ C' = C''\tau \end{array} \qquad \textit{[general resolution]}$$

An analogous lifting lemma holds for factorization.

Saturation of Sets of General Clauses

Corollary 3.30 *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,*

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

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 $D\sigma$ and $C\rho$ of N with resolvent C', or else (ii) C' is a factor of a ground instance $C\sigma$ of C.

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

Case (ii): Similar. □

Herbrand's Theorem

Lemma 3.31 *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 3.32 *Let N be a set of Σ-clauses, let $\mathcal{A}$ be a* Herbrand *interpretation. Then $\mathcal{A} \models G_\Sigma(N)$ implies $\mathcal{A} \models N$.*

Theorem 3.33 (Herbrand) *A set N of Σ-clauses is satisfiable if and only if it has a Herbrand model over Σ.*

Proof. The "⇐" part is trivial. For the "⇒" part let $N \not\models \bot$.

$$\begin{aligned} N \not\models \bot &\Rightarrow \bot \notin Res^*(N) \qquad \text{(resolution is sound)} \\ &\Rightarrow \bot \notin G_\Sigma(Res^*(N)) \\ &\Rightarrow I_{G_\Sigma(Res^*(N))} \models G_\Sigma(Res^*(N)) \qquad \text{(Thm. 3.17; Cor. 3.30)} \\ &\Rightarrow I_{G_\Sigma(Res^*(N))} \models Res^*(N) \qquad \text{(Lemma 3.32)} \\ &\Rightarrow I_{G_\Sigma(Res^*(N))} \models N \qquad (N \subseteq Res^*(N)) \quad \Box \end{aligned}$$

The Theorem of Löwenheim-Skolem

Theorem 3.34 (Löwenheim–Skolem) *Let Σ be a countable signature and let S be a set of closed Σ-formulas. 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 Theorem 3.33. □

Refutational Completeness of General Resolution

Theorem 3.35 *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 3.30: $Res(G_\Sigma(N)) \subseteq G_\Sigma(N)$

$$\begin{aligned} N \models \bot \Leftrightarrow G_\Sigma(N) \models \bot & \quad \text{(Lemma 3.31/3.32; Theorem 3.33)} \\ \Leftrightarrow \bot \in G_\Sigma(N) & \quad \text{(propositional resolution sound and complete)} \\ \Leftrightarrow \bot \in N & \quad \Box \end{aligned}$$

Compactness of Predicate Logic

Theorem 3.36 (Compactness Theorem for First-Order Logic) *Let S be a set of first-order formulas. S is unsatisfiable $\Leftrightarrow$ some finite subset $S' \subseteq S$ is unsatisfiable.*

Proof. The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let S be unsatisfiable and let N be the set of clauses obtained by Skolemization and CNF transformation of the formulas in S. Clearly $Res^*(N)$ is unsatisfiable. By Theorem 3.35, $\bot \in Res^*(N)$, and therefore $\bot \in Res^n(N)$ for some $n \in \mathbb{N}$. Consequently, $\bot$ has a finite resolution proof B of depth $\leq n$. Choose S' as the subset of formulas in S such that the corresponding clauses contain the assumptions (leaves) of B. $\Box$

3.12 Ordered Resolution with Selection

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

Ideas for improvement:

1. In the completeness proof (Model Existence Theorem 3.17) 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

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

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

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

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

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 $Res^{\succ}_{\text{sel}}$

The resolution calculus $Res^{\succ}_{\text{sel}}$ is parameterized by

- a selection function sel
- 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) $\sigma = \text{mgu}(A, B)$;

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

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

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

(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 sel.

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 sel;

(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 sel.

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 sel 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

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

we can obtain by *rotation*

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

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 $Res^{\succ}_{\text{sel}}$

Lemma 3.37 *Let D and C be variable-disjoint clauses. If*

$$\begin{array}{ccc} D & & C \\ \downarrow \sigma & & \downarrow \rho \\ \end{array}$$
$$\frac{D\sigma \qquad C\rho}{C'} \quad \textit{[propositional inference in } Res^{\succ}_{\text{sel}}\textit{]}$$

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

$$\begin{array}{c} D \qquad C \\ \hline C'' \\ \big\downarrow \tau \\ C' = C''\tau \end{array} \qquad \textit{[inference in } Res^{\succ}_{\text{sel}}\textit{]}$$

An analogous lifting lemma holds for factorization.

Saturation of General Clause Sets

Corollary 3.38 *Let* N *be a set of general clauses saturated under* $Res^{\succ}_{\text{sel}}$*, i. e.,* $Res^{\succ}_{\text{sel}}(N) \subseteq N$*. Then there exists a selection function* sel' *such that* $\text{sel}|_N = \text{sel}'|_N$ *and* $G_\Sigma(N)$ *is also saturated, i. e.,*

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

Proof. We first define the selection function sel' such that $\text{sel}'(C) = \text{sel}(C)$ 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 $\text{sel}'(C)$ to be those occurrences of literals that are ground instances of the occurrences selected by sel in D. Then proceed as in the proof of Cor. 3.30 using the above lifting lemma. □

Soundness and Refutational Completeness

Theorem 3.39 *Let* $\succ$ *be an atom ordering and* sel *a selection function such that* $Res^{\succ}_{\text{sel}}(N) \subseteq N$*. 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 3.38. □

Craig-Interpolation

A theoretical application of ordered resolution is Craig-Interpolation:

Theorem 3.40 (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. $Res^{\succ}_{\text{sel}}$ with an empty selection function sel . Then saturate $N^* \cup M$ w. r. t. $Res^{\succ}_{\text{sel}}$ 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 $C_1, \ldots, C_n \in N$, $n \geq 0$, such that $C_i \prec C$ and $C_1, \ldots, C_n \models C$.

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 $\succ$ is used for ordering restrictions and for redundancy (and for the completeness proof).

Examples of Redundancy

Proposition 3.41 *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. $Res^{\succ}_{\text{sel}}$) if

$$Res^{\succ}_{\text{sel}}(N \setminus Red(N)) \subseteq N \cup Red(N)$$

Theorem 3.42 *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 $I^{\succ}_{N}$ for $Res^{\succ}_{\text{sel}}$
- redundant clauses are not productive
- redundant clauses in N are not minimal counterexamples for $I^{\succ}_{N}$

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

(ii) Lifting: no additional problems over the proof of Theorem 3.39. □

Monotonicity Properties of Redundancy

Theorem 3.43

(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.