Resolution for General Clauses

We obtain the resolution inference rules for non-ground clauses from the inference rules for ground clauses by replacing equality by unifiabilty:

General resolution *Res*:

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

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

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.

Lifting Lemma

Lemma 3.30 *Let C and D be variable-disjoint clauses. If*

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

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

then there exists a substitution ρ such that

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

$$\downarrow \rho$$

$$C' = C''\rho$$

An analogous lifting lemma holds for factorization.

Saturation of Sets of General Clauses

Corollary 3.31 *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 $C' \in Res(G_\Sigma(N))$. Then either (i) there exist resolvable ground instances $D\theta_1$ and $C\theta_2$ of N with resolvent C', or else (ii) C' is a factor of a ground instance $C\theta$ of C.

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

Case (ii): Similar. $\square$

Soundness for General Clauses

Proposition 3.32 *The general resolution calculus is sound.*

Proof. We have to show that, if $\sigma = \mathrm{mgu}(A, B)$ then $\{\forall \vec{x}\ (D \vee B), \forall \vec{y}\ (C \vee \neg A)\} \models \forall \vec{z}\ (D \vee C)\sigma$ and $\{\forall \vec{x}\ (C \vee A \vee B)\} \models \forall \vec{z}\ (C \vee A)\sigma$.

Let $\mathcal{A}$ be a model of $\forall \vec{x}\ (D \vee B)$ and $\forall \vec{y}\ (C \vee \neg A)$. By Lemma 3.8, $\mathcal{A}$ is also a model of $\forall \vec{z}\ (D \vee B)\sigma$ and $\forall \vec{z}\ (C \vee \neg A)\sigma$ and by Lemma 3.7, $\mathcal{A}$ is also a model of $(D \vee B)\sigma$ and $(C \vee \neg A)\sigma$. Let β be an assignment. If $\mathcal{A}(\beta)(B\sigma) = 0$, then $\mathcal{A}(\beta)(D\sigma) = 1$. Otherwise $\mathcal{A}(\beta)(B\sigma) = \mathcal{A}(\beta)(A\sigma) = 1$, hence $\mathcal{A}(\beta)(\neg A\sigma) = 0$ and therefore $\mathcal{A}(\beta)(C\sigma) = 1$. In both cases $\mathcal{A}(\beta)((D \vee C)\sigma) = 1$, so $\mathcal{A} \models (D \vee C)\sigma$ and by Lemma 3.7, $\mathcal{A} \models \forall \vec{z}\ (D \vee C)\sigma$.

The proof for factorization inferences is similar. $\square$

Herbrand's Theorem

Lemma 3.33 *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 3.34 *Let N be a set of Σ-clauses, let $\mathcal{A}$ be a* Herbrand *interpretation. Then $\mathcal{A} \models G_\Sigma(N)$ implies $\mathcal{A} \models N$.*

Proof. Let $\mathcal{A}$ be a Herbrand model of $G_\Sigma(N)$. We have to show that $\mathcal{A} \models \forall\vec{x}\, C$ for all clauses $\forall\vec{x}\, C$ in N. This is equivalent to $\mathcal{A} \models C$, which in turn is equivalent to $\mathcal{A}(\beta)(C) = 1$ for all assignments β.

Choose $\beta : X \to U_{\mathcal{A}}$ arbitrarily. Since $\mathcal{A}$ is a Herbrand interpretation, $\beta(x)$ is a ground term for every variable x, so there is a substitution σ such that $x\sigma = \beta(x)$ for all variables x occurring in C. Now let γ be an arbitrary assignment, then for every variable occurring in C we have $(\gamma \circ \sigma)(x) = \mathcal{A}(\gamma)(x\sigma) = x\sigma = \beta(x)$ and consequently $\mathcal{A}(\beta)(C) = \mathcal{A}(\gamma \circ \sigma)(C) = \mathcal{A}(\gamma)(C\sigma)$. Since $C\sigma \in G_\Sigma(N)$ and $\mathcal{A}$ is a Herbrand model of $G_\Sigma(N)$, we get $\mathcal{A}(\gamma)(C\sigma) = 1$, so $\mathcal{A}$ is a model of C. $\square$

Theorem 3.35 (Herbrand) *A set N of Σ-clauses is satisfiable if and only if it has a Herbrand model over Σ.*

Proof. The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part let $N \not\models \bot$. Since resolution is sound, this implies that $\bot \notin \mathit{Res}^*(N)$. Obviously, a ground instance of a clause has the same number of literals as the clause itself, so we can conclude that $\bot \notin G_\Sigma(\mathit{Res}^*(N))$. Since $\mathit{Res}^*(N)$ is saturated, $G_\Sigma(\mathit{Res}^*(N))$ is saturated as well by Cor. 3.31. Now $I_{G_\Sigma(\mathit{Res}^*(N))}$ is a Herbrand interpretation over Σ and by Thm. 3.20 it is a model of $G_\Sigma(\mathit{Res}^*(N))$. By Lemma 3.34, every Herbrand model of $G_\Sigma(\mathit{Res}^*(N))$ is a model of $\mathit{Res}^*(N)$. Now $N \subseteq \mathit{Res}^*(N)$, so $I_{G_\Sigma(\mathit{Res}^*(N))} \models N$. $\square$

Corollary 3.36 *A set N of Σ-clauses is satisfiable if and only if its set of ground instances $G_\Sigma(N)$ is satisfiable.*

Proof. The "$\Rightarrow$" part follows directly from Lemma 3.33. For the "$\Leftarrow$" part assume that $G_\Sigma(N)$ is satisfiable. By Thm. 3.35 $G_\Sigma(N)$ has a Herbrand model. By Lemma 3.34, every Herbrand model of $G_\Sigma(N)$ is a model of N. $\square$

Refutational Completeness of General Resolution

Theorem 3.37 *Let N be a set of general clauses that is saturated w. r. t. Res. Then $N \models \bot$ if and only if $\bot \in N$.*

Proof. The "$\Leftarrow$" part is trivial. For the "$\Rightarrow$" part assume that N is saturated, that is, $Res(N) \subseteq N$. By Corollary 3.31, $G_\Sigma(N)$ is saturated as well, i. e., $Res(G_\Sigma(N)) \subseteq G_\Sigma(N)$. By Cor. 3.36, $N \models \bot$ implies $G_\Sigma(N) \models \bot$. By the refutational completeness of ground resolution, $G_\Sigma(N) \models \bot$ implies $\bot \in G_\Sigma(N)$, so $\bot \in N$. $\square$

3.12 Theoretical Consequences

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

The Theorem of Löwenheim-Skolem

Theorem 3.38 (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 3.35. $\square$

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 3.39 (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.*

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 3.37, $\bot \in Res^*(N)$, 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. $\square$

3.13 Ordered Resolution with Selection

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

Ideas for improvement:

1. In the completeness proof (Model Existence Theorem 3.20) 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*

Ordering Restrictions

In the completeness proof one only needs to resolve and factor maximal atoms
$\Rightarrow$ If we impose ordering restrictions on ground inferences, the proof remains correct:

(Ground) Ordered Resolution:

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

if $A \succ L$ for all L in D and $\neg A \succeq L$ for all L in C.

(Ground) Ordered Factorization:

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

if $A \succeq L$ for all L in C.

Problem: How to extend this to non-ground inferences?

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.

An ordering $\succ$ on atoms (or terms) is called *stable under substitutions*, if $A \succ B$ implies $A\sigma \succ B\sigma$.

Note:

- We can not require that $A \succ B$ iff $A\sigma \succ B\sigma$.
- We can not require that $\succ$ is total on non-ground atoms.

Consequence: In the ordering restrictions for non-ground inferences, we have to replace $\succ$ by $\not\preceq$ and $\succeq$ by $\not\prec$.

Ordered Resolution:

$$\frac{D \vee B \qquad C \vee \neg A}{(D \vee C)\sigma}$$

if $\sigma = \mathrm{mgu}(A, B)$ and $B\sigma \not\preceq L\sigma$ for all L in D and $\neg A\sigma \not\prec L\sigma$ for all L in C.

Ordered Factorization:

$$\frac{C \vee A \vee B}{(C \vee A)\sigma}$$

if $\sigma = \mathrm{mgu}(A, B)$ and $A\sigma \not\prec L\sigma$ for all L in C.

Selection Functions

Selection functions can be used to override ordering restrictions for individual clauses.

A *selection function* is a mapping

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

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

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

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

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

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

- a selection function sel
- and a well-founded ordering $\succ$ on atoms that is total on ground atoms and stable under substitutions.

(Ground) Ordered Resolution with Selection:

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

if the following conditions are satisfied:

(i) $A \succ L$ for all L in D;

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

(iii) $\neg A$ is selected in $C \vee \neg A$, or nothing is selected in $C \vee \neg A$ and $\neg A \succeq L$ for all L in C.

(Ground) Ordered Factorization with Selection:

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

if the following conditions are satisfied:

(i) $A \succeq L$ for all L in C;

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

The extension from ground inferences to non-ground inferences is analogous to ordered resolution (replace $\succ$ by $\not\preceq$ and $\succeq$ by $\not\prec$). Again we assume that $\succ$ is stable under substitutions.

Ordered Resolution with Selection:

$$\frac{D \lor B \qquad C \lor \neg A}{(D \lor C)\sigma}$$

if the following conditions are satisfied:

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

(ii) $B\sigma \not\preceq L\sigma$ for all L in D;

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

(iv) $\neg A$ is selected in $C \lor \neg A$, or nothing is selected in $C \lor \neg A$ and $\neg A\sigma \not\prec L\sigma$ for all L in C.

Ordered Factorization with Selection:

$$\frac{C \lor A \lor B}{(C \lor A)\sigma}$$

if the following conditions are satisfied:

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

(ii) $A\sigma \not\prec L\sigma$ for all L in C.

(iii) nothing is selected in $C \lor A \lor B$ by sel.

Lifting Lemma for $Res^{\succ}_{\mathrm{sel}}$

Lemma 3.40 *Let C and D be variable-disjoint clauses. If*

$$\begin{array}{ccc} D & & C \\ \big\downarrow \theta_1 & & \big\downarrow \theta_2 \\ \hline D\theta_1 & & C\theta_2 \\ \hline & C' & \end{array} \qquad \text{[ground inference in } Res^{\succ}_{\mathrm{sel}}\text{]}$$

and if $\mathrm{sel}(D\theta_1) \simeq \mathrm{sel}(D)$, $\mathrm{sel}(C\theta_2) \simeq \mathrm{sel}(C)$ *(that is, "corresponding" literals are selected), then there exists a substitution ρ such that*

$$\dfrac{D \qquad C}{C''} \qquad \text{[inference in } Res^{\succ}_{\mathrm{sel}}\text{]}$$

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

An analogous lifting lemma holds for factorization.

Saturation of Sets of General Clauses

Corollary 3.41 *Let N be a set of general clauses saturated under $Res^{\succ}_{\mathrm{sel}}$, i. e., $Res^{\succ}_{\mathrm{sel}}(N) \subseteq N$. Then there exists a selection function sel' such that $\mathrm{sel}|_N = \mathrm{sel}'|_N$ and $G_\Sigma(N)$ is also saturated, i. e.,*

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

Proof. We first define the selection function sel' such that $\mathrm{sel}'(C) = \mathrm{sel}(C)$ 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 $\mathrm{sel}'(C)$ 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. 3.31 using the lifting lemma above. $\square$

Soundness and Refutational Completeness

Theorem 3.42 *Let $\succ$ be an atom ordering and* sel *a selection function such that $Res^{\succ}_{\mathrm{sel}}(N) \subseteq N$. 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 if they are false in I_C and if their maximal atom occurs only once and is positive. The result for general clauses follows using Corollary 3.41. $\square$

What Do We Gain?

Search spaces become smaller:

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

we assume $P \succ Q$ and sel as indicated by $\boxed{X}$. The maximal literal in a clause is depicted in red.

In this example, the ordering and selection function even ensure that the refutation proceeds strictly deterministically.

Rotation redundancy can be avoided:

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 ordering restrictions.

Craig-Interpolation

Theorem 3.43 (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 propositional variables occurring both in F and in G, and such that $F \models H$ and $H \models G$.*

Proof. Let Π_F, Π_G, and Π_{FG} be the sets of propositional variables that occur only in F, only in G, or both in F and G. Translate F and $\neg G$ into CNF; let N and M, respectively, denote the resulting clause set. Choose an atom ordering $\succ$ for which the propositional variables in Π_F are larger than those in $\Pi_{FG} \cup \Pi_G$. Saturate N into N' w. r. t. $\mathit{Res}^{\succ}_{\mathrm{sel}}$ with an empty selection function sel. Then saturate $N' \cup M$ w. r. t. $\mathit{Res}^{\succ}_{\mathrm{sel}}$ 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 from Π_{FG}. The conjunction of these premises is an interpolant H. $\square$

The theorem also holds for first-order formulas, but in the general case, a proof based on resolution technology is complicated because of Skolemization.

3.14 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? (e. g., if they are tautologies)

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: If a ground clause C is redundant and all clauses smaller than C hold in I_C, then C holds in I_C (so C is neither a minimal counterexample nor productive).

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

Examples of Redundancy

In general, redundancy is undecidable. Decidable approximations are sufficient for us, however.

Proposition 3.44 *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. $N \cup \{C\}$.*

(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. $Res^{\succ}_{\mathrm{sel}}$) if

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

Theorem 3.45 *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 $Res^{\succ}_{\mathrm{sel}}$.

If a clause $C \in N$ is redundant, then there exist $C_1, \ldots, C_n \in N$, $n \geq 0$, such that $C_i \prec C$ and $C_1, \ldots, C_n \models C$.

If $I_C \models C_i$ by minimality, then $I_C \models C$.

In particular, C is not productive.

$\Rightarrow$ Redundant clauses are not used as premises for "essential" inferences.

By saturation, the conclusion $D' \vee C'$ of a resolution inference is contained in N (as before) or in $Red(N)$. In the first case, minimality of C ensures that $D' \vee C'$ is productive or $I_{D' \vee C'} \models D' \vee C'$; in the second case, it ensures that $I_{D' \vee C'} \models D' \vee C'$. So in both cases we get a contradiction (analogously for factorization). The rest of the proof works as before.

(ii) Lifting: no additional problems over the proof of Theorem 3.42. $\square$

Monotonicity Properties of Redundancy

When we want to delete redundant clauses during a derivation, we have to ensure that redundant clauses *remain redundant* in the rest of the derivation.

Theorem 3.46

(i) $N \subseteq M \Rightarrow \mathit{Red}(N) \subseteq \mathit{Red}(M)$

(ii) $M \subseteq \mathit{Red}(N) \Rightarrow \mathit{Red}(N) \subseteq \mathit{Red}(N \setminus M)$

Proof. (i) Obvious.

(ii) For ground clause sets N, the well-foundedness of the multiset extension of the clause ordering implies that every clause in $\mathit{Red}(N)$ is entailed by smaller clauses in N that are themselves not in $\mathit{Red}(N)$.

For general clause sets N, the result follows from the fact that every clause in $G_\Sigma(N) \setminus \mathit{Red}(G_\Sigma(N))$ is an instance of a clause in $N \setminus \mathit{Red}(N)$. □

Recall that $\mathit{Red}(N)$ may include clauses that are not in N.

Computing Saturated Sets

Redundancy is preserved when, during a theorem proving derivation one adds new clauses or deletes redundant clauses. This motivates the following definitions:

A *run* of the resolution calculus is a sequence $N_0 \vdash N_1 \vdash N_2 \vdash \dots$, such that
(i) $N_i \models N_{i+1}$, and
(ii) all clauses in $N_i \setminus N_{i+1}$ are redundant w. r. t. N_{i+1}.

In other words, during a run we may add a new clause if it follows from the old ones, and we may delete a clause, if it is redundant w. r. t. the remaining ones.

For a run, we define $N_\infty = \bigcup_{i \geq 0} N_i$ and $N_* = \bigcup_{i \geq 0} \bigcap_{j \geq i} N_j$. The set N_* of all *persistent* clauses is called the *limit* of the run.

Lemma 3.47 *Let* $N_0 \vdash N_1 \vdash N_2 \vdash \dots$ *be a run. Then* $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_\infty)$ *and* $\mathit{Red}(N_i) \subseteq \mathit{Red}(N_*)$ *for every* i.

Proof. Exercise. □

Corollary 3.48 $N_i \subseteq N_* \cup \mathit{Red}(N_*)$ *for every* i.

Proof. If $C \in N_i \setminus N_*$, then there is a $k \geq i$ such that $C \in N_k \setminus N_{k+1}$, so C must be redundant w. r. t. N_{k+1}. Consequently, C is redundant w. r. t. N_*. □

Even if a set N is inconsistent, it could happen that $\bot$ is never derived, because some required inference is never computed.

The following definition rules out such runs:

A run is called *fair,* if the conclusion of every inference from clauses in $N_* \setminus \mathit{Red}(N_*)$ is contained in some $N_i \cup \mathit{Red}(N_i)$.

Lemma 3.49 *If a run is fair, then its limit is saturated up to redundancy.*

Proof. If the run is fair, then the conclusion of every inference from non-redundant clauses in N_* is contained in some $N_i \cup \mathit{Red}(N_i)$, and therefore contained in $N_* \cup \mathit{Red}(N_*)$. Hence N_* is saturated up to redundancy. □

Theorem 3.50 (Refutational Completeness: Dynamic View) *Let* $N_0 \vdash N_1 \vdash N_2 \vdash \dots$ *be a fair run, let* N_* *be its limit. Then* N_0 *has a model if and only if* $\bot \notin N_*$.

Proof. ($\Leftarrow$): By fairness, N_* is saturated up to redundancy. If $\bot \notin N_*$, then it has a Herbrand model. Since every clause in N_0 is contained in N_* or redundant w. r. t. N_*, this model is also a model of $G_\Sigma(N_0)$ and therefore a model of N_0.

($\Rightarrow$): Obvious, since $N_0 \models N_*$. □

Simplifications

In theory, the definition of a run permits to add arbitrary clauses that are entailed by the current ones.

In practice, we restrict to two cases:

- We add conclusions of $\mathit{Res}^{\succ}_{\mathrm{sel}}$-inferences from non-redundant premises.
  $\leadsto$ necessary to guarantee fairness
- We add clauses that are entailed by the current ones if this *makes* other clauses redundant:

  $$N \cup \{C\} \;\vdash\; N \cup \{C, D\} \;\vdash\; N \cup \{D\}$$
  $$\text{if } N \cup \{C\} \models D \text{ and } C \in \mathit{Red}(N \cup \{D\}).$$

  Net effect: C is *simplified* to D
  $\leadsto$ useful to get easier/smaller clause sets

Examples of simplification techniques:

- Deletion of duplicated literals:

$$N \cup \{C \vee L \vee L\} \ \vdash \ N \cup \{C \vee L\}$$

- Subsumption resolution:

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

3.15 Hyperresolution

There are *many* variants of resolution.

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 $Res^{\succ}_{\text{sel}}$, the calculus is parameterized by an atom ordering $\succ$ and a selection function sel.

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

with $\sigma = \text{mgu}(A_1 \doteq B_1, \dots, A_n \doteq B_n)$, 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 sel, or 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 factorization 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.

3.16 Implementing Resolution: The Main Loop

Standard approach:

Select one clause ("Given clause").

Find many partner clauses that can be used in inferences together with the "given clause" using an appropriate index data structure.

Compute the conclusions of these inferences; add them to the set of clauses.

The set of clauses is split into two subsets:

- WO = "Worked-off" (or "active") clauses: Have already been selected as "given clause".
- U = "Usable" (or "passive") clauses: Have not yet been selected as "given clause".

During each iteration of the main loop:

Select a new given clause C from U;
$U := U \setminus \{C\}$.

Find partner clauses D_i from WO;
$New :=$ Conclusions of inferences from $\{ D_i \mid i \in I \} \cup C$ where one premise is C;
$U := U \cup New$;
$WO := WO \cup \{C\}$

$\Rightarrow$ At any time, all inferences between clauses in WO have been computed.

$\Rightarrow$ The procedure is fair, if no clause remains in U forever.

Additionally:

Try to simplify C using WO. (Skip the remainder of the iteration, if C can be eliminated.)

Try to simplify (or even eliminate) clauses from WO using C.

Design decision: should one also simplify U using C?

yes $\rightsquigarrow$ "Otter loop":
Advantage: simplifications of U may be useful to derive the empty clause.

no $\rightsquigarrow$ "Discount loop":
Advantage: clauses in U are really passive; only clauses in WO have to be kept in index data structure. (Hence: can use index data structure for which retrieval is faster, even if update is slower and space consumption is higher.)