Restrictions of resolution

Restrictions allow to perform a resolution step only when the clauses involved satisfy certain syntactic conditions.

A restriction is complete if the calculus with the restriction is still complete.

We consider some restrictions of propositional resolution.
Extending them to predicate logic is easy.

Positive and negative resolution

P-resolution: one of the two clauses to be resolved is positive, i.e., contains only positive literals.

N-resolution: one of the two clauses to be resolved is negative, i.e., contains only negative literals.

Theorem: P- and N-resolution are complete.

Completeness proof

Theorem: P- and N-resolution are complete.

Proof: Only for P-resolution (N-resolution similar).
Let F be an unsatisfiable formula. We show that the empty clause can be derived using P-resolution.
By induction on the number n of atomic formulas occurring in F.
Case $n = 0$ is trivial. Let $n > 0$ and let A be an atomic formula of F.
Example:

$$F = \{ \{A, \neg D\} \quad \{A, B, D\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\} \}$$

We know that $F[A/0]$ and $F[A/1]$ are unsatisfiable.

$$F[A/0] \quad = \quad \{\neg D\} \quad \{B, D\} \quad \{\neg B, D\}$$
$$F[A/1] \quad = \quad \{\neg B, \neg D\} \quad \{B\} \quad \{\neg B, D\}$$

Completeness proof

(1) Construct using P-Resolution a derivation of the empty clause from $F[A/0]$ (exists by induction hypothesis).

$$F[A/0]: \quad \{\neg D\} \qquad \{B, D\} \qquad \{\neg B, D\}$$

Completeness proof

(2) Transform the derivation from step (1) into a derivation of $\{A\}$ from F (or of the empty clause if this is possible).

(3) Add resolution steps that resolve $\{A\}$ with every clause of F containing $\neg A$.

$$F: \quad \{A, \neg D\} \quad \{A, B, D\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

This produces the clauses in $F[A/1]$.

Completeness proof

Add a derivation of the empty clause from $F[A/1]$.

$$F[A/1]: \quad \{\neg B, \neg D\} \quad \{B\} \quad \{\neg B, D\}$$

Linear resolution

Linear resolution: one of the two clauses must be the resolvent produced in the previous step (no restriction for the first step).

Theorem: Linear resolution is complete.

Completeness proof

Theorem: Linear resolution is complete.

Proof: Let F be unsatisfiable.

$$F = \{A\} \quad \{A, B, D\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

Let $F' \subseteq F$ be a minimal unsatisfiable subset (unsatisfiable core)

$$F' = \{A\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

We show: for every clause C of F' there is a linear derivation of the empty clause starting with C.

Proof by induction on the number n of atomic formulas.
Case $n = 0$ is trivial. Let $n > 0$ and let A be an atomic formula of F.
We consider two cases: $|C| = 1$ and $|C| > 1$.

Case $|C| = 1$

Let $C = \{L\}$.

$$C = \{A\}$$

We know that $F'[A/0]$ and $F'[A/1]$ are unsatisfiable.

Step 1: Choose an unsatisfiable core F'' of $F'[L/1]$.

$$F'' = F'[A/1] = \quad \{\neg B, \neg D\} \quad \{B\} \quad \{\neg B, D\}$$

Pick $C' \in F''$ such that $C' \cup \{\overline{L}\} \in F'$.
(C' exists, otherwise $F'' \subseteq F' - \{C\}$ and so by minimality of F' the core F'' is satisfiable.)

$$C' = \{\neg B, \neg D\}$$

Case $|C| = 1$ (con.)

Step 2: Construct a linear derivation of the empty clause from F'' starting with C' (exists by induction hypothesis).

$$F'' : \quad \{\neg B, \neg D\} \quad \{B\} \quad \{D\}$$

Case $|C| = 1$ (con.)

Step 3: Resolve $\{L\}$ with $C' \cup \{\overline{L}\}$, add the derivation from Step 2 to get a derivation of $\{\overline{L}\}$ from F', and resolve $\{L\}$ and $\{\overline{L}\}$.

$$F' : \quad \{A\} \quad \{\neg A \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

Case $|C| > 1$

$$\begin{aligned} F &= \{A\} \quad \{A, B, D\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\} \\ F' &= \{A\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\} \\ C &= \{\neg A, \neg B, \neg D\} \end{aligned}$$

Step 1: Pick any $L \in C$ and set $C' = C - \{L\}$.

$$L = \neg B \qquad C' = \{\neg A, \neg D\}$$

Choose an unsatisfiable core F'' of $F'[L/0]$ containing C'.
(Why must it exist?)

$$F'' = F'[\neg B/0] = F'[B/1] = \quad \{A\} \quad \{\neg A, \neg D\} \quad \{D\}$$

Case $|C| > 1$ (con.)

Step 2: Construct a linear derivation of the empty clause from F'' starting with C' (exists by induction hypothesis). Transform it into a derivation of $\{L\}$ from F'.

$$F' : \quad \{A\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

Case $|C| > 1$ (con.)

Step 3: Apply the previous case (singleton clause case) to $(F' - \{C\}) \cup \{\{L\}\}$.
(Allowed, because $(F' - \{C\}) \cup \{\{L\}\}$ unsatisfiable and $(F' - \{C\})$ satisfiable.)

$$(F' - \{C\}) \cup \{\{L\}\} : \quad \{A\} \quad \{\neg A, B\} \quad \{\neg B, D\} \quad \{\neg B\}$$

Case $|C| > 1$ (con.)

Step 4: Concatenate the derivations from steps 2 and 3.

$$F' : \quad \{A\} \quad \{\neg A, \neg B, \neg D\} \quad \{\neg A, B\} \quad \{\neg B, D\}$$

SLD-Resolution

The satisfiability problem for Horn-formulas can be solved in linear time.

The satisfiability problem for Horn-formulas of predicate logic is, however, undecidable.

SLD-resolution is defined only for Horn-formulas.

SLD-resolution: linear resolution +

- start at a negative clause (the goal clause);
- at each resolution step one of the parent clauses is an input non-negative clause (a procedure clause).

Completeness

Theorem: SLD-resolution is complete (for Horn-formulas).

Proof: Let F be an unsatisfiable Horn-formula.

(1) F contains a negative clause C.
Proof: exercise.

(2) There is a linear derivation of the empty clause starting with C.
Already proved (Why?).

(3) At each step of this derivation one of the two clauses to be resolved is an input procedure clause.
Proof: by the Horn condition all resolvents of the derivation are negative. Since negative clauses can only be resolved with non-negative clauses, the other clause must be a procedure clause, which must come from the input.