3.5 Restrictions of Resolution

Freitag, 8. Mai 2015 10:00

4 restrictions of resolution:

- linear resolution (still complete)
- input resolution (no longer complete, but still complete on Horn clauses, i.e., on the clauses used in logic programs)
- SLD resolution (complete on Horn clauses)
- binary SLD resolution (— " ——)

This is the form of resolution used in logic programming.

3.5.1. Linear Resolution

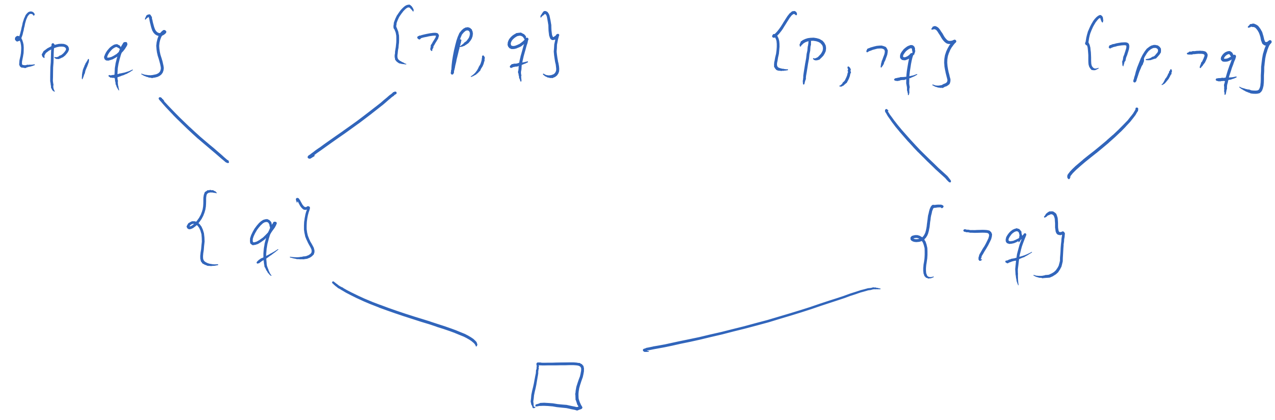
Restrict resolution in the following way: One of the parent clauses in the next resolution step must be the resolvent that was produced in the step before.

Def 3.5.1 (Linear Resolution)

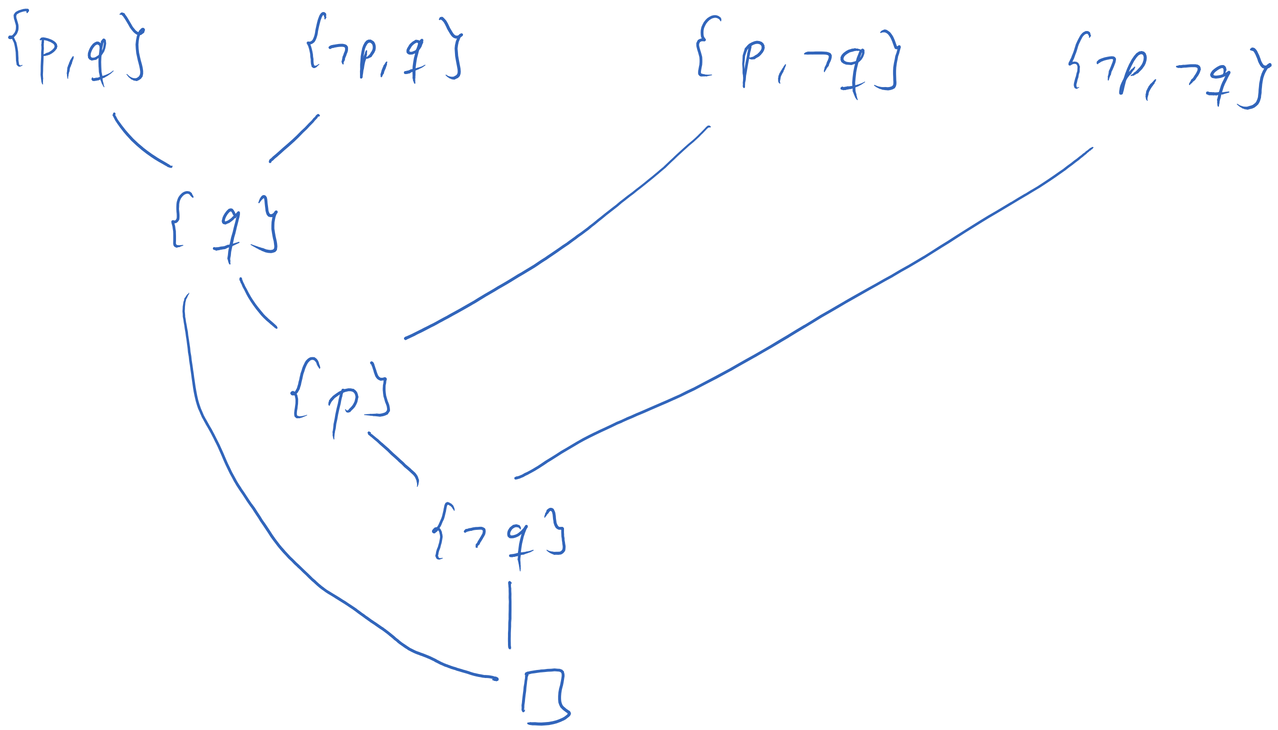
Let $\mathcal{K}$ be a clause set. $\square$ can be obtained from $K \in \mathcal{K}$ by linear resolution iff there is a sequence $K_1, \dots, K_m$ such that

- $K_1 = K \in \mathcal{K}$
- $K_m = \square$
- for all $2 \leq i \leq m$: K_i is a resolvent of K_{i-1} and a clause from $\{K_1, \dots, K_{i-1}\} \cup \mathcal{K}$

Ex 352



This is not a linear resolution proof.
Is linear resolution still complete?
Can we also derive □ by linear resolution?



Thm 353 (Soundness + Completeness of Linear Resolution)
Let $\mathcal{K}$ be a clause set. Then

$\mathcal{K}$ is unsatisfiable iff □ can be derived by linear resolution from some $K \in \mathcal{K}$.

If $\mathcal{K}$ is a minimal unsatisfiable clause set

(i.e., for every $K \in \mathcal{K}$ the set $\mathcal{K} \setminus \{K\}$ is satisfiable),
then $\Box$ can even be derived by linear resolution from <u>every</u> $K \in \mathcal{K}$.

<u>Proof</u>: "$\Leftarrow$" (Soundness): obvious, because every linear resolution proof is a proper resolution proof (and resolution is sound by Thm 3.4.10)

"$\Rightarrow$" (completeness):

- prove completeness of propositional linear resolution
- Then use the lifting lemma to lift any linear resolution proof of $\Box$ to a linear resolution proof in pred. logic.

$\boxtimes$

3.5.2. Input- and SLD-Resolution

Input Resolution: Special form of linear resolution (i.e., one parent clause must be the resolvent obtained in the last step). But now the other parent clause must be from the original clause set (i.e., it must not be a resolvent from an earlier step).

<u>Def 3.5.4</u> (Input Resolution)

Let $\mathcal{K}$ be a clause set. $\Box$ can be derived from a clause $K \in \mathcal{K}$ by <u>input resolution</u> iff there is a sequence of clauses

$K_1, \dots, K_m$ with
- $K_1 = K \in \mathcal{K}$
- $K_m = \square$
- for all $2 \leq i \leq m$:
  K_i is a resolvent of K_{i-1} and a clause from $\mathcal{K}$.

Advantage: drastic reduction of search space for the next resolution step ($\mathcal{K}$ remains constant, i.e., no new clauses are added to $\mathcal{K}$)

Disadvantage: input resolution is no longer complete

Ex 355 Consider the clauses from Ex 352:

$\{p, q\}$ $\{\neg p, q\}$ $\{p, \neg q\}$ $\{\neg p, \neg q\}$

By input resolution we can deduce in the first step:

$\{q\}$, $\{\neg q\}$, $\{p\}$, $\{\neg p\}$, $\{p, \neg p\}$, or $\{q, \neg q\}$

In the second step, we obtain a clause from $\{q\}$, $\{\neg q\}$, $\{p\}$, $\{\neg p\}$ or from the input set.

$\Rightarrow$ By input resolution one can only deduce
- unit clauses ($\{q\}, \{\neg q\}, \{p\}, \{\neg p\}$)
- clauses from the input set
- tautologies ($\{p, \neg p\}, \{q, \neg q\}$)

One never reaches $\square$.

While input resolution is not complete in general (not even in propositional logic), it is complete on Horn clauses (In LP, we only regard Horn clauses.)

<u>Def 3.56</u> (Horn clause)

A clause K is a <u>Horn clause</u> iff it contains at most one positive literal (all other literals must be negated atomic formulas).

A Horn clause is called <u>negative</u> iff it only contains negated literals (i.e., it has the form $\{\neg A_1, \dots, \neg A_k\}$ for atomic formulas $A_1, \dots, A_k$)).

A Horn clause is called <u>definite</u> iff it contains one positive literal (i.e., it has the form $\{B, \neg C_1, \dots, \neg C_n\}$ for atomic formulas $B, C_1, \dots, C_n$).

A set of definite Horn clauses corresponds to a conjunction of implications:

$$\{\ \{p, \neg q\},\ \{\neg r, \neg p, s\},\ \{s\}\ \}$$

is equivalent to

$$(p \vee \neg q) \wedge (\neg r \vee \neg p \vee s) \wedge s$$

which is equivalent to

$$(q \to p) \land (r \land p \to s) \land s.$$

This corresponds to the following logic program:

s.
s :- r, p.
p :- q.

← Restriction to Horn clauses means that we may not have negations in bodies of rules (p :- ¬q would correspond to $\{p, q\}$).

⇒ definite Horn clauses correspond to clauses of a logic program

- facts ≙ definite Horn clauses without negative literals (e.g. $\{s\}$)
- rules ≙ definite Horn clause with negative literals (e.g., $\{s, \neg r, \neg p\}$)
- queries ≙ negative Horn clause (e.g., $\{\neg p, \neg q\}$).

  ?- p, q.

  The negation of $p \land q$ would be added to the program clauses in order to prove unsatisfiability.

$\neg p \lor \neg q$, i.e., $\{\neg p, \neg q\}$

Restriction to Horn clauses improves efficiency substantially:

- input resolution instead of just linear resolution

(but unsatisfiability of Horn clauses remains undecidable in predicate logic)

- in propositional logic
  - (un)satisfiability of clauses is decidable, but NP-complete
  - (un)satisfiability of prop. Horn clauses can be checked in polynomial time

Instead of proving completeness of input resolution on Horn clauses, we restrict input resolution to SLD-resolution and then prove its completeness on Horn clauses.

Def 357 (SLD-Resolution)

Let $\mathcal{K}$ be a set of Horn clauses with $\mathcal{K} = \mathcal{K}^d \uplus \mathcal{K}^n$, where $\mathcal{K}^d$ contains the definite and $\mathcal{K}^n$ contains the negative clauses of $\mathcal{K}$. $\Box$ can be derived from $K \in \mathcal{K}^n$ by SLD-resolution iff there is a sequence $K_1, \dots, K_m$ with

- $K_1 = K \in \mathcal{K}^n$
- $K_m = \Box$
- for each $2 \le i \le m$: K_i is a resolvent of K_{i-1} and a clause from $\mathcal{K}^d$.

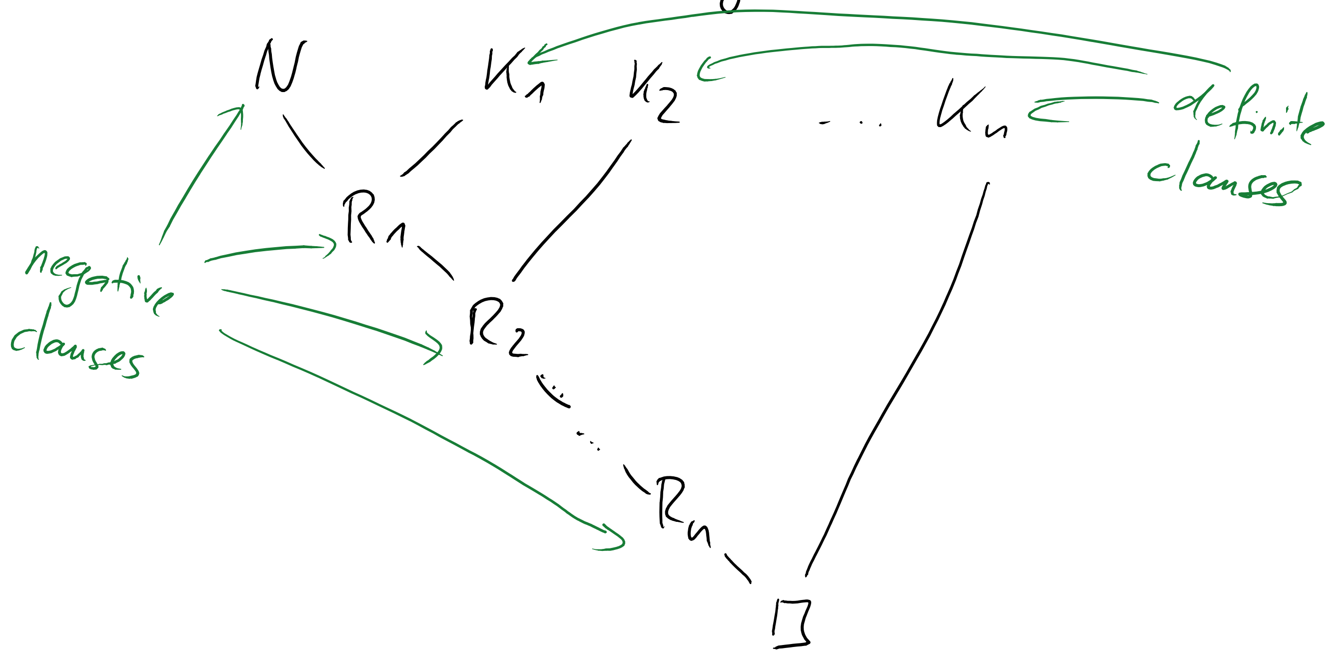
Difference to input resolution:

- Start with negative clause K_1 (i.e., no resolution between 2 definite clauses)

$\Rightarrow$ negative clauses can only be resolved with definite clauses

$\Rightarrow$ K_2 is again a negative clause

$\Rightarrow \dots \Rightarrow$ all $K_1, \dots, K_n$ are negative clauses



"SLD-resolution" stands for

<u>l</u>inear resolution with <u>s</u>election function for <u>d</u>efinite clauses

Selection function needs to solve the remaining 2 indeterminisms:

1. Which program clause should be used in the next resolution step?
2. Which literal in the negative clause should be used for the next resolution step?

<u>Thm 358</u> (Soundness + Completeness of SLD-Resolution)

Let $\mathcal{K}$ be a set of Horn clauses. Then:

$\mathcal{K}$ is unsatisfiable iff $\square$ can be derived by SLD-resolution from some negative clause $N \in \mathcal{K}$.

Proof: "$\Leftarrow$" (Soundness) obvious, since SLD-resolution is a restriction of full resolution

"$\Rightarrow$" (Completeness):

Let $\mathcal{K}_{min} \subseteq \mathcal{K}$ be a minimal unsatisfiable subset of $\mathcal{K}$. $\mathcal{K}_{min}$ must contain a negative clause N, since any set of definite Horn clauses is satisfiable (the interpretation that satisfies all atomic formulas would be a model).

By Thm 3.5.3, $\square$ can be deduced by linear resolution from any clause in $\mathcal{K}_{min}$.

$\Rightarrow$ There is a linear resolution proof of $\square$ that starts with the negative clause $N \in \mathcal{K}_{min}$.

Any such linear resolution proof is also an SLD-resolution proof (since negative clauses can only be resolved with definite clauses and the resolvent is again a negative clause).

$\blacksquare$

Resolution Algorithm can now be improved by starting with a negative clause and by only performing SLD-resolution.

In logic programming, a resolution step only removes <u>one</u> literal in each parent clause (<u>binary</u> resolution)

<u>Binary resolution</u>: like ordinary resolution, but with $m = n = 1$. (i.e., in $\nu_1(K_1)$ one removes just L_1 and in $\nu_2(K_2)$ one removes just L_1').

In general, binary resolution is not complete.

<u>Ex. 3.59</u>

$$\{\overset{L_1}{p(X)}, \overset{L_2}{p(Y)}\} \qquad \{\overset{L_1'}{\neg p(U)}, \overset{L_2'}{\neg p(V)}\}$$

$$\sigma$$

$$\square$$

$\sigma = mgu\ \{\neg p(X), \neg p(Y), \neg p(U), \neg p(V)\} = \{X/V, Y/V, U/V\}$

This was not a binary resolution step.

$$\{\underline{p(X)}, p(Y)\} \qquad \{\underline{\neg p(U)}, \neg p(V)\}$$

not a Horn clause

$\{p(Y), \neg p(V)\}$

...

$\square$ can't be derived with binary resolution.

But: binary resolution is complete for Horn clauses

Thm 3.5.10. (Soundness + Completeness of Binary SLD-Resolution)

Let $\mathcal{K}$ be a set of Horn clauses. Then:

$\mathcal{K}$ is unsatisfiable iff $\square$ can be deduced from a negative clause $N \in \mathcal{K}$ by binary SLD-resolution.