

(2) For every N_i in the sequence, $0 \leq i < p$, if

$$N_i =: -A_1, \dots, A_{k-1}, A_k, A_{k+1}, \dots, A_n,$$

then there is some definite clause $C_i = A : -B_1, \dots, B_m$ in D such that A_k and A are unifiable, and for some most general unifier σ_i of A_k and $\rho_i(A)$, where (Id, ρ_i) is a separating substitution pair, if $m > 0$, then

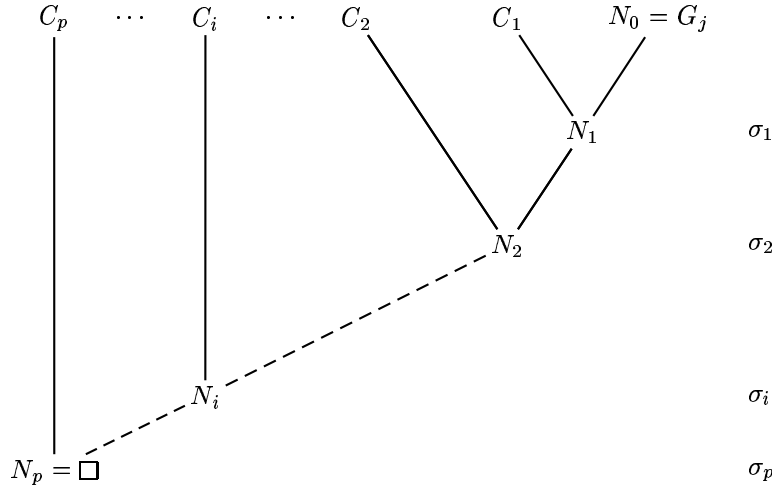
$$N_{i+1} =: -\sigma_i(A_1, \dots, A_{k-1}, \rho_i(B_1), \dots, \rho_i(B_m), A_{k+1}, \dots, A_n)$$

else if $m = 0$ then

$$N_{i+1} =: -\sigma_i(A_1, \dots, A_{k-1}, A_{k+1}, \dots, A_n).$$

An SLD-derivation is an *SLD-refutation* iff $N_p = \square$.

Note that an SLD-derivation is a linear representation of a resolution DAG of the following special form:



At each step, the clauses

$$: -A_1, \dots, A_{k-1}, A_k, A_{k+1}, \dots, A_n$$

and

$$A : -B_1, \dots, B_m$$

are resolved, the atoms A_k and $\rho_i(A)$ being canceled, since they are unified by the most general unifier σ_i . The literal A_k is called the *selected atom* of N_i , and the clauses N_0, C_1, \dots, C_p are the *input clauses*.

When the derivation is a refutation, the substitution

$$\sigma = (\rho_1 \circ \sigma_1) \circ \dots \circ (\rho_p \circ \sigma_p)$$

obtained by composing the substitutions occurring in the refutation is called the *result substitution* or *answer substitution*. It is used in PROLOG to extract the output of an SLD-computation.

Since an SLD-derivation is a special kind of resolution DAG, (a linear input resolution), its soundness is a consequence of lemma 8.5.2.

Lemma 9.4.1 (Soundness of SLD-resolution) If a set of Horn clauses has an SLD-refutation, then it is unsatisfiable.

Proof: Immediate from lemma 8.5.2. \square

Let us give an example of an SLD-refutation in the first-order case.

EXAMPLE 9.4.1

Consider the following set of definite clauses, axiomatizing addition of natural numbers:

$$\begin{aligned} C_1 &: \text{add}(X, 0, X). \\ C_2 &: \text{add}(X, \text{succ}(Y), \text{succ}(Z)) : \neg \text{add}(X, Y, Z). \end{aligned}$$

Consider the goal

$$B : \neg \text{add}(\text{succ}(0), V, \text{succ}(\text{succ}(0))).$$

We wish to show that the above set is unsatisfiable. We have the following SLD-refutation:

Goal clause	Input clause	Substitution
$: \neg \text{add}(\text{succ}(0), V, \text{succ}(\text{succ}(0)))$	C_2	
$: \neg \text{add}(\text{succ}(0), Y_2, \text{succ}(0))$	C_1	σ_1
\square		σ_2

where

$$\begin{aligned} \sigma_1 &= (\text{succ}(0)/X_1, \text{succ}(0)/Z_1, \text{succ}(Y_2)/V), \\ \sigma_2 &= (\text{succ}(0)/X_2, 0/Y_2) \end{aligned}$$

The variables X_1, Z_1, Y_2, X_2 were introduced by separating substitutions in computing resolvents. The result substitution is

$$(\text{succ}(0)/V, \text{succ}(0)/X_1, \text{succ}(0)/Z_1, \text{succ}(0)/X_2).$$

The interesting component is $\text{succ}(0)/V$. Indeed, there is a computational interpretation of the unsatisfiability of the set $\{C_1, C_2, B\}$. For this, it is necessary to write quantifiers explicitly and remember that goal clauses are negative. Observe that

$$\forall X C_1 \wedge \forall X \forall Y \forall Z C_2 \wedge \forall V B$$