# Logic Programming 2013–14
# Tutorial 5: Definite Clause Predicate Logic

## For discussion during Week 7 (Oct. 28 – Nov. 1)

1. Consider the following Prolog program and query.

```
n(e,_).
n(f(X),g(Y)) :- n(X,Y).
n(g(X),f(Y)) :- n(X,Y).

?- n(X,f(g(e))), n(X,f(f(e))).
```

   (a) Write out this program and query in the notation of first-order logic, making all quantification explicit.

   (b) Draw the full search tree generated by the program and query.

2. Consider the three substitutions displayed below.

$$\{\, \mathtt{X} = \mathtt{f}(\mathtt{Y},\mathtt{Y}) \,\} \quad \{\, \mathtt{X} = \mathtt{f}(\mathtt{g}(\mathtt{Z}),\mathtt{W}) \,\} \quad \{\, \mathtt{X} = \mathtt{f}(\mathtt{g}(\mathtt{Z}),\mathtt{g}(\mathtt{Z})) \,\}$$

   For each pair of substitutions from the above set, say whether one of the substitutions is more general than the other, and justify your answer.

3. For each of the following pairs of terms, find two different unifiers $\theta_1$ and $\theta_2$ such that $\theta_1$ is a most-general unifier, and such that $\theta_2$ is not most general.

   (a) `f(X,h(Z))` and `f(g(W),U)`

   (b) `f(X,h(Z))` and `f(g(W),h(W))`

   In what sense are the most general unifiers unique?

4.* This question concerns the *resolution* proof method, introduced by J. A. Robinson; and *SLD resolution* introduced by R. Kowalski.

   In predicate logic a *literal* is a formula that is either an atomic formula or the negation of an atomic formula. That is, a literal is either of the form $A$ or of the form $\neg A$, where $A$ is an atomic formula. A literal of the form $A$ is said to be a *positive* literal; one of the form $\neg A$ is said to be *negative*.

   A *clause* is a formula that is a finite disjuction

$$L_1 \vee \cdots \vee L_k$$

   of literals $L_1, \ldots L_k$, where $k \geq 0$. We write the $k = 0$ clause as $\perp$; it can be thought of as representing the truth value **false**.

The *resolution proof rule* combines two clauses as follows:

$$\frac{L_1 \vee \cdots \vee L_k \qquad\qquad L'_1 \vee \cdots \vee L'_n}{(L_1 \vee \cdots \vee L_{l-1} \vee L_{l+1} \vee \cdots \vee L_k \vee L'_1 \vee \cdots \vee L'_{m-1} \vee L'_{m+1} \vee \cdots \vee L'_n)\theta} \tag{1}$$

where $1 \leq l \leq k,\ 1 \leq m \leq n,\ L_l$ is an atomic formula $A,\ L'_m$ is a negated atomic formula $\neg B$, and $\theta$ is the most general unifier of $A$ and $B$.

A *resolution refutation* for a finite set of clauses $C_1, \ldots, C_N$, where $N \geq 1$, is a tree of applications of the proof rule, for which the leaves are (variable renamings of) clauses in $C_1, \ldots, C_N$, and the root is the empty clause $\perp$.

Resolution refutations are sound and complete in the sense that the following statements are equivalent.

(i) $C_1, \ldots, C_N$ has a resolution refutation.

(ii) $\forall \mathit{Vars}(C_1).\, C_1,\ \ldots\ ,\ \forall \mathit{Vars}(C_N).\, C_N \models \mathbf{false}$.
Statement (ii) is equivalently expressed as saying that the formula below is *unsatisfiable* (i.e., it has no models).

$$\forall \mathit{Vars}(C_1).\, C_1 \ \wedge\ \ldots\ \wedge\ \forall \mathit{Vars}(C_N).\, C_N$$

SLD resolution is a special case of resolution. A *definite clause* is a clause that has exactly one positive literal. A *goal clause* is a clause that has no positive literals. The notion of an *SLD-resolution refutation* is defined for a set of clauses:

$$G,\ D_1,\ \ldots\ ,D_N$$

where $G$ is a single goal clause and $D_1, \ldots, D_N$ are definite clauses.

**Definition.** An *SLD-resolution refutation*, for clauses $G, D_1, \ldots, D_N$ as above, is a resolution refutation that satisfies the additional condition that,in each application of the resolution rule (1), the clause $L'_1 \vee \cdots \vee L'_n$ is required to be goal. (This implies that $L_1 \vee \cdots \vee L_k$ must always be one of the definite clauses in $D_1, \ldots, D_N$, and thus the conclusion clause $(L_1 \vee \cdots \vee L_{l-1} \vee L_{l+1} \vee \cdots \vee L_k \vee L'_1 \vee \cdots \vee L'_{m-1} \vee L'_{m+1} \vee \cdots \vee L'_n)\theta$ is also goal.)

Work out the one-to-one correspondence between SLD-resolution refutations, as defined above, and derivations in the inference system defined in Theory Lecture 5. Your answer should consider, in turn: how $D_1, \ldots, D_N$ corresponds to a Prolog program; how $G$ corresponds to a Prolog goal; how the unsatisfiablity of the conjunction of $G, D_1, \ldots, D_N$ (as in (ii) above) corresponds to the existentially quantified Prolog goal being a logical consequence of the universally quantified Prolog program; and, finally, the correspondence between refutations and derivations.

5. (a)$^\star$ Prove technical lemmas 1 and 2, from Theory Lecture 5.

   (b)$^{\star\star}$ Formulate and prove analogues of technical lemmas 1 and 2 for general Robinson-style resolution refutations from question 4 above.