Logic Programming 2011–12
Tutorial 5: Proof Search for Predicate Logic

For discussion during Week 7 (Oct. 31 – Nov. 4)

1. Consider the following Prolog program and query.

\[
\begin{align*}
n(e,_) & . \\
n(f(X),g(Y)) & :- n(X,Y) . \\
n(g(X),f(Y)) & :- n(X,Y) . \\
\end{align*}
\]

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

\[
\begin{align*}
\{ X = f(Y,Y) \} & \ \{ X = f(g(Z),W) \} & \ \{ X = f(g(Z),g(Z)) \} \\
\end{align*}
\]

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. What is the result when a variable \( V \) is unified with a (non-constant, non-variable) term \( t \):

(a) when the occurs check is made (see Programming Lecture 2)?

(b) when the occurs check is omitted?

Consider the following program:

\[
\begin{align*}
test & :- X = f(X) . \\
\end{align*}
\]

What is the result of the query “?- test.”, in standard Prolog (with no occurs check)?

The declarative semantics for definite clauses treats an equality statement between terms as having a solution if and only if there is a substitution \( \theta \) that makes the terms identical. Explain why the omission of the occurs check makes Prolog unsound with respect to the declarative reading.