1 Programming Paradigms

Explain clearly the differences between a variable in an imperative language from a variable in a logic language.

2 Unification

Are the following two terms unifiable? If so, determine their most general unifier. If not, point out the subterms that fail to unify.

\[ f(X, g(Y, g(Y, X))) \text{ and } f(A, g(a, g(B, b))) \]

3 Meta-programming

A datalog program is a Prolog program that does not contain any function symbols. Write a Prolog meta-program that checks to see if each clause loaded into the interpreter is either a datalog fact or a datalog rule. (Assume comma operator is right associative.)

4 Expression Evaluation

The following Prolog program evaluates constant expressions:

\[
\begin{align*}
\text{eval}(A+B, V) & : - \text{eval}(A, V1), \text{eval}(B, V2), V \text{ is } V1 + V2. \\
\text{eval}(A*B, V) & : - \text{eval}(A, V1), \text{eval}(B, V2), V \text{ is } V1 * V2. \\
\text{eval}(X, X) & : - \text{integer}(X).
\end{align*}
\]

?- eval(3*4+5, V).
V = 17

Modify the program so that it allows the expression to contain variables. Variable values should be taken from an environment (a list of variable/value pairs), like this:
5 Models

Consider the following Prolog program $P$.

\[
\text{bool}(z). \\
\text{bool}(o). \\
nutt(z,o). \\
nutt(X,Y) :- nutt(Y,X).
\]

1. Give a model for $P$ with domain = \{0,1,2,...\}.

2. Give a Herbrand model for $P$.

3. Give a Herbrand interpretation for $P$ that is not a model.