
Finals (45 pts)

1 Programming Paradigms 6 pts

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

2 Unification 4 pts

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

$f(X, g(Y, g(Y, X)))$ and $f(A, g(a, g(B, b)))$

3 Meta-programming 15 pts

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 10 pts

The following Prolog program evaluates constant expressions:

```
eval(A+B, V) :- eval(A, V1), eval(B, V2), V is V1 + V2.  
eval(A*B, V) :- eval(A, V1), eval(B, V2), V is V1 * V2.  
eval(X, X) :- integer(X).
```

```
?- 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:

```
?- eval([x=3,y=4], x*y+5, V).  
V = 17
```

```
?- eval([x=3], x*y+5, V).  
no
```

5 Models

10 pts

Consider the following Prolog program **P**.

```
bool(z).  
bool(o).  
nutt(z,o).  
nutt(X,Y) :- nutt(Y,X).
```

1. Give a model for **P** with domain = $\{0,1,2,\dots\}$.
2. Give a *Herbrand* model for **P**.
3. Give a *Herbrand* interpretation for **P** that is not a model.