

Discussion Problems: “Prolog Basics”  
Declarative Methods (JHU 600.432/632)  
Prof. Jason Eisner

1. In each case, say whether the unifications succeed. If so, give the resulting variable bindings.

(a) `foo(A,bar(B,baz(4)))=foo(3,bar(4,C))`

(b) `foo(P,bar(4))=foo(Q,Q)`

(c) `b(C,D) = b(D,C)`

(d) `b(C,f(C),f(f(C))) = b(f(f(D)),f(D),X)`

(e) `g(A,g(B,C))=g(C,g(3,g(B,D)))`

(f) `foo(2,bar(3,X))=foo(2,X)`. (Different implementations of unification will behave differently on this one; discuss.)

2. Here is a Prolog program:

```
loves(John,Mary).  
loves(misery,company). % there's an old saying that "misery loves company"
```

(a) Write a Prolog query that asks: “Who loves company?”

(b) What is Prolog’s answer to this query?

3. In ECL<sup>i</sup>PS<sup>e</sup>, we considered constraint systems like

```
X::1..5, Y::1..5, Z::1..5, X #= Y+1, Y #= Z+1.
```

Let’s do this in Prolog without doing any real arithmetic. Assume that you have loaded a program consisting of the following rules:

```
plusone(1,2).
plusone(2,3).
plusone(3,4).
plusone(4,5).
```

- (a) Write a Prolog query that corresponds to the ECL<sup>i</sup>PS<sup>e</sup> constraint system  $X \# = Y+1$ ,  $Y \# = Z+1$  where  $X$  and  $Y$  are assumed to be in the range  $1..5$ .
- (b) What will Prolog answer?
- (c) Now suppose that the program is enlarged to have many more `plusone` clauses, ranging from `plusone(-1000,-999)` to `plusone(999,1000)`. So now you need to impose the constraint  $X::1..5$ ,  $Y::1..5$  explicitly. How can you do this?
- (d) What is the best order to list the constraints in, in terms of the speed of the Prolog solver?

4. Let's define a directed graph in Prolog:

```
vertex(a,1).    vertex(e,3).    vertex(h,4).    vertex(k,5).
vertex(b,2).    vertex(f,3).    vertex(i,4).    vertex(l,5).
vertex(c,2).    vertex(g,3).    vertex(j,4).    vertex(m,5).
vertex(d,2).                                vertex(n,6).
edge(X,Y) :- vertex(X,GroupX), vertex(Y,GroupY), plusone(GroupX,GroupY).
```

- (a) Draw the graph of vertices and directed edges. There should be 14 vertices (i through n).
- (b) Explain how this program is related to “joining” of constraints.
- (c) Let's define paths like this:

```
path(Q,Q).
path(S,U) :- edge(S,T), path(T,U).
```

How many answers are there to the query `path(U,e)`?