Logic Programming Question

Qualifier Examination

July 10, 2002

Question 1

Consider the following logic program P:

```
g(X,Y) :- p(X,Y).
g(X,Y) :- p(X,Z), g(Z,Y).
p(a,b).
p(b,c).
p(c,d).
p(e,f).
```

Write repsonses to each of the following:

1. provide two distinct Herbrand models for this program;
2. describe the notion of T_P operator and discuss under what conditions it admits a least fixpoint;
3. describe the result obtained from repeatedly applying the T_P operator for the above program;
4. list the elements of the least Herbrand model for the above program.

Please make explicit any assumptions you use in your responses.

Question 2

Write repsonses to each of the following:

1. describe the notion of *Computation Rule* as used in logic programming;
2. describe the computation rule used by the Prolog language and provide two distinct programs P_1 and P_2 such that
   (a) the goal $?-p$ computed in P_1 using the Prolog computation rule terminates successfully
   (b) the goal $?-p$ computed in P_2 using the Prolog computation rule leads to an infinite computation;
3. consider the following computation rule (called Basic Andorra Model): in the current goal, select the leftmost subgoal that is defined by a single program clause; if there are no subgoals defined by a single program clause, then select the leftmost subgoal in the goal.

   Provide a simple program P such that the goal $?-p,q$ executed in P using Prolog computation rule terminates with a failure while the same goal executed using Basic Andorra Model gets into an infinite computation. Provide a simple program Q which has the opposite behavior ($?-p,q$ executed in Q using Prolog enters an infinite computation while $?-p,q$ executed in Q using Basic Andorra Model terminates with a failure).

   Hint: You can use the predicate fail.

Question 3

In the description of the unification process used by Prolog we mentioned the idea of *occur-check*.

1. describe the notion of occur-check
2. discuss why Prolog does not implement occur-check
3. provide an example of an unification problem that provides an incorrect answer because of the lack of occur-check