Logic Programming
Coursework 2: Theory

This is the second coursework assignment for Logic Programming. It contributes 10% towards your grade for the course.

Available: **27 October 2014**
Due: **10 November 2014, 3pm**
Submission: Solution to be submitted on paper at the ITO.

1. (a) Consider the following Prolog program and query.

```
d(f(X), f(Y)) :- d(X, Y).
d(f(X), Y) :- d(X, Y).
d(g(X), g(Y)) :- d(X, Y).
d(a, a).

?- d(f(g(a)), Z), d(g(f(a)), Z).
```

Rewrite both program and query in the notation of first-order logic, writing all quantifiers explicitly.

[6 marks]

(b) Draw the full Prolog search tree for the above program and query, and say what response Prolog gives to the query.

[8 marks]

(c) Consider the three terms below.

$$f(g(Y), h(Y)) \quad f(g(Y), h(Z)) \quad f(g(Y), Y)$$

For each pair of these terms, determine whether the two terms are unifiable and do one of the following:

i. If the two terms are unifiable, say what the most general unifier is, and also give an example of one other unifier that is not most general.
ii. If the two terms are not unifiable then explain why not.

[9 marks]

2. (a) Let $A = \{0, 1, 2, 3\}$. Consider the following functions from the power set of A to itself.

$$f_1, f_2, f_3, f_4 \colon \mathcal{P}(A) \to \mathcal{P}(A)$$

$$\begin{array}{rcl} f_1(Y) & = & (A - Y) \cup \{1\} \\ f_2(Y) & = & \{x + 1 \mod 4 \mid x \in Y\} \cup \{0\} \\ f_3(Y) & = & \{x \in A - \{0\} \mid x \text{ divides some } y \in Y - \{0\}\} \\ f_4(Y) & = & \{x \mod 3 \mid x \in Y\} \cup \{2\} \end{array}$$

For each of these four functions do the following:

i. Prove or disprove that the function is monotone.

ii.
- List all of the fixed points.
- If there are no fixed points, give an argument/proof explaining why.
- If there is a least fixed point, identify it.
- Otherwise, give an argument/proof explaining why there is no least fixed point.

(Here "proof" means a mathematical argument - this could be a case-by-case analysis of the behavior of the function on all of the possible inputs, or a higher-level argument explaining why the function cannot have any (least) fixed points.)

[16 marks]

(b) Consider the following propositional Prolog program.

```
a.
b :- a,d.
c :- b,j.
d :- a,e.
e :- a.
j :- b,c.
```

The meaning of the program is defined by the least fixed point of a function $f \colon \mathcal{P}(\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{j}\}) \to \mathcal{P}(\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{j}\})$. Give a precise definition of the function f, and calculate sufficiently many iterated applications of f to the empty set to find its least fixed point. Show your workings.

[6 marks]

(c) Briefly explain how the calculation of a least-fixed-point can be used to implement a decision procedure for definite clause propositional logic.

[3 marks]

(d) Give one reason that it would not be appropriate to for Prolog to replace its proof-search-based strategy with a fixed-point-based decision procedure.

[2 marks]