AP/PHIL/COGS 3750 Philosophy of Artificial Intelligence
Winter 2011

Dept. of Philosophy
York University

Programming Assignment

Total marks: 40.

Due: February 14 at 1pm

Note: Your report for this assignment should be the result of your own individual work. Take care to avoid plagiarism ("copying"). You may discuss the problems with other students, but do not take written notes during these discussions, and do not share your written solutions.

1. [20 points] Write and test a Prolog program that extends the family relations example of ch. 4 in the Levesque textbook.

   Begin with the code in the file `family_asg.pl` that is available on the course web site. Add the Prolog code to define the following relations:

   - `sibling(X,Y)`, meaning that X is a sibling of Y, i.e. X and Y are two different people who share a parent in common;
   - `first_cousin(X,Y)`, meaning that X is a first cousin of Y, i.e. some parent of X and some parent of Y are siblings;
   - `descendant(X,Y)`, meaning that X is a descendant of Y, i.e. X is either a child of Y or (recursively) of someone who is a descendant of Y;
   - `ancestor(X,Y)`, meaning that X is an ancestor of Y, i.e. Y is a descendant of X;
   - `common_ancestor(X,Y,Z)`, meaning that X is a common ancestor of Y and Z, i.e. X is an ancestor of both Y and Z;
   - `closest_common_ancestor(X,Y,Z)`, meaning that X is a closest common ancestor of Y and Z, i.e. X is a common ancestor of Y and Z and no child of X is a common ancestor of Y and Z.

   You may define some auxiliary relations if that helps in defining the ones above. See p. 89-90 of the Levesque textbook for more details about these relations. Test your code thoroughly to make sure it is correct. Do not add additional child assertions to the ones already in the `family_asg.pl` file.

   Submit (i) a printed copy of your program, and (ii) a printout of a session where you load and test your program. Run at least the following test queries:

```
sibling(sue,mary).
sibling(john,X).
sibling(X,Y).
first_cousin(john,henry).
first_cousin(john,Y).
first_cousin(X,Y).
closest_common_ancestor(X,john,jane).
closest_common_ancestor(X,john,henry).
closest_common_ancestor(sam,john,Z).
```

Whenever a query contains variables, keep asking for further answers until they run out.

Also email your program code file `family_asg.pl` to `lesperan@cse.yorku.ca` (using the subject "PHIL 3750 asg 1").

2. [20 points] Write and test a Prolog program that solves the puzzle in exercise 4 on p. 124 of the Levesque textbook. Running the query `print_solution` should display who ordered each of the main courses and each of the beverages. (If you find this too difficult, just write code that solves the puzzle and displays who ordered pizza, and you will get partial credit.)

   Submit (i) a printed copy of your program, and (ii) a printout of a session where you load and test your program by running the `print_solution` query.

   Also email your program code file `lunch_asg.pl` to `lesperan@cse.yorku.ca` (using the subject "PHIL 3750 asg 1").

Note that points will be given for partial solutions, even if they do not run without errors. Submit whatever you have by the deadline. Note also that points will be deducted if your program has syntax errors, does not produce the correct answers, or is more complex than necessary.