

Concordia University
Department of Computer Science and Software Engineering
COMP 348: Principles of Programming Languages
Fall 2017
Assignment 1
Evaluation: 90 points
(5% of your final grade)
Due date and time: Monday October 16th 2017 at 23:59

Question 1 (7 pts) **Knowledge representation in prolog**

Write a Prolog program that determines who are siblings and cousins of each other from a database of Prolog facts.

The database of Prolog facts should capture all the information from the following records:

```
mary and catherine are sisters.  
john and simone are brothers.  
  
john and mary got married in 2010.  
mary gave birth to johnny in 2012.  
mary gave birth to peter in 2015.  
  
josh and catherine got married in 2011.  
catherine gave birth to william in 2012.  
  
simone and kate married in 2009.  
kate gave birth to betty in 2011.
```

(There might be many more records on sisters, brothers, marriages, and newborns.)

If two children's father or mother is the same, they are considered siblings. For example, your program should answer "yes" to the following query:

```
?- siblings(johnny, peter).  
Yes  
And "no" to the following query:  
?- siblings(johnny, william).  
No
```

You should also provide at least one rule for cousins/2, so that `cousins(S,johnny)` will output all S instantiations that are cousins of johnny.

For example:

```
?- cousins(S,johnny).  
Yes  
S=william  
S=betty
```

Question 2 (8 pts) **Unifications and resolutions in Prolog**

Which of the following pairs of terms can be unified (matched) together? Where relevant, give the variable instantiations that lead to successful unification. (Note = shows unification)

```
healthyFood(X) = healthyFood(bread)
healthyFood(bread,X) = healthyFood(Y,salad)
healthyFood(bread,X,milk) = healthyFood(Y,salad,X)
healthyFood(X) = Y
meal(healthyFood(bread),drink(milk)) = meal(X,Y)
meal(healthyFood( Z ),drink(milk)) = meal(X,Y)
meal(healthyFood(bread),drink(milk)) = meal(X, drink(Water))
meal(healthyFood(bread), Y) = meal(X, drink(water))
breakfast(healthyFood(bread),egg,milk)= breakfast(healthyFood(Y),Y, Z)
dinner( X, Y, Time) = dinner( jack, cook( egg, oil), Evening)
k(s(g), Y) = k(X, t(k))
```

Question 3 (15 pts) **Unifications and resolutions in Prolog**

Assume we have the following knowledge base in a Prolog program:

```
man(jack).
man(peter).
woman(rebeca).
woman(julia).
woman(maria).
hasWand(rebeca).
hasWand(maria).
hasWand(jack).
quidditchPlayer(jack).
quidditchPlayer(rebeca).
quidditchPlayer(maria).
quidditchPlayer(peter).
playsAirGuitar(julia).
playsAirGuitar(adam).
playsAirGuitar(rebeca).
playsAirGuitar(mary).
playsAirGuitar(jack).
wizard(jack).
hasBroom(X) :- quidditchPlayer(X).
warlock(X) :- man(X), hasBroom(X), hasWand(X).
witch(X) :- woman(X), hasBroom(X), hasWand(X).
wizard(X) :- warlock(X) ; witch(X).    % note: semicolon was used here
```

Determine the type of each of the following queries (ground/non-ground), and explain what will Prolog respond for each of these queries (write all the steps of unifications and resolutions for each query)?

```

? -wizard(jack).
? -witch(jack).
? -warlock(jack).
? -witch(maria).
? -warlock(Y).
? -witch(Y).
? -wizard(X).
? -hasBroom(X).
? -playsAirGuitar(Y), witch(Y).
? -witch(Y), witch(maria).
? -hasBroom(X), !, playsAirGuitar(Y) % note there is a cut (!) here

```

Question 4 (20 pts) **Lists and Backtracking**

Consider the following Prolog program:

```

outcome([_, E | L], [E | M]) :- !,
    outcome(L, M).

outcome(_, []).

```

- 1) After having consulted this program, what would Prolog reply when presented with the following query? Try answering this question first without actually typing in the program, but verify your solution later on using the Prolog system.

```

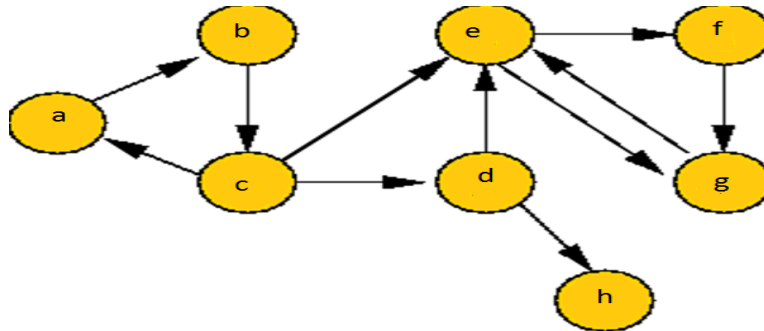
?- outcome([a, b, c, d, e, f, g], X).

```

- 2) Briefly describe what the program does and how it does what it does, when the first argument of the `outcome/2`-predicate is instantiated with a list and a variable is given in the second argument position, i.e., as in item (a). Your explanations should include answers to the following questions:
 - a) What case(s) is/are covered by the Prolog fact?
 - b) What effect has the cut in the first line of the program?
 - c) Why has the anonymous variable been used?

Question 5 (20 pts) Graph in Prolog

The Purpose of this question is to write a Prolog Program which describes a directed graph (G), with the following structure and allows us to ask some questions about this graph.



- 1) Write all possible node and edge facts that describes the structure of this graph G as examples below:

```
node(a).           % "a" is a node of this graph.
edge(a,b).         % There is an edge (directed) from node "a" to "b".
```

- 2) Complete the definition for all these rules.

```
parent(X,Y) :- % There is a directed edge from X to Y.
ancestor(X,Y) :- % X is an ancestor of Y
child(X,Y) :- % Y is parent of X.
path(X,Y) :- % find a directed path from node X to node Y.
length_of_path(X,Y) :- % Length of a path (directed) from X to Y.
length_of_path(X,Y, Z) :- % Length of a path (directed) from X to Y
will be assigned to Z
connected(X,Y) :- % There is a directed path from X to Y, or from Y to X
undirected_edge(X,Y) :- /* There is an edge (ignoring the directions)
from X to Y or from Y to X */
undirected_path(X,Y) :- /* find a path (ignoring the directions)
from node X to node Y. */
```

Important Note: There are circles in this graph including nodes some nodes such as: e, f, g. You should find a way to avoid infinite recursion by making sure that you visit each node along your paths once.

- 3) Create a knowledge base (KB) composed of all the facts and rules that you have written in parts (1), and (2), and save it in a prolog program file called "my_graphG.pl", and show a print of your program here.
- 4) Run this Prolog program, and write 4 queries about each of those rules that you have created in part (2) in order to test those rules, and show your queries and their results, and test the correctness of your results. (Note: you should write and test two ground quires (one with positive and one with negative answer) and two non-ground quires for each of the rules that you have created in Part B.)

- 5) Add the following rule to your program, and test it and describe its function?
- ```
tpath(Node1,Node2):-edge(Node1, SomeNode), edge(SomeNode,Node2).
```

### **Question 6** (20 pts) **Using Prolog for a Search Problem**

Assume, you are working with the following knowledge base:

```
house_elf(dobby).
witch(hermione).
witch('McGonagall').
witch(rita_skeeter).
magic(X):-house_elf(X).
magic(X):-wizard(X).
magic(X):-witch(X).
```

```
house_elf(dobby).
witch(hermione).
witch(mcGonagall).
witch(rita_skeeter).
wizard(dobby).
magic(X):- house_elf(X).
magic(X):- wizard(X).
magic(X):- witch(X).
```

Write the details of steps of search (unification, resolutions, and back tracking) and also the answer for each of the following queries. (You can show the details of your search process by drawing search trees for each of the following queries)

```
?- magic(Hermione).
?- magic(hermione).
```

### **Submission:**

- **Assignment must be done individually (no groups are permitted).**
  - Create one zip file, containing all files for your assignment.
  - Name your zip file this way: *a1\_studentID*, where *studentID* is your ID number, for the first assignment, student 123456 would submit a zip file named *a1\_123456.zip*
- Assignments must be submitted through the course web page(EAS) or Moodle site of the course (please check your section).

|                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Note: Assignment not submitted by the due date and in the correct format will not be graded – NO EXCEPTIONS!!!!</b></p> |
|-------------------------------------------------------------------------------------------------------------------------------|