CECS 342
Lab assignment 5
Due date: Wednesday, May 1
20 points

Prolog Programming

Install SWI-Prolog in windows 10

Step 1 Download SWI-Prolog.

Step 2 Run the Installer (Follow the instruction given on the installer).

Step 3 Open the SWI-Prolog.

Step 4 Click on File (in menu) File-->New (a new window will open save it by any name with extension .pl). A new window will open you can write your program here. After writing the program select compile (in menu in the same window) and click on “make” and again click on “compile buffer”.

Step 5 Run the query on the SWI-Prolog window.

1. [5 points] Create the following knowledge base. Name the program kbase1.

```
woman(mia).
woman(jody).
woman(yolanda).

loves(vincent,mia).
loves(marcellus,mia).
loves(pumpkin,honey_bunny).
loves(honey_bunny,pumpkin).
```

   Create the following queries:
   a. tell me which of the individuals you know about is a woman.
   b. Is there any individual X such that Marcellus loves X and X is a woman?

2. [5 points] Create the following knowledge base. Name the program kbase2.

```
loves(vincent,mia).
loves(marcellus,mia).
loves(pumpkin,honey_bunny).
loves(honey_bunny,pumpkin).
```

   a. Create a rule:

      It says that an individual X will be jealous of an individual Y if there is some individual Z that X loves, and Y loves that same individual Z too.

   b. Create the following query:

      Can you find an individual W such that Marcellus is jealous of W?

3.[5 points] Write a Prolog relation that accepts a list of integers, and counts the number of zeros in the list, e.g.,

? zeros([1, 0, 0, 5], X).

X = 2

? zeros([ ], X).

X = 0

Hint:

zeros([ ], 0).

zeros([0 | T], Z):- _____________, !, Z is Z1 + 1.

___________________:- zeros(T, Z).

```
[trace]  ?- zeros([1, 0, 0, 5], X).
   Call: (12) zeros([1, 0, 0, 5], _3936) ? creep
   Call: (13) zeros([0, 0, 5], _3936) ? creep
   Call: (14) zeros([0, 5], _5934) ? creep
   Call: (15) zeros([5], _6690) ? creep
   Call: (16) zeros([], _6690) ? creep
   Exit: (16) zeros([], 0) ? creep
   Exit: (15) zeros([5], 0) ? creep
   Call: (15) _5934 is 0+1 ? creep
   Exit: (15) 1 is 0+1 ? creep
   Exit: (14) zeros([0, 5], 1) ? creep
   Call: (14) _3936 is 1+1 ? creep
   Exit: (14) 2 is 1+1 ? creep
   Exit: (13) zeros([0, 0, 5], 2) ? creep
   Exit: (12) zeros([1, 0, 0, 5], 2) ? creep
X = 2.
```

4.[5 points] Write a Prolog relation "intersect(L1, L2, R)" that succeeds if R is the intersection of L1 and L2. (*Assume no duplicates*), e.g.,

? intersect([0, 1, 6, 3], [5, 1, 8, 2, 3, 9], X).

X = [1, 3]

? intersect([1], [4], X).

X = [ ]

Hint:

intersect([ ], X, [ ]).

intersect([X|R], Y, [X|Z]) :- ________(X, Y), !, intersect(___, Y, Z).

intersect([___], Y, Z) :- intersect(__,__,__).

```
[trace]  ?- intersect([1], [4], X).
   Call: (12) intersect([1], [4], _4134) ? creep
   Call: (13) lists:member(1, [4]) ? creep
   Fail: (13) lists:member(1, [4]) ? creep
   Redo: (12) intersect([1], [4], _4134) ? creep
   Call: (13) intersect([], [4], _4134) ? creep
   Exit: (13) intersect([], [4], []) ? creep
   Exit: (12) intersect([1], [4], []) ? creep
X = [].
```

```
[trace]  ?- intersect([0, 1, 6, 3], [5, 1, 8, 2, 3, 9], X).
   Call: (12) intersect([0, 1, 6, 3], [5, 1, 8, 2, 3, 9], _2746) ? creep
   Call: (13) lists:member(0, [5, 1, 8, 2, 3, 9]) ? creep
   Fail: (13) lists:member(0, [5, 1, 8, 2, 3, 9]) ? creep
   Redo: (12) intersect([0, 1, 6, 3], [5, 1, 8, 2, 3, 9], _2746) ? creep
   Call: (13) intersect([1, 6, 3], [5, 1, 8, 2, 3, 9], _2746) ? creep
   Call: (14) lists:member(1, [5, 1, 8, 2, 3, 9]) ? creep
   Exit: (14) lists:member(1, [5, 1, 8, 2, 3, 9]) ? creep
   Call: (14) intersect([6, 3], [5, 1, 8, 2, 3, 9], _7146) ? creep
   Call: (15) lists:member(6, [5, 1, 8, 2, 3, 9]) ? creep
   Fail: (15) lists:member(6, [5, 1, 8, 2, 3, 9]) ? creep
   Redo: (14) intersect([6, 3], [5, 1, 8, 2, 3, 9], _7146) ? creep
   Call: (15) intersect([3], [5, 1, 8, 2, 3, 9], _7146) ? creep
   Call: (16) lists:member(3, [5, 1, 8, 2, 3, 9]) ? creep
   Exit: (16) lists:member(3, [5, 1, 8, 2, 3, 9]) ? creep
   Call: (16) intersect([], [5, 1, 8, 2, 3, 9], _12496) ? creep
   Exit: (16) intersect([], [5, 1, 8, 2, 3, 9], []) ? creep
   Exit: (15) intersect([3], [5, 1, 8, 2, 3, 9], [3]) ? creep
   Exit: (14) intersect([6, 3], [5, 1, 8, 2, 3, 9], [3]) ? creep
   Exit: (13) intersect([1, 6, 3], [5, 1, 8, 2, 3, 9], [1, 3]) ? creep
   Exit: (12) intersect([0, 1, 6, 3], [5, 1, 8, 2, 3, 9], [1, 3]) ? creep
X = [1, 3].
```

Submission:

Submit the following (one copy for all the team members) to the Dropbox:

1. Prolog file for each problem
2. a pdf file that contains the prolog program, queries, and the result of queries
3. Write a completion of each team member on the comment box. Student who finishes one problem gets a 25% of completion.

Missing the following items:
Item 1 (-20 points)
Item 2 (-15 points)
Item 3 (-20 points)