Lab 2

1. **Making a .pl file and consulting.**

Make a simple text file and rename it to test.pl.

Write in file

boy(ram).

Through the Prolog Terminal, use the command

?- consult (“Absolute path of the file”).

In my case,

?-consult (D:/AI/test.pl)

Then Try:

?- boy(ram).

TRUE

The file can be directly consulted form File>Consult>Select File.

1. **First File**

Filename : first.pl

boy(ram).

girl(sita).

Terminal:

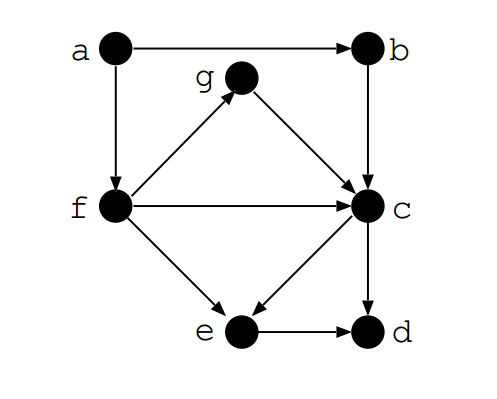
1 ?- boy(ram). // ram is a boy. So the program returns TRUE after consulting

true.

2 ?- boy(sita). //sita is not a boy. So the program returns FALSE after consulting

false.

1. **Graph Representation**



File name: graph.pl

edge(a,b). // a vertex is connected to vertex b

edge(a,f).

edge(f,e).

edge(e,d).

edge(b,c).

edge(f,c).

edge(c,e).

edge(f,g).

edge(g,c).

edge(c,d).

path(Node1,Node2):- edge(Node1,Node2). // If node1 and node 2 is connected, then there is path between them.

path(Node1, Node2) :- edge(Node1,Somenode),path(Somenode,Node2). /\* This is the recursion that checks the path between two nodes which are connected through many edges \*/

Terminal:

1 ?- edge(a,b).

true .

2 ?- edge(X,c).

X = b ;

X = f ;

X = g.

3 ?- path(a,f).

true .

4 ?- path(a,d).

true ;

1. **Family Representation**

**Filename: family.pl**

father(ram).

mother(sita).

male(amar).

male(chandra).

female(bina).

female(divya).

parent(amar,chandra).

parent(amar,divya).

parent(bina,chandra).

parent(bina,divya).

sibling(X,Y):- parent(Z,X),parent(Z,Y),different(X,Y). /\* X is sibling of Y if parents are same and X and Y are different \*/

different(X,X):- !,fail. /\* X is not different to X \*/

different(X,Y). /\* X is different than Y \*/

Terminal:

1 ?- father(ram).

true.

2 ?- parent(X,chandra).

X = amar ;

X = bina.

3 ?- parent(X,divya).

X = amar ;

X = bina.

4 ?- sibling(chandra,divya).

true .

5 ?- sibling(chandra,chandra).

false.

6 ?- siblin(chandra,amar).

Correct to: "sibling(chandra,amar)"?

Please answer 'y' or 'n'? yes

false.