# LAB2: Introduction to Prolog

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

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

Write in file

likes(ram,food).

Through the Prolog Terminal, use the command

Consult file from: File>Consult>Select File.

or

?-consult (E:/test.pl)

?- likes(ram,food).

TRUE

1. **First File**

Filename test.pl

likes(ram,hari).

likes(hari,noodles).

likes(ram,drinking).

Terminal:

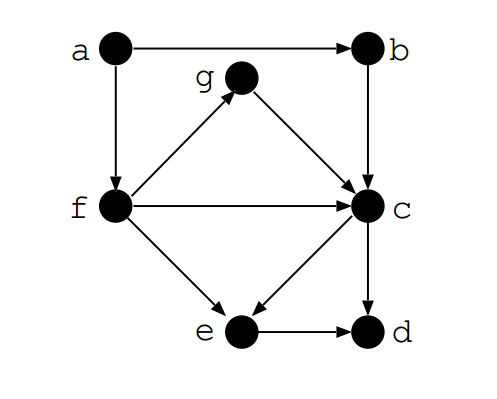
1 ?- likes(ram,hari). // ram likes hari. So the program returns TRUE after consulting

true.

2 ?- likes(hari,ram). //hari does not like hari. 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(rohit).

male(chandra).

female(binda).

female(diya).

parent(ram,diya).

parent(ram,binda).

parent(sita,chandra).

parent(sita,diya).

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,diya).

X = ram ;

X = sita.

3 ?- parent(X,diya).

X = ram;

X = sita.

4 ?- sibling(diya,binda).

true .

5 ?- sibling(chandra,chandra).

false.