**Introduction to Prolog**

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”).

?- consult('/home/bips/Desktop/test.pl').

true.

Then,

?- 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 ?- consult('/home/bips/Desktop/prolog/test.pl').

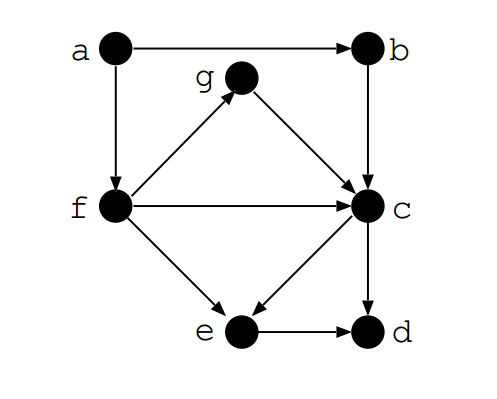
2 ?- boy(ram).

true.

3 ?- boy(sita).

False.

1. **Graph Representation**



File name: graph.pl

edge(a,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).

/\*If two vertices Node1 and Node2 are connected, there is path between them.\*/

path(Node1,Node2):- edge(Node1,Node2).

/\* This is the recursion that checks whether there is path between two nodes which are connected through many edges \*/

path(Node1, Node2) :- edge(Node1,Somenode),path(Somenode,Node2).

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**

File name: family.pl

male(amar).

male(chandra).

female(bina).

female(divya).

parent(amar,chandra).

parent(amar,divya).

parent(bina,chandra).

parent(bina,divya).

father(X,Y):-parent(X,Y), male(X).

mother(X,Y):-parent(X,Y), female(X).

/\* X is sibling of Y if parents are same and X and Y are different \*/

sibling(X,Y):- parent(Z,X),parent(Z,Y),different(X,Y).

/\* X is not different to X \*/

different(X,X):- !,fail.

/\* X is different than Y \*/

different(X,Y).

Terminal

?- consult('family.pl').  
true.

?- mother(bina, divya).  
true.

?- father(amar, divya).  
true.

?- father(amar, chandra).  
true .

?- sibling(chandra, chandra).  
false.

?- sibling(chandra, divya).  
true .

?- parent(X, chandra).  
X = amar ;  
X = bina.

?- sibling(amar, chandra).  
false.