Contents

[About Prolog 1](#_Toc453044326)

[Introduction to Prolog: 1](#_Toc453044327)

[1. Making a .pl file and consulting. 1](#_Toc453044328)

[2. First File 1](#_Toc453044329)

[3. Graph Representation 2](#_Toc453044330)

[4. Family Representation 4](#_Toc453044331)

Prolog Lab (Introduction)

# About Prolog

Prolog is a general-purpose logic programming language associated with artificial intelligence and computational linguistics.

Prolog has its roots in first-order logic, a formal logic, and unlike many other programming languages, Prolog is declarative: the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations.

The language was first conceived by a group around Alain Colmerauer in Marseille, France, in the early 1970s and the first Prolog system was developed in 1972 by Colmerauer with Philippe Roussel.

## Introduction to Prolog:

### Making a .pl file and consulting.

Create a file and name is test.pl and save it in a location (In my case)

D:\AI\Labs\test.pl

In the file write boy(ram).

**In the Prolog Terminal**

1 ?- consult('D:\\AI\\Labs\\test.pl').

true.

2 ?- boy(ram)

| .

true.

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

### First File

Create a file named first.pl and write following in the file

boy(ram).

girl(sita).

**In the Prolog Terminal**

3 ?- consult('D:\\AI\\Labs\\first.pl')

| .

true.

4 ?- boy(ram)

| .

true.

5 ?- girl(ram)

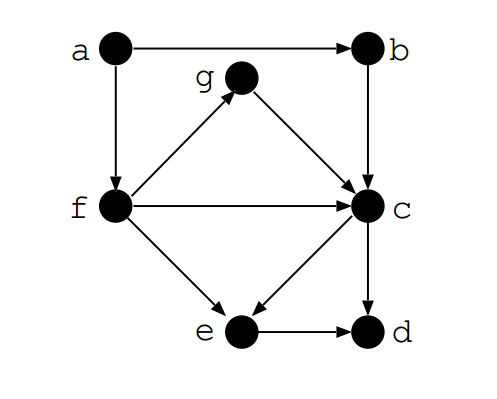
| .

false.

?- girl(sita).

true.

### 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 ;

### Family Representation

Create a file graph.pl and add the following

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).

**In the Prolog 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.