Vishwakarma Institute of Technology
Department of Computer Science Engineering (CS)

Name	Vaishnavi Vijay Arthamwar
Division	CS-A
Batch	B3
Roll No.	16
PRN No.	12220242
Department	Computer
Subject	Artificial Intelligence
Assignment No.	5

Assingment No. 5

Title: Implementation of Predicate logic using PROLOG

Theory:

Predicate logic, also known as first-order logic, is a formal system used to represent and reason about statements involving quantified variables and predicates. Prolog (Programming in Logic) is a programming language based on predicate logic and is particularly well-suited for implementing and reasoning about logical statements.

In this assignment, we aim to implement predicate logic using Prolog. The implementation involves defining predicates, rules, and facts to represent logical statements and relationships between entities.

Key Concepts:

1.Predicates: Predicates represent relationships between entities and are defined using Prolog clauses. Predicates can be unary (having one argument) or n-ary (having multiple arguments).

For example:

parent(X, Y)   % X is the parent of Y

employee(Name, Department, Salary)   % Name works in Department and earns Salary

2.Facts: Facts are statements that are known to be true. In Prolog, facts are represented as predicates without any variables.

For example:

parent(john, alice).   % John is the parent of Alice

employee('Jane Doe', 'Engineering', 80000).   % Jane Doe works in Engineering and earns $80,000

3.Rules: Rules define logical implications or relationships based on certain conditions. In Prolog, rules are represented as clauses with a head (conclusion) and a body (conditions).

For example:

grandparent(X, Z) :- parent(X, Y), parent(Y, Z).   % X is the grandparent of Z if X is the parent of some Y and Y is the parent of Z

4.Variables: Variables in Prolog begin with an uppercase letter or an underscore. Variables represent unknown values and can be instantiated with values that satisfy the given predicates.

<u>For example:</u>

parent(X, alice).   % X is a parent of Alice (X is unknown)

5.Queries: Queries are Prolog goals or statements that we want to evaluate for truthfulness. Prolog searches for solutions by unifying predicates with facts and rules.

<u>For example:</u>

?- parent(john, alice).   % Is John the parent of Alice?

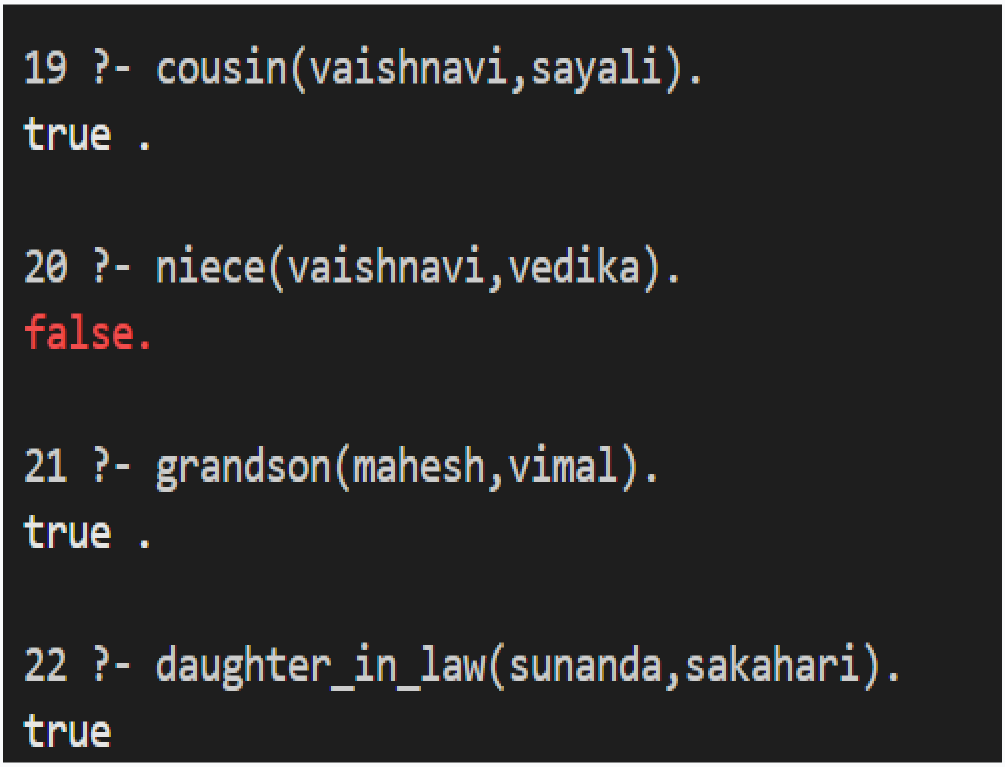
?- grandparent(john, alice).   % Is John the grandparent of Alice?

Code:

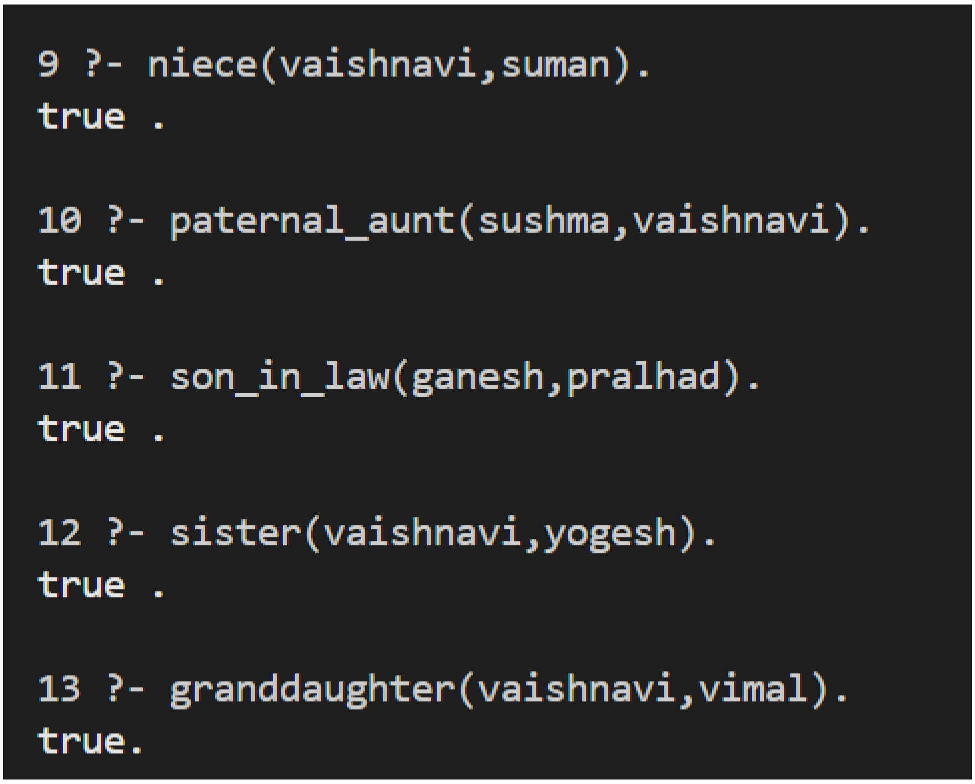
```
% Facts about the family members
male(pralhad).
male(sakahari).
male(omprakash).
male(dnyaneshwar).
male(kishan).
male(megha).
male(vijay).
male(mahesh).
male(shubham).
male(yogesh).
male(sanjay).
male(sainath).
male(ganesh).
male(gajanan).
male(shree).
male(mahesh).
male(sagar).
male(sai).
male(parth).
male(shravan).
male(samarth).

female(vimal).
female(pushpa).
female(sakhu).
female(suman).
female(kanta).
female(sunanda).
female(suchita).
female(vaishnavi).
female(soni).
female(jayshree)
female(sangita).
female(jyoti).
female(sushma).
female(vedika).
female(ishwari).
female(pranavi).
female(sayali).
female(nayan).
female(shruti).
female(gayatri).

% Parent-child relationships
parent(pralhad, kanta).
parent(pralhad, sakhu).
parent(pralhad, suman).
parent(pralhad, gajanan).
parent(pralhad, sunanda).
parent(pralhad, sai).
parent(vimal, sai).
parent(vimal, kanta).
parent(vimal, sakhu).
parent(vimal, suman).
parent(vimal, gajanan).
parent(vimal, sunanda).
parent(sakahari, vijay).
parent(sakahari, sanjay).
parent(sakahari, sainath).
parent(pushpa, vijay).
parent(pushpa, sanjay).
parent(pushpa, sainath).
parent(vijay, vaishnavi).
parent(vijay, yogesh).
parent(sunanda, vaishnavi).
parent(sunanda, yogesh).
parent(suchita, vedika).
parent(sai, vedika).
parent(suchita, ishwari).
parent(sai, ishwari).
parent(suchita, pranavi).
parent(sai, pranavi).
parent(kishan,shubham).
parent(kishan,jyoti).
parent(suman,shubham).
parent(suman,jyoti).
parent(kishan,soni).
parent(suman,soni).
parent(kishan,jayshree).
parent(suman,jayshree).
parent(dnyaneshwar,mahesh).
parent(kanta,mahesh).
parent(dnyaneshwar,shree).
parent(kanta,shree).
parent(dnyaneshwar,sagar).
parent(kanta,sagar).
parent(sakhu,parth).
parent(ganesh,parth).
parent(sakhu,sayali).
parent(sakhu,nayan).
parent(ganesh,sayali).
parent(ganesh,nayan).
parent(sainath,gayatri).
parent(sangita,gayatri).
parent(sainath,samarth).
parent(sangita,samarth).
parent(sushma,shruti).
parent(sushma,shravan).
parent(sanjay,shruti).
parent(sanjay,shravan).

% Marriage relationships
married(pralhad, vimal).
married(sakahari,pushpa).
married(dnyaneshwar, kanta).
married(kishan, suman).
married(gajanan,megha).
married(sai,suchita).
married(vijay,sunanda).
married(sainath,sangita).
married(ganesh,sakhu).
married(sanjay,sushma).
married(sai,suchita).


% Rules to define relationships
maternal_grandmother(X,Y):- female(X),parent(X,Z),female(Z),parent(Z,Y).

maternal_grandfather(X,Y):- male(X),parent(X,Z),female(Z),parent(Z,Y).

paternal_grandmother(X,Y):- female(X),parent(X,Z),male(Z),parent(Z,Y).

paternal_grandfather(X,Y):- male(X),parent(X,Z),male(Z),parent(Z,Y).
```

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

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

brother(X,Y):- male(X),parent(Z,X),parent(Z,Y),X\=Y.

sister(X,Y):- female(X),parent(Z,X),parent(Z,Y),X\=Y.

sibling(X,Y):-parent(Z,X),parent(Z,Y),X\=Y.

child(X,Y):- parent(Y,X).

husband(X,Y):- male(X),female(Y),married(X,Y).

wife(X,Y):- female(X),male(Y),married(Y,X).

grandson(X,Y):-male(X),parent(Y,Z),parent(Z,X).

granddaughter(X,Y):-female(X),parent(Y,Z),parent(Z,X).

daughter_in_law(X,Y):- female(X),married(Z,X),parent(Y,Z).

son_in_law(X,Y):-male(X),married(X,Z),parent(Y,Z).

maternal_aunt(X,Y):-
    female(X),female(Z),sister(X,Z),parent(Z,Y);
    female(X),married(Z,X),maternal_uncle(Z,Y).

paternal_aunt(X,Y):-
    female(X),male(Z),sister(X,Z),parent(Z,Y);
    female(X),married(Z,X),paternal_uncle(Z,Y).

maternal_uncle(X,Y):-
    male(X),female(Z),brother(X,Z),parent(Z,Y);
    male(X),married(X,Z),maternal_aunt(Z,Y).

paternal_uncle(X,Y):-
    male(X),male(Z),brother(X,Z),parent(Z,Y);
    male(X),married(X,Z),paternal_aunt(Z,Y).

cousin(X,Y):- parent(A,X),parent(B,Y),sibling(A,B).

niece(X,Y):-female(X),parent(A,X),sibling(A,Y).

nephew(X,Y):-male(X),parent(A,X),sibling(A,Y).

/* <examples>
?- grandson(mahesh,vimal)
?- daughter_in_law(sunanda,sakahari)
?- granddaughter(vaishnavi,sakahari)
?- nephew(mahesh,sunanda)
?- niece(vaishnavi,suman)
?- sister(vaishnavi,yogesh)
?- maternal_grandmother(vimal,vaishnavi)
?- maternal_grandmother(vimal,mahesh)
?- brother(vijay,sanjay)
?- maternal_aunt(suman,vaishnavi)
?- paternal_aunt(sushma,vaishnavi)
?- son_in_law(ganesh,pralhad)
?- son_in_law(vijay,pralhad)
*/
```

Output:



```
19 ?- cousin(vaishnavi,sayali).
true .

20 ?- niece(vaishnavi,vedika).
false.

21 ?- grandson(mahesh,vimal).
true .

22 ?- daughter_in_law(sunanda,sakahari).
true
```



```
9 ?- niece(vaishnavi,suman).
true .

10 ?- paternal_aunt(sushma,vaishnavi).
true .

11 ?- son_in_law(ganesh,pralhad).
true .

12 ?- sister(vaishnavi,yogesh).
true .

13 ?- granddaughter(vaishnavi,vimal).
true.
```

Conclusion:

In summary, By implementing predicate logic using Prolog, we can leverage its powerful inference mechanism to reason about relationships and make logical deductions based on the defined rules and facts. This assignment provides hands-on experience in applying predicate logic concepts in a programming context and demonstrates the practical utility of Prolog for logical reasoning tasks.