NAME: Meith Navlakha
SAP: 60004190068
AI Practical

EXPERIMENT 9

AIM: Program to implement Family Tree in Prolog.

THEORY:

Prolog is a logic programming language. It has important role in artificial intelligence. Unlike many other programming languages, Prolog is intended primarily as a declarative programming language. In prolog, logic is expressed as relations (called as Facts and Rules). Core heart of prolog lies at the logic being applied. Formulation or Computation is carried out by running a query over these relations.

Syntax and Basic Fields:

In prolog, we declare some facts. These facts constitute the Knowledge Base of the system. We can query against the Knowledge Base. We get output as affirmative if our query is already in the knowledge Base or it is implied by Knowledge Base, otherwise we get output as negative. So, Knowledge Base can be considered similar to database, against which we can query. Prolog facts are expressed in definite pattern. Facts contain entities and their relation. Entities are written within the parenthesis separated by comma (,). Their relation is expressed at the start and outside the parenthesis. Every fact/rule ends with a dot (.). So, a typical prolog fact goes as follows:

Format: relation(entity1, entity2, ...k'th entity).

Example:

friends(Suresh, Mahesh).

singer(Micca).

odd_number(7).

Explanation:

These facts can be interpreted as:

Suresh and Mahesh are friends.

Micca is a singer.

7 is an odd number.

Key Features:

1. Unification: The basic idea is, can the given terms be made to represent the same structure.
2. Backtracking: When a task fails, prolog traces backwards and tries to satisfy previous task.
3. Recursion: Recursion is the basis for any search in program.

Running queries:
A typical prolog query can be asked as:

- Query 1: ?- singer(sonu).
  Output: Yes.
  Explanation: As our knowledge base contains the above fact, so output was 'Yes', otherwise it would have been 'No'.

- Query 2: ?- odd_number(7).
  Output: No.
  Explanation: As our knowledge base does not contain the above fact, so output was 'No'.

Advantages:

1. Easy to build database. Doesn't need a lot of programming effort.
2. Pattern matching is easy. Search is recursion based.
3. It has built in list handling. Makes it easier to play with any algorithm involving lists.

Disadvantages:

1. LISP (another logic programming language) dominates over prolog with respect to I/O features.
2. Sometimes input and output is not easy.

Applications:

Prolog is highly used in artificial intelligence (AI). Prolog is also used for pattern matching over natural language parse trees.

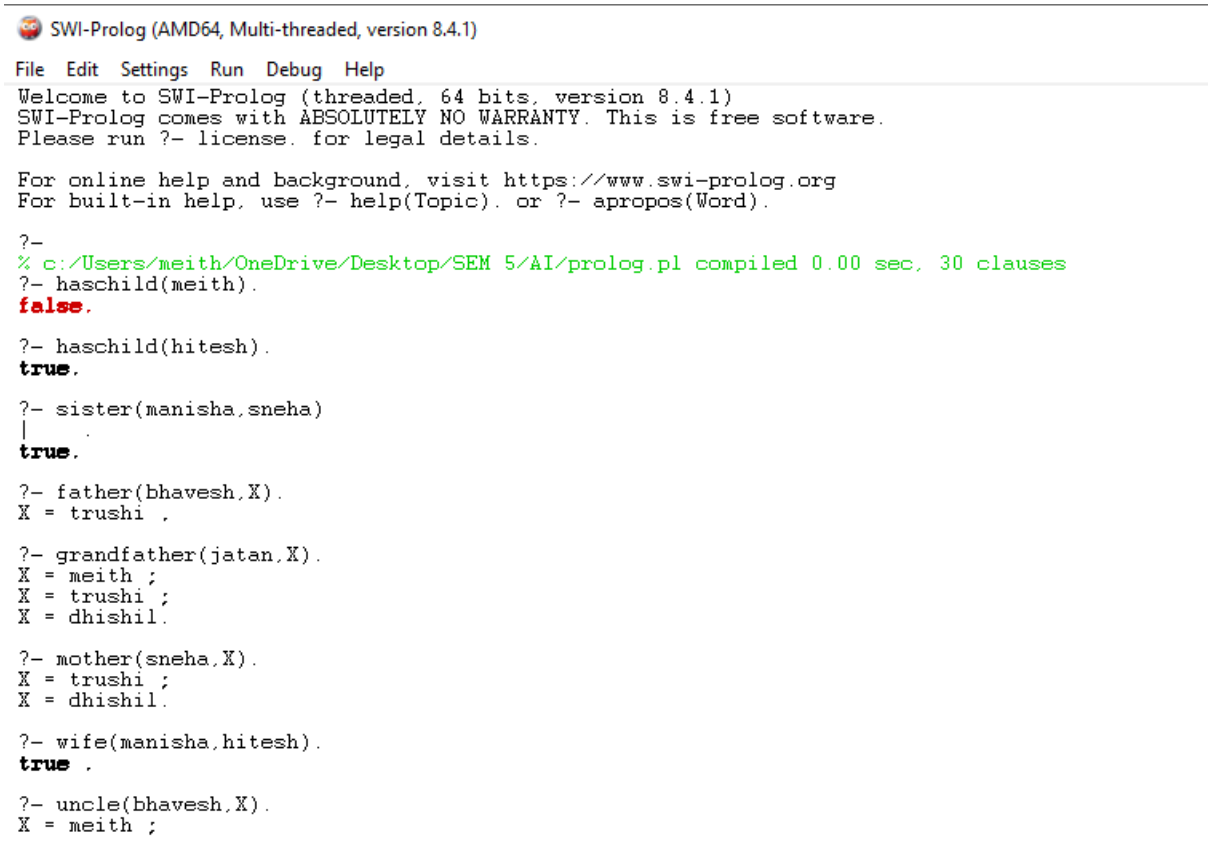
NAME: Meith Navlakha
SAP: 60004190068
AI Practical

CODE:

```
female(manisha).
female(sneha).
female(trushi).
female(neeta).
male(meith).
male(jatan).
male(hitesh).
male(bhavesh).
male(dhishil).
male(prakash).
parent(jatan,hitesh).
parent(jatan,bhavesh).
parent(prakash,manisha).
parent(hitesh,meith).
parent(prakash,sneha).
parent(manisha,meith).
parent(sneha,trushi).
parent(sneha,dhishil).
parent(bhavesh,trushi).
parent(bhavesh,dhishil).
mother(X,Y):- parent(X,Y),female(X).
father(X,Y):- parent(X,Y),male(X).
haschild(X):- parent(X,_).
sister(X,Y):- parent(Z,X),parent(Z,Y),female(X),X\==Y.
brother(X,Y):-parent(Z,X),parent(Z,Y),male(X),X\==Y.
grandparent(X,Y):-parent(X,Z),parent(Z,Y).
grandmother(X,Z):-mother(X,Y),parent(Y,Z).
grandfather(X,Z):-father(X,Y),parent(Y,Z).
wife(X,Y):-parent(X,Z),parent(Y,Z),female(X),male(Y).
uncle(X,Z):-brother(X,Y),parent(Y,Z).
```

NAME: Meith Navlakha
SAP: 60004190068
AI Practical

OUTPUT:



CONCLUSION:

In this experiment, I implemented prolog family tree code on my family tree. Prolog can generate complex relations based on a few basic relations, having its roots in first-order logic. In this experiment, the only information I provided was the gender of the individual and basic relation i.e., who is a parent of whom. Prolog then used this information and allowed us to write code and on complex relations. For example, in the experiment, the father function tells us if the person is the father of other mentioned individual or can also be used to list all the children of the specified person. Similarly for other complex relations like wife or uncle. Thus, prolog allows us to draw complex inferences from simpler relations or logic.