Batch: IAI-2 **Experiment Number: 5**

Roll Number: 16010422234 **Name: Chandana Ramesh Galgali**

Aim of the Experiment: Write a program for implementation of family tree in PROLOG using condition- action rules based agent.

Program/Steps:

This approach follows a table for lookup of condition - action pairs defining all possible condition - action rules necessary to interact in an environment.

Create a family tree program to include following rules –

- M is the mother of P, if she is a parent of P and is female
- F is Father of P, if he is parent of P and is male
- X is Sibling of Y, if they have same parent
- Then add rules for sister, brother, grandfather, grandmother, uncle, aunty, cousins etc (consider 3 generations of your own family and build the family tree)

Based on the facts, define goals to answer questions related to the family tree.

Predicates used in Program:

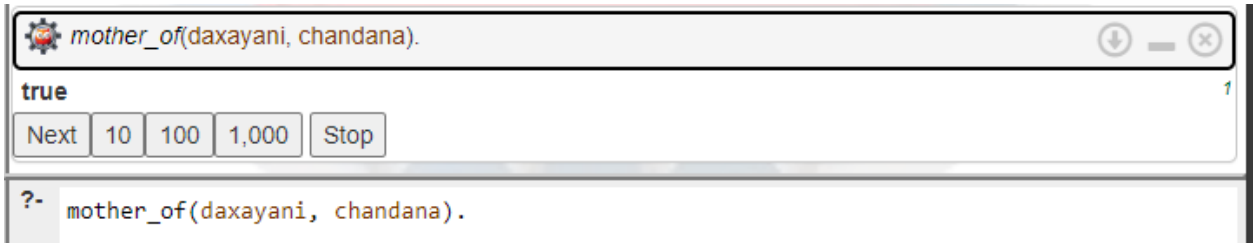
- father(person,person)
- male(person)
- child(person,person)
- female(person)
- mother(person,person)
- spouse(person,person)
- brother(person,person)
- grandparent(person,person)
- uncle(person,person)
- aunt(person,person)
- cousin(person,person)

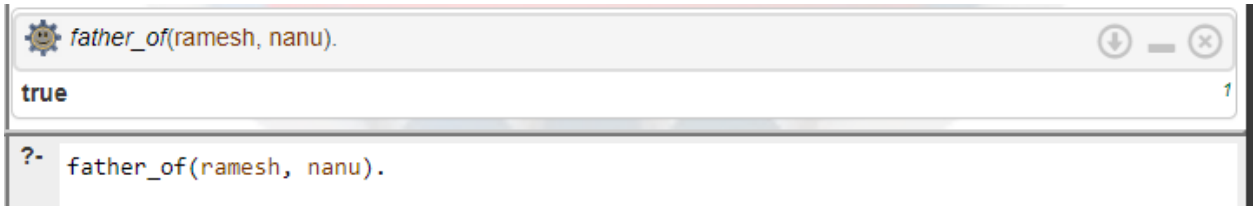
Code:

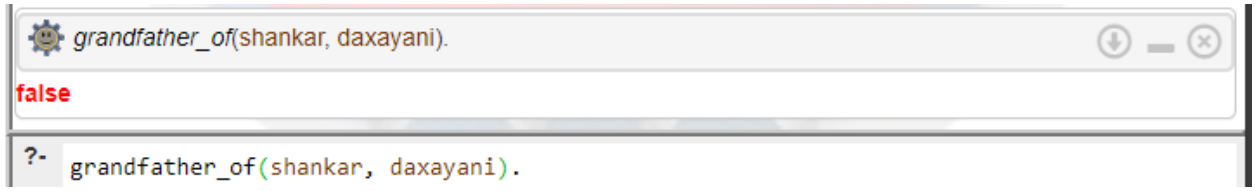
```
/* Facts */
male(ramesh).
male(shankar).
female(daxayani).
female(chandana).
female(nanu).
female(lata).

parent_of(ramesh, chandana).
```

```
parent_of(ramesh, nanu).
parent_of(daxayani, chandana).
parent_of(daxayani, nanu).
parent_of(shankar, daxayani).
parent_of(lata, daxayani).

/* rules */
father_of(X,Y):- male(X),
   parent_of(X,Y).

mother_of(X,Y):- female(X),
   parent_of(X,Y).

grandfather_of(X,Y):- male(X),
   parent_of(X,Z),
   parent_of(Z,Y).

grandmother_of(X,Y):- female(X),
   parent_of(X,Z),
   parent_of(Z,Y).

sister_of(X,Y):- %(X,Y or Y,X)%
   female(X),
   father_of(F, Y), father_of(F,X),X \= Y.

sister_of(X,Y):- female(X),
   mother_of(M, Y), mother_of(M,X),X \= Y.

ancestor_of(X,Y):- parent_of(X,Y).
ancestor_of(X,Y):- parent_of(X,Z),
   ancestor_of(Z,Y).
```

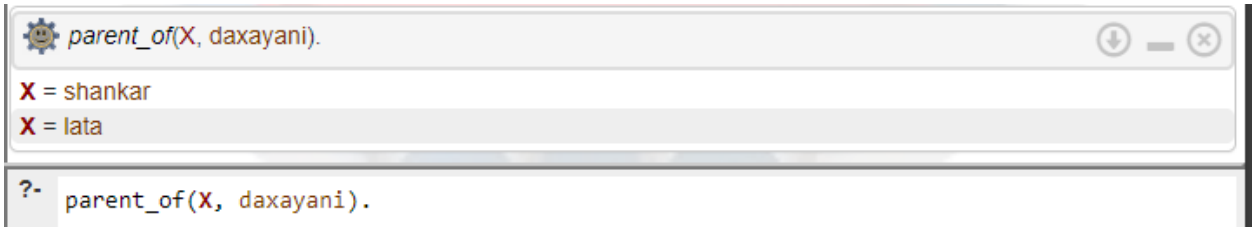
Output/Result:

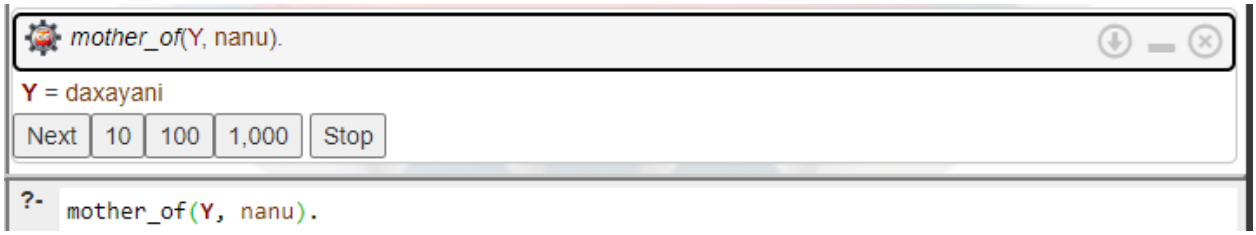



father_of(ramesh, nanu).
true
1
?- father_of(ramesh, nanu).


grandfather_of(shankar, daxayani).
false
?- grandfather_of(shankar, daxayani).


sister_of(chandana, lata).
false
?- sister_of(chandana, lata).


parent_of(X, daxayani).
X = shankar
X = lata
?- parent_of(X, daxayani).


mother_of(Y, nanu).
Y = daxayani
Next
10
100
1,000
Stop
?- mother_of(Y, nanu).


sister_of(X, chandana).
X = nanu
Next
10
100
1,000
Stop
?- sister_of(X, chandana).

Post-Lab Questions:

1. The PROLOG suit is based on

a. Interpreter

b. Compiler

c. None of the above

d. Both

2. State True or False

There must be at least one fact pertaining to each predicate written in the PROLOG program.

Ans: False

3. State True or False

In PROLOG program the variable declaration is a compulsory part.

Ans: False

4. Differentiate between a fact and a predicate with syntax.

Ans: Fact: A fact in PROLOG represents a statement that is known to be true. It consists of a predicate followed by a list of arguments enclosed in parentheses.

Example:

```
likes(john, pizza).
```

Predicate: A predicate in PROLOG is a rule or a clause that defines a relation or property. It consists of a predicate name followed by a list of parameters (variables or constants) enclosed in parentheses, optionally followed by a body that specifies conditions or actions.

Example:

```
happy(X) :- likes(X, pizza).
```

5. Differentiate between knowledge base and Rule base approach.

Ans: Knowledge Base: In the knowledge base approach, information is stored as a collection of facts and rules. Facts represent basic assertions about the world, while rules define relationships or logical implications. Queries are made against the knowledge base to retrieve information or infer new facts.

Rule Base: In the rule-based approach, the system operates by applying a set of rules to the input data or current state to derive conclusions or make decisions. Rules are typically in the form of condition-action pairs, where if certain conditions are met, corresponding actions are triggered. This approach is commonly used in expert systems and inference engines.

Outcomes: Ability to formally state the problem and develop the appropriate proof for a given logical deduction problem.

Conclusion (Based on the Results and outcomes achieved):

The experiment successfully demonstrated the feasibility and benefits of implementing a family tree in PROLOG using a condition-action rules-based agent. This project not only contributed to understanding the practical applications of logic programming in representing complex relational data but also opened avenues for further research in optimizing and expanding such systems. The findings underscore the potential of PROLOG and condition-action rules in developing intelligent systems that require efficient data representation and reasoning capabilities.

References:

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Second Edition, Pearson Publication
2. Luger, George F. Artificial Intelligence : Structures and strategies for complex problem solving, 2009, 6th Edition, Pearson Education
3. https://www.101computing.net/prolog-family-tree/
