

Batch: SY IT A2**Experiment Number: 5****Roll Number: 16010421063****Name: Arya Nair**

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

Program/ Steps:

```
/* Program For Family Tree */
```

```
male(jack).
```

```
male(oliver).
```

```
male(ali).
```

```
male(james).
```

```
male(simon).
```

```
male(harry).
```

```
female(helen).
```

```
female(sophie).
```

```
female(jess).
```

```
female(lily).
```

```
parent_of(jack,jess).
```

```
parent_of(jack,lily).
```

```
parent_of(helen, jess).
```

```
parent_of(helen, lily).
```

```
parent_of(oliver,james).
```

```
parent_of(sophie, james).
```

```
parent_of(jess, simon).
```

```
parent_of(ali, simon).
```

```
parent_of(lily, harry).
```

```
parent_of(james, harry).
```

```
/* 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.
```

```
aunt_of(X,Y):- female(X),  
    parent_of(Z,Y), sister_of(Z,X),!.
```

```
brother_of(X,Y):- %(X,Y or Y,X)%  
    male(X),  
    father_of(F, Y), father_of(F,X),X \= Y.
```

```
brother_of(X,Y):- male(X),  
    mother_of(M, Y), mother_of(M,X),X \= Y.
```

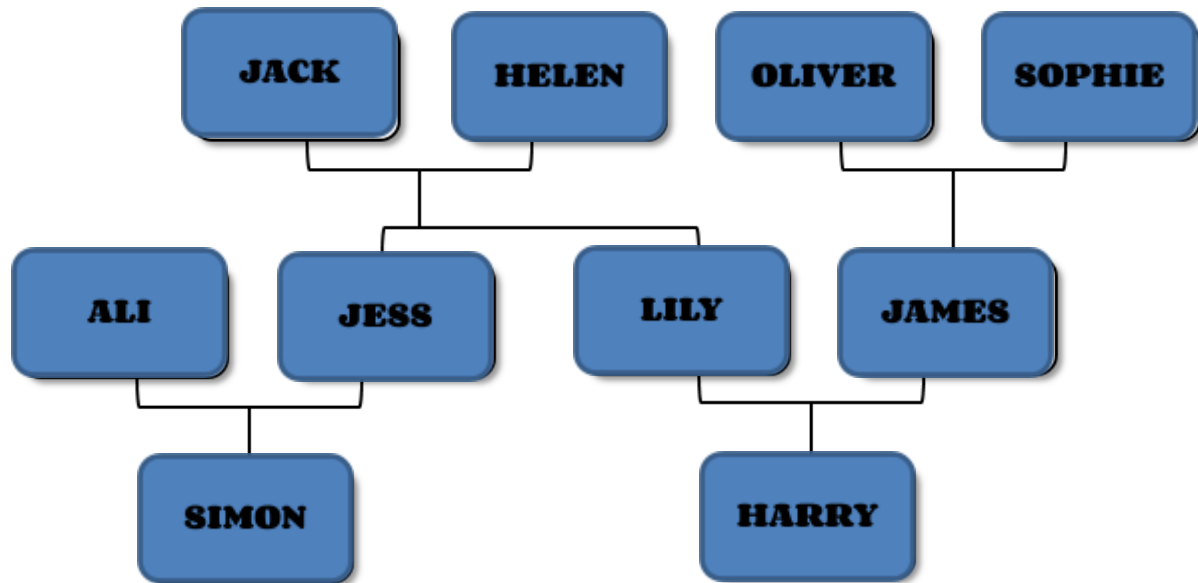
```
uncle_of(X,Y):-  
    parent_of(Z,Y), brother_of(Z,X).
```

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

Output/Result:

 <code>father_of(X,jess).</code>	  
<code>X = jack</code>	
 <code>mother_of(X,james).</code>	  
<code>X = sophie</code>	
 <code>grandmother_of(X,harry).</code>	  
<code>X = helen</code>	
 <code>grandfather_of(X,simon).</code>	  
<code>X = jack</code>	
 <code>mother_of(X,harry).</code>	  
<code>X = lily</code>	
 <code>ancestor_of(X,ali).</code>	  
<code>false</code>	

Family Tree:



Post Lab Question-Answers:

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.

True

3. State true or false

In PROLOG program the variable declaration is a compulsory part.

True

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

Fact	Predicate
Fact in prolog is substitution of predicate like table in db Table(Column1,Column2, ...) indeed Facts takes the form like Fact(Arg1,Arg2) which gives us {true,false} values ONLY for the specific constants mentioned inside "()"	A predicate denotes a property or relationship between objects. Definitions consist of clauses. A clause has a head and a body (Rule) or just a head (Fact). A head consists of a predicate name and arguments.
Syntax: parent_of(jack,jess).	Syntax: mother_of(X,Y):- female(X), parent_of(X,Y).

5. Differentiate between knowledge base and Rule base approach.

- A rule-based system is one that is built on a set of rules that have been created and that are followed by everyone. The rules are given to the user and the user is only allowed to complete the given tasks with the rules.

- A knowledge-based system is one that is built on the idea that there is no set of rules that are followed; it is one that is built on knowledge that has been obtained in order to make decisions. The knowledge-based system is one in which the context of the user's situation is taken into account and the user is given the information needed to complete the task.
-

Outcomes:

CO3: Ability to formally state the problem and develop the appropriate proof for given a logical deduction problem.

Conclusion (based on the Results and outcomes achieved):

So in this experiment we have successfully implemented a family tree in PROLOG using condition- action rules based agent and executed queries in the engine.

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/>