SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJ22ITL501 **DATE:**

COURSE NAME: Artificial Intelligence Laboratory **CLASS: TY-IT**

EXPERIMENT NO.08

CO/LO: Formulate the problem as a state space and select appropriate technique from blind, heuristic or adversarial search to generate the solution.

AIM / OBJECTIVE: Implement a suitable case study using PROLOG.

DESCRIPTION OF EXPERIMENT:

Prolog is to define a set of rules that describe the relationships between different objects or concepts in your problem domain. For example, you might define rules that specify that certain objects are bigger than others, or that some objects are the same color. Then, you can use Prolog to ask questions about these objects and their relationships, and the interpreter will use your rules to deduce the answers.

To use Prolog, you will need to have a Prolog interpreter installed on your computer. There are several different Prolog interpreters available, including SWI-Prolog, GNU Prolog and B-Prolog. Once you've installed an interpreter, you can start writing Prolog programs using a text editor and then run them using the interpreter.

Prolog is a powerful and flexible programming language that's well-suited for developing logicbased artificial intelligence applications. It allows the programmer to specify a set of rules and facts about a problem domain, and then use those rules and facts to automatically infer solutions to problems.

There is no single "syntax" for Prolog, as the language allows for a wide range of different programming styles and approaches. However, here are some basic elements of Prolog syntax that are commonly used:

- Facts are statements that are assumed to be true. In Prolog, facts are written using a predicate name followed by a list of arguments enclosed in parentheses. For example: man(john).
- Rules are logical statements that describe the relationships between different facts. In Prolog, rules are written using the predicate name followed by a list of arguments enclosed in parentheses, followed by a colon and a hyphen (:-) and the body of the rule.
  For example: not(X,Y) :- man(X), woman(Y).

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

- Variables are used to represent values that can change or be determined by the interpreter. In Prolog, variables are written using a name that begins with an uppercase letter. For example: X.
- Queries are used to ask the interpreter to find solutions to problems based on the rules and facts in the program. In Prolog, queries are written using the same syntax as facts followed by a question mark (?). For example: not(john, mary)?

EXPLANATION / SOLUTIONS (DESIGN):

Code:

simple Prolog program might look like this:

```
man(john).
woman(mary).
capital_of(france, paris).

not(X,Y) :- man(X), woman(Y).
```

Output:

The first three lines are facts, while the fourth line is a rule. The rule uses the not/2 predicate to state that if X is a man and Y is a woman, then X is not Y.

Example:

Example 1: Below food table shows the facts, rules, goals and their english meanings.

Facts	**English meanings**
food(burger).	// burger is a food
food(sandwich).	// sandwich is a food
food(pizza).	// pizza is a food
lunch(sandwich).	// sandwich is a lunch
dinner(pizza).	// pizza is a dinner
Rules	
meal(X) :- food(X).	// Every food is a meal OR Anything is a meal if it is a food
Queries / Goals	
?- food(pizza).	// Is pizza a food?
?- meal(X), lunch(X).	// Which food is meal and lunch?

?- dinner(sandwich). // Is sandwich a dinner?

Example 2 : Below student-professor relation table shows the facts, rules, goals and their english meanings.

Facts	**English meanings**
studies(charlie, csc135).	// charlie studies csc135
studies(olivia, csc135).	// olivia studies csc135
studies(jack, csc131).	// jack studies csc131
studies(arthur, csc134).	// arthur studies csc134
teaches(kirke, csc135).	// kirke teaches csc135
teaches(collins, csc131).	// collins teaches csc131
teaches(collins, csc171).	// collins teaches csc171
teaches(juniper, csc134).	// juniper teaches csc134

Rules professor(X, Y) :-
teaches(X, C), studies(Y, C). // X is a professor of Y if X teaches C and Y studies C.

Queries / Goals

?- studies(charlie, What). // charlie studies what? OR What does charlie study?

?- professor(kirke, Students). // Who are the students of professor kirke.

Example 4: holi. cat(bengal). /* bengal is a cat */ dog(rottweiler). /* rottweiler is a dog */ likes(Jolie, Kevin). /* Jolie likes Kevin */ likes(A, Kevin). /* Everyone likes Kevin */ likes(Jolie, B). /* Jolie likes everybody */
likes(B, Jolie), likes(Jolie, B). /* Everybody likes Jolie and Jolie likes everybody */
likes(Jolie, Kevin); likes(Jolie, Ray). /* Jolie likes Kevin or Jolie likes Ray */
not(likes(Jolie, pasta)). /* Jolie does not like pasta */ **Output:**

```
?- 'It is sunny'.
yes
?- 'It is cold'.
no
 ?-
```

Program Design:

Code:

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

1- First create a source file for the genealogical logicbase application. Start by adding a few members of your family tree. It is important to be accurate, since we will be exploring family relationships. Your own knowledg of who your relatives are will verify the correctness of your Prolog programs.

2- Enter a two-argument predicate that records the parent-child relationship. One argument represents the parent, and the other the child. It doesn't matter in which order you enter the arguments, as long as you are consistent. Often Prolog programmers adopt the convention that parent(A,B) is interpreted "A is the parent of B".

3- Create a source file for the customer order entry program. We will begin it with three record types (predicates). The first is customer/3 where the three arguments are

1)

```
% Facts: Define family relationships
parent(john, mary).
parent(john, tom).
parent(mary, alice).
parent(tom, susan).

% Rules: Define other relationships
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).
ancestor(X, Y) :- parent(X, Y).
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).
```

2)

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

```
% Facts: Define customers, orders, and items
customer(john_doe, '123 Main St', 'john@example.com').
customer(jane_smith, '456 Elm St', 'jane@example.com').

order(order_1, john_doe, '2024-11-14').
order(order_2, jane_smith, '2024-11-15').

item(order_1, 'phone case', 19.99).
item(order_1, 'charger', 9.99).
item(order_2, 'tablet', 199.99).

% Rules: Define relationships or calculations, such as total order cost
total_cost(Order, Total) :-
    findall(Price, item(Order, _, Price), Prices),
    sum_list(Prices, Total).
```

3)

```
% Define parent-child relationships
parent(john, mary).
parent(john, tom).
parent(mary, alice).
parent(mary, bob).
parent(tom, susan).
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Output:

1)

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

2)

sibling(mary, tom).

true

grandparent(john, alice).

true

Next | 10 | 100 | 1,000 | Stop

ancestor(john, susan).

true

Next | 10 | 100 | 1,000 | Stop

customer(john_doe, Address, Email).

Address = '123 Main St',
Email = 'john@example.com'

order(OrderID, john_doe, Date).

Date = '2024-11-14',
OrderID = order_1

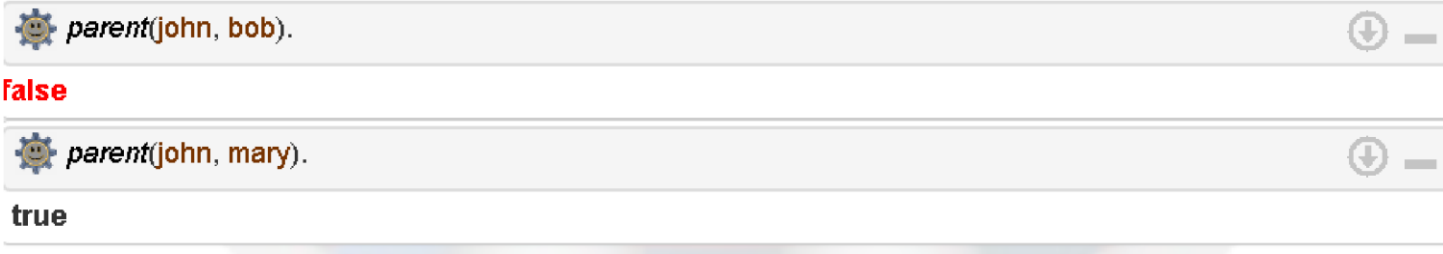
item(order_1, Product, Price).

Price = 19.99,
Product = 'phone case'

total_cost(order_1, Total).

Total = 29.979999999999997

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)



3)

Questions:

1. Ram likes apple.
2. Ram is taller then Mohan.
3. My name is Subodh.
4. Apple is fruit.
5. Orange is fruit.
6. Ram is male.

Write simple queries for following facts.

1)likes(ram, apple).
2)taller_than(ram, mohan).
3)name(myself, subodh).
4)fruit(apple).
5)fruit(orange).
6)male(ram).

CONCLUSION: We implemented facts and rules using prolog .

REFERENCES:

[1] Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", 2nd Edition, Pearson Education, 2010

[2] https://silp.iiita.ac.in/wp-content/uploads/PROLOG.pdf

[3] https://swish.swi-prolog.org/