EXPERIMENT NO :-01

Aim of Experiment :- Study of PROLOG Programming language and its Functions. Write simple facts for the statements using PROLOG.

Theory:- Prolog (short for "Programming in Logic") is a high-level programming language associated with artificial intelligence and computational linguistics.

- It is a declarative language, which means that the logic of the computation is expressed without describing its control flow. Prolog is particularly well-suited for tasks that involve pattern matching, tree-based data structures, and automatic backtracking.
- Prolog is a declarative language that means we can specify what problem we want to solve rather than how to solve it.
- Prolog is used in some areas like database, natural language processing, artificial intelligence, but it is pretty useless in some areas like a numerical algorithm or instance graphics.

Features of PROLOG :-

- PROLOG is a declarative programming language.
- It employs predicate calculus terminology.
- PROLOG is a natural handler of lists and recursions.
- This language has a built-in inference engine as well as automated backtracking.
- PROLOG has built-in parallelism.
- Unification: The main concept is to see if the phrases can be combined to reflect the same structure.
- For problems requiring inference, PROLOG allows for very efficient coding.

Some Applications of Prolog :-

Prolog is used in various domains. It plays a vital role in automation system. Following are some other important fields where Prolog is used −

- Intelligent Database Retrieval
- Natural Language Understanding
- Specification Language
- Machine Learning
- Robot Planning
- Automation System
- Problem Solving

PROLOG (PROgramming in LOGic) is a high-level programming language primarily associated with artificial intelligence and computational linguistics. It is declarative, meaning that you specify *what* we want done rather than *how* to do it.

At its core, PROLOG operates on facts, rules, and queries:

1. **Facts**: Base truths in the knowledge base.
2. **Rules**: Logical inferences derived from facts.
3. **Queries**: Questions asked to the PROLOG system to infer or retrieve data.

Example: Writing Simple Facts in PROLOG

1. **Facts about Family Relationships**

```
% Defining facts about family members.
parent(john, mary).   % John is a parent of Mary.
parent(mary, susan).  % Mary is a parent of Susan.
parent(john, michael). % John is a parent of Michael.

% Defining gender.
male(john).
male(michael).
female(mary).
female(susan).
```

2. Queries

- Who are John's children?

```
?- parent(john, X).
% X = mary ;
% X = michael.
```

- Who is Susan's parent?

```
?- parent(X, susan).
% X = mary.
```

3. Rules Using Facts

Rules allow us to derive relationships. For example:

```
% Define a grandparent rule.
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

% Define a sibling rule.
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
```

Functions in PROLOG :-

Though PROLOG doesn't have functions in the traditional sense, we can simulate them using rules to perform calculations or reasoning:

Example: Arithmetic in PROLOG

```
% Define a rule to add two numbers.
sum(X, Y, Z) :- Z is X + Y.

% Query to find the sum of 3 and 5.
?- sum(3, 5, Result).
% Result = 8.
```

Conclusion :-

PROLOG is a powerful tool for representing and querying logical relationships. By defining facts and rules, We can effectively model and infer knowledge in a domain of interest. This practical understanding forms the basis of more complex applications like expert systems, natural language processing, and decision-making tools.