

Experiment No. 2

1. Aim : Write a simple program using PROLOG as an AI Programming Language.

2. Objectives : Study of PROLOG

3. Theory /Description: PROLOG stands for Programming In Logic. It is language for symbolic, non-numeric computation. It is specially well suited for solving problems that involve objects and relation between objects. 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.

For example, it is an easy exercise in PROLOG to express spatial relationship between objects, such as the blue sphere is behind the green one. It is also easy to state a more general rule: if the object X is closer to the observer than object Y, and object Y is closer than Z, then X must be closer than Z. PROLOG can result about the spatial relationships and their consistency with respect to the general rule. Features like this makes the PROLOG a powerful language for Artificial Language AI, and non-numerical programming.

Prolog program represents a different type of relationship or property. The program uses simple and concise syntax to encode various forms of knowledge, such as identity, liking, possession, characteristics, categorization, and comparison. Prolog's logic-based nature allows it to represent and reason about this knowledge in a coherent and declarative manner, making it well-suited for knowledge representation in artificial intelligence.

Example :

The provided PROLOG program demonstrates knowledge representation using facts. PROLOG is an AI programming language well-suited for representing and reasoning about knowledge in a logical and declarative manner. In this theory, we will explain each line of the program and how it represents knowledge about various relationships.

Example:

1-seema is a girl.

2-bill likes candy.

3-John owns Candy.

4-rose is red.

5-apple is fruit.

6-Ram is taller than mohan.

1. girl(seema).

- This fact asserts that "Seema" is a girl.
- It represents a simple relationship between the term "seema" and the concept of being a girl.

2. likes(bill, candy).

- This fact states that "Bill" likes "candy."
- It establishes a liking relationship between the individual "Bill" and the object "candy."

3. owns(john, gold).

- This fact indicates that "John" owns "gold."
- It establishes a possession relationship between the person "John" and the object "gold."

4. red(rose).

- This fact describes that the "rose" is red.
- It represents a characteristic relationship, implying that the "rose" possesses the quality of being red.

5. fruit(apple).

- This fact categorizes "apple" as a fruit.
- It establishes a classification relationship, classifying the term "apple" as a member of the category "fruit."

6. taller(ram, mohan).

- This fact indicates that "Ram" is taller than "Mohan."
- It establishes a comparison relationship between the two individuals "Ram" and "Mohan" based on their heights.

4. Program:

1. likes(bill,candy).
2. seema(girl).
3. color(rose,red).
4. apple(fruit).aaaaaa
5. taller(ram,mohan).
6. owns_gold(john).

5. Output:

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (64 bits)
Compiled Jul  8 2021, 12:22:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/admin/Downloads/EXP3.pl for byte code...
C:/Users/admin/Downloads/EXP3.pl compiled, 6 lines read - 918 bytes written, 5 ms
| ?- seema(Who).

Who = girl

yes
| ?- likes(bill,What).

What = candy

yes
| ?- color(rose,what).

no
| ?- color(rose,What).

What = red

yes
| ?- apple(What).

What = fruit

yes
| ?- taller(Who,mohan).

Who = ram

yes
| ?- owns_gold(Who).

Who = john

yes
| ?- |
```

6. Conclusion: In conclusion, the experiment illustrates how PROLOG, with its declarative style, is ideal for knowledge representation. The program efficiently captures relationships and facts, allowing logical inference for decision-making. PROLOG's simplicity and querying prowess make it a powerful AI tool for building knowledge-based applications.