

## 5.5 INTRODUCTION TO LOGIC AND LOGIC PROGRAMMING

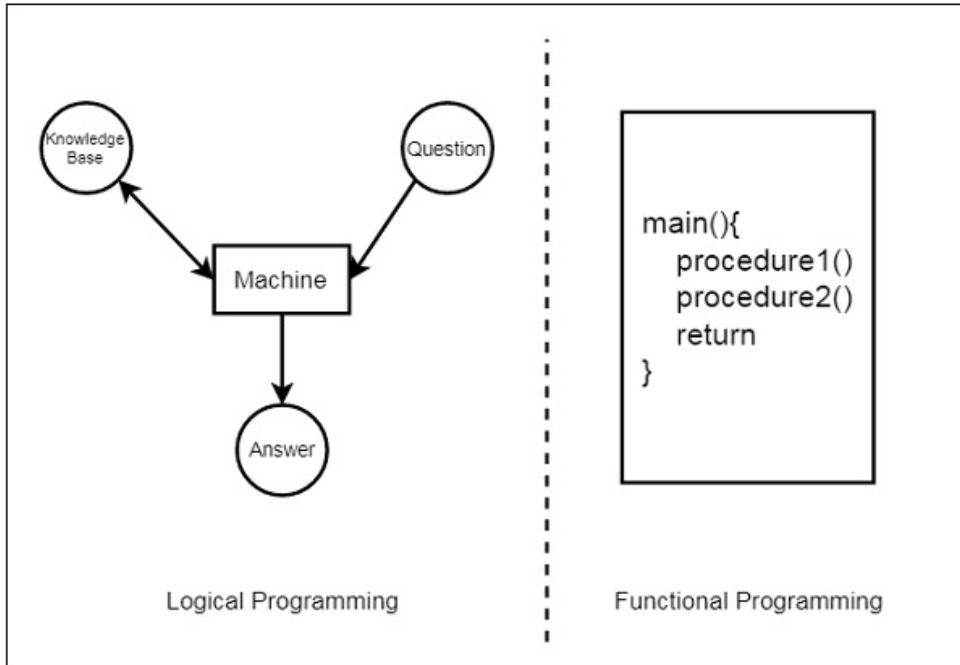
Prolog as the name itself suggests, is the short form of LOGical PROgramming. It is a logical and declarative programming language. Before diving deep into the concepts of Prolog, let us first understand what exactly logical programming is.

Logic Programming is one of the Computer Programming Paradigm, in which the program statements express the facts and rules about different problems within a system of formal logic. Here, the rules are written in the form of logical clauses, where head and body are present. For example, His head and B1, B2, B3 are the elements of the body. Now if we state that “H is true, when B1, B2, B3 all are true”, this is a rule. On the other hand, facts are like the rules, but without any body. So, an example of fact is “H is true”.

Some logic programming languages like Datalog or ASP (Answer Set Programming) are known as purely declarative languages. These languages allow statements about what the program should accomplish. There is no such step-by-step instruction on how to perform the task. However, other languages like Prolog, have declarative and also imperative properties. This may also include procedural statements like “To solve the problem H, perform B1, B2 and B3”.

Some logic programming languages are given below –

- ALF (algebraic logic functional programming language).
- ASP (Answer Set Programming)
- CycL
- Datalog
- FuzzyCLIPS
- Janus
- Parlog
- Prolog
- Prolog++
- ROOP



## 5.6 PROGRAMMING WITH PROLOG

Prolog or PROgramming in LOGics is a logical and declarative programming language. It is one major example of the fourth generation language that supports the declarative programming paradigm. This is particularly suitable for programs that involve symbolic or non-numeric computation. This is the main reason to use Prolog as the programming language in Artificial Intelligence, where symbol manipulation and inference manipulation are the fundamental tasks.

In Prolog, we need not mention the way how one problem can be solved, we just need to mention what the problem is, so that Prolog automatically solves it. However, in Prolog we are supposed to give clues as the solution method.

**Prolog language basically has three different elements –**

**Facts** – The fact is predicate that is true, for example, if we say, “Tom is the son of Jack”, then this is a fact.

Rules – Rules are extensions of facts that contain conditional clauses. To satisfy a rule these conditions should be met. For example, if we define a rule as – grandfather(X, Y) :- father(X, Z), parent(Z, Y)

This implies that for X to be the grandfather of Y, Z should be a parent of Y and X should be father of Z.

The basic elements of Prolog include:

### 1. Facts:

In Prolog, facts are statements about the world that are assumed to be true. Facts consist of predicates and arguments. Predicates represent relationships or properties, and arguments provide the specific details.

Facts are declared using the syntax:

```
predicate(argument1, argument2, ...).
```

For example:

```
parent(sanjay, maya).
```

### 2. Rules:

Rules in Prolog define relationships or conditions based on other facts or rules. They are used to derive new information or make logical inferences. Rules consist of a head (the conclusion) and a body (the condition). The head specifies the result or conclusion, while the body contains the conditions or requirements for the rule to be true.

Rules are declared using the syntax:

```
head :- condition1, condition2, ...
```

For example:

```
ancestor(X, Y) :- parent(X, Y).
```

### 3. Queries:

In Prolog, you can ask queries to the system to retrieve information or test relationships. Queries are entered in the form of goals, which are logical statements that you want to prove or find solutions for. You can ask queries using the syntax:

```
?- goal.
```

For example:

```
?- ancestor(sanjay, maya).
```

### 4. Variables:

Variables are used in Prolog to represent unknown values or placeholders. They start with an uppercase letter or an underscore (`_`) and can be used to instantiate values during the execution of a query or rule. Variables allow Prolog to find solutions and perform unification, which is the process of matching values. For example:

```
?- parent(X, maya).
```

Here, `X` is a variable, and Prolog will attempt to find a value for `X` that satisfies the query.

### 5. Logical Operators:

Prolog provides logical operators to combine conditions and goals.

The main logical operators in Prolog are:

Conjunction (`,`): Represents logical AND. It requires both conditions to be true.

Disjunction (`;`): Represents logical OR. It satisfies the goal if either condition is true.

Negation (`not` or `+`): Represents logical NOT. It negates a condition, making it false.