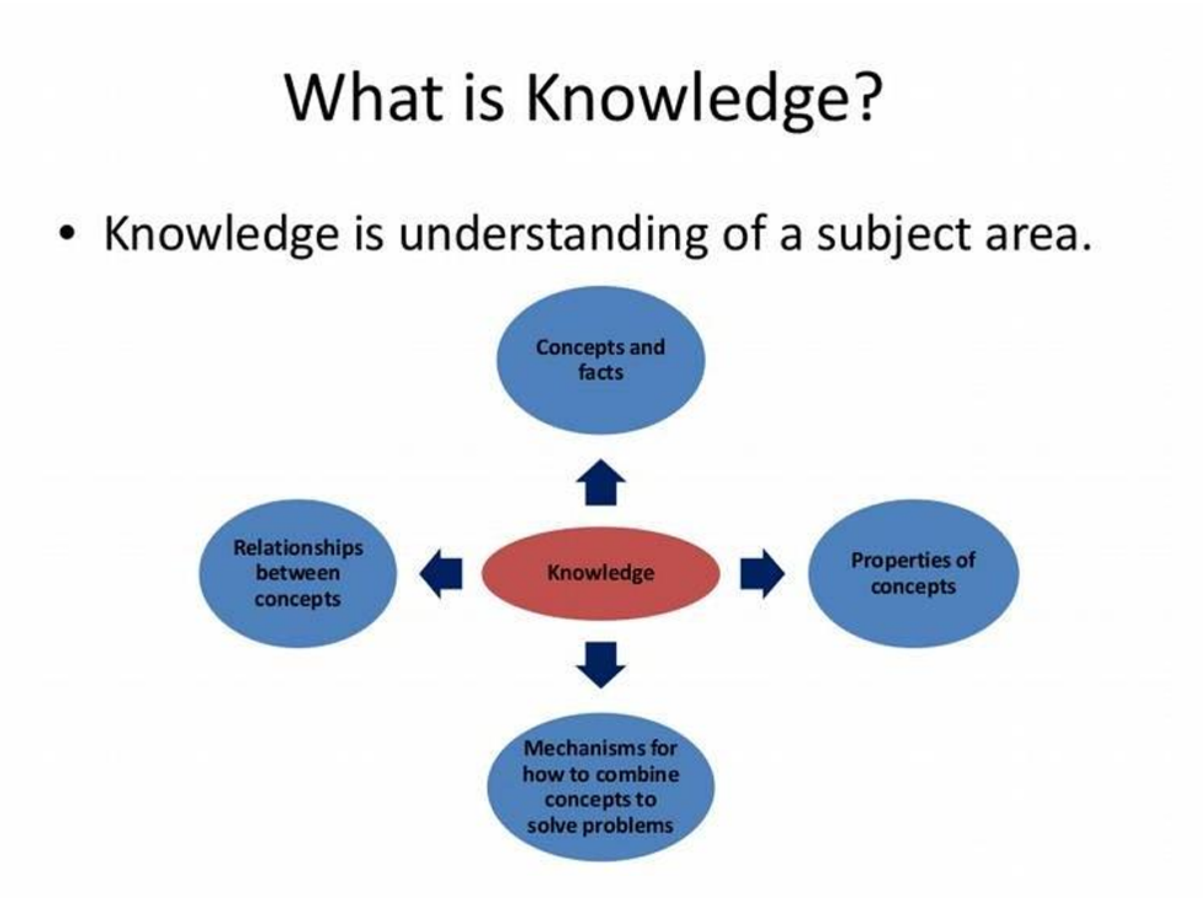
Unit 4 : Knowledge, Logic and Programming Language (Milan)



What is Knowledge?

- Knowledge is understanding of a subject area.

Concepts and facts

Relationships between concepts

Knowledge

Properties of concepts

Mechanisms for how to combine concepts to solve problems

Why do we need Knowledge Representation?

- Unlike human mind, computers cannot acquire and represent knowledge by themselves.
- It is complicated to machine process a knowledge represented in natural language.
- Human knowledge is of different types.
- Knowledge manipulation involves:
  - Knowledge acquisition: gathering, structuring and organizing knowledge.
  - Knowledge storing: putting the knowledge into computer.

- Knowledge retrieval: getting the knowledge when needed.
- Reasoning: gives conclusion, inference or explanation.

Knowledge

1. **Declarative Knowledge**

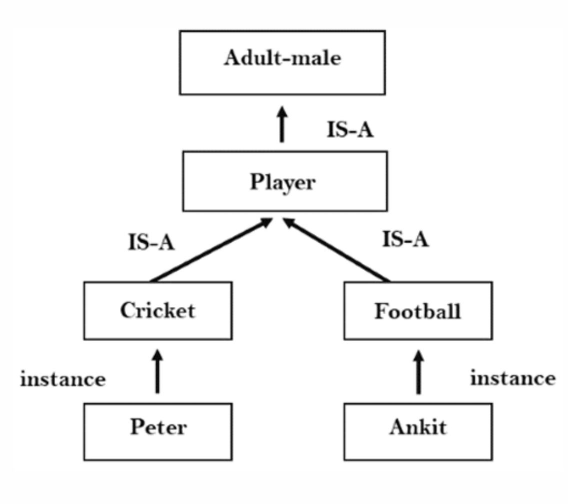
- It gives the simple facts about any organization or phenomenon.
- Declarative knowledge means representation of facts.
- The facts may be static facts or dynamic facts.
  - **Static Facts**: Do not change with time, e.g., in the college, location is permanent.
  - **Dynamic Facts**: Change with time, e.g., new courses may be added in the curriculum.

2. **Procedural Knowledge**

- The declarative knowledge does not tell anything regarding the functioning of the concerned object.
  - E.g., It does not tell how a student is examined, how the syllabus is framed, how fees deposit is made, etc.
- The procedural knowledge represents the functioning of an organization.
- It describes dynamic attributes using production rules.
  - **Example**:
    - **If**:
      - student has deposited fees
      - student has opted for a course
      - student has attended 90% of classes
      - student has passed the examination
    - **Then**: declare the student pass

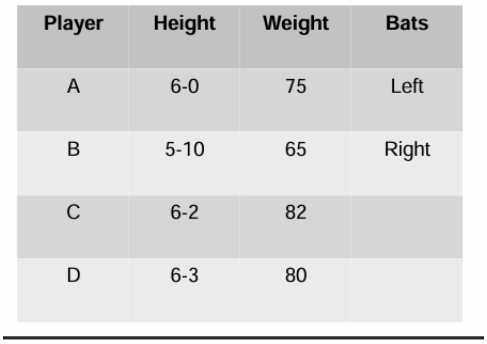
3. **Inheritable Knowledge**

- There are many situations where general concepts regarding some event, thing, or activity are already known. An object of that particular type inherits all features of that event.
    - **Example**:
        - **College**: A college has some features like classrooms, teachers, a playground, building, students, etc.
        - If we say ‘x’ is a college, then 'X' will automatically inherit all the features of the college.
        - It may be possible that 'X' has some additional features.
        - Here, the relationship ‘has’ indicates the salient features, and ‘is a’ represents the variable.



4. **Relational Knowledge**

- In this type of knowledge, the facts are represented as a set of relations in a tabular form.
- It can be used to answer simple questions like “Who is the tallest boy?”



Player	Height	Weight	Bats
A	6-0	75	Left
B	5-10	65	Right
C	6-2	82	
D	6-3	80	

5. **Heuristic Knowledge**

- This type of knowledge can be defined as experimental, rarely discussed knowledge.
- Knowledge of good guessing is heuristic knowledge. Such type of knowledge cannot be acquired from books; rather, it comes from within the individual and differs from individual to individual.
    - **Example**: How many runs the Indian Cricket team would score in a particular one-day international match against Australia.
- Heuristic knowledge represents the knowledge of some experts in a field or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

6. **Inferential Knowledge**

- This method uses an inference mechanism to use this knowledge.
- Predicate logic representation is also used to represent inferential knowledge.
- The inference procedures implement the standard logic rules of inference.

7. **Common Sense Knowledge**

- It is domain-independent knowledge.
- It is gained by our experience.
    - **Example**: Regarding the inheritable knowledge of college, additional knowledge like the concept of preliminary school education, a general idea about the concept of education, etc.

- A person gains this knowledge, knowingly or unknowingly, throughout their life.
- It is the knowledge that is most difficult to represent and code.

8. **Explicit Knowledge**

- Explicit knowledge is the one which an individual holds explicitly (clear and conscious knowledge).
- This knowledge can be expressed clearly into formal language, including mathematical expressions, grammatical statements, specifications, manuals, etc.

9. **Tacit Knowledge**

- It is understood or not expressed in any conventional form.
- It is the form of knowledge an individual possesses about which they may or may not be aware.
- This kind of knowledge is acquired by experience and involves intangible factors such as personal beliefs, perspective, and the value system.

10. **Uncertain Knowledge**

- There is one more property of knowledge, that is, it is uncertain and usually incomplete.
- What we provide is the information that is known to us. We provide the knowledge complete to the best of our capacity, but it is never absolutely complete.
- Real-world phenomena are highly uncertain.
- The kind of knowledge required to represent this is uncertain knowledge.

Techniques of Knowledge Representation

There are mainly four ways of knowledge representation which are given as follows:

1. **Logical Representation**

2. **Semantic Network Representation**
3. **Frame Representation**
4. **Production Rules**

1. Logical Representation

Logical representation is a language with some concrete rules which deals with propositions and has no ambiguity in representation. Logical representation means drawing a conclusion based on various conditions. Each sentence can be translated into logics using **syntax** and **semantics**. Logical representation can be categorized into mainly two logics:

1. **Propositional Logics**
2. **Predicate Logics**

Syntax

- Syntaxes are the rules which decide how we can construct legal sentences in the logic.
- It determines which symbols we can use in knowledge representation.
- How to write those symbols.

Semantics

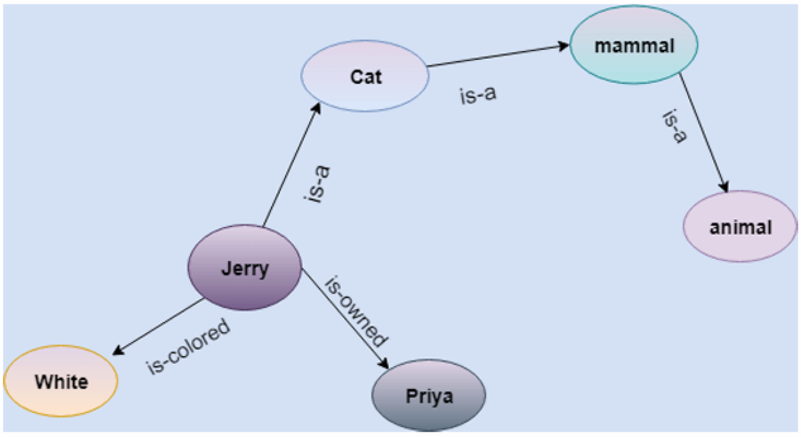
- Semantics are the rules by which we can interpret the sentence in the logic.
- Semantics also involves assigning a meaning to each sentence.

2. Semantic Network Representation

- Semantic networks are an alternative to predicate logic for knowledge representation.
- In Semantic networks, we can represent our knowledge in the form of graphical networks.
- This network consists of nodes representing objects and arcs which describe the relationship between those objects.

Statements:

- Jerry is a cat.
- Jerry is a mammal.
- Jerry is owned by Priya.
- Jerry is brown colored.
- All Mammals are animals.



3. Frame Representation

- A frame is a record-like structure that consists of a collection of attributes and its values to describe an entity in the world.
- Frames are the AI data structure that divides knowledge into substructures by representing stereotypes situations.
- Slots have names and values which are called facets.

Slots:

- **Title:** Artificial Intelligence
- **Genre:** Computer Science
- **Author:** Peter Norvig
- **Edition:** Third Edition
- **Year:** 1996
- **Page:** 1152

4. Production Rules

- Production rules system consists of (condition, action) pairs which mean, "If condition then action". It has mainly three parts:
  1. **The set of production rules**
  2. **Working Memory**
  3. **The recognize-act-cycle**
- In production rules, the agent checks for the condition, and if the condition exists, then the production rule fires and the corresponding action is carried out.
- The condition part of the rule determines which rule may be applied to a problem. The action part carries out the associated problem-solving steps. This complete process is called a **recognize-act cycle**.

Example:

- IF (at bus stop AND bus arrives) THEN action (get into the bus)
- IF (on the bus AND paid AND empty seat) THEN action (sit down).
- IF (on bus AND unpaid) THEN action (pay charges).
- IF (bus arrives at destination) THEN action (get down from the bus).

Basic Components of Knowledge

1. **Set of Data**
2. **A Form of Belief or Hypothesis**
3. **Kind of Information**

As Set of Data

- Knowledge is different from data.

- **Data** is the raw form of observations.
- **Knowledge** is the organized form of data and procedures which can be used for some useful purposes.

Example: Physician treating a patient.

As Form of Belief or Hypothesis

- Knowledge is different from belief and hypothesis.
- **Belief** is any meaningful coherent expression that can be expressed.
    - Belief may be true or false.
- **Hypothesis** is a belief that is backed with some supporting evidence but may still be false.
- **Knowledge** is true justified belief.

As Kind of Information

- **Information** is data plus meaning of the same.
- When information is capable of creating more information and can become part of some action, it falls in the category of knowledge.
- **Knowledge** is information about objects, concepts, and relationships that are assumed to exist in a particular area of interest.
- Knowledge is a richer, structured, and more contextual form of information that is required to perform the task of problem-solving.

What is Knowledge?

Difference between Data, Information, and Knowledge:

- **Data:** Primitive verifiable facts.
    - **Example:** Name of novels available in a library.
- **Information:** Analyzed data.
    - **Example:** The novel that is frequently asked by the members of the library is "Harry Potter and the Chamber of Secrets".

- **Knowledge:** Analyzed information that is often used for further information deduction.
    - **Example:** Since the librarian knows the name of the novel that is frequently asked by members, they will ask for more copies of the novel the next time they place an order.
    - Knowledge is richer, structured, and more contextual form of information that is required to perform the task of problem-solving.
    - Knowledge includes:
        - Understand knowledge
        - Use knowledge for decision making
        - Recognize objects through vision
        - Interpret situations
        - Plan strategies

Logic

1. Propositional logic

- Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions.
- A proposition is a declarative statement which is either true or false.
- It is a technique of knowledge representation in logical and mathematical form.

Example:

1. New Delhi is a capital of India.
2. The square root of 4 is 2.
3. India will be superpower by 2030.
4. No, thank you.

- Sentence 1 and 2 are propositions. They both are true.
- Sentence 3 cannot be said as 'true' or 'false', but not both.

- Sentence 4 is an assertion, so we cannot assign true or false with it.

Following are some basic facts about propositional logic:

- Propositional logic is also called Boolean logic as it works on 0 and 1.
- In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for representing a proposition, such as A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but they cannot be both.
- Propositional logic consists of objects, relations or functions, and logical connectives.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of propositional logic.
- Connectives can be said to be logical operators that connect two sentences.
- A proposition formula that is always true is called **tautology**, and it is also called a valid sentence.
- A proposition formula that is always false is called **contradiction**.
- Statements that are questions, commands, or opinions are not propositions, such as "Where is Rohini", "How are you", "What is your name", which are not propositions.

Syntax of Propositional Calculus

- **Atomic Propositions**
- **Compound propositions**

Atomic Proposition:

- Atomic propositions are the simple propositions.
- It consists of a single proposition symbol. These are the sentences which must be either true or false.

Example:

1. 2+2 is 4. It is an atomic proposition as it is a true fact.
2. “The sun is cold” is also a proposition as it is a false fact.

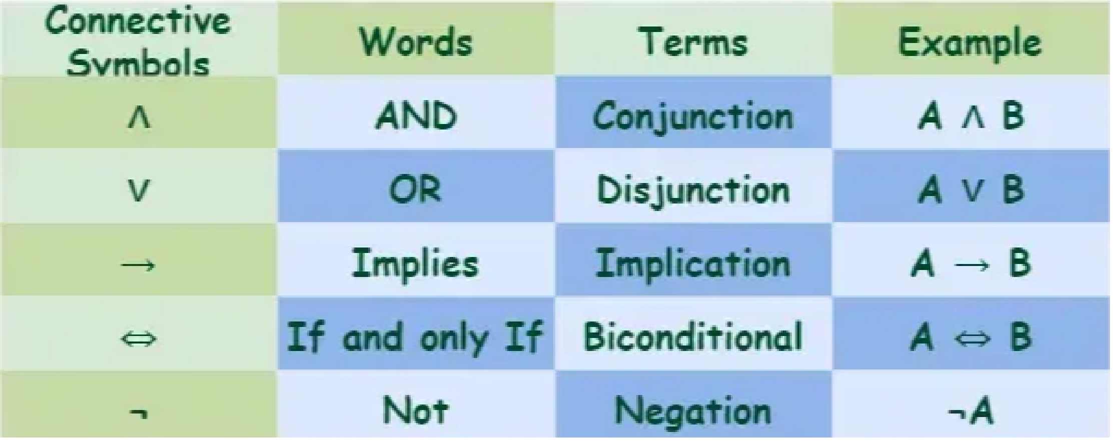
Compound proposition:

- Compound propositions are constructed by combining simpler or atomic propositions, using parentheses and logical connectives.

Examples:

1. "It is raining today, and the street is wet."
2. "Ajay is a doctor, and his clinic is in Mumbai."

Logical Connectives:



Connective Symbols	Words	Terms	Example
∧	AND	Conjunction	A ∧ B
∨	OR	Disjunction	A ∨ B
→	Implies	Implication	A → B
⇔	If and only If	Biconditional	A ⇔ B
¬	Not	Negation	¬A

- Logical connectives are used to connect two simpler propositions or represent a sentence logically.
- We can create compound propositions with the help of logical connectives.
- There are mainly five connectives, which are given as follows:

1. Negation:

- A sentence such as ¬P is called the negation of P.
- A literal can be either a positive literal or a negative literal.

p	¬p
T	F
F	T

Example: Today is not Friday

2. Conjunction:

- A sentence that has ∧ connective, such as P ∧ Q, is called a conjunction.

Example:
"Rohan is intelligent and hardworking."

It can be written as:

P = Rohan is intelligent

Q = Rohan is hardworking

→ P ∧ Q

p	**q**	**p ∧ q**
T	T	T
T	F	F
F	T	F
F	F	F

3. Disjunction:

- A sentence that has ∨ connective, such as P ∨ Q, is called disjunction, where P and Q are the propositions.

Example:
"Ritika is a doctor or an engineer."

Here,

P = Ritika is a doctor

Q = Ritika is an engineer

→ P ∨ Q

p	**q**	**p ∨ q**
T	T	T
T	F	T
F	T	T
F	F	F

4. Implication:

- A sentence such as P → Q is called an implication. Implications are also known as **if-then** rules.

Example:
"If it is raining, then the street is wet."

Let,

P = It is raining

Q = The street is wet

→ P → Q

p	**q**	**p → q**
T	T	T
T	F	F
F	T	T
F	F	T

5. Biconditional:

- A sentence such as P ⇔ Q is a biconditional sentence.

Example:
"If I am breathing, then I am alive."

P = I am breathing

Q = I am alive

→ P ⇔ Q

p	**q**	**p ↔ q**
T	T	T
T	F	F
F	T	F
F	F	T

2. Semantics of Propositional Calculus

Like syntax, we will now define semantics for propositional calculus as follows:

- **The truth assignment of negation:**
  - Where P is any propositional symbol: It is F (false), if the assignment to P is T (true);and T (true), if the assignment to P is F (false).
- **The truth assignment to conjunction:**
  - It is T (true) only when both the conjuncts have the truth-value T; otherwise, it is F (false).

- **The truth assignment of disjunction:**
  - It is F (false) only when both the disjuncts have the truth-value F; otherwise, it is T (true).
- **The truth assignment of implication:**
  - It is F (false) only when the symbol before the implication (i.e., premise) is T (true) and the symbol after the implication (i.e., consequent) is F (false);otherwise, it is T (true).
- **The truth assignment of biconditional:**
  - It is T (true) only when symbols on both sides are either T or F; otherwise, it is F (false).

3. Well-Formed Formula (WFF)

- Propositional logic uses a symbolic “language” to represent the logical structure, or form, of a compound proposition.
- Like any language, this symbolic language has rules of syntax—grammatical rules for putting symbols together in the right way.
- Any expression that obeys the syntactic rules of propositional logic is called a well-formed formula, or **WFF**.

The syntax of propositional logic is easy to learn and follows three rules:

1. Any capital letter by itself is a WFF.
2. Any WFF can be prefixed with “~” (The result will be a WFF too).
3. Any two WFFs can be put together with “^”, “∨”, “→”, or “↔” between them, enclosing the result in parentheses (This will be a WFF too).

WFF	non - WFF
A	A~
~A	(A)
(A ∧ B)	(A ∧)
(~A ∧ B)	(A ∧ B) v C)
((~A ∧ B) ∨ C)	

Properties of Operators:

- **Commutativity:**

  $P \land Q = Q \land P$

  $P \lor Q = Q \lor P$

- **Associativity:**

  $(P \land Q) \land R = P \land (Q \land R)$

  $(P \lor Q) \lor R = P \lor (Q \lor R)$

- **Identity element:**

  $P \land \text{True} = P$

  $P \lor \text{True} = \text{True}$

- **Distributive:**

  $P \land (Q \lor R) = (P \land Q) \lor (P \land R)$

  $P \lor (Q \land R) = (P \lor Q) \land (P \lor R)$

- **De Morgan's Law:**

  $\neg(P \land Q) = (\neg P) \lor (\neg Q)$

  $\neg(P \lor Q) = (\neg P) \land (\neg Q)$

- **Double-negation elimination:**

  $\neg(\neg P) = P$

4. Properties of Statements

- A formula A in a language Q is valid if it is true for every interpretation of Q.
- A formula A in a language Q is satisfiable if it is true for some interpretation of Q.
- A formula A of the language of arithmetic is decidable if it represents a decidable set, i.e., if there is an effective method which, given a substitution of the free variables of A, says that either the resulting instance of A is provable or its negation is.

Inferencing in Propositional Logic

- In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, so generating the conclusions from evidence and facts is termed as Inference.
- Inference rules are the templates for generating valid arguments.
- Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.
- In inference rules, the implication among all the connectives plays an important role.

Inferencing in Propositional Logic

- Following are some terminologies related to inference rules:
- **Implication:** It is one of the logical connectives which can be represented as P → Q. It is a Boolean expression.
- **Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as Q → P.
- **Contrapositive:** The negation of converse is termed as contrapositive, and it can be represented as ¬Q → ¬P.
- **Inverse:** The negation of implication is called inverse. It can be represented as ¬P → ¬Q.

Limitations of Propositional Logic

- We cannot represent relations like ALL, some, or none with propositional logic. Example:
  - All the girls are intelligent.
  - Some apples are sweet.
- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties or logical relationships.

Predicate Logic (First-Order Logic)

- In propositional logic, we can only represent the facts, which are either true or false.

- PL is not sufficient to represent the complex sentences or natural language statements.
- The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.
    - "Some humans are intelligent."
- To represent the above statements, PL logic is not sufficient, so we required some more powerful logic, such as first-order logic.

Predicate Logic (First-Order Logic)

- First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expressive to represent the natural language statements in a concise way.
- First-order logic is also known as Predicate logic or First-order predicate logic. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.
- First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:
    - **Objects:** A, B, people, numbers, colors, wars, theories, squares.
    - **Relations:** It can be unary relation such as: red, round, is adjacent, or n-ary relation such as: the sister of, brother of, has color, comes between.
    - **Function:** Father of, best friend, third inning of, end of, ...

Predicate Logic (First-Order Logic)

- The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic syntactic elements of first-order logic are symbols.
- **Atomic sentences:** Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.
    - We can represent atomic sentences as Predicate(term1, term2, ..., term n/Object).

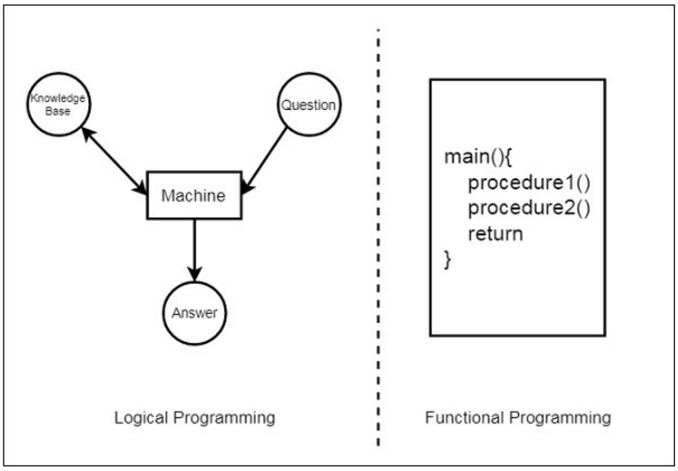
    - Example:
        - Ravi and Ajay are brothers: ⇒ Brothers(Ravi, Ajay).
        - Chinky is a cat: ⇒ Cat(Chinky).

Prolog

- Prolog is a short form of LOGical PROgramming.
- Prolog is a logic programming language.
- 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.
- Prolog is intended primarily as a declarative programming language.
- In Prolog, logic is expressed as relations (called as Facts and Rules).

Prolog

- To obtain the solution, the user asks a question rather than running a program.
- When a user asks a question, to determine the answer, the run-time system searches through the database of facts and rules.
- The first Prolog was 'Marseille Prolog', which is based on work by Colmerauer.
- The major example of fourth-generation programming language was Prolog.



```
main(){
   procedure1()
   procedure2()
   return
}
```

Prolog

- Prolog language basically has three different elements:
  - **Facts:** A fact is a 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 must be met. Example: This implies that for X to be the grandfather of Y, Z should be a parent of Y, and X should be the father of Z.

    ```
    scss
    Copy code
    grandfather(X, Y) :- father(X, Z), parent(Z, Y).
    ```

  - **Questions:** To run a Prolog program, we need some questions, and those questions can be answered by the given facts and rules.

Prolog

- Installation of Swi-Prolog in Linux:

  ```
  arduino
  Copy code
  sudo apt-get install swi-prolog
  ```

- Prolog facts are expressed in a definite pattern. Facts contain entities and their relation. Entities are written within parentheses separated by commas (,). Their relation is expressed at the start, outside the parentheses. Every fact/rule ends with a dot (.).
  - **Format:**

    ```
    scss
    Copy code
    relation(entity1, entity2, ....k'th entity).
    ```

Prolog Example

- Example:

```
scss
Copy code
friends(raju, mahesh).
singer(sonu).
odd_number(5).
```

- **Explanation**
    - Raju and Mahesh are friends.
    - Sonu is a singer.
    - 5 is an odd number.