UNIT - I

Defining Artificial Intelligence, Defining AI techniques, Using Predicate Logic and Representing Knowledge as Rules, Representing simple facts in logic, Computable functions and predicates, Procedural vs Declarative knowledge, Logic Programming

Defining Artificial Intelligence:

Artificial Intelligence (AI) is a field of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence. These tasks include reasoning, learning, problem-solving, perception, and natural language understanding. AI can be classified into different types based on functionality and capability:

- **Narrow AI (Weak AI)**: Systems designed to perform a narrow task, such as facial recognition or voice assistants. They cannot perform beyond their specific tasks.Weak AI is an AI that is created to solve a particular problem or perform a specific task.
- **General AI (Strong AI)**: A theoretical concept where a system can understand, learn, and apply intelligence across a wide range of tasks at a human level.It is a general purpose intelligence that can demonstrate human abilities. Human abilities such as learning from experience, reasoning, etc. can be demonstrated by this AI.
- **Artificial Super-intelligence**: A hypothetical level of AI where systems surpass human intelligence in all aspects.It ranges from a machine being just smarter than a human to a machine being trillion times smarter than a human.

The most common answer that one expects is **"to make computers intelligent so that they can act intelligently!"**, but the question is how much intelligent? How can one judge intelligence?

As intelligent as humans. If the computers can, somehow, solve real-world problems, by improving on their own from past experiences, they would be called "intelligent". Thus, the AI systems are more generic(rather than specific), can "think" and are more flexible.

Intelligence, as we know, is the ability to acquire and apply knowledge. Knowledge is the information acquired through experience. Experience is the knowledge gained through exposure(training). Summing the terms up, we get **artificial intelligence** as the "copy of something natural(i.e., human beings) 'WHO' is capable of acquiring and applying the information it has gained through exposure."

Need for Artificial Intelligence

- **To create expert systems** that exhibit intelligent behavior with the capability to learn, demonstrate, explain, and advise its users.
- **Helping machines find solutions** to complex problems like humans do and applying them as algorithms in a computer-friendly manner.
- **Improved efficiency:** Artificial intelligence can automate tasks and processes that are time-consuming and require a lot of human effort. This can help improve efficiency and productivity, allowing humans to focus on more creative and high-level tasks.
- **Better decision-making:** Artificial intelligence can analyze large amounts of data and provide insights that can aid in decision-making. This can be especially useful in domains like finance, healthcare, and logistics, where decisions can have significant impacts on outcomes.

- **Enhanced accuracy:** Artificial intelligence algorithms can process data quickly and accurately, reducing the risk of errors that can occur in manual processes. This can improve the reliability and quality of results.
- **Personalization:** Artificial intelligence can be used to personalize experiences for users, tailoring recommendations, and interactions based on individual preferences and behaviors. This can improve customer satisfaction and loyalty.
- **Exploration of new frontiers:** Artificial intelligence can be used to explore new frontiers and discover new knowledge that is difficult or impossible for humans to access. This can lead to new breakthroughs in fields like astronomy, genetics, and drug discovery.

Applications of Artificial Intelligence :

Artificial Intelligence has many practical applications across various industries and domains, including:

1. **Healthcare:** AI is used for medical diagnosis, drug discovery, and predictive analysis of diseases.
2. **Finance:** AI helps in credit scoring, fraud detection, and financial forecasting.
3. **Retail:** AI is used for product recommendations, price optimization, and supply chain management.
4. **Manufacturing:** AI helps in quality control, predictive maintenance, and production optimization.
5. **Transportation:** AI is used for autonomous vehicles, traffic prediction, and route optimization.
6. **Customer service:** AI-powered chat-bots are used for customer support, answering frequently asked questions, and handling simple requests.
7. **Security:** AI is used for facial recognition, intrusion detection, and cyber-security threat analysis.
8. **Marketing:** AI is used for targeted advertising, customer segmentation, and sentiment analysis.
9. **Education:** AI is used for personalized learning, adaptive testing, and intelligent tutoring systems.

az

The Future of AI Technologies:

1. Reinforcement Learning: Reinforcement Learning is an interesting field of Artificial Intelligence that focuses on training agents to make intelligent decisions by interacting with their environment.

2. Explainable AI: this AI techniques focus on providing insights into how AI models arrive at their conclusions.

3. Generative AI: Through this technique AI models can learn the underlying patterns and create realistic and novel outputs.

4. Edge AI:AI involves running AI algorithms directly on edge devices, such as smartphones, IoT devices, and autonomous vehicles, rather than relying on cloud-based processing.

5. Quantum AI: Quantum AI combines the power of quantum computing with AI algorithms to tackle complex problems that are beyond the capabilities of classical computers.

Drawbacks of Artificial Intelligence :

1. **Bias and unfairness:** AI systems can perpetuate and amplify existing biases in data and decision-making.
2. **Lack of transparency and accountability**: Complex AI systems can be difficult to understand and interpret, making it challenging to determine how decisions are being made.
3. **Job displacement**: AI has the potential to automate many jobs, leading to job loss and a need for res-killing.
4. **Security and privacy risks:** AI systems can be vulnerable to hacking and other security threats, and may also pose privacy risks by collecting and using personal data.
5. **Ethical concerns:** AI raises important ethical questions about the use of technology for decision-making, including issues related to autonomy, accountability, and human dignity.

Defining AI Techniques

AI techniques refer to the methodologies and approaches used to build AI systems. These include:

- **Machine Learning (ML)**: A method where algorithms learn from data to make decisions or predictions. Sub-fields include supervised learning, unsupervised learning, and reinforcement learning.
- **Natural Language Processing (NLP)**: Techniques for understanding and generating human language. This includes tasks like language translation and sentiment analysis.
- **Computer Vision**: Techniques for analyzing and interpreting visual data from the world, such as object recognition.
- **Robotics**: The study and design of robots that can perform tasks in the real world.
- **Expert Systems**: AI systems that use a knowledge base of human expertise to solve specific problems.
- **Chatbots:** **AI**-powered virtual assistants that can interact with users through text-based or voice-based interfaces.

Knowledge Representation

Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. **But how machines do all these things comes under knowledge representation and reasoning**. Hence we can describe Knowledge representation as following:

- Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.
- It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

What to Represent:

Following are the kind of knowledge which needs to be represented in AI systems:

- **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
- **Events:** Events are the actions which occur in our world.
- **Performance:** It describe behavior which involves knowledge about how to do things.
- **Meta-knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledge base is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Knowledge Representation in First-Order Logic (Predicate Logic)

First-order logic (FOL), also known as **predicate logic**, is a powerful formalism used for **knowledge representation** in artificial intelligence and computer science. It extends **propositional logic** by allowing the use of quantifiers and predicates, enabling the representation of complex statements about objects and their relationships. Here are the key components and concepts of knowledge representation in first-order logic:

Key Components of First-Order Logic

1. **Constants**:
   - **Definition**: Constants are symbols that represent specific objects in the domain.
   - **Examples**: If a, b, and c are constants, they might represent specific individuals like Alice, Bob, and Charlie.

2. **Variables**:
- **Definition**: Variables are symbols that can represent any object in the domain.
- **Examples**: Variables such as x, y, and z can represent any object in the domain.

3. **Predicates**:
- **Definition**: Predicates represent properties of objects or relationships between objects.
- **Examples**: P(x) could mean “x is a person”, while Q(x, y) could mean “x is friends with y”.

4. **Functions**:
- **Definition**: Functions map objects to other objects.
- **Examples**: f(x) could represent a function that maps an object x to another object, like “the father of x”.

5. **Quantifiers**:
- **Universal Quantifier (∀)**: Indicates that a statement applies to all objects in the domain. For example, ∀x P(x) means “P(x) is true for all x”.
- **Existential Quantifier (∃)**: Indicates that there exists at least one object in the domain for which the statement is true. For example, ∃x P(x) means “There exists an x such that P(x) is true”.

6. **Logical Connectives**:
- **Definition**: These include ∧ (and), ∨ (or), ¬ (not), → (implies), and ↔ (if and only if).
- **Examples**: P(x) ∧ Q(x, y) means “P(x) and Q(x, y) are both true”.

7. **Equality**:
- **Definition**: States that two objects are the same.
- **Examples**: x = y asserts that x and y refer to the same object.

Syntax of First-Order Logic

The syntax of **FOL** defines the rules for constructing well-formed formulas:

- **Atomic Formulas**: The simplest formulas, which can be predicates applied to terms (e.g., P(a), Q(x,y)).
- **Complex Formulas**: Formed by combining atomic formulas using logical connectives and quantifiers (e.g., ∀x(P(x)∨¬Q(x,f(y)))).

Semantics of First-Order Logic

The semantics define the meaning of FOL statements:

- **Domain**: A non-empty set of objects over which the variables range.
- **Interpretation**: Assigns meanings to the constants, functions, and predicates, specifying which objects the constants refer to, which

function the function symbols denote, and which relations the predicate symbols denote.

- **Truth Assignment**: Determines the truth value of each formula based on the interpretation.

Examples of Knowledge Representation in FOL

1. **Facts**: Simple statements about objects.
   - $P(a)$ (Object a has property P).
   - $Q(a, b)$ (Objects a and b are related by Q).
2. **Rules**: Implications that describe general relationships.
   - $\forall x(P(x)\rightarrow Q(x))$(If x has property P, then x also has property Q).
3. **Existential Statements**: Indicate the existence of objects with certain properties.
   - $\exists xP(x)$ (There exists an x such that $P(x)$ is true).
4. **Universal Statements**: Apply to all objects in the domain.
   - $\forall x(P(x)\lor\neg Q(x))$ (For all x, either $P(x)$ is true or $Q(x)$ is not true).

Example Knowledge Base in FOL

Consider a knowledge base representing a simple family relationship:

1. **Constants:**
   - John, Mary
2. **Predicates:**
   - Parent(x, y): x is a parent of y.
   - Male(x): x is male.
   - Female(x): x is female.
3. **Statements:**
   - Parent(John, Mary)
   - Male(John)
   - Female(Mary)

Applications of First-Order Logic in Knowledge Representation

1. **Expert Systems**: FOL is used to represent expert knowledge in various domains such as medicine, finance, and engineering, enabling systems to reason and make decisions based on logical rules.
2. **Natural Language Processing**: FOL provides a formal framework for representing the meaning of natural language sentences, facilitating semantic analysis and understanding in NLP tasks.
3. **Semantic Web**: FOL is foundational to ontologies and knowledge graphs on the Semantic Web, enabling precise and machine-interpret-able representations of knowledge.
4. **Robotics**: FOL is employed in robotic systems to represent spatial relationships, object properties, and task constraints, aiding in robot planning, navigation, and manipulation.

5. **Database Systems**: FOL-based query languages such as SQL enable expressive querying and manipulation of relational databases, allowing for complex data retrieval and manipulation.

Challenges & Limitations of First-Order Logic in Knowledge Representation

Challenges of First-Order Logic in Knowledge Representation

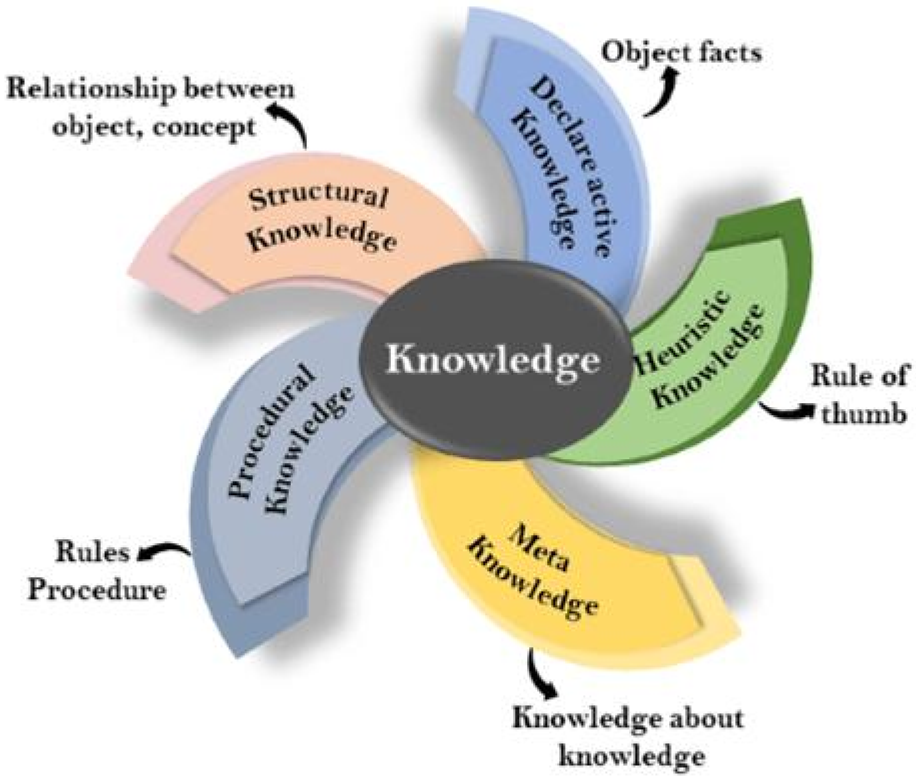
1. **Complexity**: Representing certain real-world domains accurately in FOL can lead to complex and unwieldy formulas, making reasoning and inference computationally expensive.
2. **Expressiveness Limitations**: FOL has limitations in representing uncertainty, vagueness, and probabilistic relationships, which are common in many AI applications.
3. **Knowledge Acquisition**: Encoding knowledge into FOL requires expertise and manual effort, making it challenging to scale and maintain large knowledge bases.
4. **Inference Scalability**: Reasoning in FOL can be computationally intensive, especially In large knowledge bases, requiring efficient inference algorithms and optimization techniques.
5. **Handling Incomplete Information**: FOL struggles with representing and reasoning with incomplete or uncertain information, which is common in real-world applications.

Limitations of First-Order Logic in Knowledge Representation

6. **Inability to Represent Recursive Structures**: FOL cannot directly represent recursive structures, limiting its ability to model certain types of relationships and processes.
7. **Lack of Higher-Order Reasoning**: FOL lacks support for higher-order logic, preventing it from representing and reasoning about properties of predicates or functions.
8. **Difficulty in Representing Context and Dynamics**: FOL struggles with representing dynamic or context-dependent knowledge, such as temporal relationships or changes over time.
9. **Limited Representation of Non-binary Relations**: FOL primarily deals with binary relations, making it less suitable for representing complex relationships involving multiple entities.
10. **Difficulty in Handling Non-monotonic Reasoning**: FOL is not well-suited for non-monotonic reasoning, where new information can lead to retraction or modification of previously inferred conclusions.

Types of knowledge

Following are the various types of knowledge:



1. Declarative Knowledge:

- Declarative knowledge is to know about something.
- It includes concepts, facts, and objects.
- It is also called descriptive knowledge and expressed in declarative sentences.
- It is simpler than procedural language.

2. Procedural Knowledge

- It is also known as imperative knowledge.
- Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
- It can be directly applied to any task.
- It includes rules, strategies, procedures, agendas, etc.
- Procedural knowledge depends on the task on which it can be applied.
-

3. Meta-knowledge:

- Knowledge about the other types of knowledge is called Meta-knowledge.

4. Heuristic knowledge:

- Heuristic knowledge is representing knowledge of some experts in a filed or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

5. Structural knowledge:

- Structural knowledge is basic knowledge to problem-solving.
- It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

Using Predicate Logic and Representing Knowledge as Rules

Predicate Logic is a formal system in logic that allows for expressing statements about objects and their relationships. It includes:

- **Predicates**: Functions that return true or false. For example, isTall(John) might represent the statement "John is tall."
- **Quantifiers**: Symbols like ∀ (for all) and ∃ (there exists) used to indicate the scope of a statement.

Knowledge Representation as Rules involves using logical rules to represent facts and relationships. For example:

- **Fact**: isStudent(John).
- **Rule**: isSmart(X) :- isStudent(X), studiesHard(X).
  This rule states that X is smart if X is a student and studies hard.

Example :

1. Family Relationship (Parent-Child):

- **Predicate Logic:**
  - Parent(x,y) means "x is a parent of y."
  - $\forall x,y\ (Parent(x,y) \Rightarrow Child(y,x))$
- **Rule Representation:**
  - If x is a parent of y, then y is a child of x.

2. Eligibility for Voting:

- **Predicate Logic:**
  - Citizen(x) means "x is a citizen."
  - Age(x,18) means "x is 18 years or older."
  - $\forall x\ (Citizen(x) \land Age(x,18) \Rightarrow EligibleToVote(x))$
- **Rule Representation:**
  - If x is a citizen and x is 18 years or older, then x is eligible to vote.

3. University Admission:

- **Predicate Logic:**
  - GPA(x,y) means "x has a GPA of y."
  - SATScore(x,z) means "x has an SAT score of z."
  - $\forall x,y,z\ (GPA(x,y) \land y \geq 3.5 \land SATScore(x,z) \land z \geq 1200 \Rightarrow Admitted(x))$
- **Rule Representation:**
  - If x has a GPA of 3.5 or higher and an SAT score of 1200 or higher, then x is admitted to the university.

4. Access Control System:

- **Predicate Logic:**
  - Employee(x) means "x is an employee."
  - HasID(x) means "x has an ID badge."
  - $\forall x\ (Employee(x) \land HasID(x) \Rightarrow AccessGranted(x))$
- **Rule Representation:**
  - If x is an employee and has an ID badge, then x is granted access.

5. Medical Diagnosis:

- **Predicate Logic:**
  - Fever(x) means "x has a fever."
  - Cough(x) means "x has a cough."
  - Flu(x) means "x has the flu."
  - $\forall x\ (Fever(x) \land Cough(x) \Rightarrow Flu(x))$
- **Rule Representation:**
  - If x has a fever and a cough, then x has the flu.

Representing Simple Facts in Logic

Simple facts can be represented using propositional logic or first-order logic. For example:

1. **Propositional Logic**: p, q, where p and q are atomic propositions.

Propositional logic in Artificial intelligence

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:

- It is Sunday.
- The Sun rises from West (False proposition)
- 3+3= 7(False proposition)
- 5 is a prime number.

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 a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and **logical connectives**.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.
- A proposition formula which is always true is called **tautology**, and it is also called a valid sentence.
- A proposition formula which is always false is called **Contradiction**.
- A proposition formula which has both true and false values is called
- Statements which are questions, commands, or opinions are not propositions such as "**Where is Rohini**", "**How are you**", "**What is your name**", are not propositions.

Syntax of propositional logic:

The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:

1. **Atomic Propositions**
2. **Compound(Complex) 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:

- 2+2 is 4, it is an atomic proposition as it is a **true** fact.
- "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 parenthesis and logical connectives.

Example:

- "It is raining today, and street is wet."
- "Ankit is a doctor, and his clinic is in Mumbai."

Logical Connectives:

Logical connectives are used to connect two simpler propositions or representing 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 $\neg$ P is called negation of P. A literal can be either Positive literal or negative literal.
2. **Conjunction:** A sentence which has $\wedge$ connective such as, **P $\wedge$ 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$\wedge$ Q.
3. **Disjunction:** A sentence which has $\vee$ connective, such as **P $\vee$ Q**. is called disjunction, where P and Q are the propositions.
   Example: "Ritika is a doctor or Engineer",
   Here P= Ritika is Doctor. Q= Ritika is Doctor, so we can write it as **P $\vee$ Q**.
4. **Implication:** A sentence such as P → Q, is called an implication. Implications are also known as if-then rules. It can be represented as
   If it is raining, then the street is wet.
   Let P= It is raining, and Q= Street is wet, so it is represented as P → Q
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, it can be represented as P ⇔ Q.

Connective symbols	Word	Technical term	Example
$\land$	AND	Conjunction	$A \land B$
$\lor$	OR	Disjunction	$A \lor B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

Following is the summarized table for Propositional Logic Connectives:

Truth Table:

In propositional logic, we need to know the truth values of propositions in all possible scenarios. We can combine all the possible combination with logical connectives, and the representation of these combinations in a tabular format is called **Truth table**. Following are the truth table for all logical connectives:

For Negation:

P	$\neg$ P
True	**False**
False	**True**

For Conjunction:

P	Q	$P \land Q$
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**False**

For disjunction:

P	Q	$P \lor Q$.
True	True	**True**
False	True	**True**
True	False	**True**
False	False	**False**

For Implication:

P	Q	$P \rightarrow Q$
True	True	**True**
True	False	**False**
False	True	**True**
False	False	**True**

For Biconditional:

P	Q	P⇔ Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**True**

Precedence of connectives:

Just like arithmetic operators, there is a precedence order for propositional connectors or logical operators. This order should be followed while evaluating a propositional problem. Following is the list of the precedence order for operators:

Precedence	**Operators**
First Precedence	Parenthesis
Second Precedence	Negation
Third Precedence	Conjunction(AND)
Fourth Precedence	Disjunction(OR)
Fifth Precedence	Implication
Six Precedence	Biconditional

Logical equivalence:

Logical equivalence is one of the features of propositional logic. Two propositions are said to be logically equivalent if and only if the columns in the truth table are identical to each other.

Let's take two propositions A and B, so for logical equivalence, we can write it as A⇔B. In below truth table we can see that column for ¬A∨ B and A→B, are identical hence A is Equivalent to B

A	B	¬A	¬A∨ B	A→B
T	T	F	**T**	**T**
T	F	F	**F**	**F**
F	T	T	**T**	**T**
F	F	T	**T**	**T**

Properties of Operators:

- **Commutativity:**
  - P∧ Q= Q ∧ P, or
  - P ∨ Q = Q ∨ P.
- **Associativity:**
  - (P ∧ Q) ∧ R= P ∧ (Q ∧ R),
  - (P ∨ Q) ∨ R= P ∨ (Q ∨ R)
- **Identity element:**
  - P ∧ True = P,
  - P ∨ True= True.
- **Distributive:**
  - P∧ (Q ∨ R) = (P ∧ Q) ∨ (P ∧ R).
  - P ∨ (Q ∧ R) = (P ∨ Q) ∧ (P ∨ R).
- **De Morgan's Law:**
  - ¬ (P ∧ Q) = (¬P) ∨ (¬Q)
  - ¬ (P ∨ Q) = (¬ P) ∧ (¬Q).
- **Double-negation elimination:**
  - ¬ (¬P) = P.

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.

2. First-Order Logic: Includes variables and quantifiers, such as loves(John, Mary) representing "John loves Mary."

Computable Functions and Predicates

Computable functions and predicates refer to functions and predicates that can be computed or decided by an algorithm.

- **Computable Function**: A **computable function** is a function for which there exists a finite, well-defined procedure (or algorithm) that can be followed to produce the function's output for any valid input within a finite amount of time. In other words, a function is computable if there exists an algorithm that can compute the function's value for any given input.

Example : f(x) = x + 1

- **Computable Predicate**: A **predicate** is a logical statement that can be true or false depending on the values of its variables. In the context of logic and computer science, a predicate is often a function that returns a boolean value (true or false) based on some condition.

  Example : isEven(x), which returns true if x is an even number.

Example : Sum of Two Numbers:

- **Function:**
  - Sum(x,y)=x+y
- **Description:**
  - This function takes two numbers x and y as input and returns their sum.
- **Example:**
  - Sum(3,5)=8

2. Check Even Number:

- **Predicate:**
  - IsEven(x)=true if xmod 2=0
- **Description:**
  - This predicate checks whether a number xxx is even.
- **Example:**
  - IsEven(4)=true

3. Factorial of a Number:

- **Function:**
  - Factorial(x)={1 if x=0

    x×Factorial(x−1) if x>0
- **Description:**
  - This function computes the factorial of a non-negative integer x, which is the product of all positive integers less than or equal to x.

- **Example:**
    - Factorial(5)=5×4×3×2×1=120

4. Greater Than Predicate:

- **Predicate:**
    - GreaterThan(x,y)=true if x>y
- **Description:**
    - This predicate checks whether number x is greater than number y.
- **Example:**
    - GreaterThan(7,5)=true

Procedural vs. Declarative Knowledge

Procedural knowledge refers to knowing **“*how*”** to do something. It involves the steps, methods, or procedures required to complete a task. This type of knowledge is often implicit, meaning that it can be difficult to verbalize or describe in detail. Instead, it is typically demonstrated through action.

- **Characteristics of Procedural Knowledge:**
    - Focuses on the *process* or *procedure*.
    - Often gained through practice or experience.
    - Can be thought of as "know-how."
- **Examples of Procedural Knowledge:**
    - Riding a bicycle.
    - Solving a mathematical equation using a specific algorithm.
    - Programming a computer using a particular coding language.

Declarative knowledge refers to knowing “***what***” something is. It involves facts, information, and understanding that can be explicitly stated or declared. This type of knowledge is more about awareness and comprehension of concepts, definitions, and principles.

- **Characteristics of Declarative Knowledge:**
    - Focuses on *facts* and *information*.
    - Can be easily articulated or communicated.
    - Can be thought of as "know-that."

- **Examples of Declarative Knowledge:**
  - Knowing the capital of a country.
  - Understanding the laws of physics.
  - Recognizing the symptoms of a disease.

Key Differences:

Purpose:

- **Procedural Knowledge:** How to perform actions or processes.
- **Declarative Knowledge:** Understanding of facts and concepts.

Expression:

- **Procedural Knowledge:** Often demonstrated through performance or action.
- **Declarative Knowledge:** Easily communicated through language, explanations, or writing.

Acquisition:

- **Procedural Knowledge:** Typically learned through practice and repetition.
- **Declarative Knowledge:** Often learned through study, reading, or instruction.

Examples:

Example 1: Cooking

- **Procedural Knowledge:**
  - Knowing how to bake a cake, including the steps to mix ingredients, set the oven temperature, and the sequence of adding components.
- **Declarative Knowledge:**
  - Knowing that baking powder is used as a leavening agent in baking cakes.

Example 2: Driving a Car

- **Procedural Knowledge:**
  - Knowing how to operate a car, including how to steer, accelerate, brake, and change gears.
- **Declarative Knowledge:**

- Knowing that the speed limit in a residential area is typically 25 mph.

Example 3: Programming

- **Procedural Knowledge:**
    - Knowing how to write a loop in a programming language like Python, including the syntax and structure required to implement it.
- **Declarative Knowledge:**
    - Knowing that loops are used to repeat a block of code multiple times.

Logic Programming

Logic programming is a programming paradigm that is based on formal logic. In logic programming, programs consist of a set of sentences in logical form, expressing facts and rules about some problem domain. Computation is performed through logical inference, where the program "solves" the problem by deriving conclusions from the given facts and rules.

- **Key Concepts:**
    - **Facts:** Statements that are always true. For example, "Socrates is a man" can be represented as a fact.
    - **Rules:** Conditional statements that define relationships between facts. For example, "If x is a man, then x is mortal" is a rule.
    - **Queries:** Questions asked to the system to derive new information or check the validity of certain conditions based on the facts and rules.
- **Logic Programming Works:**
    - In logic programming, you define what you want to achieve (the goal) rather than how to achieve it. The logic programming engine then searches for a solution that satisfies the goal based on the provided facts and rules.
    - The most common logic programming language is **Prolog** (short for "Programming in Logic").

Examples of Logic Programming:

Example 1: Family Relationships

- **Facts:**
    - parent(john, mary).
    - parent(mary, susan).

- **Rules:**

  - grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

- **Explanation:**

  - The facts state that John is the parent of Mary, and Mary is the parent of Susan.
  - The rule states that X is the grandparent of Y if X is the parent of Z and Z is the parent of Y.

- **Query:**

  - ?- grandparent(john, susan).
  - **Result:** Yes, John is the grandparent of Susan.

Example 2: Simple Arithmetic in Prolog

- **Facts:**

  - even(2).
  - even(4).

- **Rules:**

  - even(X) :- 0 is X mod 2.

- **Explanation:**

  - The facts state that 2 and 4 are even numbers.
  - The rule defines that a number X is even if the remainder when X is divided by 2 is 0.

- **Query:**

  - ?- even(6).
  - **Result:** Yes, 6 is even.

Example 3: Pathfinding in a Graph

- **Facts:**

  - edge(a, b).
  - edge(b, c).
  - edge(c, d).

- **Rules:**

  - path(X, Y) :- edge(X, Y).
  - path(X, Y) :- edge(X, Z), path(Z, Y).

- **Explanation:**
    - The facts define the edges of a graph, indicating direct connections between nodes.
    - The first rule states that there is a path between X and Y if there is a direct edge between X and Y.
    - The second rule states that there is a path between X and Y if there is a direct edge between X and some intermediate node Z, and a path between Z and Y.

- **Query:**
    - ?- path(a, d).
    - **Result:** Yes, there is a path from node a to node d.

Importance of Logic Programming

- **Declarative Nature:** Logic programming allows programmers to specify what the program should do rather than how to do it, which can make programs easier to write and understand.
- **Problem Solving:** It is well-suited for solving problems that involve complex relationships and constraints, such as puzzles, scheduling, and expert systems.
- **Automated Reasoning:** Logic programming is powerful in domains that require reasoning, such as artificial intelligence, where it can be used to model and solve problems automatically by deriving conclusions from known facts and rules.