What is 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 Knowledgebase is a group of the Sentences (Here, sentences are

used as a technical term and not identical with the English language).

Knowledge: Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. Following are the types of knowledge in artificial intelligence:

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 declarativesentences.
- 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:

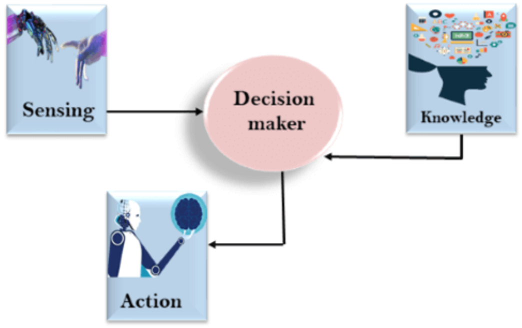
- Structural knowledge is basic knowledge to problem-solving.
- It describes relationships between various concepts such as kind of, part of, and grouping of something.
- It describes the relationship that exists between concepts or objects.

The relation between knowledge and intelligence:

Knowledge of real-worlds plays a vital role in intelligence and same for creating artificial intelligence. Knowledge plays an important role in demonstrating intelligent behavior in AI agents. An agent is only able to accurately act on some input when he has some knowledge or experience about that input.

Let's suppose if you met some person who is speaking in a language which you don't know, then how you will able to act on that. The same thing applies to the intelligent behavior of the agents.

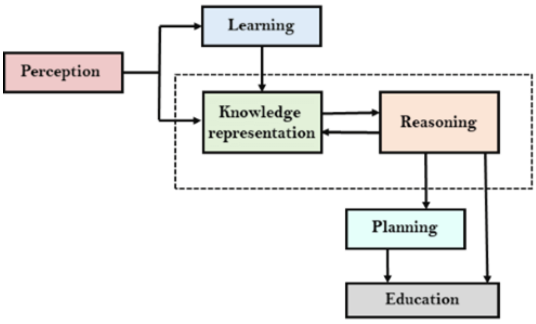
As we can see in below diagram, there is one decision maker which act by sensing the environment and using knowledge. But if the knowledge part will not present then, it cannot display intelligent behavior.



AI knowledge cycle:

An Artificial intelligence system has the following components for displaying intelligent behavior:

- Perception
- Learning
- Knowledge Representation and Reasoning
- Planning
- Execution



The above diagram is showing how an AI system can interact with the real world and what components help it to show intelligence. AI system has Perception component by which it retrieves information from its environment. It can be visual, audio or another form of sensory input. The learning component is responsible for learning from data captured by Perception comportment. In the complete cycle, the main components are knowledge representation and Reasoning. These two components are involved in showing the intelligence in machine-like humans. These two components are independent with each other but also coupled together. The planning and execution depend on analysis of Knowledge representation and reasoning.

Approaches to knowledge representation:

There are mainly four approaches to knowledge representation, which are givenbelow:

1. Simple relational knowledge:

- It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.

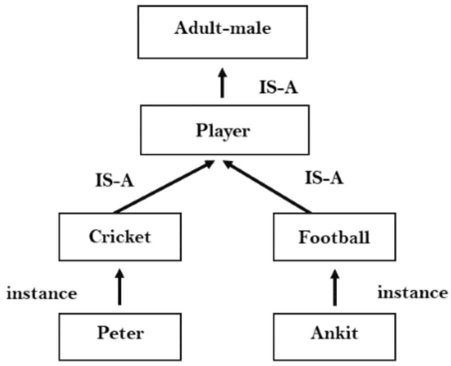
- This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
- This approach has little opportunity for inference.

Example: The following is the simple relational knowledge representation.

Player	Weight	Age
Player1	65	23
Player2	58	18
Player3	75	24

2. Inheritable knowledge:

- In the inheritable knowledge approach, all data must be stored into a hierarchy of classes.
- All classes should be arranged in a generalized form or a hierarchal manner.
- In this approach, we apply inheritance property.
- Elements inherit values from other members of a class.
- This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
- Every individual frame can represent the collection of attributes and its value.
- In this approach, objects and values are represented in Boxed nodes.
- We use Arrows which point from objects to their values.
- **Example:**



3. Inferential knowledge:

- Inferential knowledge approach represents knowledge in the form of formal logics.
- This approach can be used to derive more facts.
- It guaranteed correctness.
- **Example:** Let's suppose there are two statements:

    1. Marcus is a man
    2. All men are mortal
       Then it can represent as;
       man(Marcus)
       ∀x = man (x) ----------> mortal (x)s

4. Procedural knowledge:

- Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.
- In this approach, one important rule is used which is **If-Then rule**.
- In this knowledge, we can use various coding languages such as **LISP language** and **Prolog language**.
- We can easily represent heuristic or domain-specific knowledge using this approach.
- But it is not necessary that we can represent all cases in this approach.

Requirements for knowledge Representation system:

A good knowledge representation system must possess the following properties.

1. **1. Representational Accuracy:**
   KR system should have the ability to represent all kind of required knowledge.
2. **2. Inferential Adequacy:**
   KR system should have ability to manipulate the representational structures to produce new knowledge corresponding to existing structure.
3. **3. Inferential Efficiency:**
   The ability to direct the inferential knowledge mechanism into the most productive directions by storing appropriate guides.
4. **4. Acquisitional efficiency-** The ability to acquire the new knowledge easily using automatic methods.

Issues in knowledge representation

The main objective of knowledge representation is to draw the conclusions from the knowledge, but there are many issues associated with the use of knowledge representation techniques.

Some of them are listed below:

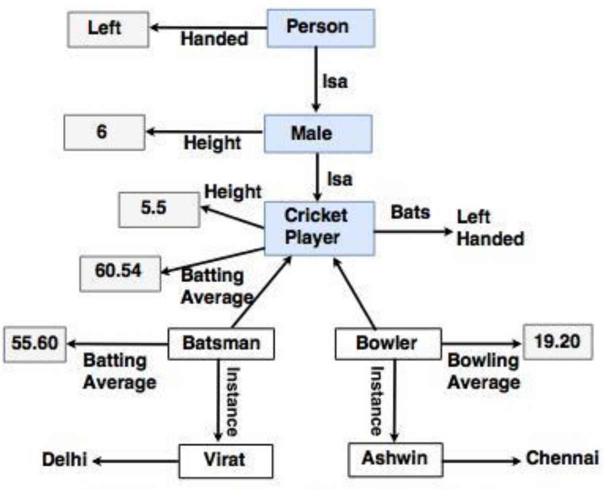


Fig: Inheratable Knowledge Representation

Refer to the above diagram to refer to the following issues.

1. Important attributes

There are two attributes shown in the diagram, **instance** and **isa.** Since these attributes support property of inheritance, they are of prime importance.

2. Relationships among attributes

Basically, the attributes used to describe objects are nothing but the entities. However, the attributes of an object do not depend on the encoded specific knowledge.

3. Choosing the granularity of representation

While deciding the granularity of representation, it is necessary to know the following:

i. What are the primitives and at what level should the knowledge be represented?

ii. What should be the number (small or large) of low-level primitives or high-level facts?

High-level facts may be insufficient to draw the conclusion while Low-level primitives may require a lot of storage.

For example: Suppose that we are interested in following facts:

John spotted Alex.

Now, this could be represented as "Spotted (agent(John), object (Alex))"

Such a representation can make it easy to answer questions such as: Who spotted Alex?

Suppose we want to know : "Did John see Sue?"

Given only one fact, user cannot discover that answer.

Hence, the user can add other facts, such as "Spotted (x, y) → saw (x, y)"

4. Representing sets of objects.

There are some properties of objects which satisfy the condition of a set together but not as individual;

Example: Consider the assertion made in the sentences:

"There are more sheep than people in Australia", and "English speakers can be found all over the world."

These facts can be described by including an assertion to the sets representing people, sheep, and English.

5. Finding the right structure as needed

To describe a particular situation, it is always important to find the access of right structure. This can be done by selecting an initial structure and then revising the choice.

While selecting and reversing the right structure, it is necessary to solve following problem statements. **They include the process on how to:**

- Select an initial appropriate structure.
- Fill the necessary details from the current situations.
- Determine a better structure if the initially selected structure is not appropriate to fulfill other conditions.
- Find the solution if none of the available structures is appropriate.
- Create and remember a new structure for the given condition.
- There is no specific way to solve these problems, but some of the effective knowledge representation techniques have the potential to solve them.

Logic Representation

Facts are the general statements that may be either True or False. Thus, logic can be used to represent such simple facts.

To build a Logic-based representation:

- User has to define a set of primitive symbols along with the required semantics.
- The symbols are assigned together to define legal sentences in the language for representing TRUE facts.
- New logical statements are formed from the existing ones. The statements which can be either TRUE or false but not both , are called propositions. A declarative sentence expresses a statement with a proposition as content;
  Example: The declarative "Cotton is white" expresses that Cotton is white. So, the sentence "Cotton is white" is a true statement.

What is Propositional Logic (PL)?

- Propositional logic is a study of propositions.
- Each proposition has either a true or a false value but not both at a time.

- Propositions is represented by variables.

  For example: Symbols 'p' and 'q' can be used to represent propositions.

There are two types of propositions:

1. Simple Preposition
2. compound Prepositions.

1. A simple preposition: It does not contain any other preposition.
For example: Rocky is a dog.

2. A compound preposition: It contains more than one prepositions.
For example: Surendra is a boy and he likes chocolate.

Connectives and the truth tables of compound prepositions are given below:
Consider 'p' and 'q' are two prepositions then,

1. Negation (¬p) indicates the opposite of p.
Truth table for negation:

p	**¬p**	**p**
0	1	0
1	0	1

2. Conjunction (p ∧ q) indicates that p and q both and are enclosed in parenthesis. So, p and q are called conjuncts .

Truth table for conjunction:

p	**q**	**p ∧ q**
0	0	0
0	1	0
1	0	0
1	1	1

3. Disjunction (p ∨ q) indicates that either p or q or both are enclosed in

parenthesis. Thus, p and q are called disjuncts.
Truth table for disjunction:

p	q	p ∨ q
0	0	0
0	1	1
1	0	1
1	1	1

4. Implication (p ⇒ q) consists of a pair of sentences separated by the ⇒ operator and enclosed in parentheses. The sentence to the left of the operator is called as an antecedent, and the sentence to the right is called as the consequent.
Truth table for implication:

p	q	p ⇒ q
0	0	1
0	1	1
1	0	0
1	1	1

5. Equivalence (p ⇔ q) is a combination of an implication and a reduction.
Truth table for Equivalence:

p	q	p ⇔ q
0	0	1
0	1	0
1	0	0
1	1	1

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:

1. a) It is Sunday.
2. b) The Sun rises from West (False proposition)

3. c) 3+3= 7(False proposition)
4. d) 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:

a. **Atomic Propositions**
b. **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. a) 2+2 is 4, it is an atomic proposition as it is a **true** fact.
2. b) "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:

1. a) "It is raining today, and street is wet."
2. b) "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 ¬ P is called negation of P. A literal can be either Positive literal or negative literal.
2. **Conjunction:** A sentence which 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.
3. **Disjunction:** A sentence which has ∨ connective, such as **P ∨ 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 ∨ 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.

Following is the summarized table for Propositional Logic Connectives:

Connective symbols	Word	Technical term	Example
∧	AND	Conjunction	A ∧ B
∨	OR	Disjunction	A ∨ B
→	Implies	Implication	A → B
⇔	If and only if	Biconditional	A⇔ B
¬ or ~	Not	Negation	¬ A or ¬ B

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	¬ P
True	**False**
False	**True**

For Conjunction:

P	Q	P∧Q
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**False**

For disjunction:

P	Q	P V Q.
True	True	**True**
False	True	**True**
True	False	**True**
False	False	**False**

For Implication:

P	Q	P→ 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**

Truth table with three propositions:

We can build a proposition composing three propositions P, Q, and R. This truth table is made-up of 8n Tuples as we have taken three proposition symbols.

P	Q	R	¬R	Pv Q	PvQ→¬R
True	True	True	False	True	False
True	True	False	True	True	True
True	False	True	False	True	False
True	False	False	True	True	True
False	True	True	False	True	False
False	True	False	True	True	True
False	False	True	False	False	True
False	False	False	True	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 $A \Leftrightarrow B$. In below truth table we can see that column for $\neg A \vee B$ and $A \rightarrow B$, 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:**
  - $P \wedge Q = Q \wedge P$, or
  - $P \vee Q = Q \vee P$.

- **Associativity:**
  - $(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$,
  - $(P \vee Q) \vee R = P \vee (Q \vee R)$

- **Identity element:**
    - $P \land True = P$,
    - $P \lor True = True$.
- **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:**
    - $\neg (\neg P) = P$.

Limitations of Propositional logic:

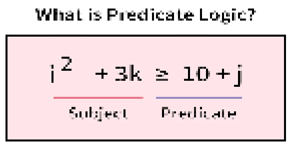
- We cannot represent relations like ALL, some, or none with propositional logic. Example:
    1. **All the girls are intelligent.**
    2. **Some apples are sweet.**
- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements in terms of their properties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

Predicate Logic In Artificial Intelligence

Logic is one of the essential ideas in the field of artificial intelligence. AI systems' reasoning and decision-making are based on logic. It offers a method for presenting and handling data in a way that enables an AI system to draw conclusions and inferences from it.

Predicate Logic in AI is one of the most significant types of logic. Predicate logic in AI is a formal framework for deducing relationships between objects and their qualities. Furthermore, it is a mathematical language that enables knowledge to be expressed precisely and unambiguously, making it perfect for usage in AI systems.

Introduction



Predicate Logic in AI is fundamentally a method for describing and modifying assertions about objects and their characteristics. It includes a collection of rules and symbols that enable us to build complex statements from simpler ones.

Predicates and variables are the core components of Predicate Logic in AI. A predicate is an assertion made regarding one or more things. For instance, the predicate "is blue" declares that a specific item possesses the attribute of being blue. A variable serves as a stand-in for an object so that assertions can apply to any object of a particular type. We can generalize assertions about vehicles by using the variable *x* to represent any car, for instance. There must be at least one object (represented by the variable *x*) with the quality of being blue, according to the adage "exists *x*, *x* is blue".

The two most common quantifiers are "for all" (represented by the symbol ∀) and "exists" (represented by the symbol ∃∃). For example, the statement "for all *x*, *x* is blue" asserts that all objects (represented by the variable *x*) have the property of being blue. Likewise, the statement "exists *x*, *x* is blue" asserts that at least one object (represented by the variable *x*) has the property of being blue.

Characteristics of Predicate Logic

Predicate Logic in AI has several characteristics that make it a powerful tool for AI applications. Some of these characteristics are:

- The Logical inference is allowed.
- More accurate knowledge representation of facts of the real world.
- Program designing is its application area.
- Better theoretical foundation.
- A predicate with no variable is called a Ground Atom.

What is Predicate Logic in AI Example with a Solution?

Let's consider an example of Predicate Logic to understand better how it works. Suppose we want to represent the following knowledge:

"All mammals are warm-blooded."

To represent this knowledge in Predicate Logic in AI, we would use the following symbols:

$M(x)$: x is a mammal. $W(x)$: x is warm-blooded.

We can then write the knowledge as:

$\forall \rightarrow \forall x(M(x) \rightarrow W(x))$

This assertion can be translated as "If x is a mammal, then x is warm-blooded for all x". The symbol →→ describes the conditional relationship between the two predicates and represents the logical connective "implies".

The answer to this example is that we have represented in Predicate Logic in AI the knowledge that all mammals are warm-blooded using the symbols $M(x)$ and $W(x)$ and the conditional relationship between them using the logical connective "implies".

What are Logical Expressions in Predicate Logic?

In Predicate Logic in AI, logical expressions are assertions made up of predicates, variables, quantifiers, and logical connectives. They are used to represent relationships between things and their properties and infer and deduce from that information.

Logical connectives are used to join together various predicates or logical expressions. For example, the following are the most prevalent logical connectives in Predicate Logic in AI:

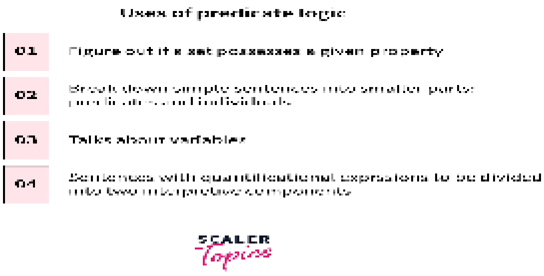
1. **"and" (∧):**
   This link expresses the logical union of two predicates. For instance, $\wedge M(x) \wedge W(x)$ represents the statement "x is a mammal and x is warm-blooded."
2. **"or" (∨):**
   This connective describes the logical disjunction of two predicates. For example, $\vee M(x) \vee B(x)$ would indicate the statement that "x is a mammal or x is a bird."

3. **"not" (KaTeX parse error: Expected 'EOF', got '¬' at position 1: ¬):**
   This connective conveys a predicate's negation. KaTeX parse error: Expected 'EOF', got '¬' at position 1: ¬M(x) for example, would indicate the statement "x is not a mammal."
4. **"implies" (→→):**
   The conditional connection between two predicates is expressed using this connective. →$M(x)$→$W(x)$ would symbolise the statement "if x is a mammal, then x is warm-blooded."

Quantifiers are words that are used to make statements about all or some of the things in a domain. "forall" (∀)and "exists" (∃) are the two most prevalent quantifiers in Predicate Logic in AI. The term "forall" refers to a statement true for all objects in a domain, whereas "exists" refers to a statement true for at least one object in a domain.

For example, to represent the assertion "all mammals are warm-blooded," we could use the statement ∀→$\forall x(M(x) \rightarrow W(x))$. The quantifier "forall" indicates that this assertion applies to all objects in the domain.

How Is Predicate Logic Used



In Artificial Intelligence, Predicate Logic is used to describe and reason about complex relationships between objects and their properties. It is especially helpful for formalizing and logically representing knowledge about the world, which can then be used to draw inferences and deductions. For example, predicate logic in AI determines whether a member of a given set holds a given property.

It breaks down basic sentences into smaller parts: predicates and individuals. Predicate logic in AI even allows you to manage generalization expressions(quantificational expressions). Predicate reasoning allows you to discuss variables(pronouns). The pronoun's value is an individual in the universe's domain that context decides.

It allows sentences containing quantificational expressions to be split into two interpretive components. For example, a simple sentence containing a variable or a pronoun (a placeholder) could be evaluated as true or false about a person taken as a value for the pronoun.

The quantificational expression directs you to restrict the domain of people under consideration to a relevant set. It indicates how many distinct pronoun values from a particular domain must be considered to determine the sentence's truth.

What Are the Quantifiers Used in Predicate Logic?

In Predicate Logic in AI, there are two primary quantifiers:

Existential Quantifier

The "Existential Quantifier" (∃):
This quantifier expresses the statement that at least one object exists in a domain that meets a particular predicate. For example, we could write $\exists\wedge\exists x(M(x)\wedge C(x))$ to indicate "there exists an object x that is both a mammal and a carnivore."

Quantifier Universal

The "Quantifier Universal" (∀):
This quantifier expresses the assertion that a predicate is true for all items in a domain. For example, to represent the assertion "all mammals are warm-blooded," we could use the statement $\forall\rightarrow\forall x(M(x)\rightarrow W(x))$

The quantifiers can be combined with logical connectives to create more complex logical expressions. For example, we could write $\exists\wedge\forall\rightarrow\exists x(M(x)\wedge\forall y(F(y)\rightarrow E(x,y)))$ to indicate "there exists an object x that is a mammal, and every fish y is an enemy of x."

What Is First-Order Predicate Logic? and Its Example?

First-order predicate logic in AI, also known as first-order logic, is a variety of formal logic that adds quantifiers and predicates to propositional logic. Predicates are used to symbolize properties of objects in first-order logic, and variables are used to represent the objects themselves. Quantifiers are words used to make statements about all or some things in a domain.

Consider the following assertion: "All dogs are mammals." In first-order logic, this assertion could be represented as follows: (Dog(x) Mammal(x)). The variable x denotes any entity in the domain, the predicate Dog(x) denotes the property of being a dog, and the predicate Mammal(x) denotes the property of being a mammal. The quantifier x conveys that this is true for all x objects in the domain.

What Is an Atomic Formula?

In predicate logic in AI, an atomic formula is an assertion that cannot be broken down into simpler statements using logical connectives or quantifiers. It comprises a single predicate symbol and an unlimited number of factors. *Cat*(*x*) and *Dog*(*y*) are atomic formulas because they cannot be broken down into simpler assertions.

A ground atomic formula is a predicate with all constant arguments. In essence, a proposition would be a predicate without supporting evidence. As a result, it is, by nature, a ground atomic formula. A non-ground atomic predicate has at least one variable argument. An atomic formula or its antithesis would be considered a literal.

Is Predicate Logic Better than Propositional Logic?

The job at hand determines whether predicate logic in AI is superior to propositional logic. Propositional logic is simpler and easy to use for simple statements without variables or quantifiers. It cannot, however, describe object relationships or reason about more complex statements. In contrast, predicate logic is more powerful and flexible, allowing for the representation and reasoning of complicated relationships between things and their properties. However, it is more complex and, for simpler statements, more challenging to use than propositional logic.

Predicate logic is generally favored in Artificial Intelligence applications that demand complex reasoning and representation of relationships between objects and their properties. Predicate logic in AI, for example, can be used in natural language processing to express the meaning of sentences more accurately and precisely than propositional logic. In addition, predicate logic can be used in database systems to describe data relationships in a more structured and organized manner.

Difference between Propositional Logic and Predicate Logic

Logical reasoning forms the basis for a huge domain of computer science and mathematics. They help in establishing mathematical arguments, valid or invalid.

1. Propositional Logic :

A proposition is basically a declarative sentence that has a truth value. Truth value can either be true or false, but it needs to be assigned any of the two values and not be ambiguous. The purpose of using propositional logic is to analyze a statement, individually or compositely.

For example :

The following statements :

1. If x is real, then x2 > 0
2. What is your name?
3. (a+b)2 = 100
4. This statement is false.
5. This statement is true.

Are not propositions because they do not have a truth value. They are ambiguous.

But the following statements :

1. (a+b)2 = a2 + 2ab + b2
2. If x is real, then x2 >= 0
3. If x is real, then x2 < 0
4. The sun rises in the east.
5. The sun rises in the west.

Are all propositions because they have a specific truth value, true or false.

The branch of logic that deals with proposition is propositional logic.

2. Predicate Logic :

Predicates are properties, additional information to better express the subject of the sentence. A quantified predicate is a proposition , that is, when you assign values to a predicate with variables it can be made a proposition.

For example :

In P(x) : x>5, x is the subject or the variable and '>5' is the predicate.

P(7) : 7>5 is a proposition where we are assigning values to the variable x, and it has a truth value, i.e. True.

The set of values that the variables of the predicate can assume is called the Universe or Domain of Discourse or Domain of Predicate.

Difference between Propositional Logic and Predicate Logic :

	Propositional Logic	**Predicate Logic**
1	Propositional logic is the logic that deals with a collection of declarative statements which have a truth value, true or false.	Predicate logic is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
2	It is the basic and most widely used logic. Also known as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
3	A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables' value.
4	Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate. There are three quantifiers : Universal Quantifier (∀) depicts for all, Existential Quantifier (∃) depicting there exists some and Uniqueness Quantifier (∃!) depicting exactly one.
5	Propositions are combined with Logical Operators or Logical Connectives like Negation(¬), Disjunction(∨), Conjunction(∧), Exclusive OR(⊕), Implication(⇒), Bi-Conditional or Double Implication(⇔).	Predicate Logic adds by introducing quantifiers to the existing proposition.
6	It is a more generalized	It is a more specialized

	representation.	representation.
7	It cannot deal with sets of entities.	It can deal with set of entities with the help of quantifiers.

Monotonic Reasoning:

In monotonic reasoning, once the conclusion is taken, then it will remain the same even if we add some other information to existing information in our knowledge base. In monotonic reasoning, adding knowledge does not decrease the set of prepositions that can be derived.

To solve monotonic problems, we can derive the valid conclusion from the available facts only, and it will not be affected by new facts.

Monotonic reasoning is not useful for the real-time systems, as in real time, facts get changed, so we cannot use monotonic reasoning.

Monotonic reasoning is used in conventional reasoning systems, and a logic-based system is monotonic.

Any theorem proving is an example of monotonic reasoning.

Example:

- **Earth revolves around the Sun.**

It is a true fact, and it cannot be changed even if we add another sentence in knowledge base like, "The moon revolves around the earth" Or "Earth is not round," etc.

Advantages of Monotonic Reasoning:

- In monotonic reasoning, each old proof will always remain valid.
- If we deduce some facts from available facts, then it will remain valid for always.

Disadvantages of Monotonic Reasoning:

- We cannot represent the real world scenarios using Monotonic reasoning.

- Hypothesis knowledge cannot be expressed with monotonic reasoning, which means facts should be true.
- Since we can only derive conclusions from the old proofs, so new knowledge from the real world cannot be added.

6. Non-monotonic Reasoning

In Non-monotonic reasoning, some conclusions may be invalidated if we add some more information to our knowledge base.

Logic will be said as non-monotonic if some conclusions can be invalidated by adding more knowledge into our knowledge base.

Non-monotonic reasoning deals with incomplete and uncertain models.

"Human perceptions for various things in daily life, "is a general example of non-monotonic reasoning.

Example: Let suppose the knowledge base contains the following knowledge:

- **Birds can fly**
- **Penguins cannot fly**
- **Pitty is a bird**

So from the above sentences, we can conclude that **Pitty can fly**.

However, if we add one another sentence into knowledge base "**Pitty is a penguin**", which concludes "**Pitty cannot fly**", so it invalidates the above conclusion.

Advantages of Non-monotonic reasoning:

- For real-world systems such as Robot navigation, we can use non-monotonic reasoning.
- In Non-monotonic reasoning, we can choose probabilistic facts or can make assumptions.

Disadvantages of Non-monotonic Reasoning:

- In non-monotonic reasoning, the old facts may be invalidated by adding new sentences.

- It cannot be used for theorem **proving**.

Monotonic Reasoning vs Non-monotonic Reasoning

	Monotonic Reasoning	**Non-Monotonic Reasoning**
1	Monotonic Reasoning is the process which does not change its direction or can say that it moves in the one direction.	Non-monotonic Reasoning is the process which changes its direction or values as the knowledge base increases.
2	Monotonic Reasoning deals with very specific type of models, which has valid proofs.	Non-monotonic reasoning deals with incomplete or not known facts.
3	The addition in knowledge won't change the result.	The addition in knowledge will invalidate the previous conclusions and change the result.
4	In monotonic reasoning, results are always true, therefore, set of prepositions will only increase.	In non-monotonic reasoning, results and set of prepositions will increase and decrease based on condition of added knowledge.
5	Monotonic Reasoning is based on true facts.	Non-monotonic Reasoning is based on assumptions.
6	Deductive Reasoning is the type of monotonic reasoning.	Abductive Reasoning and Human Reasoning is a non-monotonic type of reasoning.