## What is Knowledge Representation?

1 
- Knowledge Representation in AI describes the representation of knowledge.
- Basically, it is a study of how the beliefs, intentions, and judgments of an intelligent agent can be expressed suitably for automated reasoning.
- One of the primary purposes of Knowledge Representation includes modeling intelligent behavior for an agent.

## What is Knowledge Representation?

- Knowledge Representation and Reasoning (KR, KRR) represents information from the real world for a computer to understand and then utilize this knowledge to solve complex real-life problems like communicating with human beings in natural language.
- Knowledge representation in AI is not just about storing data in a database, it allows a machine to learn from that knowledge and behave intelligently like a human being.

## What is Knowledge Representation?

- The different kinds of knowledge that need to be represented in AI include:
  - Objects
  - Events
  - Performance
  - Facts
  - Meta-Knowledge
  - Knowledge-base

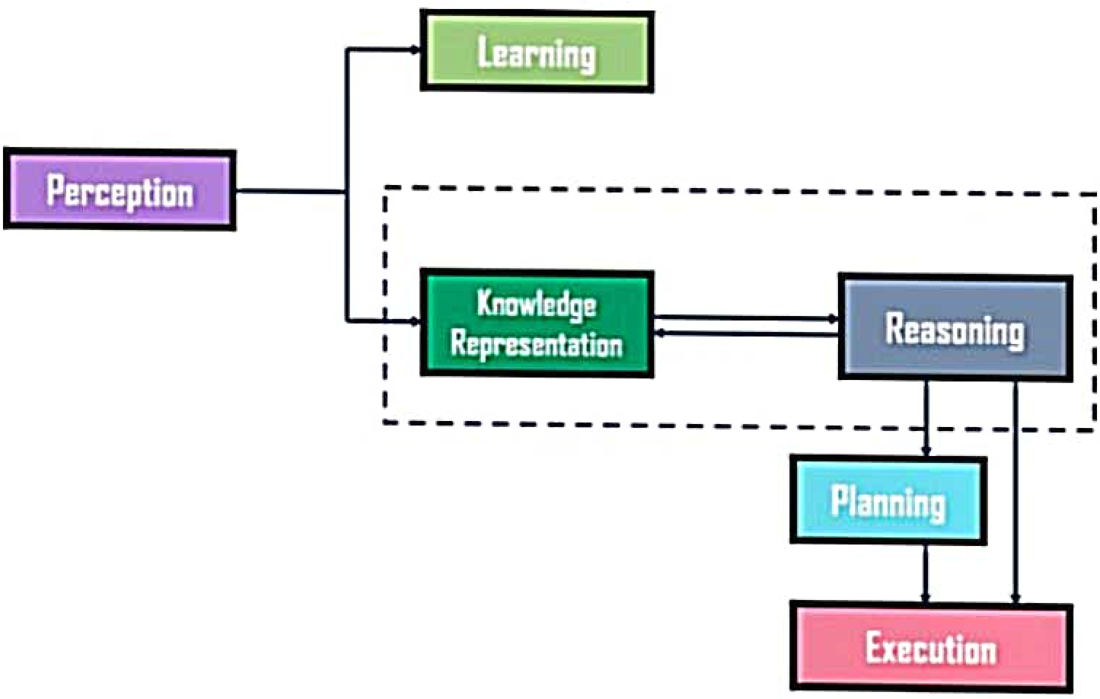
In summary, knowledge and intelligence are related but distinct concepts. Knowledge is acquired information or understanding, while intelligence is the ability to learn, reason, and solve problems. Knowledge can be represented in systems using formal languages or artificial intelligence techniques.

# Cycle of Knowledge Representation in AI

- Artificial Intelligent Systems usually consist of various components to display their intelligent behavior. Some of these components include:
  - Perception
  - Learning
  - Knowledge Representation & Reasoning
  - Planning
  - Execution



# Example:



## Example

- The Perception component retrieves data or information from the environment.
- with the help of this component, you can retrieve data from the environment, find out the source of noises and check if the AI was damaged by anything.
- Also, it defines how to respond when any sense has been detected.



## Example

- Then, there is the Learning Component that learns from the captured data by the perception component.
- The goal is to build computers that can be taught instead of programming them. Learning focuses on the process of self-improvement.
- In order to learn new things, the system requires knowledge acquisition, inference, acquisition of heuristics, faster searches, etc.



## Example

- The main component in the cycle is Knowledge Representation and Reasoning which shows the human-like intelligence in the machines.
- Knowledge representation is all about understanding intelligence.
- Instead of trying to understand or build brains from the bottom up, its goal is to understand and build intelligent behavior from the top-down and focus on what an agent needs to know in order to behave intelligently.
- Also, it defines how automated reasoning procedures can make this knowledge available as needed.

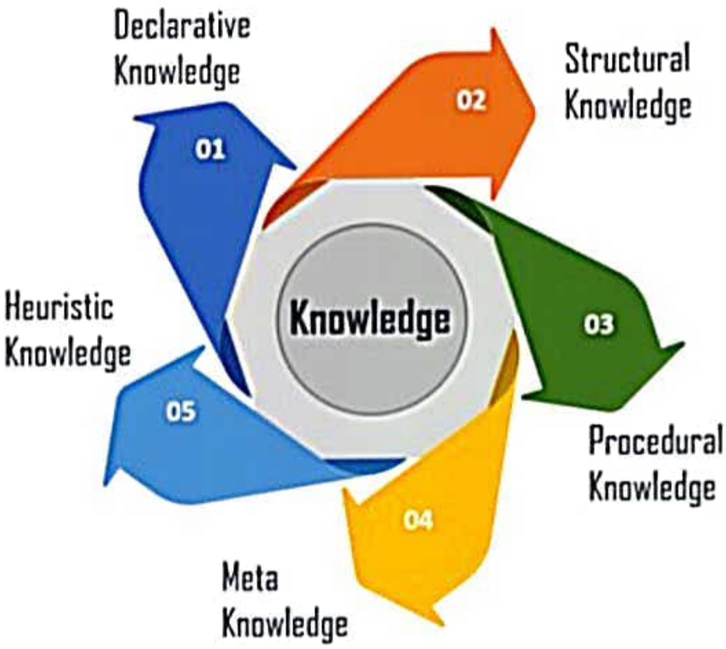


## Example

- The Planning and Execution components depend on the analysis of knowledge representation and reasoning.
- Here, planning includes giving an initial state, finding their preconditions and effects, and a sequence of actions to achieve a state in which a particular goal holds.
- Now once the planning is completed, the final stage is the execution of the entire process.

# Types of Knowledge

3






# Types of Knowledge

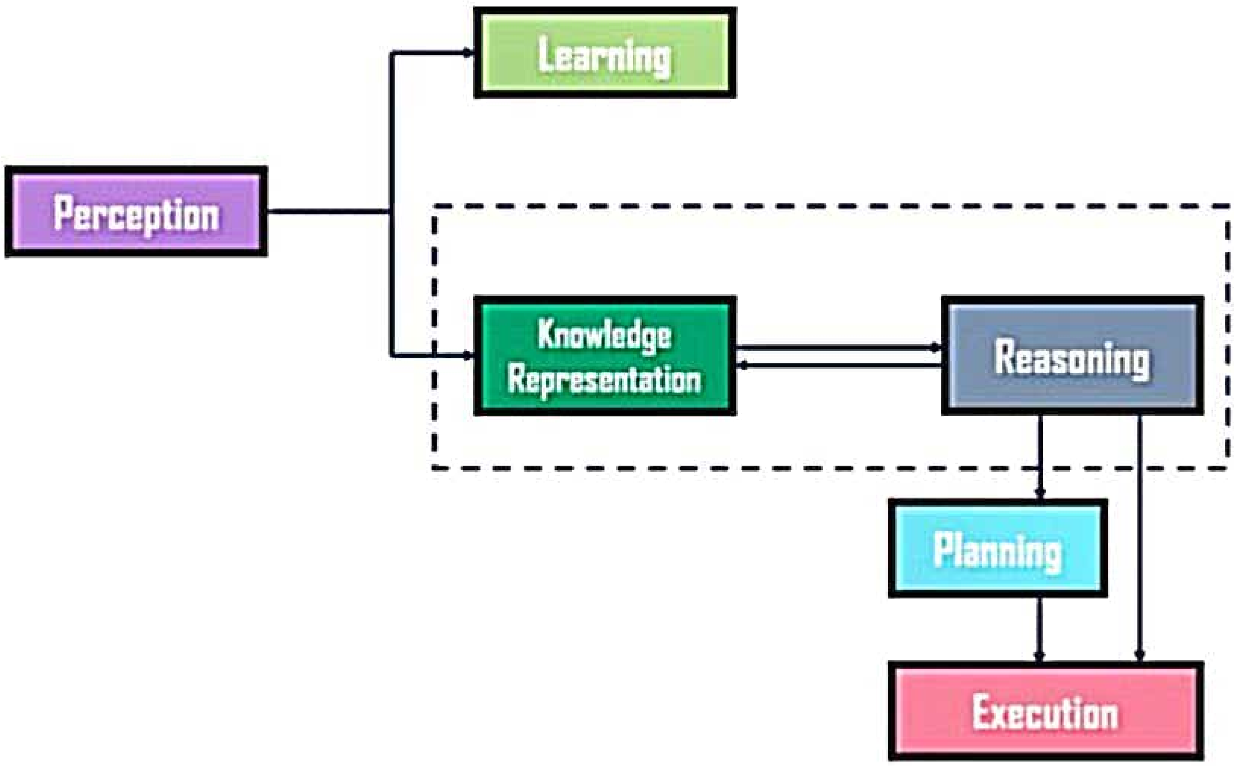
- Declarative Knowledge – It includes concepts, facts, and objects and expressed in a declarative sentence.
- Structural Knowledge – It is a basic problem-solving knowledge that describes the relationship between concepts and objects.
- Procedural Knowledge – This is responsible for knowing how to do something and includes rules, strategies, procedures, etc.
- Meta Knowledge – Meta Knowledge defines knowledge about other types of Knowledge.
- Heuristic Knowledge – This represents some expert knowledge in the field or subject.

4

- Artificial Intelligent Systems usually consist of various components to display their intelligent behavior. Some of these components include:
  – Perception
  – Learning
  – Knowledge Representation & Reasoning
  – Planning
  – Execution



## Example:



# Knowledge Representation in AI

4




# Logical Representation

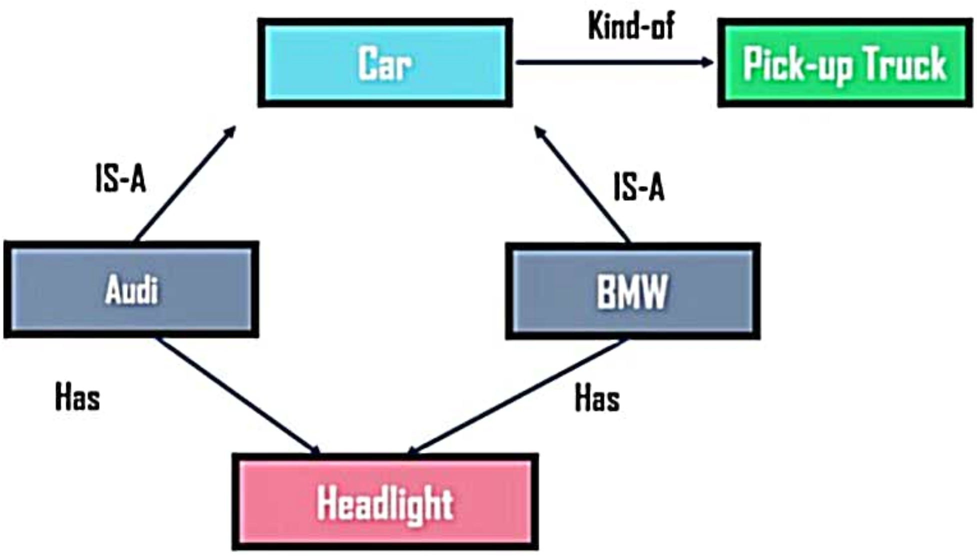
- Logical representation is a language with some definite rules which deal with propositions and has no ambiguity in representation.
- It represents a conclusion based on various conditions and lays down some important communication rules.
- Also, it consists of precisely defined syntax and semantics which supports the sound inference.
- Each sentence can be translated into logics using syntax and semantics.

## Semantic Network Representation

- Semantic networks work as an alternative of predicate logic for knowledge representation. In Semantic networks, you can represent your knowledge in the form of graphical networks.
- This network consists of nodes representing objects and arcs which describe the relationship between those objects. Also, it categorizes the object in different forms and links those objects.
- This representation consist of two types of relations:
  - IS-A relation (Inheritance)
  - Kind-of-relation



## Semantic Network Representation






Semantic Network Representation

# Frame Representation

- A frame is a record like structure that consists of a collection of attributes and values to describe an entity in the world.
- These are the AI data structure that divides knowledge into substructures by representing stereotypes situations.
- Basically, it consists of a collection of slots and slot values of any type and size.
- Slots have names and values which are called facets.

# Production Rules

- In production rules, agent checks for the condition and if the condition exists then production rule fires and corresponding action is carried out.
- The condition part of the rule determines which rule may be applied to a problem. Whereas, the action part carries out the associated problem-solving steps. This complete process is called a recognize-act cycle.
- The production rules system consists of three main parts:
  - The set of production rules
  - Working Memory
  - The recognize-act-cycle

5 Propositional logic plays a crucial role in artificial intelligence (AI) by providing a formal framework for representing and reasoning about knowledge and beliefs. In AI, propositional logic is used to model the knowledge of agents, to represent the state of a system, and to reason about the consequences of actions.

Syntax of Propositional Logic in AI:
In AI, propositional logic is used to represent the knowledge of agents in the form of logical expressions or sentences. These sentences are constructed using propositional variables, logical connectives, and quantifiers.

Propositional variables represent propositions or statements that can be either true or false, and they are used to represent the state of the system. For example, let us consider a simple example of an agent in a room with two doors, labeled "A" and "B". We can represent the state of the agent with the following propositional variables:

- A: The agent is at door A
- B: The agent is at door B

Logical connectives are used to connect propositional variables and form more complex sentences. In AI, the three main logical connectives are conjunction (^), disjunction (v), and negation (~). For example, the sentence "The agent is at door A and the door A is open" can be represented in propositional logic as follows:

- A ^ D_A

where D_A represents the statement "The door A is open".

Semantics of Propositional Logic in AI:

The semantics of propositional logic in AI is concerned with the interpretation and meaning of logical expressions or sentences. In AI, logical expressions are interpreted as assertions about the state of the system, and the truth or falsity of these assertions can be determined using truth tables or inference rules.

For example, consider the following logical expressions:

- A ^ D_A: The agent is at door A and the door A is open
- ~B v D_B: The agent is not at door B or the door B is open

These expressions can be evaluated using truth tables, which list all possible combinations of truth values for the propositional variables and the resulting truth values for the logical expressions. For example, the truth table for the expression A ^ D_A is:

| **A** | **D_A** | **A ^ D_A** |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

This table shows that the expression A ^ D_A is true only when the agent is at door A and the door A is open.

This table shows that the expression A ^ D_A is true only when the agent is at door A and the door A is open.

Inference rules are used in AI to derive new knowledge from existing knowledge using logical reasoning. For example, the rule of modus ponens can be used to infer a new sentence from two existing sentences:

- If A then B
- A

Therefore, B

In conclusion, propositional logic is a fundamental tool in AI for representing and reasoning about knowledge and beliefs. Its syntax and semantics are essential for understanding the behavior of intelligent agents and for designing intelligent systems.

6

Propositional logic is a formal system for representing and reasoning about propositions, which are statements that can be either true or false. In AI, propositional logic is used to represent knowledge by defining a set of propositional symbols (also called atomic propositions or propositional variables) and combining them using logical connectives to form more complex statements, called compound propositions.

Atomic propositions are the basic building blocks of propositional logic, and they represent simple statements that can be either true or false. For example, consider the following atomic propositions:

- P: "The sky is blue."
- Q: "It is raining."
- R: "The temperature is below freezing."

These atomic propositions can be used to represent simple facts about the world that an intelligent agent might know.

Compound propositions, on the other hand, are formed by combining atomic propositions using logical connectives such as negation, conjunction, disjunction, conditional, and biconditional. For example, the following compound propositions can be formed using the atomic propositions above:

- $\neg P$: "The sky is not blue."
- $P \wedge Q$: "It is both raining and the sky is blue."
- $R \vee \neg Q$: "Either the temperature is below freezing or it is not raining."
- $P \rightarrow Q$: "If the sky is blue, then it is raining."
- $P \leftrightarrow R$: "The sky is blue if and on ... low freezing."

Regenerate response

- ¬P: "The sky is not blue."
- P ∧ Q: "It is both raining and the sky is blue."
- R ∨ ¬Q: "Either the temperature is below freezing or it is not raining."
- P → Q: "If the sky is blue, then it is raining."
- P ↔ R: "The sky is blue if and only if the temperature is below freezing."

These compound propositions can be used to represent more complex relationships between facts or to express logical implications and equivalences.

In conclusion, propositional logic provides a powerful framework for representing knowledge in AI by defining a set of atomic propositions and combining them using logical connectives to form compound propositions. This allows intelligent agents to reason about the world and make decisions based on their knowledge and beliefs.

| 7 | 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 (∃) |

| | | |
|---|---|---|
| 4 | Scope analysis is not done in propositional logic. | 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 representation. | It is a more specialized representation. |
| 7 | It cannot deal with sets of entities. | It can deal with set of entities with the help of quantifiers. |

## Universal Quantifier:

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.

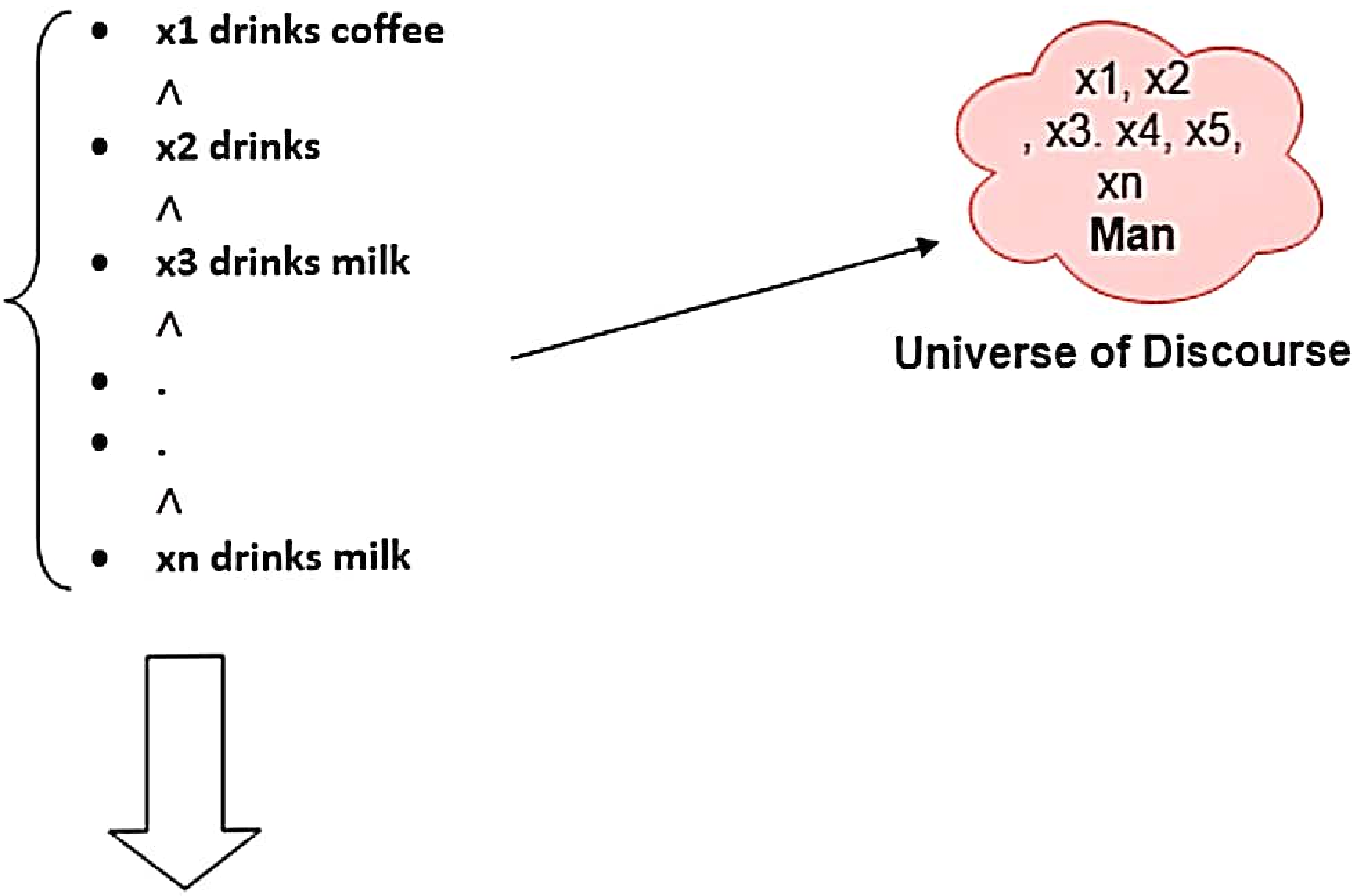
> **Note: In universal quantifier we use implication "→**

If x is a variable, then ∀x is read as:

- **For all x**
- **For each x**
- **For every x.**

**All man drink coffee.**

Let a variable x which refers to a cat so all x can be represented in UOD as below:




So in shorthand notation, we can write it as :

**$\forall x\ man(x) \rightarrow drink(x, coffee)$.**

It will be read as: There are all x where x is a man who drink coffee.

# Existential Quantifier:



Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.

It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.

> **Note: In Existential quantifier we always use AN**
> **Conjunction symbol (∧).**

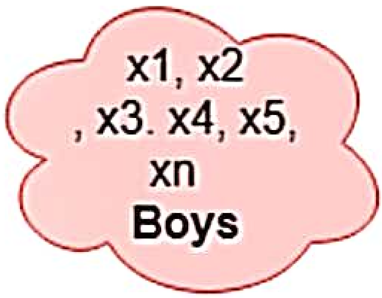
If x is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

- **There exists a 'x.'**
- **For some 'x.'**
- **For at least one 'x.'**

# Example:

**Some boys are intelligent.**

- **x1 is intelligent**
  **V**
- **x2 is intelligentV**
- **x3 is intelligent**
  **V**
- .
- .
  **V**
- **xn is intelligent**




**Universe of Discourse**

So in short-hand notation, we can write it as:

**∃x: boys(x) ∧ intelligent(x)**

i) All men are clever

9

M(x): x is a man

C(x): x is clever

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

This statement can be read as "For all x, if x is a man, then x is clever".

ii) All students like either Machine learning or Web Technologies

S(x): x is a student

M(x): x likes Machine learning

W(x): x likes Web Technologies

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

This statement can be read as "For all x, if x is a student, then x likes either Machine learning or Web Technologies".

iii) There are some students who like deep learning

D(x): x likes deep learning

$\exists x(S(x) \wedge D(x))$

This statement can be read as "There exists an x such that x is a student and x likes deep learning".

Regenerate response

This statement can be read as "There exists an x such that x is a student and x likes deep learning".

iv) No rectangle is a square.

R(x): x is a rectangle
S(x): x is a square

$\neg \exists x(R(x) \land S(x))$

This statement can be read as "There does not exist an x such that x is both a rectangle and a square".

a. **Atomic Propositions** 10

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:**

a) 2+2 is 4, it is an atomic proposition as it is a **true** fact.

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:**

a) "It is raining today, and street is wet."

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 |