~~Abhay~~

UNIT 2

What is Knowledge Representation in AI

Humans are great at tasks that require creativity, critical thinking, and empathy. They can learn from experience and adapt to new situations, and they possess emotional intelligence that allows them to understand and connect with other people on a deep level.

On the other hand, Artificial Intelligence or AI is excellent at tasks that require speed, accuracy, and scalability. It can quickly process vast amounts of data and perform complex calculations and analyses far beyond human capabilities.

Knowledge representation is a crucial element of Artificial Intelligence. It is believed that an intelligent system needs to have an explicit representation of its knowledge to reason and make decisions.

Knowledge representation provides a framework for representing, organizing, and manipulating knowledge that can be used to solve complex problems, make decisions, and learn from data.

For example, when you see a hot tea cup, a signal immediately comes from your brain cautioning you against picking it up. If we were to make AI more sophisticated(or humanist), we would be required to feed them with more and often complex information about our world to perform the complex task, which leads to the concept of Knowledge Representation in Artificial Intelligence.

- An intelligent agent needs **knowledge** about the real world for taking decisions and **reasoning** to act efficiently.
- Knowledge-based agents are those agents who have the capability of **maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and take actions. These agents can represent the world with some formal representation and act intelligently**.
- Knowledge-based agents are composed of two main parts:
  - **Knowledge-base and**
  - **Inference system**.

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).

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.

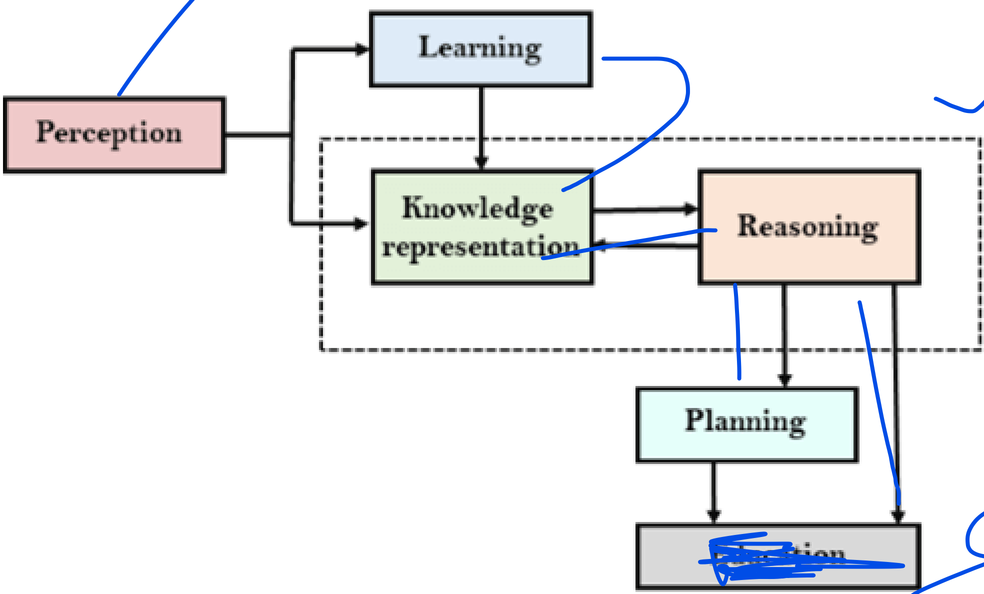
5. Structural knowledge:

- 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.

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 shows the interaction of an AI system with the **real world** and the **components** involved in showing intelligence.

- 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.
- 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.
- 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.
- 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.

Approaches to Knowledge Representation

1. Simple Relational Knowledge

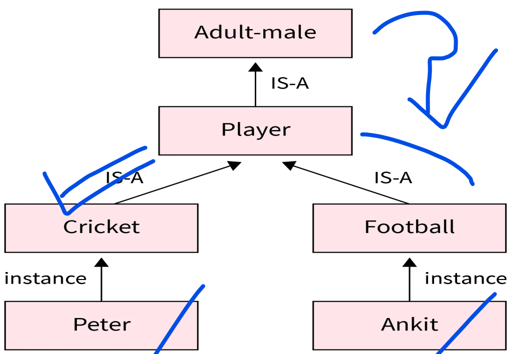
- This type of knowledge uses relational methods to store facts.
- It is one of the simplest types of knowledge representation.
- The facts are systematically set out in terms of rows and columns.
- This type of knowledge representation is used in database systems where the relationship between different entities is represented.
- There is a low opportunity for inference.

Example : The following is the simple relational knowledge representation

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

2. **Inheritable Knowledge**

- Inheritable knowledge in AI refers to knowledge acquired by an AI system through learning and can be transferred or inherited by other AI systems.
- This knowledge can include models, rules, or other forms of knowledge that an AI system learns through training or experience.
- In this approach, all data must be stored in a hierarchy of classes.
- Boxed nodes are used to represent objects and their values.
- We use Arrows that point from objects to their values.
- Rather than starting from scratch, an AI system can inherit knowledge from other systems, allowing it to learn faster and avoid repeating mistakes that have already been made. Inheritable knowledge also allows for knowledge transfer across domains, allowing an AI system to apply knowledge learned in one domain to another.



3. **Inferential Knowledge**

- Inferential knowledge refers to the ability to draw logical conclusions or make predictions based on available data or information
- In artificial intelligence, inferential knowledge is often used in machine learning algorithms, where models are trained on large amounts of data and then used to make predictions or decisions about new data.
- For example, in image recognition, a machine learning model can be trained on a large dataset of labeled images and then used to predict the contents of new images that it has never seen before. The model can draw inferences based on the patterns it has learned from the training data.
- It represents knowledge in the form of formal logic.

Example: Statement 1: Alex is a footballer.

Statement 2: All footballers are athletes. Then it can be represented as;

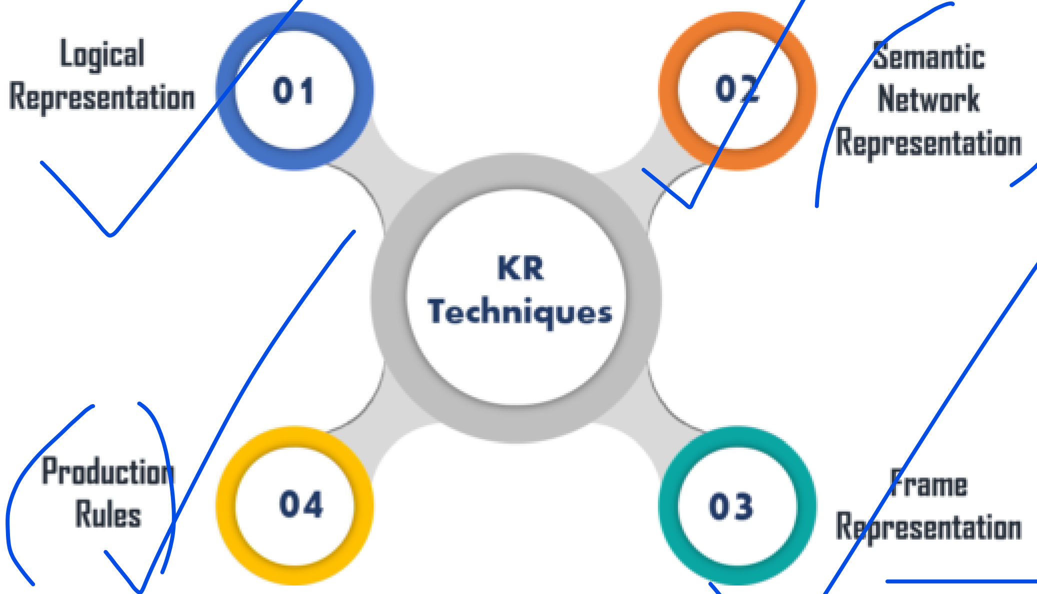
Footballer(Alex) $\boxed{\forall x}$ = Footballer (x) ———-> Athelete (x)s

4. **Procedural Knowledge:**

- In artificial intelligence, procedural knowledge refers to the knowledge or instructions required to perform a specific task or solve a problem.
- This knowledge is often represented in algorithms or rules dictating how a machine processes data or performs tasks.
- For example, in natural language processing, procedural knowledge might involve the steps required to analyze and understand the meaning of a sentence. This could include tasks such as identifying the parts of speech in the sentence, identifying relationships between different words, and determining the overall structure and meaning of the sentence.
- One of the most important rules used is the If-then rule.
- This knowledge allows us to use various coding languages such as LISP and Prolog.
- Procedural knowledge is an important aspect of artificial intelligence, as it allows machines to perform complex tasks and make decisions based on specific instructions.

Techniques of Knowledge Representation in AI

There are four techniques of representing knowledge such as:



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.

Syntax	Semantics
• It decides how we can construct legal sentences in logic. • It determines which symbol we can use in knowledge representation. • Also, how to write those symbols.	• Semantics are the rules by which we can interpret the sentence in the logic. • It assigns a meaning to each sentence.

Logical representation can be categorised into mainly two logics:

a) Propositional Logics
b) Predicate logics

Advantages:

- Logical representation helps to perform logical reasoning.
- This representation is the basis for the programming languages.

Disadvantages:

- Logical representations have some restrictions and are challenging to work with.

- This technique may not be very natural, and inference may not be very efficient.

Propositional Logic: This type of logical representation is also known as propositional calculus or statement logic. This works in a Boolean, i.e., True or False method.

First-order Logic: This type of logical representation is also known as the First Order Predicate Calculus Logic (FOPL). This logical representation represents the objects in quantifiers and predicates and is an advanced version of propositional logic.

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.

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.

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:

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

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**

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

Limitations of Propositional Logic

- **Limited expressivity:** Propositional logic is limited in its ability to represent complex relationships between objects or concepts. It can only express simple propositional statements with binary truth values (true/false). This makes it difficult to represent concepts such as uncertainty, ambiguity, and vagueness.
- **Inability to handle quantifiers:** Propositional logic is unable to handle quantifiers such as "all" or "some." For example, it cannot represent the statement "all humans are mortal" in a concise manner. This makes it difficult to reason about sets of objects or concepts.
- **Lack of support for negation:** Propositional logic does not provide an easy way to represent negation. This can make it difficult to represent negative statements and reason about them.

Predicate Logic in AI (Artificial Intelligence)

What is Predicate Logic in AI? Predicate logic in artificial intelligence, also known as first-order logic or first order predicate logic in AI, is a formal system used in logic and mathematics to represent and reason about complex relationships and structures. It plays a crucial role in knowledge representation, which is a field within artificial intelligence and philosophy concerned with representing knowledge in a way that machines or humans can use for reasoning and problem-solving.

$\wedge$	*and* [conjunction]
$\vee$	*or* [disjunction]
$\Rightarrow$	*implies* [implication]
$\neg$	*not* [negation]
$\forall$	*For all*
$\exists$	*There exists*

Basic Components of Predicate Logic

1. **Predicates:** Predicates are statements or propositions that can be either true or false depending on the values of their arguments. They represent properties, relations, or characteristics of objects. For example, "IsHungry(x)" can be a predicate, where "x" is a variable representing an object, and the predicate evaluates to true if that object is hungry.

2. **Variables:** Variables are symbols that can take on different values. In predicate logic, variables are used to represent objects or entities in the domain of discourse. For example, "x" in "IsHungry(x)" can represent any object in the domain, such as a person, animal, or thing.

3. **Constants:** Constants are specific values that do not change. They represent particular objects in the domain. For instance, in a knowledge base about people, "Alice" and "Bob" might be constants representing specific individuals.

4. **Quantifiers:** Quantifiers are used to specify the scope of variables in logical expressions. There are two main quantifiers in predicate logic:

- Existential Quantifier (∃): Denoted as ∃, it indicates that there exists at least one object for which the statement within the quantifier is true. For example, "∃x IsHungry(x)" asserts that there is at least one object that is hungry.
- Universal Quantifier (∀): Denoted as ∀, it indicates that the statement within the quantifier is true for all objects in the domain. For example, "∀x IsHuman(x) → IsMortal(x)" asserts that all humans are mortal.

Rules of inference in artificial intelligence

Rules of inference are a set of logical principles and deductive rules that draw conclusions from existing information or assertions.

Types of inference rules

Following are the types of inference rules:

1. Modus ponens
2. Modus tollens
3. Hypothetical syllogism
4. Disjunctive syllogism
5. Addition
6. Simplification
7. Resolution

Before we begin, let's suppose the following statements for all the types mentioned below:

P: It is raining.

Q: The streets are wet.

R: Roads are slippery.

Modus ponens

If P implies Q and P is true, then Q is true.

Notation

$((P \rightarrow Q) \wedge P) \Rightarrow Q((P \rightarrow Q) \wedge P) \Rightarrow Q$

Example

If it is raining, then the streets are wet (P -> Q), and it is raining (P); therefore, the streets are wet (Q).

Modus tollens

If P implies Q and Q is false, then P is false.

Notation

$((P \rightarrow Q) \wedge \sim Q) \Rightarrow \sim P((P \rightarrow Q) \wedge \sim Q) \Rightarrow \sim P$

Example

If it is raining, then the streets are wet (P -> Q), and streets are not wet (~Q); therefore, it is not raining (~P).

Hypothetical syllogism

If P implies Q and Q implies R, then P implies R.

Notation

$((P \to Q) \land (Q \to R)) \Rightarrow (P \to R)((P \to Q) \land (Q \to R)) \Rightarrow (P \to R)$

Example

If it is raining, then the streets are wet (P -> Q), and if the streets are wet, then roads are slippery (Q -> R); therefore, if it is raining, then roads are slippery (P -> R).

Disjunctive syllogism

If P or Q is true, and P is false, then Q is true.

Notation

$((P \lor Q) \land \sim P) \Rightarrow Q((P \lor Q) \land \sim P) \Rightarrow Q$

Example

It is raining or streets are wet (P∨Q), and it is not raining (~P); therefore, streets are wet (Q).

Addition

If P is true, then P or Q is true.

Notation

$P \Rightarrow (P \lor Q)P \Rightarrow (P \lor Q)$

Example

It is raining (P), therefore it is raining or streets are wet (P∨Q).

Simplification

If P and Q is true, then P is true.

Notation

$(P \land Q) \Rightarrow P(P \land Q) \Rightarrow P$

Example

It is raining and streets are wet (P∧Q); therefore it is raining (P).

Resolution

If both P or Q and not P or R is true, then Q or R is true.

Notation

$((P \vee Q) \wedge (\sim P \vee R)) \Rightarrow (Q \vee R)((P \vee Q) \wedge (\sim P \vee R)) \Rightarrow (Q \vee R)$

Example

It is raining or streets are wet (P∨Q) and it is not raining or roads are slippery (~P∨R); therefore streets are wet or roads are slippery (Q∨R).

Applications

Inference is a fundamental process that significantly impacts various applications, including natural language processing, expert systems, robotics, and computer vision.

Natural language processing (NLP): Based on context and prior knowledge, the inference understands the meaning of sentences.

Computer vision: The inference recognizes object as an image, based on patterns and features.

Robotics: Plans and actions are executed by the inference based on the environment.