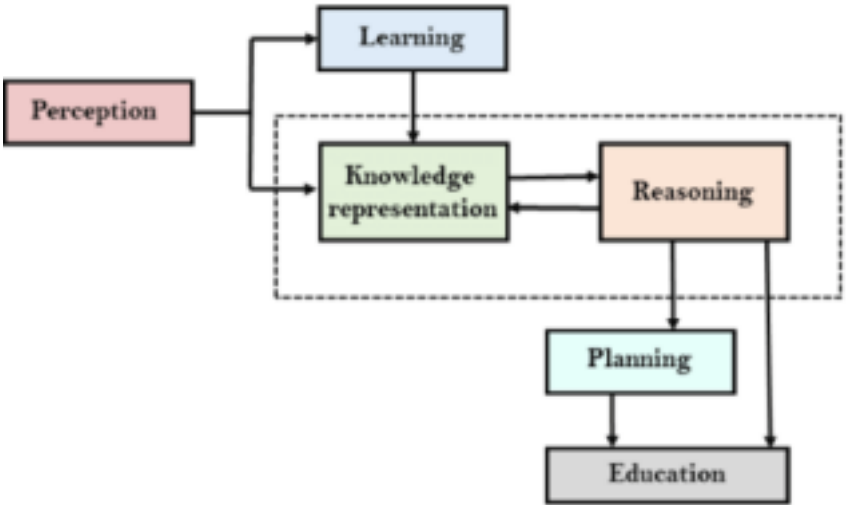
Subject: AI

UNIT III: Knowledge Representation

1. Explain AI Knowledge Cycle

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

o Perception
o Learning
o Knowledge Representation and Reasoning
o Planning
o 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.

2. What are the approaches in KR

Ans : There are mainly four approaches to knowledge representation, which are given below: **1. Simple relational knowledge:**

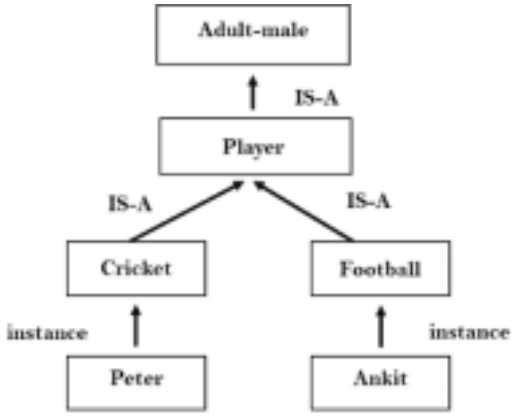
- o It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
- o This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
- o 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:

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



3. Inferential knowledge:

o Inferential knowledge approach represents knowledge in the form of formal logics. o This approach can be used to derive more facts.

o It guaranteed correctness.

o **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:

o Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.

o In this approach, one important rule is used which is **If-Then rule**.

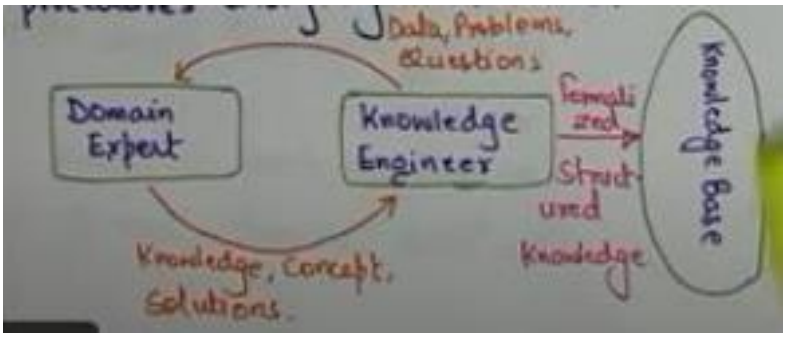
o In this knowledge, we can use various coding languages such as **LISP language** and **Prolog language**.

o We can easily represent heuristic or domain-specific knowledge using this approach. o But it is not necessary that we can represent all cases in this approach.

3. What is knowledge Acquisition its Task, and Knowledge Engineering

Ans : Knowledge Acquisition →

In artificial intelligence, knowledge acquisition is the process of gathering, selecting, and interpreting information and experiences to create and maintain knowledge within a specific domain. It is a key component of machine learning and knowledge-based systems.



There are many different methods of knowledge acquisition, including rule-based systems, decision trees, artificial neural networks, and fuzzy logic systems. The most appropriate method for a given application depends on the nature of the problem and the type of data available.

Rule-based systems are the simplest form of knowledge-based system. They use a set of rules, or heuristics, to make decisions. Decision trees are another common method, which use a series of if-then-else statements to arrive at a decision.

Artificial neural networks are a more complex form of knowledge-based system, which mimic the way the human brain learns. They are able to learn from data and make predictions based on that data. Fuzzy logic systems are another type of complex knowledge-based system, which use fuzzy set theory to make decisions.

Knowledge Acquisition Task →

- Collect—from Experts
- Interpret-Review and identify key information
- Analysis-Create Theory/strategies to solve problem
- Design-For better understanding

Knowledge Engineering →

- Knowledge engineering is a field of artificial intelligence (AI) that tries to emulate the judgment and behavior of a human expert in a given field.
- Knowledge engineering is the technology behind the creation of expert systems to assist with issues related to their programmed field of

knowledge. Expert systems involve a large, expandable knowledge base integrated with a rules engine that specifies how to apply information to each particular situation.

- Knowledge engineers may create a system that incorporates machine learning so that it can learn from experience in the same way humans do. Expert systems are used in various fields, including healthcare, customer service, financial services, manufacturing and law.

- Knowledge engineering is a branch of artificial intelligence (AI) that develops rules that are applied to data in order to imitate the thought process of a human that is an expert on a specific topic

4. What are the issues in knowledge representation

Ans : The fundamental goal of knowledge Representation is to facilitate inference (conclusions) from knowledge.

The issues that arise while using KR techniques are many. Some of these are explained below.

Important Attributed:

• Any attribute of objects so basic that they occur in almost every problem domain? • There are two attributed "instance" and "isa" that are general significance. These attributes are important because they support property inheritance.

Relationship among attributes:

• Any important relationship that exists among object attributed?

• The attributes we use to describe objects are themselves entities that we represent. • The relationship between the attributes of an object, independent of specific knowledge they encode, may hold properties like:

1. **Inverse** — this is about consistency check, while a value is added to one attribute. The entities are related to each other in many different ways.
2. **Existence in an isa hierarchy** — this is about generalization-specification, like, classes of objects and specialized subsets of those classes, there are attributes and specialization of attributes. For example, the attribute height is a specialization of general attribute physical-size which is, in turn, a specialization of physical attribute. These generalization-specialization relationships are important for attributes because they support inheritance.
3. **Technique for reasoning about values** — this is about reasoning values of attributes not given explicitly. Several kinds of information are used in reasoning, like, height: must be in a unit of length, Age: of a person cannot be greater than the age of person's parents. The values are often specified when a knowledge base is created.
4. **Single valued attributes** — this is about a specific attribute that is guaranteed to take a unique value. For example, a baseball player can at time have only a single height and be a member of only one team. KR systems take different approaches to provide support for single valued attributes.

5. Explain Knowledge Acquisition techniques/ strategies.

Ans : **Knowledge Acquisition Techniques/Stratergies**

Many techniques have been developed to deduce knowledge from an expert. They are termed as knowledge acquisition techniques. They are:

- Introspection—Expert act as both Expert and Knowledge
- Observation- Closely observe, readymade resource
- Induction
- Protocol Analysis-psychology term expert can asked to perform some task and verbalize thought process
- Prototyping-Expert + K.E=system
- Interviewing
- Diagram Based Techniques
- Matrix Based Techniques
- Hierarchy-Generation Techniques
- Protocol Analysis Techniques
- Protocol Generation Techniques
- Sorting Techniques

7. Explain the concept of predicate logic and propositional logic with suitable examples.

Ans : Predicate logic and propositional logic are two fundamental branches of mathematical logic used in philosophy, computer science, and mathematics to represent and analyze logical statements and arguments. They serve different purposes and have distinct characteristics:

1. Propositional Logic:
Propositional logic deals with propositions or statements that are either true or false. It focuses on the relationships between these propositions using logical connectives. Propositional logic is useful for simplifying complex statements and analyzing arguments. The basic components of propositional logic are:

a. Propositions: These are statements that can be true or false. They are represented by variables like P, Q, or A.

b. Logical Connectives: These are symbols used to combine or modify propositions. The main logical connectives in propositional logic are:

- Conjunction (AND): Represented by ∧ (ampersand), it combines two propositions and is true only if both propositions are true. For example:

P: It is raining.
Q: I have an umbrella.
P ∧ Q: It is raining, and I have an umbrella.

- Disjunction (OR): Represented by ∨ (pipe symbol), it combines two propositions and is true if at least one of them is true. For example:

P: It is sunny.
Q: It is raining.
P ∨ Q: It is sunny, or it is raining.

- Negation (NOT): Represented by ¬ (tilde or exclamation mark), it negates a proposition, making it true if the original proposition is false, and vice versa. For example:

P: The computer is working.
¬P: The computer is not working.

c. Implication (→) and Biconditional (↔): These are also important connectives in propositional logic but not as fundamental as conjunction, disjunction, and negation. Implication represents "if...then" statements, while biconditional represents "if and only if" statements.

2. Predicate Logic (First-Order Logic):
Predicate logic, also known as first-order logic, extends propositional logic by introducing variables, predicates, and quantifiers. It is more expressive and allows for the representation of more complex statements involving objects and relationships. The key components of predicate logic are:

a. Predicates: These are functions or relations that can take one or more variables as arguments and return a truth value (true or false). Predicates are typically represented by uppercase letters and can be followed by variables. For example:

P(x): x is a prime number.

Q(x, y): x is greater than y.

b. Quantifiers: Predicate logic uses quantifiers to specify the scope of variables. The two main quantifiers are:

- Universal Quantifier (∀): Represents "for all" or "for every." It indicates that a statement holds true for all values of a variable. For example:

∀x P(x): All x are prime numbers.

- Existential Quantifier (∃): Represents "there exists" or "there is at least one." It indicates that there exists at least one value of a variable for which a statement is true. For example:

∃x Q(x, 5): There exists an x greater than 5.

c. Complex Statements: In predicate logic, you can build complex statements by combining predicates, variables, and quantifiers. For example:

∀x (P(x) → Q(x)): For all x, if x is a prime number, then x is greater than 5.

In summary, propositional logic deals with simple true/false statements and their logical relationships, while predicate logic extends this to handle more complex statements involving variables, predicates, and quantifiers to represent relationships and properties of objects and elements in a more detailed and expressive manner.

8. How the knowledge is categorized? Briefly explain different types of knowledge. Also mention the issues in knowledge representation.

Ans : **Types of knowledge**

Following are the various types of knowledge:

1. Declarative Knowledge:

o Declarative knowledge is to know about something.

o It includes concepts, facts, and objects.

o It is also called descriptive knowledge and expressed in declarativesentences. o It is simpler than procedural language.

2. Procedural Knowledge

o It is also known as imperative knowledge.

o Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.

o It can be directly applied to any task.

o It includes rules, strategies, procedures, agendas, etc.

o Procedural knowledge depends on the task on which it can be applied.

3. Meta-knowledge:

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

4. Heuristic knowledge:

o Heuristic knowledge is representing knowledge of some experts in a filed or subject. o 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:

o Structural knowledge is basic knowledge to problem-solving.

o It describes relationships between various concepts such as kind of, part of, and grouping of something.

o It describes the relationship that exists between concepts or objects.

Issues in Knowledge Representation:

The fundamental goal of knowledge Representation is to facilitate inference (conclusions) from knowledge.

The issues that arise while using KR techniques are many. Some of these are explained below.

Important Attributed:

• Any attribute of objects so basic that they occur in almost every problem domain? • There are two attributed "instance" and "isa" that are general significance. These attributes are important because they support property inheritance.

Relationship among attributes:

• Any important relationship that exists among object attributed?

• The attributes we use to describe objects are themselves entities that we represent. • The relationship between the attributes of an object, independent of specific knowledge they encode, may hold properties like:

1. **Inverse** — this is about consistency check, while a value is added to one attribute. The entities are related to each other in many different ways.
2. **Existence in an isa hierarchy** — this is about generalization-specification, like, classes of objects and specialized subsets of those classes, there are attributes and specialization of attributes. For example, the attribute height is a specialization of general attribute physical-size which is, in turn, a specialization of physical attribute. These generalization-specialization relationships are important for attributes because they

support inheritance.

3. **Technique for reasoning about values** — this is about reasoning values of attributes not given explicitly. Several kinds of information are used in reasoning, like, height: must be in a unit of length, Age: of a person cannot be greater than the age of person's parents. The values are often specified when a knowledge base is created.

4. **Single valued attributes** — this is about a specific attribute that is guaranteed to take a unique value. For example, a baseball player can at time have only a single height and be a member of only one team. KR systems take different approaches to provide support for single valued attributes.

9. Explore the term resolution. Provide the basics of resolution in propositional logic.

Ans : Resolution is a fundamental technique used in propositional logic and first-order logic for proving the validity or satisfiability of logical formulas. It is a rule of inference that allows us to derive new logical conclusions from existing formulas. Resolution is primarily used in automated theorem proving, artificial intelligence, and logic programming.

Here are the basics of resolution in propositional logic:

1. **Clausal Form**:
Before applying resolution, logical formulas are typically converted into clausal form. A clausal form is a disjunction (OR) of clauses, where each clause is a conjunction (AND) of literals. A literal is either a propositional variable or its negation.

For example, suppose you have the following formulas:
- P ∧ Q
- ¬P ∨ R
- ¬R

You can convert them into clausal form as follows:
- (P, Q)
- (¬P, R)
- (¬R)

2. **Resolution Rule**:
The resolution rule is based on the idea of finding a contradiction (i.e., a clause containing both a literal and its negation) in a set of clauses. When you have two clauses that contain complementary literals (i.e., one contains a literal, and the other contains its negation), you can resolve them to derive a new clause. The resolution process involves three steps:

a. **Selection**: Choose two clauses that contain complementary literals. These clauses are called the parent clauses.

b. **Resolution**: Remove the complementary literals from the parent clauses and merge the remaining literals to form a new clause, which is the resolvent.

c. **Addition**: Add the resolvent to the set of clauses.

The resolution rule can be expressed as follows:
If (L1 ∨ ... ∨ Ln) and (¬L1 ∨ ... ∨ Lm) are two clauses containing complementary literals L1 and ¬L1, then their resolution results in the clause (L2 ∨ ... ∨ Ln ∨ L3 ∨ ... ∨ Lm), where L2, ..., Ln, L3, ..., Lm are the remaining literals.

3. **Repeat and Test for Contradiction**:

The process of selecting, resolving, and adding clauses is repeated until one of the following conditions is met:

- A contradiction (an empty clause) is derived, indicating that the original set of clauses is unsatisfiable.
- No more resolutions can be performed, and the process terminates.

4. **Conclusion**:

If a contradiction is derived, the original set of clauses is unsatisfiable, which means that the logical formula is valid. If no contradiction is found, the formula is satisfiable.

Resolution is a sound and complete proof procedure for propositional logic, meaning that if it finds a contradiction, the original formula is indeed unsatisfiable, and if it doesn't find a contradiction, the formula is satisfiable. This makes it a powerful tool for automated reasoning and theorem proving.