

Knowledge Based Agents(KBA)

Course Name: Artificial Intelligence

Course code: CSE-403 [SECTION - A]

Khandaker Jannatul Ritu, Lecturer Dept. of CSE at BAIUST

Knowledge Based Agents

- ☐ Knowledge-Based Agent in Artificial intelligence
- ☐ The architecture of knowledge-based agent
- ☐ Why Knowledge-Based Agent?
- ☐ Inference system
 1. Forward chaining
 2. Backward chaining
- ☐ Operations Performed by KBA
- ☐ A generic KBA
- ☐ Various levels of knowledge-based agent
 1. Knowledge level
 2. Logical level
 3. Implementation level
- ☐ Approaches to designing a knowledge-based agent
 1. Declarative approach
 2. Procedural approach
- ☐ What is knowledge representation?
- ☐ What to Represent?
- ☐ Types of knowledge
 1. Declarative Knowledge
 2. Procedural Knowledge
 3. Meta-knowledge
 4. Heuristic knowledge
 5. Structural knowledge

- ☐ The relation between knowledge and intelligence
- ☐ AI knowledge cycle
- ☐ Approaches to knowledge representation
 1. Simple relational knowledge
 2. Inheritable knowledge
 3. Inferential knowledge
 4. Procedural knowledge
- ☐ Requirements for knowledge Representation system
 1. Representational Accuracy
 2. Inferential Adequacy
 3. Inferential Efficiency
 4. Acquisitional efficiency
- ☐ Techniques of knowledge representation
 - Logical Representation
 - Semantic Network Representation
 - Frame Representation
 - Production Rules

Knowledge-Based Agent in Artificial intelligence

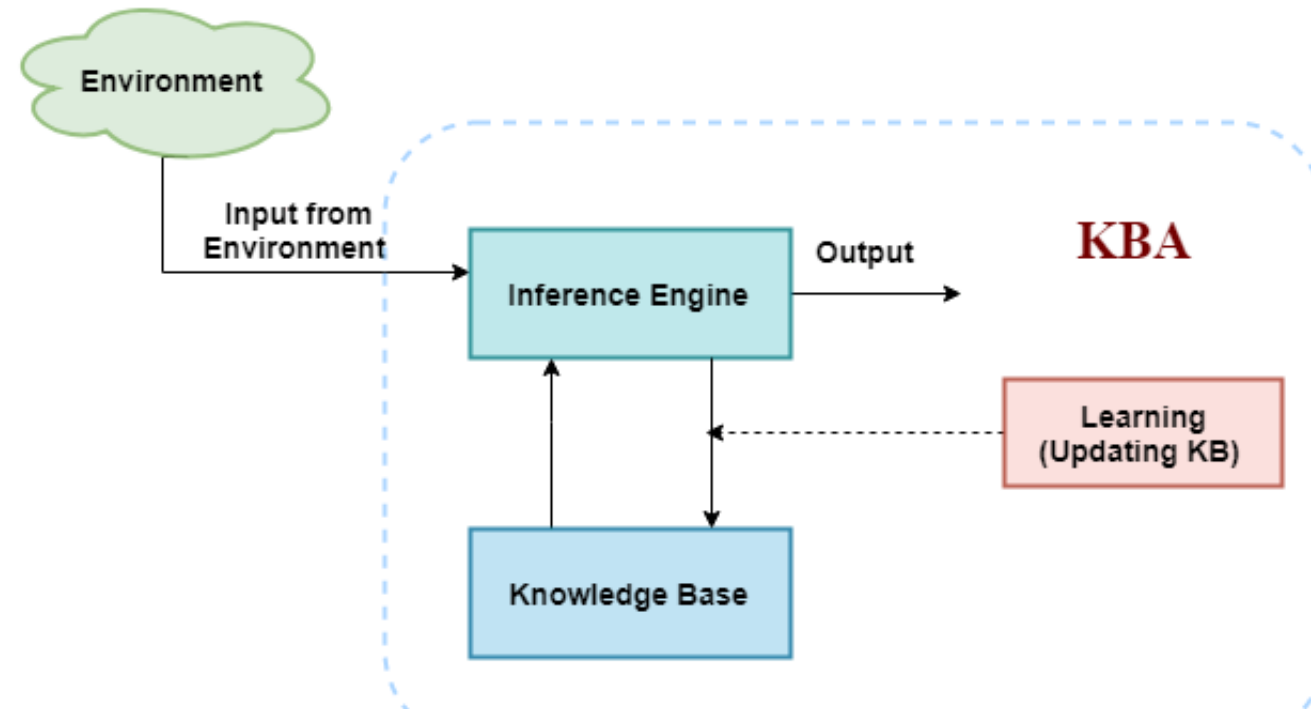
- An intelligent agent needs **knowledge** about the real world for taking decisions and **reasoning** to act efficiently.
- Knowledge-based agents are those agents who have the capability of
 1. **maintaining an internal state of knowledge,**
 2. **reason over that knowledge,**
 3. **update their knowledge after observations**
 4. **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.**
- **A knowledge-based agent must able to do the following:**
- An agent should be able to represent states, actions, etc.
- An agent should be able to incorporate new percepts
- An agent can update the internal representation of the world
- An agent can predict the internal representation of the world
- An agent can predict appropriate actions.

The Architecture Of Knowledge-based Agent:

The knowledge-based agent (KBA) take input from the environment by perceiving the environment.

The input is taken by the inference engine of the agent and which also communicate with KB to decide as per the knowledge store in KB.

The learning element of KBA regularly updates the KB by learning new knowledge.



❑ Knowledge base:

- ✓ Knowledge-base is a central component of a knowledge-based agent, it is also known as KB. It is a collection of sentences (here 'sentence' is a technical term and it is not identical to sentence in English).
- ✓ These sentences are expressed in a language which is called a knowledge representation language.
- ✓ The Knowledge-base of KBA stores fact about the world.

❑ Why use a knowledge base?

- A knowledge base **inference** is required for updating knowledge for an agent to learn with experiences and take action as per the knowledge.
- Inference means deriving new sentences from old. The inference-**based** system allows us to add a new sentence to the knowledge base. A sentence is a proposition about the world. The inference system applies logical rules to the KB to deduce new information.
- **The inference** system generates new facts so that an agent can update the KB. An inference system works mainly in two rules which are given:
 - Forward chaining
 - Backward chaining

❑ Operations Performed by KBA

Following are three operations which are performed by KBA in order to show the intelligent behavior:

- 1.**TELL:** This operation tells the knowledge base what it perceives from the environment.
- 2.**ASK:** This operation asks the knowledge base what action it should perform.
- 3.**Perform:** It performs the selected action.

A Generic Knowledge-based Agent:

```
function KB-AGENT(percept):  
    persistent: KB, a knowledge base  
        t, a counter, initially 0, indicating time  
    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))  
    Action = ASK(KB, MAKE-ACTION-QUERY(t))  
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))  
    t = t + 1  
    return action
```

The knowledge-based agent takes percept as input and returns an action as output. The agent maintains the knowledge base, KB, and it initially has some background knowledge of the real world. It also has a counter to indicate the time for the whole process, and this counter is initialized with zero.

Each time when the function is called, it performs its three operations:

- Firstly it TELLS the KB what it perceives.
 - Secondly, it asks KB what action it should take
 - Third agent program TELLS the KB that which action was chosen.
- The MAKE-PERCEPT-SENTENCE generates a sentence as setting that the agent perceived the given percept at the given time.
 - The MAKE-ACTION-QUERY generates a sentence to ask which action should be done at the current time.
 - MAKE-ACTION-SENTENCE generates a sentence which asserts that the chosen action was executed.

❑ Various levels of knowledge-based agent:

1. Knowledge level

Knowledge level is the first level of knowledge-based agent, and in this level, we need to specify what the agent knows, and what the agent goals are. With these specifications, we can fix its behavior. For example, suppose an automated taxi agent needs to go from a station A to station B, and he knows the way from A to B, so this comes at the knowledge level.

2. Logical level:

At this level, we understand that how the knowledge representation of knowledge is stored. At this level, sentences are encoded into different logics. At the logical level, an encoding of knowledge into logical sentences occurs. At the logical level we can expect to the automated taxi agent to reach to the destination B.

3. Implementation level:

This is the physical representation of logic and knowledge. At the implementation level agent perform actions as per logical and knowledge level. At this level, an automated taxi agent actually implement his knowledge and logic so that he can reach to the destination.

- **Approaches to designing a knowledge-based agent:**

- There are mainly two approaches to build a knowledge-based agent:

1. **Declarative approach:** We can create a knowledge-based agent by initializing with an empty knowledge base and telling the agent all the sentences with which we want to start with. This approach is called Declarative approach.
2. **Procedural approach:** In the procedural approach, we directly encode desired behavior as a program code. Which means we just need to write a program that already encodes the desired behavior or agent.

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).

Types Of Knowledge

1. Declarative Knowledge:

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

2. Procedural Knowledge

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

3. Meta-knowledge:

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

4. Heuristic knowledge:

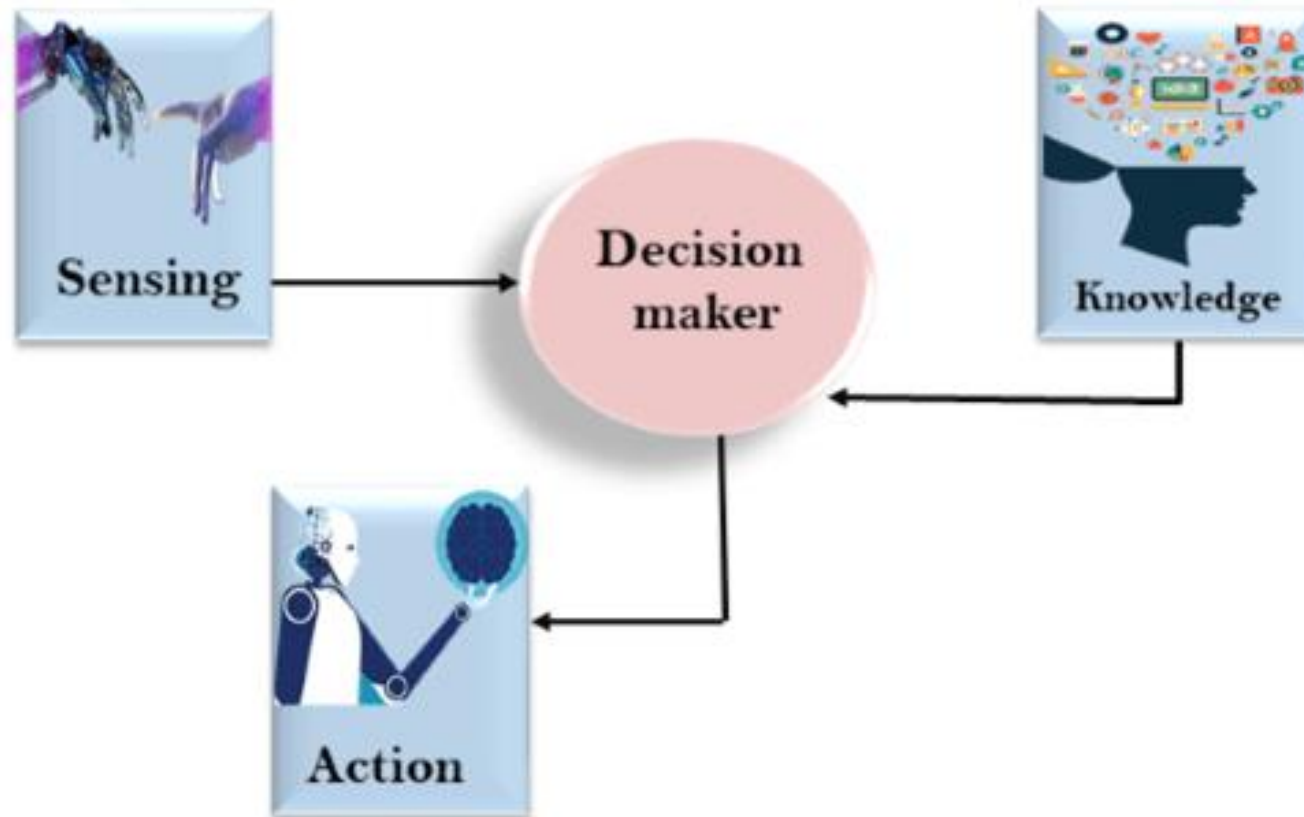
- Heuristic knowledge is representing knowledge of some experts in a field or subject.
- Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.

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.

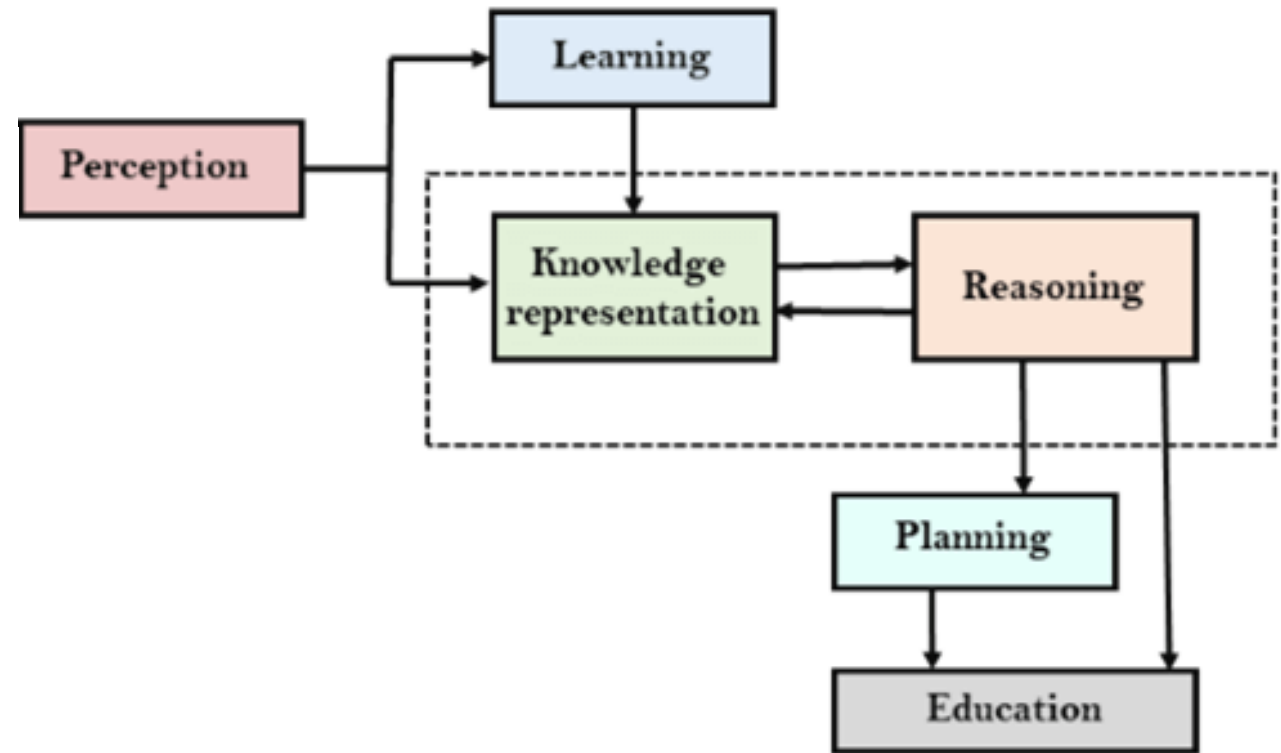


AI knowledge cycle:

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 component. 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.

❑ 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:

❑ Approaches to knowledge representation:

- 1. Simple relational knowledge
- 2. Inheritable knowledge
- 3. Inferential knowledge
- 4. Procedural knowledge

1. Simple relational knowledge:

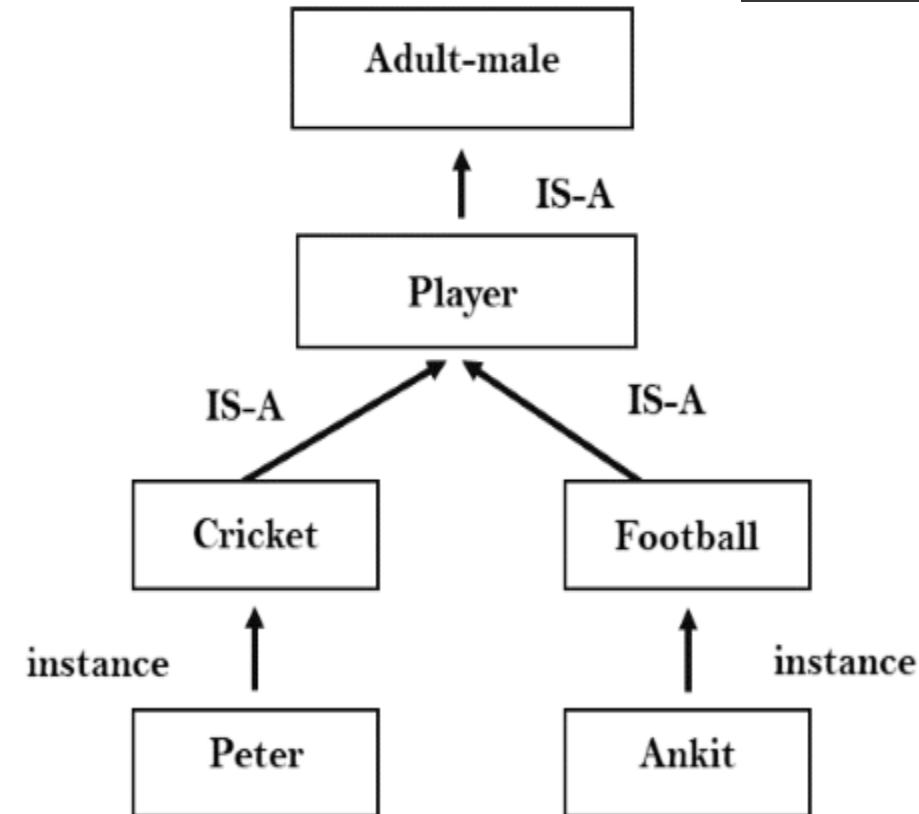
- It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
- This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.
- This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

- a. Marcus is a man
- b. All men are mortal

Then it can represent as;

`man(Marcus)`

`$\forall x = \text{man}(x) \text{ -----} \rightarrow \text{mortal}(x)$`

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. Representational Accuracy:

KR system should have the ability to represent all kind of required knowledge.

2. Inferential Adequacy:

KR system should have ability to manipulate the representational structures to produce new knowledge corresponding to existing structure.

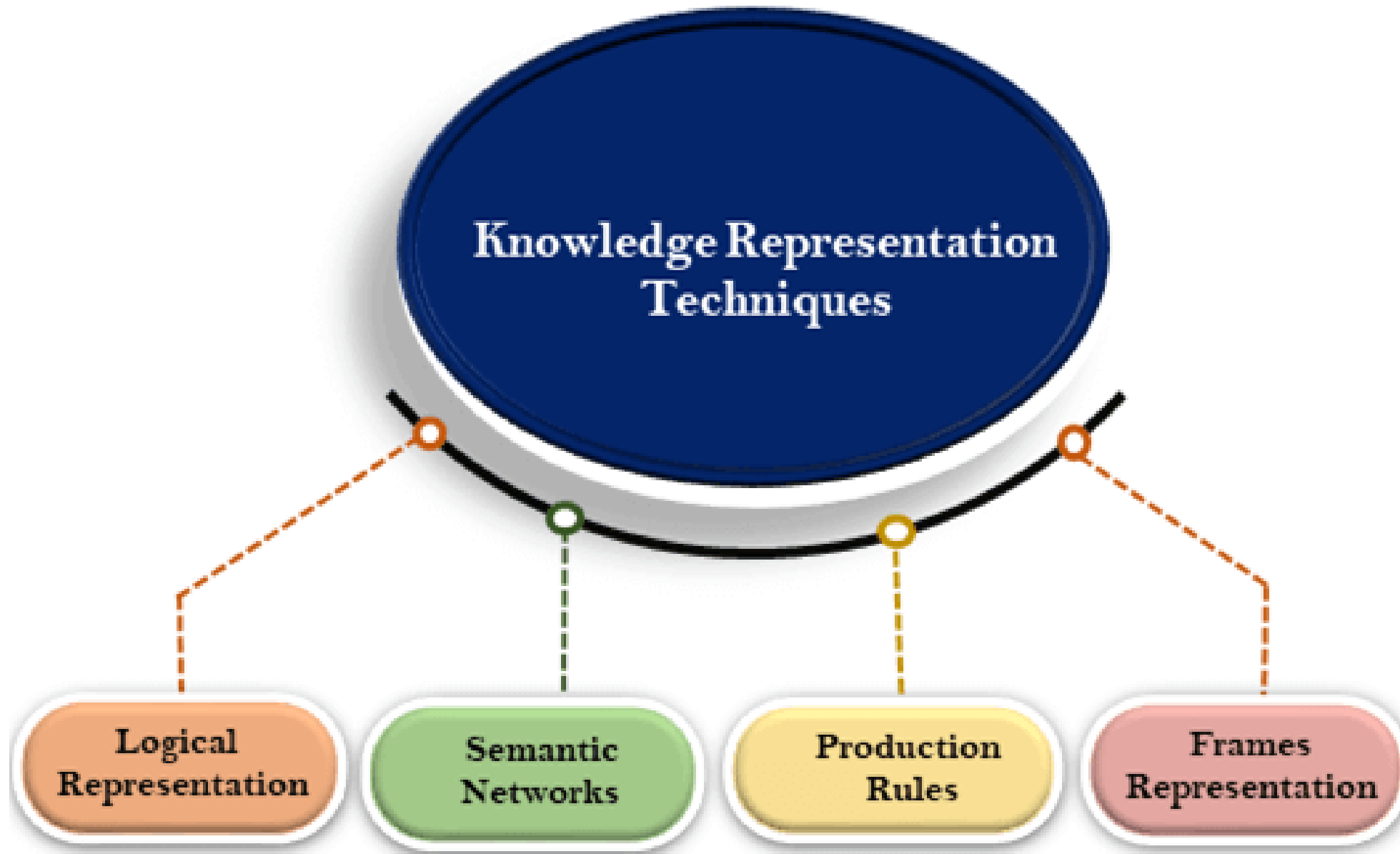
3. Inferential Efficiency:

The ability to direct the inferential knowledge mechanism into the most productive directions by storing appropriate guides.

4. Acquisitional efficiency:

The ability to acquire the new knowledge easily using automatic methods.

Techniques Of Knowledge Representation



1. Logical Representation

Logical representation is a language with some concrete rules which deals with propositions and has no ambiguity in representation. Logical representation means drawing a conclusion based on various conditions. This representation lays down some important communication rules. 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:

- Syntaxes are the rules which decide how we can construct legal sentences in the logic.
- It determines which symbol we can use in knowledge representation.
- How to write those symbols.

Semantics:

- Semantics are the rules by which we can interpret the sentence in the logic.
- Semantic also involves assigning a meaning to each sentence.

Logical representation can be categorized into mainly two logics:

- 1.Propositional Logics
- 2.Predicate logics

2. Semantic Network Representation

Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks. This network consists of nodes representing objects and arcs which describe the relationship between those objects. Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.

This representation consist of mainly two types of relations:

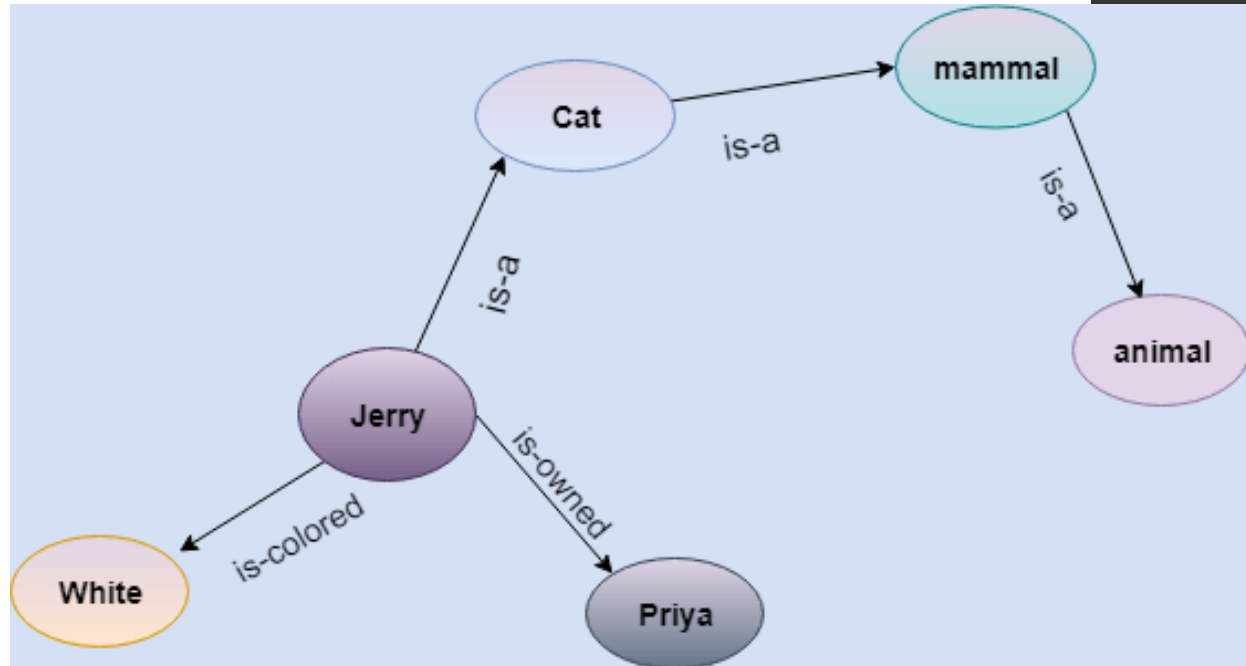
- a. IS-A relation (Inheritance)
- b. Kind-of-relation

Example: Following are some statements which we need to represent in the form of nodes and arcs.

Statements:

- 1.Jerry is a cat.
- 2.Jerry is a mammal
- 3.Jerry is owned by Priya.
- 4.Jerry is brown colored.
- 5.All Mammals are animal.

In the above diagram, we have represented the different type of knowledge in the form of nodes and arcs. Each object is connected with another object by some relation.



Drawbacks in Semantic representation:

- 1.Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
- 2.Semantic networks try to model human-like memory (Which has 10^{15} neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
- 3.These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
- 4.Semantic networks do not have any standard definition for the link names.
- 5.These networks are not intelligent and depend on the creator of the system.

Advantages of Semantic network:

- 1.Semantic networks are a natural representation of knowledge.
- 2.Semantic networks convey meaning in a transparent manner.
- 3.These networks are simple and easily understandable.

3. Frame Representation

A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world. Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations. It consists of a collection of slots and slot values. These slots may be of any type and sizes. Slots have names and values which are called facets.

Facets: The various aspects of a slot is known as **Facets**. Facets are features of frames which enable us to put constraints on the frames. Example: IF-NEEDED facts are called when data of any particular slot is needed. A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values. A frame is also known as **slot-filter knowledge representation** in artificial intelligence. Frames are derived from semantic networks and later evolved into our modern-day classes and objects. A single frame is not much useful. Frames system consist of a collection of frames which are connected. In the frame, knowledge about an object or event can be stored together in the knowledge base. The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.

Example: 1:- Let's take an example of a frame for a book

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

Example 2:
Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So following is the frame representation for this:

Slots	Filter
Name	Peter
Profession	Doctor
Age	25
Marital status	Single
Weight	78

4. Production Rules

Production rules system consist of (**condition, action**) pairs which mean, "If condition then action".

It has mainly three parts:

- The set of production rules
- Working Memory
- The recognize-act-cycle

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. And the action part carries out the associated problem-solving steps. This complete process is called a recognize-act cycle.

The working memory contains the description of the current state of problems-solving and rule can write knowledge to the working memory. This knowledge match and may fire other rules.

If there is a new situation (state) generates, then multiple production rules will be fired together, this is called conflict set. In this situation, the agent needs to select a rule from these sets, and it is called a conflict resolution.

Example:

- IF (at bus stop AND bus arrives) THEN action (get into the bus)**
- IF (on the bus AND paid AND empty seat) THEN action (sit down).**
- IF (on bus AND unpaid) THEN action (pay charges).**
- IF (bus arrives at destination) THEN action (get down from the bus).**