Knowledge Representation

Outline

- What is knowledge representation
- Different types of knowledge
- Cycle of knowledge representation
- Relation between knowledge and intelligence
- Techniques of knowledge representation
- Representation Requirements
- Approaches with example

Introduction

- Human beings are good at understanding, reasoning and interpreting knowledge.
- And using this knowledge, they are able to perform various actions in the real world. But how do machines perform the same?

What is Knowledge Representation?

- 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.

- 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.
- 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 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. For example, cars have wheels, the piano has keys, Guitars contains strings, etc.
- **Events:** Events are the actions which occur in our world. Our perception of the world is based on what we know regarding the various events that have taken place in our world. This knowledge is regarding all those events. The wars, achievements, advancement of societies, etc., are an example of this knowledge.
- **Performance:** It describe behavior which involves knowledge about how to do things. It deals with how humans and other beings and things perform certain actions in different situations. Thus, it helps in understanding the behavior side of the knowledge.

What to Represent

- **Meta-knowledge:** It is knowledge about what we know. knowledge can be divided into 3 categories: What we know, What we know that we don’t know, and knowledge that we even are unaware of and Meta knowledge deals with the first concept. Thus, meta-knowledge is the knowledge of what we know.
- **Facts:** Facts are the truths about the real world and what we represent.

Knowledge-Based Agent in AI

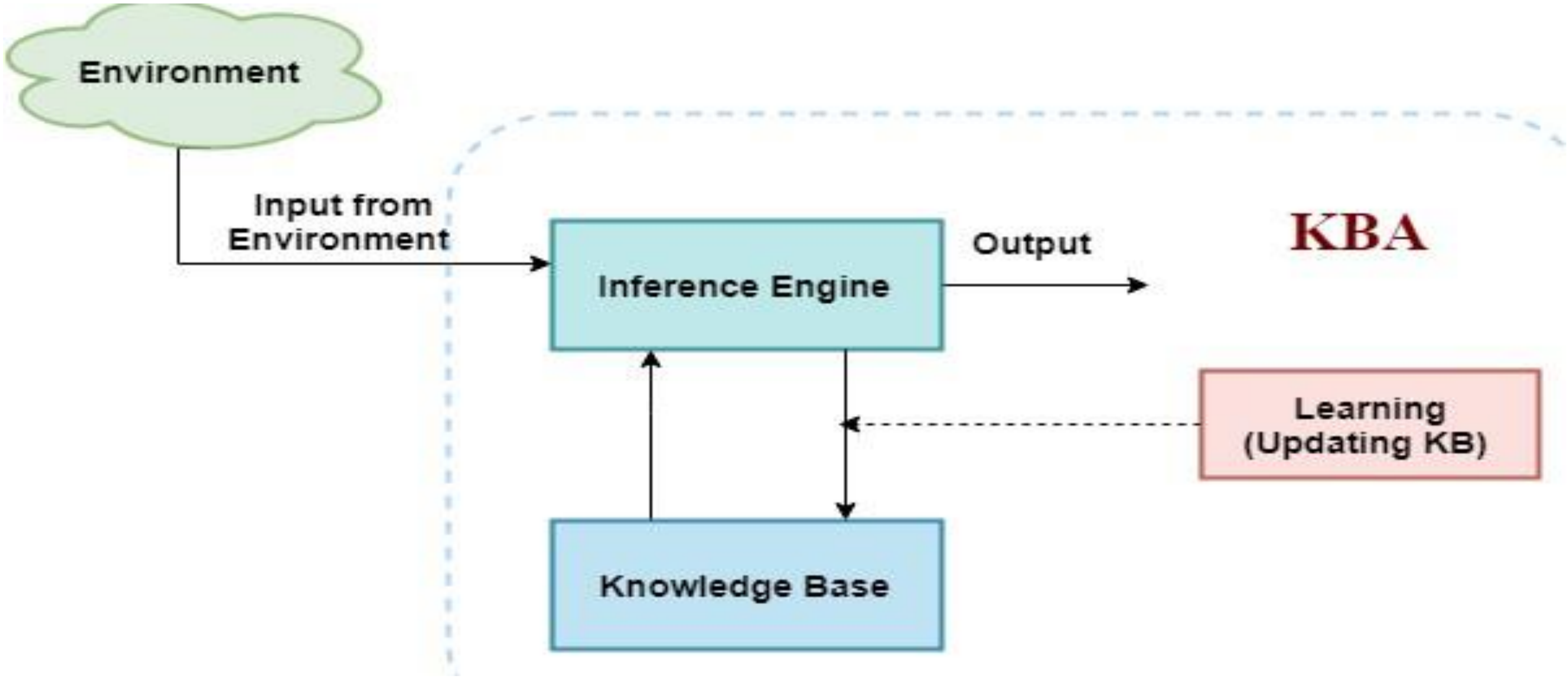
- An intelligent agent needs **knowledge** about the real world for taking decisions and **reasoning** to act efficiently.
- Knowledge-based agents are those agents who have the capability of **maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations 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

➢ Inference system.

The architecture of knowledge-based agent

- Knowledge-base is a central component of a knowledge-based agent, it is also known as KB. It is a collection of sentences. These sentences are expressed in a language which is called a knowledge representation language. The Knowledge-base of KBA stores fact about the world.



The architecture of knowledge-based agent

- **Inference system**

➢ Inference means deriving new sentences from old. Inference system allows us to add a new sentence to the knowledge base. A sentence is a proposition about the world. Inference system applies logical rules to the KB to deduce new information.

➢ Inference system generates new facts so that an agent can update the KB. An inference system works mainly in two rules which are given as:

- Forward chaining
- Backward chaining

Knowledge-based agent

- Approaches to designing a knowledge-based agent:

➢ **Declarative approach:** Declarative knowledge refers to facts or information stored in the memory, that is considered static in nature. Declarative knowledge, also referred to as conceptual, propositional or descriptive knowledge, describes things, events, or processes; their attributes; and their relation to each other.

➢ **Procedural approach:** Procedural Knowledge refers to the knowledge of how to perform a specific skill or task, and is considered knowledge related to methods, procedures, or operation of equipment.

Declarative and Procedural Knowledge

- Declarative knowledge involves knowing THAT something is the case - that J is the tenth letter of the alphabet, that Paris is the capital of France. Declarative knowledge is conscious; it can often be verbalized.
- Procedural knowledge involves knowing HOW to do something - ride a bike, for example. We may not be able to explain how we do it. Procedural knowledge involves implicit learning, which a learner may not be aware of, and may involve being able to use a particular form to understand or produce language without necessarily being able to explain it.

Declarative and Procedural Knowledge

- Real world Example: I need a cup of tea.

➢ Declarative:

1. Get me a cup of tea.

➢ Procedural:

2. Go to kitchen
3. Get sugar, milk and tea.
4. Mix them and heat over the fire till it boils
5. Put that in a cup and bring it to me

- In a declarative language, we just set the command or order and let it be on system how to complete that order. We just need our result without digging into how it should be done.

Declarative and Procedural Knowledge

- In a procedural language, we define the whole process and provide the steps how to do it. We just provide orders and define how the process will be served.
- The main difference between two approaches are, in declarative approach, we tell the computer what problem we want solved and in procedural approach, we tell the computer how to solve the problem.

Declarative and Procedural Knowledge

- Different types of knowledge can be more or less effective, given the scenario in which they're used. For example, you can score 100% in your driving theory test, yet still not be able to actually drive a car. In that case, your declarative knowledge of driving is almost useless, as you can't actually put it into practice until you have an understanding of the procedural knowledge involved in driving the car itself.
- You might know what every road sign in the US means, what every button on your dashboard does, and what lies underneath the hood, but you don't know how to parallel park or shift from 1st to 2nd gear.

Types of Knowledge

Types of Knowledge

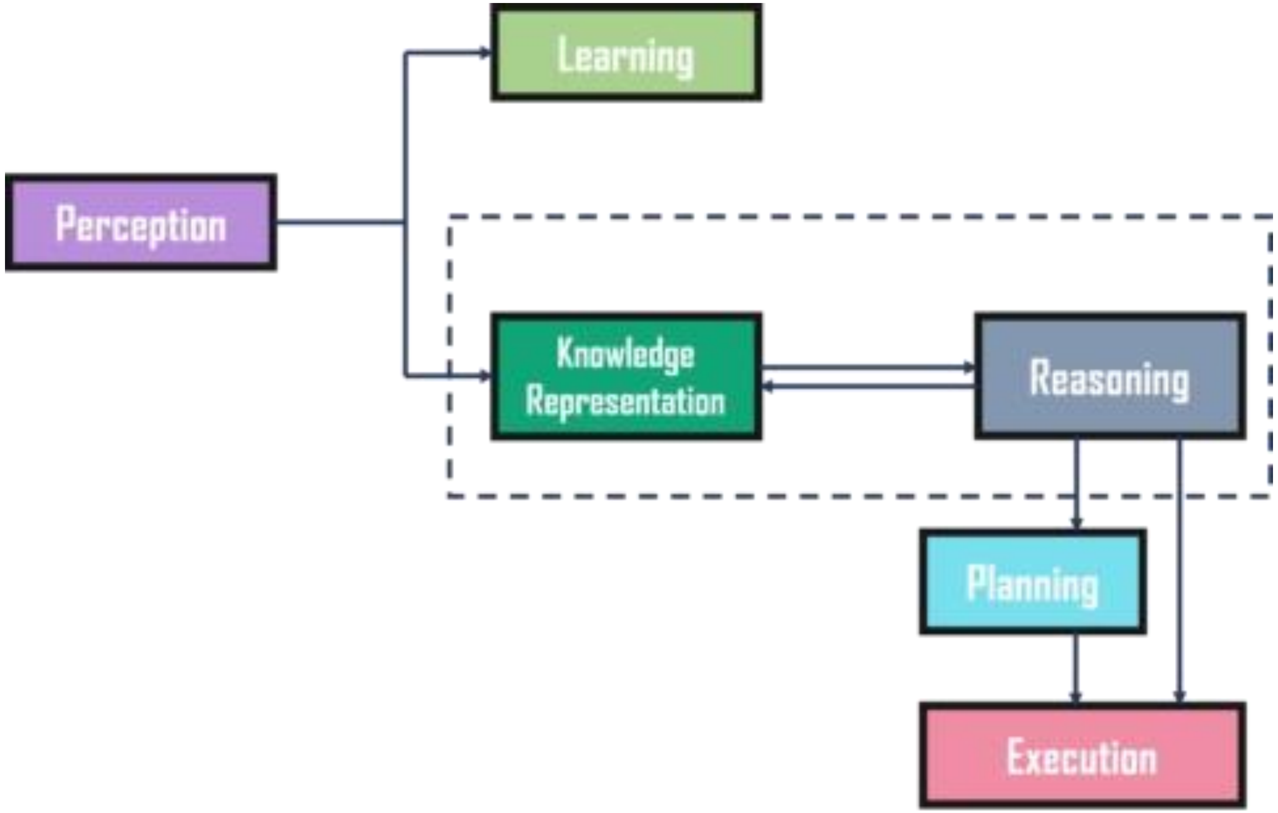
1. Declarative Knowledge
2. Procedural Knowledge
3. Meta-knowledge
4. Heuristic knowledge
5. Structural knowledge

- **1.Declarative Knowledge:**
  - It includes concepts, facts, and objects.
  - It is simpler than procedural language.
  - Ex: Mark statement of a student.
- **2. Procedural 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.
  - Ex: Steps to solve a problem statement.

- **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.
  - Ex: rule of thumb, educated guess.
- **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.

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


Perception
Learning
Knowledge Representation
Reasoning
Planning
Execution

- The **Perception** component retrieves data or information from the environment.
- with the help of this component, you can retrieve data from the environment.
- Also, it defines how to respond when any sense has been detected.

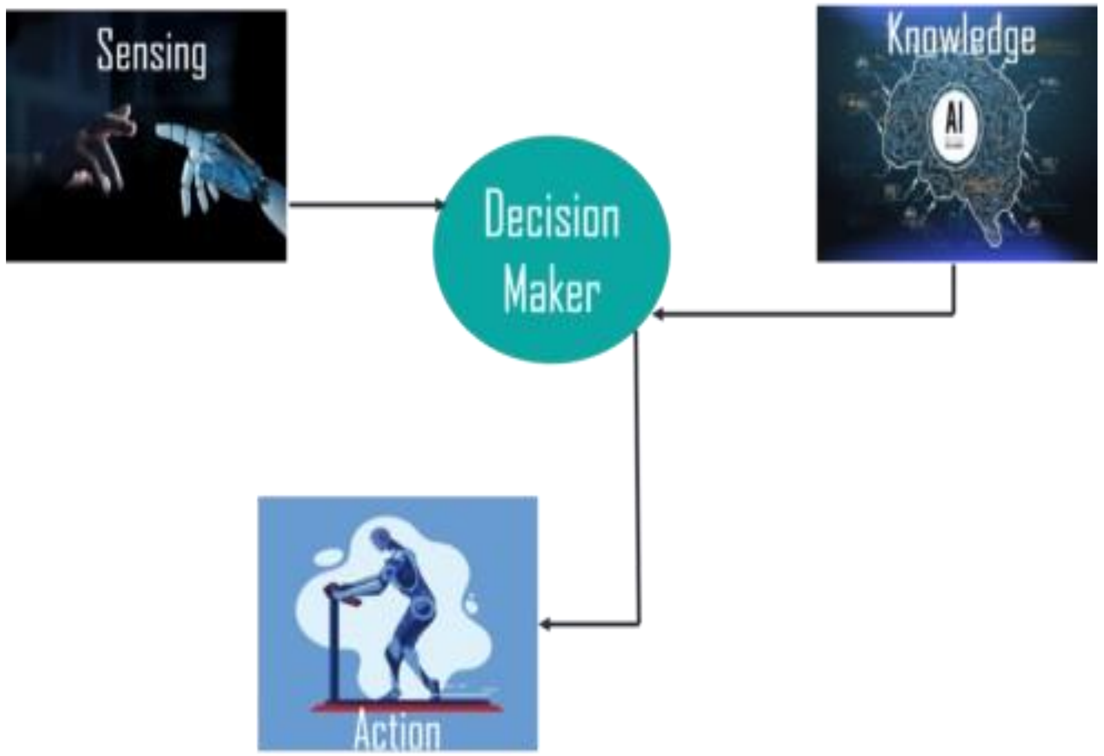
- Then, there is the **Learning** Component that learns from the captured data by the perception component.
- The goal is to build computers that can be taught instead of programming them. Learning focuses on the process of self-improvement.
- In order to learn new things, the system requires knowledge acquisition, 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 and focus on what an agent needs to know in order to behave intelligently.

- 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.

Relation between Knowledge & Intelligence

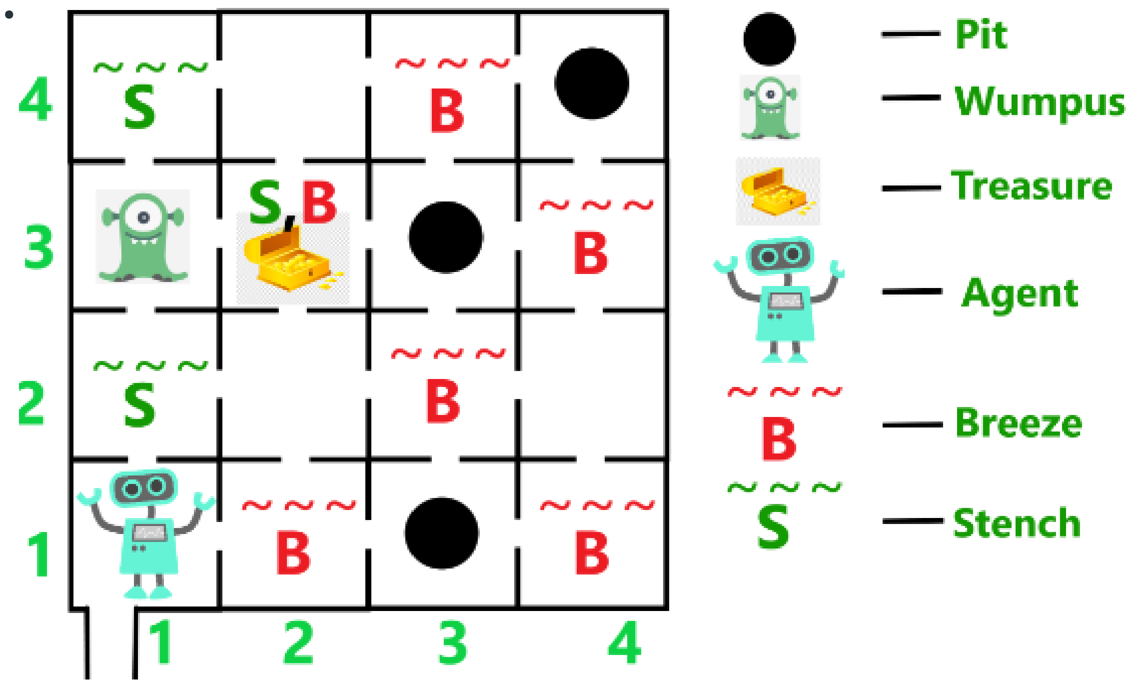
- In the real world, knowledge plays a vital role in intelligence as well as creating artificial intelligence.
- It demonstrates the intelligent behavior in AI agents or systems.
- It is possible for an agent or system to act accurately on some input only when it has the knowledge or experience about the input.


Sensing
Knowledge
AI
Decision Maker
Action

- In this example, there is one decision-maker whose actions are justified by sensing the environment and using knowledge.
- But, if we remove the knowledge part here, it will not be able to display any intelligent behavior.

The Wumpus World Environment

- The Wumpus World's agent is an example of a knowledge-based agent that represents Knowledge representation, reasoning and planning.



The Cave

- **Problem Statement:**

The Wumpus world is a cave with *16* rooms *(4×4)*.

Each room is connected to others through walkways **(no rooms are connected diagonally).**

The knowledge-based agent starts from *Room[1, 1]*. The cave has – some **pits**, a **treasure** and a beast named **Wumpus**. The Wumpus can not move but eats the one who enters its room. If the agent enters the pit, it gets stuck there. The goal of the agent is to take the treasure and come out of the cave. The agent is rewarded, when the goal conditions are met.

The agent is penalized, when it falls into a pit or being eaten by Wumpus. Some elements support the agent to explore the cave, like -The wumpus's adjacent rooms are stenchy.

The agent is given one arrow which it can use to kill the wumpus when facing it (Wumpus screams when it is killed). – The adjacent rooms of the room with pits are filled with breeze. -The treasure room is always glittery.

WW Agent PEAS Description

- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter of a gold is in same square
  - Shooting kills wumpus if agent facing it
  - Shooting uses up only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops gold in same square
- Actuators
  - Left turn, right turn, forward, grab, release, shoot
- Sensors
  - Breeze, glitter, smell, bump, scream

WW Environment Properties

- **Partially Observable:** knows only the local perceptions
- **Deterministic:** outcome is precisely specified
- **Sequential:** subsequent level of actions performed
- **Static:** Wumpus, pits are immobile
- **Discrete:** discrete environment
- **Single-agent:** The knowledge-based agent is the only single agent here whereas the wumpus is considered as the environment’s feature.

Sample Run

1,4	2,4	3,4	4,4
1,3	2,3	2,4	2,5
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

Percept: [None, None, None, None, None]

Deduce: Agent alive, so (1,1) OK
No breeze, so (1,2) and (2,1) OK

Sample Run

1,4	2,4	3,4	4,4
1,3	2,3	2,4	2,5
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

Percept: [None, None, None, None, None]

Deduce: Agent alive, so (1,1) OK
No breeze, so (1,2) and (2,1) OK

Action: Move East (turnright, goforward)

Sample Run

1,4	2,4	3,4	4,4
1,3	2,3	2,4	2,5
1,2 OK	2,2 P?	3,2	4,2
1,1 OK	2,1 A OK	3,1 P?	4,1

Percept: [None, Breeze, None, None, None]

Deduce: Pit in (2,2) or (3,1)

Sample Run

1,4	2,4	3,4	4,4
1,3	2,3	2,4	2,5
1,2 OK	2,2 P?	3,2	4,2
1,1 OK	2,1 A OK	3,1 P?	4,1

Percept: [None, Breeze, None, None, None]

Deduce: Pit in (2,2) or (3,1)

Action: Back to (1,1) then to (1,2)
(turnleft, turnleft, goforward,
turnright, goforward)

Sample Run

1,4	2,4	3,4	4,4
1,3 W	2,3	2,4	2,5
1,2 A OK	2,2 OK	3,2	4,2
1,1 OK	2,1 OK	3,1 P	4,1

Percept: [Stench, None, None, None, None]

Deduce: Wumpus in (1,3)
No pit in (2,2), pit in (3,1)

Sample Run

1,4	2,4	3,4	4,4
1,3 W	2,3	3,3	4,3
1,2 A OK	2,2 OK	3,2	4,2
1,1 OK	2,1 OK	3,1 P	4,1

Percept: [Stench, None, None, None, None]

Deduce: Wumpus in (1,3)
No pit in (2,2), pit in (3,1)

Action: Move to (2,2) (turnright, goforward)
Ignore percept for now
Move to (2,3) (turnleft, goforward)

Sample Run

1,4	2,4 P?	3,4	4,4
1,3 W	2,3 A OK G	3,3 P?	4,3
1,2 OK	2,2 OK	3,2	4,2
1,1 OK	2,1 OK	3,1 P	4,1

Percept: [Stench, Breeze, Glitter, None, None]

Deduce: Pit in (2,4) or (3,3)
Gold in (2,3)

Action: Move to (2,2) (turnright, goforward)
Ignore percept for now
Move to (2,3) (turnleft, goforward)

Sample Run

1,4	2,4 P?	3,4	4,4
1,3 W	2,3 A OK G	3,3 P?	4,3
1,2 OK	2,2 OK	3,2	4,2
1,1 OK	2,1 OK	3,1 P	4,1

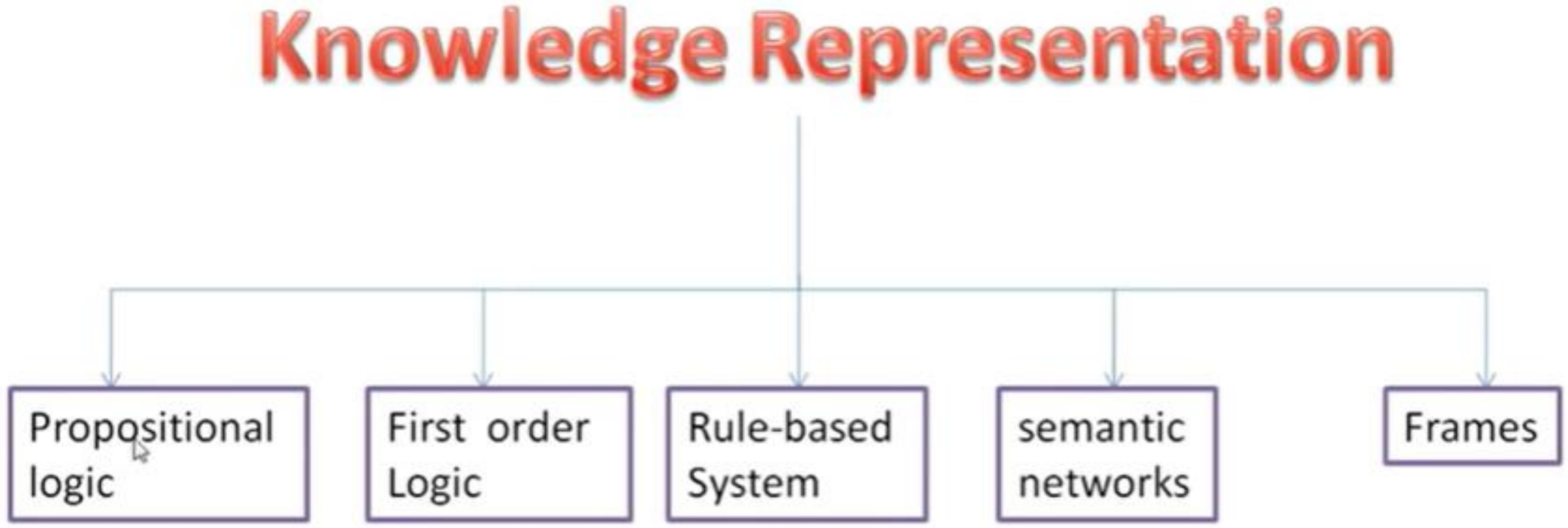
Percept: [Stench, Breeze, Glitter, None, None]

Deduce: Pit in (2,4) or (3,3)
Gold in (2,3)

Action: Move to (1,1) through OK locations

Knowledge Representation Techniques

- How this knowledge can be represented so that a machine can make sense of it. One has to keep in mind that there are numerous ways to achieve this, and no method is perfect and has its own disadvantages.



Knowledge Representation Techniques

1 Logical Representation

- It is the most basic form of representing knowledge to machines where a well-defined syntax with proper rules is used.
- This syntax needs to have no ambiguity in its meaning and must deal with prepositions.
- Thus, this logical form of presentation acts as communication rules and is why it can be best used when representing facts to a machine.
- Logical representation can be categorised into mainly two logics:

1. Propositional Logics
2. Predicate logics

Knowledge Representation Techniques

- Logical Representation can be of two types-

1. Propositional Logic: This type of logical representation is also known as **propositional calculus or statement logic.**

- It 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.
- This works in a Boolean, i.e., True or False method. (Either true or false but not both)

2. 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.

Knowledge Representation Techniques

- Logical Representation
- Advantages:

➢ Logical representation helps to perform logical reasoning

➢ This representation is the basis for the programming languages.

- Disadvantages:

➢ 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.

Logical Representation-Propositional Logic

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.

Logical Representation-Propositional Logic

- **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:
- 2+2 is 4, it is an atomic proposition as it is a **true** fact.
- "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: "It is raining today, and street is wet."

  "Ankit is a doctor, and his clinic is in Mumbai."

Paris is the capital of France and Paris has a population of over two million.

Following are some basic facts about propositional logic:

1. Propositional logic is also called Boolean logic as it works on 0 and 1.
2. 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.
3. Propositions can be either true or false, but it cannot be both.
4. Propositional logic consists of an object, relations or function, and **logical connectives**.
5. These connectives are also called logical operators.

6. The propositions and connectives are the basic elements of the propositional logic.

7. Connectives can be said as a logical operator which connects two sentences.

8. A proposition formula which is always true is called **tautology**, and it is also called a valid sentence.

9. 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.

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:
- **Negation:** A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.
- **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.

Logical Connectives

- **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**.
- **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
- **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.

Connective symbols	Word	Technical term	Example
$\wedge$	AND	Conjunction	$A \wedge B$
$\vee$	OR	Disjunction	$A \vee B$
$\rightarrow$	Implies	Implication	$A \rightarrow B$
$\Leftrightarrow$	If and only if	Biconditional	$A \Leftrightarrow B$
$\neg$ or $\sim$	Not	Negation	$\neg A$ or $\neg B$

- Truth Table

For Implication:

P	Q	$P \rightarrow Q$
True	True	**True**
True	False	**False**
False	True	**True**
False	False	**True**

For Biconditional:

P	Q	$P \Leftrightarrow Q$
True	True	**True**
True	False	**False**
False	True	**False**
False	False	**True**

Logical Connectives

- Truth Table

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 ∨ Q.
True	True	**True**
False	True	**True**
True	False	**True**
False	False	**False**

Propositional Logic

- For propositional logic, a row in the truth table is one interpretation

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Logical Connectives

- Truth table with three propositions:

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

- Logical equivalence:

A	B	¬A	¬AV B	A→B
T	T	F	**T**	**T**
T	F	F	**F**	**F**
F	T	T	**T**	**T**
F	F	T	**T**	**T**

Example

- Translation

Translation of English sentences to propositional logic:

(1) identify atomic sentences that are propositions

(2) Use logical connectives to translate more complex composite sentences that consist of many atomic sentences

Assume the following sentence:

- It is not sunny this afternoon and it is colder than yesterday.

Atomic sentences:

- p = It is sunny this afternoon
- q = it is colder than yesterday

Translation: $\neg p \land q$

Example

- Translation

Assume the following sentences:

- It is not sunny this afternoon and it is colder than yesterday. $\neg p \land q$
- We will go swimming only if it is sunny. $r \rightarrow p$
- If we do not go swimming then we will take a canoe trip. $\neg r \rightarrow s$
- If we take a canoe trip, then we will be home by sunset. $s \rightarrow t$

Denote:

- p = It is sunny this afternoon
- q = it is colder than yesterday
- r = We will go swimming
- s= we will take a canoe trip
- t= We will be home by sunset

Knowledge Representation Techniques

2. Semantic Networks

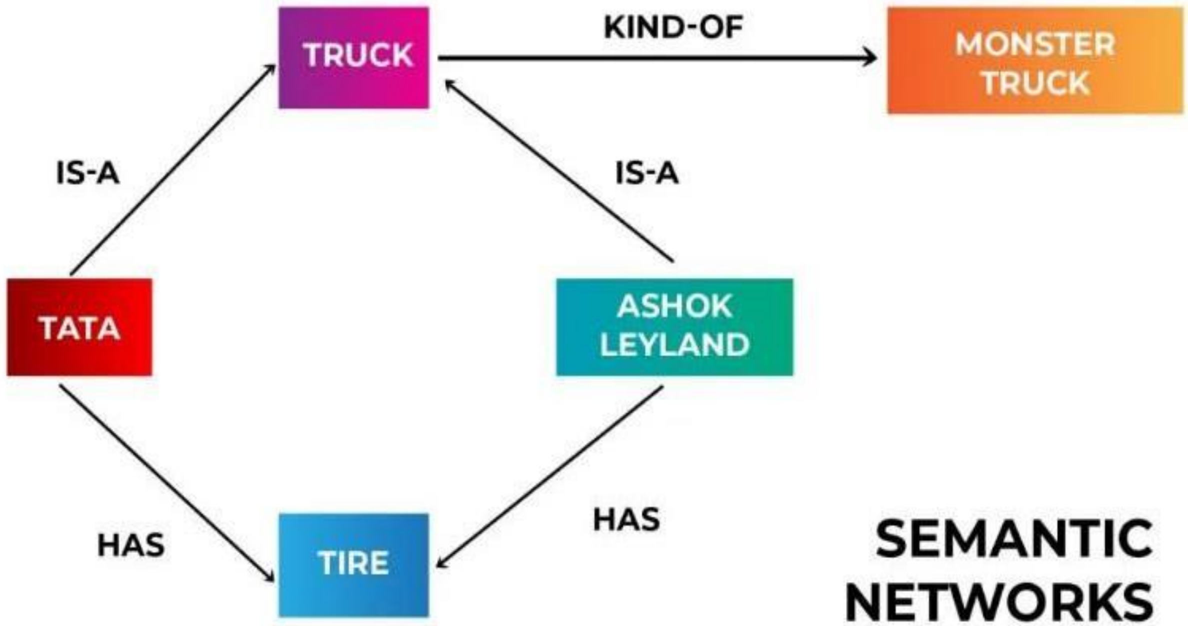
- Graphical notation of representing knowledge in interconnected nodes pattern.
- In this form, a graphical representation conveys how the objects are connected and are often used with a data network.
- Alternative of predicate logic.
- The Semantic networks consist of node/block (the objects) and arcs/edges (the connections) that explain how the objects are connected.
- The relationships found in the Semantic Networks can be of two types – IS-A and instance (KIND-OF).
- This form of representation is more natural than logical. It is simple to understand however suffers from being computationally expensive and do not have the equivalent of quantifiers found in the logical representation.

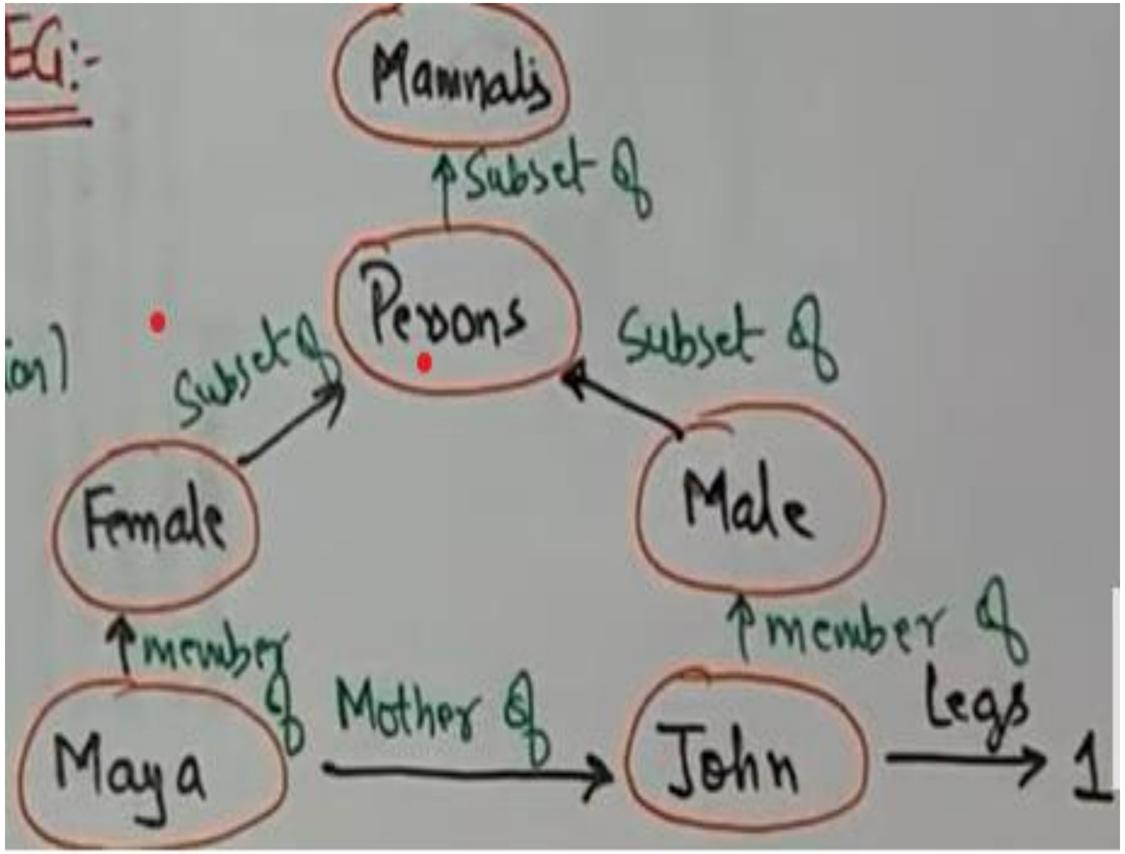
- ➢Nodes represents objects:

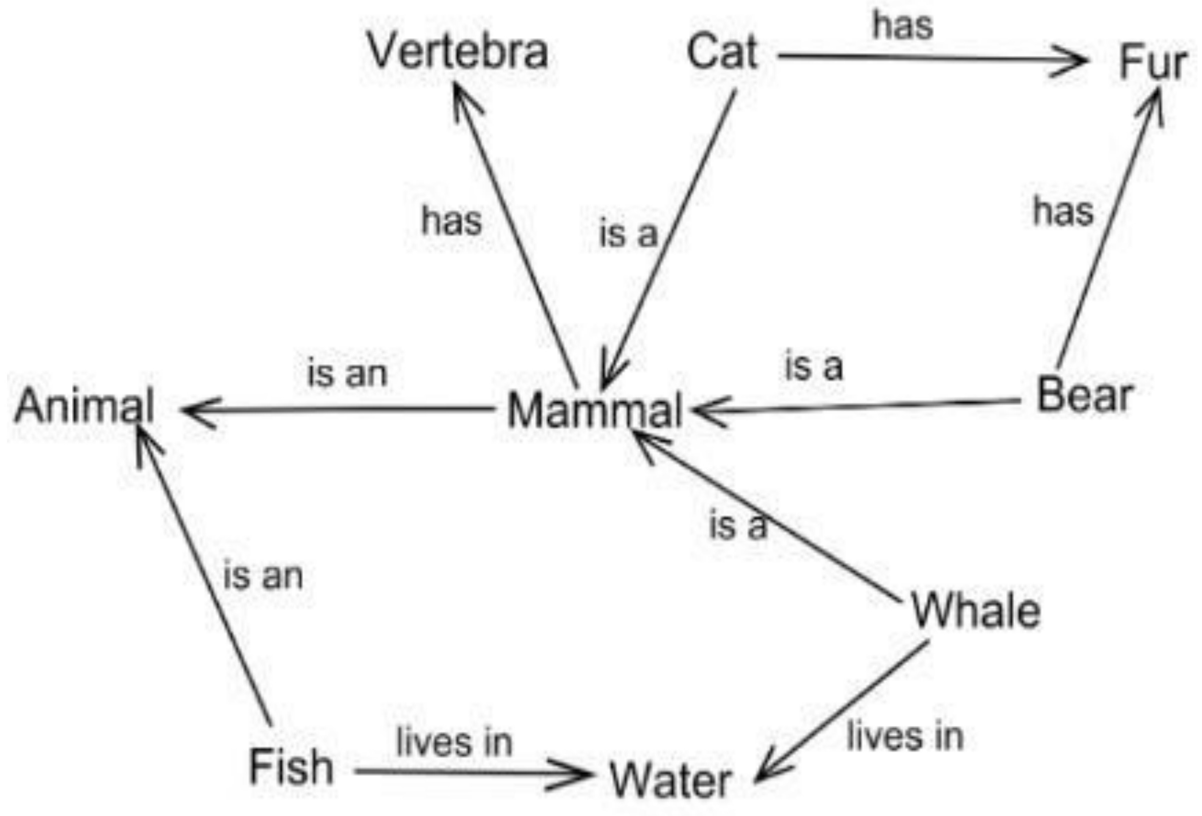
1. Circle
2. Eclipse
3. Rectangle

- ➢Arcs represents relation b/w objects.
- ➢Link labels represents relationships.

Knowledge Representation Techniques





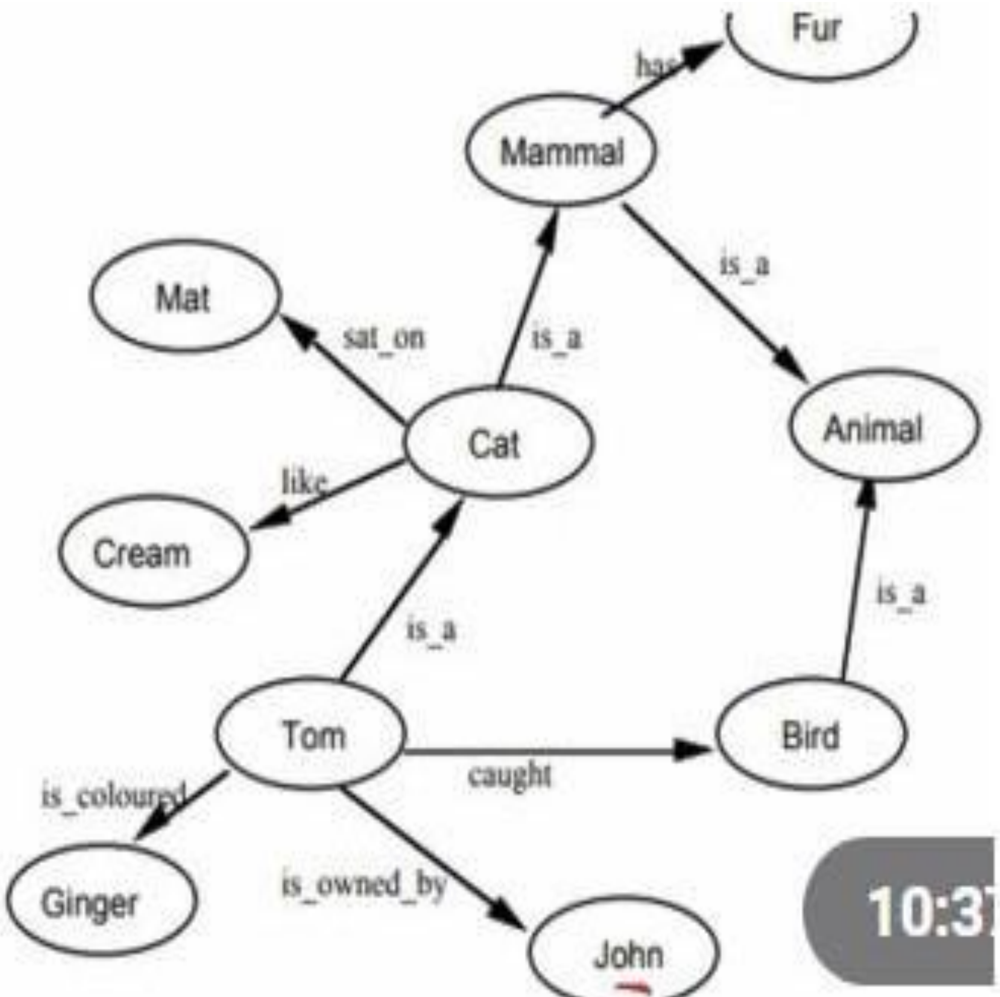

Vertebra
Cat
has
Fur
has
is a
has
is an
is a
Animal
Mammal
Bear
is a
is an
Whale
lives in
lives in
Fish
Water

Knowledge Representation Techniques

- **Semantic Networks**

Example :

Tom is a cat.
Tom caught a bird.
Tom is owned by John.
Tom is ginger in colour.
Cats like cream.
The cat sat on the mat.
A cat is a mammal.
A bird is an animal.
All mammals are animals.
Mammals have fur.



Knowledge Representation Techniques

- Advantages:
  - Semantic networks are a natural representation of knowledge
  - It conveys meaning in a transparent manner
  - These networks are simple and easy to understand

- Disadvantages:
  - Semantic networks takes more computational time at runtime
  - They do not have any equivalent quantifiers.
  - These networks are not intelligent and depends on the creator of the system.

Knowledge Representation Techniques

- **Frame Representation**
- it is a collection of attributes and values linked to it. This AI-specific data structure uses slots and fillers (i.e., slot values, which can be of any data type and shape).
- it has a similar concept to how information is stored in a typical DBMS. These slots and fillers form a structure – a frame. The slots here have the name (attributes), and knowledge related to it is stored in the fillers.
- The biggest advantage of this form of representation is that due to its structure, similar data can be combined in groups as frame representation can divide the knowledge in structures and then further into sub-structures. Also, being like any typical data structure can be understood, visualized, manipulated easily, and typical concepts such as adding, removing, deleting slots can be done effortlessly.

Example 2:

"Tweety is a yellow bird having wings to fly"

```
Tweety(
        (Species        (Value   bird))
        (color          (Value   yellow))
        (Activity       (Value   Fly))
)
```

Conti.

- E.g.

Slots	Fillers
Name	Peter
Profession	Doctor
Age	25
Marital Status	Single
Weight	78

Slots	Fillers
Title	Artificial Intelligence
Genr e	Computer Science
Auth or	Peter Norvig
Editio n	Third Edition
Year	1996
Page	1152

Knowledge Representation Techniques

- Advantages:
  - It makes the programming easier by grouping the related data.
  - Frame representation is easy to understand and visualize.
  - It is very easy to add slots for new attributes and relations also it is easy to include default data and search for missing values.

- Disadvantages:
  - In frame system inference, the mechanism can not be easily processed.
  - The inference mechanism can not be smoothly processed by frame representation.
  - It is very generalized approach.

Knowledge Representation Techniques

- **Production Rules**
- It is among the most common ways in which knowledge is represented in AI systems. In the simplest form, it can be understood as a simple if-else rule-based system and, in a way, is the combination of Propositional and FOPL logics.
- This system comprises a set of production rules, working memory, a recognize act cycle.
- For every input, conditions are checked from the set of a production rule, and upon finding a suitable rule, an action is committed.
- This cycle of selecting the rule based on some conditions and consequently acting to solve the problem is known as a recognition and act cycle, which takes place for every input.

(Condition, action)

4. Production Rule

- 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).

Knowledge Representation Techniques

- Advantages:
  - Expressed in natural language.
  - Production rules are highly modular and can be easily removed or modified.
- Disadvantages:
  - It does not exhibit any leaning capabilities and does not store the result of the problem for future uses.
  - During the execution of program many rules may be active. Thus rule based production systems are inefficient.

Approaches to Knowledge Representation

- There are different approaches to knowledge representation such as:

1. Simple Relational Knowledge
2. Inheritable Knowledge
3. Inferential Knowledge

1. Simple Relational Knowledge

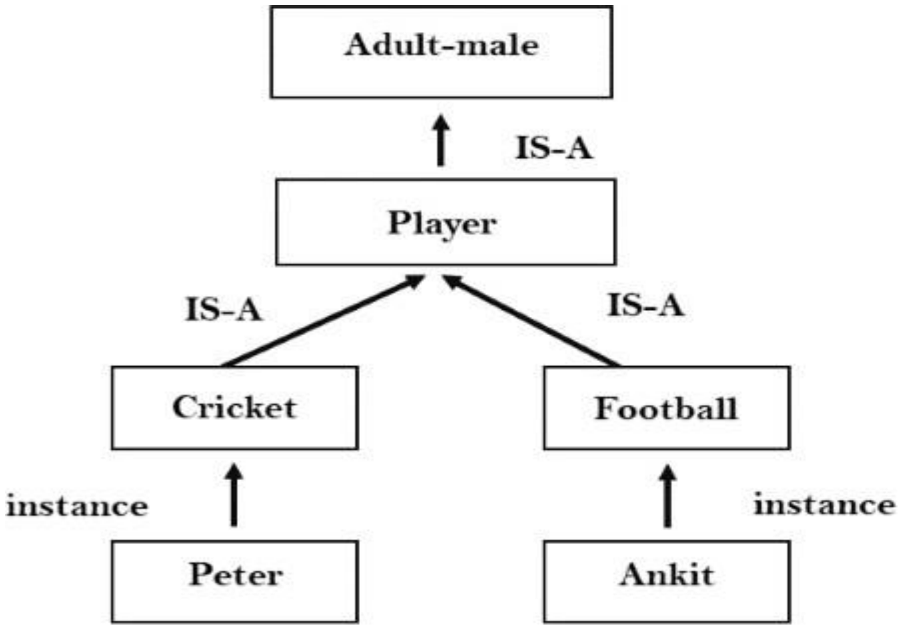
It is the simplest way of storing facts which uses the relational method. Here, all the facts about a set of the object are set out systematically in columns. Also, this approach of knowledge representation is famous in database systems where the relationship between different entities is represented. Thus, there is little opportunity for inference.

Name	Age	Emp ID
John	25	100071
Amanda	23	100056
Sam	27	100042

Approaches to Knowledge Representation

2. Inheritable Knowledge

- In the inheritable knowledge approach, all data must be stored into a hierarchy of classes and should be arranged in a generalized form or a hierarchal manner. Also, this approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation. In this approach, objects and values are represented in Boxed nodes.



- This structure is a slot-and-filler structure. Also known as Semantic network.

Approaches to Knowledge Representation

3. Inferential Knowledge

- The inferential knowledge approach represents knowledge in the form of formal logic. Thus, it can be used to derive more facts. Also, it guarantees correctness.
- Example: Statement 1: John is a cricketer.
  - Statement 2: All cricketers are athletes.
- Then it can be represented as; Cricketer(John)
  - $\forall$x= Cricketer (x) ———-> Athelete (x)s

Formalizing English Sentences

- Let’s consider a propositional language where
- p = ”Paola is happy”
- q = ”Paola paints a picture”
- r = ”Renzo is happy”.
- Formalize the following sentences:

➢ ”if Paola is happy and paints a picture then Renzo isn’t happy”

- $p \land q \rightarrow \neg r$

➢ ”if Paola is happy, then she paints a picture”

- $p \rightarrow q$

➢ ”Paola is happy only if she paints a picture”

- $p \Leftrightarrow q$

Formalizing English Sentences

- Let A =”Angelo comes to the party”,
- B=”Bruno comes to the party”,
- C=”Carlo comes to the party”
- D =”Davide comes to the party”.

1 ”If Davide comes to the party then Bruno and Carlo come too”

2 ”Carlo comes to the party only if Angelo and Bruno do not come”

3 ”If Davide comes to the party, then, if Carlo doesn’t come then Angelo comes”

4 ”Carlo comes to the party provided that Davide doesn’t come, but, if Davide comes, then Bruno doesn’t come”

5 ”A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren’t coming, Davide comes”

6 ”Angelo, Bruno and Carlo come to the party if and only if Davide doesn’t come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes”

Formalizing English Sentences

- Let A ="Angelo comes to the party",
- B= "Bruno comes to the party",
- C= "Carlo comes to the party",
- D = "Davide comes to the party".

1 "If Davide comes to the party then Bruno and Carlo come too"

- $D \rightarrow B \wedge C$

2 "Carlo comes to the party if Angelo and Bruno do not come"

- $C \rightarrow \neg A \wedge \neg B$

3 "If Davide comes to the party, then, if Carlo doesn't come then Angelo comes"

- $D \rightarrow (\neg C \rightarrow A)$

Formalizing English Sentences

- Let A =”Angelo comes to the party”, B =”Bruno comes to the party”, C =”Carlo comes to the party”, and D =”Davide comes to the party”.
- ”Carlo comes to the party provided that Davide doesn’t come, but, if Davide comes, then Bruno doesn’t come”

➢ $(C \rightarrow \neg D) \wedge (D \rightarrow \neg B)$

- ”A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren’t coming, Davide comes”

➢ $A \rightarrow (\neg B \wedge \neg C \rightarrow D)$

- ”Angelo, Bruno and Carlo come to the party if and only if Davide doesn’t come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes”

➢ $(A \wedge B \wedge C \leftrightarrow \neg D) \wedge (\neg A \wedge \neg B \rightarrow (D \rightarrow C))$

Contradiction and Tautology

- Some composite sentences may always (under any interpretation) evaluate to a single truth value:

- **Contradiction** (always ***False***)

$$P \wedge \neg P$$

- **Tautology** (always ***True***)

$$P \vee \neg P$$

$$\left.\begin{array}{l} \neg(P \vee Q) \Leftrightarrow (\neg P \wedge \neg Q) \\ \neg(P \wedge Q) \Leftrightarrow (\neg P \vee \neg Q) \end{array}\right\} \text{DeMorgan's Laws}$$

Rules of Inference

- Inference: In artificial intelligence, we need intelligent computers which can create new logic from old logic or by evidence, **so generating the conclusions from evidence and facts is termed as Inference**.
- Inference rules are the templates for generating valid arguments. Inference rules are applied to derive proofs in AI, and the proof is a sequence of the conclusion that leads to the desired goal.
- Following are some terminologies related to inference rules:

1. **Implication:** It is one of the logical connectives which can be represented as $P \rightarrow Q$. It is a Boolean expression.
2. **Converse:** The converse of implication, which means the right-hand side proposition goes to the left-hand side and vice-versa. It can be written as $Q \rightarrow P$.

Rules of Inference

3. Contrapositive: The negation of converse is termed as contrapositive, and it can be represented as $\neg Q \rightarrow \neg P$.

4. Inverse: The negation of implication is called inverse. It can be represented as $\neg P \rightarrow \neg Q$.

- From the above term some of the compound statements are equivalent to each other, which we can prove using truth table:

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$\neg Q \rightarrow \neg P$	$\neg P \rightarrow \neg Q$.
T	T	T	T	T	T
T	F	F	T	F	T
F	T	T	F	T	F
F	F	T	T	T	T

Types of Inference Rules

1. Modus Ponens:

- if P→ Q is true and P is true then we can say that Q will be true.
- Example :
- Statement-1: "If it is holiday then school is closed" ==> P→ Q
  Statement-2: " it is holiday " ==> P
  Conclusion: "school is closed " ==> Q.
  Hence, we can say that, **Proof by Truth table:**

Notation for Modus ponens: $\dfrac{P \to Q, \quad P}{\therefore Q}$

P	Q	$P \to Q$
0	0	0
0	1	1
1	0	0
1	1	1 ←

Types of Inference Rules

2. Modus Tollens: The Modus Tollens rule state that if P→ Q is true and **¬ Q is true, then ¬ P** will also true.

Notation for Modus Tollens: $\dfrac{P \to Q,\ \sim Q}{\sim P}$

- **Statement-1:** "If I am sleepy then I go to bed" ==> P→ Q
  Statement-2: "I do not go to the bed."==> ~Q
  Statement-3: Which infers that "**I am not sleepy**" => ~P
- **Proof by Truth table:**

P	Q	$\sim P$	$\sim Q$	$P \to Q$
0	0	1	1	1 ←
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1

Types of Inference Rules

3. Hypothetical Syllogism:

- It states that if P→R is true whenever P→Q is true, and Q→R is true.
- Statement-1: If you have my home key then you can unlock my home.
  P→Q
  Statement-2: If you can unlock my home then you can take my money.
  Q→R
  Conclusion: If you have my home key then you can take my money. P→R
- **Proof by Truth table:**

P	Q	R	$P \to Q$	$Q \to R$	$P \to R$
0	0	0	1	1	1 ←
0	0	1	1	1	1 ←
0	1	0	1	0	1
0	1	1	1	1	1 ←
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1 ←

Types of Inference Rules

4. Disjunctive Syllogism:

- It states that if PVQ is true, and ¬P is true, then Q will be true.
- Example
- **Statement-1:** Today is Sunday or Monday. ==>PVQ
  Statement-2: Today is not Sunday. ==> ¬P
- **Conclusion:** Today is Monday. ==> Q
- **Proof by Truth table:**

Notation of Disjunctive syllogism: $\dfrac{P \vee Q,\ \neg P}{Q}$

P	Q	$\neg P$	$P \vee Q$
0	0	1	0
0	1	1	1
1	0	0	1
1	1	0	1

Types of Inference Rules

5. Addition: The Addition rule is one the common inference rule, and it states that If P is true, then PVQ will be true.

Notation of Addition: $\frac{P}{P \vee Q}$

- **Example:**
- **Statement:** I have a vanilla ice-cream. ==> P
  Statement-2: I have Chocolate ice-cream.
  Conclusion: I have vanilla or chocolate ice-cream. ==> (PVQ)
- **Proof by Truth table:**

P	Q	$P \vee Q$
0	0	0
1	0	1
0	1	1
1	1	1

Types of Inference Rules

6. Simplification: The simplification rule state that if **P∧ Q** is true, then **Q or P** will also be true. It can be represented as:

Notation of Simplification rule: $\frac{P \land Q}{Q}$ Or $\frac{P \land Q}{P}$

7. Resolution: The Resolution rule state that if P∨Q and ¬ P∧R is true, then Q∨R will also be true. **It can be represented as**

Notation of Resolution $\frac{P \lor Q,\ \neg P \land R}{Q \lor R}$

P	¬P	Q	R	$P \lor Q$	¬P∧R	$Q \lor R$
0	1	0	0	0	0	0
0	1	0	1	0	0	1
0	1	1	0	1	1	1 ←
0	1	1	1	1	1	1 ←
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1 ←

Logical Representation-Propositional Logic

- In order to draw conclusions, facts are represented in a more convenient way as,

1. Marcus is a man.

- ➢ man(Marcus)

2. Plato is a man.

- ➢ man(Plato)

3. All men are mortal.

- ➢ mortal(men)

- Caesar was a ruler

- ➢ ruler(Caesar)

Limitations of Propositional Logic

- Propositional logic cannot express general-purpose knowledge briefly.
- We need 32 sentences to describe the relationship between wumpus and stenches.
- We would need another 32 sentences for pits and breezes
- We would need at least 64 sentences to describe the effects of actions
- Generalizations, patterns, regularities difficult to represent (all triangles have 3 sides)
- Can’t directly talk about properties of individuals or relations between individuals (e.g., “Bill is tall”)

Propositional Logic VS Predicate Logic

Propositional Logic	Predicate Logic
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.
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.
A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables' value.
Scope analysis is not done in propositional logic.	Predicate logic helps analyze the scope of the subject over the predicate. There are two quantifiers : Universal Quantifier (∀) depicts for all, Existential Quantifier (∃) depicting there exists some
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.