UNIT-II

PART A (2 Marks)

1. What is Knowledge management?

The knowledge management (KM) process comprises a set of activities for identification, gathering, creation, presentation and distribution of knowledge for the purposes of learning, reuse, and awareness.

2. What are the 5 types of knowledge?

- Declarative Knowledge
- Procedural Knowledge
- Meta-knowledge
- Heuristic knowledge
- Structural knowledge

3. What is 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.

4. Give an example for Declarative Knowledge

An example of a declarative knowledge statement is, "A car has four tires." Declarative knowledge is explicit, meaning either a person knows it or they don't.

5. What is 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.

6. What is procedural knowledge example?

It's basically “how” you know to do something. The classic example of procedural knowledge is riding a bicycle. When someone was teaching you how to ride a bicycle, no matter what they said, you probably struggled to grasp it until you'd actually done it a few times.

7. What is Meta Knowledge?

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

8. What are the examples of Meta Knowledge?

Examples of the first-level individual meta-knowledge are methods of planning, modelling, tagging, learning and every modification of domain knowledge. Indeed, universal meta-knowledge frameworks have to be valid for the organization of meta-levels of individual meta-knowledge.

9. What is 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.

10. Give example for heuristic knowledge?

Some examples of heuristic knowledge are a hypothesis, common sense, rule of thumb, and intuition. Heuristic knowledge helps a person make judgments in a sufficient manner and amount of time. A concrete example of heuristic knowledge would be when a plumber comes to give an estimate to a new customer.

11. What is structural knowledge?

Structural knowledge is knowledge of how the ideas within a domain are integrated and interrelated”

12. Give example for 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.
- Examples include temporal relations, spatial relations, family relations, social relations, administrative organizations, military hierarchies, etc.

13. What is knowledge representation?

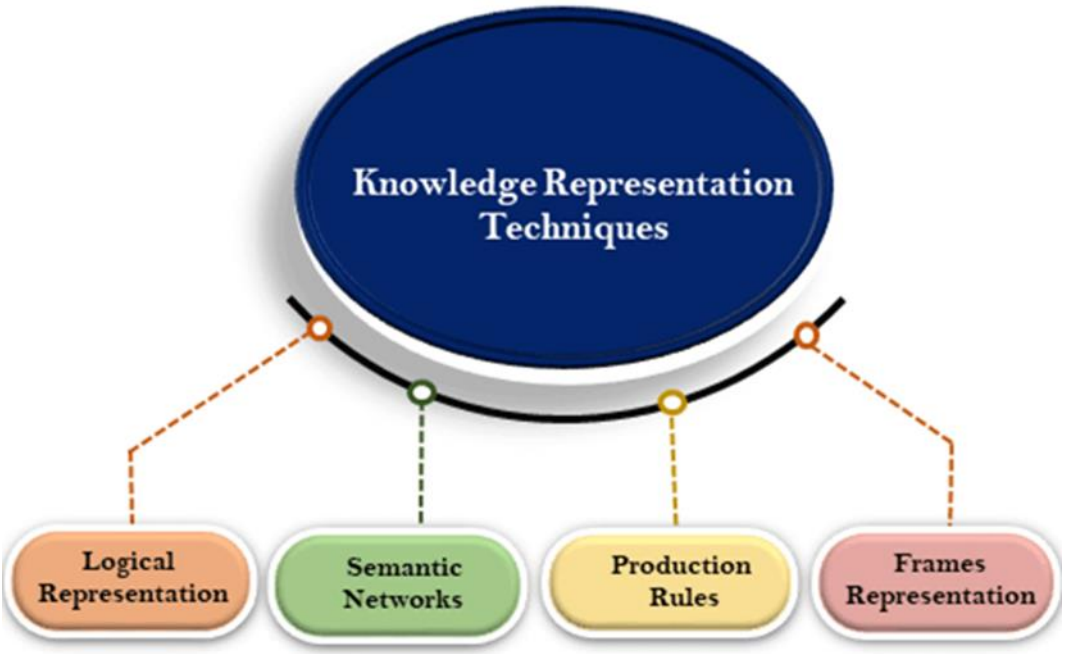
- Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.

14. What are the two main components in knowledge representation?

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.

15. What are the ways of knowledge representation?

- Logical Representation
- Semantic Network Representation
- Frame Representation
- Production Rules. Techniques of knowledge representation



16. What is 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.

17. What is the Syntax for Knowledge representation?

- 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.

18. What is the Semantics for Knowledge representation?

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

19. How logical Representation is categorised?

Logical representation can be categorised into mainly two logics:

a. Propositional Logics
b. Predicate logics

20. What are the advantages of logical representation?

Advantages of logical representation:

- Logical representation enables us to do logical reasoning.
- Logical representation is the basis for the programming languages.

21. What are the disadvantages of logical representation?

Disadvantages of logical Representation:

- Logical representations have some restrictions and are challenging to work with.
- Logical representation technique may not be very natural, and inference may not be so efficient.

22. What is Semantic Network Representation?

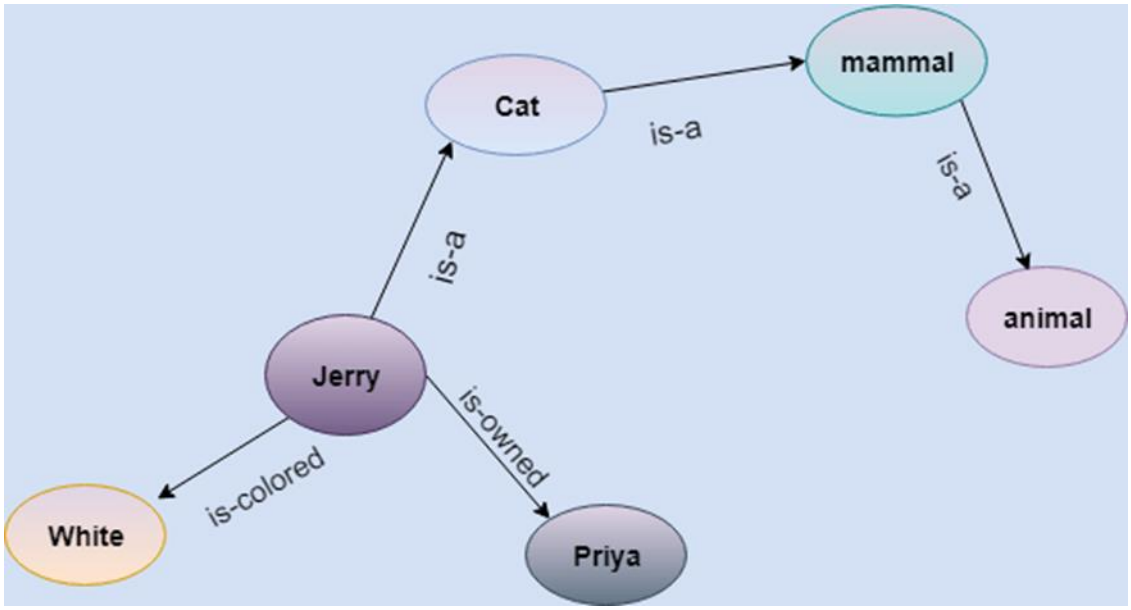
- Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks.
- This network consists of nodes representing objects and arcs which describe the relationship between those objects. Semantic networks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

23. Give the examples for Semantic Network Representation?

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

Statements:

a) Jerry is a cat.
b) Jerry is a mammal
c) Jerry is owned by Priya.
d) Jerry is brown colored.
e) 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.

24. What are the advantages of Semantic Network?

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.

25. What are the disadvantages of Semantic Network?

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 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.

26. Define a Frame?

- 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.

27. Define a Facet?

- 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.

28. What are the advantages and disadvantages of Frame System?

Advantages of frame representation:

1) The frame knowledge representation makes the programming easier by grouping the related data.
2) The frame representation is comparably flexible and used by many applications in AI.
3) It is very easy to add slots for new attribute and relations.
4) It is easy to include default data and to search for missing values.
5) Frame representation is easy to understand and visualize.

Disadvantages of frame representation:

1) In frame system inference mechanism is not be easily processed.
2) Inference mechanism cannot be smoothly preceded by frame representation.
3) Frame representation has a much generalized approach.

29. What is a Production Rule System?

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

30. What are the Advantages and Disadvantages of Production Rules?

Advantages of Production rule:

1) The production rules are expressed in natural language.
2) The production rules are highly modular, so we can easily remove, add or modify an individual rule.

Disadvantages of Production rule:

1) Production rule system does not exhibit any learning capabilities, as it does not store the result of the problem for the future uses.
2) During the execution of the program, many rules may be active hence rule-based production systems are inefficient.

31. What is 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 behaviour in AI agents. An agent is only able to accurately act on some input when he has some knowledge or experience about that input.

32. What are the four main approaches to knowledge representation?

- Simple relational knowledge
- Inheritable knowledge
- Inferential knowledge
- Procedural knowledge

33. Give an example for inferential knowledge?

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)

∀x = man (x) ----------> mortal (x)s

34. What are the Requirements for Knowledge Representation?

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.

34. What is Knowledge based agent?

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.

35. What is an inference system?

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

36. What is the 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

37. What are the various levels of knowledge based agent?

1. Knowledge level

2. Logical level

3. Implementation level

38. What is first order logic?

First-order logic is also known as Predicate logic or First-order predicate logic. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects

First-order logic also has two main parts:

- Syntax
- Semantics

39. What are the basic elements of FOL?

1. Constant 1, 2, A, John, Mumbai, cat,....
2. Variables x, y, z, a, b,....
3. Predicates Brother, Father, >,....
4. Function sqrt, LeftLegOf, ...
5. Connectives ∧, ∨, ¬, ⇒, ⇔
6. Equality ==
7. Quantifier ∀, ∃

40. Define atomic sentence?

Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.

Example: Ravi and Ajay are brothers:

=>Brothers(Ravi,Ajay).

Chinky is a cat: => cat (Chinky).

41. Define complex sentence?

Complex sentences are made by combining atomic sentences using connectives.

42. Define Subject and Predicate with example?

First-order logic statements can be divided into two parts:

Subject: Subject is the main part of the statement.

Predicate: A predicate can be defined as a relation, which binds two atoms together in a statement.

Consider the statement: "x is an integer.", it consists of two parts, the first part x is the subject of the statement and second part "is an integer," is known as a predicate.

43. What do you mean by quantifiers in First Order Logic?

A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.

These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:

1. Universal Quantifier, (for all, everyone, everything)
2. Existential quantifier, (for some, at least one).

44. What is an universal quantifier?

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

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

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

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

45. Define Existential Quantifier.

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

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

If x is a variable, then existential quantifier will be ∃x or ∃(x

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

46. Give the representation for following sentences using first order logic and quantifiers?

1.All birds fly.
In this question the predicate is "fly(bird)."
And since there are all birds who fly so it will be represented as follows.

∀x bird(x) →fly(x).

2.Every man respects his parent.
In this question, the predicate is "respect(x, y)," where x=man, and y= parent.
Since there is every man so will use ∀, and it will be represented as follows:

∀x man(x) → respects (x, parent).

3.Some boys play cricket.
In this question, the predicate is "play(x, y)," where x= boys, and y= game.
Since there are some boys so we will use ∃, and it will be represented as:

∃x boys(x) → play(x, cricket).

4.Not all students like both Mathematics and Science.
In this question, the predicate is "like(x, y)," where x= student, and y= subject.
Since there are not all students, so we will use ∀ with negation, so following representation for this:

¬∀ (x) [student(x) → like(x, Mathematics) ∧ like(x, Science)].

5.Only one student failed in Mathematics.
In this question, the predicate is "failed(x, y)," where x= student, and y= subject.
Since there is only one student who failed in Mathematics, so we will use following representation for this:

∃(x) [student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].

47. What is conceptual dependency explain with example?

Conceptual dependency provides a structure in which knowledge can be represented and also a set of building blocks from which representations can be built. A typical set of primitive actions are. ATRANS - Transfer of an abstract relationship (Eg: give) PTRANS - Transfer of the physical location of an object (Eg: go)

PART B (5 Marks)

1. **Explain the different types of knowledge with example?**

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.
- An example of a declarative knowledge statement is, "A car has four tires." Declarative knowledge is explicit, meaning either a person knows it or they don't.

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.
- The classic example of procedural knowledge is riding a bicycle. When someone was teaching you how to ride a bicycle, no matter what they said, you probably struggled to grasp it until you'd actually done it a few times.

Meta-knowledge:

- Knowledge about the other types of knowledge is called Meta-knowledge.
- 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.
- Examples of the first-level individual meta-knowledge are methods of planning, modeling, tagging, learning and every modification of domain knowledge. Indeed, universal meta-knowledge frameworks have to be valid for the organization of meta-levels of individual meta-knowledge.

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.
- Some examples of heuristic knowledge are a hypothesis, common sense, rule of thumb, and intuition. Heuristic knowledge helps a person make judgments in a sufficient manner and amount of time. A concrete example of heuristic knowledge would be when a plumber comes to give an estimate to a new customer.

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.
- Examples include temporal relations, spatial relations, family relations, social relations, administrative organizations, military hierarchies, etc.

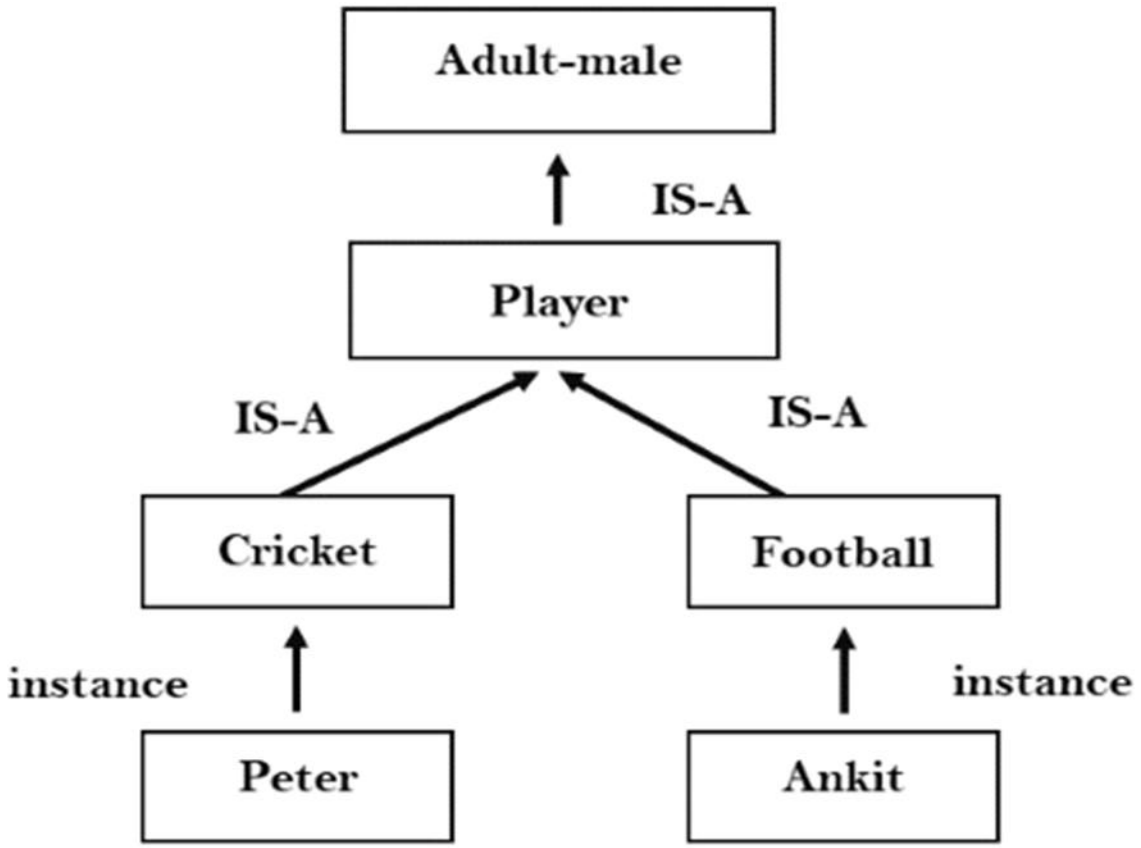
2. What are the different approaches to knowledge Representation?

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

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.

Inheritable knowledge:



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:

Marcus is a man

Allmen are mortal
Then it can represent as;

man(Marcus)
∀x = man (x) ----------> mortal (x)s

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 Prolog language.

3. **Explain Frame with example?**

   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

Example

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig

Edition	Third Edition
Year	1996
Page	1152

4. Explain Production Rules with examples?

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

Examples

- 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).

5. Explain the two types of quantifiers in FOL(First Order Logic) with example?

Quantifiers in First-order logic

1. A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.
2. These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.
3. There are two types of quantifier:

- Universal Quantifier, (for all, everyone, everything)

- Existential quantifier, (for some, at least one).

Universal Quantifier

1. Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.
2. The Universal quantifier is represented by a symbol ∀, which resembles an inverted A.
3. If x is a variable, then ∀x is read as:

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

Examples

1.All birds fly.
In this question the predicate is "fly(bird)."
And since there are all birds who fly so it will be represented as follows.

$\forall x \text{ bird}(x) \rightarrow \text{fly}(x)$.

2.Every man respects his parent.
In this question, the predicate is "respect(x, y)," where x=man, and y= parent.
Since there is every man so will use ∀, and it will be represented as follows:

$\forall x \text{ man}(x) \rightarrow \text{respects}(x, \text{parent})$.

Existential Quantifier

1. Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something.
2. It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as an existential quantifier.
3. If x is a variable, then existential quantifier will be ∃x or ∃(x)

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

Example

1. Some boys play cricket.

The predicate is "play(x, y)," where x= boys, and y= game. Since there are some boys so we will use ∃, and it will be represented as:

∃x boys(x) → play(x, cricket).

2. Only one student failed in Mathematics.

The predicate is "failed(x, y)," where x= student, and y= subject.
Since there is only one student who failed in Mathematics, so we will use following representation for this:

∃(x) [student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].

5. Explain Knowledge Representation in detail?

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).

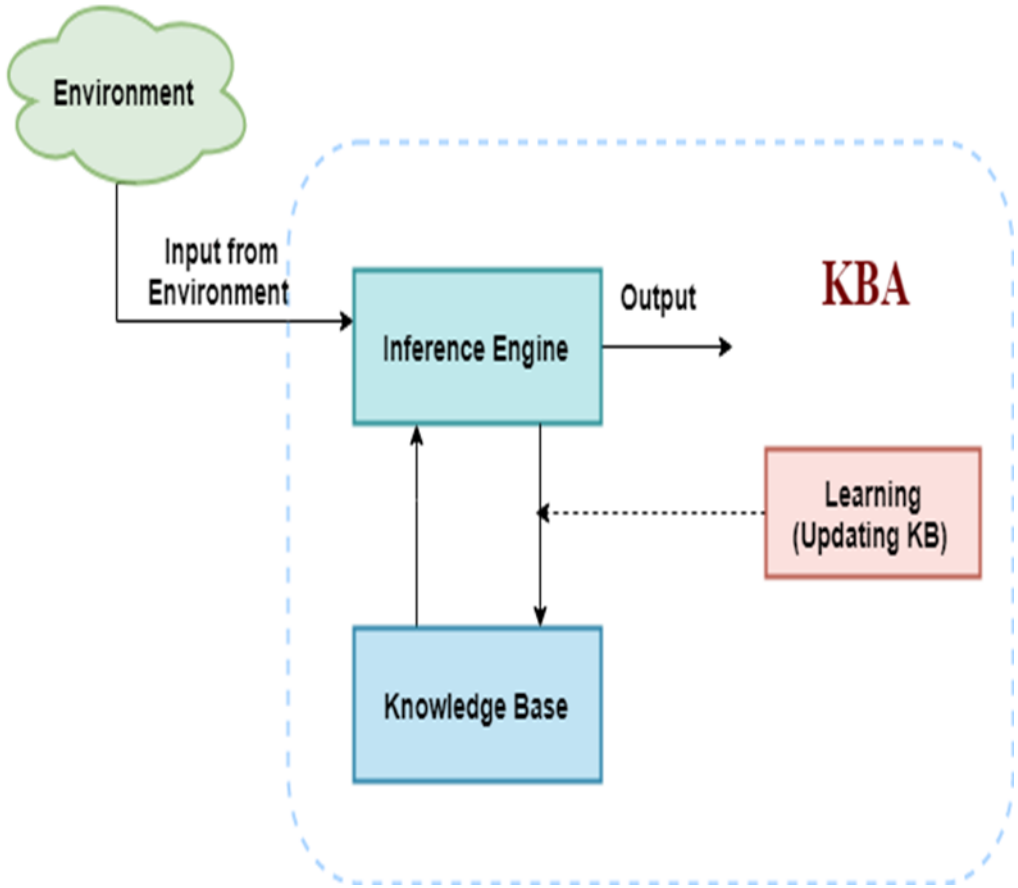
Knowledge: Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. Following are the types of knowledge in artificial intelligence:

6. Explain about 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 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:
    a. Knowledge-base and
    b. 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 deduce the internal representation of the world
- An agent can deduce appropriate actions

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:

    a. Forward chaining
    b. Backward chaining

Operations Performed by KBA

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

a. TELL: This operation tells the knowledge base what it perceives from the environment.
b. ASK: This operation asks the knowledge base what action it should perform.
c. Perform: It performs the selected action

Various levels of knowledge-based agent

1. Knowledge level

2. Logical level

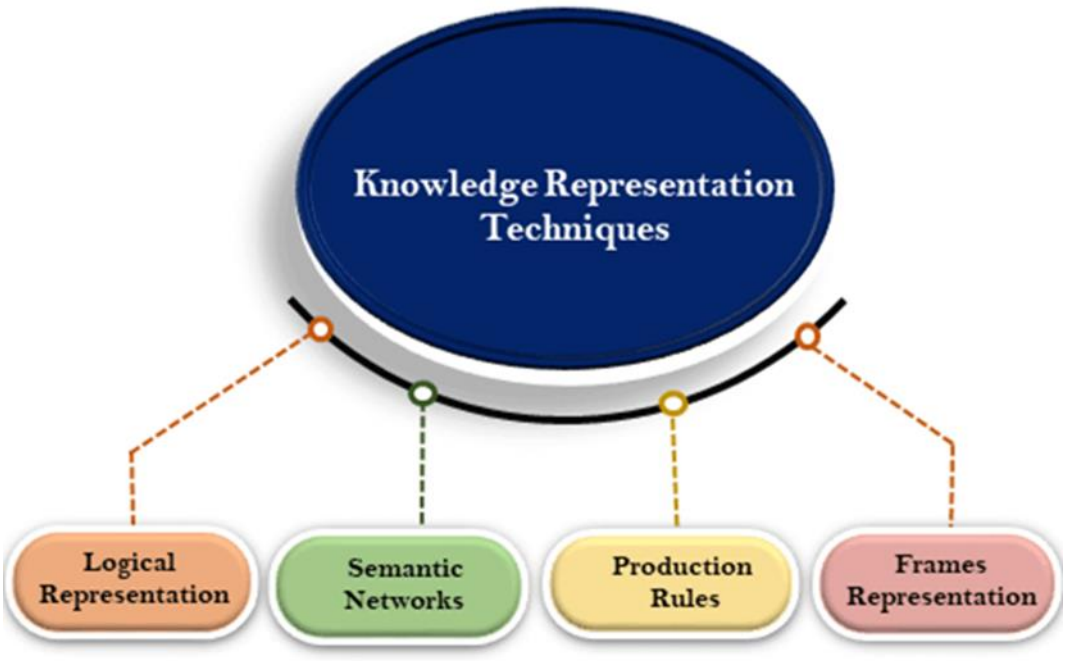
3. Implementation level

10 Marks

1. Explain the Techniques of knowledge representation?

There are mainly four ways of knowledge representation which are given as follows:

- Logical Representation
- Semantic Network Representation
- Frame Representation
- Production Rules



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 categorised into mainly two logics:

- Propositional Logics
- Predicate logics

Advantages of logical representation:

1. Logical representation enables us to do logical reasoning.
2. Logical representation is the basis for the programming languages.

Disadvantages of logical Representation:

1. Logical representations have some restrictions and are challenging to work with.
2. Logical representation technique may not be very natural, and inference may not be so efficient.

3. 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.

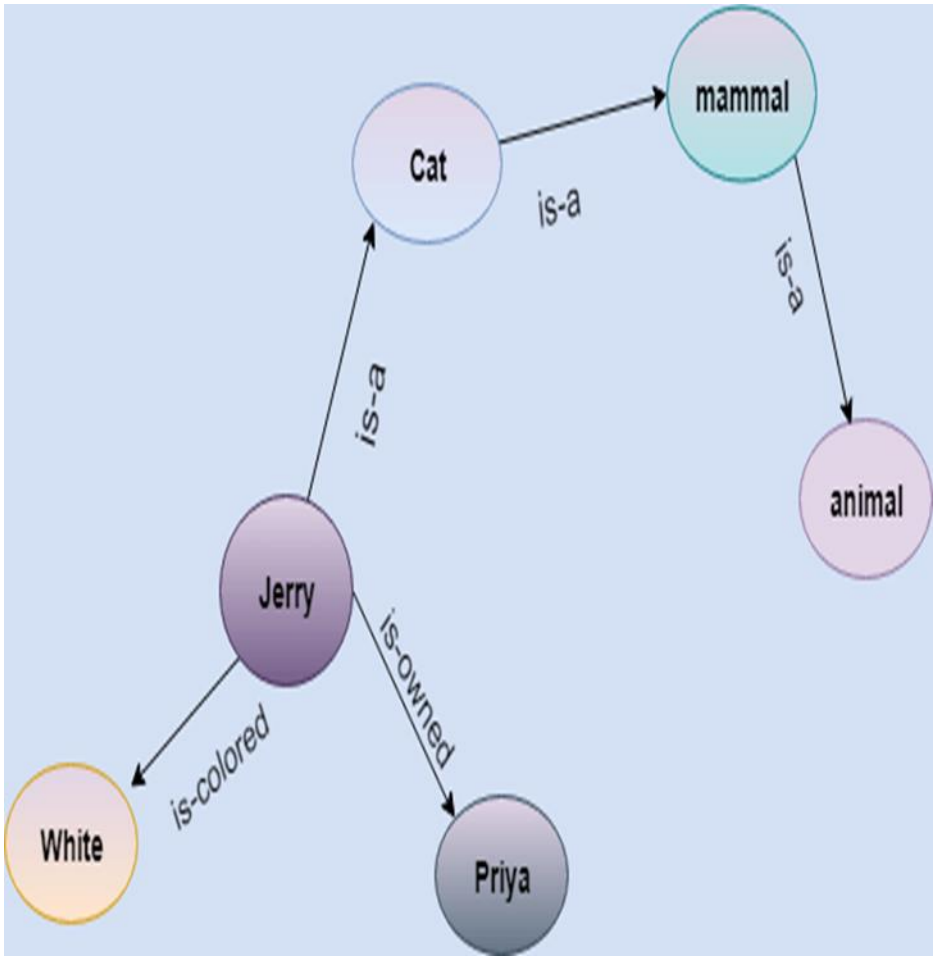
This representation consists of mainly two types of relations:

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:

a. Jerry is a cat.
b. Jerry is a mammal
c. Jerry is owned by Priya.
d. Jerry is brown colored.
e. 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 1015 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.

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition

Year	1996
Page	1152

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.

Advantages of frame representation:

- The frame knowledge representation makes the programming easier by grouping the related data.
- The frame representation is comparably flexible and used by many applications in AI.
- It is very easy to add slots for new attribute and relations.
- It is easy to include default data and to search for missing values.
- Frame representation is easy to understand and visualize.

Disadvantages of frame representation:

- In frame system inference mechanism is not be easily processed.
- Inference mechanism cannot be smoothly proceeded by frame representation.
- Frame representation has a much generalized approach.

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).

Advantages of Production rule:

a. The production rules are expressed in natural language.
b. The production rules are highly modular, so we can easily remove, add or modify an individual rule.

Disadvantages of Production rule:

a. Production rule system does not exhibit any learning capabilities, as it does not store the result of the problem for the future uses.
b. During the execution of the program, many rules may be active hence rule-based production systems are inefficient.

2. Explain the issues in Knowledge Representation?

Issues that should be raised when using a knowledge representation technique:

- Are any attributes of objects so basic that they occur in almost every problem domain?
- Are there any important relationships that exist among attributes of objects?
- At what level should knowledge be represented? Is there a good set of primitives into which all knowledge can be broken down?
- How should sets of objects be represented?
- Given a large amount of knowledge stored in a database, how can relevant parts be accessed when they are needed?

Important Attributes:

- Any attribute of objects so basic that they occur in almost every problem domain ?

a. Relationship among attributes:

- Any important relationship that exists among object attributes ?

a. Choosing Granularity :

- At what level of detail should the knowledge be represented ?

a. Set of objects :

- How sets of objects be represented?

a. Finding Right structure :

1. Important Attributes:

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

2. 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:

3. Properties of Attributes

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, For example, the attribute height is a specialization of general attribute physical-size which is, in turn, a specialization of physical-attribute. 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.

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.

4. Choosing Granularity

- At what level of detail should the knowledge be represented?
- Regardless of the KR formalism, it is necessary to know:
- At what level should the knowledge be represented and what are the primitives?
- Should there be a small number or should there be a large number of low-level primitives or High-level facts.
- High-level facts may not be adequate for inference while Low-level primitives may require a lot of storage.

Example of Granularity:

- John spotted Sue.
- This could be represented as
- Spotted (agent(John),object (Sue))
- Such a representation would make it easy to answer questions such are:
- Who spotted Sue?
- Suppose we want to know:
- Did John see Sue?

- Given only one fact, we cannot discover that answer.
- We can add other facts, such as
- Spotted(x, y) -> saw(x, y)

5. Set of objects

- How should sets of objects be represented?
- There are certain properties of objects that are true as member of a set but not as individual;
- Example: Consider the assertion made in the sentences:
- "there are more sheep than people in Australia", and
- "English speakers can be found all over the world."
- To describe these facts, the only way is to attach assertion to the sets representing people, sheep, and English.
- The reason to represent sets of objects is: if a property is true for all or most elements of a set, then it is more efficient to associate it once with the set rather than to associate it explicitly with every elements of the set.
- This is done,in logical representation through the use of universal quantifier, and
- in hierarchical structure where node represent sets and inheritance propagate set level assertion down to individual.

6. Finding Right structure

- Given a large amount of knowledge stored in a database, how can relevant parts are accessed when they are needed?
- This is about access to right structure for describing a particular situation.
- This requires, selecting an initial structure and then revising the choice

While doing so, it is necessary to solve following problems

- How to perform an initial selection of the most appropriate structure.
- How to fill in appropriate details from the current situations.
- How to find a better structure if the one chosen initially turns out not to be appropriate.
- What to do if none of the available structures is appropriate.
- When to create and remember a new structure.

3. Explain in Detail about First Order Logic?

First-Order Logic

- In Propositional logic, we represent statements using propositional logic. But unfortunately, in propositional logic, we can only represent the facts, which are either true or false.
- PL is not sufficient to represent the complex sentences or natural language statements. The propositional logic has very limited expressive power. Consider the following sentence, which we cannot represent using PL logic.
- "Some humans are intelligent", or
  "Sachin likes cricket."
- To represent the above statements, PL logic is not sufficient, so we required some more powerful logic, such as first-order logic.
  - First-order logic is another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
  - FOL is sufficiently expressive to represent the natural language statements in a concise way.
  - First-order logic is also known as Predicate logic or First-order predicate logic. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.
  - First-order logic (like natural language) does not only assume that the world contains facts like propositional logic but also assumes the following things in the world:

a. Objects: A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
b. Relations: It can be unary relation such as: red, round, is adjacent, or n-any relation such as: the sister of, brother of, has color, comes between
c. Function: Father of, best friend, third inning of, end of,

As a natural language, first-order logic also has two main parts:

a. Syntax
b. Semantics

Syntax of First-Order logic:

The syntax of FOL determines which collection of symbols is a logical expression in first-order logic. The basic elements of first-order logic are symbols. We write statements in short-hand notation in FOL.

Basic Elements of First-order logic

Constant 1, 2, A, John, Mumbai, cat,....

- Variables x, y, z, a, b,....
- Predicates Brother, Father, >,....
- Function sqrt, LeftLegOf, ...
- Connectives ∧, ∨, ¬, ⇒, ⇔
- Equality ==
- Quantifier ∀, ∃

Atomic sentences

Atomic sentences are the most basic sentences of first-order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.

We can represent atomic sentences as Predicate (term1, term2, ..., term n).

Example: Ravi and Ajay are brothers:

=> Brothers(Ravi, Ajay).

Chinky is a cat:

=> cat (Chinky).

Complex Sentences

- Complex sentences are made by combining atomic sentences using connectives.
- First-order logic statements can be divided into two parts:
- **Subject:** Subject is the main part of the statement.
- **Predicate:** A predicate can be defined as a relation, which binds two atoms together in a statement.

- Consider the statement: "x is an integer.", it consists of two parts, the first part x is the subject of the statement and second part "is an integer," is known as a predicate.

Quantifiers in First-order logic

1. A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse.

2. These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.

3. There are two types of quantifier:

•Universal Quantifier, (for all, everyone, everything)

•Existential quantifier, (for some, at least one).

Universal Quantifier

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

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

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

•For all x

•For each x

•For every x.

Examples

1. All birds fly.

In this question the predicate is "fly(bird)."

And since there are all birds who fly so it will be represented as follows.

$$\forall x \text{ bird}(x) \rightarrow \text{fly}(x).$$

2. Every man respects his parent.

In this question, the predicate is "respect(x, y)," where x=man, and y= parent.

Since there is every man so will use ∀, and it will be represented as follows:

∀x man(x) → respects (x, parent).

Existential Quantifier

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

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

3. If x is a variable, then existential quantifier will be ∃x or ∃(x)

•There exists a 'x.'

•For some 'x.'

•For at least one 'x.'

Example

1. Some boys play cricket.

The predicate is "play(x, y)," where x= boys, and y= game. Since there are some boys so we will use ∃, and it will be represented as:

∃x boys(x) → play(x, cricket).

2. Only one student failed in Mathematics.

The predicate is "failed(x, y)," where x= student, and y= subject.

Since there is only one student who failed in Mathematics, so we will use following representation for this:

∃(x) [student(x) → failed (x, Mathematics) ∧∀ (y) [¬(x==y) ∧ student(y) → ¬failed (x, Mathematics)].