Knowledge Representation

Knowledge representation and mapping

- 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.

Kind of knowledge which needs to be represented

- **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

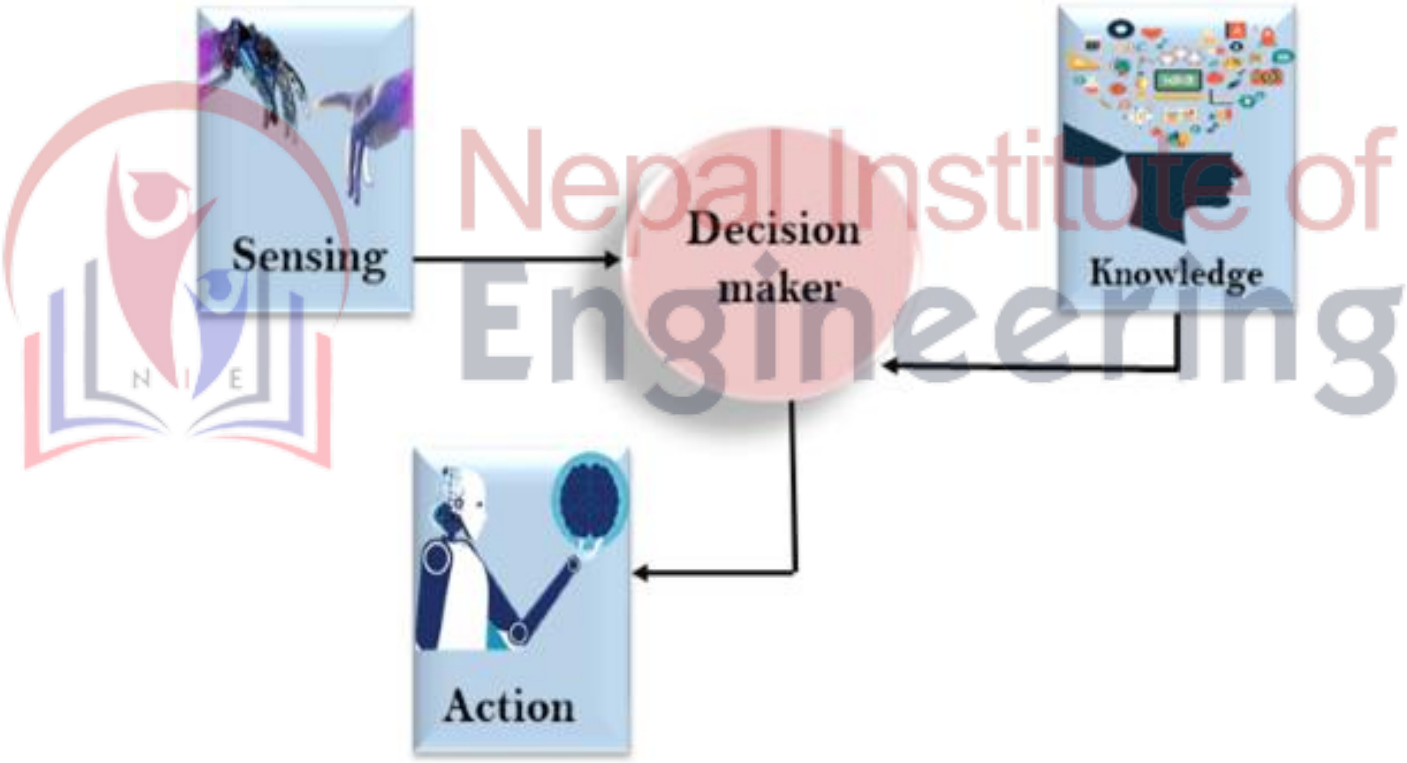
- **Declarative Knowledge:**
  - Declarative knowledge is to know about something.
  - It includes concepts, facts, and objects.
  - It is also called descriptive knowledge and expressed in declarativesentences.
  - It is simpler than procedural language.
- **Procedural Knowledge**
  - It is also known 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.

- **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.
- **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.

Relation between knowledge and intelligence

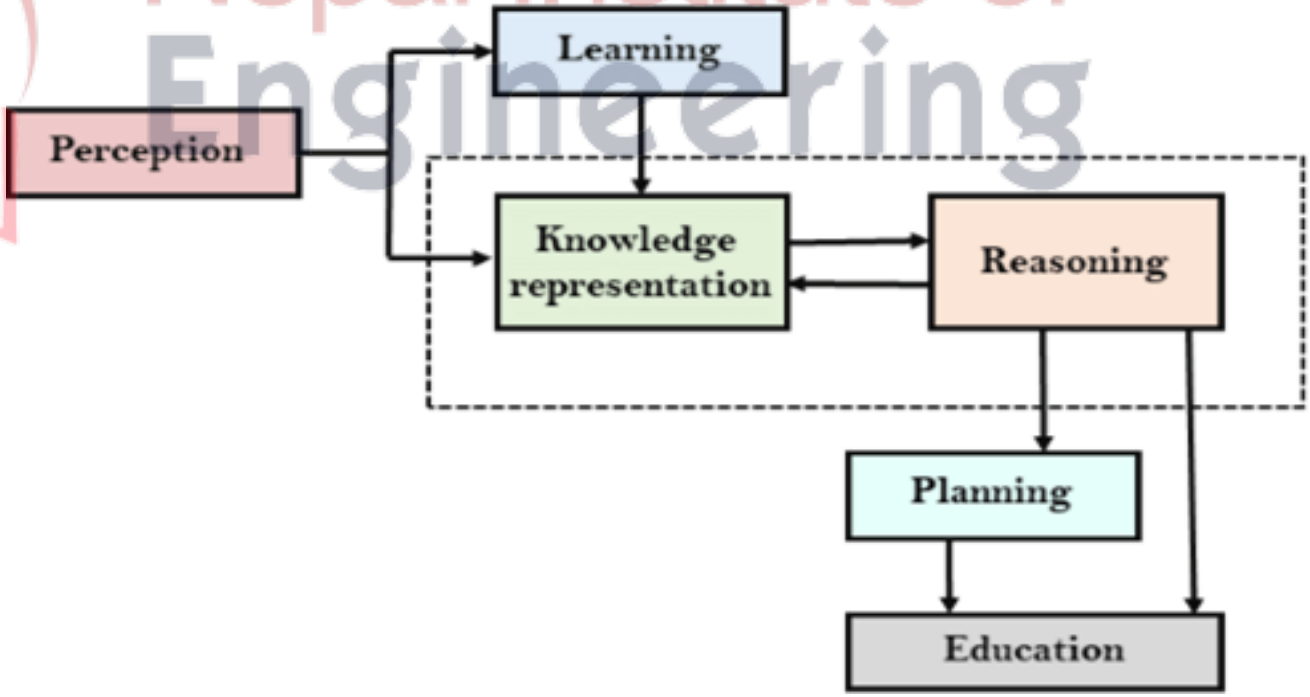
- Knowledge of real-worlds plays a vital role in intelligence and same for creating artificial intelligence.
- Knowledge plays an important role in demonstrating intelligent behavior in AI agents.
- An agent is only able to accurately act on some input when he has some knowledge or experience about that 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 diagram is showing how an AI system can interact with the real world and what components help it to show intelligence.
- AI system has Perception component by which it retrieves information from its environment.
- It can be visual, audio or another form of sensory input.
- The learning component is responsible for learning from data captured by Perception comportment.
- In the complete cycle, the main components are knowledge representation and Reasoning.
- These two components are involved in showing the intelligence in machine-like humans.
- These two components are independent with each other but also coupled together.
- The planning and execution depend on analysis of Knowledge representation and reasoning.

Approaches to knowledge representation

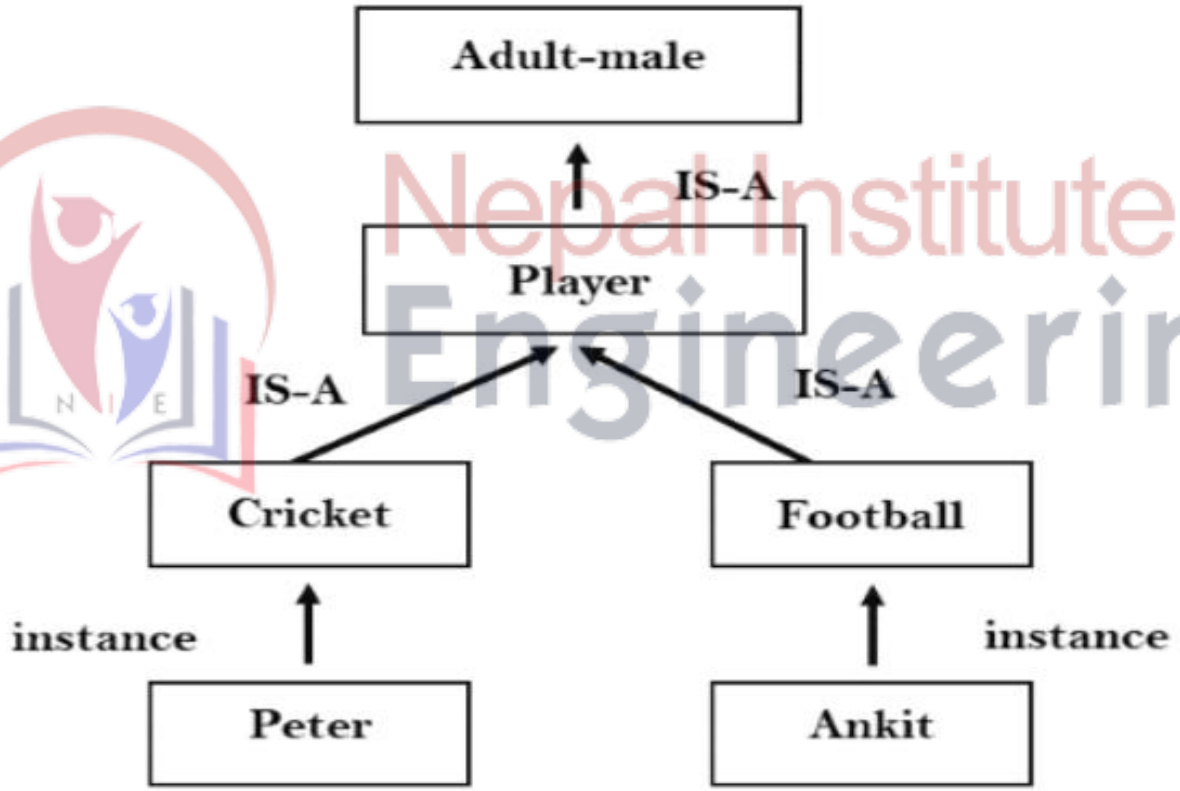
- Simple relational knowledge:
  - It is the simplest way of storing facts which uses the relational method, and each fact about a set of the object is set out systematically in columns.
  - This approach of knowledge representation is famous in database systems where the relationship between different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

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 hierarchical 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:**



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
  - All men 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 **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.

Issues in Knowledge Representation

✓Are any attributes of objects so basic that they occur in almost every problem domain? If there are, we need to make sure that they are handled appropriately in each of the mechanisms we propose. If such attributes exist, what are they ?

✓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? Is it helpful to use such primitives?

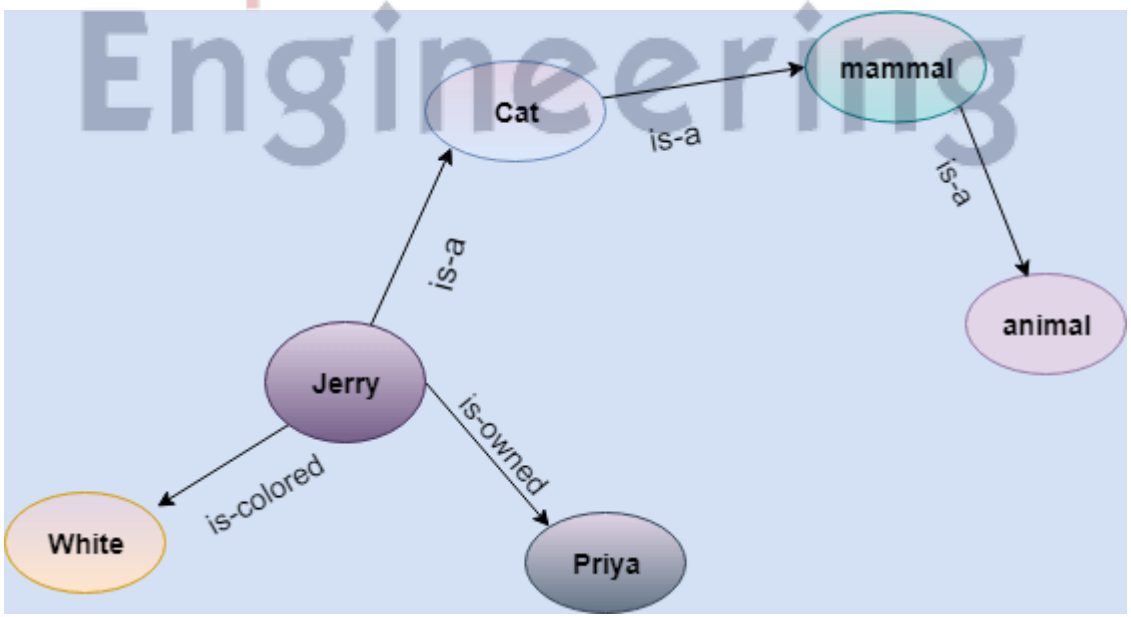
✓How should set of objects be represented?

✓Given a large amount of knowledge stored in a database, how can relevant parts be accessed whey they are needed?

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

- **Example:** Following are some statements which we need to represent in the form of nodes and arcs
- Statements:
  - Jerry is a cat.
  - Jerry is a mammal
  - Jerry is owned by Priya.
  - Jerry is white colored.
  - All Mammals are animal.



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: 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

Propositional logic

- Propositional logic (PL) 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.
- Example:
  - It is Sunday.
  - The Sun rises from West (False proposition)
  - 3+3= 7(False proposition)
  - 5 is a prime number.

Following are some basic facts about propositional logic

- Propositional logic is also called Boolean logic as it works on 0 and 1.
- 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.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and **logical connectives**.
- These connectives are also called logical operators.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.
- A proposition formula which is always true is called **tautology**, and it is also called a valid sentence.

- A proposition formula which is always false is called **Contradiction**.
- A proposition formula which has both true and false values is called contingency.
- 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.

Syntax of propositional logic

- The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:
- **Atomic Propositions**
- **Compound propositions**
- **Atomic Proposition:** Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

- **Example:**
  - 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."

Following table is of Propositional Logic Connectives

Connective symbols	Word	Technical term	Example
∧	AND	Conjunction	A ∧ B
∨	OR	Disjunction	A ∨ B
→	Implies	Implication	A → B
⇔	If and only if	Biconditional	A⇔ B
¬ or ~	Not	Negation	¬ A or ¬ B

Predicate Logic

- A predicate is an expression of one or more variables determined on some specific domain.
- A predicate with variables can be made a proposition by either authorizing a value to the variable or by quantifying the variable.

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:
  - **Objects:** A, B, people, numbers, colors, wars, theories, squares, pits, wumpus, ...
  - **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
  - **Function:** Father of, best friend, third inning of, end of, ...
- As a natural language, first-order logic also has two main parts:
  - **Syntax**
  - **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 syntactic elements of first-order logic are symbols. We write statements in short-hand notation in FOL.

Basic Elements of First-order logic

- Following are the basic elements of FOL syntax:

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

Quantifier

- The variable of predicates is quantified by quantifiers.
- There are two types of quantifier in predicate logic - Existential Quantifier and Universal Quantifier.

Existential Quantifier

- If p(x) is a proposition over the universe U.
- Then it is denoted as ∃x p(x) and read as "There exists at least one value in the universe of variable x such that p(x) is true.
- The quantifier ∃ is called the existential quantifier.
- There are several ways to write a proposition, with an existential quantifier, i.e.,
- (∃x∈A)p(x) or ∃x∈A such that p (x) or (∃x)p(x) or p(x) is true for some x ∈A.

Universal Quantifier

- If p(x) is a proposition over the universe U.
- Then it is denoted as $\forall x, p(x)$ and read as "For every $x \in U, p(x)$ is true."
- The quantifier $\forall$ is called the Universal Quantifier.
- There are several ways to write a proposition, with a universal quantifier.
- $\forall x \in A, p(x)$ or $p(x), \forall x \in A$ Or $\forall x, p(x)$ or $p(x)$ is true for all $x \in A$.

FOL inference rules for quantifier

- Following are some basic inference rules in FOL:
  - Universal Generalization
  - Universal Instantiation
  - Existential Instantiation
  - Existential introduction
- Universal Generalization
  - Universal generalization is a valid inference rule which states that if premise P(c) is true for any arbitrary element c in the universe of discourse, then we can have a conclusion as ∀ x P(x).
  - It can be represented as: .

$$\frac{P(c)}{\forall x\, P(x)}$$

  - This rule can be used if we want to show that every element has a similar property.
  - In this rule, x must not appear as a free variable.
  - Example: Let's represent, P(c): "A byte contains 8 bits", so for ∀ x P(x) "All bytes contain 8 bits.", it will also be true.

Universal Instantiation

- Universal instantiation is also called as universal elimination or UI is a valid inference rule.
- It can be applied multiple times to add new sentences.
- The new KB is logically equivalent to the previous KB.
- As per UI, **we can infer any sentence obtained by substituting a ground term for the variable**.
- The UI rule state that we can infer any sentence P(c) by substituting a ground term c (a constant within domain x) from **∀ x P(x) for any object in the universe of discourse**.
- It can be represented as:

$$\frac{\forall x\ P(x)}{P(c)}$$

- **Example:** IF "Every person like ice-cream"=> ∀x P(x) so we can infer that
  "John likes ice-cream" => P(c)

Existential Instantiation

- Existential instantiation is also called as Existential Elimination, which is a valid inference rule in first-order logic.
- It can be applied only once to replace the existential sentence.
- The new KB is not logically equivalent to old KB, but it will be satisfiable if old KB was satisfiable.
- This rule states that one can infer P(c) from the formula given in the form of ∃x P(x) for a new constant symbol c.
- The restriction with this rule is that c used in the rule must be a new term for which P(c) is true.
- It can be represented as:

$$\frac{\exists x\, P(x)}{P(c)}$$

Existential introduction

- An existential introduction is also known as an existential generalization, which is a valid inference rule in first-order logic.
- This rule states that if there is some element c in the universe of discourse which has a property P, then we can infer that there exists something in the universe which has the property P.
- It can be represented as:

$$\frac{P(c)}{\exists x P(x)}$$

- **Example: Let's say that,**
  "Priyanka got good marks in English."
  "Therefore, someone got good marks in English."

Unification

- Unification is a process of making two different logical atomic expressions identical by finding a substitution.
- Unification depends on the substitution process.
- It takes two literals as input and makes them identical using substitution.
- Let Ψ_1 and Ψ_2 be two atomic sentences and σ be a unifier such that, $\boldsymbol{\Psi_1\sigma = \Psi_2\sigma}$, then it can be expressed as $\mathbf{UNIFY(\Psi_1, \Psi_2)}$.

Resolution

- Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions.
- It was invented by a Mathematician John Alan Robinson in the year 1965.
- Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions.

- Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.
- **Clause**: Disjunction of literals (an atomic sentence) is called a **clause**. It is also known as a unit clause.
- **Conjunctive Normal Form**: A sentence represented as a conjunction of clauses is said to be **conjunctive normal form** or **CNF**.

Bayes' theorem

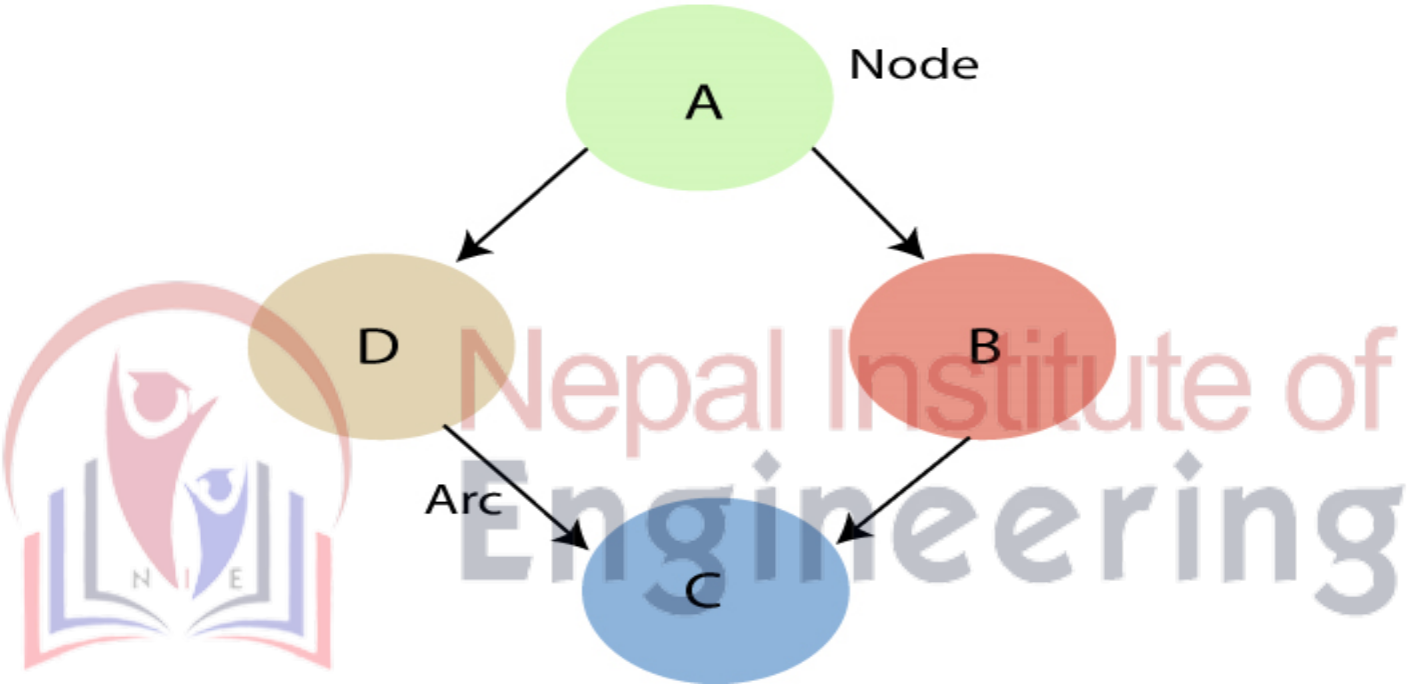
- Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.
- In probability theory, it relates the conditional probability and marginal probabilities of two random events.
- Bayes' theorem was named after the British mathematician **Thomas Bayes**.
- The **Bayesian inference** is an application of Bayes' theorem, which is fundamental to Bayesian statistics.
- It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad \text{....(a)}$$

Bayesian Belief Network

- A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph.
- It is also called a **Bayes network, belief network, decision network**, or **Bayesian model**.
- Bayesian networks are probabilistic, because these networks are built from a **probability distribution**, and also use probability theory for prediction and anomaly detection.
- Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:
  - **Directed Acyclic Graph**
  - **Table of conditional probabilities.**
- The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an **Influence diagram**.

- **A Bayesian network graph is made up of nodes and Arcs (directed links), where:**



- Each **node** corresponds to the random variables, and a variable can be **continuous** or **discrete**.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables.
- These directed links or arrows connect the pair of nodes in the graph.

- These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - **In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.**
  - **If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.**
  - **Node C is independent of node A.**
- The Bayesian network has mainly two components:
  - **Causal Component**
  - **Actual numbers**
- Each node in the Bayesian network has condition probability distribution **$P(X_i \mid Parent(X_i))$**, which determines the effect of the parent on that node.
- Bayesian network is based on Joint probability distribution and conditional probability

- Which is created by using single propositional symbol?
  a) Complex sentences
  b) Atomic sentences
  c) Composition sentences
  d) None of the mentioned
- Which is used to construct the complex sentences?
  a) Symbols
  b) Connectives
  c) Logical connectives
  d) All of the mentioned
- How many proposition symbols are there in artificial intelligence?
  a) 1
  b) 2
  c) 3
  d) 4
- Which are needed to compute the logical inference algorithm?
  a) Logical equivalence
  b) Validity
  c) Satisfiability
  d**) All of the mentioned**

- A _____ is used to demonstrate, on a purely syntactic basis, that one formula is a logical consequence of another formula.
  a) Deductive Systems
  b) Inductive Systems
  c) Reasoning with Knowledge Based Systems
  d) Search Based Systems
- The statement comprising the limitations of FOL is/are _____
  a) Expressiveness
  b) Formalizing Natural Languages
  c) Many-sorted Logic
  d) All of the mentioned
- First Order Logic is also known as _____
  a) First Order Predicate Calculus
  b) Quantification Theory
  c) Lower Order Calculus
  d) All of the mentioned
- How many terms are required for building a bayes model?
  a) 1
  b) 2
  c) 3
  d) 4

- Where does the bayes rule can be used?
  a) Solving queries
  b) Increasing complexity
  c) Decreasing complexity
  d) Answering probabilistic query
- What does the bayesian network provides?
  a) Complete description of the domain
  b) Partial description of the domain
  c) Complete description of the problem
  d) None of the mentioned
- How the entries in the full joint probability distribution can be calculated?
  a) Using variables
  b) Using information
  c) Both Using variables & information
  d) None of the mentioned
- Knowledge and reasoning also play a crucial role in dealing with _____ environment.
  a) Completely Observable
  b) Partially Observable
  c) Neither Completely nor Partially Observable
  d) Only Completely and Partially Observable

- Which is not a property of representation of knowledge?
  a) Representational Verification
  b) Representational Adequacy
  c) Inferential Adequacy
  d) Inferential Efficiency
- Which is not Familiar Connectives in First Order Logic?
  a) and
  b) iff
  c) or
  d) not
- Inference algorithm is complete only if _____
  a) It can derive any sentence
  b) It can derive any sentence that is an entailed version
  c) It is truth preserving
  d) It can derive any sentence that is an entailed version & It is truth preserving