

* What is Knowledge Representation:- Human 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.

1. Knowledge Representation and Reasoning is the part of AI which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.
2. 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 human in natural language.
3. It is also a way which describes how we can represent knowledge in AI. It is not just storing data into database. But it enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

There are some kind of knowledge which needs to be represent in AI:-

- (A) Object:- All the facts about objects in our world.
- (B) Events:- Events are the actions which occur in our world.
- (C) Performance:- It describe behaviour which involves knowledge about how to do things
- (D) Meta-knowledge:- It is knowledge about what we know.
- (E) Facts:- Facts are the truths about the real world and what we represent.
- (F) Knowledge - Base:- The Central Component of the knowledge-based agents is the knowledge base. It is represented as KB.

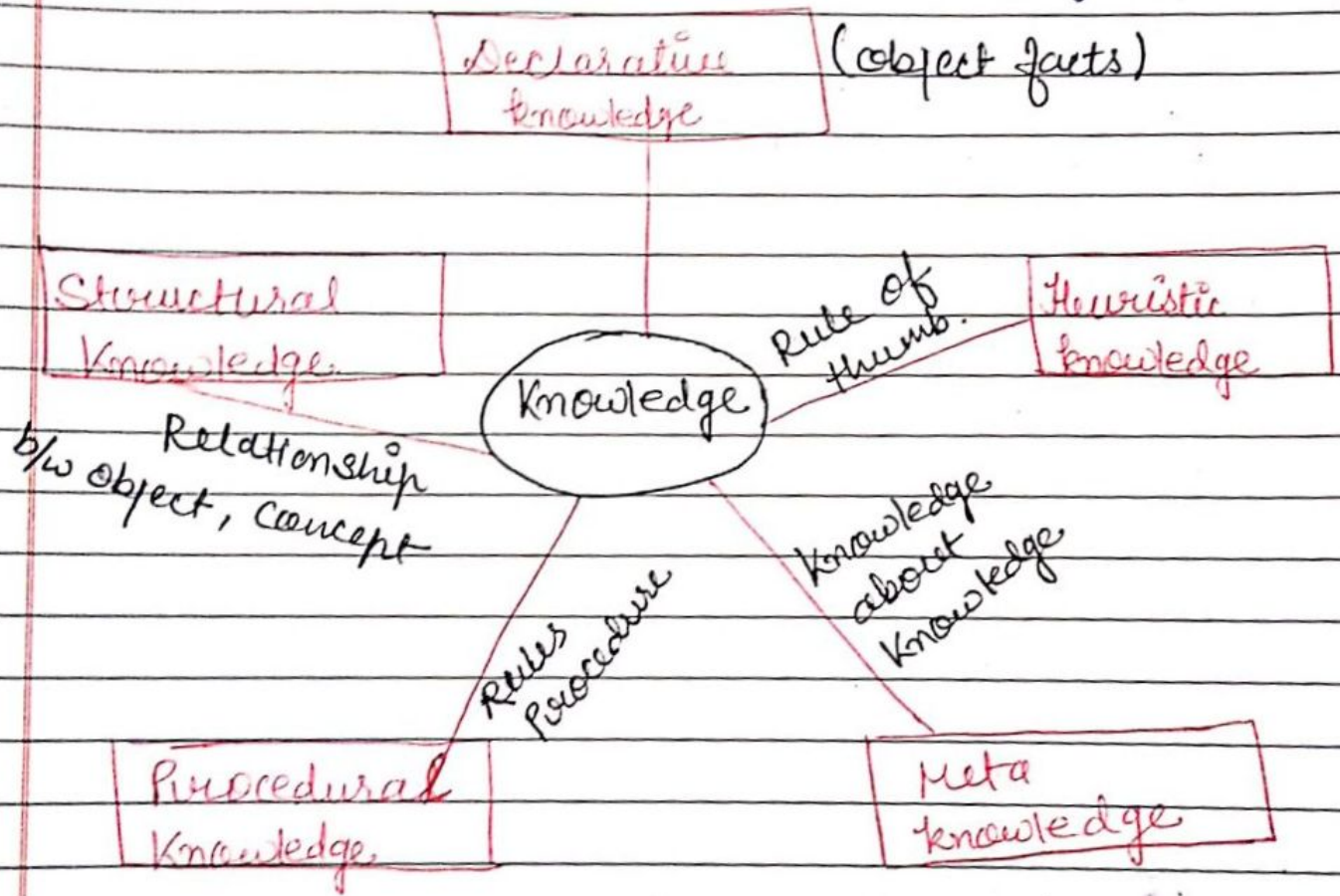
* Knowledge:- knowledge is awareness of facts, data and situations.

* Types of knowledge:-

(A) Declarative knowledge:-

- ① Declarative knowledge is to know about something.

2. It includes concepts, facts and objects.
3. It is also called descriptive knowledge and expressed in declarative sentences.
4. It is simpler than procedural language.



② Procedural Knowledge:-

1. It is also known as imperative knowledge.
2. Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.

3. it can be directly applied to any task.
4. It includes rules, strategies, procedures, agendas etc.
5. Procedural Knowledge depends on the task on which it can be applied.

③ Meta Knowledge:-

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

④ Heuristic knowledge:-

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

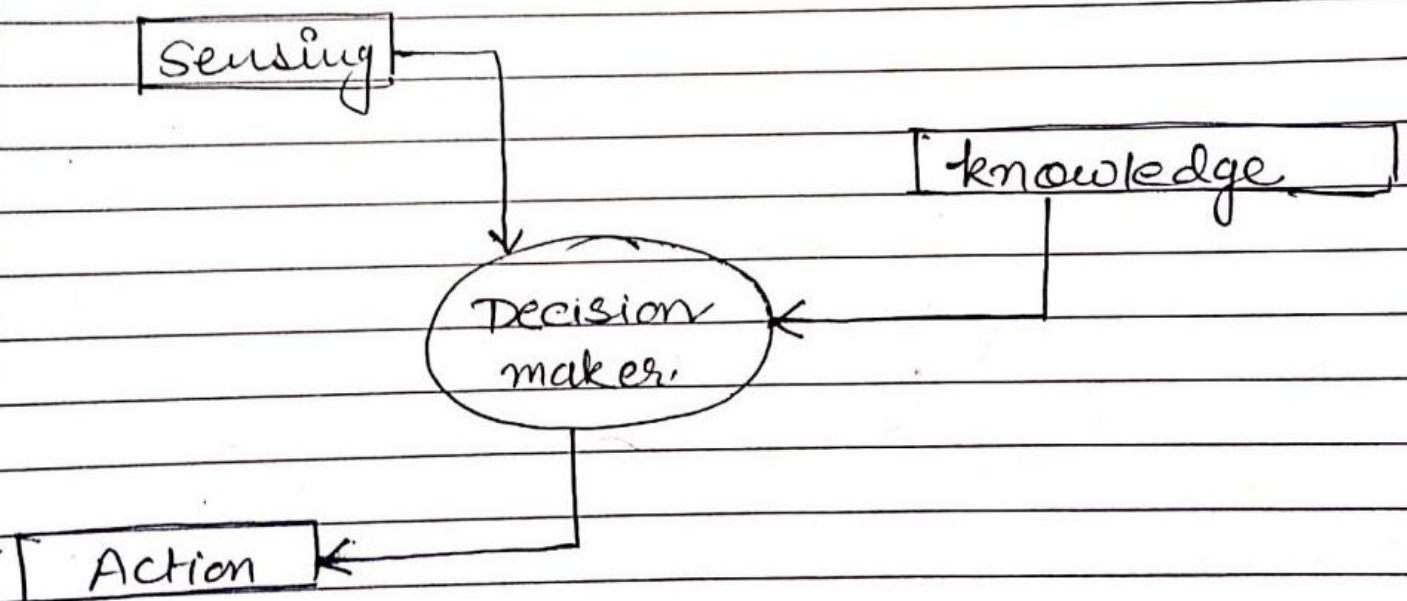
⑤ Structural knowledge:-

1. Structural knowledge is basic knowledge to problem-solving.
2. It describes relationships b/w various concepts such as kind of, part of, and grouping of something.
3. It describes the relationship that exists b/w concept.

Relation b/w Knowledge and Intelligence :-

Knowledge of Real-worlds plays a vital role in intelligence and same for creating AI. Knowledge plays an important role in demonstrating intelligent behaviours in AI agents.

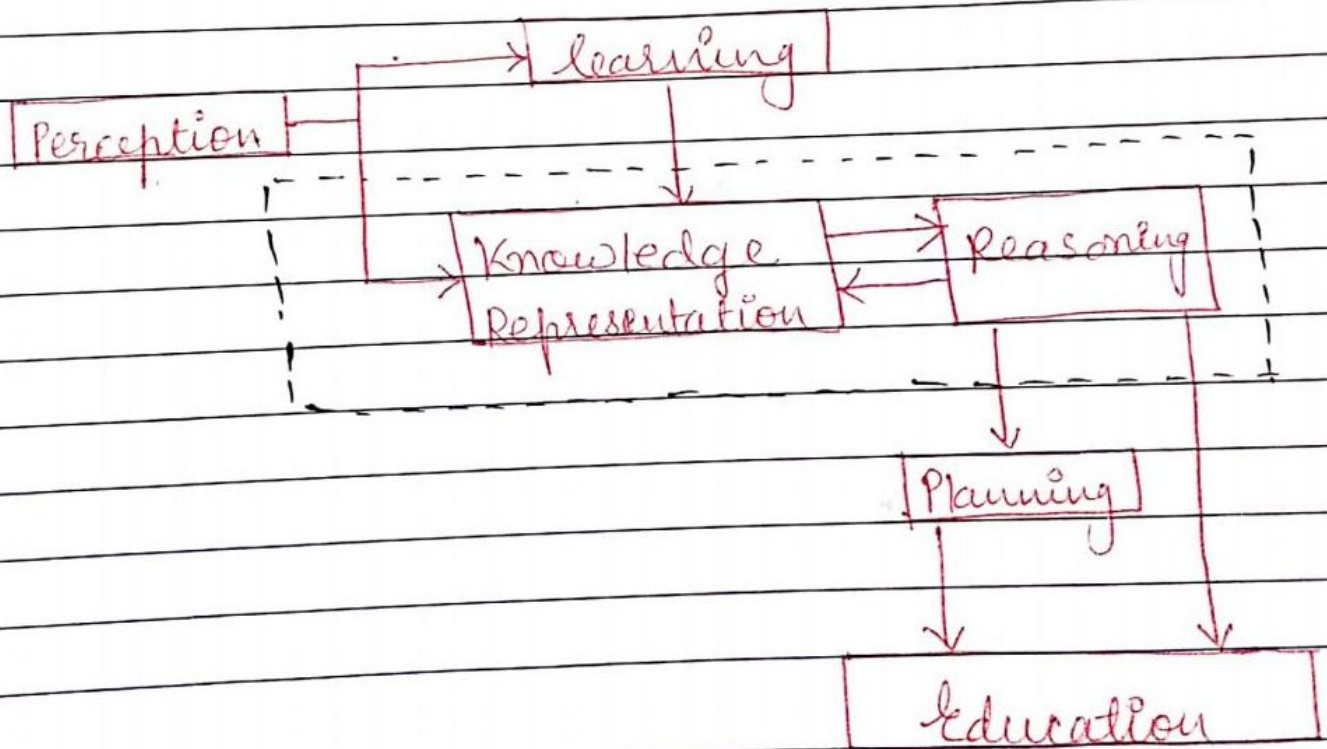
An agent is only able to accurately act on some input when he has some knowledge or experience about that input.



"Relationship b/w Knowledge and intelligence."

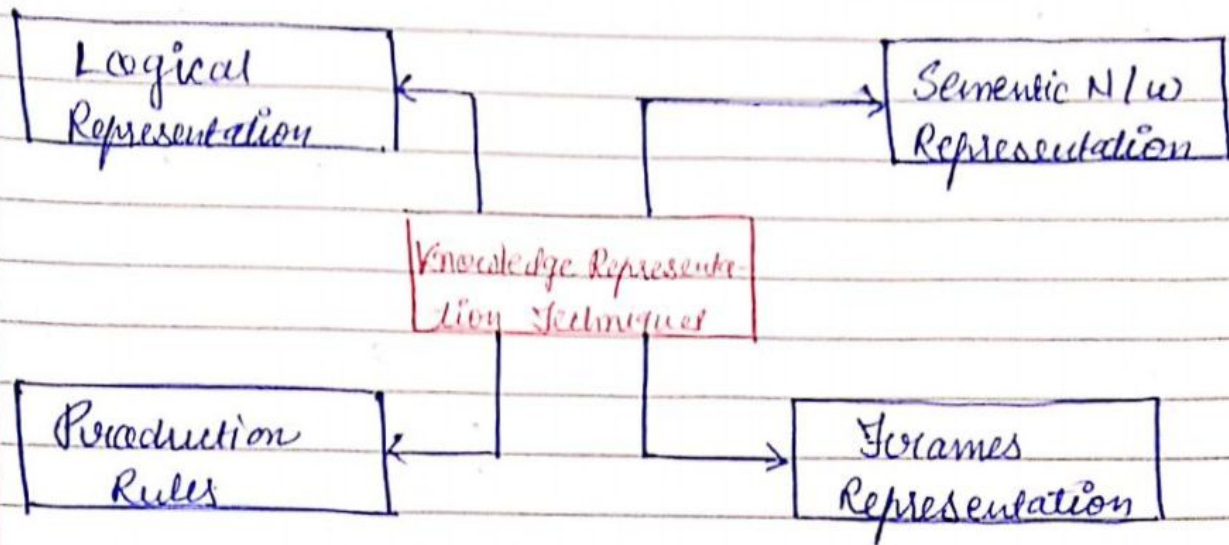
* AI Knowledge Cycle :- An AI System has the following components for displaying Intelligence behavior.

1. Perception
2. Learning
3. Knowledge Representation and Reasoning
4. Planning
5. Execution



"AI Knowledge cycle"

Techniques of Knowledge Representation in AI:-



"Techniques of KR"

① Logical Representation:- Logical Representation is a language with some definite rules which deal wth propositions and has no ambiguity in Representation.

It represents a conclusion based on various conditions and lays down some important communication rules.

Advantages:- ① Logical representation helps to perform logical reasoning.

② This representation is the basis for the programming languages.

Disadvantages:- ① Logical representation have some restrictions and are challenging to work with.

② This Techniques may not be very natural and inference may not be very efficient.

② Semantic Network Representation :- Semantic n/w work as an alternative of predicate logic for knowledge representation.

In this n/w you can represent your knowledge in the form of graphical n/w.

This network consists of nodes representing objects and arcs which describe the relationship b/w those objects. Also, it categorized the object in different forms and links those objects.

It is consists of two types of relations:-

① IS-A relation (Inheritance)

② kind-of-relation.

Advantages:- ① Semantic n/w are a natural representation of knowledge.

② These networks are simple and easy to understand.

Disadvantages:- ① Semantic n/w's take more computational time at runtime.

② These networks are not intelligent and depend on the creator of the system.

③ Frame Representation:- A frame is a record like structure that consists of a collection of attributes and value to describe an entity in the world.

These are the AI data structure that divides knowledge into substructures by

Advantages:- ① It makes the programming easier by grouping the related data.

② Frame representation is easy to understand and visualize.

③ It is very easy to add slots for new attributes and relations.

④ It is easy to include default data and search for missing values.

Disadvantages:- ① In frame system inference, the mechanism cannot be easily processed.

② The inference mechanism cannot be smoothly proceeded by frame representation.

③ It has a very generalized approach.

④ Production Rules:- 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~~ applied to a problem, whereas, the action part carries out the associated problem solving steps. This complete process is called a recognize-act cycle.

The production rules system consists of three main parts:-

① The set production rules.

② working Memory

③ The recognize-act-cycle.

Advantages:- ① The production rules are expressed in natural language.

② The production rules are highly modular and can be easily removed or modified.

Disadvantages:- ① It does not exhibit any learning capabilities and does not store the result of the problem for future uses.

② During the execution of the program, many rules may be active. Thus rule-based production systems are inefficient.

* Representation Requirements :- A good knowledge representation system must have properties such as:-

① Representational Accuracy :- It should represent all kinds of required knowledge.

② Inferential Adequacy :- It should be able to manipulate the representational structures to produce new knowledge corresponding to the 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 new knowledge easily using automatic methods.

* Approaches to knowledge Representation in AI:-

There are different approaches to knowledge Representation Such as:-

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. In this approach of knowledge representation is famous in database systems where the relationship between different entities is represented.

② Inheritable Knowledge:- In this approach all data must be sorted into a hierarchy of classes and should be arranged in a generalized form or a hierarchical manner.

It is also contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation and the objects and values are represented in boxed modes.

③ Inferential knowledge:- The inferential knowledge approach represents knowledge in the form of formal logic. Thus, it can be used to derive more facts and it guarantees correctness.

④ Procedural knowledge:-

① Procedural knowledge approach uses small programs and codes which describes how to do specific things, and how to proceed.

② In this approach, one important rule is used which is If-Then rule.

③ In this knowledge, we can use various coding languages such as LISP language and Prolog language.

④ we can easily represent heuristic or domain specific knowledge using this approach.

⑤ But it is not ~~necessary~~ necessary that we can represent all cases in this approach.

Requirements for Knowledge Representation System:-

A good Knowledge Representation System must possess the following properties:

- ① **Representational Accuracy**:- 'KR. System Should have the ability to represent all kind of required Knowledge.
- ② **Inferential Adequacy**:- Knowledge Representation 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 sorting appropriate guides.
- ④ **Acquisitional efficiency**:- The ability to acquire the new knowledge easily using automatic method

Propositional logic in AI:- Propositional logic 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 technique of knowledge representation in logical and mathematical form.

- Ex
- ① It is Sunday
 - ② The sun rises from west (false proposition)

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 this logic we use symbolic variable to represent the logic, and we can use any symbol for a representing the logic, and are ~~not~~ ~~in~~ propositional ~~pro~~ 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 ~~is~~ which is always true is called tautology, and it is also called a valid sentence.
- ⑨ A proposition formula which is always false is called contradiction.
- ⑩ A Propositional formula which has both true and false ~~satisfies~~ ~~is~~

The syntax of Propositional logic defines the allowable sentences for the knowledge representation. There are two types of propositions :-

- ① Atomic propositions
- ② Compound propositions

⑨ Atomic Proposition:- Atomic propositions are the simple propositions. It consists of a single proposition symbol. There are the sentences which must be either false or true.

Ex:- "The sun is cold" is also a proposition as it is a false fact.

⑩ Compound proposition:- Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

Ex:- "It is raining today, and street is wet".

* 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.

⑪ Negation:- A sentence such as $\neg P$ is called negation of P. A literal can be either positive literal or negative literal.

② Conjunction:- A sentence which has \wedge connective. Such as $P \wedge Q$ is called a conjunction.

Ex Rohan is intelligent and hardworking. It can be written as,

$P = \text{Rohan is intelligent,}$
 $Q = \text{Rohan is hardworking.} \rightarrow P \wedge Q$

③ Disjunction:- A sentence which has " \vee " connective, such as $P \vee Q$ is called disjunction, where P and Q are the propositions.

Ex: "Ritika is a doctor or engineer".

Here $P = \text{Ritika is Doctor.}$ $Q = \text{Ritika is doctor}$
So we can write it as $P \vee Q$

③ Implication:- A sentence such as $P \rightarrow Q$ 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 $P = \text{it is raining,}$ ~~then the street~~
and $Q = \text{Street is wet,}$ so it is represented as $P \rightarrow Q$.

- ⑤ Biconditional :- A Sentence such as $P \Leftrightarrow Q$ 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 $P \Leftrightarrow Q$.

* Properties of Operators :-

- ① Commutativity :- $P \wedge Q = Q \wedge P$ OR
 $P \vee Q = Q \vee P$.

- ② Associativity :- $(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$
 $(P \vee Q) \vee R = P \vee (Q \vee R)$

- ③ Identity element :- $P \wedge \text{True} = P$
 $P \vee \text{True} = \text{True}$.

- ④ Distributive :- ~~$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$~~
 $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$
 $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.

- ⑤ De Morgan's Law :- $\neg(P \wedge Q) = (\neg P) \vee (\neg Q)$
 $\neg(P \vee Q) = (\neg P) \wedge (\neg Q)$

- ⑥ Double-negation elimination :-
 $\neg(\neg P) = P$

Limitations of propositional logic :-

- ① We cannot represent relation like ALL, Some, or none with propositional logic.

Ex (a) All the girls are intelligent.

 (b) Some apples are sweet.

- ② Propositional logic has limited expressive power.

- ③ In propositional logic, we can't describe statements in terms of their properties or logical relationship.

* First order predicate logic (FOL) :-

- ① It is another way of Knowledge Representation in AI. It is an extension to propositional logic.

- ② FOL is sufficiently expressive to represent the natural language statements in a concise way.

- ③ FOL is also known as predicate logic or first-order predicate logic. It is a powerful language that develops information about the objects in a more easy way and can also express the relationship b/w those objects.

④ It 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-ary relation such as: The sister of, brother of, has color, comes between.

(C) Function:- Father of, best friend, third innings of, end of...

⑤ As a Natural language, first order logic also has two main parts:-

(a) Syntax

(b) Semantics.

(FOL)

Syntax of Predicate Logic:- The syntax of FOL determines which collection of symbols is a logical expression in first order logic.

Basic Elements of FOL:- following are the basic element of FOL syntax:-

Constant	1, 2, A, John, Mumbai, ...
variables	x, y, z, a, b, ...
Predicates	Brother, father, > < ...
function	3rd, left leg, ...
Connectives	$\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
Equality	$=$
Quantifier	\forall, \exists

Atomic Sentences:-

- ① Atomic Sentences are the most basic Sentence of first-order logic. These Sentences are formed from a predicate symbol followed by parameters parenthesis with a sequence of terms.
- ② we can represent atomic Sentences as predicate (term 1, term 2, ..., term n).

Complex Sentences:-

- ① Complex sentences are made by combining atomic sentences using connectives.

First order logic statements can be divided into two parts.

- ① **Subject:-** Subject is the main part of the statement.
- ② **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 predicate.

Quantifiers in first-order logic:-

- ① A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen or identify 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.

There are two types of quantifiers:-

(a) Universal Quantifier: (for all, everyone, everything)

(b) Existential Quantifiers (for some, at least one).

(A) Universal Quantifiers:- The universal Quantifier is represented by a symbol \forall , which resembles an inverted A.

If 'x' is a variable, then $\forall x$ is read as:-

\forall For all x

\forall For each x

\forall For every x

(B) Existential Quantifier:- Existential Quantifiers are the types 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 \exists , 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, the existential quantifier will be $\exists x$ or $\exists(x)$. And it will be read as:

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

Note:- ① The main connective for universal Quantifier \forall is implication \rightarrow .

② The main connective for existential Quantifier \exists is and \wedge .

* **Properties of Quantifiers:-** ① In universal Quantifier, $\forall x, \forall y$ is similar to $\forall y \forall x$.

② In existential Quantifier, $\exists x \exists y$ is similar to $\exists y \exists x$.

③ $\exists x \forall y$ is not similar to $\forall y \exists x$.

* Conceptual Dependency :- Conceptual Dependency Originally developed to represent knowledge acquired from natural language ip.

The goals of this theory are:-

- * To help in the drawing of inference from sentences.
- * To be independent of the words used in the original ip.
- * A structure into which nodes representing information can be placed.
- * A specific set of primitives
- * At a given level of granularity. (~~* Sentences~~ are represented as a series of diagrams depicting actions using ~~to~~ both abstract and real physical situations).
- * The agent and the objects are represented.
- * The actions are built up from a set of primitive acts which can be modified by tense.

Examples of primitive acts are:-

ATRANS Transfer of an abstract relationship.
Ex: give.

PTRANS Application of a physical ^{location of} ~~force~~ to an object.
Ex. ~~push~~ go.

PROPEL Application of a physical force to an object.
Ex: push

MTRANS Transfer of mental information.
Ex. tell.

MBUILD Construct new information from old.
Ex: decide.

SPEAK UHer sound.
Ex: Say

ATTEND Focus a sense on a stimulus
Ex. listen, watch.

MOVE Movement of a body part by owner.
Ex: Punch, kick.

GRASP Actor grasping an object
Ex: Clutch.

INGEST Actor ingesting an object.
Ex: Eat.

EXPEL Actor getting rid of an object from body.

Six primitive conceptual categories provides building blocks which are the set of allowable dependencies in the concepts in a sentence.

- (a) **PP** Real world objects.
- (b) **ACT** Real world actions.
- (c) **PA** Attributes of objects
- (d) **AA** Attributes of actions
- (e) **T** Times.
- (f) **LOC** Locations

~~After p11~~

Applications of Conceptual Dependency:-

- ① MARGIE (Meaning Analysis, Response Generation and Inference on English) - Model natural language understanding.
- ② SAM. (Script Applier Mechanism) - Scripts to understand stories. see next Section.
- ③ PAM (Plan Applier Mechanism) - Scripts to understand stories.

Advantages of CD:-

- ① Using ~~less~~ these primitives involves fewer inference rules.
- ② Many inference rules are already represented in CD structure.
- ③ The holes in the joints still to be established.

Disadvantages of CD:-

- ① Knowledge must be ~~developed~~ decomposed into fairly low level primitives.
- ② Impossible or difficult to find correct set of primitives.
- ③ A lot of inference may still be required.
- ④ ~~A lot of inference may~~
- ④ Representations can be complex even for relatively simple actions. ~~consider~~
- ⑤ Complex representations require a lot of storage.

Situation Calculus :- Situation calculus is that states are definable in terms of the actions required to reach them. These reachable states are called situations. Situation calculus can be seen as ~~an~~ a relational version of the feature-based ~~situation~~ representation of actions.

Here we only consider single agents, a fully observable environment, the deterministic actions.

Situation calculus is defined in terms of situations. A situation is either.

- * $init$, the initial situation or,
- * $do(A, S)$, the situation S , resulting from doing action A in situation S , if it is possible to do action A in situation S .

A situation can be associated w/ a state.

There are two main differences b/w situations and states :-

- ① Multiple situations may refer to the same state if multiple sequences of actions lead to the same state. That is equality b/w situations is not the same as equality b/w states.

- ② Not all States have corresponding Situations. A state is reachable if a sequence of actions exists that can reach that state from the initial state. States that are not reachable do not have a corresponding situation.

Some $do(A, S)$ terms do not correspond to any state. However, sometimes an agent must reason about such a (potential) situation without knowing if A is possible in state S , or if state S is ~~possible~~ possible.

A Static Relation is a relation for which the truth value does not depend on the situation; that is, its truth value is unchanging through time.

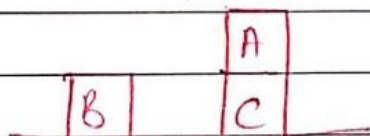
A Dynamic relation is a relation for which the truth value depends on the situation. To represent what is true in a situation, predicate symbols denoting dynamic relations have a situation argument, so that the truth can depend on the situation.

A predicate symbol with a situation argument is called a fluent.

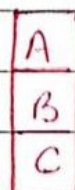
* What is Planning in AI?

- ① The planning in AI is about the decision making tasks performed by the robots or computer programs to achieve a specific goal.
- ② The execution of planning is defined as is specific about choosing a sequence of actions w/ a high likelihood to complete the specific task.

* Block-world planning problem:-



Start State



Goal State

- ① The blocks-world problem is known as Sussman anomaly.
- ② When two subgoals G_1 and G_2 are given a non-interleaved planner produces either a plan for G_1 concatenated w/ a plan for G_2 or vice-versa.
- ③ In block-world problem, three blocks labeled as

'A', 'B', 'C' are allowed to rest on the flat surface. The given condition is that only one block can be moved at a time to achieve the goal.

- (1) The start state and goal state are already shown in diagram.

Components of planning system:-

- (1) choose the best rule for applying the next rule based on the best available heuristics.
- (2) apply the chosen rule for computing the new problem state.
- (3) detect when a solution has been found.
- (4) detect dead ends so that they can be abandoned and the system's efforts is directed in more fruitful direction.
- (5) detect when an almost correct solution has been found.

Goal stack planning:- This is one of the most important planning algorithm, which is specifically used by STRIPS.

- ① The stack is used in an algorithm to hold the action and satisfy the goal. A knowledge base is used to hold the current state, action
- ② Goal stack is similar to a node in a search tree, where the branches are created if there is a choice of an action.

Non-linear planning:- This planning is used to set a goal stack and is included in the search space of ~~all~~ all possible subgoal orderings. It handles the goal inter-actions by interleaving method.

Advantage of Non-linear planning:- Non linear planning may be an optimal solution with respect to plan length (depending on search strategy used).

Disadvantages of Nonlinear planning:-

- ① It takes larger search space, ~~the~~ since all possible goal orderings are taken into consideration
- ② Complex algorithm to understand.

Algorithm:-

- (1) choose a goal 'g' from the goalset
- (2) If 'g' does not match the state, then:
 - (A) choose an operator 'o' whose add-list matches goal 'g'.
 - (B) Push 'o' on the opstack.
 - (C) Add the preconditions of 'o' to the goalset.
- (3) while all preconditions of operator on top of opstack are met in state.
 - (A) Pop operator 'o' from top of opstack
 - (B) State = apply(o, state)
 - (C) Plan = [Plan; o]

Probabilistic reasoning in Artificial Intelligence

* **Uncertainty** : Knowledge Representation logic with certainty, which means we were sure about the predicates, with this knowledge representation, we might write $A \rightarrow B$ which means if A is true then B is true, but consider a situation where we are not sure about whether A is true or not then we cannot express this statement, this situation is called **Uncertainty**.

Causes of uncertainty : —

1. Information occurred from unreliable sources
2. Experimental errors
3. Equipment fault
4. Temperature variation
5. Climate change.

Probabilistic reasoning:- Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability reasoning. We combine probability theory with logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.

Need of Probabilistic reasoning in AI:

- ① When there are unpredictable outcomes
- ② When specifications or possibilities of predicates becomes too large to handle.
- ③ When an unknown error occurs during an experiment.

In probabilistic reasoning, there are two ways to solve problems with uncertain knowledge:

- ① Bay's Rule
- ② Bayesian Network.

Probability :- ~~Prop~~ probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

We can find the probability of an uncertain event by using the below formula.

$$\text{Probability of occurrence} = \frac{\text{No. of desired outcome}}{\text{Total number of outcomes}}$$

① $P(\neg A) = \text{probability of a not happening event}$

② $P(\neg A) + P(A) = 1$

Event :- Each possible outcomes of a Variable is called event.

Sample Space :- The collection of all possible events is called Sample Space.

Random variables:- Random variables are events and objects in the real world used to represent the.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior probability:- The probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of 'B', it can be written as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

where $P(A \cap B)$ = Joint probability of A and B

$P(B)$ = Marginal Probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:-

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Bayes's theorem:- It is also known as Bayes rule, Bayes law, which is determines the probability of an event with uncertain knowledge.

Bayes's Theorem was named after the British mathematician Thomas Bayes's. The Bayesian inference is an application of Bayes's theorem, which is fundamental to Bayesian statistics.

It is a way to calculate the value of $P(B|A)$ with the knowledge of $P(A|B)$.

Bayes's theorem allows updating the probability prediction of an event by observing new information of the real world.

Applying Bayes's rule:- Bayes's rule allows us to compute the single term $P(B|A)$ in terms of $P(A|B)$, $P(B)$ and $P(A)$. This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Bayes's rule becomes:-

$$P(\text{cause}|\text{effect}) = \frac{P(\text{effect}|\text{cause}) P(\text{cause})}{P(\text{effect})}$$

* Applications of Bay's Theorm:-

- ① It is used to calculate the next step of the robot when the already executed step is given
- ② Baye's Theorm is helpful in weather forecasting.
- ③ It can solve the monty hall problem.

*

Bayesian Belief Network in Artificial Intelligence:-

" 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 Network are probabilistic, because these nw are built from a probability distribution, and also use probability theory for prediction and anomaly detection.

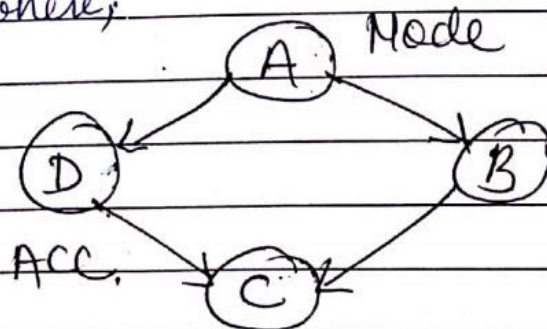
It can also be used in various tasks including prediction, anomaly detection, diagnostics, automatic insight, reasoning, time series prediction and decision making under uncertainty.

Bayesian N/w can be used for building models from data and experts opinions, and it consists of two parts

- (A) Directed Acyclic graph
- (B) Table of Conditional probabilities.

The generalized form of Bayesian N/w 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;



① Each node corresponding to the random variables, and variable can be continuous or ~~also~~ discrete.

② Arc or directed arrows represent the Causal relationship or conditional probabilities between random variables.
~~These links represents that the case~~

These directed links or arrows connect the pair of nodes in the graph.

The Bayesian NW has mainly two components:-

① Causal component

② Actual numbers.