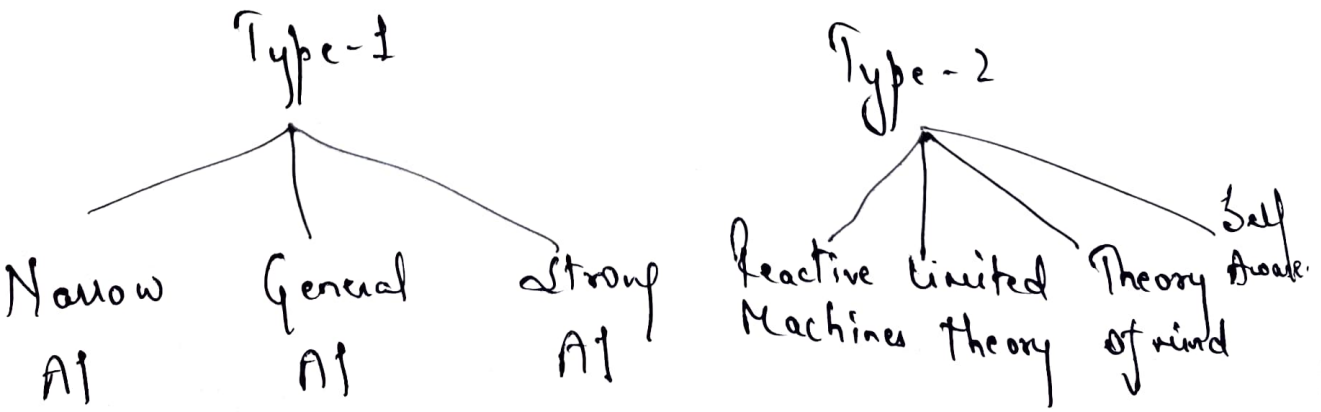
Artificial Intelligence

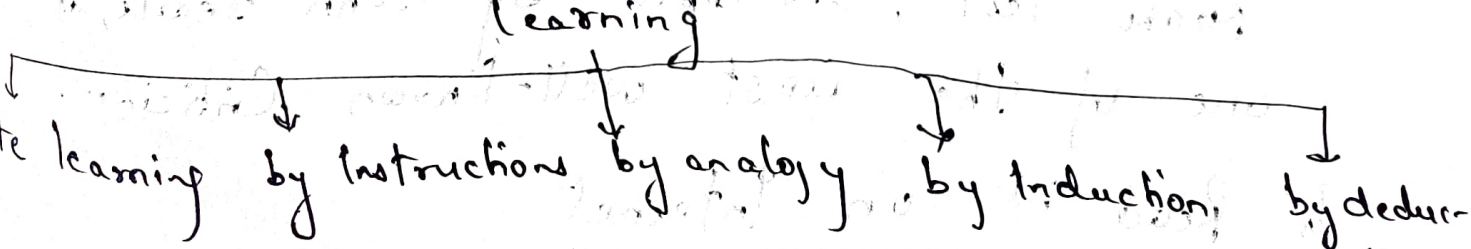
- Artificial Intelligence (AI) refers to the integration of intelligence into machines, indicating the machine's capacity to exhibit human-like skills such as reasoning, learning, planning, and creativity.

- AI can be categorised into the following levels:
  - Software level (i.e. Embodied AI) like search engines, virtual assistants, speech and facial recognition systems, picture analysis tools, and other things like that.
  - Hardware level (Embedded) includes robots, autonomous vehicles, drones, IOT etc.

- On basis of functionalities AI can be classified based on:



- Artificial Narrow Intelligence, also called weak AI or Narrow AI. Refers to simulated thinking where systems exhibits apparent intelligence without being self-aware. Eg: Chatbot
- Artificial General Intelligence, also called Strong AI: Refers to as 'true' thinking, involves behaving and reasoning like a intelligent human with a conscious and subjective mind. Eg: Self driving vehicles.
- Artificial Super Intelligence: Surpasses human intelligence. Ability to make decisions, rationalise choices, and even engage in activities like forming emotional relationships.
- Components of Intelligence:
  - learning
  - reasoning
  - problem solving
  - perception
  - language comprehension



- Learning based on feedback is normally categorised as Supervised, Unsupervised and Reinforcement (Rewards) for learning.

- Approaches to Artificial Intelligence:

  - Turing Test: for the test, three rooms will be utilised. One of these rooms contains a computer system that is described as intelligent. In each of the other two rooms, there is one person sitting. One of the people, who we'll call C, is suppose to ask question of the computer. And the other person who we'll call B, without knowing who each question is for and ofcourse, with the goal of figuring out who the computer is. On the other hand, the computer would reply in a way that would keep C from finding out who it is.

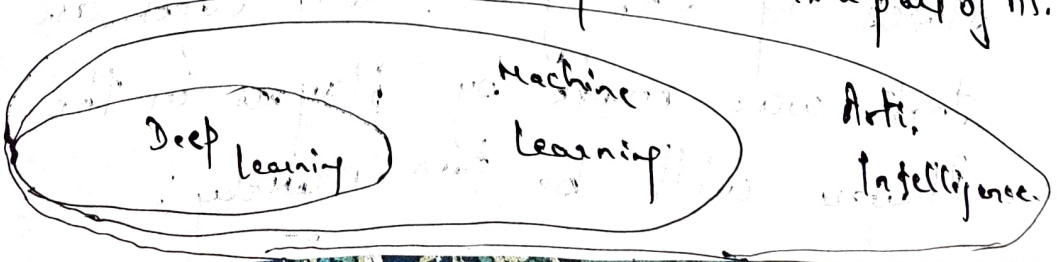
  - Criticism to Turing Test: The Chinese Room test, developed by John Searle, is one of the most well-known Criticism.
  - Read it on Internet

↳ Thinking Humanly or The Cognitive Modelling:
→ Introspection, which means to look at our own thoughts and use those thoughts to build a model.
→ Psychological Experiments, which means running tests on people and looking how they act.
→ Brain Imaging, which means to use an MRI to study how the brain works in different situations and then copy that through code.

↳ Thinking Rationally i.e. The Laws of Thought approach: Long list of logical statements that tell our ~~mind~~ minds how to work.

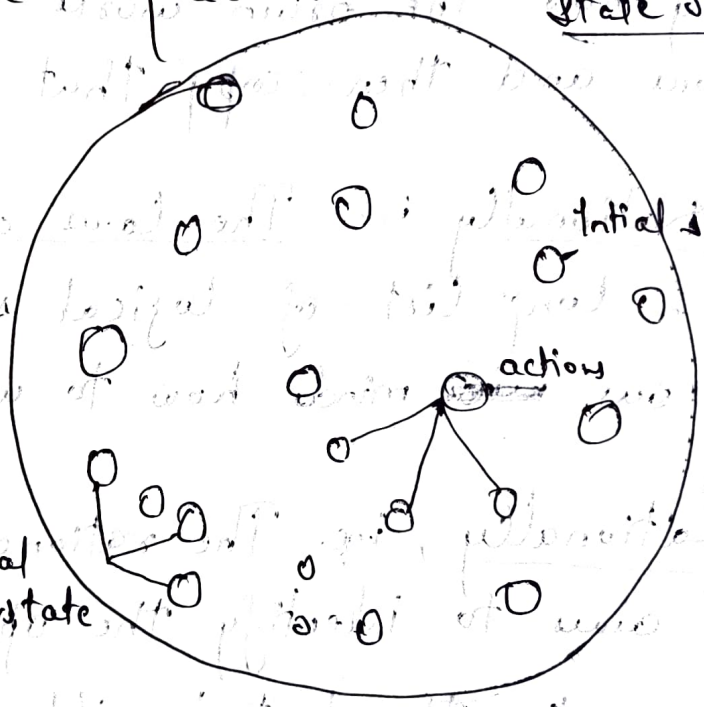
↳ Acting Rationally, i.e. The rational agent approach: aims to identify the optimal outcome, striving to make the best possible outcome, considering prevailing circumstances. The laws of thought, on the other hand, says that a thing must act in a way that makes sense.

AI, ML and DL relationships: ML is a part of AI. and DL is a type of ML.



- State Space Search: Encompasses all reachable states starting from the initial state, forming a graph where states are nodes and actions are connections between nodes. In the context of state space, a path is defined as a series of states linked by sequence of action.

State space



Example: Water - Jug Prob. (WJP).

- Min - Max Strategy: Let's understand this strategy via game tree where nodes are states of the game, and edges are moves that were made by players in the game. Players will be two namely:

↳ MIN: Decrease the chances of MAX to win the game.

↳ MAX: Increases his chances of winning game.

MINMAX Algorithm: is a backtracking algorithm where it backtracks to pick the best move out of several choices. MINMAX strategy follows DFS concept. (DFS => Depth-first search)

Steps:

↳ Generate a whole game tree, all the way down to the terminal states.

↳ Apply the utility function to each terminal state to get its value.

↳ Use the utility of the terminal states to determine the utility of the nodes one level higher up in the search tree.

↳ Continue backing up the values reached this top of the tree. At that point, MAX chooses the move that leads to highest value.

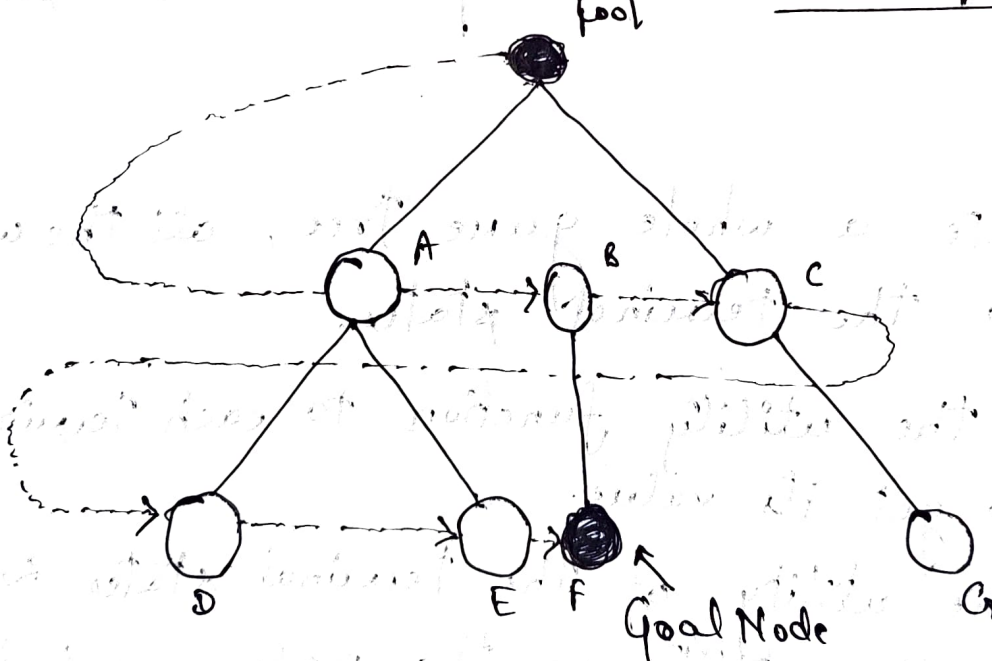
Properties: ↳ complete ↳ optimal

↳ Time Complexity $O(b^m)$, where b: branching factor

↳ Space Complexity $O(bm)$ m: max. depth of game tree

Breadth-first Search (BFS): All nodes at a particular depth in a search tree are expanded before moving on to nodes at the next level. This implies that the immediate children of nodes are thoroughly explored before considering any descendants at the next level.

Search Tree for BFS



Steps: Algorithm (BFS is a brute-search) uses data structure that works on FIFO principle.

- Initialise: Set = {s}, where s is a start state.
- Fail: If OPEN = {}, terminate with failure.
- Select: Remove a left most state (say a) from OPEN.
- Terminate: If a @ Goal node, terminate with success

↳ Expand: Generate the successor of node a; discard the successors of a if it's already in OPEN, Insert only remaining successors on right end of OPEN (i.e., QUEUE)

↳ Loop: Go to Step 2.

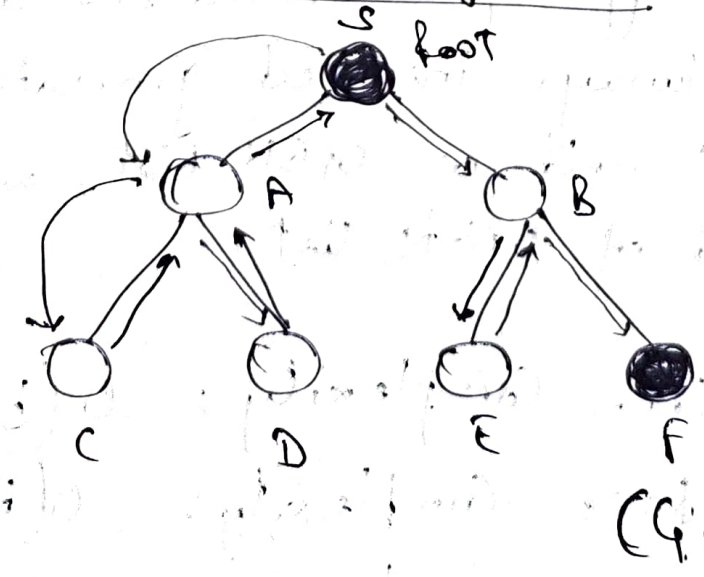
Properties:

↳ Time Complexity is number of nodes generated, ∴ TC of BFS is $O(b^d)$

↳ Space Complexity is given by: $1+b+b^2 \dots b^d = O(b^d)$.

- Depth-first Search (DFS): Characterised by the strategy of descending as deeply as possible into the search tree or graph before backtracking and exploring alternative paths.

Search Tree for DFS



DFS Algorithm:

Steps:

1. Initialise: Set = {s}, where s is a start state.
2. Fail: If OPEN = {}, terminate with failure.
3. Select: Remove a left most state (say a) from OPEN.
4. Terminate: If a ∈ Goal node, terminate with success, else.
5. Expend: Generate a successor of node a, discard the successor of a if it's already in OPEN, Insert remaining successors on left end of OPEN [i.e., STACK].
6. Loop: Go to step 2.

Note: Only diff. b/w BFS & DFS is step 5. In BFS, we always inserted generated successors at the right end of OPEN List, whereas DFS is at left end of OPEN list.

Properties:

Time Complexity: $O(b^m)$

Space Complexity: $O(bm)$

Informed (Heuristic) Search: It can solve much complex problem which could not be solved in another way. We have following informed search algo:

- Best-first Search
- A* algo
- Iterative Deepening A*

Heuristic function: Heuristic info. is provided in form of function called H.f. Denoted by $h(n)$ = estimated cost of cheapest path from node n to a goal node.

Example: Suppose you want to find a shortest path from Kolkata to Guwahati, then heuristic for Guwahati may be straight-line distance b/w. Kolkata & Guwahati i.e.:

h(Kolkata) = euclidean Distance (Kolkata, Guwahati)

- A* algo represents an OR graph algo. that is used to find a single solution (either this or that).
- AO* algo represents an AND OR graph algo that is used to find more than one solution.
- A* algo. guarantees to give an optimal solu., AO* doesn't

- Propositional Rules of Inference: described below:

↳ Modus Ponens (MP): If the propositions A→B and A are true, then B is also true.

$\alpha_1 \to \alpha_2$, $\alpha_1 \Rightarrow \alpha_2$

↳ Modus Tollens (MT): If $A \to B$ and $\neg B$ are true then $\neg A$ are also true.

$\alpha_1 \to \alpha_2$, $\sim a_2 \Rightarrow \sim \alpha_1$

↳ Disjunctive Syllogism (DS): affirms the truthfulness of the other proposition if one of the propositions in a disjunction is false.

Rule 1: $\alpha_1 \vee \alpha_2$, $\sim \alpha_1 \Rightarrow \alpha_2$

Rule 2: $\alpha_1 \vee \alpha_2$, $\sim \alpha_2 \Rightarrow \alpha_1$

↳ Addition: If a proposition is true, then its disjunction with any other proposition is also true.

Rule 1: $\alpha_1 \Rightarrow \alpha_1 \vee \alpha_2$

Rule 2: $\alpha_2 \Rightarrow \alpha_1 \vee \alpha_2$

↳ Simplification: if we have a conjunction, then both the constituent propositions are also

true.

Rule 1: $\alpha_1 \wedge \alpha_2 \Rightarrow \alpha_1$

Rule 2: $\alpha_1 \wedge \alpha_2 \Rightarrow \alpha_2$

- Conjunction: if two propositions are true, then their conjunc. is also true.

$$\alpha_1, \alpha_2 \Rightarrow \alpha_1 \wedge \alpha_2$$

- Hypothetical Syllogism (HS): Conclusion $\alpha_1 \rightarrow \alpha_3$ is true, whenever conditional statements $\alpha_1 \rightarrow \alpha_2$ and $\alpha_2 \rightarrow \alpha_3$ hold truth values.

$$\alpha_1 \rightarrow \alpha_2, \quad \alpha_2 \rightarrow \alpha_3 \Rightarrow \alpha_1 \Rightarrow \alpha_3$$

- Absorption: if literal α_1 conditionally implies another literal α_2, i.e, $\alpha_1 \rightarrow \alpha_2$ is true, then $\alpha_1 \rightarrow (\alpha_1 \wedge \alpha_2)$ also holds.

$$\alpha_1 \rightarrow \alpha_2 \Rightarrow \alpha_1 \rightarrow (\alpha_1 \wedge \alpha_2)$$

- Constructive Dilemma (CD): if proposition $(\alpha_1 \vee \alpha_3)$ and $(\alpha_1 \rightarrow \alpha_2) \wedge (\alpha_3 \rightarrow \alpha_4)$ have true values, the $(\alpha_2 \vee \alpha_4)$ holds a true value.

$$(\alpha_1 \vee \alpha_3), \quad (\alpha_1 \rightarrow \alpha_2) \wedge (\alpha_3 \rightarrow \alpha_4)$$

$$\Rightarrow \alpha_2 \vee \alpha_4$$

- Skolomisation: Refinement of Prenex Normal form (PNF) called (Skolem) Standard form, is the basis of problem solving through Resolution Method.

The Standard form of a formula of first Order Predicate Logic is obtained through the following three steps:

↳ The given formula should be converted to PNF, and then,

↳ Convert the matrix of PNF, i.e, the quantifier-free part of PNF into Conjunctive Normal form.

↳ Skolomisation: Eliminate the existential quantifiers using skolem constants and functions.

- Rule Based System: Consists of:

↳ Rule base, which is a set of IF-THEN rules.

↳ A bunch of facts.

↳ An interpreter of facts and rules functions as a mechanism that determines the appropriate rule to apply from given set of available facts.

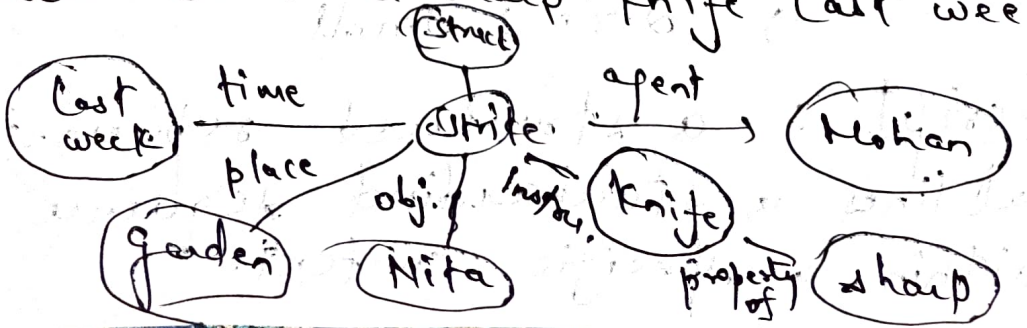
Kinds of Rule Based System:

↳ Forward Chaining System: processes sequentially starting with facts, moving through rules, & ultimately reaching goals. In FCS, rules are typically represented in standard implicational form, where antecedent or condition part comprises positive literals, and the consequent or conclusion part includes a positive literal.

↳ Backward Chaining System: does not need to update a working memory. Instead it needs to keep track of what goal it needs to prove it's main hypothesis. So, in this reasoning proceeds 'backward', beginning with the goal to be established, chaining through rules, and finally anchoring in facts.

Semantic Network: is essentially a graph, where the nodes symbolise concepts, and the labeled arcs denote binary relationships between these concepts. Eg: Mohan struck Nita in garden with a sharp knife last week.

Semantic Network



- Frames : serve as a variation of semantic networks and stand as one of the widely adopted methods for representing non-procedural knowledge in an expert system. Frames look like the data structure, record.

We may represent some knowledge about a lion in frames as follows:

↳ Mammal:
- → Subclass : Animal
- → Warm_blooded : Yes

↳ Lion:
- → Subclass : Mammal
- → Eating habbit : Carnivorous
- → Size : medium

↳ Raja :
- → instance : Lion
- → Colour : dull-yellow
- → Owner : Amar Circus

↳ Shera:
- → Instance : Lion
- → Size : Small

- Script : a structured representation that outlines a standardised sequence of events within a specific context.

Bayes' Theorem: also known as Bayes rule or Bayes law. It is used to determine the conditional probability of event A when event B has already happened. The general statement of Bayes' Theorem is:

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

P(A) and P(B) are probabilities of events A and B.
P(A|B) is probability of A when event B happens
P(B|A) is probability of event B when A happens

Dempster-Shafer Theory: is an evidence theory, it combines all possible outcomes of the problem. Hence it is used to solve prob. of, where there may be a chance that a piece of different evidence will lead to some different result.

Core functions of DS Theory:
- Basic Probability Assignment fn.
- The Belief fn.
- Plausibility fn.
- Rule of combination

- Rough Set Theory: It is a mechanism to deal with imprecise / imprecisive knowledge, & it is highly comprehensive and can be treated as distinct and independent discipline.

- fuzzy Reasoning is taken care by:

↳ Non-Monotonic Reasoning System:

Components:

↳ Knowledge Base (KB)

↳ Inference Engine (IE)

↳ Truth-Maintenance System (TMS)

↳ Default Reasoning:

↳ Closed World Assumption.