UNIT-3 :-

GAME PLAYING & CSP

* Game theory
* Optimal decision in games
* Alpha-beta search
* Monte-carlo tree search
* Stochastic games
* Partially observable games (General)
* Constraint satisfaction problems
* Constraint propagation
* Backtracking search for CSP
* Local search for CSP
* Structure of CSP.

Game theory :-

Eg: Card game

* Game playing is a search problem difined by :- (AI has to decide :)
  - → Initial state (where it has to start)
  - → Successor function (After that, where it has to go to next func)
  - → Goal test (So, we want to win the game. AI decides what is the process at or state it has to go through to win the game)
  - → path cost

→ AI has continued to improve, with aims that a player should be unable to tell wether if he's playing with a human or a computer.

→ A game must feel natural :-
  - * AI should obey laws
  - * Character aware of the Enviroment
  - * Path find (A* using algorithm)
  - * Decision making
  - * Planning

* THE GAME AI IS ALL ABOUT THE ILLUSION OF HUMAN BEHAVIOUR :)

* This needs various tech like Machine learning, Data structures & Animation

* Game types :
  1. strategy games
  2. Roleplaying games
  3. Action games
  4. Sports games
  5. Simulation
  6. Adventure
  7. Puzzle

OPTIMAL DECISION IN GAMES :-

* An optimal decision is a design that leads to atleast as good as known (or) expected outcome as all other available decision position.

* It is an important concept in decision-theory / Artificial intelligence because in order to compare the different decision outcomes.

Eg : Bubble Shooter where we have to decide where to hit the ball the decision must be optimal.

* To assign a utility value to each of them.

* Utility is nothing but an arbitary term for quantifying the desirability of a particular decision outcome.

(Simple words?
utility is a term used for disired decision outcome)

Mathematically:

Each decision (d) is the set of available options which will lead to an outcome (O)

$$O = f(d)$$

All possible outcomes from the set D.

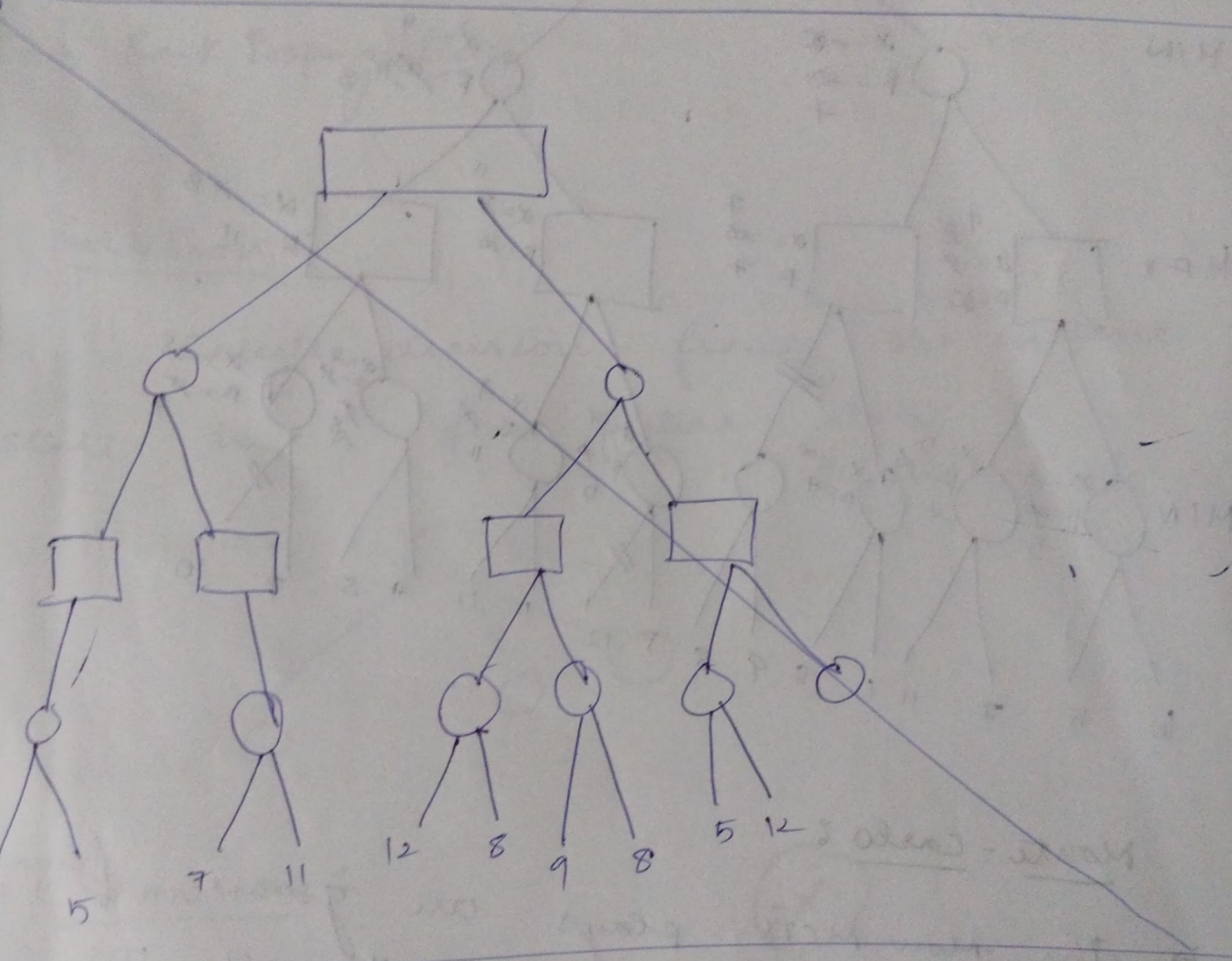
Assigning a utility $U(o)$ to every outcome,

$$U_D(d) = U_{=O}(f(d))$$

ALPHA - BETA PRUNING

Here $\alpha \rightarrow$ maximum value

$\beta \rightarrow$ minimum value



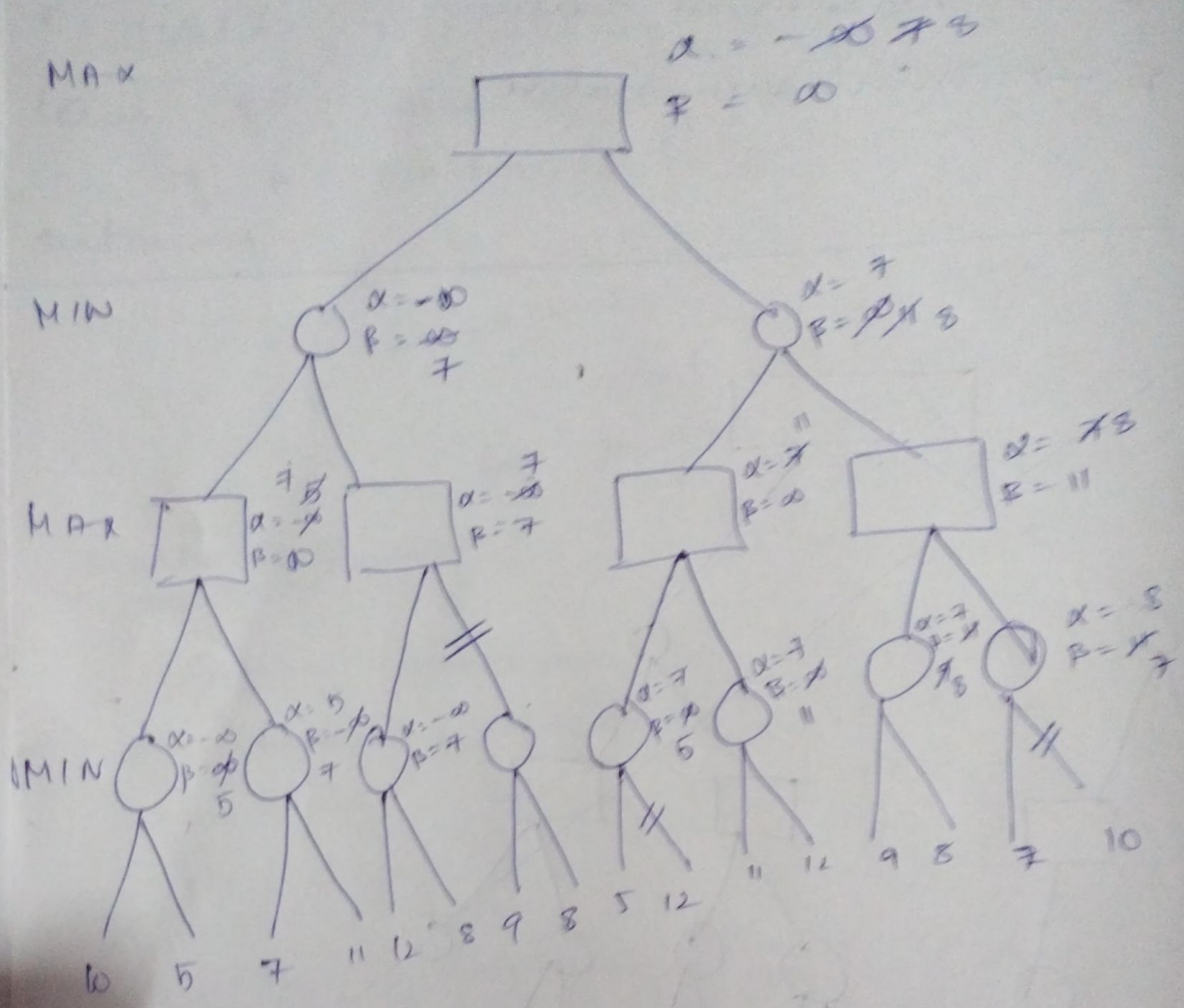
→ next page.

ALPHA BETA PRUNING

$\alpha \geq \beta$

MAX → α

MIN → β



Monte-carlo :

→ The MCTS plays the game from a given starting point all the way to the end by making random decisions.

→ It is better than min-max algorithm

There are four steps in MCTS

→ Selection

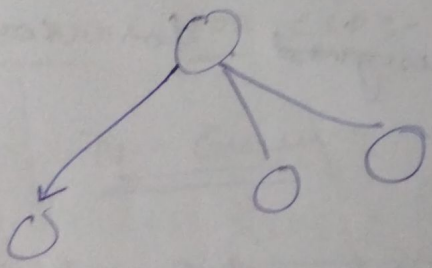
→ Expansion

→ SIMULATION

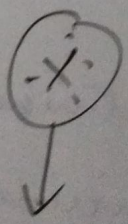
→ Back Propagation

Selection :

Selects decision from the current state to a ~~some~~ future state.



EXPANSION :



MCTS in

~~UNIT~~ - INTERNAL - 2

EXAM PREP NOTES

STOCHASTIC GAMES

* Many games are unpredictable in nature, such as (dice throw). These games are called stochastic games. The outcome of the game depends on skills as well as luc

* Winner is not only decided by the skill but also by luck.

STOCHASTIC SEARCH ALGORITHMS :-

→ These are designed for complex problems.

→ Desired properties of search metho are :

* Effectiveness
* Efficiency.

CSP) CONSTRAINT SATISFACTION PROBLEM :-

Satisfy the constraints)

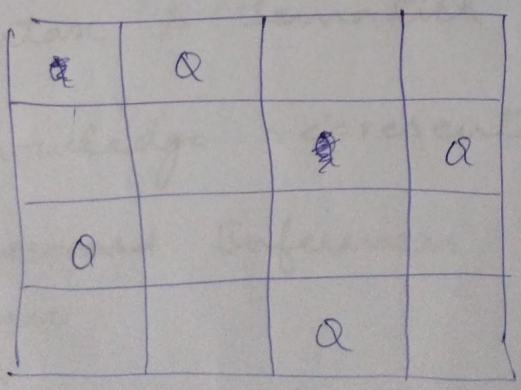
Range (0-9)

```
  SEND
+ MORE
------
 MONEY
```

```
  FORTY
+ TEN
+ TEN
------
 SIXTY
```

BACKTRACKING FOR CSP :-

N - Queen :



In this topic, Backtracking algorithm is used to solve CSP.

Eg :

LOCAL SEARCH FOR CSP :-

* It is an effective technique used to solve CSP.

* Here initially we assign value to every variable & as we move forward then values are changed.

* For Eg :

In N Queen, the initial state is random and might be changed as we move forward.

* Point of local search is to eliminate violated constraints.

UNIT-4 : (Logical Agents)

* Backward chaining
* Forward chaining
* Propositional logic.

} UNIT-3 & 4 PREP NOTES.

* Knowledge based agents
* Propositional theorem proving
* Propositional model checking
* Agents based on propositional first-order-logic
* Syntax & Semantics
* Knowledge representation & Eng
* ~~Forward~~ Inferences in first-order logic
* Resolution.

Knowledge based agents :-

* logic agent are agents with some representation of compeen knowledge about the world / Enviroment * uses inference to derive new information from the knowledge combined with new inputs.

* Knowledge base :-

set of sentences representing facts about the world.

* KNOWLEDGE - BASED AGENTS :-

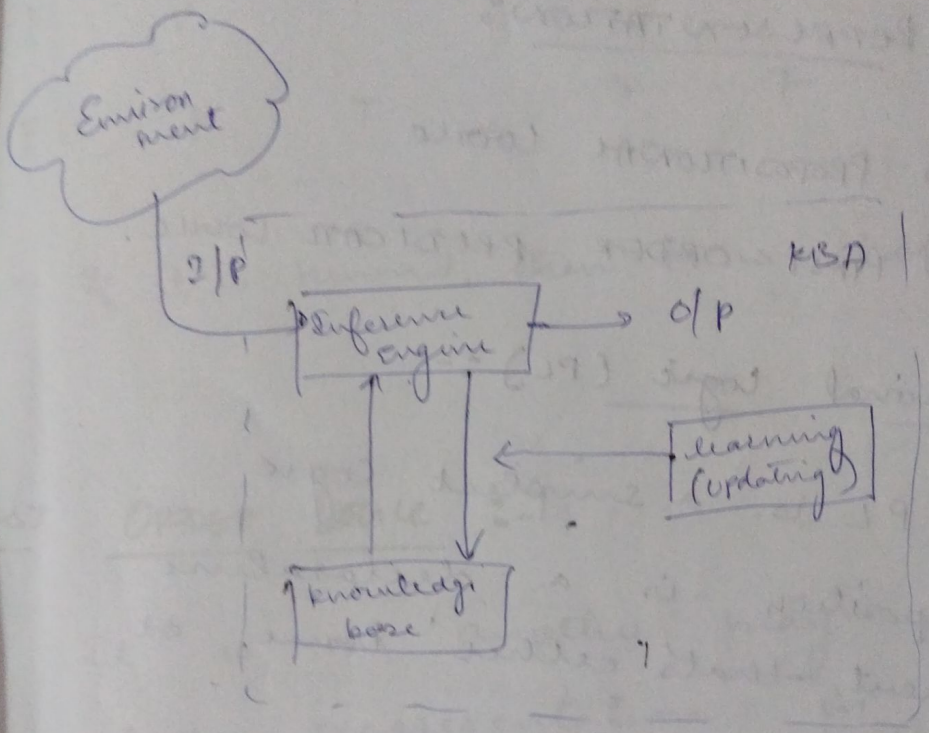
→ Intelligent agents need knowledge about the world to choose good actions / decisions

→ knowledge learning

A knowledge based agent is composed of :-

1. knowledge base : domain specific content

2. Inference mechanism : domain-independent algorithms



PROPOSITIONAL LOGIC :-

(Logical Representation)

→ It is a lang with some rules which deals with propositions

→ It consists of precisely defined syntax & semantics

→ Each sentence can be translated into logic using syntax & Sy

* Syntax : defines well formed sentence in lang

* Semantics : Defines the meaning of sentence in a world.

LOGICAL REPRESENTATION :

→ PROPOSITIONAL LOGIC

→ FIRST-ORDER PREDICATE LOGIC.

① Propositional logic (PL) :-

→ PL is a simplest logic

→ Proposition is a declarative statement that's either True or False

→ Connectives :

DRAW truth tables for these

Not	¬
And	∧
OR	∨
implies	→
If and only if	↔

Implication : If & only if :

P	Q	P → Q	P	Q	P ⟷ Q
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	T	F
F	F	T	F	F	T

Eg : If it is humid, then it is hot

FIRST ORDER LOGIC :-

→ It is an another way of knowledge representation in AI

→ It is an extension to PL.

→ FOL is also known as: Predicate logic.

→ It is powerful lang that develops info about the objects in more easy way & can also express the relationship b/w the objects.

→ PL ~~does~~ uses facts while FOL uses objects & relations.

① Objects : Colour, size.

② Relations : Sister of, brother of.

③ Function : Father of, leg of.

→ Two parts :

* Syntax
* Semantics

Syntax :

Constant : 1, 2, 3, A

Variables : x, y, z

Predicates : Boy, father

Func : Sqrt

Connectives : ¬, ⇒, ∧, ∨

Equality : =

Quantifiers : ∀, ∃

INFERENCE IN FOL :-

→ Used to deduce new facts sentences from existing sentences.

→ Basic terminologies in FOL :

* Substitution :

operation performed on terms & formulas.

* Equality :

FOL logic does not only use predicate & terms for making atomic sentences, but also uses another way, eq which is equality

Eg :

Brother (John) = Smith

UNIT - V :

KNOWLEDGE REPRESENTATION AND PLANNING

Ontological Engineering

Categories & Objects

Events

mental objects & modal logic

Reasoning system for categories

reasoning with default information

Classical planning

Algorithms for classical planning

heuristics for planning

hierarchical planning

non-deterministic domains

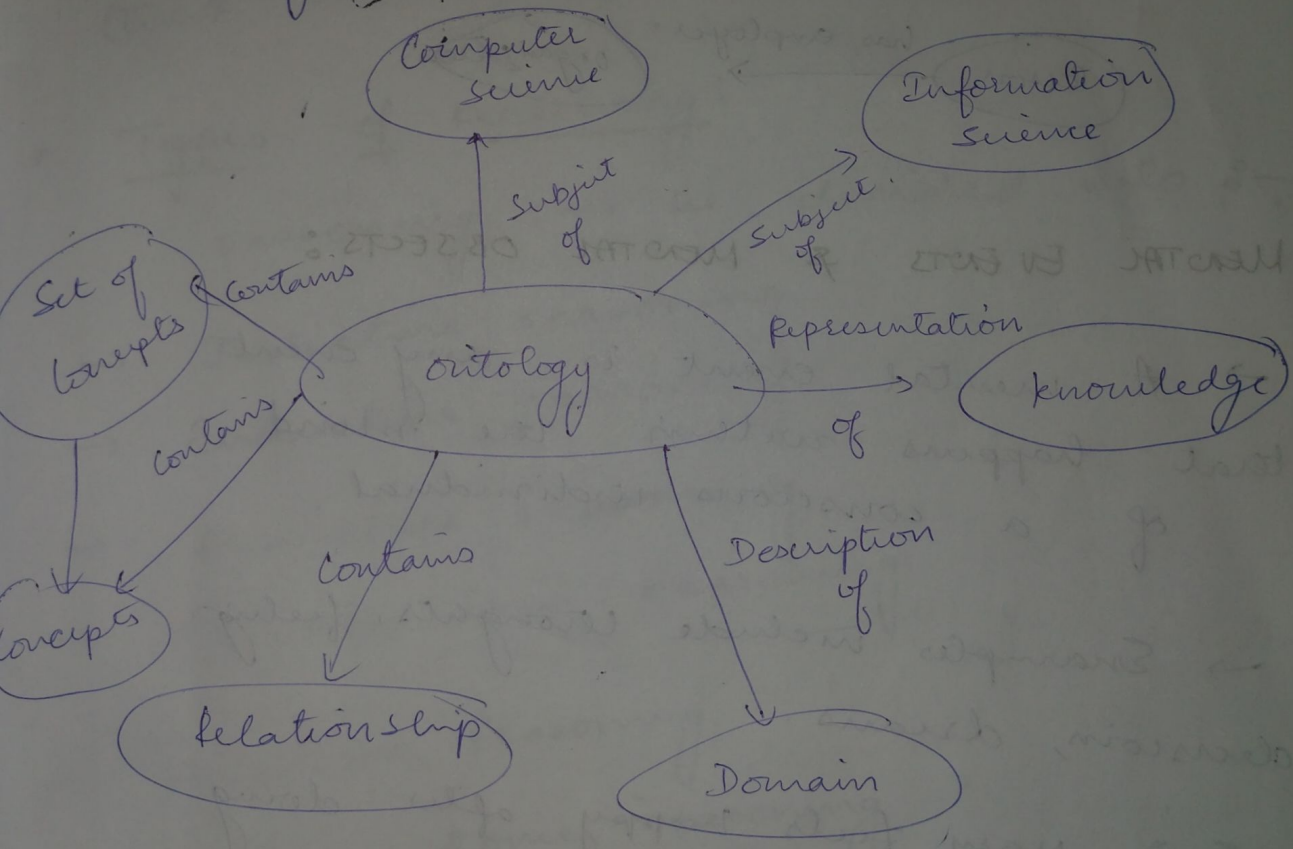
time schedule & resources

analysis.

Ontological Engineering :-

→ Ontology means "the state of being".

→ Ontology engineering is a subfield of knowledge engineering which studies methodologies of building ontology

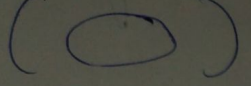


→ Ontology definition :-

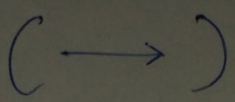
Represents knowledge as a set of concepts within a Domain & the relationship BW these concepts

Ontology is made up of two components :

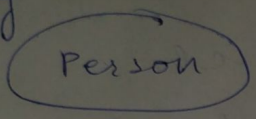
* Classes

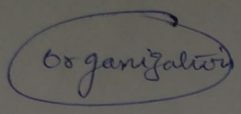


* Relationships

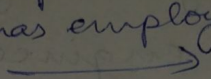


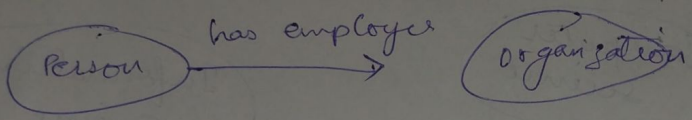
Eg :





Eg :





MENTAL EVENTS & MENTAL OBJECTS :

→ A mental event is any event that happens within the mind of a conscious individual.

→ Examples include thoughts, feelings decisions, dreams.

Eg : Mary feels happy after doing well on an exam & she smiles.

This thought is an mental event.

The smile is an physical event.

REASONING SYSTEMS FOR CATEGORIES :-

→ Reasoning is a mental process of deriving logical conclusion & making predictions from available knowledge, facts & beliefs.

→ In AI, the reasoning is essential so that the machine can think like a human brain

* Types of Reasoning :

reasoning can be divided into :-

→ Deductive reasoning
→ Inductive reasoning
→ Abductive reasoning
→ Common sense reasoning
→ Monotonic Reasoning
→ Non-monotonic reasoning

* Deductive reasoning :-

Premise 1 : All humans eats veggies

Premise 2 : Suresh is human

Conclusion : Suresh eats veggies.

* Inductive Reasoning :-

Premise : All pigeons we've seen in zoo are white

Conclusion : All pigeons are white.

* Abductive reasoning :-

Implication : Cricket ground is wet if it is raining

Axiom : Cricket ground is wet

Conclusion : it is raining.

* Common sense reasoning :-

If I put my hand in fire, then it will burn.

* Monotonic reasoning :-

Earth revolves around the sun.

* Non- monotonic reasoning :-

Birds can fly

penguins cannot fly

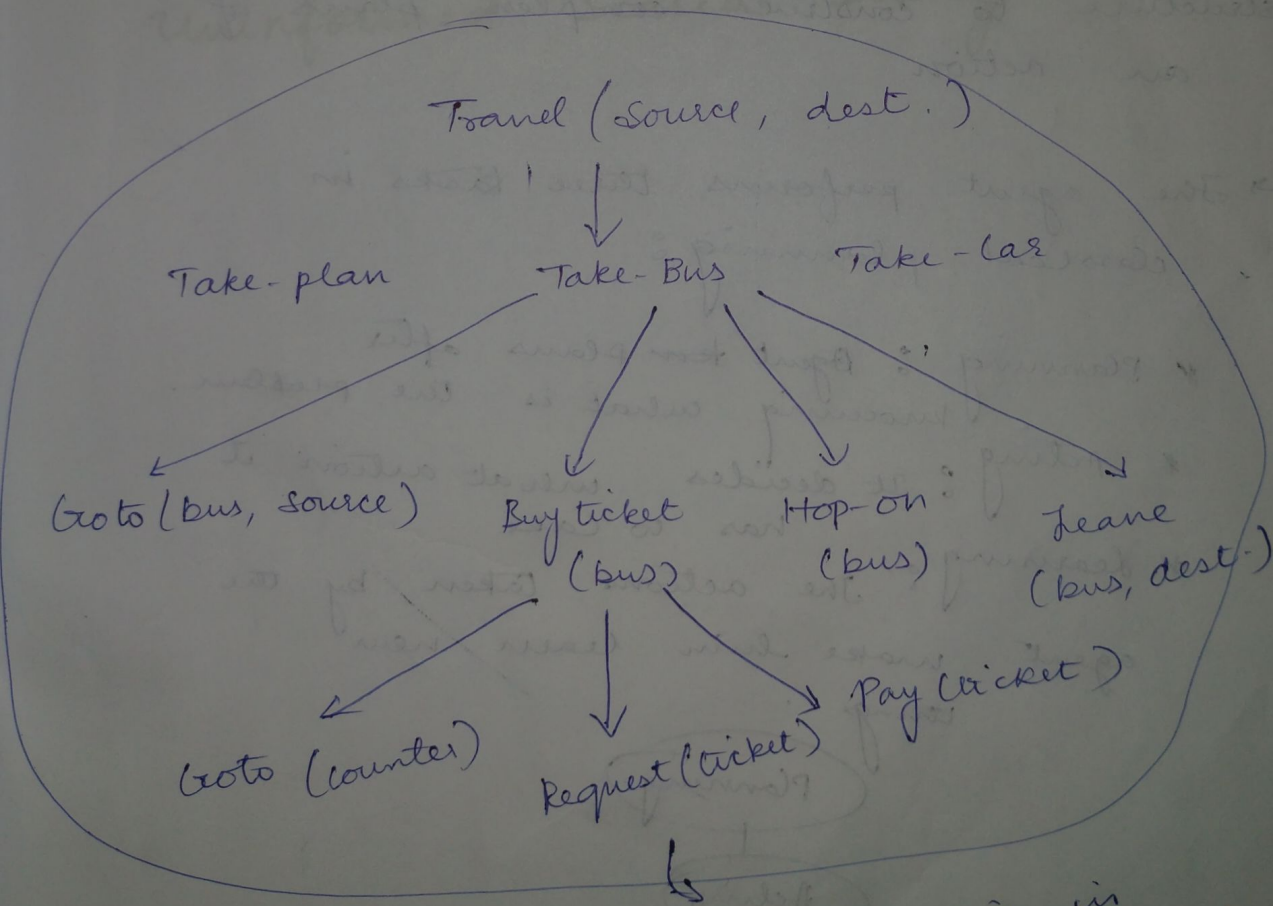
pitty is a bird so pitty can fly.

Hierarchical planning :-

* Hierarchical planning is also called as Hierarchical task network planning.

* This ~~problem~~ planning divides the problem into sub problem and is used to find the solution.

* In HTN planning, initial plan is mentioned as very high level description of what is to be done.



Explain this in words.

Advantages of Hierarchical planning :-

→ Can create the very large plans required by many real world applications

→ Easy to fix problems.

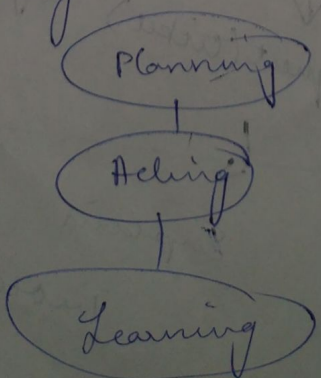
→ For complex problems, it is much more efficient.

Classical Planning :-

→ It is the planning where an agent takes advantage of the problem structure to construct complex plans of an action

→ The agent performs three tasks in classical planning :

* Planning : Agent plans after knowing what is the problem.
* Acting : It decides what action it has to take
* Learning : The actions taken by the agent make him learn new things.



UNIT-1

INTELLIGENT AGENTS

Introduction to AI

Agents & Environments ✓

Concept of Rationality ✓

Nature of Environments ✓

Structure of agents

Problem solving agents ✓

search Algorithms ✓

Uninformed Search Strategies

Agents Environment :-

→ An environment is everything in the world that surrounds the agent but it is not a part of an agent.

→ It can be described as a situation in which agent is present

* Features of Environment :-

An environment can have various features from the point of agent

* ① fully observable vs partially observable
② Static vs Dynamic
③ Discrete vs continuous
④ ~~Discrete~~ Deterministic vs stochastic
⑤ Single agent vs multi-agent
⑥ Episodic vs Sequential
⑦ known vs unknown
⑧ Accessible vs Inaccessible

① Fully Observable vs Partially observable :-

If an agent sensor can sense the compete state of environment at each point of time, it is fully observable, else it is partially observable.

② Deterministic vs stochastic :-

→ If an agents current state & selected action can completely determine the next state of environment. Then such environment is deterministic

→ A stochastic environment is random in nature & cannot be completely determined completely by an agent.

③ Episodic vs sequential :-

→ In episodic environment, there is a series of actions & only the current percept is required for the action

→ In sequential, agent requires memory of past actions to determine the next best actions.

④ Single vs multi agent :-

→ If only one agent is involved in an environment, it is single agent.

→ If multiple agent " "

⑤ static vs Dynamic :-

→ If an environment can change itself while an agent is working is called dynamic, else it is static.

⑥ Discrete vs Continuous :-

→ When the states of the environment is discrete i.e... a set of values, it is discrete else continuous.

Concept Of Rationality

* It is an objective to make an AI agent to work as per the desired actions then the agent is considered to be rational.

Rationality depends on :-

1. Performance measure
2. Agents Prior knowledge (Environment)
3. Actions that agents can perform (Actuators)
4. Agents percept sequence (Sensors)

PEAS
↓
Task environment

Agent → Environment (S) —percepts→ Actions

Actions → Changes Environment

Rational Agent :- For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given percept sequence & prior knowledge.

NATURE OF ENVIRONMENTS :-

* There are two types of Environment
  - → Natural
  - → Artificial

TYPES OF AGENTS :

Five types of agents :

- → Simple Reflex Agent
- → Model-based reflex agent
- → Goal-based agents
- → Utility-based agents
- → Learning agent

* Simple Reflex Agents :

→ The agent works only on the basis of current perception and it does not bother about the previous state in which the system was.

→ This type of agent is based upon the condition-action rule

If the condition is true
action is taken.
Else not

* Model-based reflex agent :-

→ It works by finding a rule whose condition matches the current situation.

model

→ It can handle partially observable environments.

* Goal-based agents :-

→ It focuses only on reaching the goal state hence the decision taken by the agent is based on how far it is currently from their goal or desired state.

→ Every action is intended to minimize their distance from the goal.

* Utility-based agents :-

→ It is similar to the goal-based agent but provides an extra component of utility measurement which makes them different.

→ It does not also the utility based agent is based not only on goals but also the best way to achieve the goal.

* Learning agent :-

→ It can learn from its past experiences as it has learning capabilities.

→ It starts to act with basic knowledge & then able to act & adapt through learning.

SEARCH ALGORITHMS

→ We can classify search algorithms into :

* Uninformed Search (Blind search)
&
* informed search (Heuristic search)

Search Algorithms

- Uninformed
  - BFS
  - DFS
  - Uniform cost search
  - Depth limited search
  - Iterative deepening depth first search
  - Bi directional Search
- Informed
  - Best first Search
  - A* search
  - AO* Algorithm
  - Problem reduction
  - Hill climbing

Uninformed Search : (Blind)

→ It does not contain any domain knowledge such as closeness, the location of the goal

→ It operates in a brute force way, as it only includes the info about how to travel through the leaves & how to identify the goal.

Informed Search :- (Heuristic search)

→ It uses domain knowledge, problem info is available which can guide the search.

→ Informed search can find the solution more efficiently than an uninformed search.

→ Also called as Heuristic search

BFS :-

→ Most common search strategy

→ This alg searches breadth wise in a tree or graph,

→ BFS alg starts searching from root node and expands all successor nodes.

→ Implemented using FIFO

Adv :

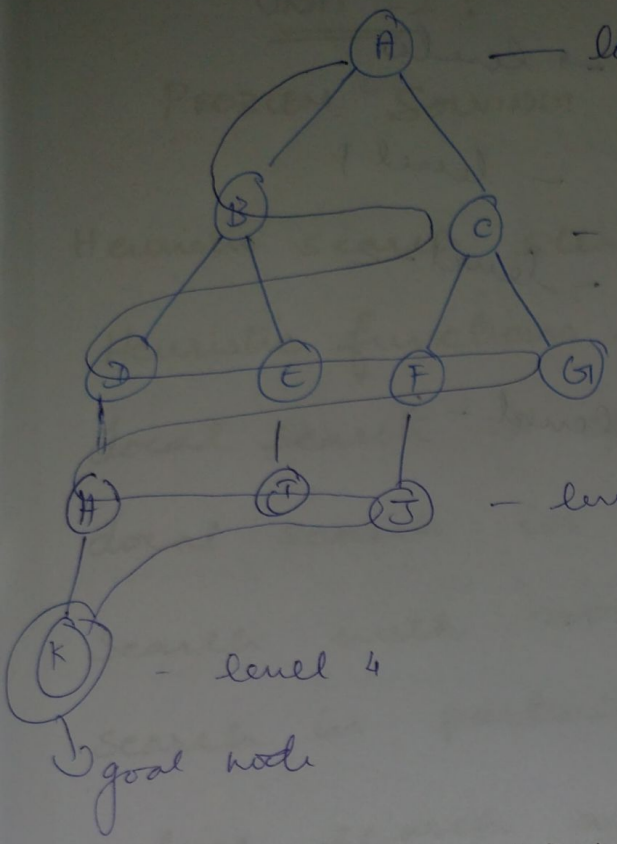
→ Provide a soln, if any soln exists.

→ If there are more than one soln, BFS will provide minimal soln which requires least steps.

Disadv :-

→ Requires a lot of memory

→ It needs a lots of time if the soln is far away.



Depth first search

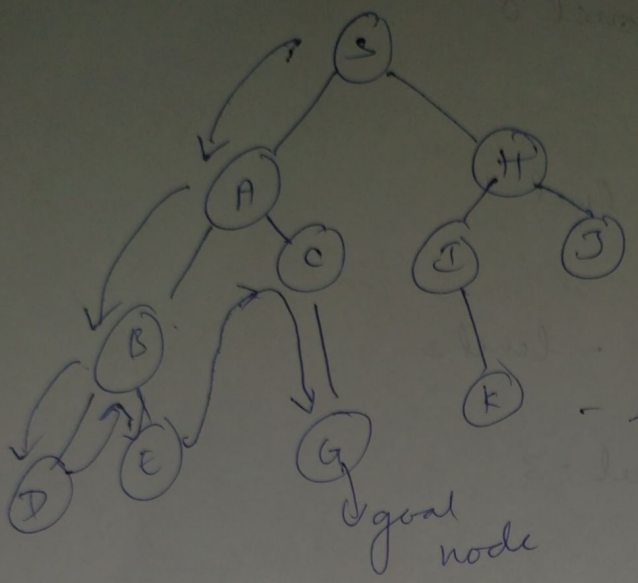
→ recursive alg for traversing a tree

→ It is called DFS because it starts from the root & follows each path to its greatest depth before moving to the next.

→ DFS ~~uses~~ similar to BFS.

Adv:
- → Requires very less memory
- → takes less time.

Disadv:
- → no guarantee of finding soln
- → It may go to infinite loop



→ level 0

- level 1

- level - 2

- level - 3

It terminates.

UNIT -2 :

PROBLEM SOLVING

Heuristic search strategies

Heuristic functions

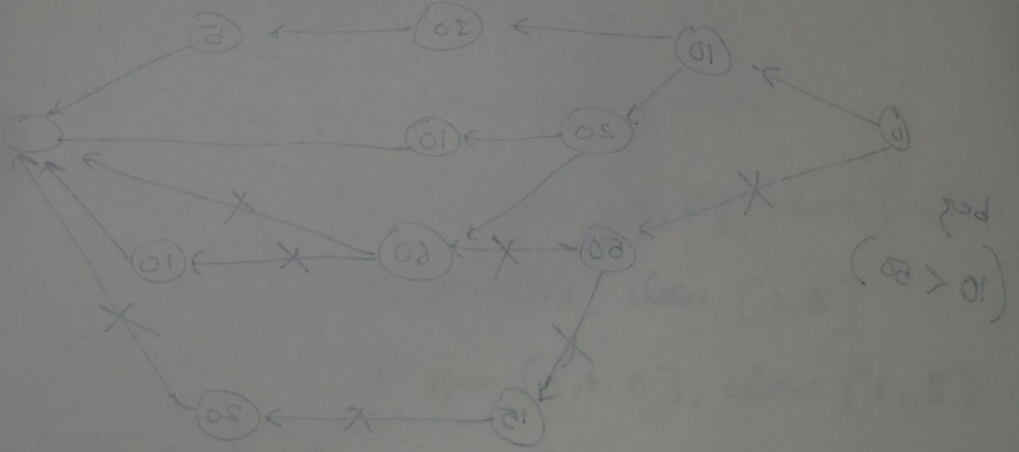
Local search & optimization problems

Local search in continuous space

Search with non-deterministic actions

Search in partially observable environments

Online search agents & unknown environments



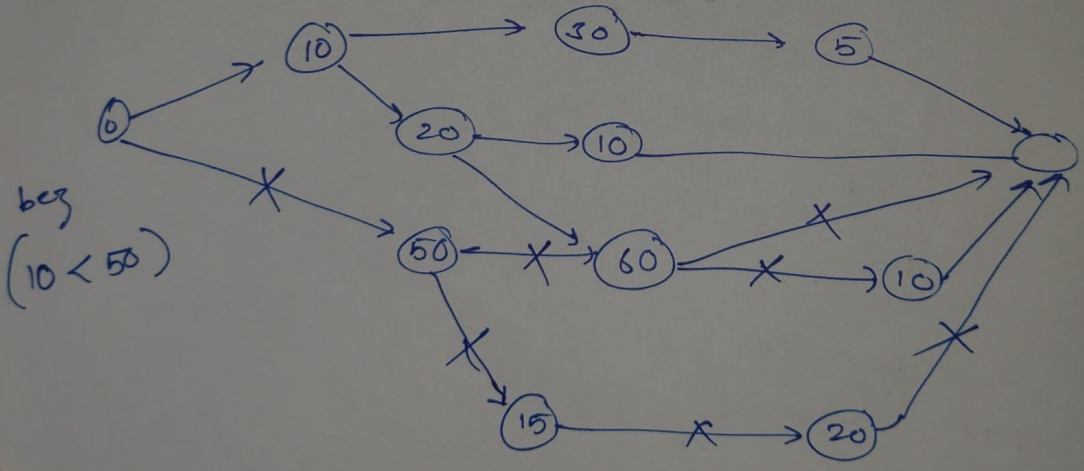
Heuristic Search techniques

→ Problem solving method that uses shortcuts / calculated guess to provide good enough solutions

→ Reduces time complexity to reach solution.

→ may not give best / optimal solution

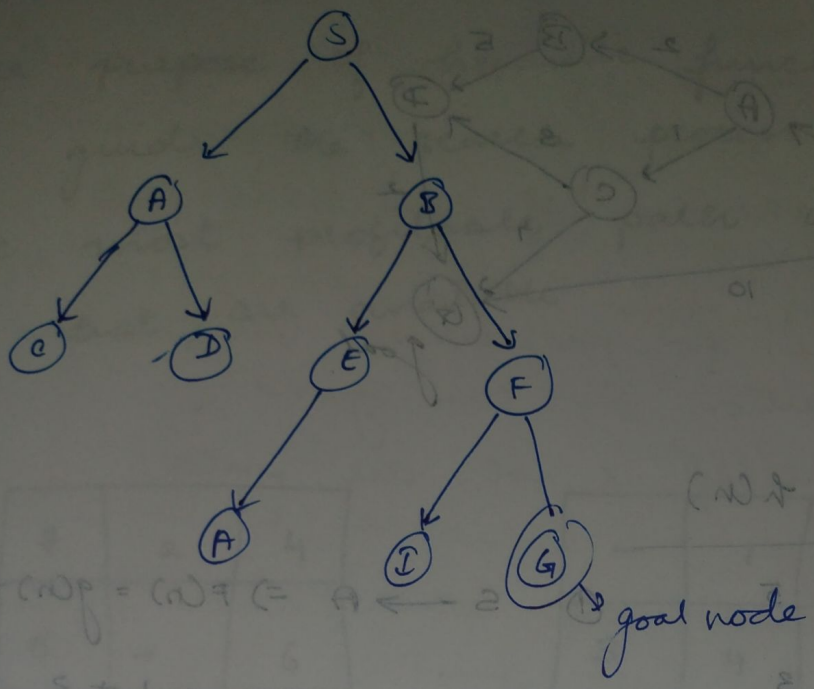
→ Also known as informed search



State - Space

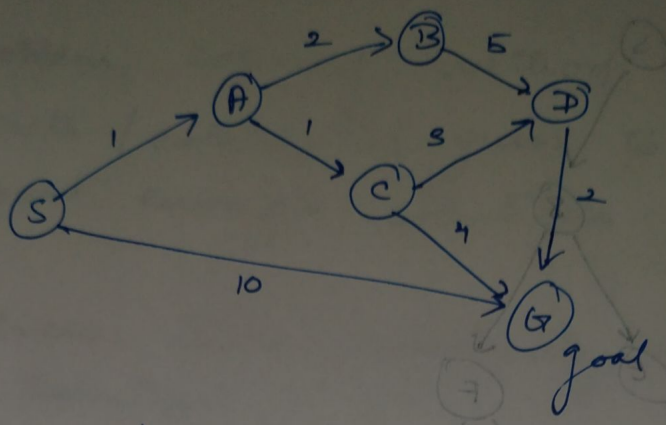
(Finds the shortest path by comparing the values)

Best first Search



node	h(n)	
S	13	=> initialization
A	12	open [A, B], close [S]
B	4	① open [A], close [S, B]
C	7	② open [E, F, A], close [S, B]
D	3	open [E, A], close [S, B, F]
E	8	③ open [E, A, I, G], close [S, B, F]
F	2	
H	4	
I	9	
G	0	

A* Search Algorithm



State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

① $S \rightarrow A \Rightarrow F(n) = g(n) + h(n)$

$= 1 + 3$

$= 4$ ✓

$S \rightarrow G \Rightarrow F(n) = 10 + 0$

$= 10$ (hold)

② $S \rightarrow A \rightarrow B \Rightarrow F(n) = 3 + 4 =$ (hold)

$S \rightarrow A \rightarrow C \Rightarrow F(n) = 2 + 2 = 4$ ✓

③ $S \rightarrow A \rightarrow C \rightarrow D \Rightarrow F(n) = 5 + 6$ (hold)

$S \rightarrow A \rightarrow C \rightarrow G \Rightarrow F(n) = 6 + 0$

$= 6$ ✓

Heuristic Search & function

The purpose of heuristic function is to guide the search process in the most profitable path among all that are available.

7	2	4
5	-	6
8	3	1

Start state

-	1	2
3	4	5
6	7	8

Goal state

$h_1 = 8$ → Total no. of nodes or moves

$h_2 = 3+1+2+2+2+3+3+2$ → (moves taken to put the no. in its desired position)

$h_2 = 18$

Optimised solution

$$\boxed{h_1 + h_2 = 26}$$

Local Search Algorithm :-

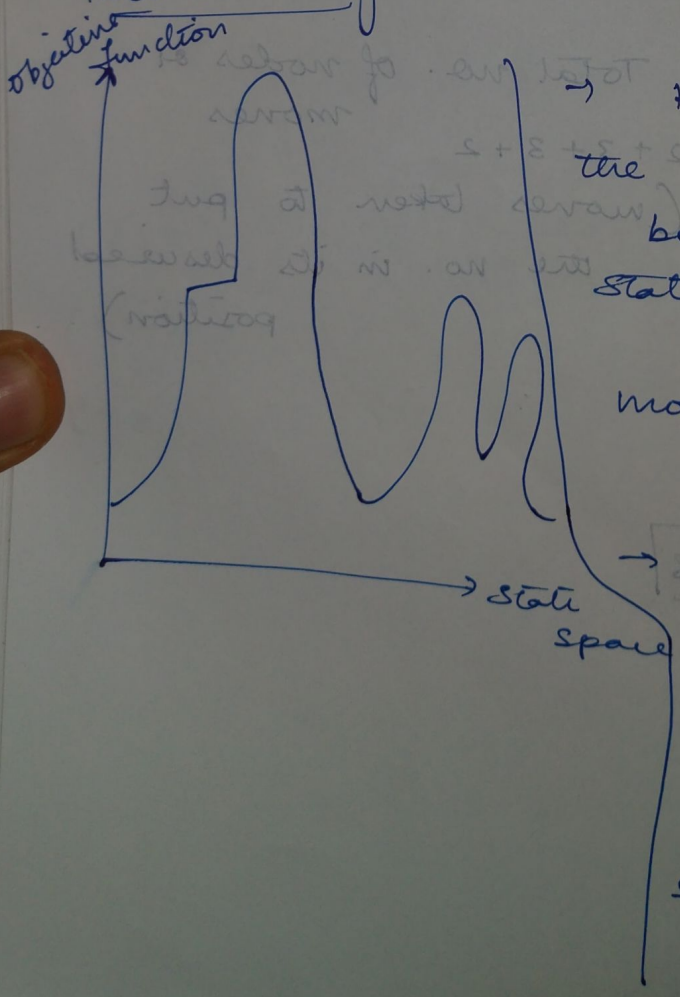
→ They operate using a single current node and generally move on to neighbours of that node.

→ Suitable for problems where the solution is the goal state itself & not the path.

→ Used for solving optimization problems

Eg: Hill climbing & ~~annea~~ simulated Annealing

Hill climbing :



→ Here we check if the neighbour state is better than current state or not, if it is better we move to the neighbour state

→ It is also called greedy local search because it grabs a good neighbor state without thinking ahead abt where to go next.

Simulated Annealing :-

* Annealing is the process used to temper or harden metals & glass by heating them to a higher temp & then gradually cooling them, thus allowing the material to reach low energy (Eg: making metal objects)

-

:→ Simulated Annealing is quite similar to hill-climbing, instead of picking the best move, it picks a random move, If the move improves the sol situation It is always accepted. Otherwise the algorithm accepts the move with some probab less than 1.