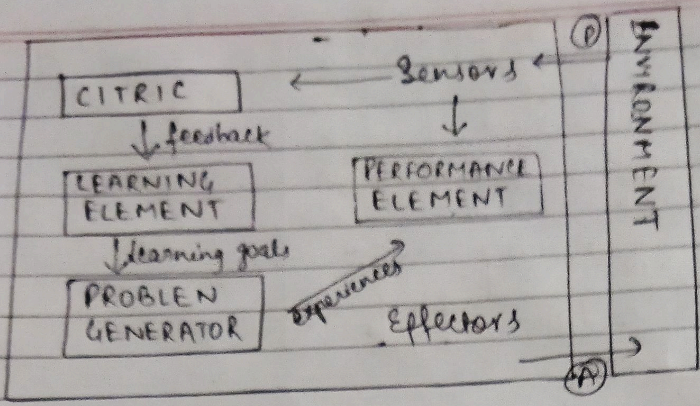


15/12/24.

PEAS: Grouping of AI Agents

→ used to group similar type of agents together.

→ PEAS → Environment - All the surrounding things & conditions.
→ Sensors - Devices through which agent perceives observ. from env.
→ Actuators → Devices, H/w, S/w, through which agent performs action on env.
→ Performance Measure

→ It is the output we get from agent.
→ or results obtained after agent processing.

SELF DRIVING CAR

P → Comfort, safety, Time, Legal Driving.
E → condition of Roads, crossings, Traffic signals.
A → ~~Actuator~~ Sterring, Brakes, Horn, Accelerator.
S → Sensors, camera, GPS, Speedometer, Odometer.

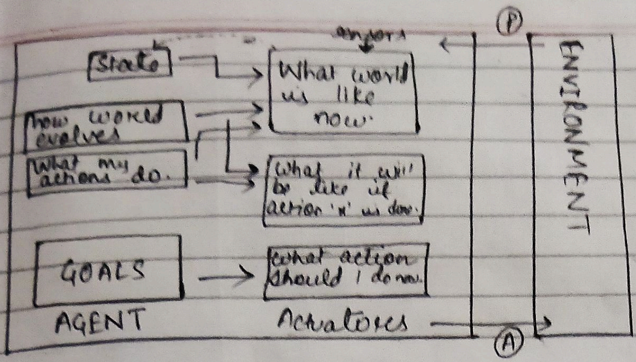
Classification of Environment in AI:-

Environment is part of the universe that surrounds intelligent system.

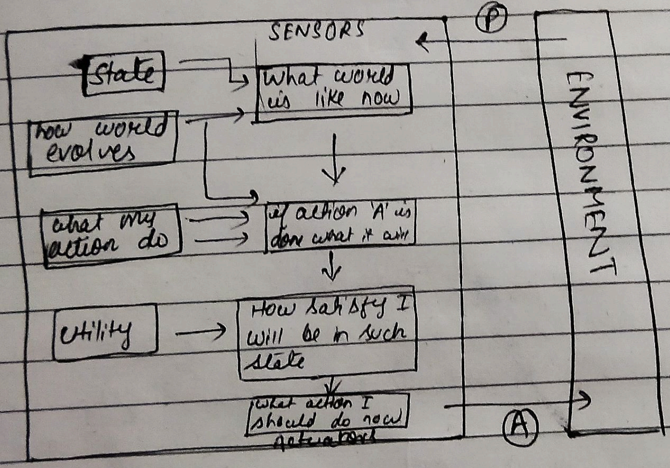
1. Accessible and Inaccessible :-
→ Agent can ~~be~~ obtain complete & accurate info. about state's env.
→ Whole part of env. may not be in reach. (inaccessible)
→ Room with temp. as state (accessible)
→ event on earth. (inaccessible)

2. Deterministic & non-deterministic :-
→ Agent's current state & selected action can completely determine env. next state.
→ Agent doesn't have to worry about uncertainty.

Non-deterministic :- nature of env. can't be decided by agent alone. [Stochastic].
- Random in Nature.

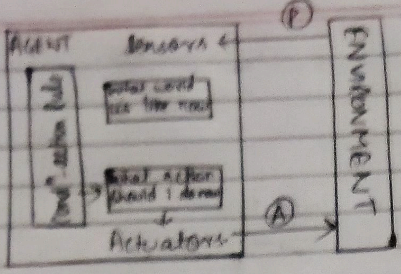


④ Utility-based Agents:-
→ Agents are more concerned about the utility (preference) for each state.
→ Act based not only on goals but also the best way to achieve goal.
→ useful when there are multiple possible alternatives and agent has to choose in order to perform best action.



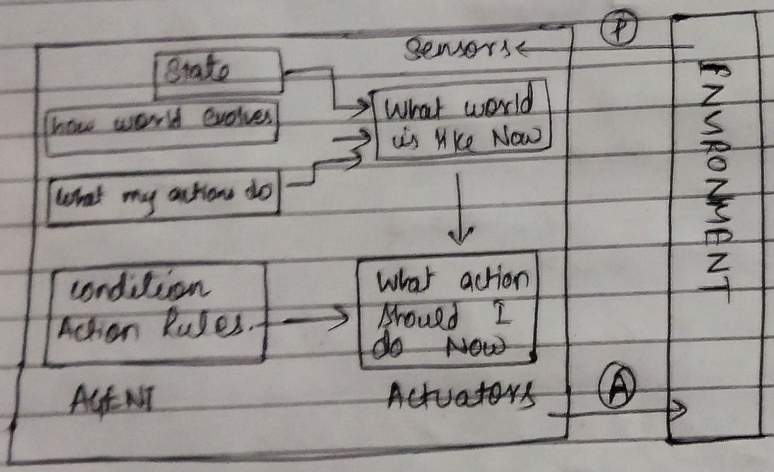
⑤ Learning Agents:-
→ can learn from its past experiences.
→ Starts to act with basic knowledge & then able to act by adapting learning.

Components:-
i) Learning element → makes improvement in system by learning from env.
ii) Critic → gives feedback about agent's performance based on standard.
iii) Performance element → selects the action to perform.
iv) Problem generators → suggest the action.] new info. gain.



② Model-based Reflex Agents

→ works by finding the rule whose cond^n watches current situation.
→ can work in partially observable env. and track situation.
→ Agent keeps track of internal state which is adjusted by each percept and that depends on percept history.
→ Model :- How things happen in world.
→ Agent state updation req. info. :- → ① how world is evolving.
→ ② how agents action affect the world.



③ Goal Based Agents :-

→ Focuses only on reaching the goal set.
→ Agent takes decision based on how far it is currently from the goal state.
→ Every action is taken to minimize distance to goal state.
→ more flexible agent.

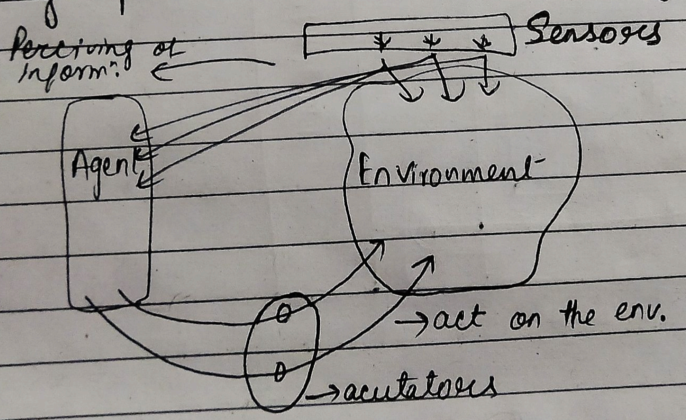
→ (i) Borrowing ideas from pshychology & ~~human~~ neuroscience.
(ii) Forgetting things, Genetics studies, Language understanding

4) Super AI → hypothetical concept.
M/C will be developed in such a way that is much more advanced than human.
machine ⟷ machine (communication.)

$\boxed{\text{Types of AI Agents}}$:-
AI is defined as study of Rational Agents.
Agents → responsible for any work output obtained from system.
→ Person
→ Firm
→ Machine / S/W
} can make decisions.

AI System is composed of
→ Agent
→ Environment.

Agent is anything like :-
i) perciving its env. through <u>sensors</u>.
ii) Acting upon that environment through <u>actuators</u>.



① <u>Simple Reflex Agents</u> :- Works only on situation/perception and ignores the history of previous state.
↳ $\boxed{\text{Condition-Action}}$ Rule. {if cond. is true take some action}

<u>Limitations</u>:
i) Very limited intelligence.
ii) No knowledge about non-perceptual parts of state.
iii) Can go to infinite loop.

(5) Expert Systems -

- designed to simulate the decision-making ability of a human expert in a specific domain.
- uses stored knowledge & logical rules to solve problems or give advice, just like a human expert would.

Representing & using domain knowledge.

domain knowledge ↓ info. & expertise req. to solve specific problems within a domain (eg. medical diagnosis, engineering).

Expert system typically represent domain knowledge using :-
- → knowledge bases
- → Inference engines.

Knowledge bases

stores all facts & rules about a specific domain (eg. medical diag.)

eg

Inference engine

"brain" of the expert system that applies the rules in knowledge base to given data / problem.
- uses logical reasoning to draw conclusions.

ex. if patient has fever & cough the IE may conclude they could have the flu.

Expert system shells

- Pre-built s/w framework that includes a ready-to-use inference engine.
- users can add their own knowledge base to create a custom expert system.
- these shells allow developers to focus to domain-specific knowledge rather than building ~~the~~ underlying str. from scratch.

Knowledge Acquisition :-

- gathering knowledge from experts / other sources & encoding it into a usable format for the expert system.
- can be time-consuming & challenging, as it requires experts to explain their reasoning in structed way that the system can understand.

Learning Decision tree :-

→ model that uses a tree-like str. to represent decisions and their possible consequences.

→ Tree is built by recursively splitting the data based on features that provide the best classific^n.

→ Goal is to create a model that can classify new data based on learned patterns.

Explaination-based learning :- (EBL)

→ improving performance by understanding the reasoning behind decisions.

→ learn from examples by explaining why a particular decision / classification was made.

Learning using relevance information :-

- focuses on identifying & using relevant features / attributes to improve learning efficiency.

- feature selection technique are used to identify which attributes are most imp. for making decisions, thus reducing the complexity of the model.

Neural net learning :-

Neural networks learn by adjusting weights b/w nodes (neurons) in a network based on I/O pairs.

Deep Learning :- a subset of neural networks, involves multiple layers of neurons & has gained popularity in complex tasks such as image recognition & natural language processing.

Genetic learning :-

- Genetic algo. mimic the process of natural evolution to search for optimal sol^n in a problem space.
- A pop. of candidates sol^n is evolved over generations using operations like selection, crossover & mutation.
- used for optimization problem. (search space is large & complex).

(3) Expert System
- designed to si
exper in a sp
- uses stored
or give advic

Representing

Expert syste
using :-

Knowledge b
stores all fa
specific doma

Expert s
- Pre-built
inference e
- users can
custom
- these shel
specific
str. fro

Knowledge
- gatherin
it into
- can be
experts
that

Example: dialogue → "Can you open the window?"
Speech Act → Literal meaning. A question about ability.
Illocutionary meaning. A polite req. to open the window.

2) Learning.

→ Learning in AI → ability of m/c to improve their performa. over time based on experience.

→ Various forms of learning in AI, especially in supervised & unsupervised settings.

Forms of learning :-

Supervised learning → model is trained on labelled data (I/O pairs), & the goal is to learn a mapping from inputs to outputs.

ex:- Categorizing email as "spam" or "not spam".
Input :- Email text → Output :- Spam/Not spam.

ex:- Predicting house prices.
Input :- Sq. footage, location → Output :- predicted price.

Unsupervised learning → model tries to find patterns/str. in unlabelled data.

ex:- Clustering :- Grouping customers based on shopping behavior (task)
ex - Input → Customer purchase history
Output → Customer Clusters (eg. "bargain hunters," "frequent shoppers").

Reinforcement Learning → Agent learns by interacting with the env. and receiving feedback (rewards/penalties) based on its actions.

ex:- Teaching AI to play chess or GO.
Input :- Current game state → Output :- Best move (based on reward)

Inductive learning :-

→ Generalizing from specific examples to broader rules.
→ Often used in decision trees & other m/c learning models
→ system infers general principles from specific obs.

Discourse Processing -

→ focuses on understanding languages at the level of paragraphs or entire conversations, taking into account the relationships b/w sentences.

→ involves tasks like :- co-reference resolution
determining which word refers to same entity.

anaphore resolution → deciding what a pronoun refers to.

ex:- "Sarah loves her dog."
"She takes it for a walk everyevening".
conference links :- "Sarah → She", "dog → it".

ex:- "The teacher called the student".
"She gave him homework".
Anaphora resolution identifies "she → the teacher", him → the stud."

Pragmatic processing -

→ deals with interpretation of lang. in context.
→ concerned with how real-world knowledge & the speaker's intentions influence the meaning of a sentence.
→ include (i) handling implicatures
⇒ what is implied but not explicitly stated.

ex:- dialogue →
Speaker 1:- "Did you finish the assignment?"
Speaker 2:- "I worked on it until midnight".
Implicature →
Speaker 2 implies that may not have completed the assignment but worked hard on it.

(ii) presupposition (assumptions that a sentence implies as true for the statement to make sense.)
Sentence :- "John stopped smoking"
Presupposition → "John used to smoke.

(iii) Speech Acts (actions performed through lang., categorized by their intent (eg. request, commands, promising).

Example: dialogue
Speech A

(2) Learning
→ Learning in A
over time ba
→ Various form
& unsupervis

forms of lear
Supervised l
(I/O pairs),
inputs to
ex:- categor

ex:- Predic

Unsupervise
unlabelled
ex:- Cluster
ex -

Reinforce
env. a

ex:-

Inductive
→ Genera
→ Often
→ system

Parse Tree:- (syntactic tree), rep. syntactic str. of a sentence based on a given grammar.
Each node in the tree corresponds to a construct in the grammar, such as noun phrase (NP) or verb phrase (VP)

ex:- the cat sleeps:-

Det N (determiner followed by Noun)

```
          S
       /     \
     NP       VP
    /  \       |
  Det   N      V
  The  cat   sleeps.
```

Context-free Grammars (CFG)
→ Type of grammar used to define syntactic rules for language.
→ consists of a set of produc^n rules
→ Specify how non-terminal symbols can be expanded into terminal &/or non-terminal symbols.

Components:- Terminals → actual word/symbol (eg. "cat", "sleeps")
Non Terminals → abstract symbols (eg. S, NP, VP).
Production rules → how a non-terminal can be expanded.
S → NP VP
Start symbol → initial non-terminal (eg. S for Sentence)

Semantic Analysis -

→ Deals with the meaning of words & sentences.
→ It goes beyond the syntax to understand the meaning conveyed by a sentence.
→ It includes - determining the correct meaning of a word based on context (Word Sense Disambiguation).
- identifying entities like names, dates, locations (Named Entity Recognition, NER)
- identifying roles played by various words in a sentence eg. subject, object (Semantic Role Labelling (SRL)).
- mapping a sentence to a formal represent^n (eg. logical form or frame).

25/12/24

Module (5)

1) Natural Language Processing.

→ Involves enabling m/c to understand, interpret, and generate human language.

→ It combines linguistics & m/c learning techniques to process & analyze large amounts of natural lang. data.

Introduction:-

→ NLP is concerned with interacⁿ b/w computers & human language, including tasks like speech recognition, text analysis, m/c translation and lang. generation.

→ It can be divided into several subfields:- Syntactic processing, semantic analysis, discourse processing and pragmatic processing.

Syntactic processing

•) Syntax refers the ~~the~~ str. of sentence, the rules that govern sentence formation.

•) It involves - parsing sentences to understand their structure.
↓ resolve sentence into its component part.

This is done by identifying constituents (grp. of words that function as a single unit) and dependencies (relationships b/w words).

•) Tools like Part-of-Speech (POS) Tagging, Parse Trees, and Context-Free Grammers (CFG) are used for syntactic analy.

POS Tagging → assigning a part of speech (eg. noun, verb, adjective) to each word in a given sentence based on its definition & context.

→ for understanding grammatical str.

ex:- POS type:- Noun (NN) : cat, table
Verbs (VB) : run, eat
Adjectives (JJ) : happy, large
Adverbs (RB) : quickly, silently.

ex:- Self-Driving Cars.

Problem → controlling the speed of a car based on dist. from another car.

Input → dist. to car ahead (d): close, medium, far.
Current speed (S): slow, medium, fast.

Output → car acceleration: slow down, maintain speed, speed up.

Fuzzy rules → if d is close & speed is fast, then slow down.
if d is medium & speed is medium, then maintain speed.
if d is far, then speed up.

(based on fuzzy reasoning, car adjusts speed to maintain a safe dist.)

Other ex:- 1) Medical diagnosis system.
2) Home Appliances → AI-powered a.c. adjust cooling levels based on room temp. & humidity
3) Robotics → robots make decisions about movt. & obstacle avoidance in uncertain env.

~~Dempst~~ DST combines the info. to calculate a new set of beliefs.

Calculations: • Belief for flu:- Bel({Flu}) = 0.7 (direct belief)
• Plausibility " " :- Pl({Flu}) = 0.7 + 0.1 (belief + uncertain)
= 0.8.

ex:- Self-Driving
Problem → controlling
Input → dist. to current
Output → car
Fuzzy rules -
(based on safe dist
Other ex.

Fuzzy sets & Fuzzy Logic

Fuzzy sets → In traditional logic, everything is black or white
something is either T or F, 1 or 0.
But in real life, many situations are gray areas, for ex -
is 50°C water hot? (maybe it's somewhat hot).

Fuzzy logic & fuzzy sets help handle this kind of reasoning.
They allow us to say 50°C water is 50% hot &
90°C water is 100% hot.

Fuzzy set → a group where elements can partially belong to the set.
each element has a degree of membership, which is a number b/w 0 and 1.

ex:- "Hot water" → 50°C → 0.5 (partially hot)
70°C → 0.8 (mostly hot)
90°C → 1.0 (completely hot)
30°C → 0.0 (not hot at all)

mimics how human think

Fuzzy Logic → reasoning method that uses fuzzy sets to make decisions / draw conclusions.
→ deals with the idea of "how much" something is true, rather than "T or F".

Examples:-
Problem:- Turning on a fan. (You want to adjust its speed based on room Temp.)

1. Fuzzy set for Temp. • cold (0°C) = 1.0, (20°C) = 0.5, (30°C) = 0.0
complete cold — not cold at all.
• warm (20°C) = 0.5, 30°C = 1.0
• Hot (30°C) = 0.0, 40°C = 1.0

Fuzzy logic in AI. Why?
→ ① Human-like decision making - handles uncertainty.
② flexibility - work with imprecise data
③ ~~-~~

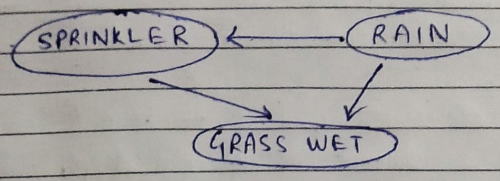
Semantics of Bayesian networks.

- Represents probabilistic relationships b/w variables.
- A network consists of nodes and directed edges.

  nodes → random variables; directed edges → dependencies b/w variables.
- Each node has a conditional probability distribution that quantifies the uncertainty about the variable, given its parents.

ex:- If Rain & Sprinkler are nodes, and WetGrass is a dependent node, the network might look like this:-

- P(WetGrass | Rain, Sprinker): The probability that the grass is wet given whether it rained | sprinkler was on.



Dempster - Shafer Theory (Evidence Theory)

- provides a framework for reasoning & uncertainty, particular in sol^n. where probabilities are difficult to quantify.

- It uses belief func^n. to represent the degree of belief in a hypothesis and combines evidences from multiple sources to update the belief.

- can handle situations where there is ambiguity | conflicting evidence

ex:- Medical Diagnosis -

Situation - A doctor wants to diagnose a patient's condition.

Possible illness:- • Θ = {Flu, cold}

Θ → set of all possible outcomes (Frame of Discernment)

Beliefs:- based on symptoms
- 70% belief the patient has flu
- 20% belief they have a cold
- 10% unsure about both.

BPA (Basic prob. Assignment):- $m(\{Flu\}) = 0.7$

$m(\{cold\}) = 0.2$

$m(\{Flu, cold\}) = 0.1$

If it rains (R), then ground will be wet (W). (R → W)
It is Raining (R)
PT Ground is wet (W).

Sol^n. 1. R → W (Premise 1)
2. R (Premise 2)
3. From R and R → W, we conclude W (ground is wet)
Conclusion:- The ground is wet (W).

② **Probabilistic Reasoning**.
Allows AI system to make decisions / predictions in situations where the knowledge is uncertain / incomplete.

Representing knowledge in an uncertain domain:-
In real life, we often deal with uncertainty
- situations where we can't be 100% sure about something.
In AI, we need ways to represent & reason about this uncertain knowledge to make decisions / predictions.

It is imp. bcz.
→ some info. may be missing / incomplete (eg. weather forecasts).
→ events might have probabilities instead of certainties (eg. chance of Rain)
→ decisions often need to be made based on limited knowledge (eg. medical diagnosis)

Ways to represent uncertainty
① **Probability** → using no. b/w 0 & 1 to represent how likely something is.
ex- Probability of rain tomorrow is 70%. → P(Rain) = 0.7.

② **Bayesian networks** → A graph that shows how events are connected and how likely they are based on each other.
ex - if it rains, there's high chance the road will be wet".

③ **Fuzzy logic** → represents degree of truth, not just "true" or "false".
ex:- "Weather is hot" can be partly true (eg. 0.8 on a scale of 0 to 1).

④ **Rules with confidence scores** →
Rules that include a level of certainty.
Ex - if a patient has a cough, there's 80% chance they have a cold."

④ complex ex. with Quantifiers

Premises:- ① $\forall x(\neg Human(x) \vee Mortal(x))$ (All humans are mortal)

② $Human(Socrates)$ (Socrates is a human.)

Resolution:

- convert to clausal form:-
  - C1 :- $\neg Human(x) \vee Mortal(x)$
  - C2 :- $Human(Socrates)$
- Substitute x = Socrates in C1.
  - $\neg Human(Socrates) \vee Mortal(Socrates)$

Resolve with C2

- Mortal (Socrates)

conclusion:- Socrates is Mortal.

Natural Deduction-

- way to solve logic prblms. step by step.
- using rules that resemble how we naturally think & reason.
- It allows you to start with given statements (premises) & use logic reasoning to reach a ~~desthat~~ conclusion.

Inference Rules in Natural Deduction-

① Introduction Rules → used to introduce logical operators:-

- conjunction intro. ($\wedge$ - intro.) :- from A and B, infer $A \wedge B$.
- Disjunction " ($\vee$ - intro.) :- from A, infer $A \vee B$ (or $B \vee A$)
- Implication " ($\rightarrow$ - intro.) :- from A, if B can be derived, infer $A \rightarrow B$.
- Negation " ($\neg$ - intro.) :- If assuming A leads to contradicⁿ, infer $\neg A$

② Elimination Rules → used to remove logical operators.

$\wedge$ - Eliminⁿ. :- ~~From~~ $A \wedge B$, infer A or B.

$\vee$ - " :- from $A \vee B$, infer C if C follows both A & B.

$\rightarrow$ - " (Modus Ponens):- from $A \rightarrow B$ and A, infer B.

③ Double Negation Elimination:-

- From $\neg\neg A$, infer A.

④ Reduction ad Absurdum:-

if assuming $\neg A$ leads to a contradiction, infer A.

Resolution →

- rule of inference used in logic to derive conclusions from a set of premises.
- complete method of deductive reasoning in 1st order logic.
- resol? principle works by finding complementary literals in clauses & combining them into a new clause.

① Ex. in propositional logic

Premises :- 1) $A \vee B$ (Either A or B is true)

2) $\neg A$ (Not A, A is false)

Resolution:- combine the premises:-

- from $A \vee B$ and $\neg A$, resolve A (∵ $\neg A$ makes A false).

The result is B (only B can be true).

conclusion:- B is True.

② Ex. in Predicate Logic

Premises:- 1) $\forall x (P(x) \vee Q(x))$ (for all x, either P(x) or Q(x) is true)

2) $\neg P(a)$ (for a specific a, P(a) is false).

Resolution:-

Substitute $x = a$ in the 1st premise

- $P(a) \vee Q(a)$

combine with 2nd premise ($\neg P(a)$):-

from $P(a) \vee Q(a)$ and $\neg P(a)$, resolve P(a).

conclusion:- Q(a) is true

③ Ex. using clausal form:-

Premises:- ① ¬Likes (John, X) ∨ Likes (Mary, X) (if John doesn't like X, Mary likes X.)

② Likes (John, Ice cream) (John likes ice cream)

Resolution:-

convert both premises in clausal form.

- Clause ① :- ¬Likes (John, X) ∨ Likes (Mary, X)
- " ② :- Likes (John, Ice Creame)

Substitute X = Ice-cream into C1.

¬Likes (John, Ice cream) ∨ Likes (Mary, Ice cream)

Resolve with C2 :-

- Likes (Mary, Ice cream)

conclusion - Mary likes Ice cream.

24/12/24

[Module-4]

① Using predicate logic
- is a formal system used to represent & reason about knowledge.
- extends propositional logic by incorporating quantifiers and predicates
- to express relationships b/w objects & their properties.

Representing simple facts in logic
• Atomic facts → In predicate logic, simple facts are represented using predicates, variables and constants.
For ex - a fact like "John is a student" could be represented as:-
  • Student(John): Here, Student is predicate & John is constant.

• General facts → facts can also be generalized with variables.
For ex, "All humans are mortal" can be represented as:-
  • $\forall x (Human(x) \rightarrow Mortal(x))$
  • This means "for all x, if x is a human, then x is mortal".

Representing instant & ISA Relationships
Instant Relationship → Describes a specific instance / individual.
For ex:- "John is a human":-
  • Human (John):- John is an instance of human.

ISA Relationship (IS-A Relationship) → Represents inheritance or classification
For ex:- "John is a student":
  • Student (John) implies Human (John).
  • ISA typically represents hierarchies, like "A dog is an animal" in terms of an ISA relationship.

Computable functions & predicates
• Computable functions → Functions can be defined within predicate logic that can compute values.
For ex:- a fun. that computes the sq. of a no. could be represented as:-
  • Square (x, y): Means that y is the square of x.

• Predicates → used to express properties / relationships.
ex:- father (x, y) → relationship "x is the father of y".

Approaches to Knowledge Representation.

① Logical Representation →
uses rules & facts in form of logic. ex. if it rains, ground is wet.

② Semantic Networks →
info is stored like a mind map with nodes (concepts) & links (relationships)
ex. Dog → is-a → Animal.

③ Frame-based Representation →
Like a structured template with slots for attributes.
ex - A "car" frame has slots for "color", "Brand", and "Model".

④ Production Rules →
~~Like~~ a If-then rules for decision-making
ex - if it's dark, then turn on ~~the~~ lights.

⑤ Ontologies →
rep. knowledge about a domain using categories & relationships
ex - bio. ontology links "Plants" and "Photosynthesis".

Issues in Knowledge Representation.

① Representation of Uncertainty → real-world knowledge is often uncertain and incomplete. AI systems need ways to represent & reason with uncertain knowledge.
Techniques like → Probabilistic reasoning (Bayesian networks)
→ Fuzzy logic
used to handle uncertainty.

② Incompleteness → sometimes, knowledge may be incomplete / partial. Systems must be able to make reasonable inferences / guesses based on limited info.

③ Dynamic knowledge ~~Context Dependence~~ → Knowledge is not static & it may change over time (eg. new discoveries). AI system needs to be able to handle dynamic updates to their knowledge base.

④ Context Dependence → Knowledge may vary based on context in which it is applied
ex. meaning of a word can change depending on the context.

⑤ Common sense Knowledge → Representing everyday knowledge that is typically intuitive for humans but difficult for m/c to encode
(ex - if it's raining, people usually carry umbrella).

Knowledge Representation Issues →

1) Expressiveness → how well the represent^n. can capture complex facts.
2) Scalability → ability to handle large-scale knowledge bases efficiently.
3) Uncertainty → handling incomplete / ambiguous info.
4) Computability → balancing the richness of rep^n. with feasibility of comput^n.
5) Inference → deriving new knowledge from existing knowledge.
6) Ambiguity & Vagueness → Real-world knowledge is often uncertain, incomplete or imprecise making rep^n. difficult.

how AI stores knowledge
Representation & Mapping → how AI use that knowledge to solve problem.
↳ how AI connects the knowledge it has stored (represen^n.) to real world problems

how a m/c stores and organizes knowledge/info. about the world.

Think of it like a recipe book.
The recipes are written in a way that humans can understand & follow.
In the same way, "lang." or format to store knowledge so it can use it to solve problems.
↳ Representation.

Think of mapping as using a recipe to actually cook food. AI uses its stored knowledge to make decisions or solve tasks.

Exampl of Representation

① List & Tables
Ex - Shopping list tells you what to buy.
For AI: Data can be stored as lists / tables for quick lookup.

② Rules
Ex: "If it's raining, take an umbrella."
AI: Rules help AI make decisions, like how expert systems work.

③ Network & connections
A mind map connects ideas (Dog → Animal → Living Thing)

AI: Network helps store relationships b/w things

④ Pictures / Graphs
A road map shows places & how to get from one to another.
For AI: A map can help robots navigate.

Ex. of Mapping

① Turning data into Action.
ex - When GPS ~~sh~~ maps your current location & shows direc^n. to your destination.
For AI :- m/c matches stored road data with your locat^n. to guide you

② Understanding language :
When you say "Turn on lights" a smart home syst. maps your words to an action (turning on lights)
For AI :- It maps your spoken lang to a stored command.

③ Problem solving - A chess AI maps the current board (rep^n.) to the best move by using rule of game.

23/10/24

Module-3

① Knowledge and Reasoning - form the backbone of intelligent systems, enabling them to simulate human-like understanding & decision-making.

Knowledge Representation - method of encoding info. about world into a format that machines can process to make decisions.

Methods:- 1> Logical Representation → rep. facts as simple T/F statements (eg. "It is raining").

predicate logic → expresses relationships & quantifiers (eg. "All humans are mortal: ∀x (Human(x) → Mortal(x))").

2> Semantic Networks → rep. concepts as nodes & relationship as edges in graph
→ex:- A "Dog" node connected to an "Animal" node via "is-a" edge.

3> Frames → rep. stereotypical situations/obj. using structured format.
ex- A "Restaurant" frame with slots for "Menu", "Waiter", "Bill".

4> Rules (Produc. Syst.) → knowledge is rep. as "if-then" rules
→ex - if Temp. > 30°C, then turn on the a.c.

Reasoning - process of drawing conclusions from knowledge.

Types of reasoning -

① Deductive " → starts with general facts/rules & deduces specific conclusion.
ex- "All men are mortal. Socrates is a man. ∴ Socrates is mortal."

② Inductive Reasoning → derives general rules from specific observations
ex- observing the sun rises every morning & concluding that it will rise tomorrow

③ Abductive Reasoning → Infers the best explanation for observed data.
ex:- Noticing wet ground & concluding it probably rained.

④ Probabilistic Reasoning → Deals with uncertainity using probability theory
Tools:- Bayesian networks, Markov Models.

Knowledge Represen
1> Expressiveness →
2> Scalability → ab
3> uncertainty → b
4> Computability →
5> Inference → de
6> Ambiguity &

how AI stores knowledge
Representation
how a m/c st
Think of it
The recipes
in the same
use it to s
↳ Repres

Think of
stored kno

Exampl of
① List &
Ex - Shoppi
for AI: D
for quic

② Rules
Ex:- if
AI:- Ru
how ex

③ Netwo
A min

AI: N

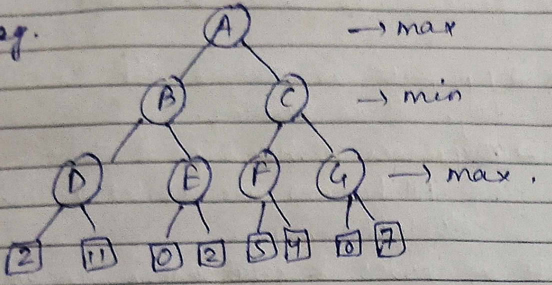
④ Pic
A
ge
for

Minimax search procedure

(BFS is not used bcz. it is level-wise)

→ Backtracking algorithm.
→ Best move strategy used.
→ Max will try to maximize its utility (best move)
→ Min. will try to minimize utility (worst move)

eg.

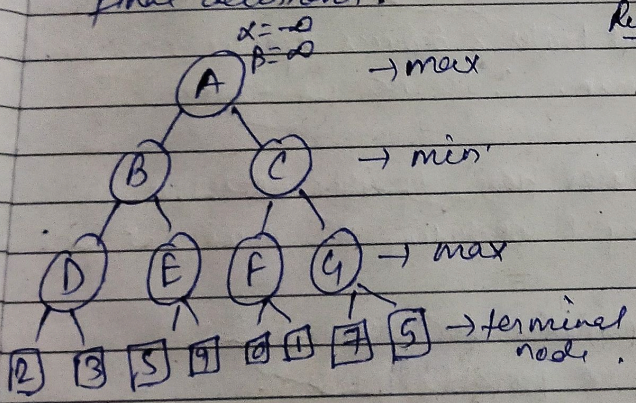


TC :- $O(b^d)$

SC :- $O(bd) \approx$ DFS.

α-β pruning

→ modification of minimax algo.
→ α → best choice (highest utility / value) maximize.
→ β → best choice (lowest ") minimize.
→ there is a techn. by which without checking each node of game tree, we can compute correct minimax decision & this techn. is called pruning
→ Removes all the nodes which are not really affecting the final decisions.



Rule ① while backtracking, node values will be passed to upper nodes instead of α-β values. (while moving backward)

② we will only pass α, β values to child nodes (while moving forward)

Additional refinements

Iterative deepening → combines benefits of DFS & BFS
→ performing DFS repeatedly with increasing depth limit.
→ ensures that shallowest soln. is found while still maintaining memory efficiency.

ex.

Local search for CSPs

→ an optimisⁿ technique
→ starts with complete assignment of values to variables & iteratively improves it.
→ Unlike backtracking which constructs partial solⁿ step by step,
local search operates directly on complete assignments & focuses on
minimizing constraints violations.
→ useful for solving large scale CSPs
other systematic method (backtracking) may be computationally expensive

[Steps]

Initialisⁿ → starts with complete assignment
Evaluatⁿ → calculate no. of violated constraints
Neighbor selection → generate neighbouring assignments by ~~generating~~ modifying current assignments.
Move to neighbor → select a neighbor with fewer violations (lower cost) & update current assignment.
Termination → stop when :- all constraints are satisfied
:- a predefined no. of iterations / time limit is reached.

③ Adversarial Search.

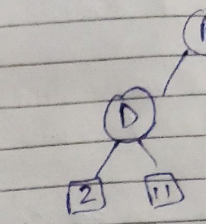
Games → Games like chess, checkers, tic-tac-toe provide ideal env. for adversarial search where 2 agents compete against each other

Optimal Decision & strategies in Games →
each agent aims to maximize its utility while minimizing the opponent's utility - leads to strategic decision making

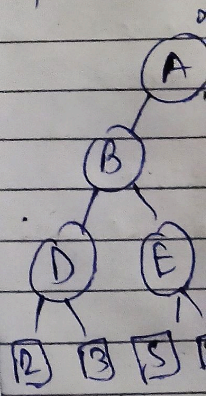
→ critical concept in AI
→ for problems involving competitive env. (games)
→ focus on finding optimal strategies
→ multiple agents (players) with opposing interests are involved

Minimax search p
→ Backtracking al
→ Best move stra
→ Max will try
→ Min will try

eg.



α-β pruni
→ modification
→ α → best
→ β → best
→ there is a
tree, we
is called
→ Removes
final dec



Additio
Iteratin

ex

Applications → Optimization problems → Travelling Salesman Problem (TSP)
→ Knapsack problem
→ Job Scheduling & Resource Allocation
→ Machine Learning → Evolving neural networks (Neuroevolution)
→ Bioinformatics → DNA seq. alignment
↳ Protein str. prediction.
→ Robotics
→ Game Theory.

Constraint Satisfaction problems (CSP)

→ Type of mathematical problem
→ defined by set of variables, their possible values, & constraints that restrict the values the variable simultaneously take.
→ CSPs widely used in AI, Operational search & optimiz^n^ to solve problem

Components of CSP :-

① Variables :- Denoted as $X_1, X_2, \dots, X_n$.
(represent the entities whose value needs to be assigned)

② Domains :- Each variable X_i has domain D_i, (set of possible value it can take)
$X_1 \in \{1, 2, 3\}$.

③ Constraints :- Relationships / rules that specify allowable combin^n^. of values for subsets of variables.
ex :- $X_1 \neq X_2$ or $X_3 + X_4 \leq 10$.

④ Goal - Find an assignment of values to all variables such that all constraints are satisfied.

Example :- Sudoku Puzzle

1. variable → each cell in grid is variable.
2. domain → values for each cell range from 1 to 9.
3. constraints → all cells in a → row / → column / → 3×3 subgrid must have distinct values

• Simulated annealing search -
- probabilistic local search algo. inspired by annealing in metallurgy.
→ starts with a random solⁿ & iteratively explores neighbouring solⁿ.
→ allows occasional moves to worse solⁿ, particularly early in process, to escape local optima.
→ probability of accepting worse solⁿ ↓ over time (analogous to cooling in physical annealing)
→ Time complexity :- $O(b^k)$ k → no. of iterations, b → branching factor.
Space " :- $O(1)$ (only stores current solⁿ)
- Adv. → can escape local optima
→ explore broader part of solⁿ space.
- disadv. → req. careful tuning of parameters (cooling schedule)
→ may take a long time to coverage to an optimal solⁿ.

~~Local beam search~~ Steps →
1. Start with initial solⁿ & a high temp.
2. Generate a neighbouring solⁿ
3. If new solⁿ is better, accept it.
4. If new solⁿ is worse, accept it with probab. proportⁿ to temp.
5. Gradually ↓ the temp.

• Local Beam Search -

• Genetic Algorithm -
• Inspired by process of natural selection.
• Works with pop. of candidate solⁿ, evolving them over generations using genetic operations like selection, crossover & mutation
• effective for solving complex optimizⁿ. problems where search space is vast, non-linear & discontinuous.

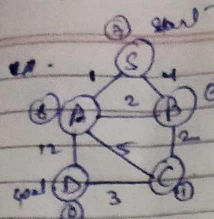
Steps
1) Initialization → generate an initial pop. randomly / based on domain-specific heuristic
2) Evaluation → Compute fitness of each indivi. in pop. using fitness function.
3) Selection → Select individuals based on their fitness score for reprodⁿ.
4) Crossover → perform this to create offspring from selected parents
5) Mutation → Apply this to offsprings to introduce diversity.
6) Replacement → replace the old pop. with new pop. (offspring)
7) Termination → Repeat 2-6 until terminⁿ condition is met :-
(max. no. of generation
solⁿ with satisfactory fitness
no significant improvement in fitness over generation)

Applications → Optim...
→ M...
→ Bi...
→ Ro...
→ ...

Constraint S...
→ Type of ma...
→ defined by ...
that restrict...
→ CSPs wide...

Components
① Variables :- ...
(represent...
② Domains : ...
③ Constraint...
④ Goal - ...

Example :- ...
1. Variable...
2. domain...
3. Constrain...



$S \to A : 1+6 = 7$

$S \to B : 4+2 = 6$ ✓

exploring $(S \to B)$

$S \to B \to C = 4+2+1 = 7$

$\boxed{S \to B \to C \to D = 4+2+3+0 = 9}$

exploring $(S \to A)$

$S \to A \to B = 1+2+2 = 5$ ok

$S \to A \to C = 1+5+1 = 7$

$\boxed{S \to D = 1+12 = 13}$

exploring $(S \to A \to B)$

$\boxed{S \to A \to B \to C \to D = 1+2+2+3+0 = 8}$ ✓

exploring $(S \to A \to C)$

$\boxed{S \to A \to C \to D = 1+5+3+0 = 9}$

Shortest route is $\boxed{S \to A \to B \to C \to D}$ 8.

Memory bound heuristic search: [Local search algo. & optimiz^n problem]
- Hill climbing search
- Simulated Annealing search
- Local Beam search.
- Genetic Algorithms.

Typically focus on finding a sol^n. by iteratively improving the current state based on local changes, without considering the entire search space.

~~Local~~

•) **Hill Climbing Search**

→ Simple local search algo.

→ Starts from initial sol^n. & iteratively moves to neighbouring sol^n. with
- •higher value (maximization)
- •lower value (minimization)

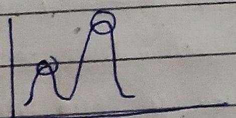
→ algo. stops when reaches local maximum (or minimum) (where no better neighbouring sol^n. exists).

→ Time complexity: $O(b)$ — branching factor

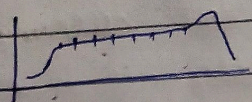
Space complexity :- $O(1)$ (∵ only current sol^n. needs to be stored).

→ Advantages:
- •Simple to use
- •computationally efficient
- •useful for problems with larger / continuous space)

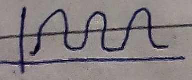
→ Limitations :-
1) Local maxima.



2) Plateau →



3) Ridge



Types

1) Simple Hill climbing → considers only immediate neighbour & moves if it improves sol^n.

2) Steepest - Ascent Hill climbing → Evaluates all neighbours & selects the best one

3) Stochastic-hill climbing → Randomly selects a neighbor & moves if it improves sol^n.

Search Strategy	Optimal	complete	Time complexity	Space complexity	Adv.	disadv.
BFS	yes	yes	$O(b^d)$	$O(b^d)$	• Guarantees shortest path • complete.	• Memory intensive • impractical for large spaces.
DFS	No	yes	$O(b^m)$	$O(bm)$	• Memory efficient • explores deeper path.	• can be non optimal • might go in ∞ loops
DLS	No	yes	$O(b^l)$	$O(bl)$	• Prevents ∞ loop • memory efficient	• may miss sol^n if depth limit is too low
Bidirec^n.	yes	yes	$O(b^{d/2})$	$O(b^{d/2})$	• faster than unidirec^n.	• Req. goal state • complex implement.

② Heuristic Search Strategies

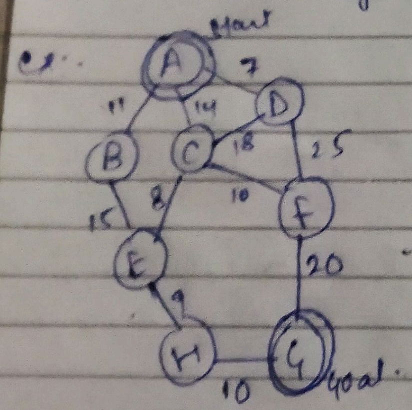
· Greedy Best-First search.

→ uses a heuristic to estimate the cost from the current node to the goal. & choose the node with lowest estimated cost.

→ faster

→ doesn't always guarantees an optimal sol^n.

ex.



Straight line dist.	Open	closed
A→G = 40	[A]	[ ]
B→G = 32	[C, B, D]	[A]
C→G = 25	[F, E, B, D]	[A, C]
D→G = 35	[G, E, B, D]	[A, C, F]
E→G = 19	[E, B, D]	[A, C, F, G]
F→G = 17		
G→G = 0		
H→G 10		

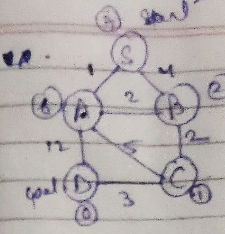
So, $\boxed{A \xrightarrow{14} C \xrightarrow{10} F \xrightarrow{20} G}$

44.

Time space complexity $= O(b^d)$

A* search - combines advantage of both BFS & greedy search.
- uses heuristic $h(n)$ & cost $g(n)$ to reach the current node to determine the next node to explore.
- guarantees an optimal sol^n. if heuristic is admissible (never overestimates the cost).

ex.



Shortest rou

Memory bound

Typically focus by iteratively current state changes, witho the entire

~~Local~~

•) Hill Climbin

→ Simple local

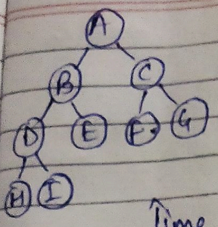
→ Starts fron

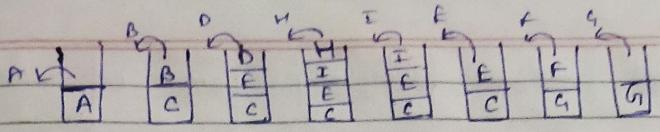
→ algo. stops (where no

→ Time compl

→ Advantages

→ Limitation

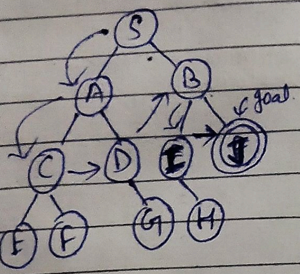




$A \to B \to D \to H \to I \to E \to C \to F \to G$.

Time complexity = $O(b^d)$.

Depth-limited Search → Similar to DFS,
→ with predetermined depth limit to avoid ∞ paths.
→ may not find sol^n. if limit is too small.



Keeping the limit upto level 2

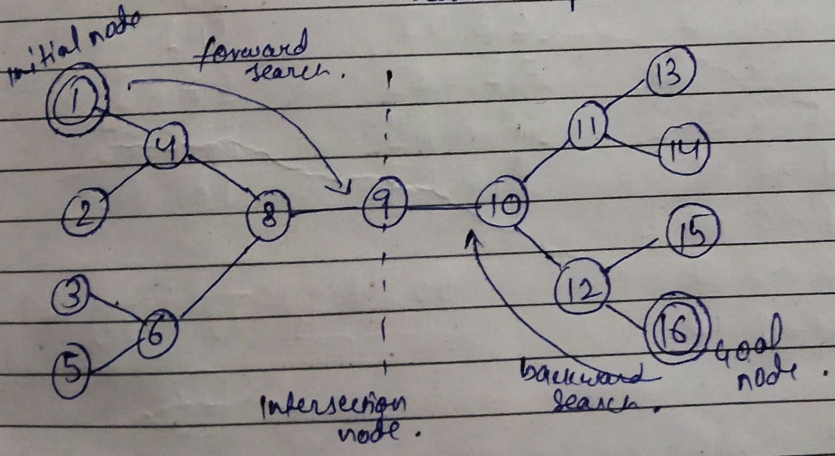
$S \to A \to C \to D \to B \to I \to J$

Time comp. → $O(b^d)$

Space comp. → $O(bd)$.

Bidirectional Search → Searches from both the start state & the goal state simultaneously, aiming to meet in the middle.
→ can be more efficient than unidirec^n. search in some scenarios.
→ any search tech. can be used (BFS, DFS, ...)
→ Search stop cond^n. :- when graph intersects.

2 searches
- forward search (start → goal)
- backward search (Goal to start)



Time complexity $O(b^{d})$.

Comparing Uniform Search Strategies.
BFS, DFS, DLS, Bidirectional Search

22-12-2024

Module-(2)

(1) Search Techniques.

Solving problems by searching:-

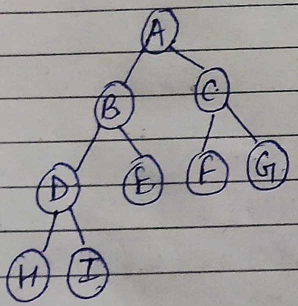
- Problem Solving Agents → aim to find sol^n to problems by exploring the search space.
- Searching for solutions → explores diff. states of the problem & applies actions to transition from 1 state to another, ultimately finding the goal state.

Uniform Search Strategies → BFS, DFS, DLS, Bidirectional search.
(uninformed search strategies)

* No info. (blind search)
* Time consuming
* No knowledge

*More complex.

Breadth-first search (BFS) → Explores all nodes at present depth level before moving at the next depth level.
→ Guarantees shortest path in unweighted graph
→ memory intensive
→ FIFO (Queue)
→ complete & optimal sol^n.

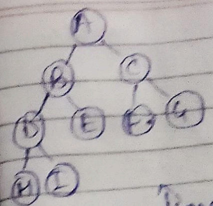


A ← Queue.
B C ← (after removing A)
C D E ← (" " B)
D E F G ← (" " C)
E F G H I ← (" " I)
F G H I ← (" " E)
G H I ← (" " F)
H I ← (" " G)
I ← (" " H)

A → B → C → D → E → F → G → H → I

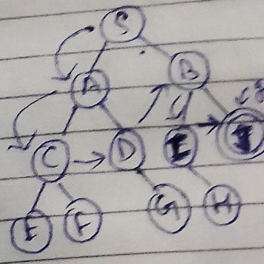
Time complexity = $O(b^d)$.

Depth-first search (DFS):- Explores as far as possible down one branch before backtracking.
→ memory-efficient
→ may not find shortest path
→ LIFO (Stack)
→ Incomplete & non-optimal.



Time compl

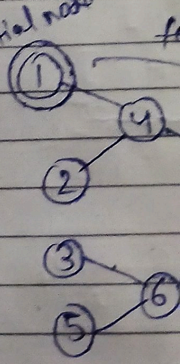
Depth-limited Search



Bidirectional S

2 Searches

Forward Search (Start→goal)



Comparing

Known env.
agent has complete knowledge of env. rules, actions & effects are predictable.
ex- Board games (checkers / chess)

Unknown env.
agent has incomplete knowledge of env. & must learn through interaction.
ex- autonomous robot navigation.
(req. agent to learn & adapt to new, uncharted situations).

Types of Agents:-
→ Simple Reflex Agents
→ Model based Agents
→ Goal based Agents
→ Utility based Agents
→ Learning agents

③ Problem Solving

Problems → A problem is typically defined in terms of state space, where states represent possible configurations & actions transition b/w them.

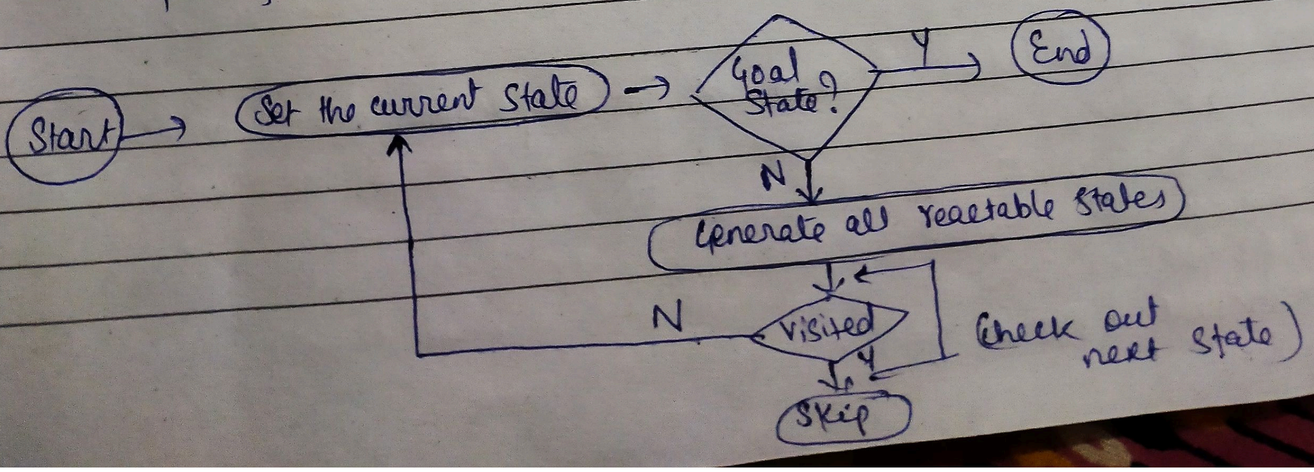
Problems as State space search → Representing the problem as a search through possible ~~outcom~~ states with a goal of finding the sol^n path.

Production system → set of rules & a state space used to define a problem and its sol^n.

S : {S, A, Actions(s), Result (S, a), cost (S, a)}

Problem characteristics → Vary in terms of size, complexity, whether deterministic or ND.

Issues in design of search programs →
→ efficiency
→ handling large state space
→ finding optimal sol^n. in an acceptable amnt. of time.



Deterministic
outcome of an action is predictable.
ex- Tic-Tac-Toe (outcome of each move is completely determined by current state of board).

Stochastic
have uncertainity
ex- Stock market trading (outcome of investment decisions is influenced by many unpredictable factors, ex- market trends, ext. events)

Static
env. remains unchanged while the agent is thinking / making decisions.
ex- Solving a mathematical eq.
(the problm doesn't change as Agent processes it).

Dynamic
env. changes over time, potentially even while the agent is deciding what to do.
ex- Online multiplayer video games (game world constantly evolves with other players taking actions that affect the env. in real time).

Discrete
State of env. changes in distinct steps, there is finite set of possible states & actions
ex- Board games (chess)
(Limited option for each move)

continuous
states & actions can vary continuously without discrete steps / limits.
ex- driving a car in real-world (car can be at any point in continuous space, can take continous range of speed)

Single Agent
only 1 Agent interacts with env.
agent's action don't directly affect other agents.
ex- A robotic Vacuum cleaner (operates independently)

Multi-Agent
multiple agents interact with the env.
actions may affect one another's outcomes.
ex- Autonomous vehicles on road. (each vehicle interacts with others avoiding collisions & following traffic rules)

Episodic
agent's current decision making doesn't affect future actions (episode is independent).
ex- Medical diagnosis (in some cases)
(each patient's visit is treated as an independent episode e- Diagnosing a patient with specific illness without considering their past history)

sequential
agent's action have consequences that affect future decisions.
(must plan ahead to achieve its goal).
ex- chess (each move depends on previous ones, decisions made early in game influence the future)

Known env.
agent has complete kn
rules, actions & eff
ex- Board games (ch

Types of Agents:

③ Problem So
Problems → A
st

Problems as
possible outc

Production S

S: {

Problem ch

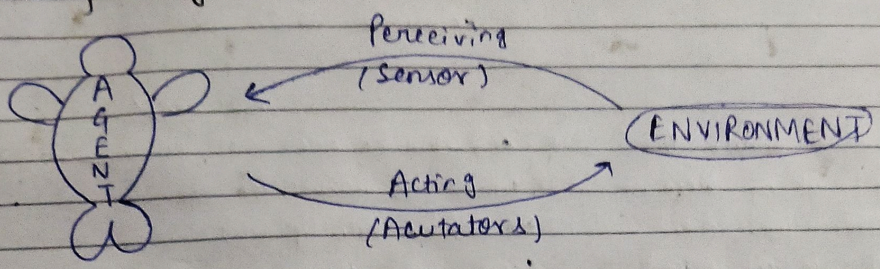
Issues
→ e
→ v
→

Start

iii) Evaluation function → win for X → +1
win for O → -1
draw → 0

iv) Game strategy →
• Perfect play → draw.
• Strategy of AI → maximize its chance of winning

② Intelligent Agents



Agent is something that perceives its env. through sensor & acts upon it using actuators.

AI agent must be able to perceive the environment.
Make decisions based on sensed data
Action must be generated & performed based on decisions.
Goals of Agents → high performance
best/optimized result
rational action.

Nature of environment :-
→ Fully vs. Partially observable.
→ Static vs. Dynamic
→ Discrete vs. continuous
→ Episodic vs. Sequential
→ Known vs. unknown
→ Accessible vs. Inaccessible
→ Single Agent vs. Multi Agents.
→ Deterministic vs. Stochastic.

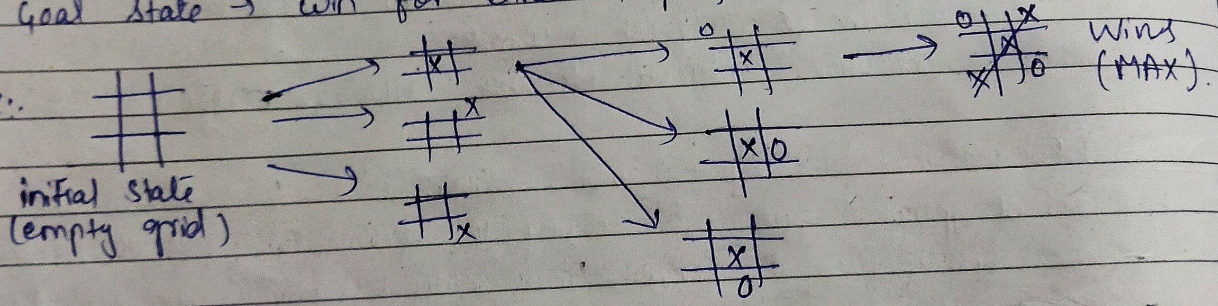
Fully observable → agent has access to all info. about the state of env. at any given time.
ex - chess game (bcz entire game is visible to both players at all time).
Partially observable → agent has partial info. about state of env. (complex decision)
ex - self-driving car (car has sensors (cameras, LIDAR) to perceive its surrounding but can't see everything like what's happening behind building or out of sensor range).

⑥ Fuzzy logic - allows rasoning with imprecise info.
- used in systems where binary T/F logic is insuffici.
- ex- in control system, where variables might be partially true (eg. "T is slightly high").

⑦ computer vision - allow machines to interpret & understand visual world.
- Image segmentation → Dividing an image into meaningful parts.
- Object detection → Identifying objects within an image.
- Face Recognition → Identifying & verifying faces in images or videos.

Tic-Tac-Toe Problem:-
→ 2 player game where one player called MAX marks a letter 'X' and the opponent called MIN marks a letter 'O'.
→ 3x3 Grid - where 2 players put their letters.
→ player with same marks (3) in complete ROW | COL. | DIAGONAL wins the Game.
→ Starts state → Empty Grid
→ Goal state → win for either of player < MAX (3 consecutive X) / MIN (3 consecutive O)

ex:-



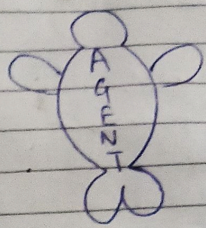
i) Problem represented as $\boxed{\text{Game Tree}}$ → each node ⇒ game state. → each edge ⇒ possible action (placing a mark in empty cell).

ii) Search techniques → $\boxed{\text{Minimax Algorithm}}$ → Algo. explores possible moves using a heuristic function. (Giving +ve scores → winning state, -ve scores → losing state)

→ $\boxed{\alpha-\beta \text{ Pruning}}$ → optimization of Minimax algo.
↳ it eliminates branches of game tree that don't need to be explored bcz they can't influence the final decision. (reduces no. of nodes).

iii) Evaluation function

② Intelligent Agents

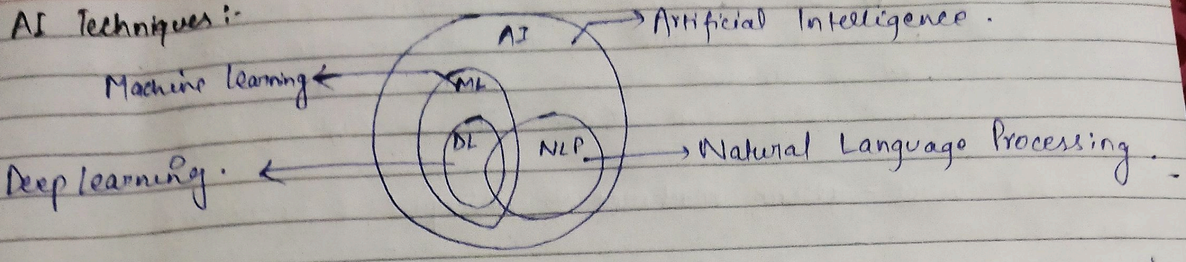


AI agent must b
Make decisions
Action must b
Goals of Agen

Nature of en

Fully observab
ex - chess
Partially o
ex - Self-

AI Techniques :-



① Machine Learning

Supervised learning → model is trained on labelled data, and the algo. learns to map i/p to o/p.

→ common techniques → decision tree, SVM (Support vector m/c), neural network.

unsupervised learning → models work with unlabeled data & tries to find hidden patterns / structure.

→ ex- clustering algo. (K-means) & dimensionality.

Reinforcement learning → learns by interacting with env. & receiving feedback.

② Deep Learning → subset of ML

→ It uses artificial neural networks with many layers.

→ for image & speech recognition.

→ Convolutional Neural Networ (CNNs) & RNNs (recurrent) are widely used.

③ NLP → Tokenization → breaking text into smaller units. (words / sentences)

→ Part-of-speech Tagging → Identifying the grammatical role of each word.

→ Named Entity Recognition (NER) → " " entity like names, dates & locations.

→ M/C Translation → Translating text from 1 lang. to another.

→ Sentiment Analysis → Determining the sentiment (+ve, -ve, neutral) in text.

④ Expert Systems → rely on if-then rule-based resning, simulating the decision-making abilities of human experts.

⑤ Search Algorithm → used to explore problem spaces & find solutions.

- BFS → explores all possible nodes level by level
- DFS → explores as far down a branch as possible before backtracking.
- A* Search → combines benefits of BFS + Greedy algo.

21-12-2024

Module-I

1) Overview of Artificial Intelligence

AI → Intelligence.
Artificial

[Intelligence exhibited by Machines]
Humans / comp. Systems.

→ Reasoning
→ Learning
→ Problem Solving
→ Perception.

Problems of AI :-

Problem arises due to → inherent complexity of mimicking human cognition and decision-making in machines.

1) Understanding & Replicating human Intelligence

- Human cognition complexity :- Human cognition is complex & involves a vast range of variables like emotions, sensory inputs, experiences. Fully replicating human intelligence is a signi. challenge due to its multi-D nature.
- Generalization :- Generalizing across diff. domains (like human do) is extremely difficult. AI system lack common-sense reasoning & struggle to adapt to new and unforeseen situations.

2) Bias → in training data can cause AI system to produce unfair or discriminatory outcomes, especially in sensitive areas like hiring or law enforcement.

3) Black Boxes → Many AI models, (deep learning models) operate as black boxes, "making it difficult to understand how they make decisions. This lack of transparency raises concerns around accountability in critical situation.

4) Ethical concerns → autonomous vehicles where decisions may involve life or death scenarios.

5) Computational Resources & Efficiency → can be expensive & environmentally taxing. Scalability & efficiency remain challeng as AI systems grow more complex.

AI Techniques
Machine
Deep learning
1) Machine L
Supervised
Unsupervis
Reinforcem
2) Deep Lea
3) NLP
4) Expe
5) Sear
• B
• D
• A