05/05

Brijesh R. Sharma
Class - D15C
Roll No. 48
Assignment - 1

Q1) What is AI? Considering the COVID-19 pandemic situation, how AI helped to survive and renovated our way of life with different application?

Ans: AI (artificial intelligence refers to the simulation of human intelligence in machines that can perform tasks such as learning, reasoning, problem solving, perception, language understanding, and decision making. ~~AI~~

AI helped during the Covid-19 Pandemic in various ways:

1] Healthcare and Diagnosis -

a) AI helped in detecting COVID-19 symptoms quickly by analyzing medical images (like x-rays and CT-scans).

b) It helped in researching and identifying potential treatments and vaccines faster by analyzing vast amounts of medical data.

c) AI-driven chatbots answered public queries, assessed symptoms.

2) Predictive Analysis and Data Analysis -

AI analyzed global data to predict virus spread patterns and help governments implement preventive measures

3) Remote Work and Education

a) AI tools such as real time caption, scheduling assistants, enhanced remote work & productivity.

b) AI tools for automated assessments, interactive learning provided personalized education

4) Contactless Technology

~~c)~~ Robots were deployed in hospitals to deliver medicines, ~~defect~~ disinfect rooms, and assist in surgeries, reducing human exposure to virus.

5) E-commerce

a) AI optimized supply chains by predicting demand, ensuring timely delivery of essential goods and managing warehouse inventory.

b) AI recommended products based on user behaviour, helping in online sales.

2) what are AI agents terminology, explain with examples

Ans AI agents is an entity that perceives its environment through sensors, processes the information, and takes action using actuators to achieve a specific goal.

Key terminologies related to AI agents;

a) Agents - It is anything that can perceive its environment and act upon it. AI agents can be software-based or hardware based.

e.g. self-driving cars,

b) Environment - The environment is everything that surrounds an AI agent and affect its performance.

e.g. chess board, opponent, moves, the rules

c) Perception:

Perception is the agent ability to gather information from the environment through sensors.

e.g. facial recognition system perceives images through camera.

d) Sensors: Sensors are the input devices that collect data from the environment

e.g., voice assistant uses mic.

5) Actuators : Actuators are components that allow the agent to take action based on decisions

6) Rationality - An agent is considered rational if it makes the possible decision, based on the available information and a given objective.

e.g. A Stock trading AI

3) How AI technique is used to solve 8 puzzle problem?

Ans 8-puzzle problem is a classic AI problem with a 3x3 grid contains 8 numbered tiles and an empty space.

AI techniques for ~~search~~ solving the 8-puzzle problem:

1) Uninformed Search (Blind Search) Methods

a) Breadth-First Search (BFS)

b) Depth-First Search (DFS)

c) Iterative Deepening Depth-First search (IDDFS)

2) Informed Search (Heuristic-Based) ~~Model~~ methods

a) Best-First Search (Greedy Search)

b) A* Algorithm

But A* Algorithm with the Manhattan Distance heuristic is the most efficient approach for balancing cost and estimation effectively.

4) What is PEAS descriptor? Give PEAS descriptor for following.

Ans→ The PEAS (performance, measure, environment, actuators, sensors) description is used to define the components of an agent, helping to analyze its working environment and functions:

a) Performance measure (p) : The criteria for evaluating the agent's success.

b) Environment (E): The external surroundings where the agent operates.

c) Actuators (A): The mechanisms through which the agent interacts with the ~~intera~~ environment.

d) Sensors (s) - The device used to perceive the environment.

1) Taxi drivers :

p: safety, fuel efficiency, passenger satisfaction

e - roads, traffic signals, other vehicles, pedestrians, weather.

a : steering wheel, accelerator, brakes, horn, indicator, wipers, etc

s: GPS, cameras, Lidar, speed sensors, fuel gauge,

2) Medical diagnosis system

p - diagnosis, accuracy, speed of response, patient, etc

e - patients, symptoms, medical databases

a - music output system, speakers, digital files

s - user feedback, musical trends, mood analysis, input instruments.

4) An aircraft autolander

p : smooth and safe landing, fuel, efficiency, passenger comfort.

e - runway, weather conditions, altitude, air traffic

a - flaps, landing gear, engine throttle, air ~~breaks~~ brakes, s - GPS, ~~alt~~ altimeter, wind sensors, gyroscopes, radar, etc.

5) An essay evaluator -
p : accuracy in grading, fairness, grammar and detection
e - essays, answer sheets, writing rules, academic guide
a - score display, feedback generator,
s - Optimal character recognition, (OCR, NLP tools.

6) Robotic sentry gun for the keck lab
p - accuracy in target detection, response-time
e - lab premises. ~~instruct~~ intruders, authorized personnel, obstacles
a - gun turret, ~~a~~ alarm system, movement motors
s - motion detectors, thermal cameras, facial recognition, infrared sensors.

5) Categorize a shopping bot for an offline bookstore according to each of the six dimensions (fully / partially ~~obser~~ observable, deterministic / stochastic, episodic / sequential, static / dynamic, discrete / continuous, single / multi-agent).

Ans a) Observability - partially observable
The bot may not have complete information about books on shelves, customer preferences or stock updates without external input.
b) Deterministic vs Schotastic - ~~sequential sto~~ stochastic
Book availability may change due to manual sales, external purchases, Or misplaced books making environment uncertain.

c) Episodic vs sequential - Sequential
Each customer interaction affects the next steps (e.g. book recommendations depend on previous queries).

d) Static vs Dynamic - Dynamic
The environment changes (books sell out, new books arrive, customer preference shift) making it dynamic

e) Discrete vs ~~Di~~ continuous - Discrete
The bot deals with a finite set of actions (searching books, checking stock).

f) Single Agent vs MultiAgent - MultiAgent
The bot interacts with multiple customers, bookstore staff. and possibility other inventory systems.

5) Differentiate between model based and utility based agent.

Ans

Model Based Agent	Utility Based Agent
1) Uses an internal model of the environment to make decisions.	1) Uses a utility function to measure the desirability of different states and select the best action.
2) Chooses actions based on a representation of how the world works.	2) Chooses action that maximize its expected utility.
3) Maintains a model of the environment (how actions affect future state)	3) Uses a utility function to compare possible outcome.
4) Focuses on achieving a goal using state ~~transnt~~ transition models	4) Focuses on maximising long term benefits rather than just

	reaching a goal.
5) Eg. A self driving car that predicts patterns and plan routes	5) A stock trading bot that evaluates different portfolios to maximise returns.

7) Explain the architecture of knowledge based agent and learning agent.

Ans Architecture of knowledge based agent —

A knowledge based agent (KBA) is an AI system that uses stored knowledge to make informed decisions. It consists of the following components:

1) KB : It stores facts, rules and heuristics about the world.

2) Inference engine - It applies logical reasoning to device new knowledge from stored facts.

3) Perception (sensors) - It collects information from the environment.

4) Actuators - Performs actions based on inferences.

5) Knowledge acquisition module - Updates and expands the knowledge base with new data.

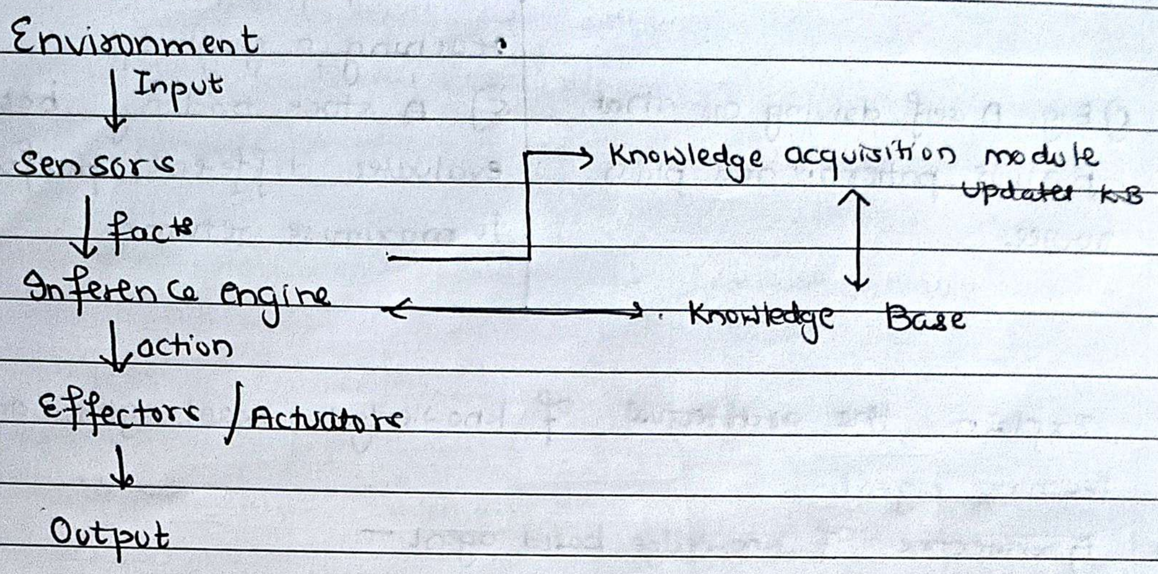
Working process

1) The agent perceives the environment.

2) It queries the knowledge base for relevant information

3) The inference engine applies logic to decide an action

4) The action is executed and KB is updated if needed.



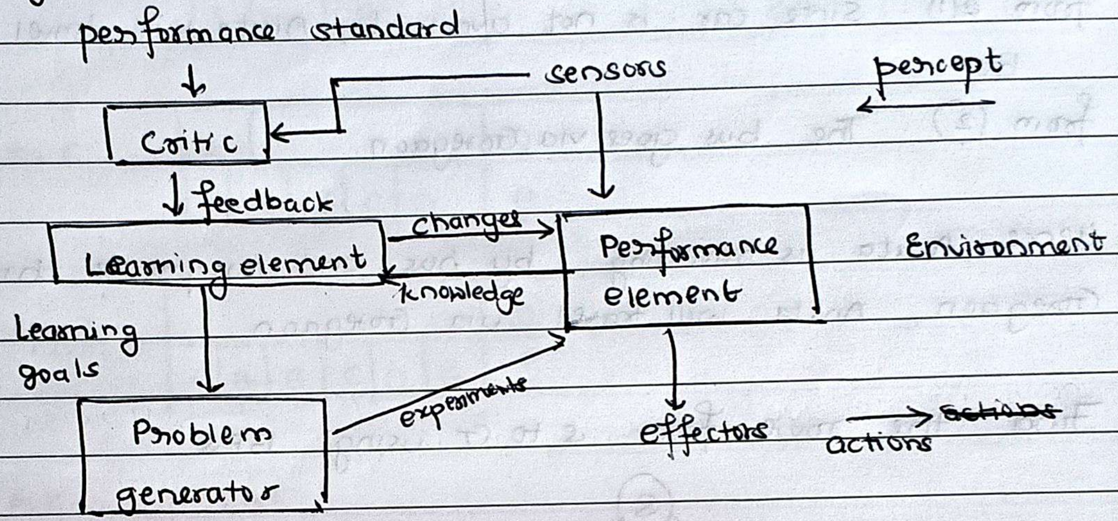
Architecture of learning agent.
A learning agent improve its performance outcomes by learning from past experiences.

Components of learning agent:
1) Learning element - Improves the agent's knowledge by analyzing past experiences.
2) Performance element - chooses actions based on the learned knowledge.
3) Critic - Evaluates the agent's actions by comparing outcomes with expected results.
4) Problem generator - suggests new exploratory actions to improve learning

Working process
1) The performance element makes decisions and take actions.
2) The critic evaluates the results and provides feedback.

3) The learning element updates the knowledge based on feedback.
4) The problem generator suggests new strategies to improve performance.

Agent:



8) Convert the following predicates.

a) Anita travels by car if available otherwise travels by bus,

Available (car) → Travels (Anita, car)

¬ Available (car) → Travels (Anita, Bus) — (1)

b) Bus goes via Andheri and Goregaon.

goes via(bus, Andheri) ∧ goes via(Bus, goregaon) — (2)

c) car has a puncture, so it is not available

puncture (car) → ¬ Available (car) — (3)

4)

Will Anita travel via Goregaon

Applying forward chaining :

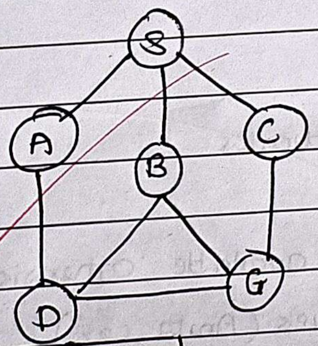
from (3) : We know that car has a puncture, so available (car) is false.

from ~~2~~(1) : Since car is not available, Anita will travel by bus.

from (2) : The bus goes via Goregaon.

Hence, Anita is travelling by bus and bus passes through Goregaon, Anita will travel via Goregaon.

9) Find the route from S to G using BFS.



Ans

	Queue (Q)
	processed (P)

Step 1:

S		Q
		P

Step 2:

A	B	C		Q
S				P

Step 3:

B	C	D		Q
S	A			P

Step 4:

C	D	G	Q
S	A	B	P

Step 5:

D	G			Q
S	A	B	C	P

Step 6:

G					Q
S	A	B	C	D	P

Step 7:

						Q
S	A	B	C	D	G	P

Adjacency List

S → {A, B, C)

A → {D}

B → {D, G}

C → {G}

D → {G}

From BFS and adjacency list

shortest path is S → B → G

other paths are S → C → G and S → B → D → G and

S → A → D → G

10) What do you mean by depth limited search? Explain iterative deepening search with example.

Ans→ Depth Limited Search-

Depth limited search is a variation of DFS where we improve a depth limit to avoid going too depth into an infinite or large search space.

Working:

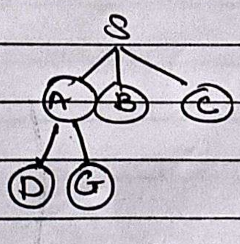
1) The algorithm follows a DFS strategy but limits the depth of recursion.

2) If the goal is not found within the limit, it returns failure or cutoff.

4) Thus helps in avoiding infiniteloops in problems with large or infinite depth.

Example:

Consider a graph where we want to find a path from S to G with depth limit of 2.



1) If the depth limit is 1, ~~we~~ we only explore S→A, B, C but cannot reach G.

2) If the depth limit of 2, we explore S ⟶ A ⟶ D, G

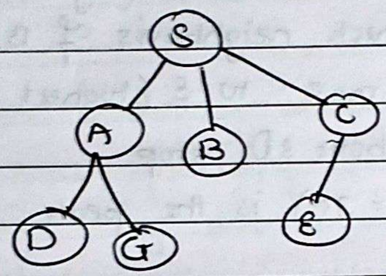
Iterative Deepening Search (IDS)

Iterative Deepening search (IDS) combine the benefits of depth-first search (DFS) and BFS. It repeatedly performs DLS with increasing depth limits until the goal is found.

Working:

1) Start with depth-limit = 0 and perform DLS.
2) Increase the depth limit and perform DLS again.
3) Repeat until goal is found.

e.g.



Depth limit = 0 → only node S is checked
Depth limit = 1 → explores A, B, C but G is not found
Depth limit = 2 explores D, G, E and G is found.

11) Explain Hill Climb and its drawback in detail with example. Also state limitations of steepest-ascent hill climbing?

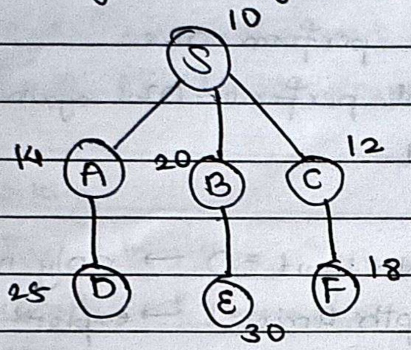
Ans- Hill climbing is an optimization algorithm that continuously moves towards higher-valued states i.e. better solutions ~~wi~~ until a peak (local maximum) is reached. It is a greedy algorithm that ~~peaks~~ evaluates neighbouring states and chooses the one with the highest value. It is widely used in AI, especially in problems like pathfinding, scheduling problems.

Working of Hill Climbing

1) Start with an initial state (solution).
2) Evaluate the neighbouring states of the current state.
3) Move to the best neighbouring state that improves the solution.
4) Repeat the process until no better neighbour is found on

predefined goal state is reached.

e.g. Consider a graph where each node has value representing its height and goal is to find the highest valued node.



1) Start at node S(10)

2) Check its neighbours: A, B, C then move to B (highest value)

3) Check neighbours of B: E(30) then move to E (highest value = 30)

4) E has no better neighbour so stop.

5) Final result: Node E (value = 30) is the peak.

Drawbacks:

1) Local Maxima - If node D (25) was chosen from A instead of B, it would be stuck there, not reaching 30.

2) Plateau - If multiple nodes had the same value, the algorithm might get confused.

3) Ridges - The algorithm cannot take downward moves to explore better paths.

4) No backtracking - Hill climbing does not remember previous states, so if it gets stuck, it cannot backtrack to explore better paths.

Steepest - Ascent Hill Climbing and its limitations -

steepest-ascent hill climbing is a variation where the algorithm evaluates all neighbouring states and moves to one with the highest improvement.

Limitations:

1) Since, it evaluates all neighbours, it takes more time and resources.

2) Even though it selects the best move at each step, it cannot escape local maxima.

3) Fails in plateaus and ridges

12) Explain simulated annealing and write its algorithm.

Ans → Simulated Annealing (SA) is an optimization algorithm inspired by ~~metal~~ metallurgical annealing, where materials are heated and then cooled to remove defects. It helps escape local maxima by ~~allocati~~ allowing occassional downward move to explore better solutions.

Working:

1) Start with an initial solution

2) Set a high temperature (T), which gradually cools down

3) Select a random neighbour of the current solution.

4) Calculate the energy difference (ΔE) between the new and current solution.

a) If the new solution is better, accept it.

b) If the new solution is worse, accept it with a probability $P = e^{-\Delta E/T}$ where e is Euler's number, T is current temperature

c) Reduce the temperature gradually

5) Repeat until the temperature is very low or a stopping condition is not met.

13) Explain A* with an example.

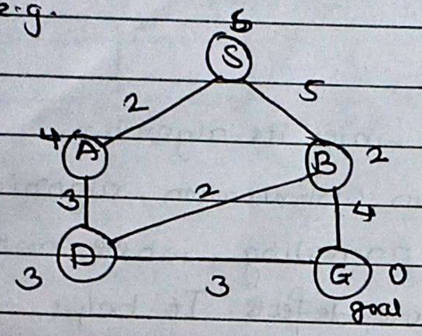
Ans → A* is widely used graph search and pathfinding algorithm that finds the shortest path between a start node and a goal node. It is an informed search algorithm that uses both.

1) Cost to reach the node g(n).

2) Estimated cost from the node to the goal h(n).

Formula : $f(n) = g(n) + h(n)$

e.g.



Steps :

1) Initialize open list
2) Expand node with lowest f(n)
3) Update g(n), h(n) and f(n) for neighbours
4) Repeat until goal node is reached

Node	g(n)	h(n)	f(n)
start (s)			
Expand A(S→A, cost = 2)	0	6	6
Expand B(S→B, cost=5)	2	4	6
Expand D(A→D, cast = 2+3 = 5)	5	2	7
Expand G(B→G, cost = 5+4 = 9)	5	3	8
Goal reached.	9	0	9

14) Explain minimax algorithm and draw game tree for Tic tac toe game.

Ans → Minimax is a decision making algorithm used in two-player games like Tic-Tac-Toe, chess and connect four. It helps in finding the best possible move for a player ~~for~~ by assuming that the opponent also ~~players~~ plays optimally. The algorithm alternates between maximising (for AI) and minimizing (for opponent) and each state have a value.

+1 → AI wins

-1 → opponent wins

0 → draws

Algorithm steps:

a) generate game tree for all possible moves.

b) Evaluates Terminal states:

→ If AI wins, return +1

→ If opponent wins, return -1

→ If draw, return 0

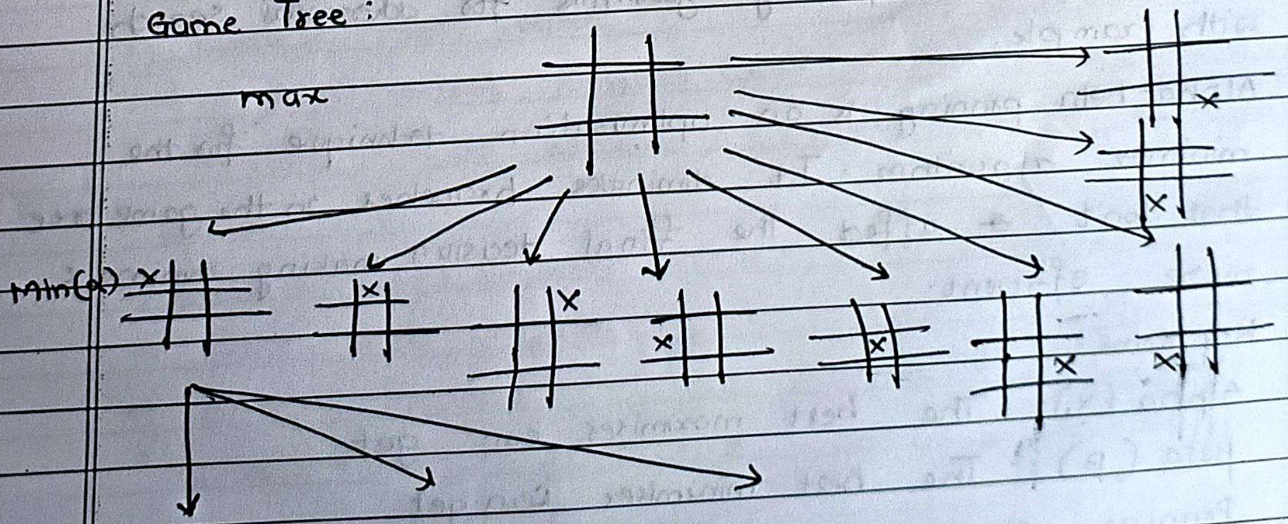
c) Backpropogate value

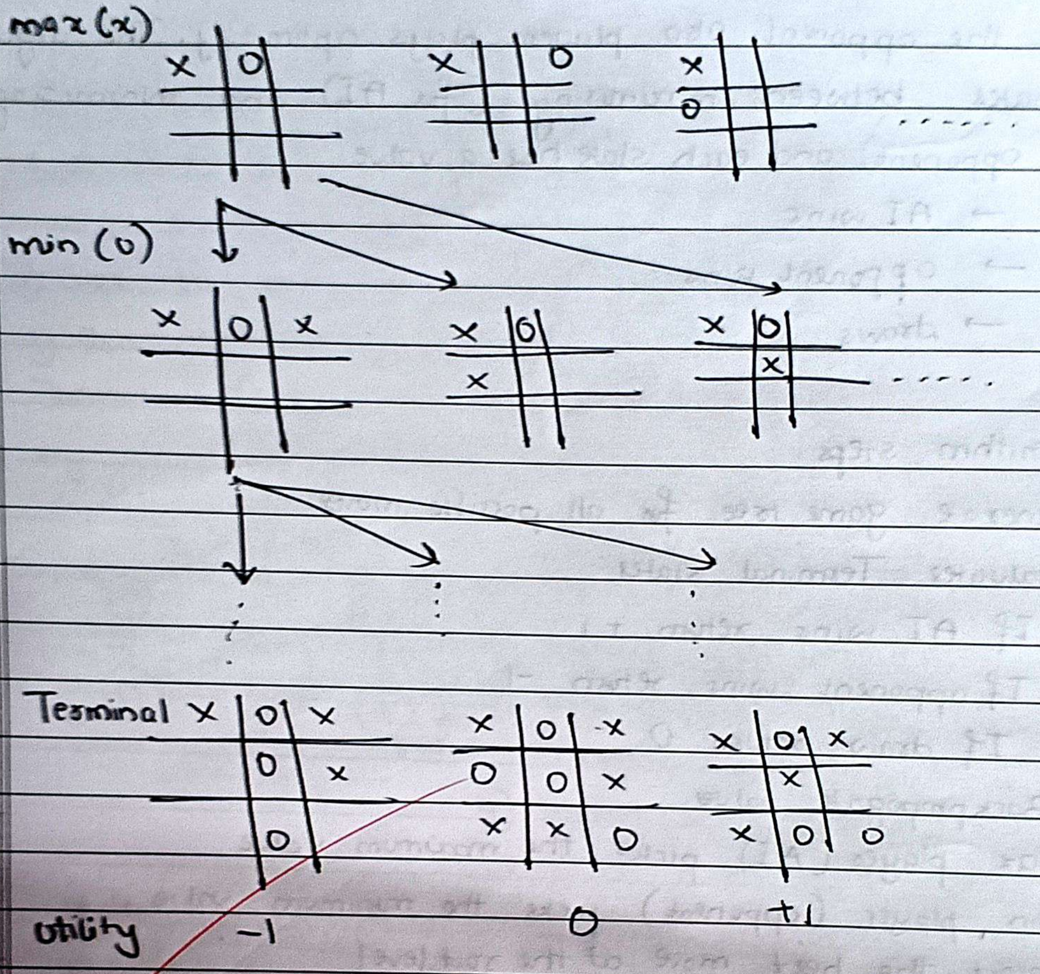
→ Max player (AI) picks the maximum value

→ Min player (opponent) picks the minimum value

d) Select the best move at the root level

Game Tree:





15) Explain alpha-beta pruning algorithms for adversarial search with example.

Ans: Alpha-beta pruning is an optimization technique for the minimax algorithm. It eliminates branches in the game tree that don't affect the final decision, making minimax more efficient.

Key terms:

Alpha (α) : The best maximiser can get

Beta (β) : The best minimiser can get

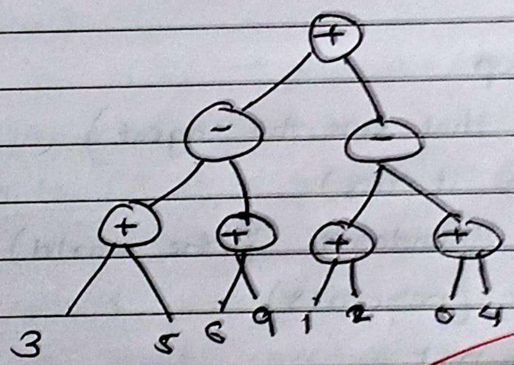
Pruning : stopping evaluation of nodes that won't affect the result.

Working:

1) Initialize $\alpha = -\infty$ and $\beta = \infty$
2) Traverse the game tree using minimax
3) Update α and β: if $\alpha >= \beta$ at any node, prune
4) Continue to the next branches.
5) Return the best move.

Example:

Consider the following game-tree where max (θ) wants to maximize the score, and min (θ) wants to minimize it



1) Initialise: Alpha (α) = $-\infty$ and beta (β) = $+\infty$
2) Evaluate left subtree
   → min(θ) chooses between (3,5) → selects 3 (smallest value)
   → max(θ) chooses between (3,6) → selects 6 (largest value)
   → update $\alpha = 6$
3) Evaluate right subtree
   → first node evaluated is 1
   → beta is updated
   → since $\alpha(6) \geq \beta(1)$, we prune (skip checking 2, 0 and 4)

6) Explain wumpus world environment giving it PEAS description Explain how percent sequence is generated.

Ans, The wumpus world is a grid based environment used in AI to demonstrate logical agent-based reasoning. It is a partially observable, stochastic, sequential environment. where an AI agent must navigate a cave-like world while avoiding danger.

Structure of wumpus world

1) Grid based (4x4 or larger)

2) contains

a) Gold (Gold pick it up

b) wumpus (A monster that kills the agent)

c) pits (If agent falls, it dies)

d) Walls (~~Bandwidt~~ Boundaries of the world)

3) sensory perceptions (percepts)

a) Breeze → near a pit

b) stench → near wumpus

c) glitter → gold is nearby

d) bump → hits a wall

e) scream → wumpus is dead.

PEAS description

P : +1000 for finding gold, -1000 for falling into a pit or encountering wumpus, -1 for every move, -10 for shooting arrow e : A 4x4 grid with the agent, ~~w~~ wumpus, pits, gold and walls.

a : move (up, down, left, right), grab (gold), shoot (arrow), climb (exit).

s : breeze, stench, glitter, bump, screen

A percept sequence is a history of all sensor inputs received by the agent over time.

e.g.

1) Agent starts at (1,1) → percepts = (Breeze) (pit is nearly wumpus)
2) moves to (1,2) → percepts = (Breeze, stench) (pit + near)
3) moves to (2,2) → percepts = (stench) (near wumpus)
4) moves to (3,2) → percepts = (glitter) (gold is nearby)

Agent collects percepts at each step and makes logical decisions based on past percepts to avoid dangers and find goal.

17) Solve the following crypto-arithmetic problems

SEND + MORE = MONEY

Ans 1) Assign unique digits to letter

Each letter represents a unique digits (0-9)

The goal is to find a valid assignment where sum is correct

→ S,E,N,D, M, O, R, Y are distinct digits

→ S and M cannot be 0 (since they are ~~at~~ leading digits)

FOR EDUCATIONAL USE

2) Convert to column wise addition
Arranging the numbers in column format:

SEND
+ MORE
MONEY

In expanded form,

$(1000S + 100E + 10N + D) + (1000M + 100O + 10R + E) =$
$10000M + 1000O + 100N + 10E + Y$

3) Solve step by step
1) Identify M
→ Since MONEY have five digits
→ M must be 1 (because S+M carries over)
2) Finding O
→ Since MORE contributes a carry to MONEY, the sum of SEND + MORE must be over 9999.
→ The only digit left for O that works is 0.
3) Determining S
→ Since S+M=10 and M=1 its means S=9
4) Finding other value
Using logical constraints and testing values, the correct assignment is
S=9, N=6, M=1, R=8
E=5, D=7, O=0, Y=2

FOR EDUCATIONAL USE

4) Verify

$$\begin{array}{r} 9567 \\ +\ 1085 \\ \hline 10652 \end{array}$$

The sum is 10652 which matches MONEY

$\therefore S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2$

19) Consider the following axioms:
All people who are graduating are happy
All happy people are smiling
Someone is graduating
Explain the following
1) Represent these axioms in first order predicate logic
2) Convert each formula to clause form
3) Prove that "is someone smiling?" using resolution technique
Draw the resolution tree.

Ans: 1) Representing the axioms in first order predicate logic.
$G(x)$; x is graduating
$H(x)$; x is happy
$S(x)$: x is smiling
Using these predicates the axioms can't be written as
1) Axiom 1 : "All people who are graduating are happy"
$\forall(x)\ (G(x) \rightarrow H(x))$
2) Axiom 2 ; "All Happy people are smiling"
$\forall x\ (H(x) \rightarrow S(x))$
3) Axiom 3 ; "Someone is graduating"
$\exists x\ G(x)$

Sundaram FOR EDUCATIONAL USE

2) Convert each formula to clause form:

Axiom 1 :- $\forall x\ (G(x) \rightarrow H(x))$

Convert implication

$\forall x\ (\neg G(x) \vee H(x))$

convert to clause form

$\neg G(x) \vee (H(x))$ clause 1

Axiom 2 : $\forall x\ (H(x) \rightarrow S(x))$ becomes

$\neg H(x) \vee S(x)$ clause 2

Axiom 3 : $\exists x\ G(x)$

Existential qualifier is eliminated

$G(a)$ clause 3

Step 3 :

To prove that someone is smiling, we need to show $\exists x\ S(x)$. Using proof by contradiction, we assume the negation of this standard $\sim S(x)$.

Applying resolution:

We start with known clauses

clause 1 : $\neg G(x) \vee H(x)$

clause 2 : $\neg H(x) \vee S(x)$

clause 3 : $G(a)$

Negated goal : ~~ps~~ $\neg S(a)$

Now we apply resolution step by step

1) Resolve (clause 1) with (clause 3) :

$\neg G(a) \vee H(a)$

$G(a)$

Resolution ; Remove $G(a)$ since it cancels $\neg G(a)$

FOR EDUCATIONAL USE

New clause : H(a)

2) Resolve (Clause 2) with H(a) :

$\neg H(a) \vee S(a)$

$H(a)$

Resolution : Remove H(a) since it concates $\neg H(a)$

New clause : S(a)

~~Reso~~

3) Resolve S(a) with $\neg S(a)$ (Negated goal):

$S(a)$

$\neg S(a)$

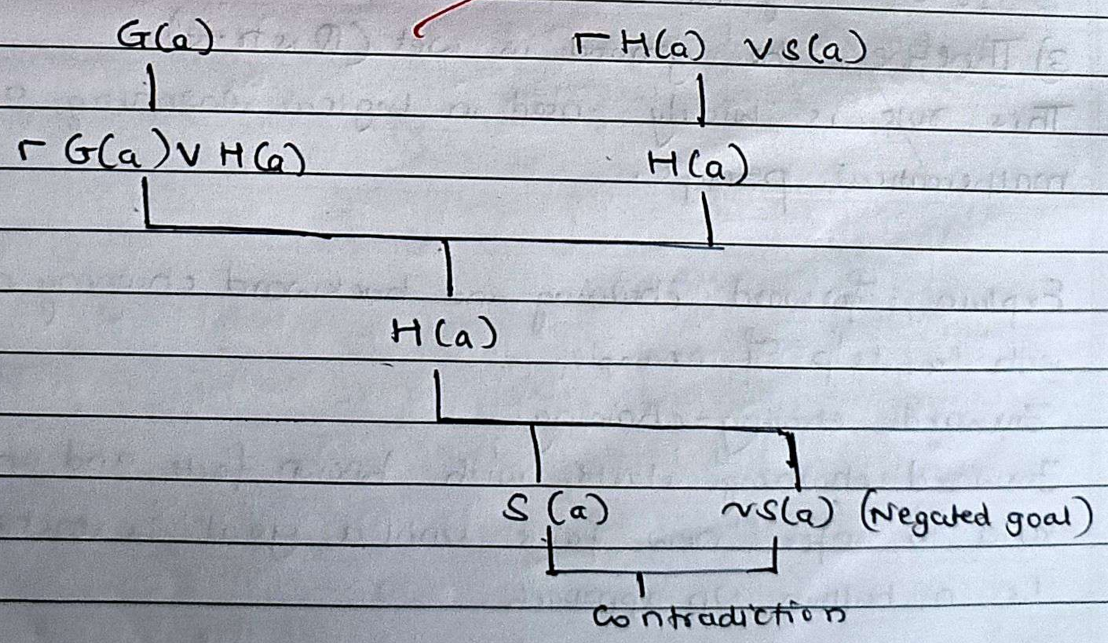
contradiction (Empty clause)

Since, we desired a contradiction, our intial assumption ($\neg S(a)$) is false, proving that :

$\exists x \, S(x)$

i.e. someone is smiling

Step 4) Resolution Tree



New clause : H(a)

2) Remove (clause 2) with H(a) :

¬H(a) V V(a)

H(a)

Resolution

20) Explain modus ponen with suitable example:

Ans) Modus Ponen is a ~~functional~~ fundamental rule of inference in logic:

It follows the structure:

1) If P, the Q (conditional statement)
2) If P is tree (promise)
3) Therefore, Q is true (conclusion)

Example:

statements :

1) If it rains, the ground will be wet (if P, then Q)
2) It is raining (P is true)
3) Therefore, the ground is wet (Q is true).

This rule is widely used in logical reasoning and mathematical proofs.

21) Explain forward chaining and backword chaining algorithm with ~~te~~ help of example.

Anss ~~Forward chaing~~ chaining:

Forward chaining starts with known facts and applies rules to infer new facts until a goal is reached. It is a bottom up-approach.

eg.

consider a medical diagnosis system

Rules:

R1: If a person has a ~~fear~~ fever and cough then they may have flue.

R2: If a person has a flu, then, they should take rest.

Facts:

1) A patient has fever.
2) The patient has a cough

Process:

1) The system checks R1: since the ~~path~~ patient has a fever and cough → They may have the flue
2) The system checks R2: since the patient has the flu → They should take rest.

conclusion: The patient should take rest.

Algorithm:

1) Initialize: start with a set of known facts in KB.
2) Match rules: Identify rules whose conditions match the known facts.
3) Apply ~~rules~~ rules: If conditions are satisfied, infer the new fact (conclusion)
4) Update KB: Add the newly inferred fact to the KB.
5) Repeat: Continue the process ~~with~~ ~~or~~ until either
   → The goal fact is derived or
   → No new facts can be inferred.

Backward Chaining:

Backward chaining starts with the goal and works backward to determine if there is evidence to support. It is a top-down approach.

Algorithm:

1) Start with the goal: Identify the target fact that needs to be proven.

2) Check if the goal is already a known fact:

→ If yes, stop (goal is achieved).

→ If no, proceed to next step.

3) Find rules that conclude the goal

→ Identify rules where the goal is the conclusion

→ Check if the rules conditions are met.

4) Verify premises:

→ If all premises are known facts, apply rule and derive goal.

→ If some premises are unknown, set them as new subgoals and repeat the process

5) Continue recursively until:

→ The goal is proven (success), or

→ No supporting facts are found (failures)

e.g. Diagnosing disease

Goal: Determine if the patient has flu

Process: 1) The system checks R1: If a person has fever and enough, then may have the flu.

2) It asks: "Does the patient have fever?" Yes

3) It asks: "Does the patient have cough?" Yes

4) Since both conditions are met, the system confirms. "The patient has flu".