Shivam R. Prajapati AIDS-1 D15C41

23/03/25

Assignment No. 1

05/05

Page No.
Date

(1) What is AI? Considering the covid-19 pandemic situation, how AI helped to evolve and renovated our way of life with different applications

Soln:- Artificial Intelligence (AI) is a branch of computer science that enables machines to perform tasks that typically require human intelligence. These tasks include learning, reasoning, problem solving, perception, language understanding and decision making.

1. AI-powered algorithms analyzed news, social media, and official reports to detect early signs of outbreaks.
2. Drug Discovery and vaccine development :- AI accelerated the development of COVID-19 treatments and vaccines by analyzing protein structures and predicting potential drug candidates.
3. Medical diagnosis :- AI powered diagnostic tools analyzed Xrays and CT scans to detect COVID-19 infections faster than traditional methods.
4. Contact Tracing and spread prediction :- AI models helped track virus spread using data from mobile apps, GPS and social media.
5. Remote Work and online learning :- AI-powered collaboration tools (Zoom, Teams) enabled remote work and virtual meetings.
6. Mental Health and wealth being :- AI chatbots like Wysa and Woebot provided mental health support to people dealing with stress and isolation

Page No.
Date

② What are AI agents terminology, explain with examples.

Soln:- An AI agent is an entity that perceives its environment through sensors and acts upon it using actuators to achieve a specific goal. AI agents operate based on decision making models, algorithm and data.

① Agent:-
An agent is any system that perceives its environment and takes actions to achieve a goal.
Eg:- A self driving car perceives traffic signals and road conditions and takes actions like stopping and accelerating.

② Environment:-
The environment is everything that surrounds an AI agent and affect its performance.
Eg:- In a chess playing AI, the chessboard, opponent, moves, the rules from the environment.

③ Perception:-
Perception is the agents ability to gather information from the environment through sensors.
Eg:- A facial recognition system perceives images through a camera.

④ Sensors:-
Sensors are the input devices that collect data from the enviroment.
Eg:- A voice assistant (like Alexa) uses a microphone

Page No.
Date

(5) Actuators :-
Actuators are components that allow the agent to take action based on decisions
Eg :- In a robot, its arm and wheels are actuators.

(6) Rational agent :-
A rational agent takes the best possible action to maximise its performance measure.
Eg :- AI stock trading system makes decisions to maximise profit.

(7) Utility function :-
A utility function measures how good agent is for achieving a goal.
Eg :- In a game AI, winning a match has a high utility score.

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

Soln :- The 8 puzzle problem is a classic problem in artificial intelligence that involves sliding tiles on a 3×3 grid to reach a goal state.

Representation of the 8 puzzle problem :-
(1) Initial state :- Any random arrangement of tiles
(2) Goal state :- A predefined arrangement (eg numbers in order)
(3) Operators :- Move the blank side (up, down, left, right)
(4) state space :- All possible tile arrangements

Page No.
Date

(5) Cost function :- Typically, each move has a cost of 1

Common heuristics for A* :-

(1) Misplaced Tile Heuristic (h_1) :- counts the number of misplaced tiles.

(2) Manhattan Distance (h_2) :- measures the sum of distances each tile is from its goal position. More accurate than misplaced tiles - heuristic.

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

Soln :- The PEAS (performance, measure, environment, actuators, sensors) descriptor is used to define the components of an agent, helping to analyze its working environment and functions.

- Performance measure (P) :- The criteria for evaluating the agent's success.
- Environment (E) :- The external surroundings where the agent operates.
- Actuators (A) :- The mechanisms through which the agent interacts with the environment
- Sensors (S) :- The device used to perceive the environment.

(1) Taxi driver

P : safety, fuel efficiency, passenger satisfaction

E : Roads, traffic signals, other vehicles, pedestrians, weather

A: steering wheel, accelerator, brakes, horn, indicators, wipers

S: GPS, cameras, LiDAR, speed sensors, fuel guage, odometer etc.

② Medical diagnosis system :-

P: Diagnosis, accuracy, speed of response, patient recovery etc.

E: Patients, symptoms, medical databases, hospital environments

A: Display screen, speaker (for communication with doctor and patients)

S: Patients history, test reports, X-rays, MRI scans, symptom inputs

③ A Music composer :-

P: Quality of generated music, creativity, audience engagement

E: Musical database, sound libraries, listener preferences

A: Music output system (MIDI, 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, airtraffic

A: Flaps, landing gear, engine & throttle, air brakes.

S: GPS, altimeter, wind sensors, gyroscopes, radar.

⑤ An essay evaluator :-

P: Accuracy in grading, fairness, grammar and colaborate detection

E: Essays, answer sheets, writing rules, academic guidelines

A: Score display, feedback generator. S: Optimal character recognition (OCR) NLP tools.

⑥ Robotic sentry gun for the kock lab :-
P: Accuracy in target detection, response-time
E: Lab premises, intruders, authorized personnel, obsta.
A: Gun turret, 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 observable, deterministic / stochastic, episodic / sequential, static / dynamic, discrete / continous, single / multi-agent)

Soln :-

ⓐ Observability :- partially observable.
The bot may not have complete information about books on shelves, customer preferences or stock updates without external input.

ⓑ Deterministic vs stochastic :- stochastic
Book availability may change due to manual sales, external purchases, or misplaced books, making environment uncertain.

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

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

ⓔ Discrete vs continous : Discrete
The bot deals with a finite set of actions (searching books, checking stock).

Page No.
Date

(4) Single Agent VS Multiagent :- Multiagent.
The bot interacts with multiple customers, bookstore staff and possibility other inventory systems.

(5)

Q.6 Differentiate Model based and utility based agent.

Soln:-

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 actions that maximize its expected utility
(3) Maintains a model of the environment (how actions affect future states).	(3) Uses a utility function to compare possible outcomes.
(4) Focuses on achieving a goal using state 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.

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

Soln:-

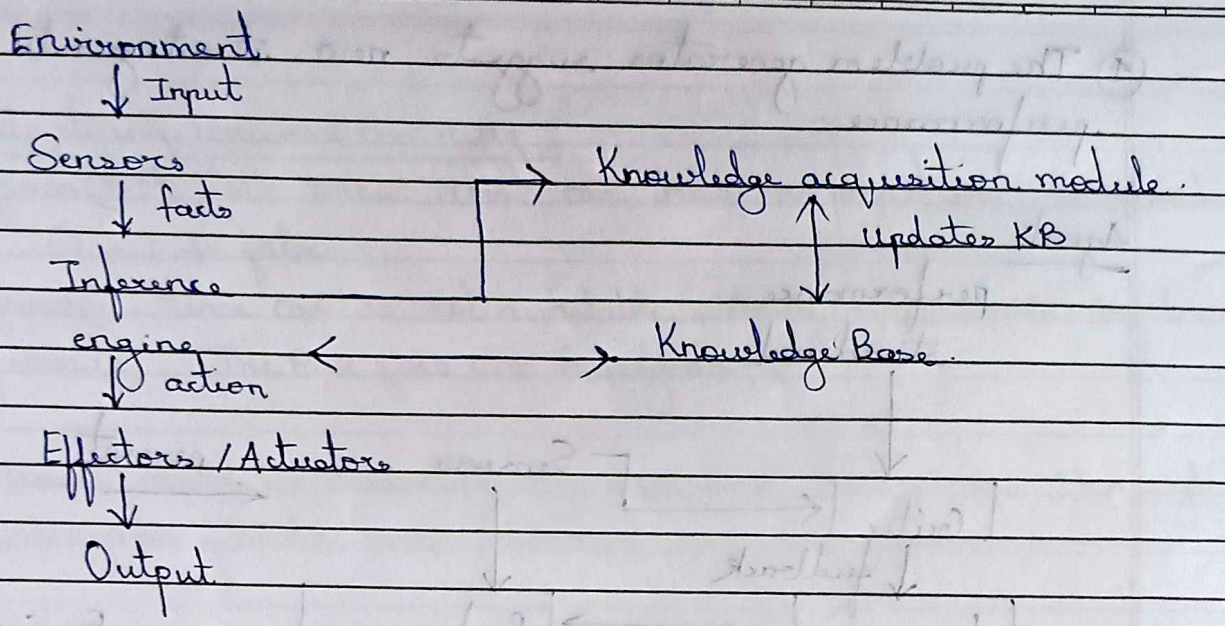
Architecture of knowledge based agent

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 derive 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 improves its performance overtime by learning from past experiences.

Components of the learning agent :-

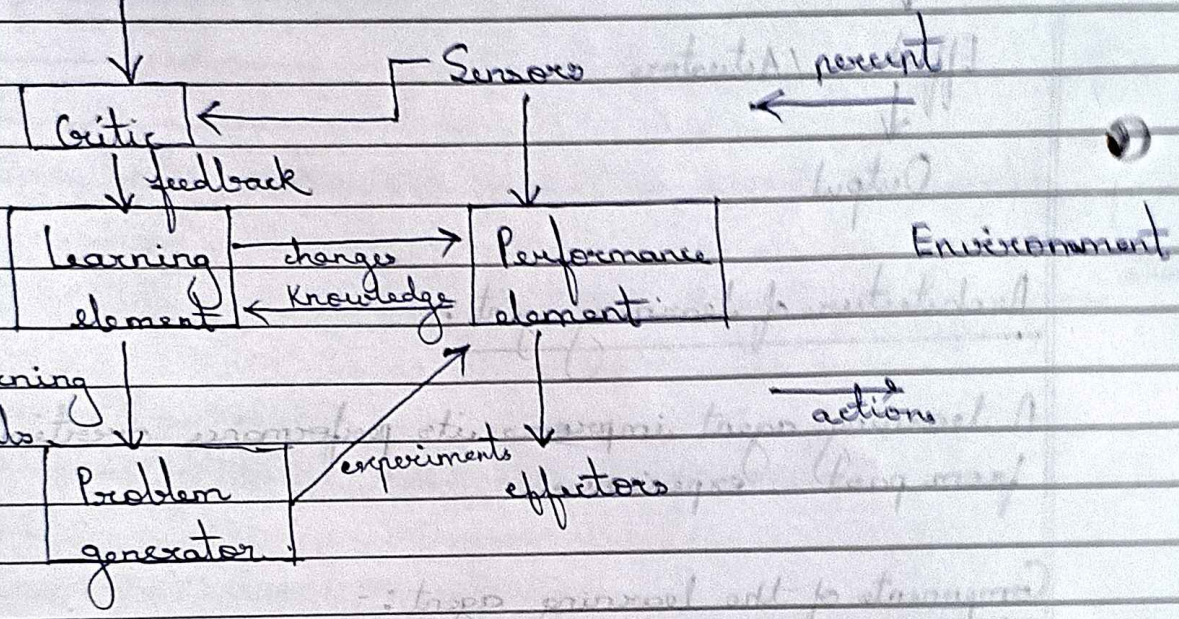
1. Learning element :- Improves the agent's knowledge by analyzing past experiences
2. Performance element :- chooses actions based on the learned knowledge.
3. Critic :- Evaluates the agent's actions by comparing outcomes with expected results
4. Problem generator :- suggests new exploratory actions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

Working process :-

1. The performance element makes decisions and takes 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



⑤ Convert the following to predicates.

a) Anita travels by car if available otherwise travels by bus.
Available(car) → Travels (Anita, car)
~ Available (car) → Travels (Anita, ~~car~~ Bus) — ①

b) Bus goes via Andheri and goregoan.
Goes via (Bus, Andheri) ∧ Goes via (Bus, Goregoan) — ②

c) Car has a puncture, so it is not available
puncture(car) → ~ Available (car) — ③

will Anita travel via Goregoan.

Applying forward chaining :

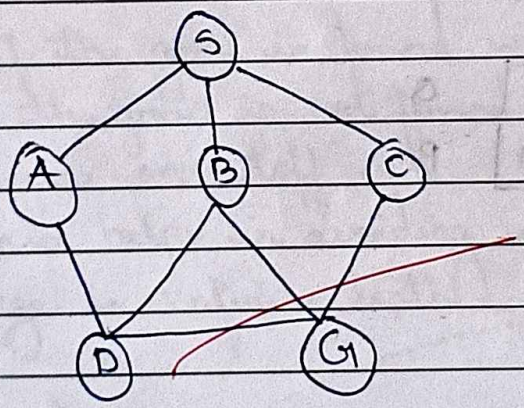
from ③ :- We know that car has a puncture, so available (car) is false.

from ① : Since car is not available, Anita will travel by bus.

from ② : The Bus goes via Goregaon.

Hence, Anita is traveling by Bus and Bus passes through Goregaon, Anita will travel via Goregaon.

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



Soln :- step ① 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.

Soln :- **Depth limited search (DLS) :-**

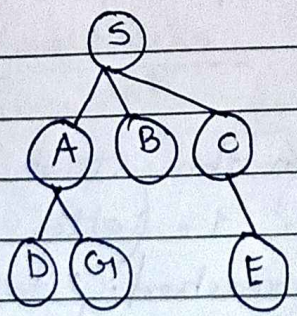
Depth limited search is a variation of DFS where we impose a depth limit to avoid going too deep into an infite or large search space.

Working :-

1. The algorithm follows a DFS strategy but limits the depth of recursion
2. If the goal is found within the limit, the search stops
3. If the goal is not found within the limit, it returns failures or cutoff
4. This helps in avoiding ~~infite~~ infinite loops 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



- If the depth limit is 1, we only explore S → A, B, C but cannot reach G.
- If the depth limit of 2, we explore S → A → D, G.

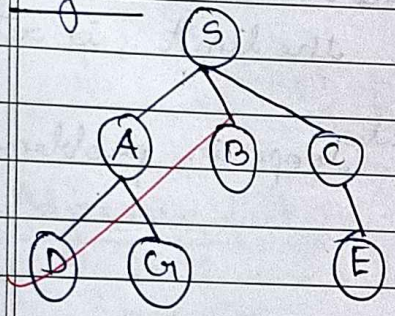
Iterative deepening search (IDS):-

Iterative deepening search (IDS) combines 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.

Eg:-



- 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 find G.

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

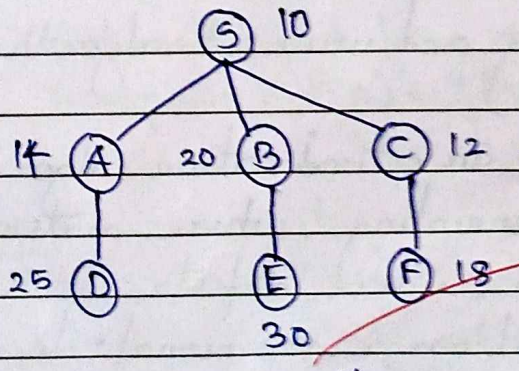
Soln:-

Hill Climbing is an optimization algorithm that continuously moves towards higher-valued states i.e better solutions until a peak (local maximum) is reached. It is a greedy algorithm that 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 or predefined goal state is reached.

Eg:- 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 = 20)
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 the 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.

Soln :- Simulated annealing (SA) is an optimization algorithm inspired by metallurgical annealing, where materials are heated and then cooled to remove defects. It helps escape local maxima by allowing occasional downward moves 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.

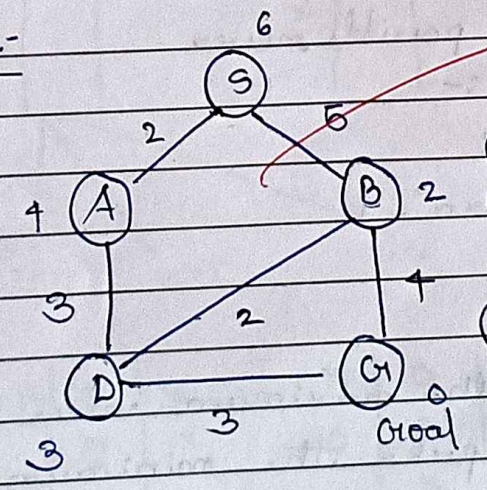
(5) Reduce the temperatures gradually
(6) Repeat until the temperature is very low or a stopping condition is met.

Q.13 Explain A* with an example.

Soln :- 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~~ 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)

Eg :-



Steps
(1) Initialize open list
(2) expand node with lowest f(n)
(3) update g(n), h(n) and f(n) for neighbouring
(4) Repeat until goal node is reached

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

(14) Explain minimax algorithm and draw game tree for Tic-Tac-Toe Game

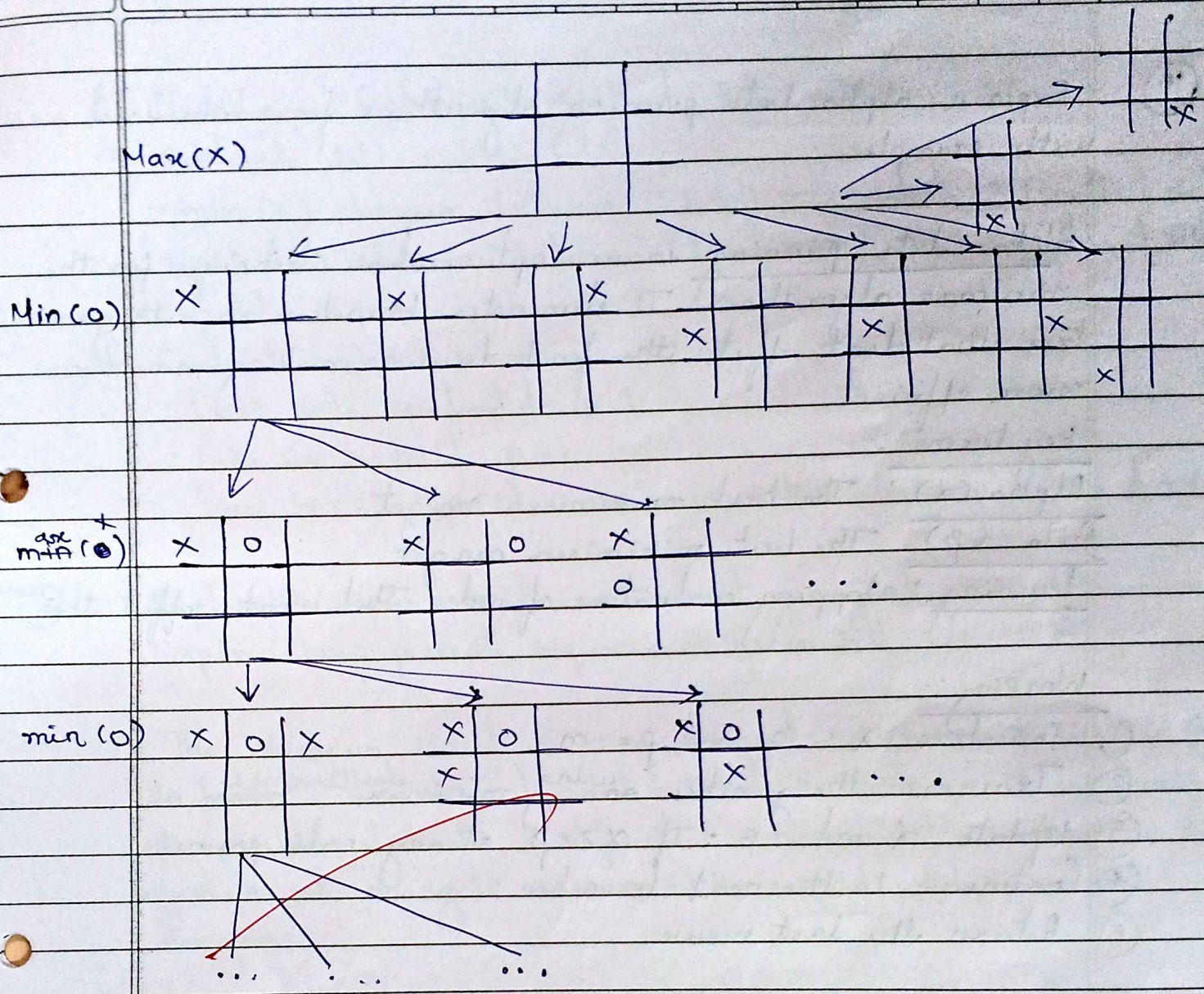
Soln:- Minimax is a decision making algorithm used in two-player games like Tic-Tac-Toe, chess, and connect four. It helps in finding the best possible move for a player by assuming that the opponent also plays optimally. The algorithm alternates between maximizing (for AI) and minimizing (for opponent)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

+1 → AI wins
-1 → opponent wins
0 → Draw

Algorithm steps

1. Generate game tree for all possible moves.
2. Evaluates Terminal states :-
   - If AI wins, return +1
   - If opponent wins, return -1
   - If draw, return 0
3. Backpropagate values
   - Max player (AI) picks the maximum value.
   - Min player (opponent) picks the minimum value.
4. Select the best move at the root level.

Game tree :-



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

Sol:- 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

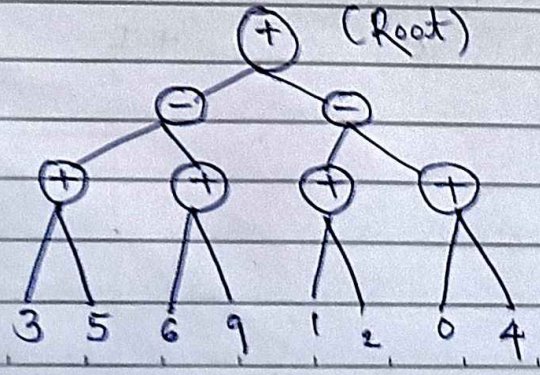
Pruning : stopping evaluation of nodes that won't affect the result

Working :-

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

Example :-

Consider the following gametree where max(⊙) wants to maximise the score, and min(⊙) wants to minimise it.



(1) Initialize : Alpha (α) = $-\infty$ and, Beta (β) = $+\infty$

(2) Evaluate left subtree
- min (O) chooses between (3, 5) ⟶ selects 3 (smallest value)
- max (O) chooses between (3, 6) ⟶ selects 6 (largest value)
- update $\alpha = 6$

(3) Evaluate right subtree
- first node evaluated is 1
- beta is updated $\beta = 1$
- Since $\alpha(6) \geq \beta(1)$, we prune (skip checking 2, 0 and 4)

(16) Explain Wumpus world environment giving it PEAS description. Explain how percept sequence is generated.

Soln :- 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 dangers.

Structure of wumpus world :-
(1) Grid Based (4×4 or larger)
(2) Contains
a) Gold (Goal pick it up)
b) Wumpus (A monster that kills the agent)
c) pits (If agent falls, it dies)
d) Walls (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 the arrows.

E: A 4×4 grid with the agent, wumpus, pits, gold and walls.

A: Move (up, down, left, right), grab (gold), shoot (arrow), climb (exit)

S: Breeze, stench, glitter, bump, scream.

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

Eg:-

1. Agent starts at (1,1) → percepts = (Breeze) (pit around)
2. Moves to (1,2) → percepts = (Breeze, stench) (pit = wumpus)
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.

Soln:-

Step①: Assign unique digits to letters.
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 leading digits.

Step ②:- Convert to column wise Addition.
Arranging the numbers in column format:

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

In expanded form,

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

Step ③:- Solve step by step.

① Identifying 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 it means S = 9

④ Finding other values.
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

Step ④:- Verify

```
 9567
+1085
-----
10652
```

The sum is 10652, which matches MONEY.

∴ 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.

Soln :- Step① :- 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 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"
$\exists x G(x)$

Step② :- Convert each Formula to Clause Form.
Axiom 1 :- $\forall x (G(x) \rightarrow H(x))$
Convert implication
$\forall x (\sim G(x) \vee H(x))$
Convert to clause form
$\sim G(x) \vee (H(x))$ (clause 1

Axiom 2: $\forall x (H(x) \rightarrow S(x))$ becomes
$\sim H(x) \vee S(x)$ Clause 2.

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

Existential quantifier 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 statement $\sim S(x)$

Applying resolution :-

We start with known clauses

Clause 1 : $\sim G(x) \vee H(x)$

Clause 2 : $\sim H(x) \vee S(x)$

Clause 3 : $G(a)$

Negated Goal : $\sim S(a)$

Now we apply resolution step by step :

① Resolve (Clause 1) with (Clause 3) :

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

$G(a)$

Resolution : Remove $G(a)$ since it cancels $\sim G(a)$

New Clause : $H(a)$

② Resolve (Clause 2) with $H(a)$:

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

$H(a)$

Resolution : Remove $H(a)$ Since it cancels $\sim H(a)$

New clause : $S(a)$

③ Resolve S(a) with ~S(a) (Negated Goal):

S(a)

~S(a)

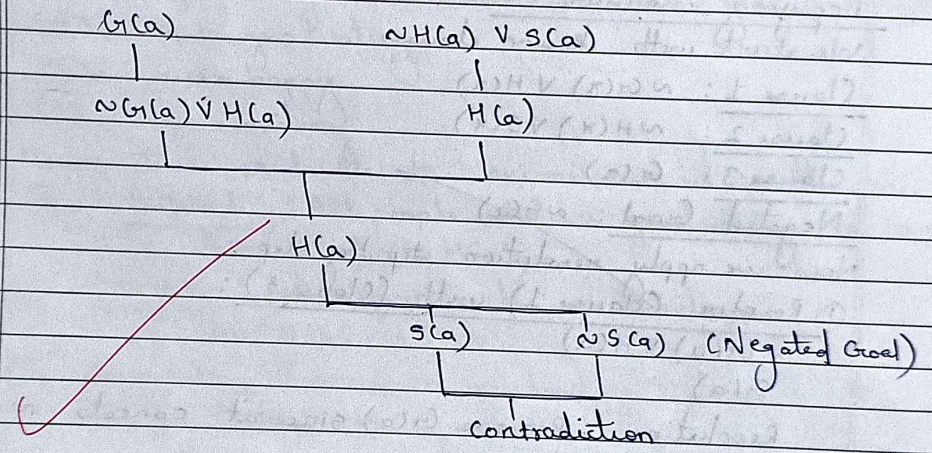
Contradiction (Empty clause)

Since we derived a contradiction, our initial assumption (~S(a)) is false, proving that:

$\exists x\, S(x)$

i.e Someone is smiling

Step ④:- Resolution Tree



Q.20 Explain Modus Ponen with suitable example.

Soln:- Modus Ponens is a fundamental rule of inference in logic. It follows the structure:

① If P, then Q (Conditional statement)

② If P is true (Promise)

③ Therefore, Q is True (Conclusion:

Example :-

statement :-

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.

Q.21. Explain forward chaining and backward chaining algorithm with the help of example

Soln :- forward 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 fever and cough then they may have the flue.
R2 : If a person has a flu, then they should take rest

Facts :
① A patient has a fever
② The patient has a cough.

Process:-

① The system checks R1: Since the patient has a fever and cough → They may have the flu
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 the KB
② Match rules: Identify rules whose conditions match the known facts.
③ Apply rules:- If a conditions are satisfied, infer the new fact (conclusion)
④ Update KB: Add the newly inferred fact to the KB.
⑤ Repeat: Continue the process 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 rule's conditions are met.

④ Verify premises:-
- If all premises are known facts, apply rule and derive the goal.
- If some premises are unknown, set them as new sub-goals and repeat the process.

⑤ Continue recursively until:
- The goal is proven (success), or
- No supporting facts are found (failure)

Example: Diagnosing disease

Rules:
R1: If a person has a fever and cough, then they may have the flu.
R2: If a person has the flu, then they should take rest
R3: If a person has a sore throat and fever, then they may have a throat infection

Facts:
- A patient has a fever
- The patient has a cough.

Goal: Determine if the patient has the flu.

Process:

① The system checks R1: "If a person has fever and cough, then they may have the flu".

② It asks: "Does the patient have a fever?"
→ Yes.

DISC 41

③ It asks: "Does the patient have a cough?" → Yes.
④ Since both conditions are met, the system confirms "The patient has the flu".

***