Name: Sahil Ramnathgari Roll no: 42 Class: DISC

AIDS Assignment

Q1) What is AI? Considering the COVID-19 pandemic situation, how did AI help to survive & renovate our way of life with different applications?

⇒ AI is the replication of human intelligence in machines, enabling them to learn, reason, solve problems, & make decisions without direct human interventions.
AI played a vital role in managing the COVID-19 crisis by detecting outbreaks early through data analysis, accelerating drug & vaccine development, automating medical diagnoses with imaging & predictive models, supporting telemedicine via AI chatbots, optimizing supply chains for essential goods, & enhancing remote work & online education through AI-driven automation, making daily life more adaptable & efficient.

Q2) What are AI agents? terminology? Explain with examples.

⇒ AI agents are systems that perceive their environment, process information, and take actions to achieve specific goals. Key terminologies include:

1) Agent: An entity that interacts with the environment
2) Environment: The external system in which the agent operates.
3) Perception: Data collected from sensors.
4) Actuators: Components that execute actions
5) Rationality: The ability to make optimal decision based on available information
6) Autonomy: The degree of independence an agent has

Q3) How is the AI technique used to solve the 8-puzzle problem?

⇒ The 8-puzzle is solved using AI search techniques that explore possible moves to reach the goal state. Common AI techniques include:

1) Uniformed Search:
- Breadth-First Search (BFS): Explores all possible moves level by level but is inefficient for large problems.
- Depth-First Search (DFS): Explores one path deeply before backtracking but may get stuck in loops

2) Informed search (Heuristic-Based)
- Best-First Search (Greedy Algorithm): selects moves based on heuristic values like misplaced tiles.

3) A*: Uses the heuristic function $f(n) = g(n) + h(n)$, where $g(n)$ is the cost to reach the current state & $h(n)$ is the estimated cost to the goal.

Q4) PEAS descriptors for given AI 8 agents.

⇒ 1) Taxi Driver:
Performance Measure: - Safety, speed, Customer satisfaction
Environment :- Roads, traffic, pedestrains, weather
Actors Acuators:- Steering, acceleration, brakes & signals
Sensors: GPS, cameras, Speedometer
2) Medical Diagnosis System:
Performance measure: Accuracy of diagnosis, patient recovery rate.
Environment: Patient data, medical reports, symptoms
Actuators: - Display diagonis, prescribe medication.
Sensors:- Patient input, test result, doctor's note.
3) A music Composer:
Performance Measure: Music quality, originality.
Environment: Musical genres, user preference.
Actuators: Generating notes, melodies, instrument selection
Sensors:- User feedback, music databases, emotional tone analysis
4) A Aircraft Autolander.
Performance Measure: Grammar accuracy, relevance.
Environment: Weather Conditions, runway, air traffic
Actuators: Score assignment, grammar suggestion
Sensors: Text input, grammar & Semantic analyzers.

Q5. Categorize a shopping bot for an offline bookstore according to each of the six dimensions

⇒ A shopping bot for an offline bookstore can be categorized according to the six AI environment dimensions as follows:-

1) Observability: Partially Observable :- The bot may not have full knowledge of stock availability customer preferences, or real-time changes in the bookstore
2) Deterministic vs Stochastic: stochastic:- Book availability, customer behaviour, & price changes introduce randomness, making the environment unpredictable
3) Episodic vs sequential: Sequential : Each decision affects future interactions, meaning past actions influence subsequent outcomes.
4) Static VS Dynamic : Dynamic:- The environment can change as books sell out, customers move, or bookstore policies update while the bot is making decisions.
5) Discrete vs Continuous : Discrete. The bot operates with distinct choices, such as recommending a book, checking stock or processing a purchase.
6) Single -Agent vs Multi Agent: Multi Agent:- The bot interacts with customers, bookstore staff & possibly other AI systems, making it a multi-agent system.

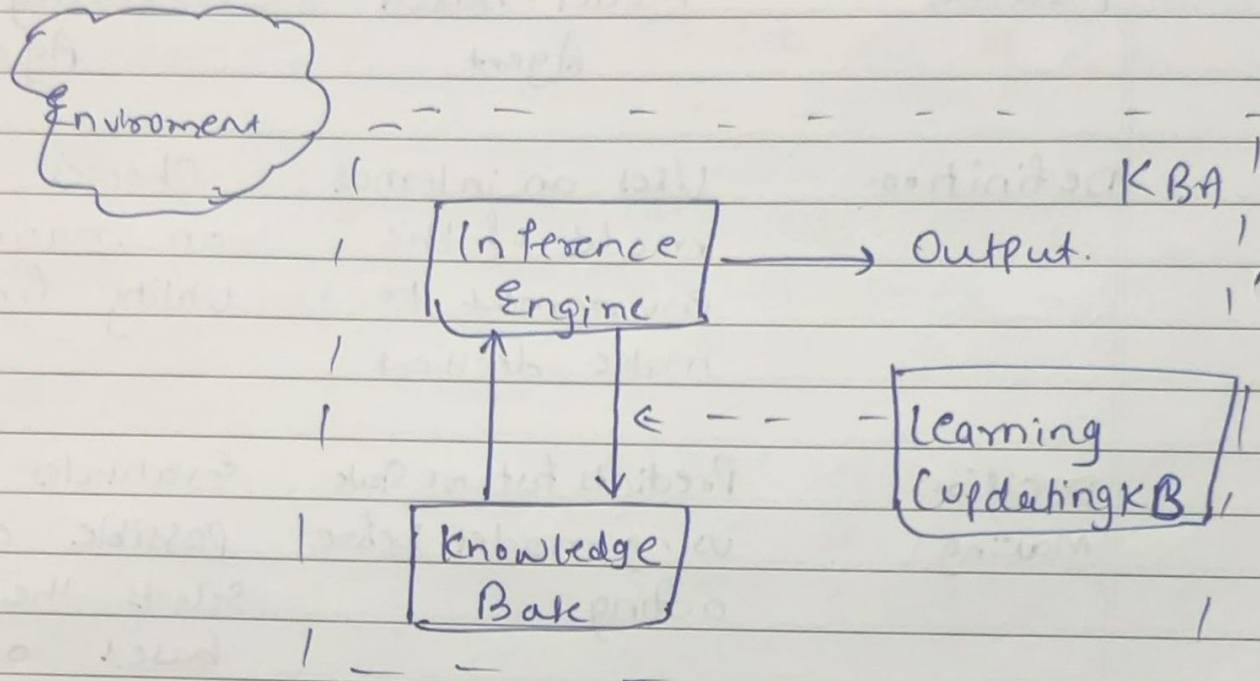
Q6) Differentiate Model based & Utility based Agent.

⇒

Feature	Model-based Agent	Utility-based Agent
Definition	Uses an internal model of the environment to make decisions	Chooses action based on maximizing a utility function
Decision Making	Predicts future state using model before acting	Evaluates multiple possible actions & selects the best based on utility
Goal Oriented	Yes, but focused on how the environment works.	Yes. but aims to achieve the best possible outcome
Optimality	May not always take the best possible action, only a feasible one	Always selects the action that provides the highest benefits.
Example	A self-driving car using a traffic model to predict congestion	A stock trading AI selecting the trade with the highest expected profit.

Q2) Explain the architecture of a knowledge based agent & learning Agent

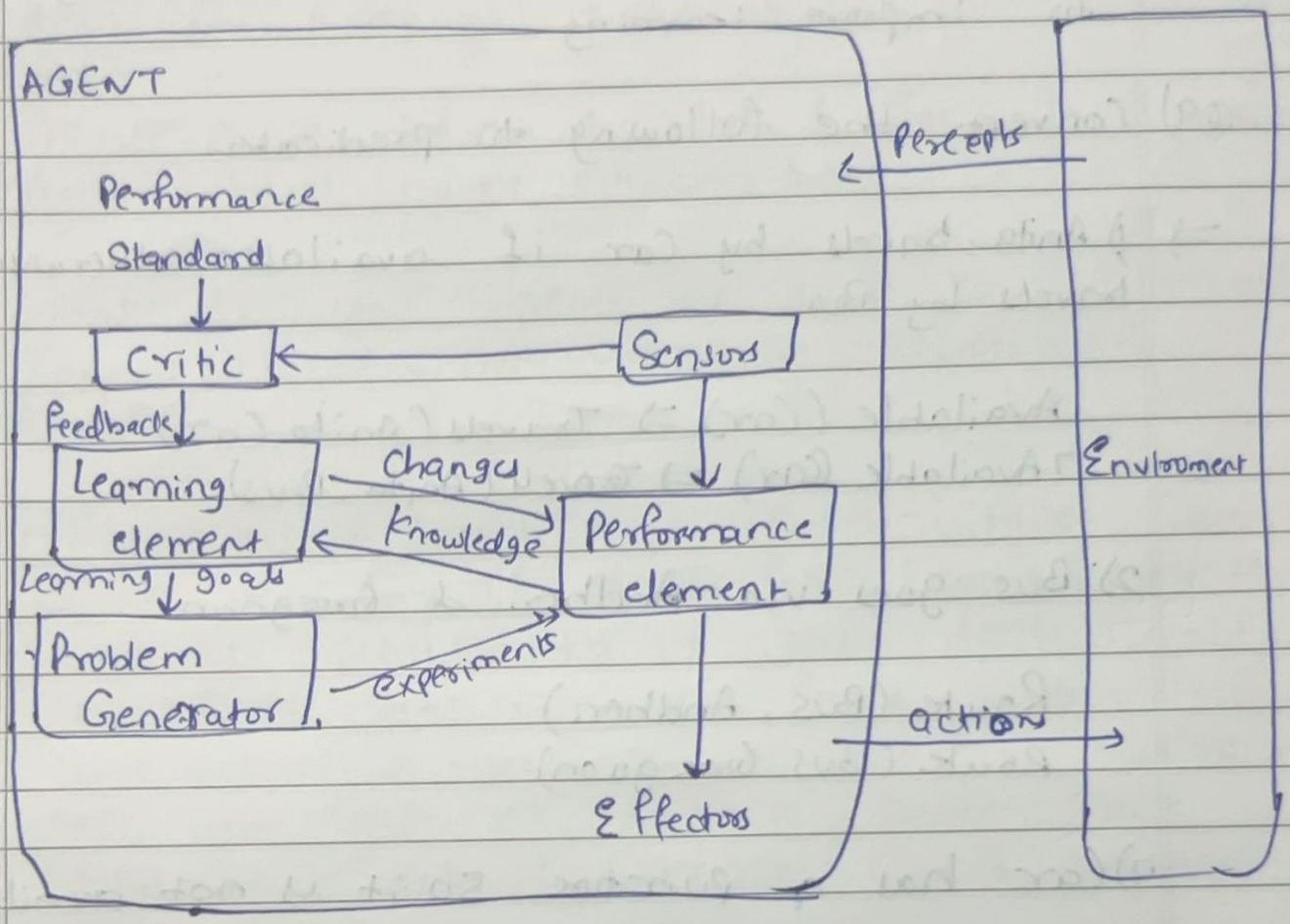
→)



A Knowledge Based Agent (KBA) uses stored knowledge to make decisions & reason about the environment. Its architecture consists of

1) Knowledge Base (KB): stores facts, rules & background knowledge knowledge in a structured format.
2) Inference Engine: Applies logical reasoning to derive conclusions from the knowledge base.
3) Preception Module: Gathers inputs from sensors or external sources.
4) Action Module: Executes action based on derived conclusion
5) Learning Mechanism: Updates the knowledge base with new information.

Architecture of Learning Agent.



A Learning Agent improves its performance over time by learning from past experience. It has four main components.

1) Learning Element: Learns from interaction and updates knowledge.
2) Performance Element: Uses the learned knowledge to make decisions.
3) Critic: Evaluates the agent's performance by comparing actions with desired outcomes.
4) ~~Problem~~

4) Problem statement: Suggests new exploratory actions to improve learning

Q9) Convert the following to predicates.

⇒ 1) Anita travels by car if available, otherwise, she travels by Bus.

Available (Car) ⇒ Travels (Anita, Car)
¬Available (Car) ⇒ Travels (Anita, Bus)

2) Bus goes via Andheri & Goregaon

Route (Bus, Andheri)
Route (Bus, Goregaon)

3) Car has a puncture, so it is not available

Puncture (Car) ⇒ ¬Available (Car)
Given: Puncture (Car)
Therefore ¬Available (Car)

Forward Reasoning:
1) From (3) ¬Available (car) \neg Available (Car) ¬Available (car)
2) From (1) ¬Available (car) ⇒ Travels (Anita, Bus) \neg Available (Car) \ Rightarrow Travels (Anita Bus) ¬Available (car).
⇒ Travels (Anita, Bus), so Anita must travel by Bus.

3) From (2), Route (Bus, Goregaon) Route (Bus, Goregaon) Route (Bus, Goregaon), meaning the Bus travels via Goregaon

4) Since Anita travels by Bus, & goes via Goregaon Anita will travel via Goregaon

Q11 what do you mean by depth limited search? Explain Iterative ~~Dept~~ Deepening search with example.

⇒ Depth-Limited Search (DLS):- Depth-Limited Search is a variation of Depth-first Search where the search is restricted to a specific depth limit LLL. If a goal is not found within the limit, the search terminates. It prevents infinite loops in deep or infinite ~~space~~ State spaces but risks failing to find a solution if LLL is too low

Iterative Deepening Search (IDS): Iterative Deepening Search combines the space efficiency of DFS & completeness of BFS by repeatedly running DLS with increasing depth limits $L = 0, 1, 2, \ldots$ $L = 0, 1, 2, \ldots$ dots $L = 0, 1, 2 \ldots$ until the goal is found.

Q12) Explain Hill climbing & its drawbacks in detail with example. Also state limitations

⇒ Hill climbing Algorithm:- Hill climbing is a local search algorithm that continuously moves towards the best neighboring state with a a higher heuristic value, aiming to reach a global optimum

Steps:-

1) Start from an initial state.
2) Evaulate neighboring states & move to the one with the highest heuristic value
3) Repeat until no better neighbor exists.

Example:

Imagine a mountain climbing scenario where a hiker moves uphill based on the steepest slope. If they reach a peak that is not the highest, they might get stuck

Drawbacks of Hill climbing

1) Local Maximum:-

The algorithm may stop at a peak that is not the global optimum

Example:- A small hill before a taller mountain.

2) Plateau Problem:-
A flat region where all neighboring states have the same heuristic value, causing the search to halt.

3) Ridges:-
The algorithm may fail to climb diagonally when only direct moves are considered
Example:- climbing a staircase with missing steps.

4) No Backtracking:-
Once it ~~on~~ moves forward, it cannot recover from a bad decision
Example:- Choosing the wrong pin in a maze with no way back.

Q13) Explain simulated annealing & write its algorithm

⇒ Simulated Annealing is a probabilistic ~~optimo~~ optimization algorithm inspired by the annealing process in metallurgy. It helps in finding a global optimum by allowing occasional moves to worse situation to escape local optima.

Algorithm:
1) Initialize the current state & temperature T.
2) Repeat until stopping condition

- Select a random neighbor state.
- Compute energy difference $\Delta E = E_{new} - E_{current}$.
- If $\Delta E < 0$, accept the new state
- Else, accept it with the probability $e^{-\Delta E/T}$
- Reduce the Temperature.

3) Return the best-found solution

Thus method avoids getting trapped in local maxima by probabilistically exploring worse solutions early on. (9)

Q14) Explain A Algorithm with an example.

⇒ The A Algorithm is an informed search algorithm that finds the optimal path by considering both the cost to reach a node ($g(n)$) and the estimated cost to the goal ($h(n)$), using the formula.

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

where

$g(n)$ = actual cost from start node to n
$h(n)$ = estimated cost from n to goal (heuristic)

Example:

In a graph search problem, if S is the start & G is the goal, A* expands nodes based on the lowest f(n) value, ensuring the shortest path is found efficiently.

Q15 Explain Minimax Algorithm & draw game tree for Tic Tac - Toe.

⇒ The Minimax Algorithm is used in adversarial search to determine the optimal move by assuming both players play optimally

Maximizer: (e.g. X) tries to get the highest score
Minimizer (eg O) tries to reduce the score

Algorithm:

1) Generate the game tree up to a depth limit
2) Assign heuristic values to leaf nodes
3) Back propagate values:
   Maximizer picks the maximum values.
   Minimizer picks the minimum values.
4) The root node selects the best move based on propagated values.

Game Tree: A minimax tree for Tic-Tac-Toe would show all possible board states, evaluating the best move to each step

Q 16) Explain Alpha-Beta Pruning Algorithm for adversarial search with an example

⇒ Alpha - Beta Pruning optimized Minimax by skipping unnecessary branches, reducing computations. It uses two values:

Alpha (α) : Best score Maximizer can achieve.
Beta (β) : Best score Minimizer can achieve.

Algorithm:

1) Perform Minimax search
2) Prune branches where:
   - If Maximizer's best (α) ≥ Minimizer best (β), futher evaluation is skipped.
   - If Minimizer's best (β) ≤ Maximizer best (α), futher evaluation is skipped.
3) This reduces time complexity without affecting the final decision.

Example: In a game tree, if a branch has already provided a worse outcome than the best known move, it is ignored to save time.

Q17) Explain WUMPUS World Environment giving its PEAS description. Explain how percept sequence is generated?

⇒ The wumpus world is a grid-based AI environment where an agent explores a cave while avoiding hazards like pits & the wumpus monster.

PEAS Descriptor:

- Performance Measure: Reaching the gold safely, minimizing steps.
- Environment: A 4x4 grid with pits, & wumpus.
- Actuators: Move forward, turn, grab, shoot, climb
- Sensors: Perceive stench (wumpus nearby), breeze (pit is nearby), glitter (gold found)

Percept Sequence Generation

1) Agent starts at (1,1), Sensing its surroundings
2) if stench is detected, the wumpus is nearby
3) if breeze is detected, a pit is nearby
4) if agent infers safe path & navigates towards the gold while avoiding hazards.

Q18 Solve the following Crypto-~~algorithm~~ arithmetic problem: SEND + MORE = MONEY.

⇒ In this problem, each letter represents a unique digit (0-9). we solve it by assigning values:

Using constraints from column-wise addition, the solution is

$$\begin{array}{r} 9\,5\,6\,7 \\ +\quad 1\,0\,8\,5 \\ \hline 1\,0\,6\,5\,2 \end{array}$$

Thus, the correct letter-digit mapping is

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

Q19 Consider the following ~~axioms~~ axioms:

1) Represent these axioms in First order predicate Logic (FOPL):

1) $\forall x (Graduating(x) \rightarrow Happy(x))$
2) $\forall x (Happy(x) \rightarrow Smiling(x))$
3) $\exists x (Graduating(x))$

2) Convert each formula to clause form

1) $\neg Graduating(x) \vee Happy(x)$
2) $\neg Happy(x) \vee Smiling(x)$
3) Graduating(A) (assuming 'A' is the individual who is graduating)

3) Prove that "Is someone smiling?" using resolution technique.

From 3: Graduating(A)
Using 1: Happy(A) (Modus Ponens)
Using 2: Smiling(A)

Thus, someone is smiling

Resolution Tree:

1) Graduating(A) Given)
2) Graduating(A) $\rightarrow$ Happy(A) $\rightarrow$ Happy(A)
3) Happy A) $\rightarrow$ Smiling(A) $\rightarrow$ Smiling(A)

Q 20) Explain Modus Ponens with a suitable example.

⇒ Modus Ponens (Law of Detachment) is a rule of inference stating:

Example:

1) If it rains, the ground gets ~~de~~ wet (Premise: Rain → Wet Ground).
2) It is raining (Premise: Rain)
3) Conclusion: The ground is wet (Appling Modus Ponens)

Q 21) Explain Forward chaining & Backward chaining algorithm with an example.

⇒ Forward chaining:

Data - driven inference: Starts from known facts and applies rules to reach a goal
Used in expert systems (~~eg.~~ (eg. medical diagonsis)

Example:

1) Fact: "Sore throat"
2) Rule: "If Sore throat → infection"
3) New fact: "Infection
4) Rule: "if infection → need antibiotics"
5) ~~New~~ Conclusion: "Need antibiotics"

Backward ~~fa~~ Chaining:

- Goal - driven interference: Starts from the goal & works backward to find supporting facts.
- Used in AI reasoning & theorem proving

Example:

Goal: Does the patient need antibiotics?
1) Check: Does the patient have an infection?
2) Check: Does the patient have a sore throat?
3) if both hold, ~~Continue~~ conclude "need antibiotics"

Thu reduces ~~AI~~ unnecessary computations by on exploring relevant facts.