Name: Rakshit K Sharma
DISC
49

05/05

Assignment - 1

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

→ Artifical Intelligence is a branch of computer science that enables machines to mimic human intelligence, including problem solving, decision making and learning from experience. AI systems utilize algorithms, data, and computational power to perform tasks that typically require human intelligence. During COVID-19 pandemic, AI played a crucial role in helping people, buissenesses and governments survive and adapt to new challenges. It helped in

1. Healthcare & Medical Innovations
2. AI in bussiness and remote works
3. AI in education and E learning
4. AI in supply chain and logistics
5. AI in Mental health and Social well being

Q2. What are AI agents terminology, explain with examples.

→ 1. Environment
Everything external to agent with which it interacts.
Eg. Chatbot - users
2. Sensors
Used by agent to perceive the environment
Eg. Camera

3. Actuators

Allow agent to take action or interact with environment

4. Percept

Input received by agent's sensors

Eg. Camera images

5. Agent Function

Maps percept received to actions to be performed

6. Performance

~~How~~ Analyzes how well the agent has performed given the circumstances

Eg. Accuracy

Q3. How AI technique is used to solve 8 puzzle problem?

→ The 8 puzzle problem consists of a 3x3 grid with 8 numbered tiles and one empty space. The goal is to move the tiles to reach a predefined arrangement using valid moves (up, down, left, right)

Initial State

1	8	3
4	6	2
7	5	

Final State

1	2	3
4	5	6
7	8	

AI can use both uninformed or informed search to find a solution.

1. Uninformed Search

These search space without prior or additional knowledge of the best path.

→ Breadth First Search

- Explores all possible moves level by level

- Guarantees the shortest solution but is memory intensive

→ Deapth First Search

- Explores one path deeply before backtracking
- May not find the optimal solution ad can get stuck in loop.

At every state, AI will create new nodes for every state. If tiles are moved either up, down, left, right depending on the different situations. Every step will create a new state ad these states can be applied to an algorithm to find the goal.

Q4. what is PEAS description? Give PEAS description for following:

→ PEAS is a framework to describe the structure of an intelligent agent in an AI system. PEAS stand for

→ Performance Measure : Defines how the success of agent is measured

→ Environment : The surrounding in which agent operates.

→ Actuators : The components that allow agent to take action

→ Sensors : The components to perceive environment.

PEAS description for

1. Taxi Driver AI

P : Safety, feul efficiency, travel time, legal compliance

E : Roads, traffic, pedestrians, other vehicles

A : Steering, accelerator, brakes, horn

S : GPS, cameras, speedameter, feul gauge

2. Medical Diagnosis System
P: Accuracy, speed, recovery rate
E: patient data, medical records, symptoms
A: Display screen, reports, ~~tre~~ recommendations
S: User input, medical imaging
3. AI Music Composer
P: Quality, user preference, harmonics, pitch
E: Music database, feedback, instruments
A: Music file, sheet generation, MIDI file
S: genre, music patterns, instrument sound libraries
4. Aircraft autolander
P: Safe landing, smooth touchdown, taxi to alignment runway
E: Runway, weather, altitude
A: landing gear, flaps, thrusters, rudder
S: Altimeter, GPS, gyroscope, airspeed
5. Essay Evaluator
P: Accuracy, grammatical corrections, coherence
E: Submitted essays, grading
A: Score output, feedback
S: Text input, NLP, images.
6. Robotic Sentry Gun for the Keck lab
P: Accuracy in target identification
E: Laboratory surroundings, intruders
A: Gun turrent, warning alarms
S: Motion sensors, infrared cameras.

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

→ Observability – Partially observable
Deterministic/Stochastic – Stochastic
Episodic/Sequential – Sequential
Static/Dynamic – Dynamic
Discrete/Continuous – Discrete
Single/Multi Agent – Multi Agent

Q6. Differentiate Model based and utility based agent.

Model Based Agent	Utility Based Agent
→ Uses an internal model of the environment to make decision	→ uses a utility function to evaluate and select best option
→ Choose actions based on predicted future states	→ Choose actions based on maxi. utility
→ Focuses on state representation	→ Focuses on optimizing
→ Low complexity	→ High complexity
→ Eg. Self driving car on predicted road layouts	→ Self driving car finding fastest route

Q7. Explain the architecture of a knowledge based agent and learning agent.

→ A knowledge based agent is an intelligent system that stores, retrieves and applies knowledge to make decisions.

Components of Knowledge Based agent

1. Knowledge Base

Stores facts, rules ad heuristics. about the environment

2. Inference Engine

Uses logical reasoning to derive new knowledge from existing facts

3. Perception module

Gathers data from sensors or user inputs

4. Query processing ad decision making unit

Uses KB ad inference rules to generate responses or actions

5. Action Execution

Perform the choosen action based on the inference result

A learning agent improves its performance over time by learning from past experiences and adapting its behaviour.

Components of Learning Agent

1. Learning Element

Improves the agent's knowledge by updating models or rules based on new experience

2. Performance Element

Executes actions based on the agent's current knowledge

3. Critic

Evaluates agent performance ad provides feedback.

4. Problem Generator

Suggets exploratory actions to improve learning

Q9. Convert the following to predicates

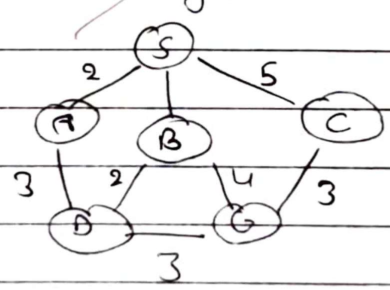
a. Anita travels by car if available otherwise travels by bus

b. Bus goes via Andheri and Goregaon.

c. Car has puncture so it is not available
Will Anita travel via Goregaon? Use forward reasoning

→ Predicate:
travels (Anita, vehicle)
available (car) → travels (Anita, car)
¬available (car) → travels (Anita, bus)
via (Bus, Andheri) , via (Bus, Goregaon
puncture (car) → ¬available (car)
Forward reasoning:
From 4th we derive
¬available(car)
Using predicate 3 we conclude
travels (Anita, bus)
Since via (Bus, Goregaon) is given and Anita is travelling by Bus
∴ Anita will travel via Goregaon

Q10. Find the route from S to G using BFS



S → A (2)
S → B (5)
S → C (5)
A → D (3)
B → D (2)
B → G (4)
C → G (3)
D → G (3)

Using best first search
1. Expand node S
Priority Queue : (A,2) (B,5) (C,5)
Visited : S
2. Take minimum cost, expand A
Priority Queue : (B,5) (C,5) (D,3)
Visited : S, A
3. Expand D
Priority Queue : (B,5) (C,5) (G,3)
Visited : S, A, D
4. ~~Expand~~ G is the goal node
Visited : S, A, D, G

∴ path from S to G
S → A → D → G
Path cost = 2 + 3 + 3 = 8

Q11. What do you mean by depth limited search ? Explain iterative Deepening search with example.

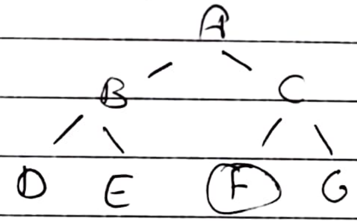
→ Depth limited search is a variation of Depth first search that limits the depth of recursion to a predefined value h. It prevents the algorithm from going too deep in a infinite or large search tree.
Key features of DLS
- Solves the infinite depth problem of DFS
- Uses a depth limit h to control recursion
- Does not gaurantee a shortest path if the solution exists at a greater depth than h.

Iterative deepining search is a combination of Depth first search and Breadth first search. It runs multiple Depth limited searches with increasing depth limits until the goal is found.

Features:

- Combines DFS space efficiency with BFS completeness
- Finds the shortest path like BFS but avoids large memory usage.
- Useful in problems where the depth of the solution is unknown.

Eg.



h = 0 → Only explores A (goal not found)

h = 1 → explores A, B, C (goal not found)

h = 2 → Explores A, B, C, D, E, F (goal found)

Q12. Explain Hill climbing and its drawbacks in detail with example. Also state limitations of steepest ascent hill climbing.

→ Hill climbing is an informed search algorithm that continuously moves towards the direction of increasing value (or decreasing cost) until it reaches a peak (maximum). It is commonly used for optimization problems.

Principle:

1. Start with an initial solution (state)
2. Evaluate neighbouring states based on an evaluation function
3. Move to the neighbour with the highest value (greedy approach)
4. Repeat until no better neighbour exists.

Eg. Consider a mountain climbing analogy where the goal is to reach the highest peak using only local observations.
Suppose we want to find the maximum of the function

$f(x) = -(x-3)^2 + 9$

→ Start at $x=0$, function value $f(0)=0$
→ Check neighbours $x=1$ & $x=-1$
→ Move to $x=1$ (higher value)
→ continue moving to $x=2$, $x=3$
→ At $x=3$ we reach the peak where $f(3)=9$

Draw backs of Hill climbing

1. Local Maxima

The algorithm can get stuck at a local maximum instead of finding the global maximum

2. Plateau

A flat region where all neighbouring states have the same value. The algorithm gets stuck because no better move exist.

3. Ridges

The search can be trapped between two higher peaks but cannot move in the right direction due to its greedy nature.

Limitations of Steepest Ascent hill climbing

Steepest hill climb is a variant where the best possible move is choosen in each step. However it suffers from aditional limitations

1. More computationally expensive.
2. Susceptible to local maxima
3. Slow progress in plateaues

Q13. Explain simulated Annealing and write its algorithm.

→ Simulated Annealing is a probabilistic search algorithm used for optimization problems. It is an improved version of hill climbing that allows downhill moves to escape local maxima and explore better solutions. It is inspired by the annealing process in metallurgy, where materials are heated and slowly cooled to remove defects and reach a stable, low energy state.

Algorithm:

1. Start with an initial solution and temperature T
2. Generate a random neighbouring solution
3. Compare the new solution with the current one
   - If better accept it
   - If worse, accept it with a probility $P = e^{-\Delta E/T}$
4. Gradually decrease the temperature (cooling)
5. Repeat until the system is "frozen"

Q14. Explain A* algorithm with an example

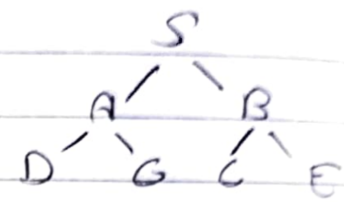
→ A* is a graph / tree search algorithm used in pathfinding and graph traversal. It finds the shortest path from a start node to a goal node by considering both:

- Cost to reach the node ($g(n)$)
- Estimated cost from the node to the goal ($h(n)$)

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

A* is both complete and optimal, meaning it always finds the shortest path if the heuristic is admissible.

Given cost :



g-value

- $S \to A = 2$, $S \to B = 5$
- $A \to D = 3$, $A \to G = 3$
- $B \to C = 4$, $B \to E = 3$

h-value

- $h(S) = 6$, $h(A) = 4$, $h(B) = 2$, $h(D) = 4$, $h(C) = 2$, $h(G) = 1$
$h(E) = 3$

f(n) for nodes :

$S \to 6$ $\quad$ $G \to 4$

$A \to 6$ $\quad$ $C \to 11$

$B \to 7$ $\quad$ $E \to 6$

$D \to 9$

Final path = $S \to A \to G$

Q15. Explain Min Max algorithm and draw game tree for TIC TAC TOE game.

→ Minimax is a decision making algorithm used in two player, zero sum games like Tic Tac Toe, chess, and checkers. It assumes both players play optimally and aims to minimize the opponents maximum gain while maximizing its own minimum gain.

Features :

→ Two players
  Maximizer - tries to maximise the score
  Minimizer - tries to minimize the score

→ Every node represents possible moves. Every level represents one players turn. Leaf nodes are game outcomes.

-1 for each move, -10 for shooting arrow.
Environment : 4x4 with wumpus, pits, gold, wall
Actuators : move forward, turn left / right, grab, shoot
Sensors : Breeze, stench, glitter, bump, scream
Percept sequence :
[breeze, stench, glitter, bump, scream]
Eg. [No, Yes, Yes, No, No]
Wumpus nearby and gold in current room

Q18. Solve the following Crypto arithmetic problem.
SEND + MORE = MONEY

->
```
   SEND
+  MORE
-------
  MONEY
```
1. Since M comes through carry and maximum carry can be 1
∴ M = 1

D + E = Y or Y + 10
N + R + carry = E or E + 10
E + O + carry = N or N + 10
S + 1 + carry = O or O + 10

2. Since O appears in the second position, and there is no carry affecting this position, O = 0
3. From the thousands place
S + 1 = O
∵ O = 0, S = 9

4. To satisfy the middle column conditions, $E=5$ fits correctly when checking the sum of N and R.

5. $N=6$

Since N appears at hundreds place, we check the addition

6. From the rightmost column

$D+E=Y$

$\therefore 7+5=12$ (carry 1) so $Y=2$

7. The only remaining digit that satisfies all conditions without repetition is $R=8$

$\therefore$

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

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

Q19. Consider the following axioms :

All people who are graduating are happy

All ~~peo~~ happy people are smiling

Someone is graduating

Explain the following

→ 1. All people who are graduating are happy

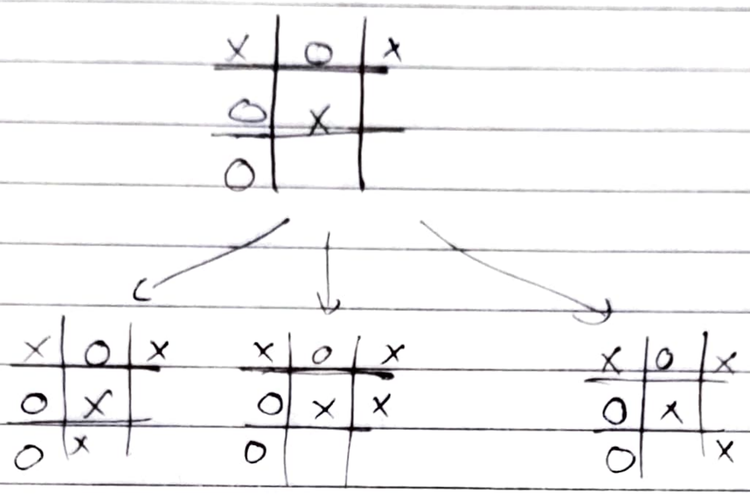
$\forall x\,(\text{graduating}(x) \rightarrow \text{happ}(x))$

2. All happy people are smilling

$\forall x\,(\text{happ}(x) \rightarrow \text{smiling}(x))$

3. Someone is graduating

$\exists x\,(\text{graduating}(x))$



X wins

1. Generate all possible game states from current state
2. Recursively evaluate each state using minimax
3. Assign scores
   - If maximizer score (X) → choose maximum score
   - If minimizer score (O) → choose minimum score
4. Backtrack scores up the tree to decide the best move

Q16. Explain Alpha beta pruning algorithm ad draw game tree for adversarial search.

→ Alph beta pruning is an optimization technique for the Minimax algorithm that reduces the number of nodes evaluated in a game tree. It does this by pruning (eliminating) branches that won't effect the final decision, making Minimax faster ad more efficient. The minimax evaluates all possible moves in a game tree, which makes it computationally expensive. Alpha-beta pruning skips unnecessary calculations by stopping evaluation for branches that won't change the final decision.

$\alpha \rightarrow$ Best value found so far for maximizer
$\beta \rightarrow$ Best value found so far for the minimizer
Pruning occurs when
$\alpha \geq \beta$, meaning a better move is already available, so further evaluation is unnecessary.

Q17. Explain WUMPUS world environment giving its PEAS description. Explain how percept sequence is generated?

→ The wumpus world is a grid based environment used in AI to demonstrate logical reasoning and agent based problem solving. It is a partially observable, stochastic and sequential environment where an agent navigates a world filled with hazards.

Components :
- → Grid based : 4x4 grid
- → Agent : moves throughout the world and tries to find goal
- → Wumpus : A monster who will eat the agent if it enters its room
- → Pits : Deadly traps resulting in loss
- → Gold : Agents has to collect goald
- → Breeze : indicates nearby pit
- → Stench : indicates nearby wumpus
- → Glitter : indicates gold
- → Arrow : one arrow to shoot the wumpus.

PEAS description :
P : 1000 points for finding gold, −1000 for falling in pit or getting eaten by wumpus

Convert to clause form

1. ¬ Graduating (x) ∨ happy (x)
2. ¬happy (x) ∨ smiling (x)

~~Resolution~~ : 3. Graduating (A)

Resolution :

From 3

we substitute x ← A in clause 1

¬ Graduating (A) ∨ happy (A)

since graduating A is true,

∴ happy (A)

From clause 2

¬happy (A) ∨ smiling (A)

since happ(A) is true

∴ smiling (A)

Therefore we prove, someone is smiling

Graduating (A)

↓

¬gratuating (A) ∨ happ(A)

↓

happy (A)

↓

¬happy (A) ∨ smiling (A)

↓

smiling (A)

∴ someone is smiling.

Q20. Explain Modus ponen with suitable example

→ Modus Ponens is a fundamental rule of inference in logic. It follows the form:
1. If P, then Q ($P \rightarrow Q$)
2. P is true (P)
3. Therefore, Q is true (Q)
Eg.
~~For~~ Rule : If it rains, the ground will be wet.
$P \rightarrow Q$
Fact : It is raining (P is true)
∴ Q is also true
i.e the ground is wet.
This rule is widely used in mathematical proofs, programming and real world decision making scenarios.

Q21. Explain forward chaining and backward chaining algorithm with the help of example.

→ Forward chaining and Backward chaining are reasoning algorithm used in Artificial Intelligence and Expert systems for inference in rule based systems.

1. Forward chaining starts with known fact and applies inference rules to extract more data until a goal is reached.
Process:
start with a given fact
apply rules to derive new fact
Continue until goal is reached or no rule can be applied.

Eg. Diagnosing
1. If a patient has fever and cough, then they might have flu.
2. If a person has a sore throat and a runny nose, then they might have a cold
3. If a person has flu, then they should rest and take fluids
Fever $\wedge$ cough $\rightarrow$ Flu
Sore throat $\wedge$ Runny nose $\rightarrow$ Cold
Flu $\rightarrow$ rest & fluids
Fact: patient has fever, patient has cough.
Chaining:
Patient has fever and cough
$\therefore$ R1, we deduce patient has flu
From R3, we conclude patient needs to rest and take fluids

2. Backward chaining starts with a goal and works backwards by checking which facts or rules support the goal.
Process:
Start with the goal
Check the rules that can derive goal
If rule's conditions are not known, check for facts or other rules that support those conditions.
repeat until goal is proven or no more rules apply.
Eg. Diagnosing
1. Does the patient have flu can be concluded if the patient has fever and cough
2. It is known facts that patient has fever and cough
$\therefore$ we conclude that patient has flu.