Name: Viven Hotwani
Class: D15C
Roll No: 17

AI-DS ASSIGNMENT 1

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

→ AI (Artificial Intelligence) is the simulation of human intelligence processes by machines, especially computer systems, including learning, reasoning, problem-solving, perception and decision-making.

During the COVID-19 pandemic, AI played a crucial role in:

(i) Diagnosis and Detection: AI-powered imaging tools analyzed CT scans and X-rays to detect COVID-19 patterns.

(ii) Drug and Discovery: AI accelerated the identification of potential treatments and vaccines by analyzing vast datasets.

(iii) Contact Tracing: Mobile apps used AI to track and predict virus spread patterns.

(iv) Chatbots: AI-powered virtual assistants provided reliable information and ~~symp~~ symptom checks.

(v) Supply Chain optimization: AI-helped manage medical supply distribution and vaccine logistics.

FOR EDUCATIONAL USE

2) what are AI agents terminology, explain with examples?

→ AI agents are entities that perceive the enviorment through sensors and act upon it through actuators. Key terminology

(i) Agent: A system that perceive and acts (e.g. a robot vaccum cleaner).

(ii) Enviornments: The world it operates in (e.g. room for vaccum)

(iii) percepts: Inputs the agent receives (e.g. detection).

(iv) Actions: what agent can do (e.g. move, dirt).

(v) Performance Measure: How success is evaluat (eg. cleanliness).

(vi) Sensor: Device that perceives (eg. came

(vii) Acutator: Device that acts (eg. motor)

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

→ The 8 puzzle problem is solved using AI searching techniques:

(i) State representation: Each tile arrangement is a state.

(ii) Initial and goal states: Defined initial and goal state configurations for more heuristics.

(iii) Operators: Legal moves (blank tile can move up/down/left/right)

(iv) Search algorithms:
- BFS/DFS: ~~Ep~~ Explore possible moves.
- A*: Uses heuristic (e.g. Manhattan distance) to find the most optimal path.
- Heuristic search: optimizes cost to goal for efficiency.

Eg: A* with ~~Manhatta~~ Manhattan distance heuristic efficiently finds the shortest sequence of moves to reach the desired goal state.

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

→ PEAS describe an agent's performance measure, Enviornment, Actuators and Se

(i) Taxi Driver:
- P: Safe, fast, legal, comfortable trip, maximum
- E: Roads, Traffic, pedestrians, weather
- A: Steering, accelator, break, signal, dis
- S: Cameras, speedometer, GPS, odometer, accelerometer

(ii) Medical diagnosis system:
- P: Accurate diagnosis, patient health improvement
- E: Patient symptoms, medical history, test results.
- A: Display diagnosis, recommend treatments
- S: Input symptoms, lab test interfaces

(iii) A music composer:
- P: Creative, original, pleasing compositio
- E: Musicals trends, user preferences
- A: Generate musical notes, scores
- S: Input preferences, existing musical database.

(i) An aircraft autolander:
- P: Safe, smooth landing, correct runway
- E: Weather, runway, aircraft position/speed
- A: Control surfaces, throttle
- S: Altimeter, GPS, radar, ILS

(ii) An essay evaluator:
- P: Accurate, consistent grading
- E: Student essays, grading rubrics
- A: Display grades/comments.
- S: Text input, grammar/style checkers.

(iii) A robotic sentry gun for the Keck Lab:
- P: Detect and neutralize threats, minimize false positives
- E: Lab perimeter, intruders
- A: Alarm, disable mechanisms
- S: Motion detectors, cameras, thermal sensor

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

(i) Observable: Partially observable (can't see all inventory at once).

(ii) Deterministic: ~~So~~ Stochastic (customer behaviour unpredictable).

(iii) Episodic/sequential: sequential (current recommendations affect future ones)

(iv) static/Dynamic: Dynamic (inventory changes, customers come/go)

(v) Discrete/continuous: Discrete (finite books/categories).

(vi) Single/multi-agent: Multi-agent (interacts with customers, staff)

6) Differentiate Model based and utility based agent:

→ Model-based Agent:

(i) Maintains an internal ~~world~~ model of the real world.

(ii) Uses model to predict outcomes of actions.

(iii) Does not considers desirability of states.

(iv) EG: Chess program that simulates moves.

daram®

FOR EDUCATI...

utility - based Agent:

(i) Includes a utility function to evaluate states.

(ii) ~~Eso~~ Chooses action that maximises expected utility.

~~(iii) Example: Investment bot~~

(iii) Includes desirability of states to maximise utility.

(iv) Eg: Investment bot selecting highest return option.

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

→ Knowledge - based Agent:

Its architecture consists of four parts:-

(i) Knowledge Base: It stores the facts and rules.

(ii) Inference Engine: It applies logical rules to KB.

(iii) Perception: It adds new facts to th[e]
KB.

(iv) Action: selects actions based on inferen[ce]

Learning Agent:

Its architecture also consists of four parts:-

(i) Performance Element: It makes the decis[ion]

(ii) Learning Element: Improves the performa[nce] element by learning an[d] adopting.

(iii) Critic: It provides feedback on the performance.

(iv) Problem generator: It suggests new experience to the agent.

8) Convert the following to predicates:

a) Anita Travels by car if available otherwi[se] travels by bus.

→ Travels (Anita, Car) ← Available (Car) ∧ ¬ Travels (Anita, Bus) Travels (Anita, Bus) ← ¬ Available (Car)

b) Bus goes via Andheri and Goregaon.

→ Route (Bus, Andheri) ∧ Route (Bus, Goregaon)

c) Car has puncture so it's not available.

→ ¬ Available(Car) ← Puncture (Car)

Q Will Anita travel via Goregaon? Use forward reasoning.

→ Given Puncture(Car) is true:

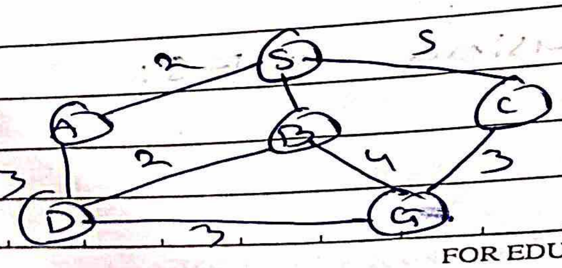
(i) From c: ¬ Available (Car)

(ii) From a: Travels (Anita, Bus)

(iii) From b: Since Anita travels by Bus and Bus goes via Goregaon, Anita travels via Goregaon.

∴ Yes, Anita will travel via Goregaon.

a) Find route from ~~st~~ S to G using BFS.



FOR EDUCATIONAL USE

BFS Traversal from S to G:

Start at S
- Queue: [S]
- Visited: [ ]

(i) Dequeue S, enqueue its neighbours: A, B, C
- Queue: [A, B, C]
- Visited: [S]
- Parent: {A: S, B: S, C: S}

(ii) Dequeue A, enqueue its neighbours D
- Queue: [B, C, D]
- Visited: [S, A]
- Parent: {D: A}

(iii) Dequeue B, enqueue G (not D again since its already queued).
- Queue: [C, D, G]
- Visited: [S, A, B]
- Parent: {G: B}

We've reached G! So we backtrack using the parent map

Path to G using BFS:
- S → B → G.

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

→ Depth limited search (DLS):

(i) DFS with a depth limit l.

(ii) Nodes at depth l are treated as having no successors.

(iii) Prevents infinite depth in infinite trees.

Iterative Deepening search (IDS):

(i) Performs DLS with l=0 by default, taking on its own.

(ii) If goal is not found, it increments l and repeats the search.

(iii) It combines benifits of both BFS (completeness) and DFS (memory efficiency)

Example: Searching a tree with depth 3

(i) Try l=0, check root only.

(ii) I=1 : check root and children

(iii) I=2 : check two levels.

(iv) I=3 : check three levels, here the goal will be found.

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

→ Hill climbing :

(i) It is a local search algorithm that continously moves in the direction of increasing value.

(ii) It moves and search ~~s~~ in the upward direction going from children to parent ~~s~~ nodes or like climbing a hill.

(iii) It terminats when no higher neighbours exists.

Eg: Finding maximum in function f(x):

(i) start at random point x.

FOR EDUCATIONAL USE

(ii) Move to neighbour y if f(y) > f(x)
(iii) Repeat until no better neighbour

Drawbacks:

(i) Local maxima: Can't find global optimum.

(ii) Ridges: Sequence of local maxima

(iii) Plateaus: Flat areas with no uphill

Steepest-ascent limitations:

(i) Computationally expensive (evaluates all neighbours)

(ii) Still susceptible to local maxima

(iii) May oscillate if multiple equal optio

Explain simulated Annealing and write i algorithm.

Simulated Annealing:

(i) It is inspired by a metal cooling process.

(ii) At first, it allows some 'bad' move early to escape local optima.

(iii) Probability of accepting worse moves decreases over time.

Algorithm:

1. Initialize Temprature T, initial state

2. While T > T_min :
   a. Generate random neighbour s'
   b. Calculate ΔE = cost (s') - cost (s)
   c. If ΔE < 0, accept s'
   d. Else, accept s' with probability e^(-ΔE
   e. Reduce T according to cooling schedul

3. Return best state found.

13) Explain A* Algorithm with an example.

A* algorithm:

(i) Best-first search using f(n) = g(n) + h(
- g(n) = cost from start to n
- h(n) = heuristic estimate to goal.

(ii) Optimal if h(n) is admissible (never overestimates)

Eg: Pathfining on a grid:

(i) Start at initial node
(ii) Expand node with lowest f(n)
(iii) For each neighbour, calculate f(n)
(iv) continue until goal is reached.

4) Explain Minimax Algorithm and draw game tree for Tic Tac Toe game.

→ Minimax Algorithm:

(i) It is a recursive algorithm for decision making in games.

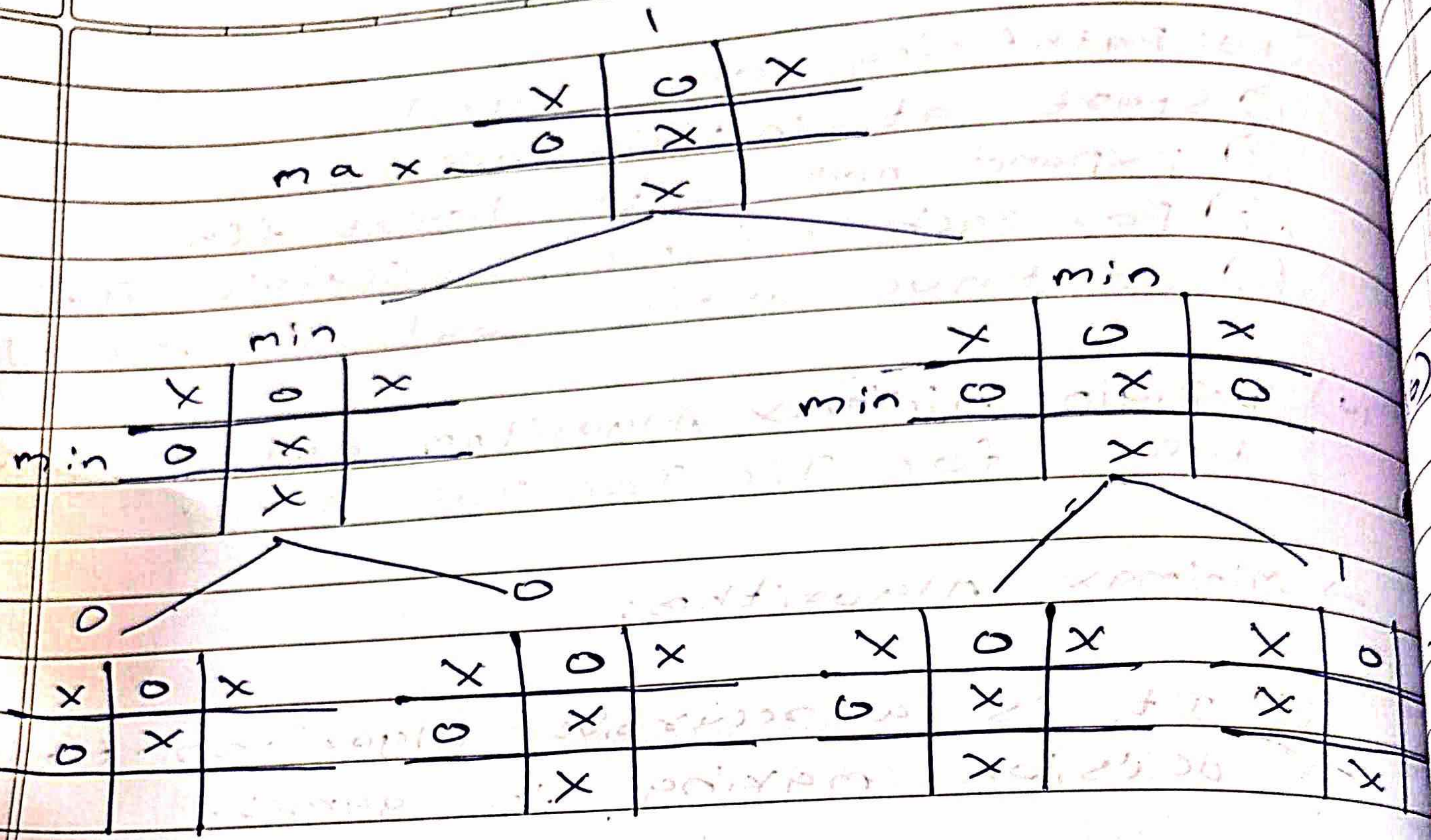
(ii) Max player chooses move to maximize minimum payoff.

(iii) Min Player chooses to minimize maximum payoff.

(iv) They both try to compete each others actions to make the game more competitive.

The game tree for tic tac toe using the min max algorithm is as follows:-

FOR EDUCATIONAL USE



15) Explain Alpha beta pruning algorithms for adversial search with example.

Alpha-beta pruning:

(i) It is an optimization of the minmax algorithm.

(ii) It just prunes the branches which are unnecessary and don't affat the final decision.

(iii) α: best value for MAX found so far

(iv) β: best value for MIN found so far.

Example:-
when exploring MIN node, if we find a value $\leq \alpha$, we can prune remaining siblings because MAX won't choose this path.

Explain wumpus world enviornment giving its PEAS description. Explain how percept sequence is generated?

WUMPUS world PEAS:

- P: Find gold, avoid pits) wumpus, minimize steps.
- E: Grid with pits, wumpus, gold
- A: Move, turn, grab, shoot, climb
- S: Strench (near wumpus), Breeze (near p Glitter (Gold), Bump (wall), Scream (wumpus killed).

Percept sequence:

At each step, agent receives percept vector:
- [Strench, Breeze, Glitter, Bump, Scream]

This was generated based on current location and enviorment state (initial).

18) 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.

→ (i) $\forall x\,(Graduating(x) \rightarrow Happy(x))$

(ii) $\forall x\,(Happy(x) \rightarrow Smiling(x))$

(iii) $\exists x\; Graduating(x)$

2) Convert each formula to clause form.

→ (i) $\neg Graduating(x) \vee Happy(x)$

(ii) $\neg Happy(y) \vee Smiling(y)$

(iii) $Graduating(a)$ (for some constant a)

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

(i) Goal: $\exists z\; Smiling(z)$

(ii) Negated goal: $\neg Smiling(z)$

(iii) Resolve $\neg Smiling(z)$ with $\neg Happy(y) \vee Smiling(y)$

$\{y/z\} \rightarrow \neg Happy(z)$

(iv) Resolve $\neg Happy(z)$ with $\neg Graduating(x) \vee Happy(x)$

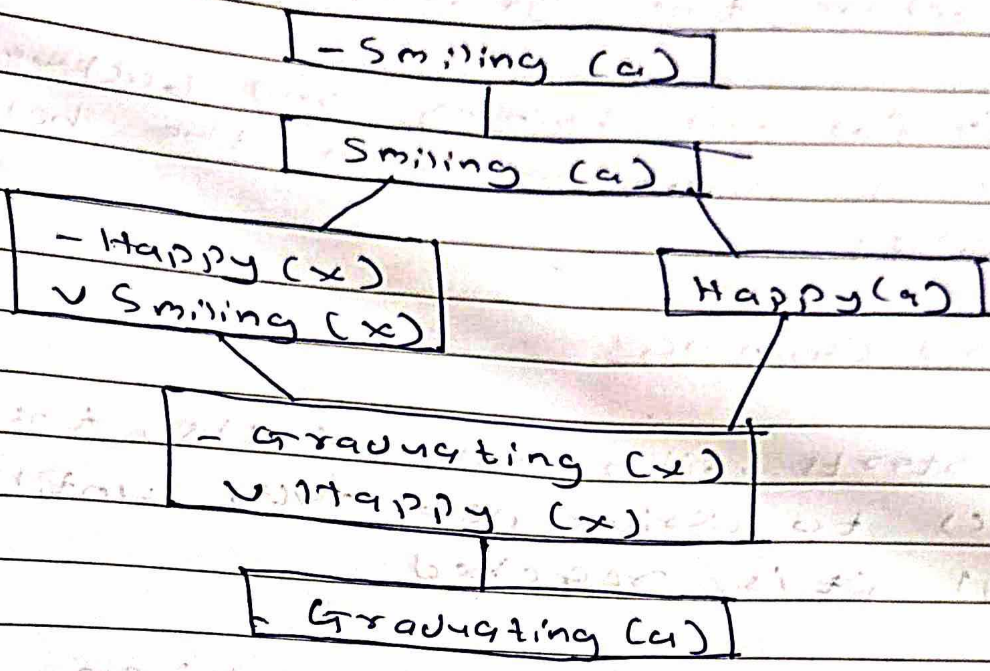
$\{x/z\} \rightarrow \neg Graduating(z)$

(v) Resolve $\neg Graduating(z)$ with $Graduating(a)$! $\{z/a\}$

empty clause

(vi) Contradiction proves $\exists z\; Smiling(z)$

Resolution tree:



1(a) Explain Modus ponen with suitable example.

→ Modus ponen is a fundamental rule of inference in propositional logic and first order logic. It states:

If P implies Q (P ⇒ Q), and P is true then Q must be true.

Eg:

(i) Premise 1 (conditional):
If it rains, then the ground will be wet.
(P → Q)

(ii) Premise 2 (Fact):
It is raining. (P)

(iii) Conclusion:
Therefore, the ground is wet. (Q)

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

→ Forward Chaining:

(i) It starts with known facts, then applies rules to derive new facts, until the goal ~~it~~ is reached.

(ii) It is data/knowledge driven.

(iii) Example:
Rules:
1. ~~A∧P~~ A ∧ B → C
2. C∨D → E

Facts: A, B
Process:
- From A and B, Derive C
- From C, derive E.

Backward Chaining:

(i) Starts with goal, and then worked backward to find its supporting facts

(ii) It is goal/final state driven.

(iii) Example:

Goal: Prove E.

- E requires C or D
- Try to prove C
  - C requires A and B
  - A and B are known
- Therefore E is true.

) Solve the following crypto arithmet function: SEND + MORE = MONEY

	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

(i) Assign unique digits (by calculatio and assumptions).

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

(ii) Verify the sum.

	9	5	6	7
+	1	0	8	5
1	0	6	5	2