Roll No: 24
DISC

Nishant S Khetal

SDS Page No.
Date / /

AIDS - ASSIGNMENT 05/05

1. What is AI? Considering the covid 19 pandemic situation how AI helped to survive and renovated our way of life with defence application?

→

Artificial Intelligence (AI) is a branch of computer science that focuses on creating intelligent systems capable of performing tasks that typically require human intelligence. These tasks include problem solving, decision making, learning perception, etc.

Role in surviving & renovating life during COVID 19.

i) Healthcare & medical diagnosis.
- CTscan analysis & X-ray diagenisis helped in rapid detection of condition cases.

ii) Virus Spread Control.
- Serial distance monitoring tools helped to monitor & explore law's in public places.

iii) Remote work & education.
- Work from home optimization & AI in online education help people continue with their work & education remotely

iv) Fake news detection.
- AI helped to identify rapidly spreading fake news on social media.

2. What are AI Agents terminology? Explain with examples
- Environment.
. Everything which surrounds the agent & influences its actions. It can be complete or partially observable.
ex: checkboard environment for chess playing AI.

- Percepts:-
- Percepts are the raw data that an agent gets from its sensors
- eg: self driving car percepts include, GPS signals, etc.
- Actuator:
- They are the components that allow an agent to take actions in the environment.
- eg: A robotic arm uses motors as actuators to pick objects
- Goal:
- The final state of an agent to achieve is called as goal.
- eg: victory is a chess game.

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

→

- The 8 puzzle problem is a state space research problem in AI where a 3×3 grid contains 8 tiles numbered from 1 to 8. 1 empty space.
- Objective is to rearrange the tiles to reach a predefined goal state.
- AI techniques :-
- Uninformed search methods such as :-
- BFS : Expand the shallowest nodes first.
- DFS : Explores as deep as possible before backtracking.
- IDS : combines DFS & BFS to increase depth first gradually.
- Informed search methods :-
- BFS : best first search based on heuristic function that appears closest to the goal.
- A* search : $f(n) = g(n) + h(n)$, based on both heuristic h cost to node.

Initial state:

1	2	3
5	6	0
4	7	8

Goal state:

1	2	3
4	5	6
7	8	0

i) Compute heuristic of each possible move.
ii) Expand the state with the lowest f(n) & repeat.

Q.4 What is PEAS descriptor? Give PEAS descriptor for the following?
→ Performance Measure: how success of agent is evaluated.
Environment: Surroundings in which agent operates.
Actuators: Component that allows agent to take actions.
Sensors: Component that allows agent to perceive the environment.

i) Taxi driver agent

Performance	Environment	Actuators	Sensor
- Safe driving - travel time - traffic rules	- traffic signal - roads - weather	- steering wheel - accelerator - brakes	- cameras - GPS - fuel guage

ii) Medical diagonsis system

Performance	Environment	Actuators	Sensor
- health of patient - accuracy of diagonsis - recommended treatment	- patient data - symptoms - test reports	- display screen - alarm system - robotic arms	- heart rate - monitors - lab results

iii) Music composes agent

Performance	Environment	Actuators	Sensor
- originality - listens engagement - quality.	- musicdb - user preferences	- speaker system - digital music - interface	- microphones - user feedback - recognition slw.

iv) Aircraft autolander

Performance	Environment	Actuators	Sensor
- Smooth landing. - accuracy	- runway - air traffic - wind conditions	- landing gear - flap - air brake	- altimeter - GPS - camera.

v. Essay evaluator

Performance	Environment	Actuators	Sensors
- grading - grammar - plagarism deck	- plagarism databases - rubric ceiteria	- display screen - text to speech systems	- optical character recognition (OCR)

vi. Robotic sentry gun

Performance	Environment	Actuators	Sensors
- neutralize threats - target tracking - false alarms	- lab area - protocol intruders	- gun mechanism - alarm sites - tracking system	- camera - thermal sensors

Q.5. Categorize a shopping bot for an offline shopping but for an offline bookstore according to the following dimensions.

→

- Observability : Partially observable. Relies on limited sensors input.
- Deterministic or stochastic : stochastic customer pref is unpredictable.
- Episodic V/s sequential : sequential, Decision affects future actions.
- Static V/s Dynamic : Dynamic Customer behaviour is continuously evolving.
- Discrete V/s continuous : Finite no of choices such as books, authors, payment options, etc.
- Single V/s multiagent : But interacts with multiple agents and customers, store employes & other bots.

Q.6 Differentiate between model based & utility based agent.

Model based agent	Utility based agent
- Agent that maintains an internal model of the env to understands its current state & predict future states.	- Agent that selects actions based on utility fn aiming to maximize long term satisfaction or benefit.
- Model updates its knowledge about the environment.	- Measures how desirable different states are.
- Less complex	- More complex.
- Doesn't concess long term rewards.	- Focus on long term rewards.
- eg: self driver car.	- eg: shopping recommendation sys.

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

→

Knowledge based agent: Stores knowledge & reasons ☐ it making decisions based on logical inference.

- Knowledge base (KB): Stores facts, rules & heuristics about the environment.
- inference engine: Uses logical reasoning techniques like forward & backward chaining.
- perception: gathers data from the environment.
- actions: Execute action based on inferred knowledge.
- knowledge update mechanism: Updates itself as new facts are learned.
- learning based agent: Agent that improves its perf overtime by learning from experience, data & feedback.
- learning element: responsible for improving agents perf by analysing past experiences using ML techniques.
- critic: provides feedback on agents actions by evaluating success or failures.

- problem generation: Supports new experience for learning & exploration.

Q8 → Convert the following to predicates.

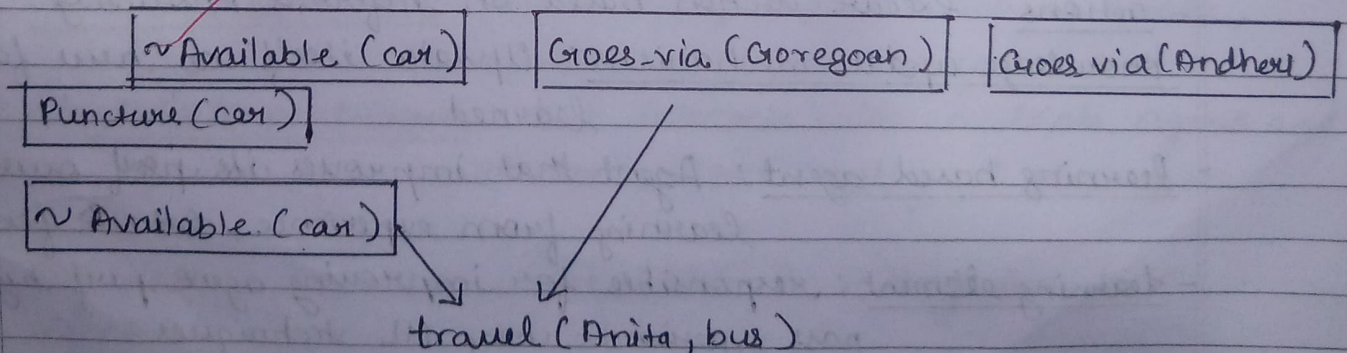
Anita travels by car if available otherwise travels by bus.
travels (x,y) → Person x travels by y.
Available (y) → y (a vehicle) is vechicle available.
Goes_via (y,z) → Vehicle y goes via z.
Puncture (y) → y (a vehicle).

a. ~ Available (car) → Travels (Anita, bus).

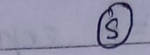
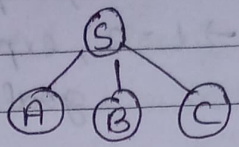
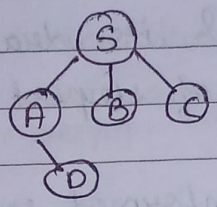
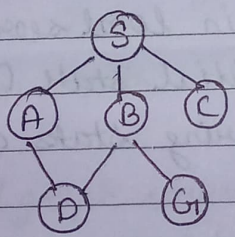
b. Bus goes via Andheri & Goregoan
Goes_via (bus, Andheri)
Goes_via (bus, Goregoan).

c. Car has a puncture, so its not available.
Puncture (car)
~ Available (car)

Forward reasoning: will Anita travel via Goregoan.



Q.10. Find the route from S to G_1 using BFS [Breadth first search].

Steps	Representation	Stack
i Load S		[S]
ii Pops load A, B, C		[A, B, C]
iii. Pop A & Expand D		[B, C, D]
iv. Pop B Expand G_1		[C, D, G_1]

G_1 is goal node ∴ Route from S-G_1 is S → B → G_1

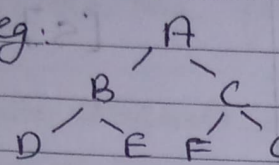
Q.11 What do you mean by depth limited search? Explain iterative deepening search with example.

→

Depth limited search is a variation of Depth first search that limits depth of exploration to a fixed level L. It prevents infinite loops in infinite state spaces & reduces mem. usage.

- start with initial node until depth limit is reached.
- if goal is found, return success.
- backtrack & explore other nodes.
- if limit is reached, without finding goal, return failure.

- Iterative deepening search (IDS) combines both DFS & BFS by gradually inc. depth limit until goal is found. It follows same algo as DLS, except if limit is reached, limit → 1 & algo repeats.

eg: 

→ L=0 explore A, goal not found.
→ L=1, explore A-B, A-C goal not found
→ L=2, explore B→D, B→E, C→F, C→G
goal G found at L=2

Q.12. Explain hill climbing & its drawbacks in detail with example Also state limitations of steepest ascent hill climbing.

→ Hill Climbing is an informed search algorithm used for optimization problems in local search.
- We start with an initial state (random or given).
- Evaluate the neighbouring states and choose one with the highest value.
- Move to the best neighbouring state & repeat.
- eg: 8 queen problem where we need to add 8 queens on the chessboard such that no two queens attack each other.
  - We start with a random placement of queens.
  - Then move to a neighbouring state by modifying the position of queen.
  - repeat until conflicts cannot be reduced further.
- Drawbacks: Local maxima: get stuck at peak which isn't global optimal plateau: All neighbouring states with same value
  - ridges: complex landscape.

Limitation of steepest ascent hill climbing:
Only selects neighbour with the highest improvement.
More computationally expensive: Need to evaluate the neighbour.

- Sensitive to initial data: Bad starting leads to poor solution
- Existing problems of hill climbing algorithm.

Q.13 Explain simulated annealing & write its algorithm.

→ Its an optimized version of hill climbing designed to escape its limitation. Its inspired by annealing process in metallurgy where a neutral is heat & slowly cooled to remove defect.

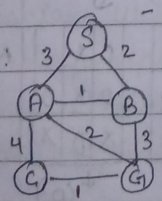
Algorithm: Start with an ~~in~~ initial solution S.
- Generate a random neighbour S' state S.
- If S' is better than S, accept it.
- If S' is worse, accept it with probability
  $P = e^{(-Delt)}$.
- return the best found solution.

Q.14 Explain A* algorithm with an example.

→

- A* is a path and graph traversal algorithm which combines both - Uniform cost search (UCS)
  - Greedy best first search (BFS).
- A* evaluates such node using the function
  $f(n) = g(n) + h(n)$
- Algorithm:- Initialize start node into an open list.
  - Compute $f(n) = g(n) + h(n)$. select node with the smallest $f(n)$.
  - Move it to closed list & expand its neighbours.
  - If neighbor is goal, return path.
  - Continue till goal is reached.

eg:



$h(S) = 4$, $h(A) = 2$, $h(B) = 1$, $h(C) = 2$
$h(G) = 0$.

i. Initialize S, expand A, B. $f(A) = 3+2 = 5$
$f(B) = 2+1 = 3$

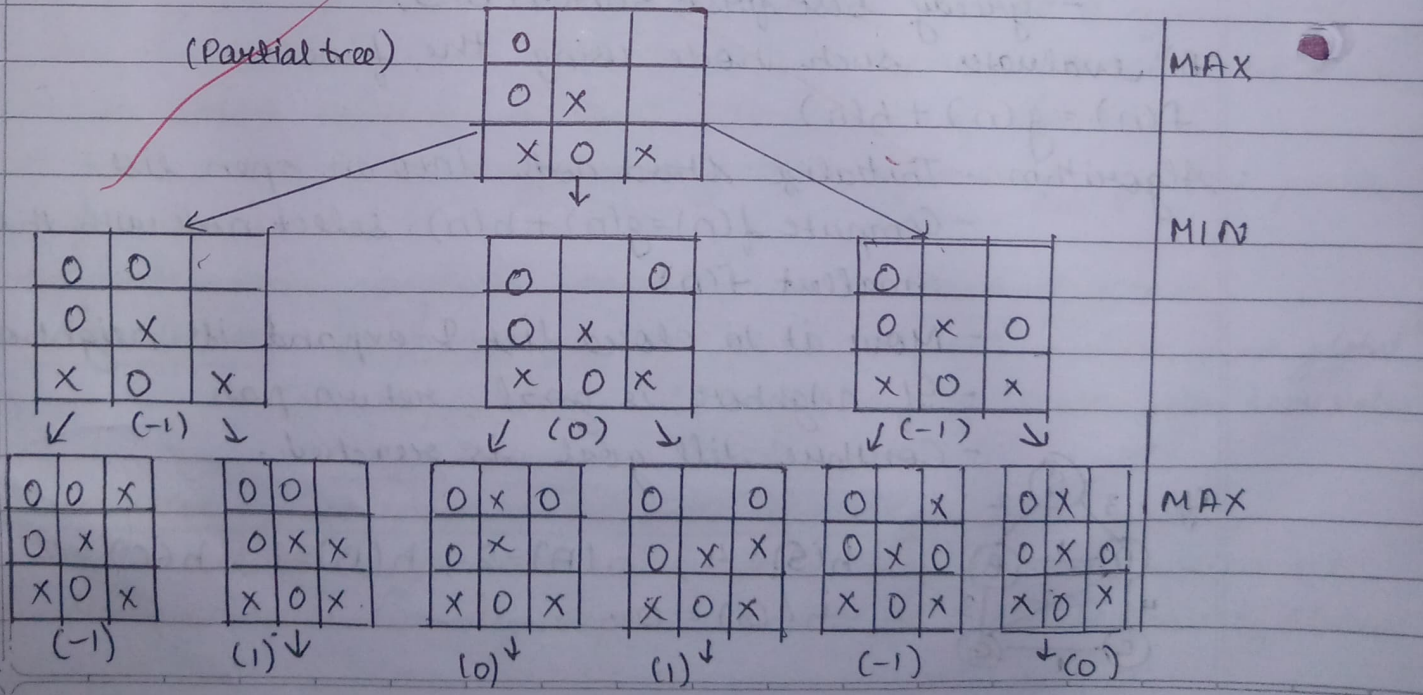
ii. Expand B to $f(G) = (2+3) + 0 = 5$
Goal is found.
Path is $S \to B \to G$.
Cost is 5.

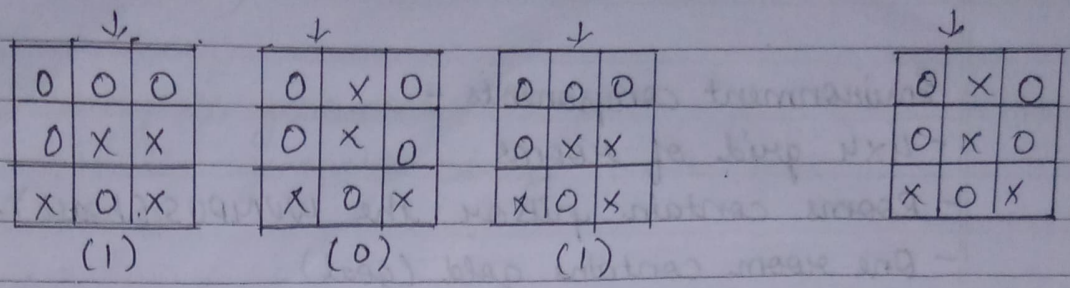
Q.15 Explain minimax algorithm & draw game tree for TicTacToe

→

Minimax is a decision making algorithm used in game theory & AI. It helps in finding the optimal solution by -
- player 1 tries to max their score.
- player 2 tries to minimize their score.

Algorithm :- Generate all possible states. [Game tree].
- Assign scores to terminal nodes.
- Backtrack → Max picks the highest valued node & viceversa for min.
- Repeat until the root where Max chooses.





∴ Node 2 option has the most optimal choice.

Q.16 Explain Alpha beta pruning algorithms for adversial search with example.

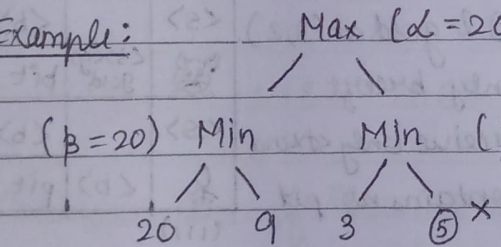
→

Its an optimization technique for minimax algorithm. It reduces no of nodes evaluated in a game tree by eliminating unneccessary branches.

Algorithm: Set two values α & β for Max & Min respectively.

- If $\alpha \geq \beta$ for any subtree or node, stop evaluating and prune it.
- Prune all nodes which do not influence the final decision.

Example:



- since right subtree is providing a min of 3.
∴ 3 or less & $\alpha = 20$.
∴ Node 5 will not be evaluated & be pruned.

Q.17 Explain WUMPUS world environment giving PEAS descriptive. Explain how percept sequence is generated.

→ The WUMPUS is a grid based environment used to demo AI agents, logic based reasoning & decision making.

Environment components -
- 4×4 grid of rooms.
- Rooms contain pits or the WUMPUS (beast).
- One room contains gold (goal).
- Agent starts at (1,1) & can move horizontally & vertically.
- One arrow to kill the WUMPUS.
- Rooms surrounding the WUMPUS have stench & rooms surrounding the pit have breeze.

PEAS

Performance	Environment	Actuators	Sensors
- Find gold in min actions. - Avoid WUMPUS & pits.	- 4×4 grid - WUMPUS, pits, gold.	- move front, back up, down, - grab, shoot - climb.	- detect stench or breeze. - glitter (gold) - scream (kill WUMPUS)

Percept sequence :

- Assume agent starts at (1,1) & gold at (3,2)
- Move to (1,2) → Perceive only breeze
- Move back & up (2,1) → Perceive only stench
- Logically infer that (2,2) contains no pit or windows.
- Similarly continue till reaching (3,2) goal.

<S>		<b>	
<S><g> gold		pit	<b>
<S>		<b>	
<b>		pit	<b>

(111)

Q.18. Solve the following crypto arithmetic problems.

→

SEND + MORE = MONEY

```
          +(1)  +(1)
     S    E     N     D
    (9)  (5)   (6)
+   M    O     R     E
    (1)  (0)   (8)   (5)
-------------------------
M   O    N     E     Y
(1) (0)  (6)   (5)
```

Since carry is generated, M = 1.

∴ S + M = 0 generates carry.

S + 1 = 0

S = 9,

O = 0

Now E + 0 = N

But E + N ∵ 2 numbers cannot have same value.

∴ S + 0 + generated carry & E + 1 = N.

∴ N + R = E, generate carry.

- Consider R = 8 & N = 2

Result in E = 0, not possible

∴ N + R = E, generates carry

- Consid If N = 3, result in E = 1, not possible - only case possible is

N = 6 & E = 5

- Now D + 5 = Y H should generate carry

∴ D = 7 & H = 2

0	O
1	M
2	N Y
3	
4	
5	E
6	N
7	D
8	R
9	S

∴ 9567 + 1085 = 10652

Q.19 Consider the following axioms.

All people who are graduating are happy

$\forall x$ (graduating(x) → happy(x)).

All happy people are smiling.

$\forall x$ (happy(x) → smiling(x)).

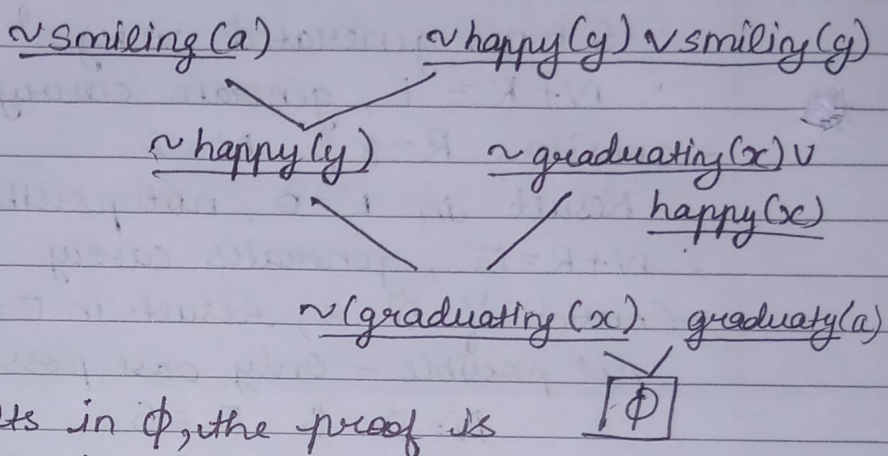
Someone is graduating $\exists x$ (graduating(x)).

Convert to clause form

~ graduating (x) ∨ happy (x).
~ happy (y) ∨ smiling (y).
graduating (a).

- Prove 'is someone smiling' → ~ smiling (a).

- Resolution tree



Since the tree results in Φ, the proof is validated ∴ Someone is smiling is true.

Q.20. Explain modes poren with suitable example.

→ Its a fundamental rule of logic stating $P \rightarrow Q = P \vdash Q$
If P implies Q & P is True then Q also must be T.
example:
- If it rains, the ground will be wet.
  Rains → wetground.
- It is raining.
  Rains → True.

∴ The conclusion is that the ground is wet.

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

→ Forward chaining is when reasoning starts with known facts & applies inference rules to new facts until the goal is reached.

Algorithm: Start with a set of known facts.
- Apply inference rules that match current facts.
- Derive new facts & add them to knowledge base.
- Repeat until goal is achieved.

Backward chaining is when reasoning starts with the goal (query) & works backwards to check if the goal can be derived from known facts.

Algorithm: Start with the goal. If goal is known fact, return true.
- If not find a rule where the goal appears on conclusion.
- Recursively check promises of the rule.
- If all promises are true, return goal = true.

Example: i If person has fever & cough they have flu.
ii If person has flu, they should take rest.
Facts :- John has fever. John has cough.
Prove : John should take rest.

→ Forward chaining: Fever(John), Cough(John)
Fever(x) ∧ cough(x) → flu(x)
Flu(John) → Rest(John)

→ Reverse chaining: Rest(John) ∴ Flu(John)
∴ Fever(John) & Cough(John) both are true, hence statement is proved.