Name :- Ishan Kiran Joshi Div - D15C Roll No: 21

21

05/05

AIDS Assignment

A.Y. 2024-25

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

→ Artificial Intelligence (AI) is the simulation of human intelligence in machines that can perform tasks like learning, reasoning, problem-solving, and decision-making. During the COVID-19 pandemic, AI played a crucial role in healthcare, research, and daily life. AI-powered algorithms helped predict virus spread, analyze medical data, and accelerate vaccine development. Chatbots and virtual assistants provided accurate health information, reducing the burden on healthcare workers. AI based diagnostic tools improved early detection of infections through X-ray and CT scan analysis. In workplaces, AI-enabled automation and remote working tools ensured business continuity. E-Commerce and delivery services used AI to manage demand and optimize supply chains. AI-powered robots assisted in sanitization and patient care in hospitals. Social media platforms used AI to detect misinformation and spread awareness. Overall, AI transformed various sectors, making life safer and more efficient during the crisis.

Q2] What are AI Agents terminology, explain with examp

→ AI Agents are intelligent Systems that perceive their environment through sensors and take actions to achieve specific goals using actuators.

1] Agent - An entity that senses and acts upon environment.

2] Sensors - Devices that collect data from the environment.

3] Actuators - Components that execute actions based on decisions.

4] Percept - The agent's perception of the environment at a given time.

5] Environment - The external surroundings in which the agent operates.

6] Performance Measure - A metric to evaluate the agent's success.

Example:-

A self-driving car is an AI Agent. Its sensors (camera) perceives traffic, its actuators (steering, brakes) control movement, its percept includes traffic conditions, and its environment is the road. The agent function maps perceptions to action (e.g. stopping at a red light), while the performance measure evaluates driving safety and efficiency.

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

→ The 8-puzzle problem is a classic example of AI search techniques used to find the shortest path to reach the goal state. It consists of a 3x3 grid with 8 numbered tiles and one empty space, where the goal is to arrange the tiles in a specific order by sliding them. AI techniques like Breadth-First search (BFS), Depth First Search (DFS), A* Algorithm, and Greedy Best-First Search are used to solve this problem. BFS explores explores all possible moves level by level, ensuring the shortest path is found. DFS goes deep into one path before backtracking, which may not guarantee the shortest solution. The A* Algorithm is the most efficient as it uses both path cost and heuristic to find the optimal solution. AI helps automate the search process, reducing human effort and ensuring the best possible moves are taken to reach the goal state efficiently.

Q4] What is PEAS description? Give PEAS description following:

→ PEAS stands for Performance Measure, Environment, Actuators, and Sensors, which are used to describe the working of an AI Agent.

- Performance Measure: Criteria to evaluate the agent's success.
- Environment:- The surroundings in which the agent operates.
- Actuators: Components used by the agent to take action.
- Sensors: Devices that collect data from the environment.

PEAS For:

1] Taxi Driver AI

Performance Measure: Safety, fuel efficiency, travel time, passenger satisfaction.

Environment: Roads, traffic signals, pedestrians, weather conditions.

Actuators:- Steering, brakes, accelerator, indicators.

Sensors: Cameras, GPS, speedometer, LIDAR, fuel sensor.

2) Medical Diagnosis System

Performance Measure: Accuracy of diagnosis, treatment effectiveness, speed of diagnosis.

Environment: Patient medical records, symptoms, test reports.

Actuators: Displaying diagnosis, suggesting treatments, alerting doctors.

Sensors: Patient input, lab test results, medical imaging.

3] AI Music Composer
Performance Measure:- Musical harmony, Creativity
Environment:- Musical notes, genres, existing components.
Actuators: Digital synthesizer, audio output, note generator.
Sensors:- MIDI input, music style data, user preferences

4] Aircraft Autolander.
Performance Measure: Smooth landing, passenger safety.
Environment: Runway, weather, wind speed, altitude.
Actuators: Flaps, landing gear.
Sensors: GPS, Speed sensors, weather radar.

5] Essay Evaluator AI
Performance Measure:- Grammar accuracy, relevance readability.
Environment: Essays, writing styles, predefined evaluation criteria.
Actuators:- Text analysis tools, grading system, Feedback generator.
Sensors:- Optical Character recognition, Natural Language processing.

6] Robotic Sentry Gun For Keck Lab.
Performance Measure:- Accuracy, threat detection efficiency.
Environment: Laboratory perimeter, intruders, weather conditions.
Actuators: Turret rotation, firing mechanism, alarms.
Sensors: Motion detectors, thermal cameras, sound sensors.

Q5] 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/continuous, single/multi agent.

→ A shopping bot designed for an offline bookstore:

1] Observability:- Partially observable - Limited access to real-time stock updates.

2] Determinism: Stochastic - Book availability and customer interactions are uncertain.

3] Episodic vs Sequential: Sequential - Past interactions influence future recommendations.

4] Static vs Dynamic: Dynamic - Inventory and customer preferences change over time.

5] Discrete vs Continuous: Discrete - Transactions and book selections occur in countable steps.

6] Single vs Multi-Agent:
Multi agent - Interacts with customers, staff, and AI systems.

Q6] Differentiate Model based and Utility based Agent.

Feature	Model-Based Agent	Utility-Based Agent
Definition	Uses an internal model of the environment to make decisions.	Choose actions based on a utility function that evaluates outcomes.
Decision Making	Considers past and current percepts to determine the next action.	Selects the action that maximizes overall utility.
Complexity	Less complex as it follows predefined rules and states	More complex due to the need for utility calculation and comparision.

Feature	Model-Based Agent	Utility Based Agent
Environment Handling	Works well in partially observable environments by maintaining a model.	Focuses on optimizing performance even in uncertain conditions.
Example	A robot vaccum that maps a room layout to navigate efficiently	A self-driving car that chooses routes based on traffic, fuel efficiency, and safety.

Q7] Explain the architecture of a knowledge based agent and Learning agent.

→ A Knowledge-Based Agent (KBA) uses a structured knowledge base to reason and make decisions. It operates using logical inference and stored facts.

Architecture Components:-

1] Knowledge Base (KB):- A collection of facts and rules about the world

2] Inference Engine - Uses reasoning to derive new facts from the knowledge base.

3] Percept Processing Unit - Gathers percepts (input data) from the environment.

4] Query Mechanism - Allows external users or systems to query the knowledge base.

5] Action Selector - Determines the best action based on logical reasoning.

A Learning Agent improves its performance over time by adapting based on past experiences. It can refine

its decision-making process dynamically.
Architecture Components:
1] Learning Element - Improves the agent's knowledge by updating its model based on new experiences.
2] Performance Element - Executes actions based on the agent's current knowledge.
3] Critic - Evaluates the agent's actions and provides feedback
4] Problem Generator - Suggests new experiences for exploration to enhance learning.

Q8] What is AI? Considering the COVID-19 pandemic situation, how AI helped to survive and renovated our way of life with different applications?

→ Artificial Intelligence (AI) is the simulation of human intelligence in machines that can perform tasks like learning, reasoning, problem-solving, and decision making. During the COVID-19 pandemic, AI played a crucial role in healthcare, research, and daily life. AI-powered algorithms helped predict virus spread, analyze medical data, and accelerate vaccine development. Chatbots and virtual assistants provided accurate health information, reducing the burden on healthcare workers. AI based diagnostic tools improved early detection of infections through X-ray and CT scan analysis. In workplaces, AI enabled automation and remote working tools ensured business continuity. E-commerce and delivery services used AI to manage demand supply chain.

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 is not available.
Will Anita travel via Goregaon?
Use forward reasoning.

→ Predicates :-

- Travels (Anita, x) → Anita travels by X.
- Available(X) → Mode of transport X is available.
- Via(X, Y) → Mode of transport X goes via Y.
- Puncture (Car) → Car has a puncture.

Given Information as Predicates:

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

Via (Bus, Andheri) ∧ Via (Bus, Goregaon)

Puncture (Car) → ¬ Available (Car)

Forward Reasoning:-

1] Given Punction (Car), we conclude ¬ Available(Car).
2] Since ¬ Available(Car), from statement (1), Anita will travel by bus.

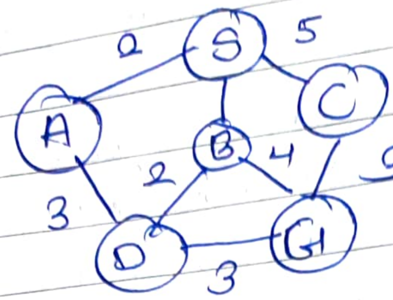
Travels (Anita, Bus)

Via (Bus, Goregaon)

Final Answer:

Yes, Anita will travel via Goregaon.

Q10] Find the route from S to G using BFS



→ 1] Start at S: Add S to the queue.
· Queue [S]

2] Expand S: Visit its neighbors A, B, C
· Queue [A, B, C]
· Parent mapping:
S → A, S → B, S → C

3] Expand A: Visit D.
· Queue [B, C, D]
· Parent Mapping: A → D

4] Expand B: Visit G (goal found).
· Queue [C, D, G]
· Parent Mapping: B → G

Since we found G, we trace back the path using the parent mapping.

Extract the path:-

The Shortest BFS path from S to G is:-

S → B → G

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

→ · Depth-Limited Search (DLS)

Depth-Limited Search is a variant of Depth-First Search (DFS) that explores nodes only upto a specified depth limit. It helps avoid infinite loops in graphs with cycles and is useful when the search space is large.

Algorithm:-

1] Start from root node. 2] Expand nodes depth-wise up to the specified depth limit 3] If the goal is found, return the solution. 4] If not, backtrack and try other branches. 5] If the goal is beyond the limit, it remains undiscovered.

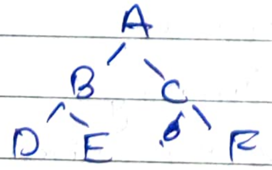
· Iterative Deepening Search (IDS)

Iterative Deepening Search (IDS) is a combination of DFS and BFS. It repeatedly performs Depth-Limited Search, increasing the depth limit with each iteration until the goal is found.

Algorith:

1] Set an initial depth limit d=0. 2] Perform Depth-Limited Search (DLS) with the current depth limit. 3] If the goal is found, return the solution. 4] If not, increase the depth limit (d=d+1) and repeat DLS. 5] Continue until the goal is found or all nodes are explored.

eg:



D=1 Goal = E

· Iteration 1 (Depth (d) = 0) → Explore A

· Iteration 2 (Depth (d) = 1) → Explores A, B, C

Now, Limit is reached but Goal is not satisfied.
we increase depth :
$d = d+1$
$\therefore d = 1+1$
$\therefore d = 2$.

• Iteration 3 (Depth (d) = 2) → Explores A, B, C, D, E, F
Now, the Goal is found in Iteration 3.

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

→ Hill Climbing is a greedy search method that only moves to better states, ignoring bad moves.

Algorithm :
1] Start with an initial state. 2] Evaluate neighbour states using the heuristic function. 3] Move to the neighbour with the best heuristic value. 4] Repeat until :
• The goal is found, or No better neighbour exists.

Example : (Graph Representation)
A (h=5) → B (h=4) → C (h=3) → D (h=2) → Goal (h=0)
Start at A (h=5), Move to B (h=4), Move to C (h=3), Move to D (h=2), Now we choose the next node with the lowest heuristic → Goal (h=0). Reached the Goal.

Final Path :
A → B → C → D → Goal.

DrawBacks of Hill Climbing :
1] Local Maxima - The algorithm may stop at a peak that is not the highest (global maximum).
2] Plateau (Flat region) - If the function has flat region, the

algorithm cannot decide where to move and gets stuck.

3] Ridges - Some solutions require diagonal moves, but hill climbing only moves in one direction (up/down).

Steepest-Ascent Hill Climbing and Its Limitations

It is a variant where instead of moving to any higher state, it moves to the highest-valued neighbour.

Limitations:-

1] Computational Cost - Evaluating all neighbours at each step increases processing time.

2] Sensitive to Noise - Small fluctuations in evaluation can lead to incorrect moves.

3] Local maxima issue - Still suffers from getting stuck at a local peak.

4] Plateau Problem - If all neighbouring states have the same value, the algorithm cannot proceed.

Q13] Explain Simulated annealing with an example.

→ Simulated Annealing is an optimization technique inspired by the annealing process in metallurgy, where materials are heated and then slowly cooled to achieve a stable state with minimal energy. In computational problems, SA is used to find optimal solutions while avoiding local minima or maxima by occasionally accepting worse solutions.

Algorithm:-

1] Start with an initial solution and a high temperature T.

2] Generate a neighbouring solution and calculate its cost.

3] If the new solution is better, accept it or if the new solution is worse, accept it with a probability
$P = e^{-\Delta E/T}$

4] Gradually decrease the temperature using a cooling factor. 5] Repeat until the temperature is near zero or no improvements occurs.

Example:
Imagine you're a hiker trying to reach the lowest point in a mountain range.

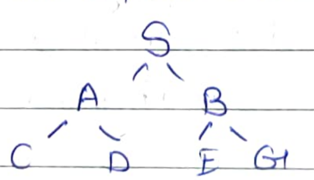
· Hill Climbing moves only downhill, getting stuck in the first valley. Simulated Annealing allows random jumps, so the hiker might climb a hill first before finding a deeper valley, eventually reaching the lowest possible point.

Q14] Explain A* Algorithm with an example.

→ The A* Algorithm is a graph traversal technique used to find the shortest path from a start node to a goal node. It uses a function.

$f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost from the start node, and $h(n)$ is the estimated cost to the goal. A* selects the node with the lowest $f(n)$ value, balancing efficiency and accuracy.

Eg:



Cost values:

· $g(n)$ = actual cost from S.
· $h(n)$ = estimated cost to G.

Node	g(n)	h(n)	f(n) = g(n) + h(n)
S	0	6	6
A	2	4	6
B	3	2	5
C	5	3	8
D	6	2	8
E	5	1	6
G	6	0	6

A* Algorithm Execution.

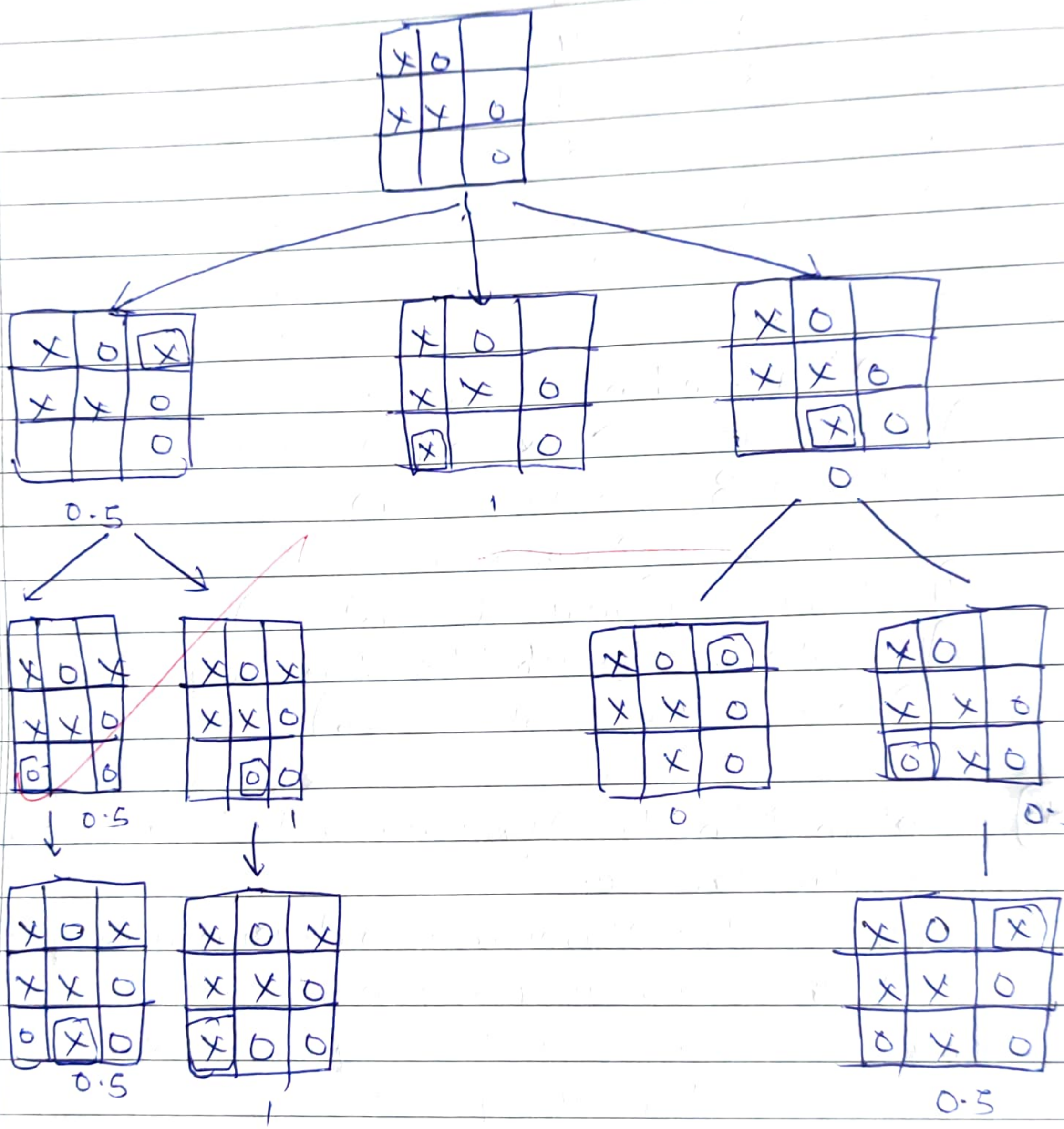
1) Start at S → Expand A and B.
2) Choose B (smallest f(n) = 5).
3) Expand B → Consider E and G.
4) Choose G (smallest f(n) = 6)
5) Goal reached with the shortest path:
S → B → G (Total cost = 6)

Q15) Explain Min Max Algorithm and draw game tree for Tic Tac Toe.

→ The MiniMax Algorithm is a decision making algorithm used ~~to~~ in two-player games like Tic-Tac-Toe, Chess, and checkers. It evaluates game states and chooses the optimal move by assuming both players play optimally.

- The Maximizer (e.g. "X") tries to maximize the score
- The Minimizer (e.g. "O") tries to minimize the score.
- A game tree is generated, where each node represents a possible board state.
- The algorithm recursively explores all moves until reaching a terminal state (win, lose, or draw).

Game Tree of Tic-Tac-Toe



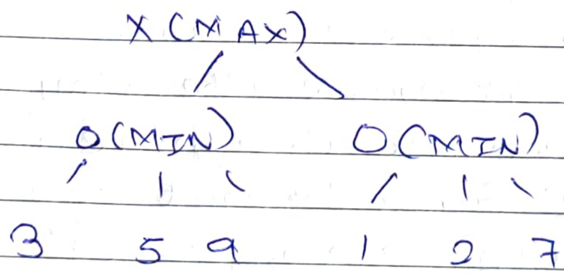
Q10] Explain Alpha beta pruning algorithms for adversarial search with example.

→ Alpha-beta Pruning is an optimization technique for the MiniMax Algorithm that reduces the number of nodes evaluated in a game tree. It eliminates branches that won't affect the final decision, making the algorithm faster without changing the result.

- Alpha (α): The best value the Maximizer can guarantee.
- Beta (β): The best value the Minimizer can guarantee.
- Nodes are pruned (ignored) if further evaluation cannot improve the decision.

Example: Tic-Tac-Toe (Game Tree)

Consider a game tree where X (Max) and O (Min) are playing:



Step 1: X checks the leftmost branch. O selects 3 as the minimum.

Step 2: X explores the next branch; O sees 5, so X chooses max (3,5) = 5.

Step 3: While checking the right subtree, if O finds a value ≤ 5, it prunes (ignores) the rest because X already has a better option.

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

→ The Wumpus world is a grid-based AI environment used for reasoning in uncertain situations. It consists of 4x4 grid where an agent (AI) must navigate to find gold while avoiding dangers like the Wumpus (a monster) and pits.

PEAS

Performance:- +1000 for grabbing gold, -1000 for falling into a pit or Wumpus, -1 for each move, -10 for using an arrow.

Environment:- 4x4 grid, with pits, a Wumpus, and a gold piece. The agent starts at (1,1) facing right.

Actuators:- Move forward, turn left/right, grab gold, shoot arrow, Climb out.

Sensors:- Breeze (near pits), Stench (near Wumpus), Glitter (gold nearby), Bump (wall hit), Scream (Wumpus Die)

· Percept Sequence Generation.

The percept sequence is the history of sensors readings as the agent moves through the world.

Example:- Suppose the agent moves from (1,1) → (1,2) → (2,2) → (3,2):-

- (1,1) → No percepts (safe square).
- (1,2) → Breeze (pit nearby).
- (2,2) → Stench (Wumpus nearby).
- (3,2) → Glitter (goal found).

Using this information, the agent makes logical inferences to avoid danger and reach the goal.

Q19] Solve the Following Crypto-arithmetic problems

1.] SEND + MORE = MONEY.

→ a] Assigning Letters to Digits

Each letter in the problem represents a unique digit

· S, E, N, D, M, O, R, Y unique digits from 0 to 9.

SEND
+ MORE
MONEY

Since MONEY is a 5-digit number, M must be 1 because otherwise, the sum would not produce a five-digit result.

b] Setting up Column-wise Addition.

Column 1: Units Place (D + E = Y or Y+10 if there's a carry)

· The rightmost column of digits in SEND and MORE must sum to Y, possibly with a carry to the next column.

· Column 2: Tens Place (N + R + Carry1 = E or E+10)

· The sum of N and R, plus any carry from the previous column, must result in E.

· Column: Hundreds Place (E + O + Carry2 = N or N+10)

· The sum of E and O, plus any carry from the previous column, must result in N.

· Column 4: Thousands Place (S + M + Carry3 = O or O+10)

· Since M is already determined to be 1, this equation simplifies

c] Assigning Logical values to letters

1] M = 1 [as determined earlier]

2] Checking S+M = O, we substitute S+1 = O

· The maximum value of S can be 9, so O = 0

3] Checking E + O + Carry = N, we set E + O = N, meaning N = E.

4] Checking N + R + Carry = E, we set N + R = E

5]. Checking D + E = Y, Solve for Y.

After testing possible values:

Letter	Value
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 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.

→ 1] $\forall x\,(\text{Graduating}(x) \rightarrow \text{Happy}(x))$
$\forall x\,(\text{Happy}(x) \rightarrow \text{Smiling}(x))$
$\exists x\,(\text{Graduating}(x))$

2]
a] Convert $\text{Graduating}(x) \rightarrow \text{Happy}(x)$
· Using implication removal:-
$\neg \text{Graduating}(x) \vee \text{Happy}(x)$
$\neg \text{Happy}(x) \vee \text{Smiling}(x)$
· Introduce a constant a
$\text{Graduating}(a)$

3] We need to prove $\exists x$, Smiling(x). This means we must derive Smiling (a) using the resolution method.

a] $\neg$ Graduating(x) $\vee$ Happy(x)

b] $\neg$ Happy(x) $\vee$ Smiling(x)

c] Graduating (a)

Resolve (c) and (a)

- Substituting $x=a$ in (a)

$\neg$ Graduating (a) $\vee$ Happy (a)

Happy(a)

Resolve (b) and (a)

- Substituting $x=a$ in (b)

$\neg$ Happy (a) $\vee$ Smiling (a)

Smiling (a)

Since we derived Smiling (a), we have proved that someone is smiling.

Resolution Tree

(Graduating(a))   (Clause c)

|

($\neg$ Graduating(x) $\vee$ Happy(x))   (Clause a)

|

(Happy(a))

|

($\neg$ Happy(x) $\vee$ Smiling(x))   (Clause b)

|

(Similing (a)) ✓

Q20] Explain Modus ponen with suitable examples

→ Modus Ponens is a fundamental rule of inference in propositional logic. It states that if we have:

1] A conditional statement: $P \to Q$ (if P then Q)

2] The antecedent is true: P (P is true)

Then, we can conclude that Q (Q must also be true).

This rule is formally written as:

$$\frac{P \to Q \qquad P}{\therefore Q}$$

Example of Modes Ponens

1] Logical Argument

a] If it rains, the ground will be wet. $(P \to Q)$

b] It is raining. (P)

c] Therefore, the ground is wet. (Q)

"If it rains, the ground will be wet", "It is raining"

∴ "The ground is wet"

2] Mathematical Form:

a] If a number is even, then it is divisible by 2.

· $(P \to Q$, where P = "Number is even", Q = "Divisible by 2")

b] The number 8 is even.

· (P is true)

c] Therefore, 8 is divisible by 2

· (Q is true)

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

→ Forward Chaining is a data driven approach. It starts with known facts and applies rules to derive new facts until the goal is reached.

Algorithm for Forward Chaining.

1] Start with known facts in a knowledge base.

2] Check if any rule's premises (conditions) match the known facts.

3] Apply the rule to infer new facts.

4] Repeat steps 2-3 until the goal is reached or no new facts can be inferred.

Example.

Knowledge Base (Rules):

1] If it is raining, the ground is wet. → Rain (x) → WetGround(x).

2] If the ground is wet, the roads are slippery. → WetGround(x) → SlipperyRoad(x)

3] It is raining. → Rain (A)

Inference:

· Given Rain (A), rule 1 implies WetGround (A).

· Since WetGround(A) is now true, Rule 2 implies SlipperyRoad(A)

· Conclusion : The roads are slippery.

Backward Chaining is a goal-driven approach. It starts with a goal and works backward to check if known facts can support it.

Algorithm for Backward Chaining:

1] Start with the goal (query).

2) Check if the goal is directly known (a fact). If yes, return true.

3) If not, check if any rule concludes the goal.

4) Recursively verify if the rule's premises are true.

5) If all premises are true, conclude the goal, otherwise fal[se]

Example:-

Query:- Are the roads slippery?
(SlipperyRoad(A))

Inference Process-

- SlipperyRoad(A) requires WetGround(A) (Rule 2).
- WetGround(A) requires Rain(A) (Rule 1).
- We check if Rain(A) is a fact in the knowledge base. Since it is, we conclude:-
  - WetGround(A) is true → SlipperyRoad(A) is true.