III B.Tech I Semester Supplementary Examinations, July/August 2021

ARTIFICIAL INTELLIGENCE

(Common to CSE & IT)

Time: 3 hours **Max Marks: 70**

Answer any FIVE questions. All questions carry equal marks

5 * 14 = 70 Marks

1. **(a)** Differentiate between the depth first search and breadth first search with the help of suitable example. **[14]**

   (b) Explain the concept of utility-based agent with an example.

2. **(a)** Justify the need for computable functions and predicates in logic. **[14]**

   (b) How forward chaining is different from backward chaining inference method?

3. **(a)** Explain Bayesian networks with example. **[14]**

   (b) How do AI systems deal with uncertainty? Explain with an example.

4. **(a)** Explain advantages and disadvantages of Hidden Markov Model. **[14]**

   (b) Explain general model of learning agent with neat sketch.

5. **(a)** Discuss about the information needed for visual control of a vehicle on a freeway. **[14]**

   (b) Define robot and explain properties of environments.

6. **(a)** Explain about A* algorithm in detail. **[14]**

   (b) Explain wumpus world with suitable example.

7. **(a)** Explain conditional probability with example. **[14]**

   (b) What is supervised learning. Explain decision tree with example.

8. **(a)** What are the desirable properties of knowledge representation? **[14]**

   (b) How do you represent knowledge in an uncertain domain? Explain.

*****

1. (a) Differentiate between Depth First Search (DFS) and Breadth First Search (BFS) with a suitable example.

Depth First Search (DFS):

Strategy: DFS explores a graph by diving deep into one branch before backtracking. It continues exploring deeper into the graph until it reaches a dead end (a node with no children or a goal node).

Data Structure Used: DFS uses a stack data structure (either explicitly or via recursion). A stack stores the nodes to visit, and the most recently visited node is explored next.

Memory Usage: Since DFS only stores a single path, it uses less memory compared to BFS.

Completeness: DFS may not always find a solution, especially in infinite or cyclic graphs. It may get stuck in an infinite loop without finding the goal.

Optimality: DFS does not guarantee the shortest path in an unweighted graph. It might find a solution, but not the best one.

Example: In a graph with nodes A, B, C, D, and E:

A → B → D

A → C → E

DFS might explore A → B → D before it explores A → C → E, missing the shorter path (A → C → E).

Breadth First Search (BFS):

Strategy: BFS explores the graph level by level, first visiting all nodes at distance 1 from the start, then all nodes at distance 2, and so on.

Data Structure Used: BFS uses a queue (FIFO) to manage nodes. Nodes are explored in the order they are added to the queue.

Memory Usage: BFS can consume more memory because it stores all nodes at each level simultaneously.

Completeness: BFS always finds a solution if one exists in a finite graph, as it explores all possible nodes in order.

Optimality: BFS guarantees the shortest path in an unweighted graph since it expands nodes level by level.

Example: Given the same graph, BFS would explore nodes in the following order: A → B → C → D → E. It finds the shortest path to E or D without missing any optimal solution.

1. (b) Explain the concept of utility-based agent with an example.

A Utility-Based Agent is an agent that makes decisions based on a utility function. The utility function quantifies the satisfaction or "happiness" of an agent's state, helping it to make decisions that maximize its overall well-being, even when the agent faces multiple options.

<u>Utility Function</u>: The utility function assigns a numerical value to each possible state, with higher values indicating more preferred states.

<u>Action Selection</u>: The agent will choose actions that maximize its expected utility, considering current and future states.

Example: Imagine a robot vacuum cleaner tasked with cleaning a house.

It can:

Move left, right, or clean a specific spot.

Its utility function might be defined as:

1 for cleaning a dirty spot.

-1 for using too much battery.

-0.5 for encountering an obstacle. The robot uses this utility function to determine which action to take next, always aiming to maximize the cleanliness of the house while minimizing battery usage and avoiding obstacles.

This kind of agent balances multiple goals (cleaning, battery conservation, and obstacle avoidance) by evaluating the utility of each action.

2. (a) Justify the need for computable functions and predicates in logic.
In AI, logic is crucial for representing knowledge and making inferences. The use of computable functions and predicates enables AI systems to manipulate logical statements effectively.
Predicates:
A predicate represents a relationship or property about objects. It's a statement that can be true or false (e.g., "isPrime(5)" checks if 5 is prime).In AI, predicates are used to represent facts and rules about the world. They help the agent reason about the world and perform logical deductions (e.g., "isEven(x)" could represent whether a number is even).
Computable Functions:
A function computes a value based on input. In logic, functions enable the agent to perform arithmetic, evaluate conditions, and manipulate data.For example, a function like "sum(x, y)" computes the sum of two numbers, which helps the agent solve problems requiring arithmetic.
Justification:
Computable functions and predicates make AI reasoning possible by formalizing knowledge and allowing the agent to apply logical operations.
Inference and reasoning rely on these tools, as they enable the agent to deduce new facts or make decisions based on known facts.
2. (b) How forward chaining is different from backward chaining inference method?
Forward Chaining:
Direction: Data-driven (bottom-up approach).Process: Starts with known facts and applies inference rules to generate new facts until the goal is reached.Usage: Often used in rule-based systems where the agent uses its current knowledge to derive more facts.Example: If the facts are "It rains" and "Rain causes puddles," forward chaining would infer "There are puddles."
Backward Chaining:
Direction: Goal-driven (top-down approach).Process: Starts with a goal (hypothesis) and works backward through inference rules to determine what facts need to be true to support the goal.Usage: Used in expert systems or in situations where the agent is trying to prove a hypothesis.
Example: If the goal is "There are puddles," backward chaining will check if "It rains" is true and if "Rain causes puddles" holds.
Key Difference: Forward chaining is data-driven, expanding known facts, while backward chaining starts from a goal and tries to prove or disprove it using available facts.

3. (a) Explain Bayesian networks with an example.

Bayesian Networks are probabilistic graphical models that represent a set of variables and their conditional dependencies. Each node in the network represents a variable, and directed edges represent dependencies between variables. Bayesian networks are used for reasoning under uncertainty.

Nodes: Represent random variables (e.g., symptoms, weather conditions)

Edges: Represent conditional dependencies between variables.

Conditional Probability Tables: Each node has a conditional probability table that quantifies the likelihood of a variable given its parents.

<u>Consider a simple medical diagnosis system with the following variables:</u>

Rain (R): Whether it rains today.

Sprinkler (S): Whether the sprinkler is on.

Wet Grass (W): Whether the grass is wet.

The dependencies might be:

Rain and sprinkler independently affect whether the grass is wet.

The probability of wet grass is conditional on whether it rained or the sprinkler was on.

The Bayesian network would look like this:

R → W ← S

From the network, we can calculate the probability of the grass being wet given that we know whether it rained or the sprinkler was on using Bayes' theorem.

3. (b) How do AI systems deal with uncertainty? Explain with an example.

AI systems deal with uncertainty using models and techniques such as probabilistic reasoning, fuzzy logic, and belief networks.

<u>Probabilistic Reasoning</u>: Systems like Bayesian networks use probability distributions to represent uncertainty and make predictions based on incomplete information.

<u>Fuzzy Logic</u>: Fuzzy systems allow the modeling of uncertainty by representing variables with degrees of truth rather than binary values (e.g., “temperature is warm” can be 0.7 true rather than just true or false).

<u>Example</u>: In a medical diagnosis system, uncertainty arises due to incomplete or noisy data. For instance, symptoms might not perfectly indicate a particular disease. A Bayesian network can help by providing probabilities for different conditions, taking into account the uncertainty of the evidence (e.g., the likelihood of a disease given the presence or absence of certain symptoms).

4. (a) Explain advantages and disadvantages of Hidden Markov Model (HMM).

Advantages of HMM:

Probabilistic Model: HMMs provide a framework for dealing with uncertainty in sequential data by modeling the system as a series of hidden states and observations.

Temporal Modeling: HMMs are great for time-series or sequential data where past states influence future states, such as speech recognition or weather forecasting.

Efficiency: The Viterbi algorithm allows for efficient decoding of hidden states.

Disadvantages of HMM:

Assumption of Independence: HMM assumes that the current state depends only on the previous state (Markov property), which might not always hold in complex systems.

Parameter Estimation: Training HMMs requires a large amount of data, and incorrect estimation of model parameters can degrade performance.

Computational Complexity: For large state spaces or long sequences, HMMs can become computationally expensive.

4. (b) Explain general model of learning agent with a neat sketch.

A Learning Agent consists of the following components:

Performance Element:

The agent interacts with the environment and takes actions based on its current state.

Learning Element:

It uses feedback from the environment to improve its performance over time.

Critic:

The critic evaluates the actions taken by the agent and provides feedback on whether the actions were good or bad.

Problem Generator:

The problem generator suggests new actions that might lead to better performance or exploration of new solutions.

Sketch:

The agent interacts with the environment, receives feedback (critic), and uses the learning element to adapt.

The learning element adjusts the agent's actions to improve future performance.

5. (a) Discuss the information needed for visual control of a vehicle on a freeway.

For visual control of a vehicle on a freeway, the following information is required:

Lane Detection: Identifying the boundaries of the lane and the vehicle's position within it.

Traffic Signs: Recognizing speed limits, exits, and other road signs.

Obstacles and Vehicles: Detecting other vehicles, pedestrians, or obstacles on the road.

Road Conditions: Detecting wet, icy, or damaged road surfaces.

Distance and Speed: Estimating the distance from other vehicles and calculating the relative speed to maintain a safe distance.

5. (b) Define robot and explain properties of environments.

Robot: A robot is an autonomous or semi-autonomous machine capable of performing tasks in the real world, typically equipped with sensors for perception and actuators for action.

Properties of Environments:

Observable: Whether the environment can be fully observed by the agent at any given time.

Static vs. Dynamic: Whether the environment changes over time, either by external forces or other agents.

Discrete vs. Continuous: Whether the environment's state changes in discrete steps or continuously.

Deterministic vs. Stochastic: Whether the environment’s behavior is predictable or involves uncertainty.

Episodic vs. Sequential: Whether the agent’s actions are independent or whether they depend on past actions.

6. (a) Explain A algorithm in detail.*

The A* (A-star) algorithm is one of the most popular and widely used algorithms in artificial intelligence for pathfinding and graph traversal. It is used to find the shortest path from a start node to a goal node in a weighted graph, ensuring optimality and efficiency. A* is a best-first search algorithm that combines aspects of Dijkstra's algorithm (which focuses on exploring the shortest paths) and greedy search (which looks for promising paths using a heuristic function).

<u>Key Features of the A Algorithm:*</u>

Optimality: A* guarantees finding the shortest path, given that the heuristic function is admissible (i.e., it never overestimates the actual cost to reach the goal).

Completeness: A* will always find a solution if one exists, as long as the graph is finite and the cost function is non-negative.

Efficiency: A* is more efficient than algorithms like Dijkstra's because it uses a heuristic to prioritize nodes that are likely to lead to the goal.

<u>Heuristic Function:</u>

h(n) represents an estimate of the cost from node n to the goal.

<u>Cost Function:</u>

g(n) represents the cost to reach node n from the start node.

<u>Total Cost Function:</u>

f(n)=g(n)+h(n) is the sum of the cost so far and the estimated cost to the goal.

A* expands nodes with the lowest f(n) value, ensuring it explores the most promising paths first.

6. (b) Explain Wumpus world with a suitable example.

The Wumpus World is a well-known example used in the study of artificial intelligence (AI) to model a simple, grid-based environment in which an agent operates. It is used to demonstrate decision-making under uncertainty, reasoning, and the use of sensors and actions. The Wumpus World is typically represented as a grid (usually 4x4), where the agent must navigate, avoiding hazards, to reach a goal or retrieve an object.It consists of:<u>Wumpus</u>: A dangerous creature that kills the agent if it enters its square.<u>Pits</u>: Locations where the agent can fall to its death. <u>Gold</u>: The agent's goal is to find and grab the gold.<u>Breeze and Stench</u>: Percepts that indicate nearby pits or the presence of the Wumpus.

<u>Example</u>: In a 4x4 grid, the agent starts in a safe square. The grid may have some cells containing a Wumpus or pits, and the agent needs to infer their locations based on percepts like breeze and stench to safely navigate and retrieve the gold.

7. (a) Explain conditional probability with example.
Conditional Probability: The probability of an event A given that another event B has occurred is represented as $P(A \mid B)$. It quantifies the likelihood of event A occurring under the condition that event B has happened.
Formula:
$P(A \mid B)=P(A\cap B) / P(B)$ Where $P(A\cap B)$ is the joint probability of both A and B, and P(B) is the probability of B.
Example: If a deck of cards is shuffled, the probability of drawing an Ace (event A) given that the card drawn is a spade (event B) is:
P(Ace|Spade)= P(Ace and Spade) / P(Spade)
= (1/52)/(1/4)
=1/13

7. (b) What is supervised learning? Explain decision tree with example.
Supervised Learning: In supervised learning, the model is trained using labeled data, where the input data comes with known output labels. The goal is for the model to learn a mapping from inputs to outputs based on the training data.
Decision Tree: A decision tree is a tree-like structure used for classification or regression tasks. Each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents the predicted class or value.
<u>Example</u>: Given a dataset of weather conditions (temperature, humidity, wind speed) and whether people played tennis, a decision tree might use features like "humidity" and "wind speed" to predict whether people will play tennis or not.

8. (a) What are the desirable properties of knowledge representation?

Desirable Properties:

Representational Adequacy: The system must be able to represent all relevant facts, relationships, and concepts.

Inferential Adequacy: The system must allow for logical inferences to be drawn from the represented knowledge.

Computational Efficiency: The system should be efficient in terms of both time and space to store and retrieve knowledge.

Expressiveness: The system should have the ability to represent complex, abstract concepts.

8. (b) How do you represent knowledge in an uncertain domain? Explain.

In an uncertain domain, knowledge is represented using techniques that account for ambiguity or incomplete information. These techniques include:

Probabilistic Models: Bayesian networks and Markov models represent uncertainty by assigning probabilities to variables and their relationships.

Fuzzy Logic: Fuzzy logic allows for partial truth values, where a statement can be true to a degree, not just true or false.

Belief Networks: A belief network uses probability to model the uncertainty in the belief of an agent about the world, especially when knowledge is incomplete or uncertain.

Example: In medical diagnosis, a Bayesian network can model the probability of various diseases given symptoms, allowing for decisions based on uncertain evidence.