DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE
ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA
EVEN SEM 2025-26
CA1 Examination

Course: B. Tech. **Class: B.Tech** **Semester: VII**
Branch: C.S.E. **Subject Code & Name: BTCOC701 Artificial Intelligence**
Max Marks: 30 **Date:** **Duration: 01:30 Hrs**

Instructions to the Students:
(**1**) Illustrate your answers with neat sketches wherever necessary.
(2) Figures to the right indicate full marks.
(3) Assume suitable data if necessary.
(4) Preferably, write the answers in sequential order.

Q.1 Objective type questions. (All questions are compulsory)

1. Good behavior in agents is measured by: CO1 1 Mark
a) Randomness in actions b) The concept of rationality c) Physical size of the agent d) Speed of response only

2. Which environment is considered most difficult for an AI agent to operate in? CO1 1 Mark
a) Fully observable, deterministic environment b) Partially observable, stochastic environment
c) Static, episodic environment d) Simple, discrete environment

3. The structure of an agent typically includes:
a) Only sensors and actuators b) Architecture and program
c) Hardware components only d) Database and compiler CO1 1 Mark

4. Constraint propagation in CSPs refers to: CO2 1 Mark
a) Expanding the search tree b) Using inference to reduce the domain of variables before or during search
c) Ignoring constraints to speed up the solution d) Running random assignments

5. Backtracking search for CSPs works by: CO2 1 Mark
a) Assigning values to all variables simultaneously
b) Incrementally building candidates and abandoning when constraints are violated
c) Ignoring conflicts and proceeding to the next step d) Using only local optimization

6. Local search algorithms for CSPs are particularly useful when: CO2 1 Mark
a) The state space is continuous or very large b) The solution requires full search tree exploration
c) There are no constraints at all d) Only uninformed strategies are applicable

Q .2 Solve Any two of the following.

A. Define rationality in detail & discuss the impact of different types of environments in terms of deterministic vs. stochastic, fully vs. partially observable on rational decision-making. CO1 6 Marks

Ans:
Definition
 Rationality refers to the ability of an intelligent agent to make decisions that maximize its performance measure, given the knowledge it has, the percepts it receives, and the actions available.

 A **rational agent** is one that does *the right thing* — meaning it chooses the action that is expected to achieve the best outcome (according to its performance measure).

Factors Affecting Rationality
Rationality depends on four main factors:
1. **Performance Measure** → Defines success (e.g., shortest time, maximum cleanliness, safety).
2. **Percept Sequence** → What the agent has perceived so far.
3. **Knowledge of the Environment** → What the agent knows about how the world works.
4. **Available Actions** → What actions the agent can perform.

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE
ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA
EVEN SEM 2025-26
CA1 Examination

Impact of Environment Types on Rational Decision-Making
The nature of the environment greatly influences how rational decisions are made. Two important dimensions are:
1. Deterministic vs. Stochastic Environments
 Deterministic Environment
o Next state of the environment is *completely determined* by the current state and agent’s action.
o Example: A chess game (no randomness — moves fully decide the outcome).
o **Impact on Rationality**:
 Easier for an agent to act rationally → because outcomes of actions are predictable.
 Rational strategy: Plan ahead and compute exact consequences.
 Stochastic Environment
o Environment has *randomness* or uncertainty in outcomes.
o Example: Driving in traffic (other drivers’ behavior, weather, accidents).
o **Impact on Rationality**:
 Agent must deal with probabilities and uncertainties.
 Rational strategy: Choose actions that **maximize expected utility** instead of guaranteed outcomes.
 Requires probabilistic reasoning (Bayesian networks, Markov decision processes).

2. Fully Observable vs. Partially Observable Environments
 Fully Observable Environment
o Agent’s sensors can perceive the *entire state of the environment*.
o Example: Tic-Tac-Toe (the whole board is visible).
o **Impact on Rationality**:
 Easier to be rational since the agent has complete information for decision-making.
 No need to guess or maintain hidden state.
 Partially Observable Environment
o Agent’s sensors provide *incomplete or noisy information* about the environment.
o Example: Poker (opponent’s cards are hidden), autonomous driving (fog hides pedestrians).
o **Impact on Rationality**:
 Agent must maintain an **internal state or belief model** to represent hidden/unobserved aspects.
 Rational decisions involve estimating probabilities of unseen states.
 Example: Self-driving car uses sensors + prediction models to estimate unseen cars/pedestrians.

B. Define and explain the functionality of a goal-based agent. CO1 6 Marks
Ans : **Idea**: "Action depends on achieving specific goals."
 Structure:
o **Sensors** → perceive environment.
o **Internal Model (State Information)**.
o **Goals** → describe desired situations or outcomes.
o **Search/Planning Module** → selects actions that help achieve goals.
o **Actuators** → perform chosen actions.
 Working:
o Uses reasoning: *Given my current state and goal, what sequence of actions will achieve it?*
o Example: A robot delivery agent.
 Goal: Deliver package to Room 102.
 Uses pathfinding (A*, Dijkstra’s) to choose route.
 Advantages:
o More flexible than reflex agents.
o Can pursue new goals dynamically.

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE
ARVIND GAVALI COLLEGE OF ENGINEERING, SATARA
EVEN SEM 2025-26
CA1 Examination

C. What is PEAS? Draw and explain model based reflex agent. CO1 6 Marks

PEAS stands for **Performance measure, Environment, Actuators, Sensors**. It is a framework used to **define the task environment** of an intelligent agent in a structured way.
When designing an agent, we need to be clear about:
1. **What the agent is trying to achieve** (Performance).
2. **Where it operates** (Environment).
3. **How it can act** (Actuators).
4. **What it can perceive** (Sensors).

1. Performance Measure (P)
 Defines the **criteria for success**.
 Tells us how we evaluate whether the agent is doing a good job.
 Examples:
o For a self-driving car → safety, speed, fuel efficiency, passenger comfort.
o For a chess-playing agent → winning the game.

2. Environment (E)
 The **external context** in which the agent operates.
 It can be fully observable or partially observable, deterministic or stochastic, static or dynamic, etc.
 Examples:
o For a vacuum cleaner → rooms, dirt, obstacles, charging dock.
o For a medical diagnosis agent → hospital database, patient information.

3. Actuators (A)
 The **tools through which the agent acts** on the environment.
 Examples:
o For a robot → wheels, arms, grippers, speakers.
o For a software agent → display messages, send commands, update logs.

4. Sensors (S)
 The **devices or mechanisms** that let the agent perceive the environment.

 Examples:
o For a robot → cameras, microphones, GPS, LiDAR.
o For a software agent → input files, user queries, API data.

PEAS Example: Self-Driving Car
 Performance Measure (P): Safety, legality, passenger comfort, fuel efficiency, time to destination.
 Environment (E): Roads, traffic, pedestrians, weather, traffic laws.
 Actuators (A): Steering wheel, accelerator, brake, horn, display.
 Sensors (S): Cameras, radar, LiDAR, GPS, speedometer, microphone

Q. 3 Solve Any two of the following.

A. What is a constraint satisfaction problem (CSP)? Define it and provide examples of CSPs in real-world applications. CO2 6 Marks

B. Prove the following statement with suitable example, Breadth- First-Search is a special case of uniform cost search CO2 6 Marks

Breadth-First Search is a systematic search algorithm that explores a graph (or tree) level by level, starting from the root (or start node). It is an uninformed search strategy (no heuristics), and is widely used because it guarantees an optimal solution (if all step costs are equal). Outline of BFS Algorithm 1. Initialize a queue with the start node. 2. Repeat until the queue is empty: o Remove (dequeue) the first node from the queue. o If it is the goal node, return the solution (success). o Otherwise, expand the node: ♣ Generate all its successors (children). ♣ Add them to the end of the queue (if not already visited). function BFS(start, goal): queue ← [start] visited ← {start} while queue is not empty: node ← queue.pop_front() if node == goal: return "Goal found" for each neighbor in successors(node): if neighbor not in visited: visited.add(neighbor) queue.append(neighbor) return "Goal not found" • Start at S, queue = [S] • Expand S → add A, B, C → queue = [A, B, C] • Expand A → add D → queue = [B, C, D] • Expand B → no goal → queue = [C, D] • Expand C → add G → queue = [D, G] • Expand D → no goal → queue = [G] • Expand G → Goal found! How BFS Provides an Optimal Solution • BFS explores nodes in increasing order of depth (level). • If all actions have the same cost (uniform cost = 1), then the first time BFS encounters the goal node, it is guaranteed to be along the shortest path. • Reason: BFS never skips a shallower node in favor of a deeper one. Example: If path costs are uniform: • BFS will find the shortest path in terms of number of edges. If path costs are not uniform: • BFS is not optimal. (In that case, Uniform Cost Search is used).

C. What is uniform cost search? Discuss how this is different from BFS. CO2 6 Marks

Ans:

Uniform Cost Search (UCS):
Uniform Cost Search is a search algorithm that explores a graph by expanding the node with the **lowest path cost** (g(n)) from the start node. It uses a **priority queue** (min-heap) to always expand the least-cost node first. UCS guarantees finding the **optimal solution** (least-cost path) when the step costs are non-negative.

Difference between UCS and BFS:

Aspect	**Uniform Cost Search (UCS)**	**Breadth-First Search (BFS)**
Cost considered	Expands nodes based on **path cost (g(n))**	Expands nodes level by level (ignores cost)
Optimality	Always finds the **least-cost/optimal path**	Finds the **shortest path only when all step costs are equal**
Data structure	Uses a **priority queue** (min-heap)	Uses a **FIFO queue**
Efficiency	May be slower due to cost calculations	Faster but may not give optimal path if costs vary

---

✅ **In short:**
- BFS = shortest path in terms of number of edges.
- UCS = optimal path in terms of total cost.