

ARTIFICIAL INTELLIGENCE

1st Internal Assignment

16 Questions | 16 Answers | Verified against the supplied 51-page module

PART A - 2 MARK QUESTIONS

Short definition + key point/example. Write 3-5 lines in the exam.

1. Define Artificial Intelligence with an example.

- **Artificial Intelligence (AI)** is a branch of computer science that creates intelligent systems or agents which can perceive their environment, reason, learn and act to achieve a goal. **Example:** Google Maps studies traffic and road conditions to suggest a suitable route and estimated travel time.

2. State any four applications of AI.

- Four applications are:
1. **Medical diagnosis** - assists in analysing symptoms and difficult cases.
 2. **Robotics** - helps robots perform complex and critical tasks.
 3. **Autonomous control** - supports vehicles and machines in controlling their movement.
 4. **Planning and scheduling** - prepares efficient plans for operations and logistics.

3. Mention any four ethical issues in Artificial Intelligence.

- Four ethical issues are: **bias and discrimination**, **privacy and data protection**, **lack of transparency** (black-box decisions), and **accountability** when an AI system causes harm. Safety, job displacement and deepfakes are also important issues.

4. What is an agent? Give an example.

- An **agent** is anything that perceives its environment through **sensors** and acts upon that environment through **actuators**. Example: a self-driving car uses cameras and sensors to observe the road and uses steering, brakes and acceleration to act.

5. Define percept and percept sequence.

- A **percept** is the information received by an agent through its sensors at one particular time. A **percept sequence** is the complete history of all percepts received by the agent up to the current time.

6. What is a rational agent? Give its different types.

- A **rational agent** selects the action expected to maximize its performance measure using its percept sequence and built-in knowledge. Main agent types are **simple reflex**, **model-based reflex**, **goal-based** and **utility-based** agents.

7. List different types of task environment.

- Task environments are classified as: **fully/partially observable**, **deterministic/stochastic**, **episodic/sequential**, **static/dynamic**, **discrete/continuous**, and **single-agent/multi-agent**.



8. Define fringe and leaf node in a search tree.

- The **fringe (frontier)** is the collection of nodes that have been generated but not yet expanded. A **leaf node** is a node in the fringe that currently has no generated successor nodes.

9. What is meant by Uniform Cost Search (UCS)?

- **Uniform Cost Search** is an uninformed search method that always expands the frontier node with the **lowest total path cost $g(n)$** . It uses a priority queue and is complete and optimal when every step cost is positive.

☆ PART B - 5 MARK QUESTIONS ☆

Use definition, main points, example and a short conclusion.

1. Discuss different types of AI with suitable examples.

- AI can be classified into three types according to capability:

Type	Meaning	Example / status
Narrow AI (Weak AI)	Works on one task or a small set of related tasks. It cannot apply its intelligence to a completely different task.	Examples: Google Maps, Siri, face recognition, spam filters and recommendation systems.
General AI (Strong AI / AGI)	A theoretical AI that could understand, learn, reason and solve many kinds of problems like a human.	Example: a future robot that could learn mathematics, cook, drive and learn a new language without a separate program for every task.
Super AI (ASI)	A theoretical AI that would be more intelligent than the best humans in almost every intellectual field.	Status: no confirmed real example. A future system might discover medicines and solve major scientific problems faster than experts.

Conclusion: Present-day systems are Narrow AI. AGI and ASI remain theoretical; their development would require strong safety, control and ethical rules.

2. Explain in detail weak and strong agent.

- **Weak agent:** A hardware- or software-based system that acts intelligently without claiming human-like mental states.
- **Autonomy:** works without direct human intervention and controls its own actions and internal state.
 - **Social ability:** communicates with humans or other agents.
 - **Reactivity:** notices environmental change and responds in time.
 - **Pro-activeness:** takes initiative, creates goals and acts to achieve them.

Strong agent: An agent described as having mental properties such as knowledge, belief, intention and obligation. It may also show mobility, veracity and rationality.

- **Mobility:** can move across machines or platforms.
- **Veracity:** does not knowingly send false information.

- **Rationality:** chooses actions that support its goals.
- Strong AI is also associated conceptually with consciousness, self-awareness, sentience and sapience.

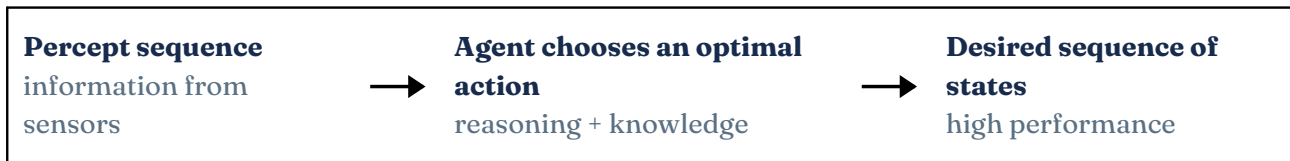
Difference: A weak agent shows useful intelligent behaviour in a limited computational sense; a strong agent is assumed to possess broader mental and human-like qualities.

3. Explain a rational agent with the optimal performance triangle.

→ A rational agent chooses the action expected to produce the best result according to a performance measure. The “right” action is not fixed for all agents because tasks and environments differ.

Rationality depends on four things:

- the performance measure that defines success;
 - the agent's prior knowledge of the environment;
 - the actions available to the agent; and
 - the percept sequence received up to the present.
- **Definition:** For every possible percept sequence, a rational agent selects the action expected to maximize its performance measure, using the evidence in the percept sequence and its built-in knowledge.

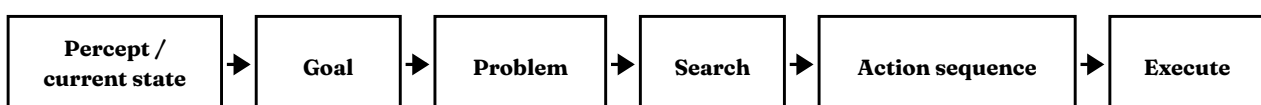


Important: Rationality maximizes expected performance; perfection or omniscience would require knowing the actual outcome in advance, which is impossible in practice.

4. Explain the steps involved in designing a problem-solving agent.

→ A problem-solving agent designs its solution through the following five steps:

1. **Goal setting:** The agent studies the environment and decides what it wants to achieve.
2. **Goal formulation:** It clearly defines the goal by observing the current state and deciding the performance measure.
3. **Problem formulation:** It represents the initial state, possible actions, transition/result model, goal test and path cost.
4. **Search:** A search algorithm explores possible action sequences and returns a solution path.
5. **Execution:** The agent carries out the selected actions. After execution, it can observe the new state and formulate another goal.



PART C - 10 MARK QUESTIONS

Write with definitions, classification, working, examples, merits/limits and conclusion.

1. Discuss in detail different types of task environment.

→ A **task environment** is the problem or world in which an agent operates. Its properties guide the choice of agent design and AI technique.

Fully observable vs partially observable

Fully observable: sensors provide the complete relevant state, e.g., tic-tac-toe. Partially observable: information is missing, hidden or noisy, e.g., poker or driving in fog.

Deterministic vs stochastic

Deterministic: current state and action completely decide the next state, e.g., image analysis.

Stochastic: outcomes contain uncertainty, e.g., road driving. If uncertainty is mainly due to other agents, the environment may be called strategic.

Episodic vs sequential

Episodic: each decision is independent of previous episodes, e.g., checking separate manufactured parts. Sequential: one action affects later decisions, e.g., chess or route planning.

Static vs dynamic

Static: the environment does not change while the agent thinks, e.g., a crossword puzzle. Dynamic: it may change during deliberation, e.g., driving. Semidynamic: the world stays fixed but performance changes with time, e.g., chess with a clock.

Discrete vs continuous

Discrete: states, time, percepts or actions are countable, e.g., tic-tac-toe. Continuous: values and changes can vary continuously, e.g., speed and steering in vehicle control.

Single-agent vs multi-agent

Single-agent: one main agent acts, e.g., boat driving. Multi-agent: several agents cooperate or compete, e.g., fantasy football (cooperative team) or trading (competitive).

Complexity generally increases in the following order:

- Fully observable to partially observable.
- Deterministic to stochastic.
- Episodic to sequential.
- Static to dynamic.
- Discrete to continuous.
- Single-agent to multi-agent.

Example classification - self-driving taxi: partially observable, stochastic, sequential, dynamic, continuous and multi-agent. The car has incomplete sensor information, uncertain road events, linked decisions, continuous motion and many other road users.

2. Explain with neat diagrams: (A) Simple Reflex Agent and (B) Model-Based Reflex Agent.

→ A. Simple Reflex Agent

A simple reflex agent selects an action only from the **current percept**. It uses condition-action rules of the form: IF condition, THEN action. It ignores the earlier percept history.

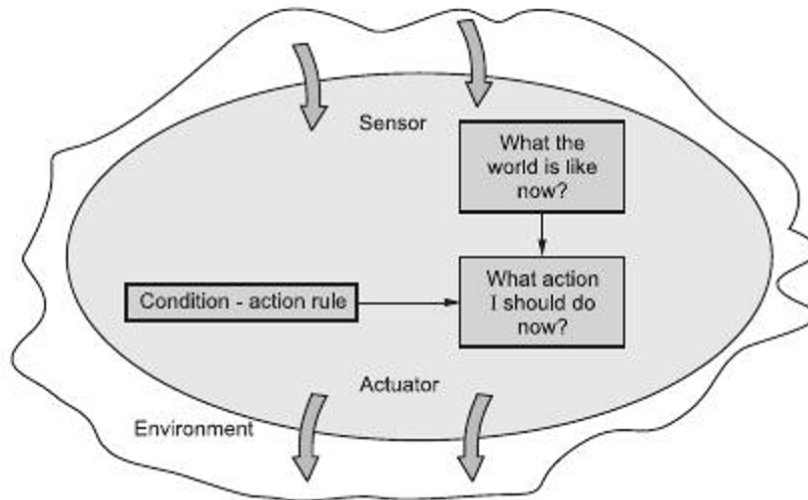


Fig. 1.10.3 Simple reflex agent

Figure 1: Simple reflex agent - diagram screenshot from the supplied module.

Working:

- Sensor receives the current percept from the environment.
- The percept is interpreted as the current state.
- A matching condition-action rule is selected.
- The rule returns an action, and the actuator changes the environment.

Example: In an ATM, IF the entered PIN matches the account, THEN allow the permitted transaction.

Advantages: simple, fast and inexpensive. **Limitations:** works reliably mainly in fully observable environments; missing information may cause wrong actions or loops because it has no memory.

Simple-Reflex-Agent(percept):

1. state = INTERPRET-INPUT(percept)
2. rule = RULE-MATCH(state, rules)
3. action = RULE-ACTION(rule)
4. return action

B. Model-Based Reflex Agent

A model-based reflex agent maintains an **internal state** representing aspects of the world that are not currently visible. It updates this state using the previous state, current percept, last action, a model of how the world changes and the effects of actions.

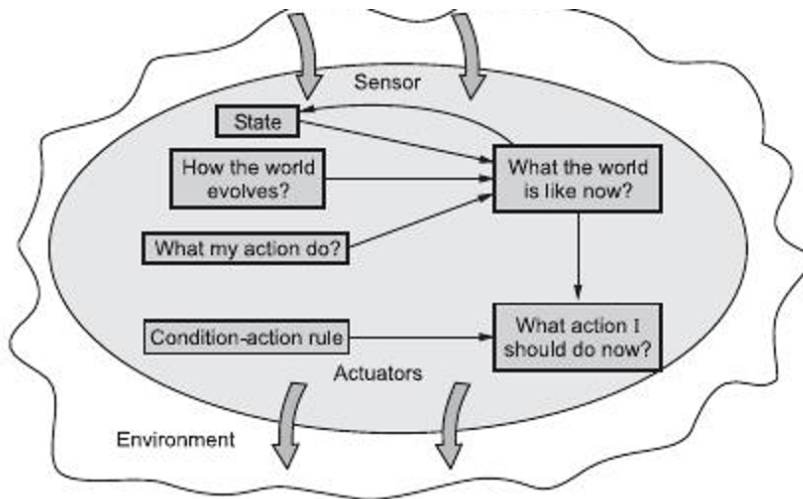


Fig. 1.10.4 Model - based reflex agent

Figure 2: Model-based reflex agent - diagram screenshot from the supplied module.

Working:

- The sensor receives a new percept.
- The agent updates its internal state: old state + current percept + last action + world model.
- It matches the updated state with a condition-action rule.
- The actuator performs the selected action.

Example: A driving agent remembers nearby vehicles that are temporarily hidden and estimates how the road situation may change.

Advantages: handles partially observable environments and uses memory. **Limitation:** performance depends on the accuracy of its internal model.

Model-Based-Reflex-Agent(percept):

1. state = UPDATE-STATE(state, last_action, percept, model)
2. rule = RULE-MATCH(state, rules)
3. action = RULE-ACTION(rule)
4. return action

Main difference: A simple reflex agent reacts only to “what the world is like now”; a model-based reflex agent also remembers and estimates the unseen world.

3. Explain Breadth-First Search with algorithm and suitable example.

→ **Breadth-First Search (BFS)** is an uninformed search strategy that expands the shallowest unexpanded node first. It visits nodes level by level: root, all nodes at depth 1, all nodes at depth 2, and so on.

Data structure: BFS stores the fringe in a **FIFO queue**. New nodes are inserted at the rear and the oldest node is removed from the front.

Algorithm:

BFS(G, n):

1. For $i = 1$ to n , set visited[i] = false.
2. For $i = 1$ to n : if visited[i] = false, call BFS-Visit(i).

BFS-Visit(start):

1. Create an empty FIFO queue Q.
2. Mark start visited and enqueue it.
3. While Q is not empty:
 - a. Dequeue the front node u and visit it.
 - b. For each unvisited neighbour v of u, mark v visited and enqueue v.

Suitable example from the supplied module: D and J are goal states. The numbered circles show the BFS visiting order.

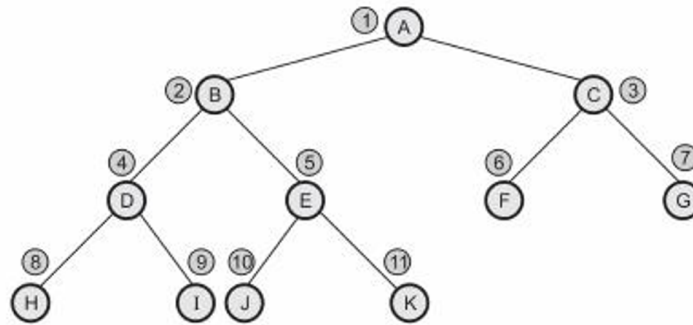


Fig. 2.3.1 Breadth-first search

Figure 3: Original Breadth-First Search tree from the supplied module.

Expansion order: A, B, C, D, E, F, G, H, I, J, K. BFS reaches **D** first through **A -> B -> D**, so it returns the shallowest goal before the deeper goal J.

Property	Result
Completeness	Complete if the branching factor b is finite.
Optimality	Optimal when all actions have equal cost; otherwise the shallowest path may not be the cheapest.
Time complexity	$O(b^{(d+1)})$, where d is the depth of the shallowest goal.
Space complexity	$O(b^{(d+1)})$; memory is the main weakness because BFS stores many frontier nodes.

Advantages: easy to implement, complete, and finds the shortest path in number of steps when step costs are equal.

Disadvantages: requires large memory and becomes expensive when the branching factor or goal depth is large.



Build & Design By : **Astra**

