QUESTION BANK

P23CAE22 — Artificial Intelligence

L:3 T:0 P:0 C:3 | 45 Periods

UNIT I
Introduction to AI

PART A 2-Mark Questions with Answers

Q1. Define Artificial Intelligence. *[CO2 | 2M]*

Answer:

AI is the field of computer science concerned with building machines that simulate intelligent behavior — reasoning, learning, problem-solving, perception, and language understanding.

Four approaches to defining AI: Thinking humanly (cognitive science), Thinking rationally (logicist), Acting humanly (Turing Test), Acting rationally (rational agent — most adopted).

The rational agent approach: AI designs agents that choose actions to maximize expected performance given percepts and knowledge.

Q2. What is the Turing Test and who proposed it? *[CO2 | 2M]*

Answer:

Proposed by Alan Turing in 1950 in 'Computing Machinery and Intelligence'.

A human interrogator communicates via text with a hidden human and machine. If the interrogator cannot reliably identify the machine, it is considered intelligent.

Tests 'acting humanly'. Modern AI has largely moved to the rational agent standard as Turing Test is hard to formalize.

Q3. Define a rational agent. *[CO2 | 2M]*

Answer:

A rational agent selects the action that maximizes its expected performance measure given its percept history and built-in knowledge.

Rationality depends on: performance measure, prior knowledge, actions available, and percept sequence so far.

Rationality does NOT mean omniscience — it means making the best possible decision with available information.

Q4. What is the PEAS framework? Give an example. *[CO2 | 2M]*

Answer:

PEAS = Performance measure, Environment, Actuators, Sensors — used to describe any AI agent completely.

Example (Self-driving car): Performance: safe, fast, legal driving. Environment: roads, traffic, weather. Actuators: steering, brakes, accelerator. Sensors: cameras, LIDAR, GPS, speedometer.

PEAS is the first step in designing any intelligent agent system.

Q5. What are the properties of task environments? *[CO2 | 2M]*

Answer:

Fully/Partially observable, Deterministic/Stochastic, Episodic/Sequential, Static/Dynamic, Discrete/Continuous, Single/Multi-agent.

Example: Chess is fully observable, deterministic, sequential, static, discrete, multi-agent.

Example: Self-driving is partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

Q6. List the five types of agents in AI. *[CO2 | 2M]*

Answer:

1. Simple Reflex: acts on current percept via condition-action rules (Thermostat).
2. Model-based Reflex: maintains internal world model (Self-driving with partial obs.).
3. Goal-based: plans actions to achieve explicit goals (Route GPS).
4. Utility-based: maximizes utility function handling tradeoffs (Trading system).
5. Learning Agent: improves via experience (AlphaGo, modern chatbots).

Q7. Name four disciplines contributing to AI and their contributions. *[CO2 | 2M]*

Answer:

Mathematics: formal logic, probability theory, computation theory — foundations of reasoning.

Neuroscience: neural architecture, synaptic learning — inspiration for neural networks.

Economics: decision theory, utility, game theory — basis of rational decision making.

Linguistics: natural language grammar and semantics — basis of NLP systems.

Q8. What is the concept of rationality vs omniscience? *[CO2 | 2M]*

Answer:

Omniscience: knowing the actual outcome of every action — impossible in reality.

Rationality: making the best decision given available information and expected outcomes.

A rational agent can be wrong — it acts correctly given what it knows, not based on complete knowledge of the world.

Q9. Mention four current state-of-the-art AI achievements. *[CO2 | 2M]*

Answer:

1. AlphaGo/AlphaZero defeated world champions in Chess and Go using MCTS + deep learning.
2. GPT-4 achieves near-human performance in text generation, reasoning, and translation.
3. Tesla/Waymo autonomous vehicles use AI for real-time driving decisions.
4. AI-based medical diagnosis systems detect cancer in radiology with accuracy exceeding human radiologists.

Q10. What are the risks of AI? *[CO2 | 2M]*

Answer:

Economic: job displacement due to automation of routine tasks.

Social: AI bias leading to discrimination in hiring, lending, facial recognition.

Security: autonomous weapons, cyberattacks using AI.

Existential: loss of human control over powerful AI systems (alignment problem).

Privacy: mass surveillance using AI-powered cameras and data analysis.

PART B 16-Mark Questions (Write Answers Below)

Q1. Explain the foundations of AI from various disciplines. Describe the history of AI from 1943 to present, including AI winters and the current deep learning era. *[CO2 | 16M]*

(a) Contributions of Mathematics, Philosophy, Neuroscience, Economics, Psychology, Linguistics to AI. (6M)

(b) Historical milestones: 1943-1970 (birth), 1966-1973 (first winter), 1980s (expert systems), 1990s-2000s (ML), 2010s+ (deep learning). (6M)

(c) State-of-the-art applications: AlphaGo, GPT, autonomous vehicles, medical AI. (4M)

Answer: ______________________________

Q2. Explain the concept of intelligent agents. Describe the PEAS framework and classify task environments with examples. *[CO2 | 16M]*

(a) Intelligent agent definition, percepts, actions, agent function, agent program. (4M)

(b) PEAS for: self-driving car, medical diagnosis system, chess program. (6M)

(c) All six environment properties with formal definitions and real examples. (6M)

Answer: ______________________________

Q3. Explain the five types of AI agents with diagrams. Describe the learning agent model in detail. *[CO2 | 16M]*

(a) Simple reflex and model-based reflex agents with architecture diagrams. (6M)

(b) Goal-based and utility-based agents — how they improve on reflex agents. (4M)

(c) Learning agent: four components (learning element, performance element, critic, problem generator). (6M)

Answer: ______________________________

Q4. What is Artificial Intelligence? Discuss the four approaches to AI. Explain the Turing Test and its limitations. *[CO2 | 16M]*

(a) Four approaches: think humanly, think rationally, act humanly, act rationally — pros and cons. (6M)

(b) Turing Test: description, interrogation scenario, Total Turing Test extension. (4M)

(c) Limitations of Turing Test and why rational agent approach is preferred in modern AI. (6M)

Answer: ______________________________

Q5. Discuss the risks and benefits of AI in modern society. Explain the nature of environments and classify the environment for a game-playing agent and an autonomous vehicle. *[CO2 | 16M]*

(a) Benefits: automation, healthcare, scientific discovery, safety. Risks: job loss, bias, weapons, privacy. (6M)

(b) Game-playing agent: PEAS + environment classification (fully obs., deterministic, sequential, static, discrete, multi-agent). (4M)

(c) Autonomous vehicle: PEAS + environment classification (partial obs., stochastic, sequential, dynamic, continuous, multi-agent). (6M)

Answer: ____

UNIT II
Solving Problem by Searching

PART A 2-Mark Questions with Answers

Q1. Define a search problem and its five components. *[CO3 | 2M]*

Answer:

A search problem consists of: Initial state (s0), Actions (ACTIONS(s)), Transition model (RESULT(s,a)=s'), Goal test (IS_GOAL(s)), Path cost (step cost c(s,a,s')).

The state space graph: all states reachable from the initial state via actions.

A solution is a sequence of actions from initial to goal state; optimal solution has lowest path cost.

Q2. Compare BFS and DFS search strategies. *[CO3 | 2M]*

Answer:

BFS: FIFO queue; Complete (finite b); Optimal (unit cost); Time O(b^d); Space O(b^d) — exponential memory.

DFS: LIFO stack; Not complete (infinite depth/loops); Not optimal; Time O(b^m); Space O(bm) — linear memory.

BFS is better for shallow goals; DFS is memory efficient but can go infinitely deep.

Q3. What is Iterative Deepening Search (IDS)? *[CO3 | 2M]*

Answer:

IDS runs depth-limited search with increasing limits l = 0, 1, 2, ... until goal is found.

Properties: Complete, Optimal (unit cost), Time O(b^d), Space O(bd) — combines BFS guarantees with DFS memory efficiency.

Repeated work is acceptable: only adds ~11% overhead for b=10 despite revisiting earlier nodes.

Q4. Define admissible and consistent heuristics. *[CO3 | 2M]*

Answer:

Admissible: h(n) never overestimates true cost h*(n). h(n) <= h*(n) for all n. Guarantees A* optimality in tree search.

Consistent (monotone): h(n) <= c(n,a,n') + h(n') for every successor n'. Implies admissibility. Guarantees A* optimality in graph search.

Example: Manhattan distance for 8-puzzle is both admissible and consistent.

Q5. What is A* search and when is it optimal? *[CO3 | 2M]*

Answer:

A* expands nodes in order of f(n) = g(n) + h(n) where g = past cost, h = heuristic future cost.

Optimal when h is admissible (tree search) or admissible and consistent (graph search).

A* is complete and optimal — the most widely used informed search algorithm.

Q6. What is uniform cost search? *[CO3 | 2M]*

Answer:

UCS expands the node with the lowest path cost g(n) using a min-priority queue.
Complete and optimal — always finds minimum-cost path. Equivalent to Dijkstra's algorithm.
Time/Space: $O(b^{(1 + C^*/e)})$ where C* is optimal cost and e is minimum step cost.

Q7. Differentiate greedy best-first search from A*. *[CO3 | 2M]*

Answer:

Greedy best-first: f(n) = h(n) only — expands node closest to goal by estimate. Fast but not optimal and not complete.
A*: f(n) = g(n) + h(n) — considers both cost incurred and estimate to goal. Complete and optimal.
Greedy may find a suboptimal path quickly; A* finds the optimal path more slowly.

Q8. What is bidirectional search? *[CO3 | 2M]*

Answer:

Bidirectional search runs two simultaneous searches: one forward from initial state, one backward from goal.
They meet in the middle. Time and space: $O(b^{(d/2)})$ — much better than $O(b^d)$ for BFS.
Challenge: backward search requires reverse actions; detecting when frontiers meet is complex.

Q9. Define node, frontier, and explored set in search. *[CO3 | 2M]*

Answer:

Node: data structure holding state, parent, action, path-cost g, and depth.
Frontier (open list): nodes generated but not yet expanded — available for exploration.
Explored set (closed list): states already expanded — prevents revisiting and cycles.

Q10. What are heuristics for 8-puzzle? Which dominates? *[CO3 | 2M]*

Answer:

h1 (misplaced tiles): count of tiles not in goal position. Admissible — a tile needs at least one move.
h2 (Manhattan distance): sum of horizontal + vertical distances of each tile from its goal. Admissible and more informed.
h2 dominates h1: h2(n) >= h1(n) for all n (Manhattan >= number of misplaced). h2 expands fewer nodes.

PART B 16-Mark Questions (Write Answers Below)

Q1. Explain all uninformed search strategies: BFS, DFS, Depth-Limited, IDS, and Uniform Cost Search. Compare them in a table. *[CO3 | 16M]*

(a) BFS: queue, algorithm, properties, example trace on simple graph. (4M)
(b) DFS and Depth-Limited: stack, algorithm, problems with completeness. (4M)
(c) IDS: algorithm, why it's preferred, time overhead analysis. (4M)

(d) UCS: priority queue, algorithm, comparison table for all five strategies. (4M)

Answer: ____

Q2. Explain A* search in detail. Prove its optimality. Illustrate with the Romania route-finding problem. *[CO3 | 16M]*

(a) A* algorithm with f(n)=g(n)+h(n), priority queue, graph search version with explored set. (4M)

(b) Optimality proof: with admissible h, A* never expands a suboptimal goal before the optimal. (6M)

(c) Trace A* on Romania map from Arad to Bucharest using SLD heuristic — show all expansions. (6M)

Answer: ____

Q3. Explain heuristic function design. Derive and compare h1 (misplaced tiles) and h2 (Manhattan distance) for 8-puzzle. Explain dominance. *[CO3 | 16M]*

(a) Design of heuristics from relaxed problems. Admissibility proof for h1 and h2. (6M)

(b) Dominance definition and proof that h2 dominates h1 for 8-puzzle. (4M)

(c) Effective branching factor concept. Comparison of search performance for h1 vs h2 vs exact. (6M)

Answer: ____

Q4. Explain greedy best-first search with example. Discuss inadmissible heuristics and weighted A*. Explain search contours. *[CO3 | 16M]*

(a) Greedy best-first: algorithm, example trace, why not optimal. (4M)

(b) Weighted A*: f(n)=g(n)+W*h(n), epsilon-suboptimal guarantee, tradeoff analysis. (6M)

(c) Search contours: how A* expands in circles around start; inadmissible heuristics and their effects. (6M)

Answer: ____

Q5. Describe problem formulation for 8-puzzle, N-Queens, and Missionaries and Cannibals. Explain bidirectional search. *[CO3 | 16M]*

(a) 8-puzzle: state space, initial/goal state, actions, path cost, BFS/A* comparison. (4M)

(b) N-Queens: incremental formulation, state space size, constraint propagation in backtracking. (6M)

(c) Missionaries and Cannibals: state, constraints, solution. Bidirectional BFS analysis. (6M)

Answer: ____

UNIT III
Local Search and Optimization

PART A 2-Mark Questions with Answers

Q1. What is local search and when is it used? *[CO3 | 2M]*

Answer:

Local search maintains only the current state (not path), moves to neighboring states based on objective value.

Used when the path to the solution is irrelevant — only the goal state matters (optimization).

Advantages: very low memory O(1); can find good solutions in large continuous/discrete spaces. Examples: N-Queens, scheduling, VLSI design.

Q2. What is hill-climbing? List its problems. *[CO3 | 2M]*

Answer:

Hill-climbing repeatedly moves to the highest-valued neighbor. Terminates when no neighbor is better.

Problems: Local maxima (no better neighbor but not global optimum), Plateaux (flat regions), Ridges (narrow peaks).

Solutions: random restarts, stochastic hill climbing, simulated annealing.

Q3. Explain simulated annealing. *[CO3 | 2M]*

Answer:

Inspired by metallurgical annealing. Allows 'bad' moves with probability P = e^(delta_E/T).

At high temperature T: explores widely (like random walk). At low T: like hill climbing.

Cooling schedule: T decreases over time. Guaranteed to reach global optimum if schedule is slow enough.

Q4. What is the minimax algorithm? *[CO3 | 2M]*

Answer:

Minimax computes optimal strategy for two-player zero-sum deterministic perfect-information games.

MAX tries to maximize value; MIN tries to minimize it. Recurse to terminal states, back up values.

Complete, optimal against optimal opponent. Time O(b^m), Space O(bm).

Q5. What is alpha-beta pruning? *[CO3 | 2M]*

Answer:

Alpha-beta eliminates branches in minimax that cannot affect the final decision.

Alpha: best value MAX guaranteed so far. Beta: best value MIN guaranteed so far. Prune when alpha >= beta.

With perfect move ordering: O(b^(m/2)) — doubles effective depth vs minimax O(b^m).

Q6. What are AND-OR search trees? *[CO3 | 2M]*

Answer:

AND-OR trees handle nondeterministic environments where action outcomes are uncertain.

OR nodes: agent's choice of action. AND nodes: all possible outcomes must be handled (environment's choice).

Solution is a contingency plan (conditional): if outcome X do Y, else do Z.

Q7. What is MCTS? List its four phases. *[CO3 | 2M]*

Answer:

Monte Carlo Tree Search incrementally builds a search tree using random simulations without an evaluation function.

Four phases: 1. Selection (UCB1 formula), 2. Expansion (add child node), 3. Simulation (random playout), 4. Backpropagation (update win/visit statistics).

Used in AlphaGo. Converges to minimax with sufficient simulations.

Q8. What is local beam search? *[CO3 | 2M]*

Answer:

Local beam search keeps k states simultaneously, generates all their successors, selects k best overall.

Unlike k independent restarts, information is shared between beams — promising regions get more focus.

Stochastic beam search: selects k successors probabilistically to maintain diversity.

Q9. What is the erratic vacuum world problem? *[CO3 | 2M]*

Answer:

Erratic vacuum: Suck action either deposits dirt on clean square or cleans dirty square — nondeterministic.

Agent cannot use a fixed action sequence; needs a contingency plan.

Solved using AND-OR search: plan branches on all possible action outcomes.

Q10. What is Expectiminimax? *[CO3 | 2M]*

Answer:

Expectiminimax extends minimax to stochastic games with chance nodes (dice, cards).

At chance nodes: expected value = sum over outcomes of P(outcome) * EXPECTIMINIMAX(resulting state).

Used in Backgammon (TD-Gammon). More complex than minimax due to chance outcomes.

PART B 16-Mark Questions (Write Answers Below)

Q1. Explain local search algorithms: hill-climbing variants, simulated annealing, and local beam search. Compare with systematic search. *[CO3 | 16M]*

(a) Hill-climbing: steepest ascent, stochastic, first-choice, random-restart — algorithms and properties. (6M)

(b) Simulated annealing: algorithm with probability formula, cooling schedule, convergence proof. (6M)

(c) Local beam vs stochastic beam vs k random restarts. Table comparing local and systematic search. (4M)

Answer: ____

Q2. Explain adversarial search. Describe the minimax algorithm with full pseudocode and trace it on a sample game tree. *[CO3 | 16M]*

(a) Two-player zero-sum games: components, utility function, game tree definition. (4M)

(b) Minimax algorithm: MINIMAX(s), MAX-VALUE, MIN-VALUE pseudocode. (6M)

(c) Full trace on a depth-3 game tree showing backed-up minimax values. (6M)

Answer: ____

Q3. Explain alpha-beta pruning in detail. Show mathematically how it improves minimax. Illustrate with a worked example showing pruned branches. *[CO3 | 16M]*

(a) Why minimax is infeasible for chess (b=35, m=100). Need for pruning. (2M)

(b) Alpha-beta algorithm with pseudocode; alpha and beta propagation rules. (6M)

(c) Full worked example: trace on same game tree, mark pruned nodes, prove O(b^(m/2)). (8M)

Answer: ____

Q4. Explain Monte Carlo Tree Search. Describe all four phases in detail. Explain how AlphaGo combined MCTS with deep neural networks. *[CO3 | 16M]*

(a) MCTS overview: why no evaluation function needed. Planning vs simulation. (2M)

(b) Selection (UCB1 formula), Expansion, Simulation (rollout), Backpropagation — detailed algorithms. (8M)

(c) AlphaGo architecture: policy network (move selection) + value network (position evaluation) + MCTS. (6M)

Answer: ____

Q5. Explain AND-OR search trees for nondeterministic environments. Describe the erratic vacuum problem and solve it using contingency planning. Also explain stochastic games. *[CO3 | 16M]*

(a) Nondeterministic environments: erratic vacuum cleaner problem formulation. (4M)

(b) AND-OR search tree algorithm with worked example — find contingency plan for vacuum. (6M)

(c) Stochastic games: Expectiminimax algorithm, chance nodes, Backgammon example. (6M)

Answer: ____

UNIT IV
Constraint Satisfaction Problems

PART A 2-Mark Questions with Answers

Q1. Define a Constraint Satisfaction Problem (CSP). *[CO3 | 2M]*

Answer:

CSP = (Variables X={X1...Xn}, Domains D={D1...Dn}, Constraints C).
A solution is a complete consistent assignment: every variable has a value from its domain, all constraints satisfied.
Examples: Map coloring (Australia), Sudoku, N-Queens, job scheduling.

Q2. What is arc consistency and the AC-3 algorithm? *[CO3 | 2M]*

Answer:

Arc (Xi,Xj) is consistent if for every value x in Di there exists y in Dj satisfying the constraint.
AC-3 enforces arc consistency by repeatedly removing inconsistent values from domains.
Time complexity: O(cd^3). Core technique in Sudoku solvers and CSP preprocessing.

Q3. What is backtracking search for CSPs? *[CO3 | 2M]*

Answer:

Backtracking = DFS + constraint checking. Assigns one variable at a time.
When current assignment violates a constraint, backtrack to previous variable.
More efficient than generate-and-test: prunes large portions of search space early.

Q4. What is the MRV heuristic? *[CO3 | 2M]*

Answer:

MRV (Minimum Remaining Values): choose the variable with fewest legal values in its domain.
Also called fail-first: likely to fail early, pruning large subtrees quickly.
Degree heuristic breaks MRV ties: choose variable with most constraints on remaining variables.

Q5. Compare forward checking and MAC. *[CO3 | 2M]*

Answer:

Forward checking: after assigning Xi, remove values from neighbors' domains inconsistent with Xi's value.
MAC (Maintaining Arc Consistency): runs full AC-3 after each assignment — stronger than forward checking.
MAC detects more failures earlier but is more expensive per step.

Q6. What is propositional logic? Give its connectives. *[CO4 | 2M]*

Answer:

Propositional logic uses atomic propositions (true/false) combined by connectives.
Connectives: NOT (negation), AND (conjunction), OR (disjunction), -> (implication), <-> (biconditional).
Entailment: KB |= alpha means alpha is true in every model where KB is true.

Q7. What is resolution in propositional logic? *[CO4 | 2M]*

Answer:
Resolution: (P OR Q) AND (NOT P OR R) yields (Q OR R) — complete inference rule for propositional logic.
Resolution refutation: negate goal, add to KB, convert to CNF, resolve until empty clause (contradiction).
DPLL uses resolution with unit propagation for SAT solving.

Q8. What is First-Order Logic? How does it differ from propositional logic? *[CO4 | 2M]*

Answer:
FOL adds objects (constants), properties/relations (predicates), functions, and quantifiers (forAll, exists).
Propositional: true/false facts only. FOL: can express 'All kings are persons' as forAll x: King(x)->Person(x).
FOL is more expressive but undecidable in general (vs propositional logic which is decidable).

Q9. What is unification? Give an example. *[CO4 | 2M]*

Answer:
Unification finds substitution theta that makes two logical expressions identical.
Example: UNIFY(Knows(John,x), Knows(John,Jane)) = {x/Jane}.
UNIFY(Knows(John,x), Knows(y,z)) = {y/John, x/z} or {y/John, z/x}.

Q10. Compare forward and backward chaining in FOL. *[CO4 | 2M]*

Answer:
Forward chaining: data-driven, start from facts, apply rules, derive new facts — complete for Horn clauses.
Backward chaining: goal-driven, start from query, find rules to support it, recurse on subgoals — used in Prolog.
Forward: better when goal unknown; Backward: better with specific query and many possible facts.

PART B 16-Mark Questions (Write Answers Below)

Q1. Define CSPs and explain constraint propagation. Describe the AC-3 algorithm with pseudocode. Apply it to the map-coloring problem. *[CO3 | 16M]*

(a) CSP definition: variables, domains, binary/unary/global constraints. Solution definition. (4M)
(b) Node consistency, arc consistency definitions. AC-3 algorithm with pseudocode. (6M)
(c) Apply AC-3 to Australia map-coloring — trace domain reductions step by step. (6M)

Answer: ____

Q2. Explain backtracking search for CSPs. Describe all improvement heuristics: MRV, degree, LCV, forward checking, and MAC. *[CO3 | 16M]*

(a) Basic backtracking algorithm — pseudocode and example with 4-Queens. (4M)
(b) Variable ordering: MRV and degree heuristic. Value ordering: LCV — with examples. (6M)
(c) Inference: forward checking vs MAC — algorithms, comparison, when each is better. (6M)

Answer: ____

Q3. Explain problem structure in CSPs. Describe tree-structured CSPs and cutset conditioning. Explain min-conflicts local search. *[CO3 | 16M]*

(a) Decomposing into independent subproblems using constraint graph. (4M)
(b) Tree-structured CSPs: algorithm, topological sort, O(nd^2) solution — proof. (6M)
(c) Cutset conditioning. Min-conflicts heuristic — algorithm, example with N-Queens, why fast. (6M)

Answer: ____

Q4. Explain propositional logic: syntax, semantics, inference. Describe DPLL algorithm for SAT. *[CO4 | 16M]*

(a) Syntax: atoms, connectives, complex sentences. Semantics: models, truth tables for all connectives. (4M)
(b) Entailment, validity, satisfiability. Inference rules: Modus Ponens, Resolution with CNF. (6M)
(c) DPLL: unit propagation, pure symbol elimination, backtracking. WalkSAT local search for SAT. (6M)

Answer: ____

Q5. Explain First-Order Logic: syntax, semantics. Describe forward chaining, backward chaining, and resolution for inference in FOL. *[CO4 | 16M]*

(a) FOL syntax: constants, predicates, functions, quantifiers. Semantics: models, interpretations, satisfaction. (4M)
(b) Forward chaining algorithm for Horn clauses — trace example 'Who does John know?'. (6M)
(c) Backward chaining, unification, FOL resolution — CNF conversion, resolution example. (6M)

Answer: ____

UNIT V
Knowledge Representation and Reasoning

PART A 2-Mark Questions with Answers

Q1. What is ontological engineering? [CO5 | 2M]

Answer:

Ontological engineering creates formal, shared vocabulary describing a domain — defining categories, objects, events, and relationships.

Upper ontology: general categories (objects, events, time, mental states). Domain ontology: specific to medicine, law, geography etc.

Tools: OWL (Web Ontology Language), SUMO, OpenCyc. Foundation for semantic web and knowledge graphs.

Q2. What are semantic networks? [CO5 | 2M]

Answer:

Semantic networks use graph structure: nodes represent objects/concepts; edges represent relationships (is-a, has-a, part-of, instance-of).

Support automatic inheritance of properties along is-a links.

Limitation: cannot easily express negation, quantification, or complex constraints. Less expressive than FOL.

Q3. What is the difference between CWA and OWA? [CO5 | 2M]

Answer:

CWA (Closed World Assumption): what is not known to be true is assumed false. Used in databases, Prolog.

OWA (Open World Assumption): unknown facts are simply unknown — not true nor false. Used in OWL, knowledge bases.

Example: DB lists no flights to Mars. CWA: no flights exist. OWA: we simply don't know.

Q4. What is classical planning? Define STRIPS. [CO6 | 2M]

Answer:

Classical planning finds a sequence of actions to achieve a goal under deterministic, fully observable conditions.

STRIPS representation: Initial state (ground atoms), Goal (atoms to achieve), Actions (Preconditions + Add effects + Delete effects).

Example: PickUp(A): Pre: Clear(A), HandEmpty; Add: Holding(A); Del: Clear(A), HandEmpty.

Q5. What is GraphPlan? [CO6 | 2M]

Answer:

GraphPlan constructs a planning graph alternating between fact layers and action layers until goal appears.

Planning graph compactly represents all possible plans. Plan extracted via backward search through layers.

More efficient than naive state-space search; useful for finding minimum-length plans.

Q6. State Bayes' Rule and explain it. *[CO5 | 2M]*

Answer:

Bayes' Rule: P(H|E) = P(E|H) * P(H) / P(E)

Where H = hypothesis (cause), E = evidence (effect). Reverses the direction of conditioning.

Use: compute P(disease|symptom) from P(symptom|disease) (from medical studies) and P(disease) (prevalence).

Q7. What is Naive Bayes and why is it 'naive'? *[CO5 | 2M]*

Answer:

Naive Bayes classifies by: P(C|x1,...,xn) proportional to P(C) * product_i P(xi|C).

It is 'naive' because it assumes all features are conditionally independent given the class — usually violated in practice.

Despite the naive assumption, it works well for text classification, spam filtering, and medical diagnosis.

Q8. Define prior and conditional probability. *[CO5 | 2M]*

Answer:

Prior probability P(A): probability of A without any additional evidence — based on background knowledge only.

Conditional probability P(A|B) = P(A AND B) / P(B): probability of A given B is observed.

Example: P(rain) = 0.3 (prior). P(rain|cloudy) = 0.7 (conditional on cloudiness).

Q9. What is default reasoning? *[CO5 | 2M]*

Answer:

Default reasoning handles typical cases while allowing exceptions without exhaustive enumeration.

Example: 'Birds fly by default' — but Tweety the penguin is an exception.

Mechanisms: Default Logic (Reiter), Circumscription, Non-monotonic logic. Traditional FOL is monotonic and cannot retract conclusions.

Q10. What is automated planning and how does it differ from search? *[CO6 | 2M]*

Answer:

Automated planning explicitly represents actions with preconditions and effects (STRIPS/PDDL).

Search: states are opaque; transitions applied blindly. Planning: states have internal structure — enables planning-specific heuristics.

Planning can use regression, graphplan, SATPLAN — all exploit the structured action representation.

PART B 16-Mark Questions (Write Answers Below)

Q1. Explain knowledge representation using ontological engineering, semantic networks, and frame systems. Discuss reasoning with default information. *[CO5 | 16M]*

(a) Ontological engineering: upper ontology, categories, events, mental objects, modal logic. (4M)

(b) Semantic networks and frame systems: inheritance, slots, fillers, examples. (6M)

(c) Default reasoning: default logic, circumscription, CWA vs OWA, non-monotonic inference. (6M)

Answer: ____

Q2. Explain automated classical planning. Describe STRIPS representation and planning algorithms: forward search, backward search, and GraphPlan. *[CO6 | 16M]*

(a) Classical planning definition, STRIPS representation with Block World example. (4M)

(b) Forward state-space search and backward regression search — algorithms and comparison. (6M)

(c) GraphPlan: planning graph construction (fact/action layers), mutex relations, plan extraction. (6M)

Answer: ____

Q3. Explain the fundamentals of probability theory for AI. Discuss joint distributions, marginalization, conditional independence, and Bayes' Rule with applications. *[CO5 | 16M]*

(a) Probability axioms, prior and conditional probability, product rule, chain rule. (4M)

(b) Full joint distribution: computing queries via marginalization and conditioning — worked example. (6M)

(c) Bayes' Rule: derivation, medical diagnosis application, spam filtering application. (6M)

Answer: ____

Q4. Explain the Naive Bayes model in detail. Describe training, prediction, and application to spam filtering. Discuss assumptions and limitations. *[CO5 | 16M]*

(a) Naive Bayes probabilistic model: generative model, conditional independence assumption. (4M)

(b) Training: MLE estimation of P(C) and P(xi|C) by counting. Laplace smoothing for zero counts. (6M)

(c) Spam filtering: bag-of-words features, classification example, precision/recall tradeoff. Limitations of independence assumption. (6M)

Answer: ____

Q5. Explain heuristics for planning: ignore delete lists, landmark heuristics. Discuss planning under uncertainty and how Bayesian methods support decision-making. *[CO5 | 16M]*

(a) Planning heuristics: ignore delete lists (relaxed problem), subgoal independence, pattern databases. (6M)

(b) Why classical planning fails in uncertain real-world environments. (2M)

(c) Bayesian decision-making: expected utility, acting under uncertainty, comparison with classical planning. (8M)

Answer: ____