Q1. Define an intelligent agent. Explain the PEAS framework with an example. (5 marks)

Answer:

- Intelligent agent:
  - Perceives its environment via sensors and acts via actuators.
  - Selects actions to maximize a performance measure over time.
  - Operates autonomously and adapts to changes.
- PEAS framework:
  - P (Performance measure): Criteria for success (e.g., amount of dirt cleaned, energy used, collisions avoided).
  - E (Environment): The external context (e.g., a grid of rooms, each either clean or dirty, with possible walls).
  - A (Actuators): How the agent affects the world (e.g., wheels to move north/south/east/west; vacuum motor to suck dirt).
  - S (Sensors): How the agent perceives the world (e.g., dirt detector, bump sensor, position locator).
- Example (Vacuum-Cleaner Agent):
  - P: Maximize cleaned tiles, minimize energy use and collisions.
  - E: Rooms arranged in a grid (tiles that are clean or dirty) with walls.
  - A: Move(N / S / E / W), Suck (vacuum).
  - S: DirtDetector (detects if current tile is dirty), BumpSensor (detects collisions), PositionLocator (knows location).

Q2(a). Compare BFS and DFS on memory usage, completeness, optimality, and time complexity. (5 marks)

Answer:

- Memory usage:
  - BFS stores all nodes at current depth -> O(b^d) in worst case (b = branching factor, d = solution depth).
  - DFS stores only one path plus unexplored siblings -> O(b*d) memory.
- Completeness:

- BFS is complete if the branching factor is finite (it will eventually reach every level).
- DFS is not guaranteed complete in infinite-depth graphs or when cycles exist (unless modified with cycle checks).

- Optimality:
  - BFS is optimal if all step costs are equal (it finds the shallowest goal).
  - DFS is not optimal; it may find a longer path before the shortest.
- Time complexity:
  - BFS: O(b^d).
  - DFS: O(b^m), where m is maximum depth; in the worst case, m >> d, so potentially worse than BFS.

Q2(b). Apply Greedy Best-First Search with given heuristics. Show the selected path and discuss optimality. (5 marks)

Answer:

- Graph edges with costs:
  - A -> A (1), A -> C (4), B -> D (2), C -> D (1), D -> Goal (3).
- Heuristic values (h):
  - A: 7 B: 4 C: 2 D: 1 Goal: 0
- Greedy-BFS trace (start at A):
  1. Start = A (h=7). Neighbors: A (h=7), C (h=2).
  2. Pick the node with lowest h -> C. Path so far: A -> C.
  3. At C (h=2). Neighbors: D (h=1). Choose D. Path: A -> C -> D.
  4. At D (h=1). Neighbor: Goal (h=0). Choose Goal. Final path: A -> C -> D -> Goal.
- Optimality check:
  - Path cost = A->C (4) + C->D (1) + D->Goal (3) = 8.
  - No alternative path from A to Goal; B is not connected to A, so the greedy route is also optimal in this case.

Q3. Role of First-Order Predicate Logic (FOPL) in knowledge representation: syntax, semantics, and quantifiers. (5 marks)

Answer:

- Syntax & Semantics:
  - Constants: Represent specific objects (e.g., Alice, Milk).
  - Variables: Placeholders (e.g., x, y).
  - Predicates: Relations on objects (e.g., Likes(Alice, x)).
  - Functions: Map objects to objects (e.g., FatherOf(x)).
  - Connectives: AND, OR, NOT, -> (implies).
  - Quantifiers: for all (for all x), there exists (there exists x).
  - An interpretation assigns each constant and function a domain element, and each predicate a truth value.
  - A sentence is true if it evaluates to true under the interpretation (e.g., for all x (Cat(x) -> Mammal(x)) means 'All cats are mammals').
- Universal vs. Existential Quantifiers:
  - for all x P(x): P(x) holds for every x in the domain (e.g., all patients with high fever have infection).
  - there exists x P(x): There exists at least one x for which P(x) is true (e.g., there exists a patient with tuberculosis).
- Effectiveness:
  - Universal quantification captures general rules (e.g., for all x (Fever(x) AND Cough(x) -> FluLikely(x))).
  - Existential quantification captures existence claims (e.g., there exists x (HighRisk(x) AND Age(x) > 65)).
  - Strengths: Precise, supports inference.
  - Limitations: Can be verbose for large domains and handling uncertainty requires extensions.

Q4. Bayesian probability - medical test problem. Compute P(disease | positive). (5 marks)

Answer:

- Given:
  - P(D) = 0.01, P(not D) = 0.99.
  - P(+ | D) = 0.99, P(- | not D) = 0.95 -> P(+ | not D) = 0.05.
- Bayes' theorem:

  P(D | +) = [P(+ | D)P(D)] / [P(+ | D)P(D) + P(+ | not D)P(not D)]

  = (0.99 * 0.01) / (0.99 * 0.01 + 0.05 * 0.99)

  = 0.0099 / (0.0099 + 0.0495)

= 0.0099 / 0.0594 ~ 0.1667 (16.67%).

Q5. Use of certainty factors (CF) in expert systems. Role in decision making under uncertainty. (5 marks)

Answer:

- Certainty Factors (CF):
  - Numeric values in [-1, +1] representing confidence in a rule or fact.
  - Positive CF indicates evidence for; negative CF indicates evidence against.
- How CFs work:
  - Each rule has a CF indicating confidence in its conclusion given premises.
  - When multiple rules fire, CFs are combined (e.g., MYCIN's combination formulas).
- Role in decision making:
  - Aggregation of evidence: accumulates partial evidence from various rules.
  - Thresholding: if final CF > threshold, system asserts conclusion.
  - Explanation: traces CF contributions for rationale.
- Example:
  - Rule1: IF fever AND cough THEN flu CF = 0.7
  - Rule2: IF sore throat AND headache THEN flu CF = 0.6
  - Combined CF ~ 0.88 for stronger belief if both rule conditions met.

Q6. Designing a medical expert system: knowledge acquisition method, rule-based inference, handling incomplete/uncertain data. (5 marks)

Answer:

- Knowledge acquisition:
  - Structured interviews with doctors to extract symptom-disease correlations.
  - Case study analysis of patient records to validate rules.
- Rule-based inference structure:
  IF Symptom(X) = True AND LabTest(Y) > Threshold THEN PossibleDisease = Z CF = c
  IF PossibleDisease = Z AND [additional conditions] THEN ConfirmDiagnosis = Z CF = c'.
- Handling incomplete or uncertain data:
  - Use default rules when data is missing (e.g., assume normal if no report).

- Attach CFs to rules and propagate through inference.
- If data missing, system requests additional tests or outputs 'unknown' and uses highest-CF path.

Q7(a). Single-layer perceptron output. (3 marks)

Answer:
- Inputs = [1, 0, 1], Weights = [0.4, 0.6, -0.5], Bias = -0.2.
- Net input: (1 * 0.4) + (0 * 0.6) + (1 * -0.5) + (-0.2) = 0.4 - 0.5 - 0.2 = -0.3.
- Activation (step): output = 1 if net >= 0, else 0 -> net < 0 => output = 0.

Q7(b). Backpropagation concept and role in training neural networks. (2 marks)

Answer:
- Backpropagation:
  - Computes gradient of loss with respect to each weight by propagating error backwards through layers.
  - Uses chain rule to adjust weights to minimize loss (e.g., MSE or cross-entropy).
- Role:
  - Enables multi-layer networks to learn internal representations by updating weights layer by layer.
  - Automates gradient-descent-based training of deep networks.

Q8. Fuzzy set A = {(1, 0.2), (2, 0.5), (3, 1), (4, 0.7), (5, 0.3)}. Find support, core, alpha-cut for alpha = 0.4. (5 marks)

Answer:
- Support: all elements with mu > 0 -> {1, 2, 3, 4, 5}.
- Core: elements with mu = 1 -> {3}.
- 0.4-cut: elements with mu >= 0.4 -> {2 (0.5), 3 (1), 4 (0.7)}.

Q9. Declarative vs Procedural knowledge: definitions, examples, and expert-system tasks. (5 marks)

Answer:
- Declarative knowledge:

  - Definition: Facts and relationships (the "what").
  - Example: "Patient X has fever," "Normal blood pressure = 120/80."
  - Expert-system task: Storing domain facts in the knowledge base.
- Procedural knowledge:
  - Definition: Sequences of steps or algorithms (the "how").
  - Example: Algorithm to compute a risk score based on age and symptoms.
  - Expert-system task: Inference control or action selection (e.g., conflict resolution, rule-firing strategy).
- Justification:
  - Declarative: needed for representing patient data and domain rules.
  - Procedural: needed for controlling reasoning and generating recommendations.