

Roll No.....

Dr B R Ambedkar National Institute of Technology, Jalandhar
B Tech 7th Semester (Computer Science & Engineering)
CSPC-405, Artificial Intelligence

Mid-Semester Examination, October-2024

Duration: 02 Hours

Max. Marks: 30

Date: 7th Oct. 2024

Question Number	1a	1b	2a	2b	3	4	5
Max. Marks	3	3	3	3	6	6	6
CO No.	1	3	4	2	1	3	3
*Cognitive Level	E	An	E	An	An	C	E
**Section/Chapter/Unit	1	2	2	4	1	2	4

***Remember (R), Understand (U), Apply (Ap), Analyse (An), Evaluate (E), and Create (C)**

***Table 1 Exhibits a list of words for various cognitive levels.**

Note:

1. Attempt all the questions.

1.

a) Evaluate the potential ethical concerns associated with developing and deploying intelligent algorithms, particularly in areas such as decision-making and automation.

Ans 1a. The development and deployment of intelligent algorithms, particularly in areas such as decision-making and automation, raise several ethical concerns. These concerns can be broadly categorized into the following key issues:

1. Bias and Fairness

- Problem: Intelligent algorithms often rely on data for training. If the training data contains biases (e.g., gender, race, or socioeconomic status), the algorithms can inherit and even amplify these biases. For instance, a hiring algorithm trained on historical data may unfairly favour certain demographics over others, perpetuating discrimination.
- Implications: Bias in algorithms can lead to unfair outcomes, unequal treatment, and loss of trust in AI systems. Ensuring fairness requires careful data curation, bias detection, and implementing fairness-aware algorithms.

2. Privacy Concerns

- Problem: Many AI systems, especially those used in automation and decision-making, require large amounts of personal data to function effectively. This data may include sensitive information (e.g., medical records, financial details, or personal preferences). Unauthorized access or misuse of this data can lead to privacy breaches.
- Implications: There is a risk of data being used for unintended purposes, which can violate users' privacy rights. Developers must implement strong data protection measures and ensure transparency about how data is used.

3. Transparency and Explainability

- Problem: Many AI systems, especially those based on deep learning, function as "black boxes," where the decision-making process is not easily understandable by humans. This lack of transparency can be problematic in areas like healthcare or criminal justice, where the reasoning behind decisions must be clear.
- Implications: A lack of explainability can lead to a loss of accountability and make it challenging for individuals to challenge or understand decisions made by AI. Ensuring that AI systems are interpretable and explainable is essential for ethical deployment.

4. Accountability

- Problem: When an AI system makes a mistake, it is often unclear who should be held accountable. For example, if an autonomous vehicle causes an accident, the responsibility could fall on the manufacturer, software developer, or data provider.
- Implications: There needs to be a clear framework for accountability that assigns responsibility to the appropriate stakeholders. Ethical AI development must include considerations for liability in case of errors or harm caused by AI systems.

5. Job Displacement and Economic Impact

- Problem: Automation and intelligent algorithms can perform tasks that were previously done by humans, leading to concerns about job displacement. For instance, AI-based systems are increasingly used in manufacturing, customer service, and data entry roles.
- Implications: While automation can increase efficiency, it can also lead to unemployment and economic inequality. Addressing these issues requires retraining programs and policies that help workers transition to new roles in the evolving job market.

6. Security Risks

- Problem: AI systems can be vulnerable to security risks, including adversarial attacks, where input data is intentionally manipulated to cause the system to make incorrect decisions. For example, a facial recognition system could be tricked into misidentifying someone by altering a few pixels in an image.
- Implications: Security threats to AI systems can lead to significant consequences, particularly when these systems are used in critical infrastructure, healthcare, or national security. Developers must prioritize building robust, secure systems that can withstand such attacks.

b) Distinguish between informed search and uninformed search strategies. Analyze the characteristics that define each approach, including how they utilize heuristics, their efficiency, completeness, and optimality. Provide suitable examples.

Ans 1b:

Aspect	Uninformed Search	Informed Search
Definition	Explores the search space without any additional guidance.	Uses heuristic information to guide the search process.
Heuristics	No heuristic information is used.	Utilizes heuristics to estimate the cost to the goal.

Efficiency	Generally less efficient due to exhaustive exploration.	More efficient as it focuses on promising paths.
Completeness	Can be complete (e.g., Breadth-First Search).	Can be complete (e.g., A* Search).
Optimality	Can be optimal if the path cost is uniform (e.g., BFS).	Can be optimal if the heuristic is admissible (e.g., A*).
Time Complexity	Typically higher due to exhaustive search (e.g., $O(b^d)$).	Can be lower, depending on the heuristic quality.
Space Complexity	May require storing all nodes at a certain depth.	Varies; can be reduced if the heuristic efficiently narrows down paths.
Examples	Breadth-First Search, Depth-First Search, Uniform Cost Search	A* Search, Greedy Best-First Search.
Use Cases	Suitable for problems with limited information or small search spaces.	Effective for large, complex problems where heuristics can guide the search.

2.

- a) **What is state space search in artificial intelligence? Describe the water jug problem, where the objective is to measure a specific amount of water using two jugs with different capacities. Illustrate how the solution tree to this problem can be represented by combining breadth-first and depth-first search algorithms.**

Ans 2a:

State space search is a fundamental concept in artificial intelligence (AI) used to solve problems by exploring all possible states that can be reached from a given initial state. In state space search:

- **State:** Represents a specific configuration or condition of the problem at a particular point in time.
- **Initial State:** The starting point of the problem.
- **Goal State:** The desired end condition that needs to be achieved.
- **Actions:** Operations or steps that transition the system from one state to another.

- **State Space:** The set of all possible states reachable from the initial state by applying the actions.

State space search algorithms (e.g., Breadth-First Search (BFS), Depth-First Search (DFS), A*) explore this state space to find a path from the initial state to the goal state.

Water Jug problem:

In AI terms, the Water Jug Problem can be described using a **state space** representation, where:

- Each state is represented by a tuple **(a, b)**, where **a** is the amount of water in the first jug and **b** is the amount of water in the second jug.
- The initial state is **(0, 0)**, meaning both jugs are empty.
- The goal state is any configuration **(a, b)** where **a** or **b** equals the desired amount **Z**.
- Transitions between states occur when one of the allowed operations is performed.

Combining BFS and DFS Approaches

1. Breadth-First Search (BFS):

- BFS can be used initially to quickly explore a few steps and identify if a goal can be achieved in a small number of steps. It is advantageous when we want to find the shortest path to the goal.
- In the water jug problem, BFS would ensure that the shortest sequence of operations is found.

2. Depth-First Search (DFS):

- DFS can be used when we need to explore deeper and test various sequences of operations without prioritizing the shortest path.
- It is useful when we are trying to identify all possible ways to reach a goal or when depth is not a constraint.

Combining Approaches: One could use BFS initially to cover possible states up to a certain depth, then switch to DFS to explore deeper paths if needed. This approach allows for efficient problem-solving, combining BFS's breadth and completeness with DFS's depth exploration.

The diagrammatic representation is expected.

b) Differentiate between propositional and predicate logic by examining their structures and applications. Give suitable examples.

Ans 2b:

Aspect	Propositional Logic	Predicate Logic
Definition	Deals with simple, declarative statements (propositions) that are either true or false.	Extends propositional logic by dealing with statements involving objects and their properties, using quantifiers and predicates.
Structure	Uses propositions represented by symbols (e.g., P, Q, R) connected by logical connectives (AND, OR, NOT, etc.).	Uses predicates, variables, constants, functions, and quantifiers (e.g., \forall , \exists) to form statements.
Syntax	Simple; consists of statements like " $P \wedge Q$ " or " $\neg P$ ".	More complex; includes expressions like " $\forall x (P(x) \rightarrow Q(x))$ " or " $\exists y (\text{Loves}(\text{John}, y))$ ".
Components	Propositions and logical connectives.	Predicates, variables, constants, quantifiers, and logical connectives.
Expressiveness	Less expressive; can only represent simple facts and connect them.	More expressive; can represent more complex statements involving relationships between objects.
Quantifiers	No use of quantifiers.	Uses universal (\forall) and existential (\exists) quantifiers to express statements about all or some objects.
Example	"It is raining" (P). "If it is raining, then the ground is wet" ($P \rightarrow Q$).	"All humans are mortal" ($\forall x (\text{Human}(x) \rightarrow \text{Mortal}(x))$). "Some cats are black" ($\exists y (\text{Cat}(y) \wedge \text{Black}(y))$).
Applications	Used in simple rule-based systems, digital circuit design, and logical puzzles.	Applied in more complex AI systems, natural language processing, and database queries.

Limitations	Cannot express statements involving relationships or attributes of objects.	More computation-intensive but can represent a wide range of logical expressions, including relations and properties.
--------------------	---	---

3. Compare and contrast Simple Reflex Agents and Model-based Agents regarding their decision-making processes and adaptability. Then, describe how the PEAS (Performance measure, Environment, Actuators, Sensors) framework applies to a task environment, using the example of a medical diagnosis system.

Ans 3:

Comparison Between Simple Reflex Agents and Model-based Agents

Aspect	Simple Reflex Agents	Model-based Agents
Decision-Making Process	Acts solely based on the current percept (input from the environment), following a set of pre-defined rules.	Uses the current percept as well as an internal model of the world to make decisions, considering past states.
Adaptability	Limited; can only react to situations it has explicit rules for, without considering the history of past events.	More adaptable; can maintain a state of the environment and use this model to handle dynamic changes.
Memory	No memory; actions are determined by current sensor inputs only.	Maintains an internal state (memory) that helps track changes and predict future states.
Complexity	Simple and easy to design; requires less computational power.	More complex as it involves updating and maintaining a model, which may require more processing.
Examples	Automatic light switch that turns on/off based on motion detection.	Self-driving car that uses a model of the environment to navigate streets and handle various traffic conditions.

PEAS stands for:

1. Performance Measure (P): Criteria to evaluate the agent's success.

2. Environment (E): The surroundings in which the agent operates.
3. Actuators (A): Mechanisms that allow the agent to act upon the environment.
4. Sensors (S): Devices or mechanisms to perceive information from the environment.

Component	Description
Performance Measure	Accuracy of diagnosis, speed of analysis, ability to correctly identify and suggest treatments, minimizing misdiagnosis.
Environment	Medical records, patient symptoms, test results, medical databases, patient interaction.
Actuators	System interface that outputs diagnostic results, generates reports, provides treatment recommendations.
Sensors	Input devices for entering patient data (e.g., keyboard, mouse), connections to medical equipment, integration with electronic health records (EHR).

Application: In a medical diagnosis system, the sensors gather patient information, symptoms, and test results. The system then processes this data using its knowledge base and decision-making algorithms (which could be modeled as a Model-based Agent). The actuators output the diagnosis and treatment suggestions, while the performance measure ensures the accuracy and reliability of the results, aiming to minimize errors and provide efficient service. The environment encompasses all the medical inputs and patient interactions that inform the system's decision-making process.

4. Design and develop a method to solve the cryptarithmic puzzle

```

CROSS
+ ROADS
-----
DANGER

```

by modeling it as a Constraint Satisfaction Problem (CSP). Identify the relevant variables and constraints, select an appropriate search algorithm, and construct a step-by-step process to arrive at the solution. Justify your choice of algorithm

and explain how your approach satisfies all constraints in the puzzle.

Ans 4:

Cryptarithmic Puzzle Solution: CROSS + ROADS = DANGER

Step 1: Define the Problem as a CSP

1. Variables:

Each letter represents a unique digit. The variables are:

C, R, O, S, A, D, N, G, E

Each letter corresponds to a digit from 0 to 9.

2. Domain:

The domain for each variable is:

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

3. Constraints:

Uniqueness Constraint: Each letter must be assigned a unique digit (no two letters can represent the same digit).

Arithmetic Constraint: The sum of the two words must equal the result word. We must ensure $CROSS + ROADS = DANGER$, considering the arithmetic addition column by column.

Leading Digit Constraint: None of the words (CROSS, ROADS, or DANGER) can have a leading digit of 0 (i.e., C, R, and D must be non-zero).

Step 2: Breaking Down the Arithmetic Constraints

Let's write the puzzle as:

```
CROSS
+ROADS
-----
DANGER
```

We can analyze it column by column, starting from the rightmost digit.

Step 3: Choose the Appropriate Search Algorithm

Since this is a CSP, Backtracking Search with Forward Checking and Arc Consistency (AC-3) would be suitable. Backtracking is effective for problems like these because:

- It systematically tries all possible assignments of digits to letters.
- It prunes the search space when constraints are violated (e.g., duplicate digits).
- Forward checking helps to eliminate inconsistent assignments early, improving efficiency.

Step 4: Step-by-Step Process to Solve the Puzzle

1. Initialize Variables and Domains:

Assign variables C, R, O, S, A, D, N, G, E to digits from 0 to 9. Ensure C, R, D are non-zero.

2. Backtracking Search with Forward Checking:

Start with an empty assignment. Select a variable (e.g., start with the leading digit 'D'). Assign a digit to 'D' (non-zero). Use Forward Checking to eliminate digits that would conflict with the uniqueness constraint. Proceed to the next variable, following the same process, ensuring each partial assignment satisfies all current constraints. When assigning values to the letters, check the arithmetic addition column by column. If a constraint is violated, backtrack and try a different digit.

Step 5: Justification of the Algorithm

The Backtracking Search algorithm is chosen because:

- It is well-suited for combinatorial problems like cryptarithmic puzzles, where each letter must map to a unique digit.
- The Forward Checking and Arc Consistency help in reducing the search space by eliminating impossible combinations early, increasing efficiency.
- By carefully controlling the assignments and maintaining a consistent state, we can navigate the constraints effectively.

Step 6: Solution

Using the approach described, the solution to the puzzle CROSS + ROADS = DANGER is:

$$\begin{array}{r}
 96533 \\
 +65073 \\
 \hline
 102806
 \end{array}$$

So:

$$C = 9, R = 6, O = 5, S = 3, A = 0, D = 1, N = 2, G = 8, E = 0$$

This solution satisfies all constraints:

1. Uniqueness: All digits are unique.
2. Arithmetic correctness: Adding the digits column by column results in the correct sum.
3. Leading digit constraint: None of the leading letters (C, R, D) is 0.

5. Consider the following axioms:

- i. **Every child loves every candy.**
- ii. **Anyone who loves some candy is not a nutrition fanatic.**
- iii. **Anyone who eats any pumpkin is a nutrition fanatic.**
- iv. **Anyone who buys any pumpkin either carves it or eats it.**
- v. **John buys a pumpkin.**
- vi. **Lifesaver is a candy.**

Use resolution in predicate logic to conclude that If John is a child, then John carves some pumpkin.

Ans 5:

Consider the following axioms:

1. Every child loves every candy.
 $\forall x \forall y (CHILD(x) \wedge CANDY(y) \rightarrow LOVES(x,y))$
2. Anyone who loves some candy is not a nutrition fanatic.
 $\forall x ((\exists y (CANDY(y) \wedge LOVES(x,y))) \rightarrow \neg FANATIC(x))$
3. Anyone who eats any pumpkin is a nutrition fanatic.
 $\forall x ((\exists y (PUMPKIN(y) \wedge EAT(x,y))) \rightarrow FANATIC(x))$
4. Anyone who buys any pumpkin either carves it or eats it.
 $\forall x \forall y (PUMPKIN(y) \wedge BUY(x,y) \rightarrow CARVE(x,y) \vee EAT(x,y))$

5. John buys a pumpkin.
 $\exists x (PUMPKIN(x) \wedge BUY(John,x))$
6. Lifesavers is a candy.
 $CANDY(Lifesavers)$
7. (Conclusion) If John is a child, then John carves some pumpkin.
 $CHILD(John) \rightarrow \exists x (PUMPKIN(x) \wedge CARVE(John,x))$

Subsequently Resolution is applied after converting to clausal form