AI PART-B

SET-1:

2) a. Classical planning

Definition

Classical Planning is an AI approach where an agent searches for a **sequence of actions** that transforms an **initial state** into a **goal state** in a well-defined environment. It assumes:

- The world is **fully observable** (the agent knows everything).
- Actions have **deterministic effects** (they always produce the expected result).
- The environment is **static** (it does not change unless the agent acts).

Key Features

- Uses **search algorithms** (e.g., Breadth-First Search, Depth-First Search, A*).
- Typically uses **STRIPS (Stanford Research Institute Problem Solver)** or **PDDL (Planning Domain Definition Language)**.
- The planner finds an optimal **sequence of actions** to achieve the goal.

Example of Classical Planning

Robot Navigation Problem

Consider a **robot** in a warehouse that needs to move a box from location A to location B.

- **Initial State:** The robot is at A, and the box is at A.
- **Actions:** The robot can "Pick Up Box," "Move to B," and "Drop Box."
- **Goal State:** The robot is at B, and the box is also at B.

Using **state-space search**, the planner finds the optimal sequence:

1. Pick Up Box
2. Move to B
3. Drop Box

This is a **sequential** approach where each action follows a predefined plan.

2) b. 4 key points about planning

•Uncertainty:

Unlike traditional planning where actions have predictable outcomes, in nondeterministic domains, actions can have multiple possible outcomes, depending on factors outside the agent's complete control.

•Partial Observability:

Often, an agent may not have complete information about the environment state, requiring sensing actions to gather more information before deciding on the next action.

•Conditional Planning (Contingency Planning):

A primary approach to handle nondeterminism, where the plan includes branches based on different possible outcomes, allowing the agent to adapt its actions depending on what **happens in the environment.**

•Replanning:

Since the environment may change unexpectedly, the agent may need to re-evaluate its plan based on new information received during execution.

4) a. Concept of probabilistic reasoning

Probabilistic reasoning is a mathematical framework used to model uncertainty and make informed decisions based on available evidence. It relies on probability theory to quantify uncertainty and update beliefs when new data is obtained.

Key Features:

- Uses **probability distributions** to represent uncertainty.
- **Combines prior knowledge** with new data (Bayesian inference).
- Supports **optimal decisions** under uncertainty.
- Handles **large-scale, complex problems** efficiently.

Importance in Decision-Making:

- **Managing Uncertainty:**
  - Useful in domains like medicine, finance, weather, etc.
  - Allows logical reasoning with incomplete info.
- **Decision Support:**
  - Computes **expected outcomes** for rational choices.

    - Used in **risk assessment** and **AI systems**.
- **Using Prior Knowledge:**
    - Example: Rare disease testing—probabilistic models give more accurate results.
- **Scalable to Complex Systems:**
    - Tools: **Bayesian Networks**, **Hidden Markov Models**.

Examples:

- Medical diagnosis
- Weather prediction
- Financial risk analysis
- Self-driving cars
- Robotics

4) b. Bayes' Rule

Bayes' Theorem is a mathematical formula that helps determine the conditional probability of an event based on prior knowledge and new evidence.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Where, • P(A) and P(B) are the probabilities of events A and B also P(B) is never equal to zero,
• P(A|B) is the probability of event A when event B happens,
• P(B|A) is the probability of event B when A happens.

Bayes Theorem can be derived for events and random variables separately using the definition of conditional probability and density.

From the definition of conditional probability, Bayes theorem can be derived for events as given below:

$P(A|B) = P(A \cap B)/ P(B)$, where $P(B) \neq 0$

$P(B|A) = P(B \cap A)/ P(A)$, where $P(A) \neq 0$

Here, the joint probability $P(A \cap B)$ of both events A and B being true such that,

$P(B \cap A) = P(A \cap B)$

$P(A \cap B) = P(A \mid B) P(B) = P(B \mid A) P(A)$

$P(A|B) = [P(B|A) P(A)]/ P(B)$, where $P(B) \neq 0$

Example:

Suppose a disease affects 1% of the population. A test gives:

- **True positive rate:** 99% (P(Positive|Disease) = 0.99)
- **False positive rate:** 5% (P(Positive|No Disease) = 0.05)

Let's find the probability a person has the disease if they test positive (P(Disease|Positive)).

$$P(D) = 0.01, \quad P(\neg D) = 0.99$$

$$P(Pos|D) = 0.99, \quad P(Pos|\neg D) = 0.05$$

$$P(Pos) = P(Pos|D)P(D) + P(Pos|\neg D)P(\neg D) = (0.99)(0.01) + (0.05)(0.99) = 0.0099 + 0.0495 = 0.0594$$

$$P(D|Pos) = \frac{P(Pos|D)P(D)}{P(Pos)} = \frac{0.99 \times 0.01}{0.0594} \approx 0.1667$$

So, even if someone tests positive, the actual chance they have the disease is only **16.67%**, showing how Bayes' Rule adjusts belief based on prior probabilities.

Significance of Bayes' Rule

- Updates beliefs based on **new evidence**.
- Helps in **decision-making under uncertainty**.
- Core concept in **machine learning, AI, medical diagnosis, spam filtering**, etc.

6) Knowledge Engineering Process in FOL

1. Problem Definition

- Define what the system should know and do.
- Example: Diagnosing diseases based on symptoms.

2. Knowledge Acquisition

- Collect information from experts, databases, or documents.
- Example: Rules for identifying diseases, treatments, etc.

3. Knowledge Representation (using FOL)

- Represent knowledge using:
  - **Terms**: Objects (e.g., John)
  - **Predicates**: Relationships (e.g., Cough(John))
  - **Variables**: Placeholders (e.g., x)
  - **Quantifiers**: ∀ (for all), ∃ (exists)
  - **Connectives**: AND (∧), OR (∨), NOT (¬), → (implies)
- Example Rule:
  Cough(x) ∧ Fever(x) → Flu(x)

4. Knowledge Structuring

- Organize knowledge logically (e.g., hierarchies or categories).
- Example: Group diseases under "viral" or "bacterial".

5. Inference and Reasoning

- Use FOL techniques to derive conclusions:
  - **Forward Chaining**: From facts → conclusions
  - **Backward Chaining**: From goal → supporting facts

    - **Resolution**: Refutation-based inference
- Example:
  Given Cough(John) and Fever(John) → infer Flu(John)

6. Validation and Verification

- **Validation**: Check knowledge accuracy with real-world data.
- **Verification**: Ensure the system reasons and responds correctly.

7. Maintenance and Updating

- Update knowledge as new info is discovered.
- Example: New medical research updates diagnosis rules.

SET-2:

3) a. Classical planning

Same answer from set 1

3) b. Forward and backward state-space search

- **Forward Search:**
  1. Begins at the **initial state** and expands toward the goal.
  2. Uses breadth-first, depth-first, or heuristic search.
- **Backward Search:**
  1. Starts from the **goal state** and works backward to reach the initial state.
  2. Efficient when fewer goal states are defined.
- **Comparison:**

Criteria	Forward Search	Backward Search
Direction	Initial → Goal	Goal → Initial
Efficiency	Inefficient for large states	More efficient in some cases
Heuristic Use	Can use heuristics	Requires goal-based heuristics

5) a. Knowledge represented in an uncertain domain

Uncertain Domain in AI

- Involves incomplete, noisy, ambiguous, or unpredictable data.
- Common in real-world applications like medicine, robotics, and finance.

Why Handle Uncertainty?

- Enables AI to make informed, adaptable, and reliable decisions.
- Essential for prediction, diagnosis, planning, and reasoning.

Probabilistic Methods for Uncertainty

1. Probabilistic Reasoning

- Uses probability theory to handle uncertainty in decision-making.

- Example: Predicting disease likelihood based on symptoms.

2. Bayesian Networks (BNs)

- Graphical models with nodes (variables) and edges (dependencies).
- Efficiently compute probabilities with new evidence.
- Example: Flu ↔ Cough ↔ Fever in a medical BN.

3. Hidden Markov Models (HMMs)

- Models time-based data with hidden states.
- Widely used in speech recognition, bioinformatics.
- Example: Hidden phonemes → Observed sound waves.

4. Markov Decision Processes (MDPs)

- Used for sequential decision-making in stochastic environments.
- Includes states, actions, transition probabilities, and rewards.
- Example: Robot path planning in uncertain terrain.

5. Fuzzy Logic

- Handles approximate reasoning using degrees of truth (0–1).
- Fuzzy sets and rules replace binary logic.
- Example: "Warm" temperature = 0.7 truth value.

Applications

- **Medical Diagnosis** – BNs for disease prediction.
- **Autonomous Vehicles** – MDPs & Fuzzy Logic for decision-making.
- **NLP & Speech** – HMMs for language modeling.
- **Robotics** – Probabilistic reasoning for navigation.
- **Finance** – Risk analysis and fraud detection.

5) b. Semantics of Bayesian Networks

A **Bayesian Network (BN)**:

- Is a **Directed Acyclic Graph (DAG)**.

- Represents variables as **nodes**.
- **Edges** show **direct dependencies** between variables.
- Each node has a **Conditional Probability Table (CPT)** showing how it depends on its parent(s).
- The **Joint Probability Distribution (JPD)** is factored using the structure of the network.

🔗 Factorization Rule

For variables A, B, C:
If the graph shows $A \to B \to C$, then:

$$P(A, B, C) = P(A) \cdot P(B|A) \cdot P(C|B)$$

Example:

🍫 Simple Example: Chocolate & Mood

Variables:

1. **A: Ate Chocolate (Yes/No)**
2. **B: Happy Mood (Yes/No)**

Structure:

```css
A → B
```

This means: **Happiness depends on whether you ate chocolate.**

🔢 Probabilities:

- $P(A = Yes) = 0.6, P(A = No) = 0.4$
- $P(B = Yes|A = Yes) = 0.9, P(B = No|A = Yes) = 0.1$
- $P(B = Yes|A = No) = 0.2, P(B = No|A = No) = 0.8$

✅ Let's Find Joint Probability:

What's the probability that:

- You **ate chocolate** and
- You are in a **happy mood**?

$$P(A = Yes, B = Yes) = P(A = Yes) \cdot P(B = Yes|A = Yes)$$

$$= 0.6 \cdot 0.9 = 0.54$$

✅ So the joint probability $P(A = Yes, B = Yes) = 0.54$

6) Backward chaining and forward chaining

3. a. Difference between backward chaining and forward chaining

Feature	Forward Chaining	Backward Chaining
Direction	Starts from facts and moves towards the goal	Starts from the goal and moves towards facts
Approach	Data-driven (bottom-up)	Goal-driven (top-down)
Inference Process	Applies rules to infer new facts	Searches for facts to support a given goal
Goal Knowledge	The goal is not defined at the start	The goal is defined at the start
UseCase	Exploring all possibilities	Solving a specific problem or goal

SET-3:

3) a. Classical planning (from set 1)

3) b. Forward and backward state-space search (from set 2)

4) a. Concept of probabilistic reasoning (from set 1)

4) b. Bayes' Rule (from set 1)

(or)

5) a. Knowledge represented in an uncertain domain (from set 2)

5) b. Semantics of Bayesian Networks (from set 2)

7) Backward chaining and forward chaining (from set 2)