|  |  |
| --- | --- |
| **EXPT NO:2** | **Propositional Logic and Knowledge-Based Agent** |
| **DATE: 23.09.2025** |

**PRE-LAB QUESTIONS (PROVIDE BRIEF ANSWERS TO THE FOLLOWING QUESTIONS)**

1. Describe the two main components of a knowledge-based agent and the function of each.

A knowledge-based agent has a **knowledge base (KB)** storing facts and rules, and an **inference engine** that applies logical reasoning on KB to derive new knowledge and guide decisions.

1. Convert the following statement into a single propositional logic sentence: "The system is armed if and only if the sensor is active or the system is in manual override mode."

Let A = system armed, S = sensor active, M = manual override.  
The statement is:  
**A ↔ (S ∨ M)**  
(System armed if and only if sensor active or manual override).

1. What is a truth table, and how can you represent the IMPLIES (→) operator using only OR (∨) and NOT (¬)?

A truth table lists all possible truth values for propositions and outcomes. The implication **P → Q** is equivalent to **¬P ∨ Q**, using only OR and NOT.

1. Briefly explain the difference between Forward Chaining and Backward Chaining as inference methods.

**Forward chaining** starts from known facts, applying rules until a goal is reached. **Backward chaining** begins with the query/goal, working backward through rules to see if facts can support it.

1. What is a key limitation of propositional logic when trying to represent complex environments with many objects (e.g., "All doors are closed")?  
   Propositional logic cannot easily represent relationships or general rules about multiple objects. It lacks variables and quantifiers, making it impractical for expressing universal statements like “All doors are closed.”

**IN-LAB EXERCISE:**

**BJECTIVE:**  
To create a Knowledge-Based Agent that decides the actions of an autonomous delivery drone based on its sensor data and a set of operational rules, using Propositional Logic.

**PROCEDURE:**

**1. Scenario:**

* **Problem Statement:** An autonomous delivery drone needs to decide whether to Proceed, Return\_To\_Base, or Land\_Immediately based on its current status.
* **Sensor Data (Percepts):**
  + Battery level is low.
  + GPS signal is strong.
  + Weather is clear.
  + Package has been delivered.
* **Rules (Knowledge Base):**
  + If the battery is low, the drone must Return to Base.
  + If the package has been delivered, the drone must Return to Base.
  + If the weather is NOT clear, the drone must Return to Base.
  + If the battery is NOT low AND the GPS signal is weak (NOT strong), the drone must Land Immediately.
  + If the battery is NOT low AND the GPS signal is strong AND the weather is clear AND the package has NOT been delivered, the drone should Proceed.

**2. Implementation Steps:**

1. **Define propositional symbols:**

BL = Battery is Low

GS = GPS Signal is Strong

WC = Weather is Clear

PD = Package is Delivered

P = Action: Proceed

RTB = Action: Return to Base

LI = Action: Land Immediately

1. **Encode rules in propositional logic:**

Rule 1: BL → RTB

Rule 2: PD → RTB

Rule 3: ¬WC → RTB

Rule 4: (¬BL ∧ ¬GS) → LI

Rule 5: (¬BL ∧ GS ∧ WC ∧ ¬PD) → P

1. **Implement a Python function decide\_drone\_action(battery\_low, gps\_strong, weather\_clear, package\_delivered)** that takes the boolean state of the drone's sensors as input.
2. **Inside the function, use a prioritized if-elif-else structure** to determine the drone's action. The order of rules is important for safety (e.g., checking for low battery should be a top priority). The function should return a string representing the chosen action.
3. **Test the agent** with different sensor inputs to ensure it makes safe and logical decisions. For example:
   * Test Case 1: decide\_drone\_action(battery\_low=False, gps\_strong=True, weather\_clear=True, package\_delivered=False) → Should decide to Proceed.
   * Test Case 2: decide\_drone\_action(battery\_low=True, gps\_strong=True, weather\_clear=True, package\_delivered=False) → Should decide to Return to Base.

**CODE:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**OUTPUT:**

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

**POST-LAB QUESTIONS (PROVIDE BRIEF ANSWERS TO THE FOLLOWING QUESTIONS)**

1. In the implemented scenario, a new rule is added: "The agent can only perform its primary action (e.g., Proceed or Diagnose) if its communication link to the server is active." How would you add a new propositional symbol and modify an existing complex rule to accommodate this?

Introduce a new symbol **CL = Communication Link Active**. Modify existing rules by adding **∧ CL** to their conditions, e.g., (¬BL ∧ GS ∧ WC ∧ ¬PD ∧ CL) → Proceed.

1. The Python implementation likely used a series of if-elif-else statements. How does this structure implicitly handle conflicts where multiple rules might be true? What is a potential drawback of this approach?

The if-elif-else structure gives priority to the first matching condition, avoiding conflicts. A drawback is rigidity: rule order dictates behavior, possibly ignoring equally valid but lower-priority actions.

1. Propositional logic assumes facts are strictly True or False. How does this model fail to handle uncertainty (e.g., a 70% chance of rain or a faulty sensor that is sometimes wrong)?

Propositional logic cannot represent probabilities or uncertainty. It treats percepts as binary, so concepts like “70% chance of rain” or unreliable sensors are lost, limiting realistic reasoning in uncertain environments.

1. Our agent made a one-time decision based on a snapshot of its environment. How would the knowledge base and inference process need to be modified to reason about states over time (e.g., "If the battery was NOT low 5 minutes ago, but it IS low now, then land immediately")?

Temporal reasoning requires extending the knowledge base with **time-indexed facts** and temporal logic operators. The inference engine must track history and compare states across time to capture dynamic environmental changes.

1. Based on the agent you built, explain the difference between an agent that acts logically versus one that acts rationally. Could your logical agent make an irrational decision?

A logical agent follows rules consistently, while a rational agent chooses actions maximizing expected outcomes. A purely logical agent may act irrationally if rules ignore consequences or environmental uncertainties.

**ASSESSMENT**

|  |  |  |
| --- | --- | --- |
| **Description** | **Max Marks** | **Marks Awarded** |
| Pre Lab Exercise | **5** |  |
| In Lab Exercise | **10** |  |
| Post Lab Exercise | **5** |  |
| Viva | **10** |  |
| **Total** | **30** |  |
| **Faculty Signature** | |  |