Name : Simanta Chowdhury

Id: 22010118

**What is Knowledge Representation in AI?**

Knowledge representation in AI is about how to describe facts and rules about a problem so that a computer can use them to make decisions or draw conclusions. It’s like creating a set of instructions and facts that an AI can reason through.

**What Does Each File Do?**

**logic.py**

Solves the core library that defines the ‘Sentence’ class and its subclasses. The program represents knowledge as instances of sentence objects. The ‘model\_check’ function allows for exhaustively testing every possible truth assignment to verify logical entailment. ❑ How It works: ▪ The program creates logical sentences using the classes, combines them with logical rules. ▪ Uses model\_check to test if a query can be concluded from the knowledge base. ▪ This allows it to deduce facts, solve puzzles, or validate logical statements.

Fig : logic.py

**clue.py**

Solves a version of the Clue game where figure out the suspect, weapon, and room. The relationships and constraints between suspects, rooms, and weapons are represented as logical sentences within the And clauses of the KB. The program deduces which entities could or could not be involved in the crime based on given and derived knowledge.

How It Works:

* Uses symbols like ColMustard (suspect), kitchen (room), knife (weapon).
* Adds rules that only one suspect, one room, and one weapon can be the answer.
* Adds known facts (e.g., ColMustard isn’t the suspect).
* Checks each symbol to see if it can be deduced to be true, false, or uncertain.

Fig: clue.py code

Result:

**mastermind.py**

Solves a simplified version of the Mastermind game. Each potential position-color pairing is represented as a logical symbol. The relationships are encoded to ensure that each position has one color and no color repeats in different positions.

How It Works:

* Uses symbols to represent color positions (e.g., red0 means red is in position 0).
* Adds rules to ensure each color appears only once and only in one position.
* Adds known clues and restrictions (e.g., “blue is not in position 0”).
* Deduces the possible color combination.

Fig: mastermind.py code

Result:

**puzzle.py**

Solves a logic puzzle involving people and house assignments (e.g., who is in Gryffindor). The problem is represented as a set of logical constraints and rules. The relationships between people and house assignments are expressed using logical symbols and implications.

How It Works:

Uses symbols like GilderoyGryffindor to represent who might be in which house. ▪ Adds rules that each person is in only one house and no house has more than one person. ▪ Adds known facts and checks what can be deduced.

Fig: puzzle.py code

Result:

**How Does It Work?**

Symbols: Represent facts, like ColMustard for a suspect or red0 for red in position 0.

Logic Sentences: Combine symbols with logic rules (e.g., If A, then B, A and B, A or B).

Knowledge Base: A collection of these rules and known facts.

Model Checking: The program tries out different combinations to see if the facts and rules imply something (e.g., whether ColMustard could be the suspect).

Simplified Example :

Imagine:

* + - “If it rains, the grass is wet.”
    - “The grass is wet.” Find out, “Is it raining?”

The program would:

* + - Represent “It rains” and “Grass is wet” as symbols.
    - Use logic to check if, given that the grass is wet, it must be raining.

Key Points:

Knowledge Representation in these programs maps real-world problems into a logical framework. Model Checking automates the process of logical entailment, allowing the AI to reason about what must or may be true based on the known constraints. Propositional Logic is sufficient for these problems but could be extended with First-Order Logic for more complex scenarios involving variables and quantifiers. This framework demonstrates the foundational elements of AI knowledge representation by structuring and deducing logical facts to solve puzzles and problems programmatically.