PRACTICAL 9

Topic: Simple Inference Engine

**Components of the Inference Engine**

1. **Knowledge Base**:
   * Stores facts (e.g., It\_rains) and rules (e.g., It\_rains → Streets\_wet).
2. **Inference Rules**:
   * **Modus Ponens**: If PPP and P  ⟹  QP \implies QP⟹Q, infer QQQ.
   * **Modus Tollens**: If P  ⟹  QP \implies QP⟹Q and ¬Q\neg Q¬Q, infer ¬P\neg P¬P.
3. **Algorithm**:
   * Process facts and rules iteratively.
   * Infer new facts until no more can be inferred.

Code:

class SimpleInferenceEngine:

    def \_\_init\_\_(self):

        self.knowledge\_base = set()  # Known facts

        self.rules = []  # List of rules in the form (premise, conclusion)

    def add\_fact(self, fact):

        """Add a fact to the knowledge base."""

        self.knowledge\_base.add(fact)

    def add\_rule(self, premise, conclusion):

        """Add a rule in the form premise → conclusion."""

        self.rules.append((premise, conclusion))

    def infer(self):

        """Perform inference using Modus Ponens and Modus Tollens."""

        new\_inferences = set()

        for premise, conclusion in self.rules:

            # Modus Ponens: If premise is true, infer conclusion

            if premise in self.knowledge\_base and conclusion not in self.knowledge\_base:

                new\_inferences.add(conclusion)

            # Modus Tollens: If ¬conclusion is true, infer ¬premise

            if f"¬{conclusion}" in self.knowledge\_base and f"¬{premise}" not in self.knowledge\_base:

                new\_inferences.add(f"¬{premise}")

        # Add new inferences to the knowledge base

        self.knowledge\_base.update(new\_inferences)

        return new\_inferences

    def run\_inference(self):

        """Iteratively apply inference until no new facts can be derived."""

        while True:

            new\_inferences = self.infer()

            if not new\_inferences:

                break

# Example usage

engine = SimpleInferenceEngine()

# Adding rules

engine.add\_rule("It\_rains", "Streets\_wet")  # It\_rains → Streets\_wet

engine.add\_rule("Lights\_on", "Room\_bright") # Lights\_on → Room\_bright

# Adding facts

engine.add\_fact("It\_rains")         # It\_rains is true

engine.add\_fact("¬Streets\_wet")     # Streets\_wet is false (¬Streets\_wet)

# Run inference

engine.run\_inference()

# Print results

print("Knowledge Base:", engine.knowledge\_base)

Output:

