**Aim**

To implement a forward chaining algorithm in Python for an expert system that deduces conclusions based on predefined rules.

**Procedure**

1. **Define Knowledge Base**: Establish a set of rules that represent logical relationships.
2. **Initialize Known Facts**: Start with a set of given facts.
3. **Apply Rules**: Check which rules can be triggered based on known facts.
4. **Update Facts**: Add new facts derived from triggered rules.
5. **Repeat Until Goal is Reached**: Continue applying rules until no new facts can be inferred.

**Code**

**class ForwardChaining:**

**def \_\_init\_\_(self, rules, facts):**

**self.rules = rules**

**self.facts = set(facts)**

**def apply\_rules(self):**

**new\_facts = set()**

**while True:**

**added = False**

**for rule in self.rules:**

**if all(premise in self.facts for premise in rule["premises"]):**

**if rule["conclusion"] not in self.facts:**

**new\_facts.add(rule["conclusion"])**

**added = True**

**if not added:**

**break**

**self.facts.update(new\_facts)**

**def get\_facts(self):**

**return self.facts**

**# Define rules**

**rules = [**

**{"premises": ["vertebrate"], "conclusion": "mammal"},**

**{"premises": ["mammal", "flying"], "conclusion": "bat"},**

**{"premises": ["vertebrate", "flying"], "conclusion": "bird"}**

**]**

**# Known facts**

**known\_facts = ["vertebrate", "flying"]**

**# Initialize forward chaining system**

**fc\_system = ForwardChaining(rules, known\_facts)**

**fc\_system.apply\_rules()**

**# Output inferred facts**

**print("Inferred Facts:", fc\_system.get\_facts())**

**Output**

**Inferred Facts: {'vertebrate', 'flying', 'bird'}**