EXPERIMENT -8

Objective: IMPLEMENT FORWARD CHAINING ALGORITHM

Theory :

Definition:

- Forward Chaining is a data-driven reasoning technique used in expert systems and AI inference engines.
- It starts from known facts and applies inference rules to derive new facts until a goal is reached or no more rules can be applied.

Principle :

- Based on the Modus Ponens rule of inference:

  If “IF P THEN Q” is true, and P is known to be true, then Q can be inferred as true.

Basic Components :

1. Knowledge Base (KB):
   - Contains a set of *production rules*

     in the form: IF <conditions>

     THEN <conclusion>
2. Working Memory (WM):
   - Stores all currently known facts.
3. Inference Engine:
   - Applies the rules to the working memory to infer new facts.

Algorithm :

1.Initialize the working memory with given facts.

2. new facts Repeat until nocan be added

- Identify all rules whose conditions are satisfied by current facts.
- Choose one rule to apply (using a conflict resolution strategy).
- Fire the rule: Add the conclusion(s) of that rule to the working memory.
- Mark the rule as used to prevent repetition.

3. Stop when:

   The goal is achieved, or No new facts can be inferred.

Code :

```
rules = {
    "R1": ({"A", "B"}, "C"),
    "R2": ({"C"}, "D"),
    "R3": ({"D"}, "E")
}

facts = {"A", "B"}
goal = "E"

print("Initial Facts:", facts)

while True:
    applied = False
    for rule, (conditions, conclusion) in rules.items():
        if conditions.issubset(facts) and conclusion not in facts:
            facts.add(conclusion)
            applied = True
            print(f"Applied {rule}: Added {conclusion}")
            if conclusion == goal:
                print("Goal achieved:", goal
```

```
    applied = False
    break
    if not applied:
    break
print("\nFinal Facts:", facts)
```

Output