**Lab Assignment 7: Implementation of Forward Chaining Algorithm**

**1. Objective:**

The objective of this lab assignment is to implement the **Forward Chaining** algorithm, a common inference technique used in artificial intelligence and rule-based systems. This algorithm is used to derive conclusions from a set of known facts and rules.

**2. Problem Statement:**

Forward Chaining is a data-driven inference technique that starts with the known facts and applies inference rules to extract more data until a goal is reached. It is widely used in expert systems and knowledge-based systems for reasoning.

**3. Theory:**

**3.1. Forward Chaining Overview:**

In Forward Chaining, the algorithm uses a set of facts and rules to derive new facts. The process continues until no new facts can be generated or a specific goal is achieved.

* **Facts**: Statements that are known to be true (e.g., "It is raining").
* **Rules**: Conditional statements that define relationships between facts (e.g., "If it is raining, then the ground is wet").

**3.2. Algorithm Steps:**

1. **Initialize** the list of known facts.
2. **Repeat** the following until no new facts can be derived:
   * For each rule, check if the rule's conditions are satisfied by the known facts.
   * If a rule is satisfied, add the conclusion (new fact) to the list of known facts.
3. **Stop** when no new facts can be generated or a specific goal is reached.

**4. Algorithm Design:**

**4.1. Pseudocode:**

Here is the pseudocode for the Forward Chaining algorithm:

plaintext

Copy code

function forwardChaining(rules, facts):

new\_facts = facts.copy()

changed = true

while changed:

changed = false

for rule in rules:

if rule.condition is satisfied by new\_facts:

if rule.conclusion not in new\_facts:

new\_facts.append(rule.conclusion)

changed = true

return new\_facts

**4.2. Data Structures:**

* **Facts**: A list or set to store the known facts.
* **Rules**: A list of rules where each rule consists of a condition and a conclusion.

**5. Implementation:**

Below is a Python implementation of the Forward Chaining algorithm:

python

Copy code

class Rule:

def \_\_init\_\_(self, condition, conclusion):

self.condition = condition # A list of conditions

self.conclusion = conclusion # The conclusion of the rule

def forward\_chaining(rules, facts):

new\_facts = set(facts) # Use a set for faster lookups

changed = True

while changed:

changed = False

for rule in rules:

# Check if all conditions are satisfied

if all(cond in new\_facts for cond in rule.condition):

if rule.conclusion not in new\_facts:

new\_facts.add(rule.conclusion) # Add new fact

changed = True

return new\_facts

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

# Define facts

initial\_facts = {"It is raining", "It is cloudy"}

# Define rules

rules = [

Rule(["It is raining"], "The ground is wet"),

Rule(["It is cloudy"], "It may rain"),

Rule(["The ground is wet"], "The grass is slippery"),

Rule(["It may rain", "The ground is wet"], "There might be a flood"),

]

# Perform forward chaining

derived\_facts = forward\_chaining(rules, initial\_facts)

# Display results

print("Derived Facts:")

for fact in derived\_facts:

print(fact)

**6. Expected Output:**

When the above implementation is executed, it will display the derived facts based on the initial facts and rules. For the provided example, the expected output might be:

csharp

Copy code

Derived Facts:

It may rain

The ground is wet

The grass is slippery

There might be a flood

**7. Conclusion:**

The Forward Chaining algorithm effectively infers new facts from existing ones using a set of rules. This approach can be extended to more complex knowledge bases and rules, making it a powerful tool in AI systems and expert systems.

**8. References:**

* Russell, S. J., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
* Rich, E., & Knight, K. (1991). *Artificial Intelligence*. McGraw-Hill.