**Assignment 7**

**Problem Statement:**  
Implement the **Forward Chaining Algorithm**.

**Theory**

**1. Knowledge Representation & Inference**

* AI systems use **facts** and **rules** to represent knowledge.
* **Facts:** Statements that are known to be true.
* **Rules:** Conditional statements of the form
* Inference engines use reasoning to derive **new facts** from existing ones.

**2. Forward Chaining**

* **Forward Chaining** is a **data-driven reasoning** method.
* Start with known facts and apply rules to infer new facts.
* Process continues until no new facts can be derived.
* Used in **expert systems** such as medical diagnosis, fault detection, etc.

**Steps of Forward Chaining:**

1. Start with an initial set of facts.
2. Check each rule: if all conditions are satisfied, infer the conclusion.
3. Add the conclusion to the fact base if not already present.
4. Repeat until no new facts are added.

**Algorithm**

1. Initialize knowledge base with facts.
2. Repeat until no new facts are added:
   * For each rule, check if all conditions are satisfied.
   * If yes, add conclusion to fact base.
   * Print applied rules.
3. Stop when no more rules can be applied.

**Code (C++ Implementation)**

#include <iostream>

#include <vector>

#include <string>

#include <map>

using namespace std;

// Rule structure

struct Rule {

vector<string> conditions; // IF conditions

string conclusion; // THEN conclusion

};

// Check if all conditions of a rule are satisfied by known facts

bool conditionsSatisfied(const vector<string>& conditions, const map<string, bool>& facts) {

for (auto& cond : conditions) {

if (facts.find(cond) == facts.end() || facts.at(cond) == false)

return false;

}

return true;

}

// Forward chaining algorithm

void forwardChaining(vector<Rule>& rules, map<string, bool>& facts) {

bool newFactAdded = true;

while (newFactAdded) {

newFactAdded = false;

for (auto& rule : rules) {

if (conditionsSatisfied(rule.conditions, facts) && facts[rule.conclusion] == false) {

cout << "Applying rule: IF ";

for (size\_t i = 0; i < rule.conditions.size(); i++) {

cout << rule.conditions[i];

if (i != rule.conditions.size() - 1) cout << " AND ";

}

cout << " THEN " << rule.conclusion << endl;

facts[rule.conclusion] = true;

newFactAdded = true;

}

}

}

}

// Main function

int main() {

// Initial facts

map<string, bool> facts;

facts["Has\_Fever"] = true;

facts["Has\_Cough"] = true;

facts["Flu"] = false;

facts["Has\_Rash"] = true;

facts["Measles"] = false;

// Rules

vector<Rule> rules = {

{ {"Has\_Fever", "Has\_Cough"}, "Flu" },

{ {"Has\_Fever", "Has\_Rash"}, "Measles" }

};

cout << "Initial facts:\n";

for (auto& f : facts)

if (f.second) cout << "- " << f.first << endl;

cout << "\n";

// Run Forward Chaining

forwardChaining(rules, facts);

cout << "\nFinal inferred facts:\n";

for (auto& f : facts)

if (f.second) cout << "- " << f.first << endl;

return 0;

}

**Sample Output**

Initial facts:

- Has\_Cough

- Has\_Fever

- Has\_Rash

Applying rule: IF Has\_Fever AND Has\_Cough THEN Flu

Applying rule: IF Has\_Fever AND Has\_Rash THEN Measles

Final inferred facts:

- Flu

- Has\_Cough

- Has\_Fever

- Has\_Rash

- Measles

**Conclusion**

* The **Forward Chaining Algorithm** was successfully implemented.
* Starting from **known facts**, new facts like *Flu* and *Measles* were inferred using rules.
* This demonstrates **data-driven reasoning** in AI.
* Forward chaining is widely used in **expert systems**, e.g., **medical diagnosis** where symptoms lead to diseases.