Experiment 1

AIM:

To implement a DFA in C++ and conduct a membership test for two strings.

THEORY:

In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called Deterministic Automaton. As it has a finite number of states, the machine is called Deterministic Finite Machine or Deterministic Finite Automaton.

Formal Definition of a DFA

A DFA can be represented by a 5-tuple (Q, ∑, δ, q0, F) where -

- Q is a finite set of states.
- ∑ is a finite set of symbols called the alphabet.
- δ is the transition function where δ: Q × ∑ → Q
- q0 is the initial state from where any input is processed (q0 ∈ Q).
- F is a set of final state/states of Q (F ⊆ Q).

A DFA is represented by digraphs called state diagram.

- The vertices represent the states.
- The arcs labelled with an input alphabet show the transitions.
- The initial state is denoted by an empty single incoming arc.
- The final state is indicated by double circles.

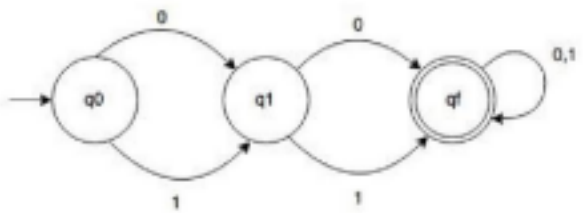
STEP 1:

Let us consider a simple Deterministic Finite Automata, for some Language: having more than one character, where all characters belong to the set {0,1}.

STEP 2:

Write a regular expression (RE) for the above Language as

RE= (0+1) (0+1) (0+1)*



DFA STATE DIAGRAM for L = (0+1) (0+1) (0+1)*, ∑ = {0,1}

Above is the DFA for the given language, which accepts only those words which consists of more than one character.

STEP 3:

Hence the machine

M = ({q0, q1, qf}, {0,1}, δ, q0, qf)

Where δ is given by the state table

STEP 4:

The following the state table for the above DFA:

δ =

Present State	New State	
→	0	1
q0	q1	q1
q1	qf	qf
qf	qf	qf

STEP 5:

Perform the Membership Test for following two strings:

a. 0110

b. 1

CODE:

```
#include <iostream>
#include <string>
using namespace std;

#define fast_io ios::sync_with_stdio(false); cin.tie(nullptr);

void solve() {
  string s;
  cin >> s;
  int state = 0;
  for (int i = 0; i<s.size(); i++) {
     if (s[i] != '0' && s[i] != '1') {
        cout << "Rejected" << endl;
        return;
     }
     switch (state) {
        case 0:
```

```
            state = 1;
            break;
        case 1:
            state = 2;
            break;
        case 2:
            state = 2;
            break;
      }
    }
    if (state == 2) cout << "Accepted" << endl;
    else cout << "Rejected" << endl;
}

int main() {
    fast_io;
    solve();
    return 0;
}
```

```
rahulgupta@Samirs-MacBook-Air Complier Design % cd "/Users/rahulgupta/Docum
ents/All-Files/C:C++/Complier Design/" && g++ qn1.cpp -o qn1 && "/Users/rah
ulgupta/Documents/All-Files/C:C++/Complier Design/"qn1
0101
Accepted
rahulgupta@Samirs-MacBook-Air Complier Design % cd "/Users/rahulgupta/Docum
ents/All-Files/C:C++/Complier Design/" && g++ qn1.cpp -o qn1 && "/Users/rah
ulgupta/Documents/All-Files/C:C++/Complier Design/"qn1
1
Rejected
```