



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 $(Q, \Sigma, \delta, q_0, F)$ where -

- Q is a finite set of states.
- Σ is a finite set of symbols called the alphabet.
- δ is the transition function where $\delta: Q \times \Sigma \rightarrow Q$
- q_0 is the initial state from where any input is processed ($q_0 \in Q$).
- F is a set of final state/states of Q ($F \subseteq Q$).

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.

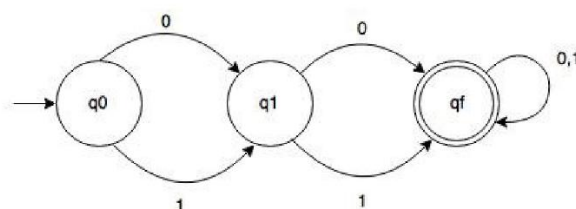
STEP 1:

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 $L = (0+1) (0+1) (0+1)^*$, $\Sigma = \{0,1\}$

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

$M = (\{q_0, q_1, q_f\}, \{0,1\}, \delta, q_0, q_f)$

Where δ is given by the state table

STEP 4:

The following the state table for the above DFA:

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

$\delta =$

STEP 5:

Perform the Membership Test for following two strings:

a. 0110

b. 1