CHAPTER

2

# Finite Automata (FA)

**(i) Derterministic Finite Automata (DFA)**

A DFA, $M = \langle Q, q_0, \Sigma, F, \delta \rangle$

Where, Q = set of states (finite)

$q_0 \in Q$ = the start/initial state

$\Sigma$ = input alphabet (finite)

(use only those symbols which create a particular string)

$F \subseteq Q$ = the set of final state

(Final state can be 0 (no final state) or more than 1 or it can have all states as final states.

$\delta$ = transition function (responsible for making the transition/movement from one state to other state

If machine halt at any final state that means it is accepting that string and if it halts at any other state it simply rejects the string.

**Note:** Behaviour of the machine is controlled by '$\delta$' responsable for switching of state from one to other state.

$\delta : Q \times \Sigma \rightarrow Q$ where, $Q$ = input state, $\Sigma$ = input alphabet

e.g. $\delta(q_i, a) = q_j$ where, $q_i$ = state, $a$ = reading and $q_j$ = move to other state

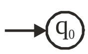
**Note:** Deterministic (exactly one choice)

(If you are at a state of read the input then after that you have to do the movement)

**Total :** On every state we need to define the transition for every symbol of input alphabet $\varepsilon$ i.e. we need to explain explicitly that where you want go after reading input.

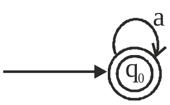
**Notations:**

(i) State : 

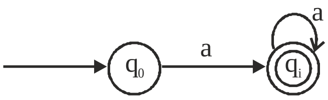
(ii) Initial state : 

(iii) Final state : 

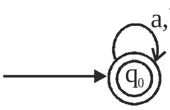

whenever, initial state becomes final state, null string $\lambda$ is accepted.

**Note: (i)** $\Sigma\{a\}^*$ 

Therefore, total $= a^* \{\lambda, a, a^2, a^3, ........\}$

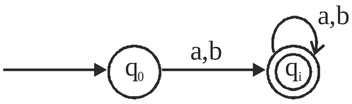
**(ii)** 

Therefore, $a^\dagger = \{a, a^2, a^3, ..........\}$

**(iii)** 

$\Sigma = \{a, b\}^* = \{\lambda, a, b, a^2, b^2, ab, ba..........\}$ (universal language)

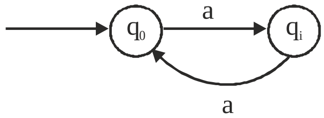
**(iv)**



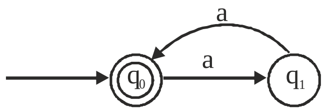

$$\Sigma = \{a, b\}^{+} = \{a, b, a^2, b^2, ab, ba.......\}$$

(not final, so can't accept (a))

**(v)**




**(vi)**

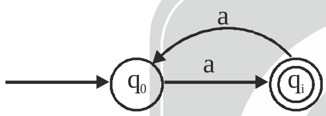



$$\left.\begin{array}{l} L = \{a^{2n} \mid n \geq 0\} \\ L = \{a^{n} \mid n \text{ is even } \#\} \\ L = \{a^{n} \mid n \bmod 2 = 0\} \end{array}\right\}$$

All the langauge are same

$$L = \{\lambda, a^2, a^4, a^6, a^8............\}$$

**(vii)**




$$\left.\begin{array}{l} L = \{a^{2n+1} \mid n \geq 0\} \\ L = \{a^{n} \mid \text{n is odd } \#\} \\ L = \{a^{n} \mid n \bmod 2 = 1\} \end{array}\right\} L = \{a, a^3, a^5, a^7, ..........\}$$

Although a given FA corresponds to only one language, a given language can have many FAs that accept it. Note that you must always be careful about the empty string: should the FA accept or not. In the preceding example, the empty string is accepted because the start state is also an accept state.

Another useful technique is remembering specific symbols. In the next example you must forever remember the first symbol; so the FA splits into two pieces after the first symbol.

## SOLVED PROBLEMS

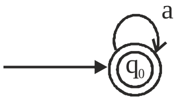
1. Construct the DFA for the following

(i) $L = \{a^m b^n \mid m \geq 0, n \geq 0\} = a^* b^*$

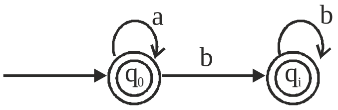
(ii) $L = \{a^n \mid n \geq 0 \text{ and } n \neq 4\}$

(iii) $L = \{a^n \mid n \bmod 5 \geq 2\}$

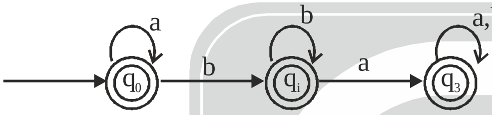
(iv) $L = \{w \in \{a\}^* \mid |w| \bmod 7 = 1\}$

**Soln.** (i) 

$\{\lambda, a, a^2, a^3 .......\}$

Therefore, 

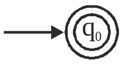
$\{\lambda, b, b^2, b^3 .......\} \Rightarrow \{ab, a^2b, a^2b^2, ab^2 .....\}$

Therefore, 

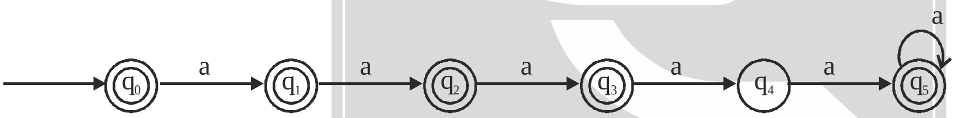

(Total concept follow, we need to define each and every state explicity in DFA).

(ii) $L = \{a^n \mid n \geq 0 \text{ and } n \neq 4\}$

$\{0, a, a^2, a^3, a^5, a^6, .........\}$



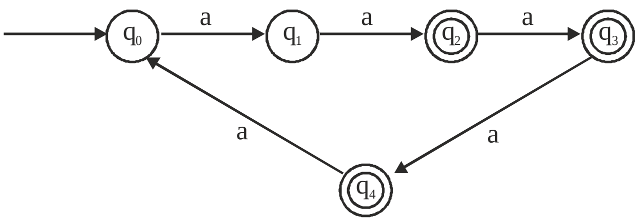
$\Sigma = \{a\}$




Therefore, 5 final state, 1 non-final state

(iii) $L = \{a^n \mid n \bmod 5 \geq 2\}$ $\Sigma = \{a\}$

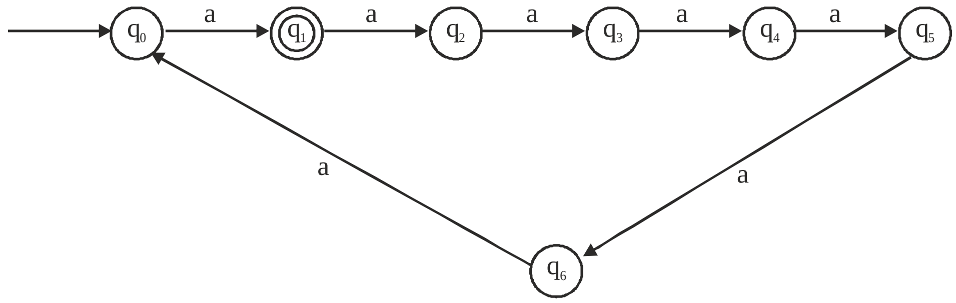
$\langle 0, 1, 2, 3, 4 \rangle$ Therefore, total 5 states.




(iv) $L = \{w \in \{a\}^* \mid |w| \bmod 7 = 1\}$

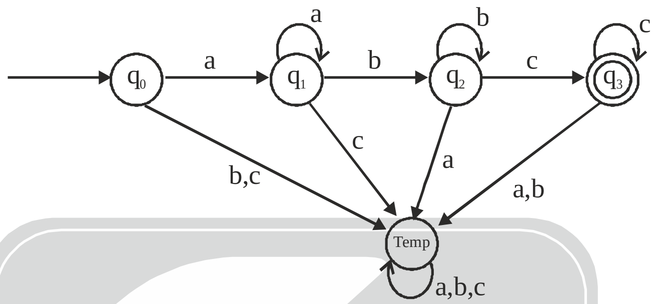
$w = a^m$ $\Sigma = \{a\}$ $\therefore n \bmod 7 = 1$

$\therefore$ $|w| = n$ $\therefore$ $\{0, 1, 2, 3, 4, 5, 6\} \Rightarrow$ Total 7 states



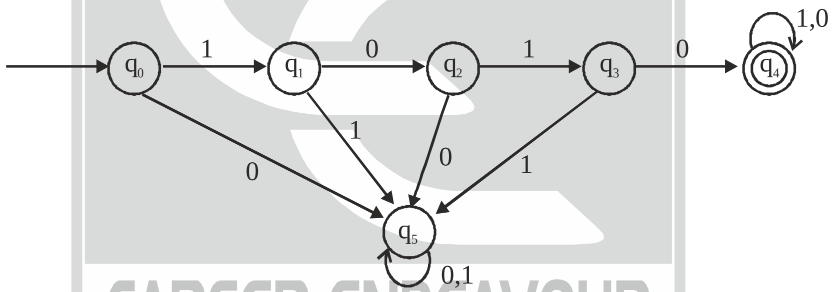

2. $L_2 = \{a^+ b^+ c^+\}$

$\{a^m b^n c^p \mid m \geq 1, n \geq 1, p \geq 1\}$



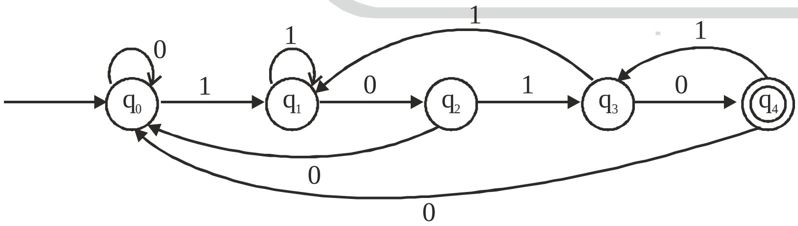

3. Design a DFA for the following language:
(Set of all binary strings which starts with 1010)

$\{1010w \mid w \in \{0,1\}^*\}$



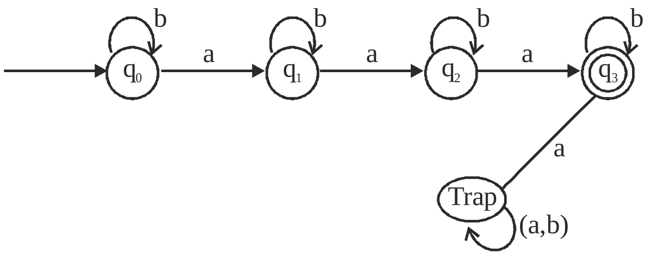



4. Design a DFA that ends with 1010

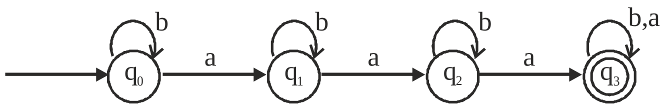



5. Design a DFA for the following over $\Sigma = \{a, b\}$

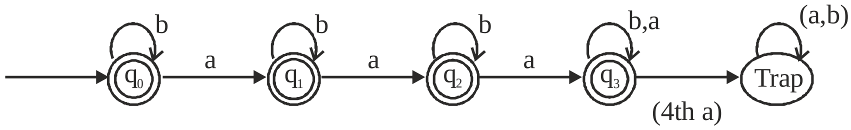
(i) The set of all strings containing exactly 3 a's.



(ii) Atleast 3 a's:



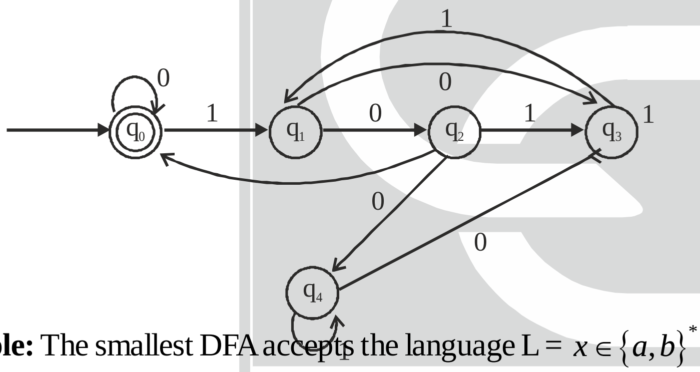

(iii) Atmost 3 a's:




**Transition table/tabular :** It is a matrix that lists the new state given the current state and the symbol read.

**Example :** Transition table for the FA that accepts all binary strings that begin and end with the same symbol.

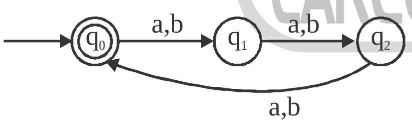
| State | Input 0 | Input 1 |
|---|---|---|
| S | A | C |
| A | A | B |
| B | A | B |
| C | D | C |
| D | D | C |




| | 0 | 1 |
|---|---|---|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_2$ | $q_3$ |
| $q_2$ | $q_4$ | $q_0$ |
| $q_3$ | $q_1$ | $q_2$ |
| $q_4$ | $q_3$ | $q_4$ |

**Example:** The smallest DFA accepts the language L = $x \in \{a,b\}^*$, length of n is multiple of 3.

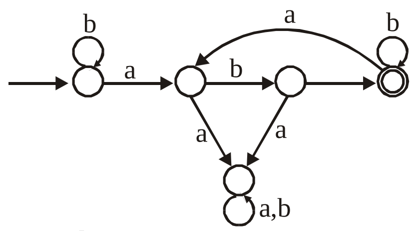
**[GATE-2002]**




(a) 2 (b) 3 (c) 4 (d) 5

**Soln.** **(b)**

**Example:** Consider the machine M **[GATE-2005]**




The language recognized by above machine is:

(a) $\left\{ \text{w} \in \{\text{a, b}\}^* / \text{ every a in w, is followed by exactly two b's} \right\}$

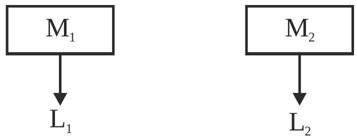
(b) $\left\{ \text{w} \in \{\text{a, b}\}^* \text{ every a in w, is followed by at least two b's} \right\}$

(c) $\left\{ \text{w} \in \{\text{a, b}\}^* \text{ w contains the substring 'abb'} \right\}$

(d) $\left\{w \in \{a,b\}^* \;\; w \text{ does not contain 'aa' as a substring}\right\}$

**Soln.** **(b)**

**Combining the machines:**




$\bar{L}_1$ **(complement of a FSM)**

I = the machine of language L is given

O = the machine of language $\bar{L}$

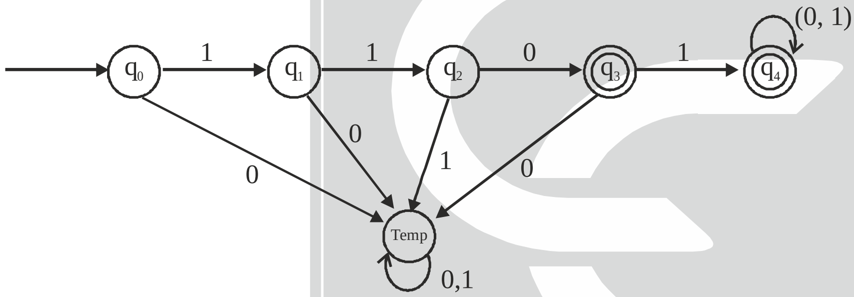
**Steps:**

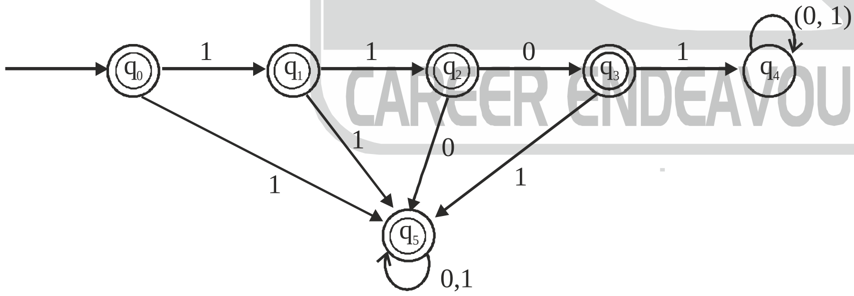
(i) Make the final states to non-final states and

(ii) Non-final states to final states (Final $\leftarrow$ non-final)

**Problem:** Design a DFA which accepts the set of all binary string which does not starts with 1101.

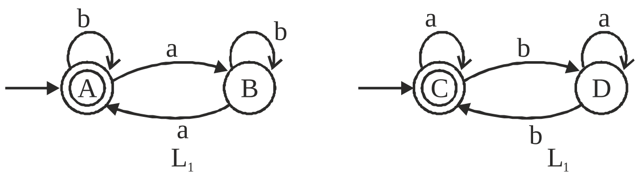
**Soln.**




**Problem:** Design a DFA for $L = \left\{w \in \{a,b\}^* \mid N_a(w) \text{ is even AND } N_b(w) \text{ is even}\right\}$

**Soln.**




$L_1 \cap L_2 \Rightarrow m \times n$

$A \xrightarrow{a} B$ ; $C \xrightarrow{a} C$ . Therefore, BC

Here, first we will check that on A state, as we read ‘a’ we move to B on C state, as we read ‘a’ we move to C.

Therefore, AC state complex move to BC state.

$A \xrightarrow{b} A$ ; $C \xrightarrow{b} D$

$B \xrightarrow{b} B \qquad B \xrightarrow{a} A$

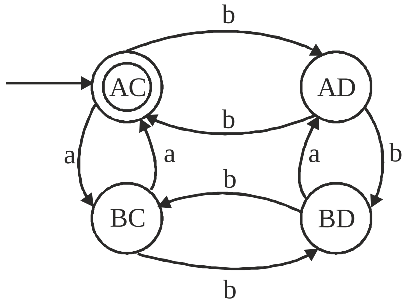
$D \xrightarrow{b} C \qquad C \xrightarrow{a} C$

$B \xrightarrow{b} B \qquad B \xrightarrow{a} A$

$D \xrightarrow{b} C \qquad D \xrightarrow{a} D$

$A \xrightarrow{b} A \qquad A \xrightarrow{a} B$

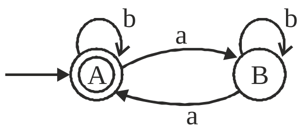
$D \xrightarrow{b} C \qquad D \xrightarrow{a} B$



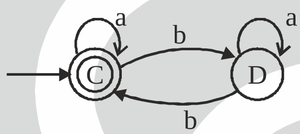

If $L_1$ machine have m state and $L_2$ machine have n state, then the combined machine $(L_1 \cap L_2)$ have m×n state.

**• If the OR operation is applied then**

$$L = \left\{ w \in (a,b)^* \mid N_a(w) \text{ is even OR } N_b(w) \text{ is even} \right\}$$

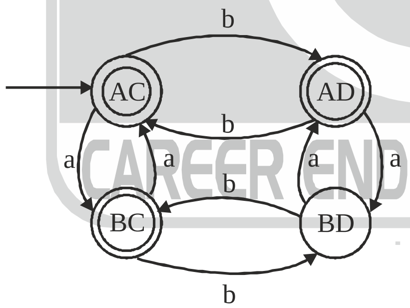



$L_1$




$L_2$

Therefore, the number of final states will change.
As here, A and C are final state, so where the A and C come, it will become final state






- $L_1 - L_2 = \{ w \mid w \in L_1 \text{ and } w \notin L_2 \}$
  So, $L_1$ is final and $L_2$ is non final > $L_1$ and $L_2$
  If $L_2 - L_1 \therefore BC$ will be final.

- $L_1 \oplus L_2 = (L_1 - L_2) \cup (L_2 - L_1)$
  $\therefore \quad L_1 \oplus L_2$ = AD and BC both will be final states.

**Non-Deterministic Finite Automata (NFA)**

A machine, $M = (Q, q_0, \Sigma, F, \delta)$

Where, Q = the set of states (finite)

$q_0 \in Q$ = initial/starting state

$\Sigma$ = input alphabet (finite)

$F \subseteq Q$ = final state is the subset of Q

$\delta$ = transition function $\rightarrow$ responsible for moving the movement from one state to another state.

$DFA = \{\delta = Q \times \Sigma \rightarrow Q(\text{total})\}$ in this null transition is not allowed.

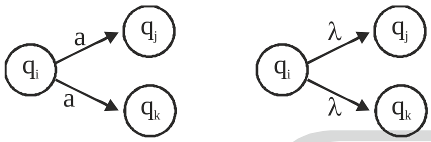
$NFA = [\delta = Q \times \Sigma] \rightarrow 2^Q \rightarrow (\text{power set of Q})$

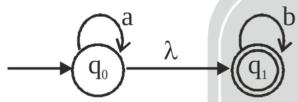
- Power of set Q, $2^Q = P(Q) = \{\phi, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$

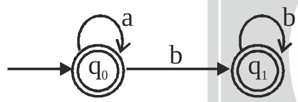
  $\boxed{Q = \{q_0, q_1\}}$

- In NFA we have zero (0) choice, one choice, or more than one choice depending upon the states, so it is called non-deterministic.

**Transition in NFA :**

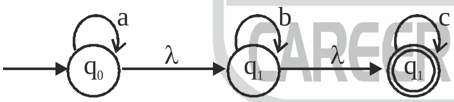
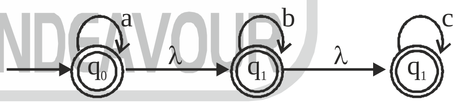


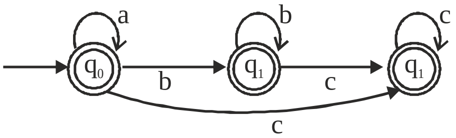
e.g.  For $a^*b^*$

e.g.  Also for $a^*b^*$

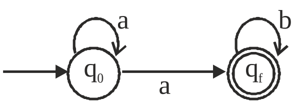
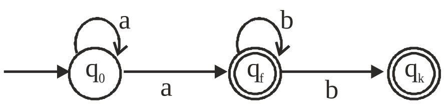
**Note:** If in DFA, all the states are final states, then langauge is considered as universal language. $\left(\Sigma^* = \{a, b\}^*\right)$. But, in NFA, if all the states are final states then it is not necessary that it is a universal language (like a*b*).

- **Design a NFA for a*b*c*:**

(1)  OR 

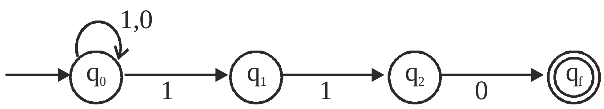
(2) 

In NFA, trap state is not considered becuase if transition is not defined for a particular I/P symbol or a state, then given string is rejected automatically through the machine i.e. check for abaab.

e.g.  or 

The machine rejects abaab.

**Problem:** Design an NFA over $\Sigma = \{0,1\}$ accepting the set of all binary strings ending with 110.

**Soln.** 

Here, we need not to think about other transitions.