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

Σ = 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.

δ = 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 'δ' responsable for switching of state from one to other state.

$\delta : Q \times \Sigma \to Q$ where, Q = input state, Σ = 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)

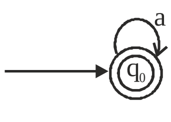
Total : On every state we need to define the transition for every symbol of input alphabet ε i.e. we need to explain explicitly that where you want go after reading input.

Notations:

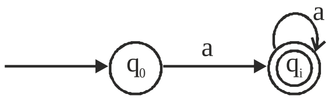
(i) State : q_i (ii) Initial state : $\to q_0$ (iii) Final state : q_s

whenever, initial state becomes final state, null string λ is accepted.

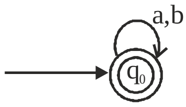
Note:

(i) 

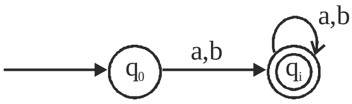
Therefore, $a^* = \{\lambda, a, a^2, a^3, ...\}, \Sigma = \{a\}$

(ii) 

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

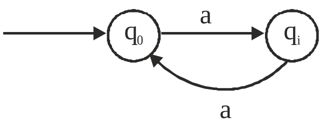
(iii) 

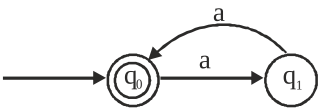
Therefore, $\{a, b\}^* = \{\lambda, a, b, a^2, b^2, ab, ba..........\}, \Sigma = \{a, b\}$ (universal language)

(iv) 

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

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

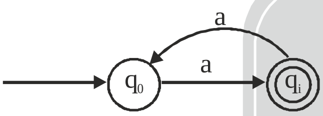
(v) 

(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

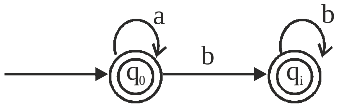
1. Construct the DFA for the following

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

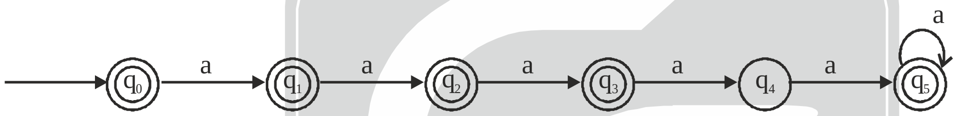
(iv) $L = \{w \in \{a\}^* \mid |w| \bmod 7 = 1\}$

Soln. (i)  $\{\lambda, a, b, ab, aab, abb, ...\}$

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

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

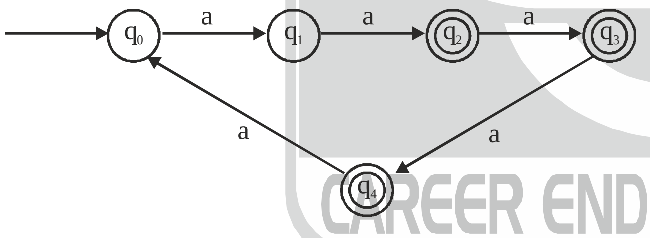
$\{\lambda, 0, a, a^2, a^3, a^5, a^6, ...\}$, $\Sigma = \{a\}$



Therefore, 5 final state, 1 non-final state

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

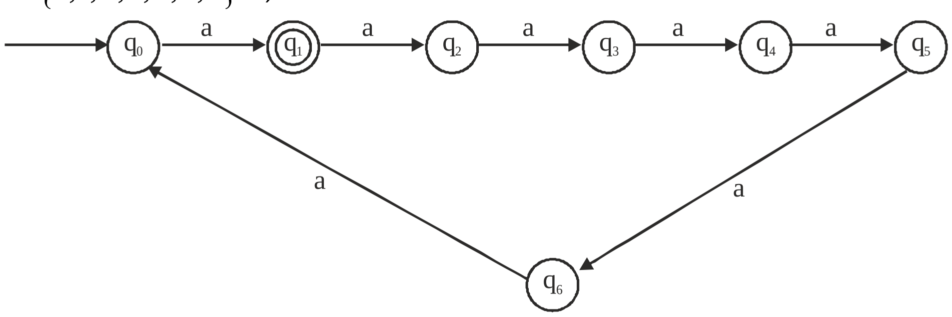
$\{0, 1, 2, 3, 4\}$. Therefore, total 5 states.



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

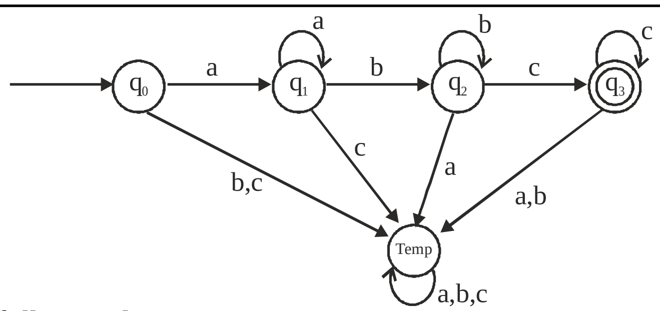
$\Sigma = \{a\}$, $\therefore n \bmod 7 = 1$

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



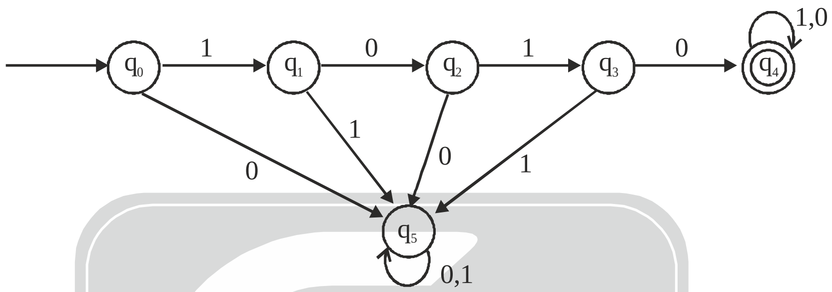
2. $L_2 = \{a^+ b^+ c^+\}$

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

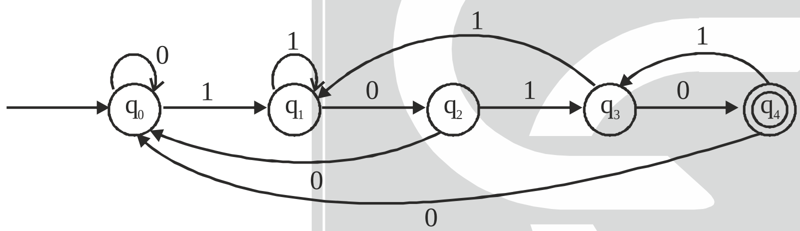


3. Design a DFA for the following language:
(Set of all binary strings which starts with 1010)

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

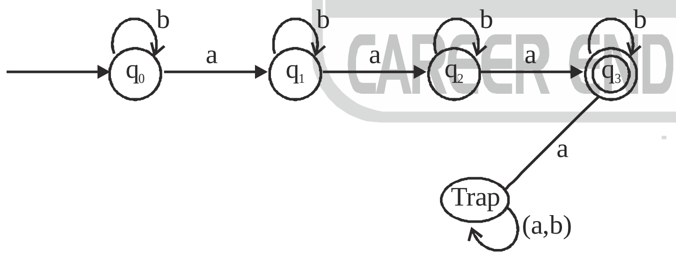


4. Design a DFA that ends with 1010

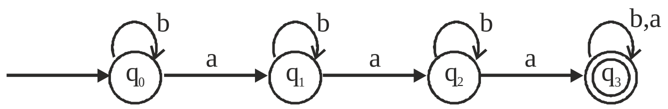


5. Design a DFA for the following over $\Sigma = \{a, b\}$

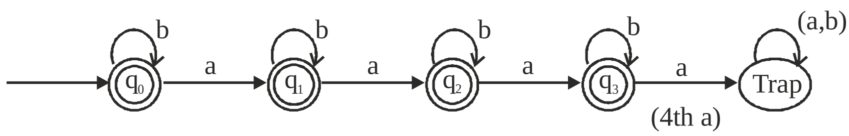
(i) The set of all strings containing exactly 3 a's.



(ii) Atleast 3 a's:

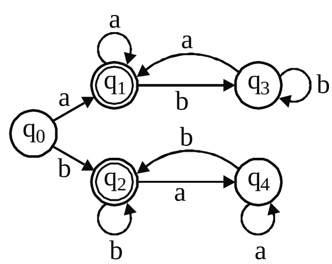


(iii) Atmost 3 a's:



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

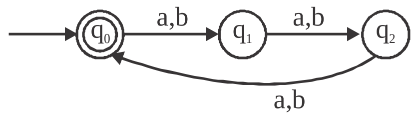
Example : Transition table for the FA that accepts all binary strings that begin and end with the same symbol.



	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_4	q_2
q_3	q_1	q_3
q_4	q_4	q_2

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

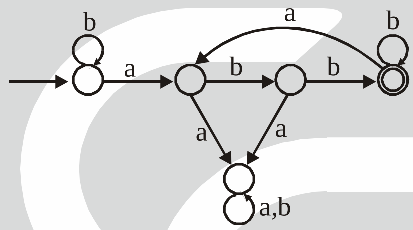
[GATE-2002]



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

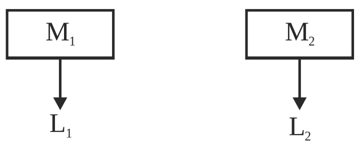
(b) $\left\{w \in \{a, b\}^* \text{ every a in w, is followed by at least two b's}\right\}$

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

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

Soln. **(b)**

Combining the machines:



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

Steps:

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

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