DFA	NFA
1) $\delta : Q \times \Sigma - Q$	⇒ $\delta : Q \times \Sigma - 2^Q$
2) Reading one input moves to only one state	⇒ on reading one input it can goto zero (or) more states
3) All states has to read all i/p's	⇒ It may/may not read all inputs
4) For an accepted strings there is only one path	⇒ For an Accepted string has many paths out of which only one is correct path.

Non Deterministic Finite Automata (NFA)

NFA is defined by five tuples

$(Q, \Sigma, \delta, S, F)$

Q - Finite non empty set of states

Σ - Finite set of i/p

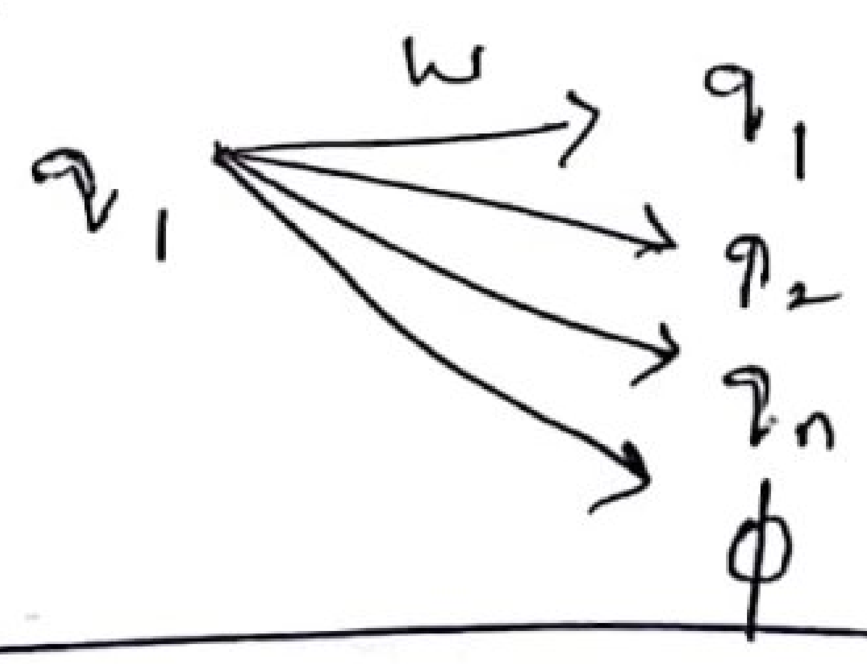
δ - Mapping $Q \times \Sigma \rightarrow 2^Q$ (is powerset of Q)

S - Start state

F - Set of Final states

$q_1 \xrightarrow{w} q_2$

DFD

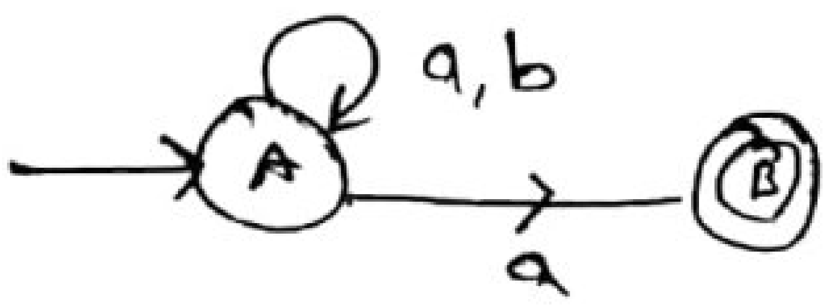


Some problem needs solution on back tracking (or) by exhaustive search

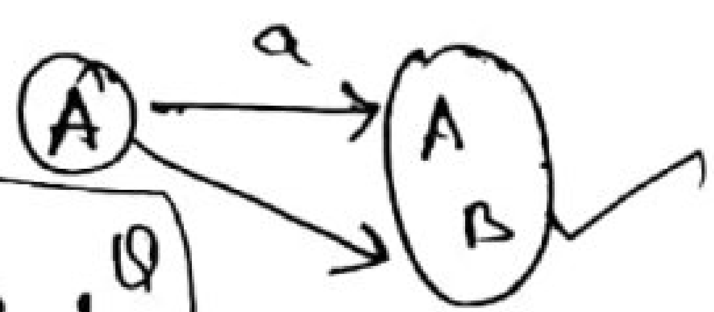
Q) $\Sigma = \{a, b\}$

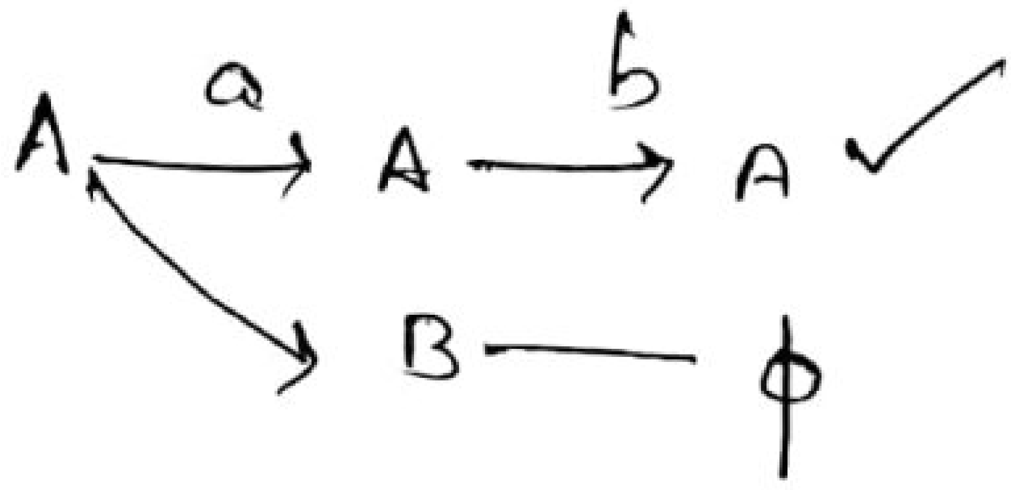
$L = \{$ end with 'a' $\}$

$L = \{a, aa, ba, \ldots\}$



NFA

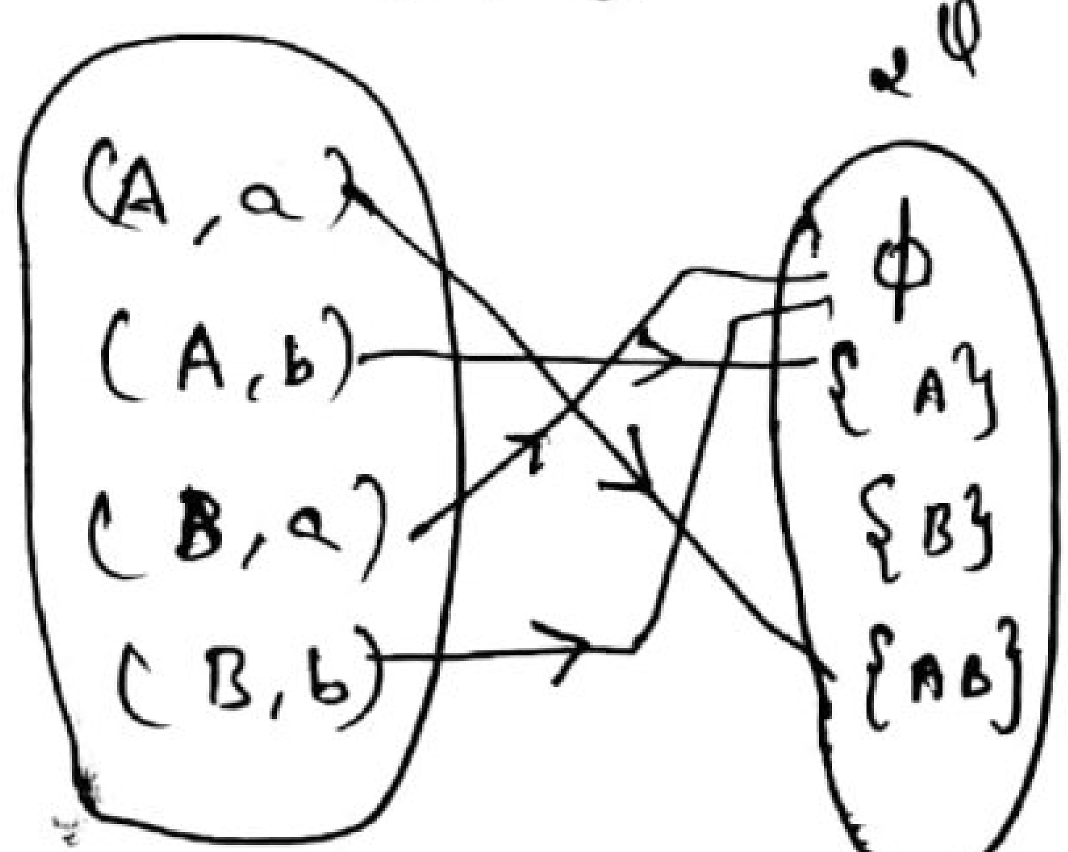




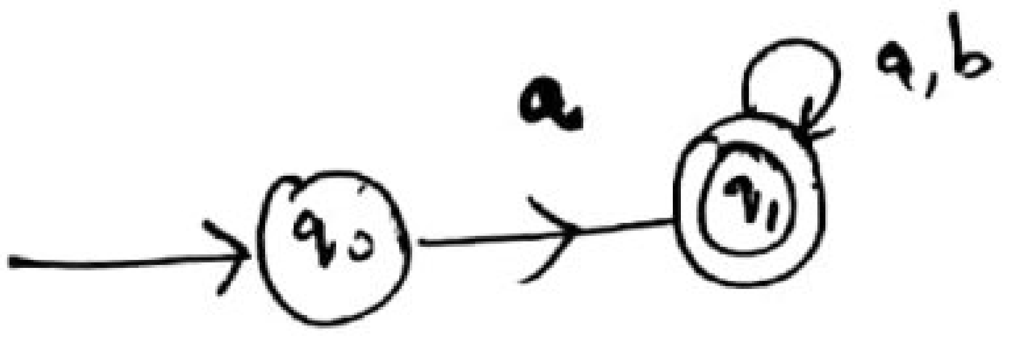
$\boxed{\delta \Rightarrow Q \times \Sigma \rightarrow 2^Q}$

$Q = \{A, B\}$ $\quad \Sigma = \{a, b\}$

$Q \times \Sigma = \{$



Q) $L = \{$ str start with 'a' $\}$

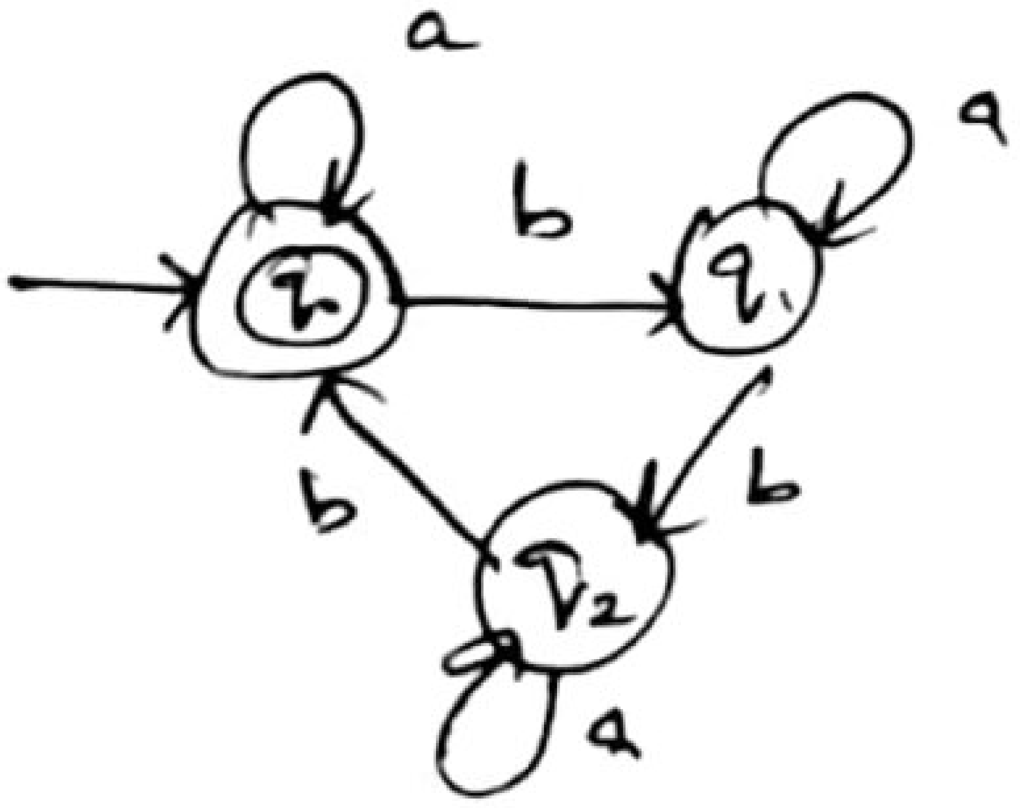


$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_1$ ✓

$q_0 \xrightarrow{b} \phi \quad a$

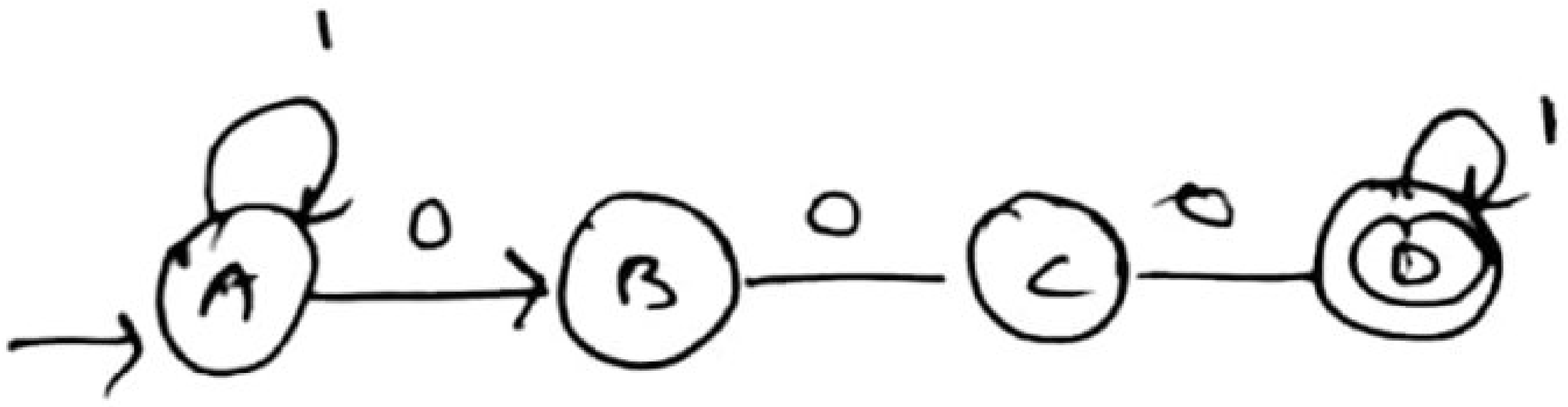
dead configuration

Q) No of b's divisible by 3.

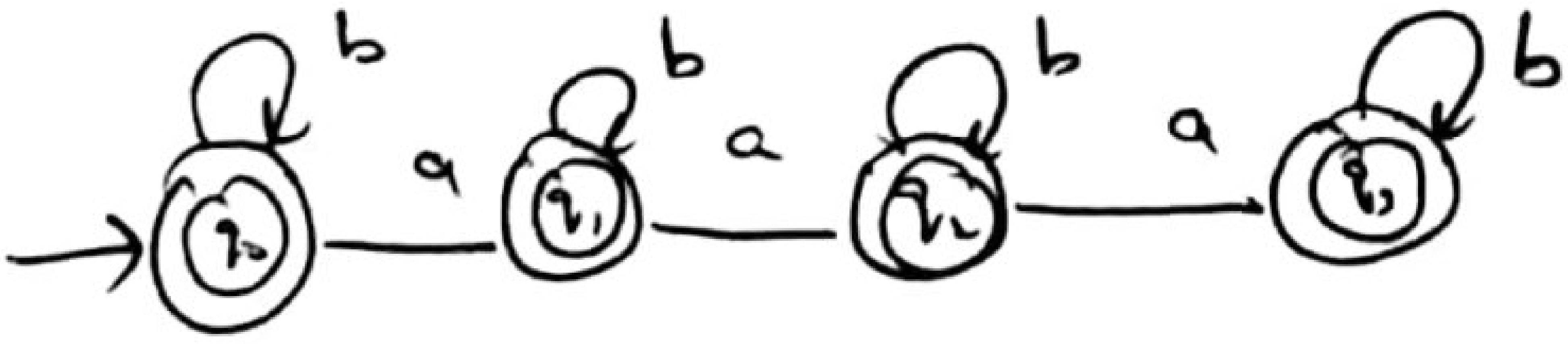


Q) Substring 1101

Q) Exactly 3 consecutive 0's

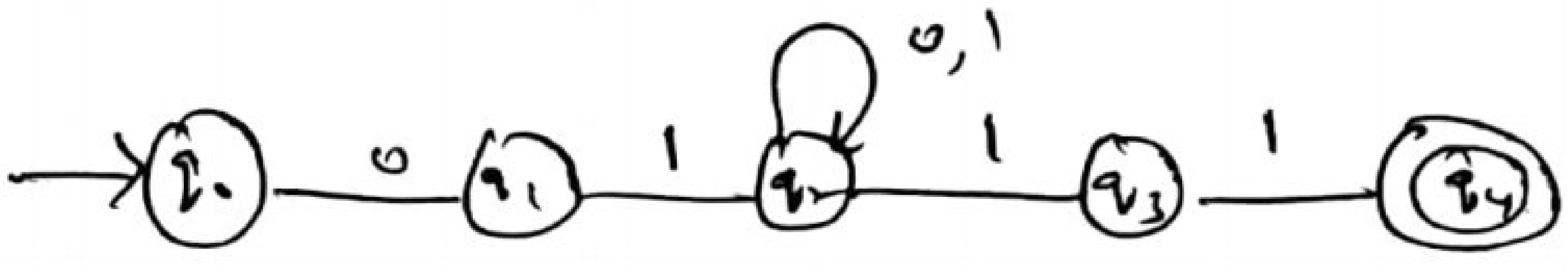


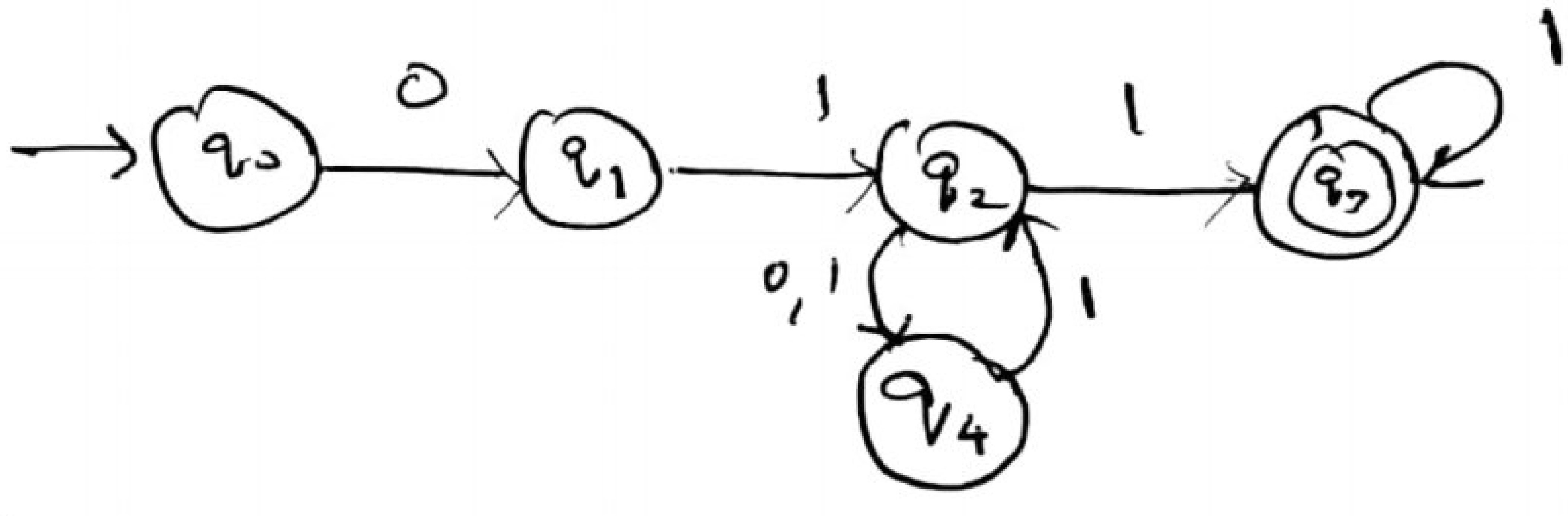
Q) Not more than 3 a's



Q) Not containing 110

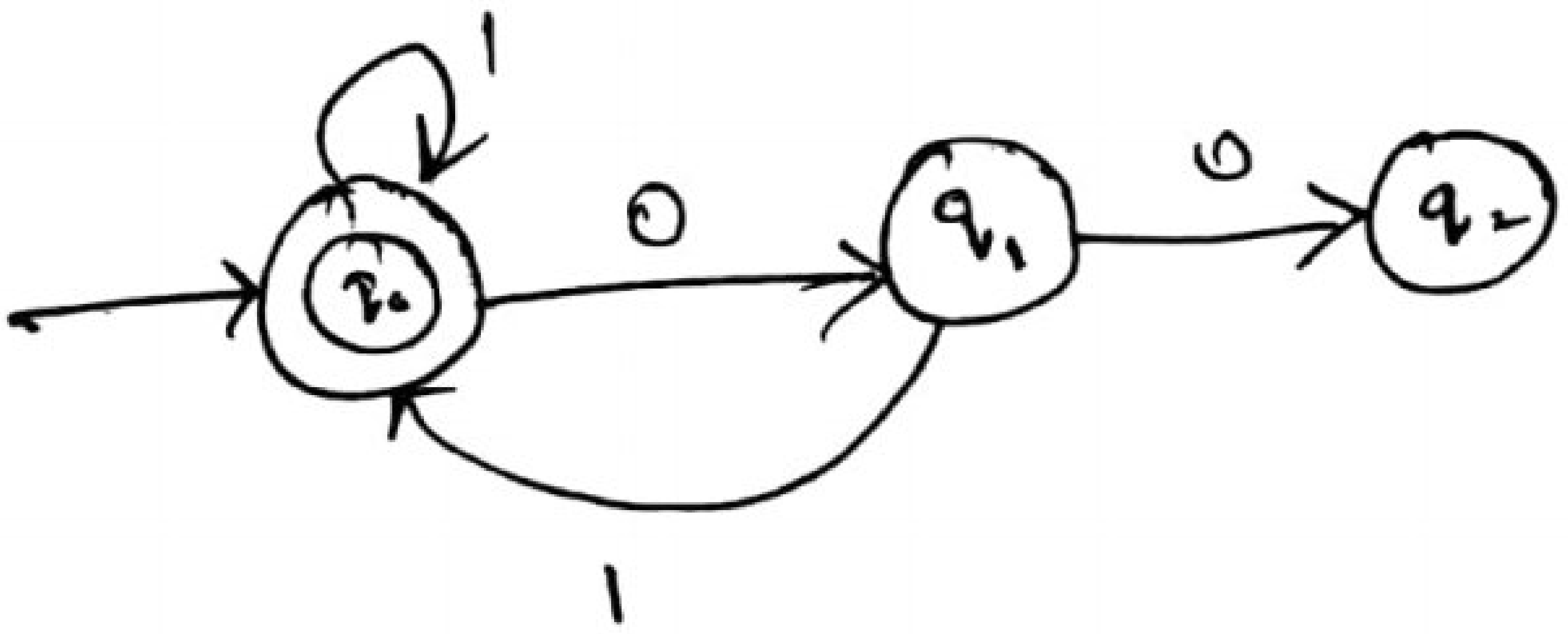
Q) Begin with 01 ends with 11





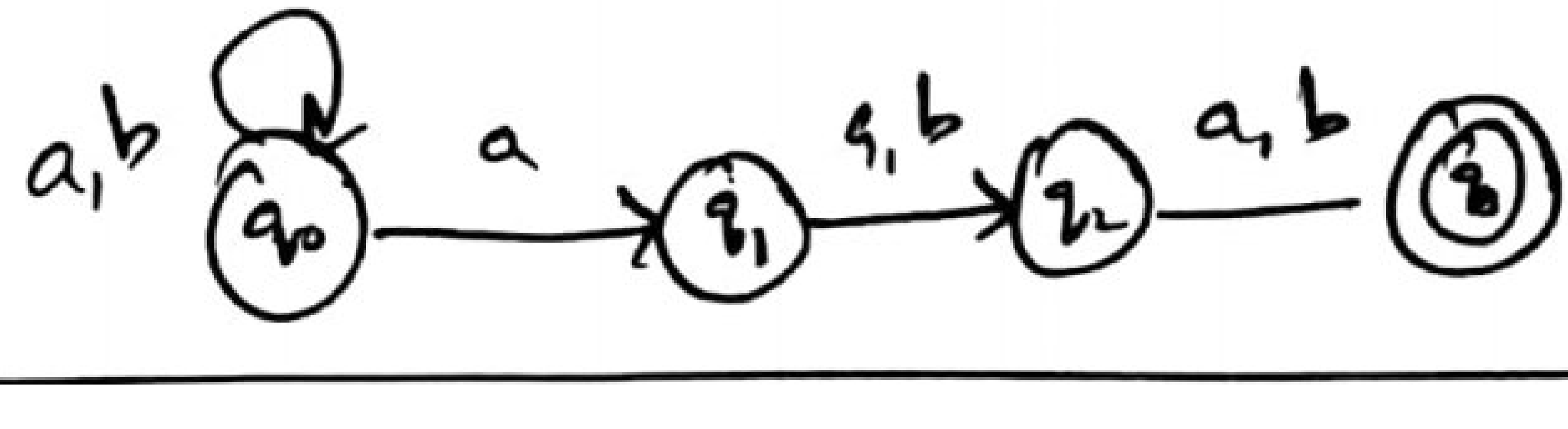
Assignment

Q) Determine an NFA, M accepting $L = \{ w \mid w \in \{0,1\}^* \}$ where every 0 in w has a '1' immediately to its right



Q) obtain an NFA which should accept a Language L, where $L = \{ x \in \{a,b\}^*$ $|x| \geq 3 \}$ and third symbol of x from the right is 'a'

$L = \{$



Which one is powerful

both NFA & DFA are equally powerful.

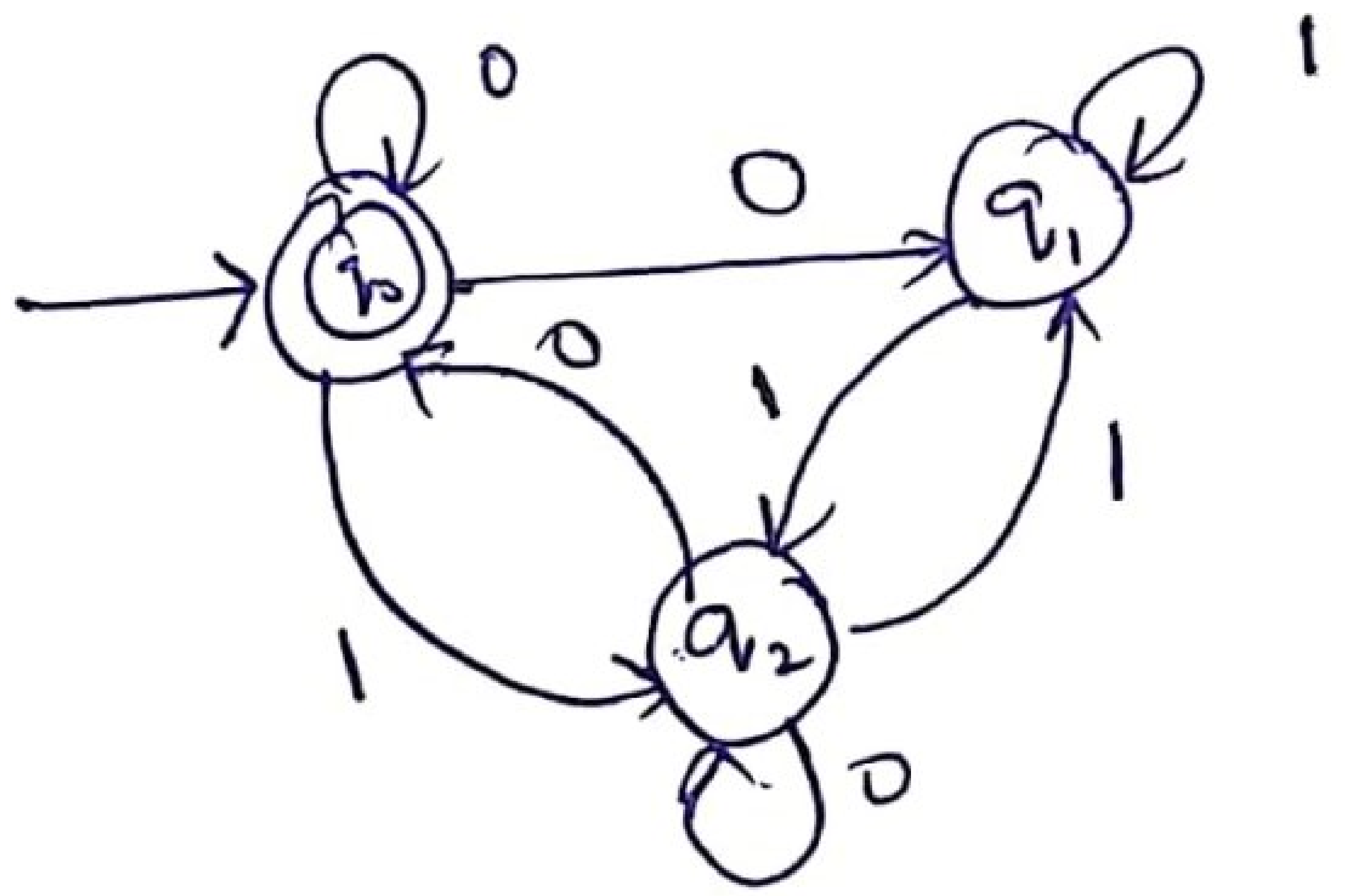
$$\checkmark \ NFA \xrightarrow{\text{Convert}} DFA$$

$$\times \ DFA \longrightarrow NFA$$

$$NFA \cong DFA$$
equivalent.

Ans

Q) For the NFA shown, check whether the i/p string 0100 is accepted (or) not ?



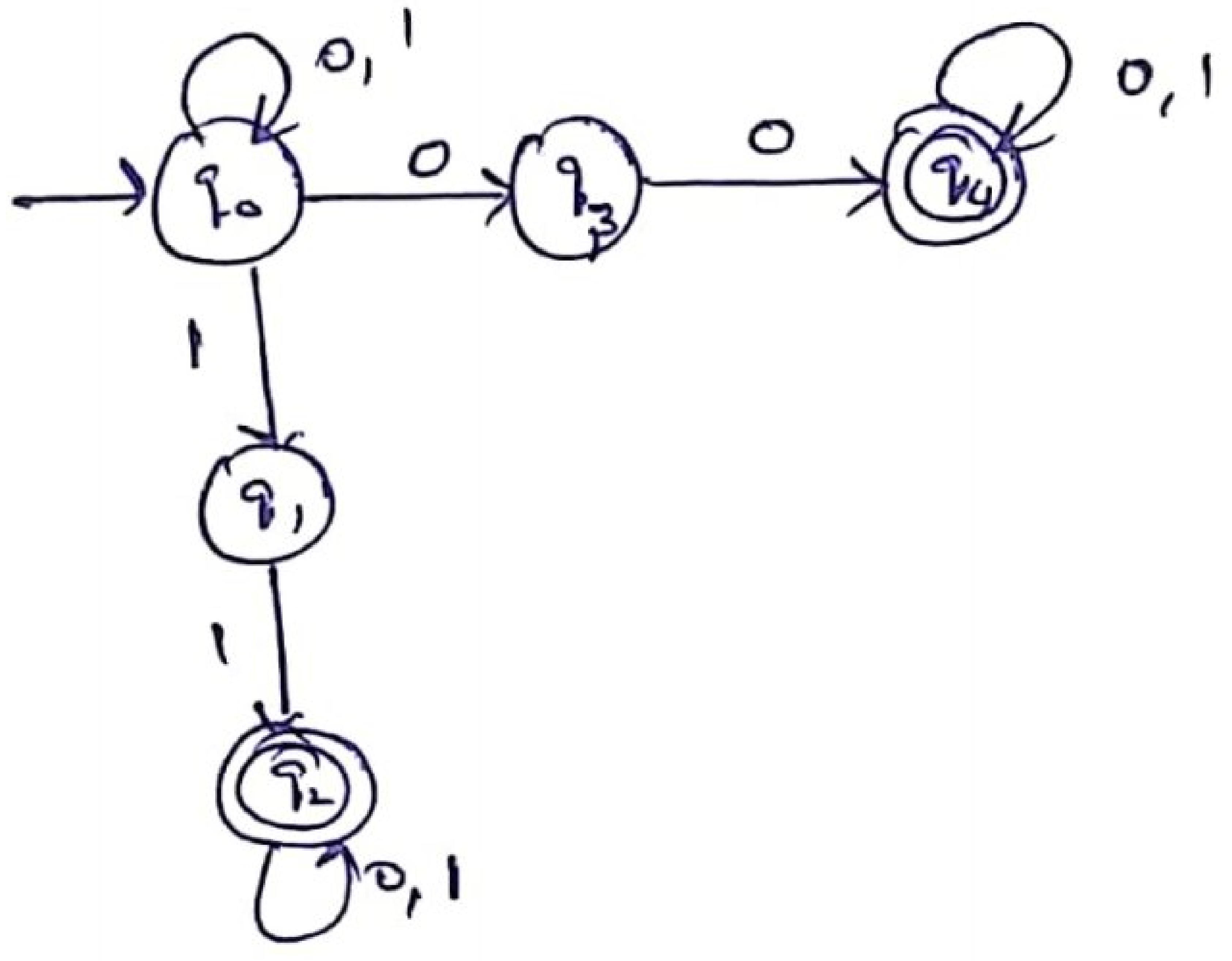
Ans

Q) NFA with states {1, 2, 3, 4, 5} & input Σ = {a, b} has the transition table.

calculate
δ(1, ab)
δ(1, abab)

states	a	b
→1	1, 2	1
2	3	3
3	4	4
4	5	ϕ
*5	ϕ	5

Q) Consider the given NFA to check whether $w = 01001$ is valid (or) not



Sol $M = (Q, \Sigma, \delta, S, F)$ is NFA

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

$S = \{q_0\}$

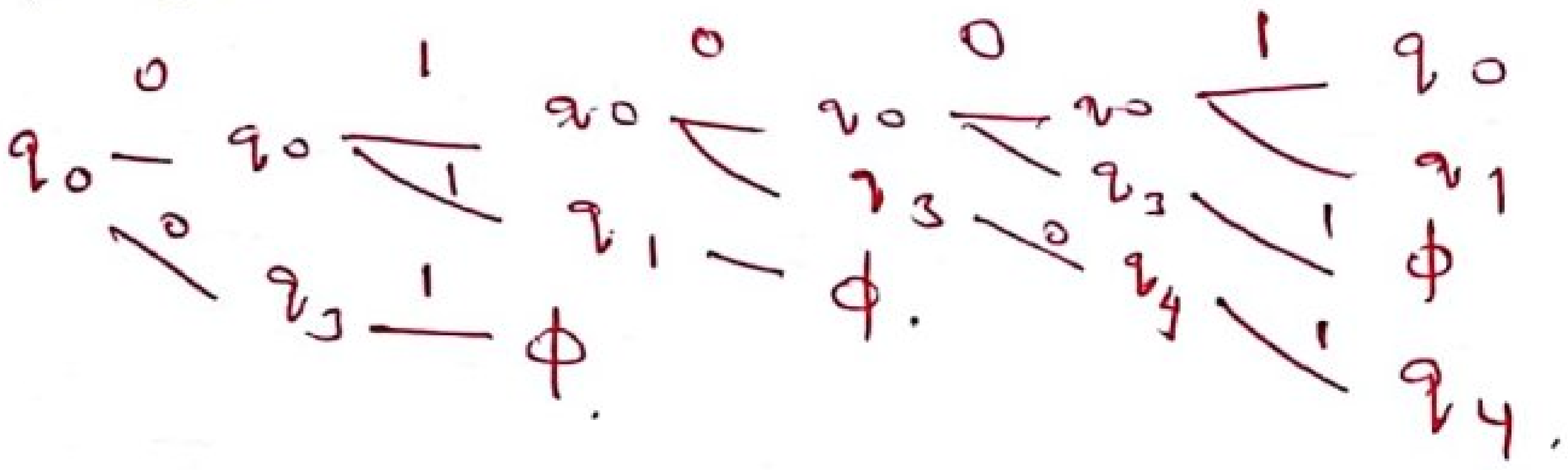
$F = \{q_2, q_4\}$

Transition table

State	i/p	
	0	1
q_0	q_0, q_3	q_0, q_1
q_1	ϕ	q_2
$* q_2$	q_2	q_2
q_3	q_4	ϕ
$* q_4$	q_4	q_4

Method 1

Q// 01001



Method 2

$\delta(q_0, 0) = \delta(\delta(q_0, 0))$
$= \{q_0, q_3\}$

$\delta(q_0, 01) = \delta(\delta(q_0, 0), 1)$
$= \delta(\{q_0, q_3\}, 1)$
$= \delta(q_0, 1) \cup \delta(q_3, 1)$
$= \{q_0, q_1\} \cup \phi$
$= \{q_0, q_1\}$

$\delta(q_0, 010) = \delta(\delta(q, 01), 0)$
$= \delta(\{q_0, q_1\}, 0)$
$= \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \{q_0, q_3\} \cup \phi = \{q_0, q_3\}$

$\delta(q_0, 0100) = \delta(\delta(q, 010), 0)$
$= \delta(\{q_0, q_3\}, 0)$
$= \delta(q_0, 0) \cup \delta(q_3, 0)$
$= \{q_0, q_3\} \cup \{q_4\} = \{q_0, q_4, q_3\}$

$\delta(q_0, 01001) = \delta(\delta(q, 0100), 1)$
$= \delta(\{q_0, q_3, q_4\}, 1)$
$= \delta(q_0, 1) \cup \delta(q_3, 1) \cup \delta(q_4, 1)$
$= \{q_0, q_1\} \cup \{q_4\} \cup \{q_4\}$
$= \{q_0, q_1, q_4\}$

$\delta(q_0, 01001) \cap F =$
$= \{q_0, q_1, q_4\} \cap \{q_4, q_4\}$
$= q_4$

Hence Accepted

EXTENDED TRANSITION FUNCTION FOR NFA

BASICS :-

$$\hat{\delta}(q, \epsilon) = \{q\}$$

Being in a state Reading no i/p the system remains in the same state.

INDUCTION :- Suppose $w = xa$, where a is the final i/p and x is rest of i/p

$$\hat{\delta}(q, x) = \{P_1, P_2, \ldots P_K\}$$

Let $\bigcup_{i=1}^{K} \delta(P_i, a) = \{r_1, r_2, \ldots r_n\}$

then $\hat{\delta}(q, w) = \{r_1, r_2, \ldots r_m\}$

$\hat{\delta}(q, w)$ is computed by first computing $\hat{\delta}(q, x)$ and by the following any transition from any of those states that is labelled a.

Language :-

Language Acc by NFA is

$$L(A) = \{w : \hat{\delta}(q_0, w) \cap F \neq \phi\}$$

EQUIVALENCE OF NFA & DFA

Since every DFA is an NFA, it is clear that the class of languages accepted by NFA's includes the regular sets

INDUCTIONS :- Let L be a set accepted by NFA then there exists a DFA that accepts L.

PROOFS :-

This is implemented by using subset Construction method, because it involves constructing all subsets of the set of states of the NFA

Let $N = \{Q_N, \Sigma, \delta_N, q_0, F_N\}$

$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$

The states of D are all the subsets of the set of set states of N

$$Q_D = 2^{Q_N}$$

An element of Q_D will be denoted by

$$[q_1, q_2, \ldots q_i]$$

where $q_1, q_2 \ldots q_i$ are in Q_D

Define $\delta_D([q_1, q_2, \ldots, q_i], a) =$

$$[P_1, P_2 \ldots P_j]$$

if and only if

$$\delta_N(\{q_1, q_2, \ldots q_i\}, a) = \{P_1, P_2, \ldots P_j\}$$

δ_D is computed by applying δ_N to each state of Q_D represented by $[q_1, q_2 \cdots q_i]$. On applying to each of $q_1, q_2 \cdots q_i$ and taking the union,

We get $[p_1, p_2 \cdots p_i]$ in Q_D

To show

$$\delta_D(\{q_0\}, x) = [q_1, q_2, \cdots q_i]$$

if and only if

$$\delta_N(q_0, x) = \{q_1, q_2, \cdots q_i\}$$

This is proved by the method of induction

F_D is the set of subsets S of Q_N such that $S_n \cap F_n \neq \phi$