Configuration of DFA

→ A configuration or an Instantaneous Discription of a DFA gives the information about the current state and the unread part of the string being processed.

→ Formally configration of a DFA $M = (Q, \Sigma, \delta, q_0, F)$ is an element of $Q \times \Sigma^*$.

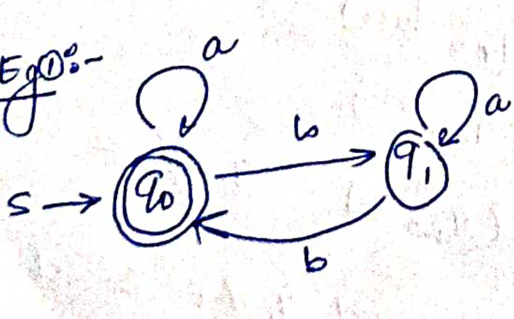
→ For a given input string $x \in \Sigma^*$ the initial configuration is (q_0, x) and a final configuration of M is of the form (P, ϵ), where $P \in F$.

→ Let $C = (P, x)$ and $C' = (q, y)$ be two configurations, we say that M moves from C to C' in one step is detoned as. $C \vdash_M C'$ iff $\delta(P, a) = q$, $x = ay$, $x, y \in \Sigma^*$ and $a \in \Sigma$.

→ Note that, $\vdash_M : Q \times \Sigma^+ \rightarrow Q \times \Sigma^*$ is a function, i.e. for every configuration except of the from (q, ϵ) there is a unique determined next configuration.

→ A configuration of the form (q, ϵ) signifies that M has consumed all its inputs, and hence the operation ceases at this point.

o Eg①:-

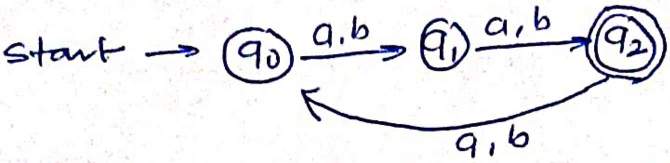


o if M is given the input aabba, its initial configuration is $(q_0, aabba)$ then.

$$(q_0, aabba) \vdash_M (q_0, abba)$$
$$\vdash_M (q_0, bba)$$
$$\vdash_M (q_1, ba)$$
$$\vdash_M (q_0, a)$$
$$\vdash_M (q_0, \epsilon)$$

∴ Therefore,
$(q_0, aabba) \vdash^*_M (q_0, \epsilon)$,
and so aabba is accepted by M.

Eg:- 2



$\Sigma = \{ w \in \{a,b\}^* : |w| \bmod 3 = 2 \}$

Problem :: check for the input aabaa and bbabba is accepted by M or not.

- if M is given the input aabaa, its initial configuration will be $(q_0, aabaa)$, then.

$$
\begin{aligned}
(q_0, aabaa) &\vdash_M (q_1, abaa) \\
&\vdash_M (q_2, baa) \\
&\vdash_M (q_0, aa) \\
&\vdash_M (q_1, a) \\
&\vdash_M (q_2, \epsilon) \quad \therefore q_2 \in F.
\end{aligned}
$$

As $(q_0, aabaa) \vdash_M^* (q_2, \epsilon)$ so string aabaa is accepted by M.

- if M is given the input bbabba, its initial configuration will be $(q_0, bbabba)$, then,

$$
\begin{aligned}
(q_0, bbabba) &\vdash_M (q_1, babba) \\
&\vdash_M (q_2, abba) \\
&\vdash_M (q_0, bba) \\
&\vdash_M (q_1, ba) \\
&\vdash_M (q_2, a) \\
&\vdash_M (q_1, \epsilon) \quad q_1 \notin F.
\end{aligned}
$$

As $(q_0, bbabba) \vdash_M^* (q_1, \epsilon)$ and $q_1 \notin F$ so string bbabba is rejected by M.

Non-Deterministic Finite Automata (NFA)

→ NFA includes non-deterministic transitions.

→ A transition is called non-deterministic if there are several (Possibly zero) next states from a state on an input symbol or without any input.

→ A NFA is much more convenient to design than a DFA.

→ A transition without an input is called "ε-transition".

→ Formally, a NFA is a quintuple $N = (Q, \Sigma, \delta, q_0, F)$, where,

① Q is finite set called set of states.

② Σ is a finite set called the input alphabet.

③ $q_0 \in Q$, called the initial state,

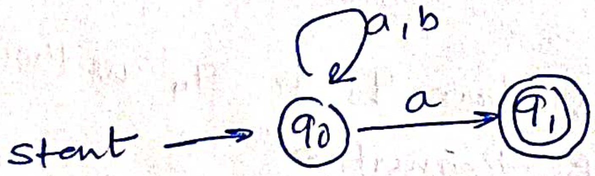
④ $F \subseteq Q$, called the set of final/accept states,

⑤ $\delta : Q (\Sigma \cup \{\epsilon\}) \rightarrow \underbrace{2^Q}_{\text{power set of } Q}$, called transition function.

∴ δ will return set of states for a given input at any state.

Eg①:- Let $\Sigma = \{a, b\}$, $\mathcal{L} = \{$ set of all strings end with 'a' $\}$

Let $\mathcal{L} = \{a, aa, ba, aaa, aba, baa, bba, \dots\}$



δ	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{\}$	$\{\}$

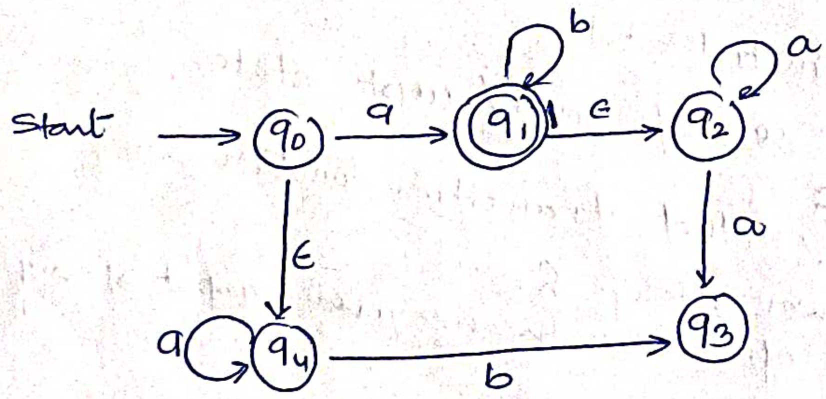
• Let $Q = \{q_0, q_1\}$

$2^Q = \{\phi, \{q_0\}, \{q_1\}, \{q_0, q_1\}\}$

∴ Every DFA is also NFA.

Eg:- Let $\Sigma = \{a, b\}$, $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $F = \{q_1, q_3\}$ and δ be given by the following transition table:

δ	a	b	ϵ
q_0	$\{q_1\}$	$\{\}$	$\{q_4\}$
q_1	$\{\}$	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_2, q_3\}$	$\{q_3\}$	$\{\}$
q_3	$\{\}$	$\{\}$	$\{\}$
q_4	$\{q_4\}$	$\{q_3\}$	$\{\}$



→ Some non-deterministic transitions in this NFA.

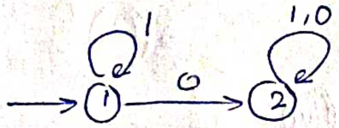
1. There is no transition from q_0 on input symbol b.
2. There are multiple (two) transitions from q_2 on input symbol a.
3. There is a transition from q_0 to q_4 without any input i.e., "ϵ-transition"

- Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Given an input string $w = a_1 a_2 \dots a_k$ and a state $p \in Q$, the set of next states $\hat{\delta}(p, w)$ can be easily computed using a tree structure, called computation tree.

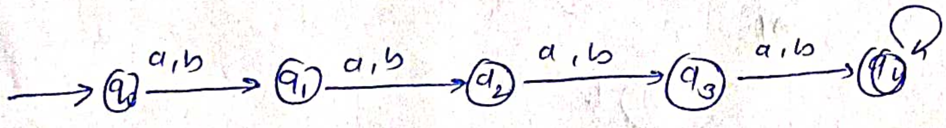
Problem Solving - (2)

Date : 03.11.2021.

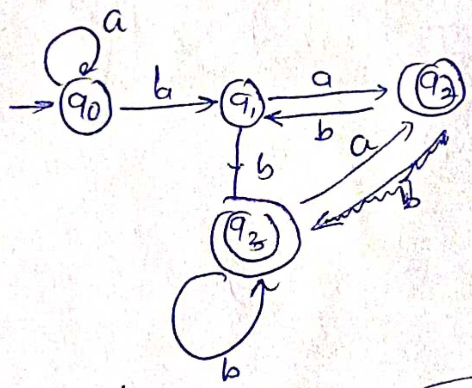
(1) mini DFA.



(2)

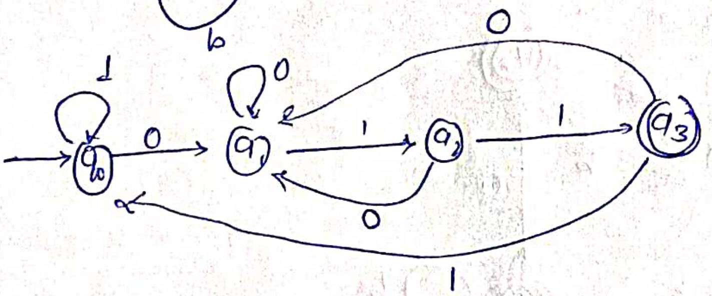


(3)



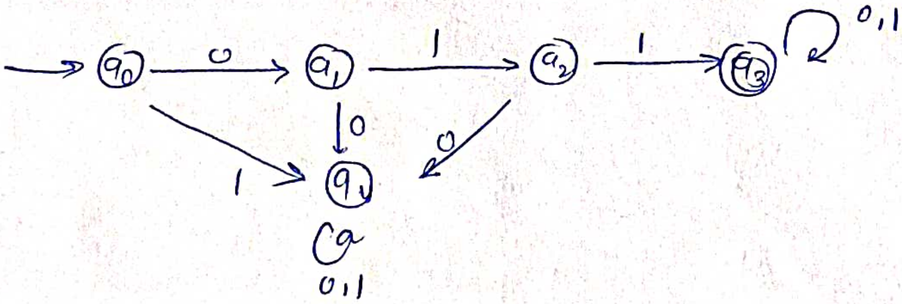
(a+b)* b (a+b)

(4)

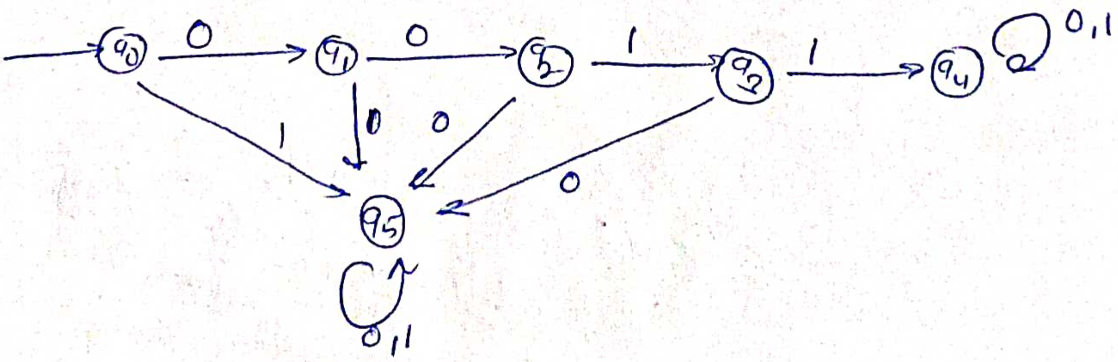


(011).

(5)

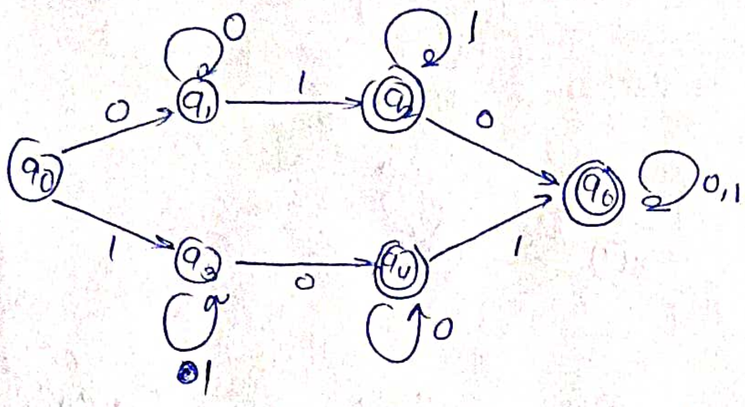


(6)

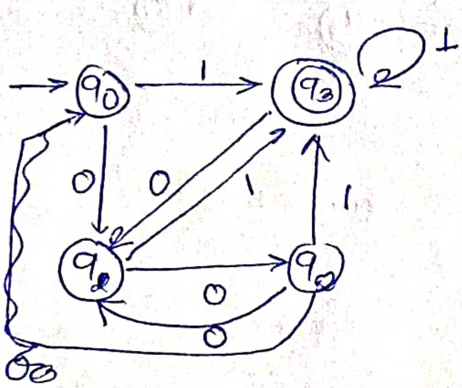


st

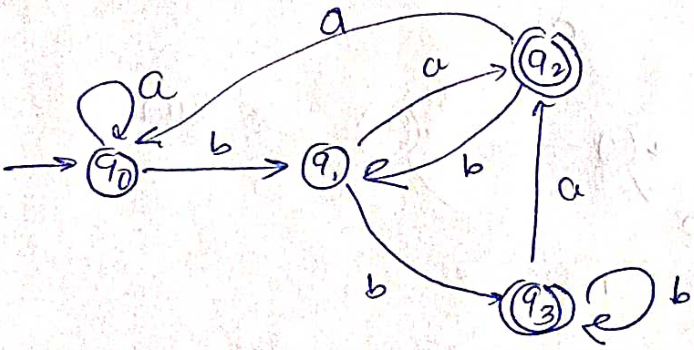
(7)



(8)



(9)

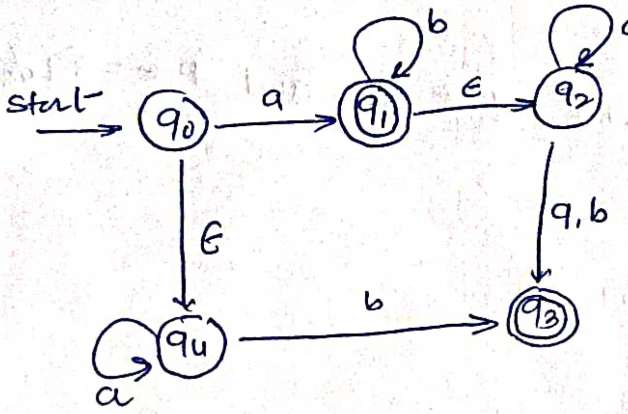


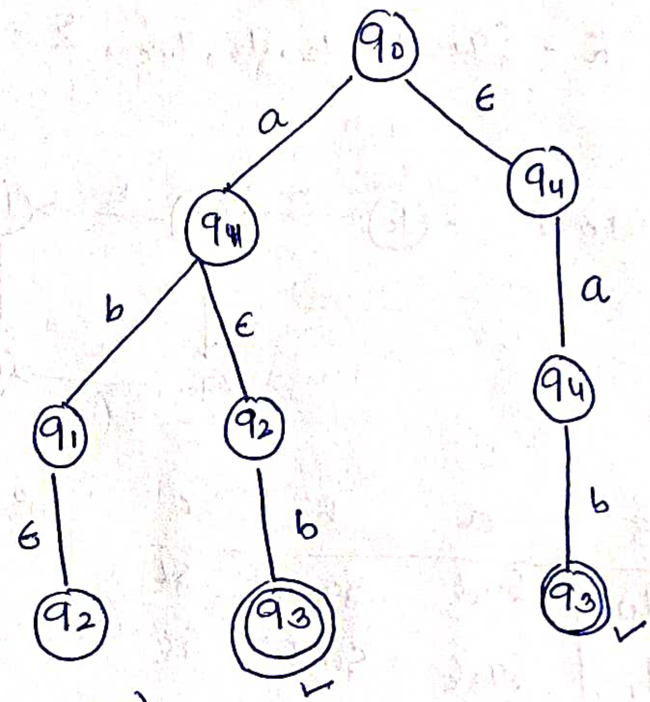
(10)

- The computation tree is defined as follows:

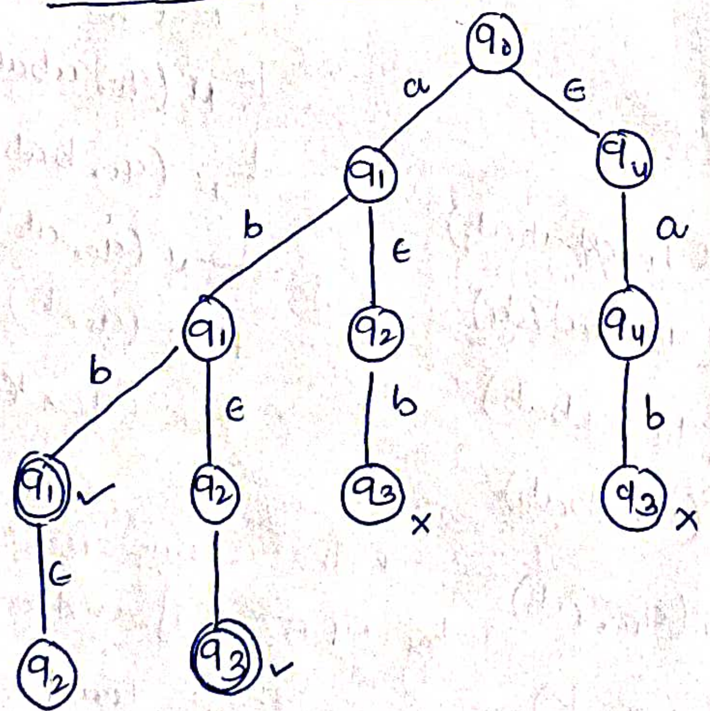
1. p is the root node.
2. children ~~from~~ of p are precisely those nodes which are having transitions from p via ϵ or a_1.

Computation tree of $\hat{\delta}(q_0, ab)$.





Computation tree of $\hat{\delta}(q_0, abb)$



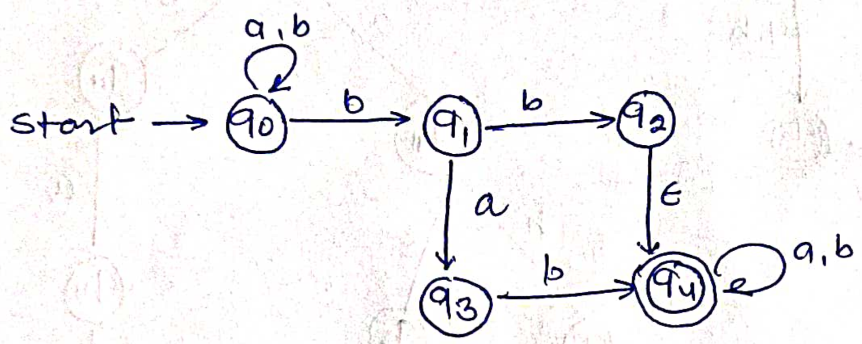
Configuration of NFA :-

→ A configuration of N is an element of $Q \times \Sigma^*$

→ The relation $\vdash_N$ between configurations is defined as follows:

$(q, w) \vdash_N (q', w')$ iff there is a $a \in \Sigma \cup \{\epsilon\}$ such that $w = aw'$ and $\delta(q, a) = q'$

Eg:- $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b\}$, $q_0 \in Q$, and $F \subseteq = \{q_4\}$.



δ	a	b	ϵ
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\{\}$
q_1	$\{q_3\}$	$\{q_2\}$	$\{\}$
q_2	$\{\}$	$\{\}$	$\{q_4\}$
q_3	$\{\}$	$\{q_4\}$	$\{\}$
q_4	$\{q_4\}$	$\{q_3, q_4\}$	$\{\}$

- When N is given the string bababab as input, several different sequence of move may be possible.

$(q_0, bababab) \vdash_N (q_0, ababab)$
$\vdash_N (q_0, babab)$
$\vdash_N (q_0, abab)$
$\vdash_N (q_0, bab)$
$\vdash_N (q_0, ab)$
$\vdash_N (q_0, b)$
$\vdash_N (q_0, \epsilon)$

$(q_0, bababab) \vdash_N (q_1, ababab)$
$\vdash_N (q_3, babab)$
$\vdash_N (q_4, abab)$
$\vdash_N (q_4, bab)$
$\vdash_N (q_4, ab)$
$\vdash_N (q_4, b)$
$\vdash_N (q_4, \epsilon)$.

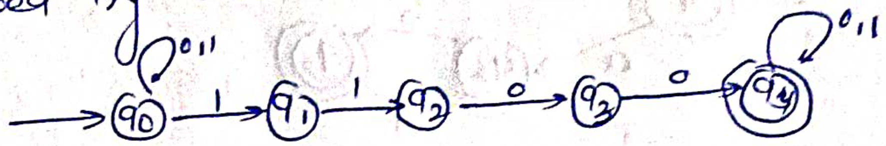
* The string is accepted by NFA N if and only if there is at least one sequence of moves leading to a final state,

∴ bababab $\in L(N)$.

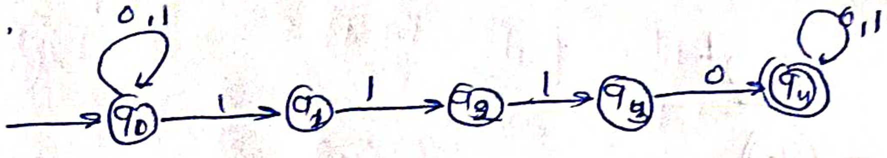
Problems on NFA

(P1) :- Design NFA with $\Sigma = \{0,1\}$ in which double '1' followed by double '0'.

Sol^n :-

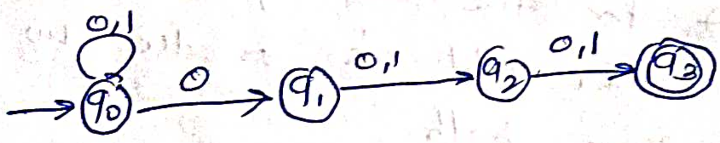


(P2) :- Design an NFA in which all the string contain substring 1110.

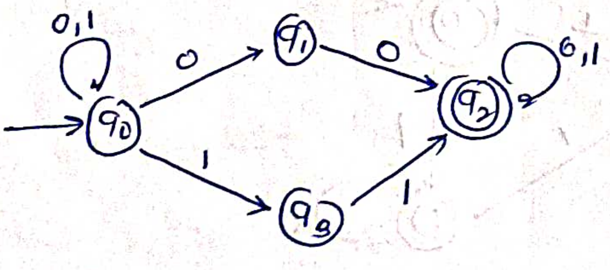


(P3) :- Design an NFA in which the third symbol from the right is always 0.

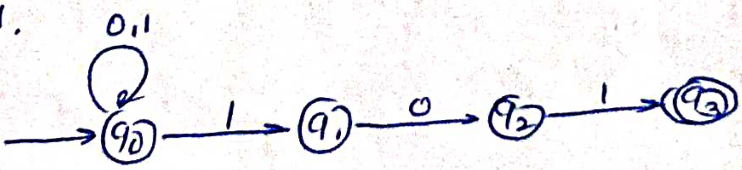
0/1 0 0/1 0/1



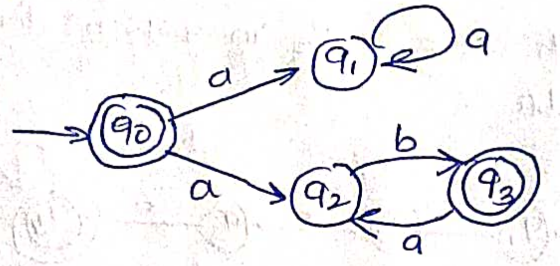
(P4) :- Design an NFA with $\Sigma = \{0,1\}$ in which every string contain either 00 or 11.



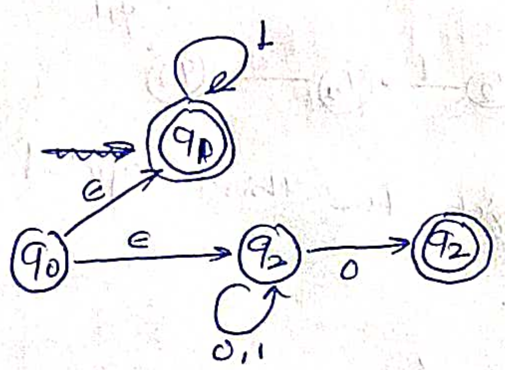
(P5) :- Design NFA to accept all binary string that ends with 101.



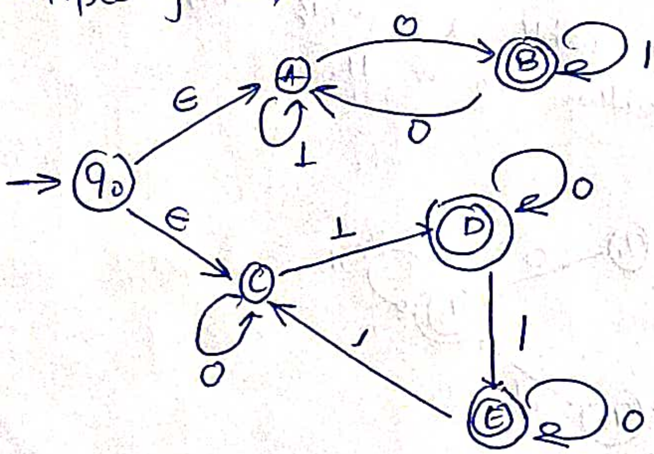
(P6) Design NFA for $a^* + (ab)^*$



(P7) NFA to accept all binary string that contains only 1's or ends with 0.



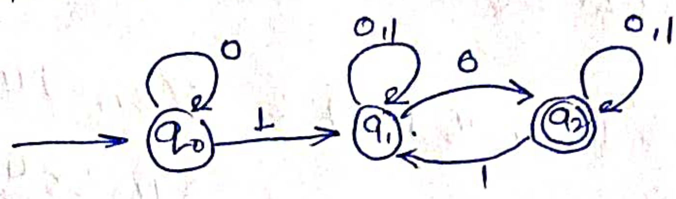
(P8) Design NFA for the set of all binary strings that have either odd 0's or the no of 1s is not a multiple of 3, or both.



:- NFA to DFA conversion :-

→ Let $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA which accepts the Language $L(M)$. There should be equivalent DFA denoted by $M' = (Q', \Sigma', \delta', q_0', F')$ such that $L(M) = L(M')$.

→ Steps of conversion :-



Step ①: Initially $Q' = \phi$

Step ②:- Add q_0 of NFA to Q'. Then find the transitions from the start state.

Step ③:- In Q', find the possible set of states for each input symbol. If this set of states is not in Q', then add it to Q'.

Step ④:- In DFA final state will be set of all states which contain F (final states of NFA).

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
q_2	q_2	$\{q_1, q_2\}$

∴ Now we will obtain δ' transition from state q_0.

$\delta'([q_0], 0) = [q_0]$
$\delta'([q_0], 1) = [q_1]$

The δ' transition for state q_1.

$\delta'([q_1], 0) = [q_1, q_2]$ (new state generated).
$\delta'([q_1], 1) = [q_1]$

The δ' transition for state q_2.

$\delta'([q_2], 0) = [q_2]$

$\delta'([q_2], 1) = [q_1, q_2]$

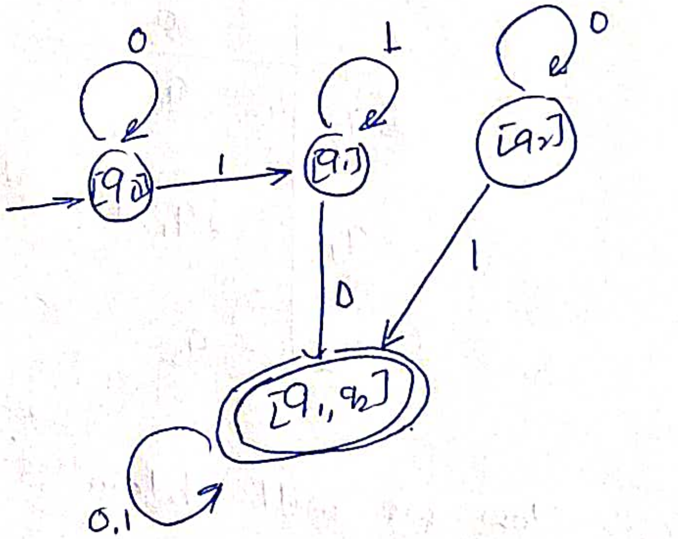
Now we will obtain δ' transition on $[q_1, q_2]$.

$$\delta'([q_1, q_2], 0) = \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \{q_1, q_2\} \cup \{q_2\}$$
$$= [q_1, q_2]$$

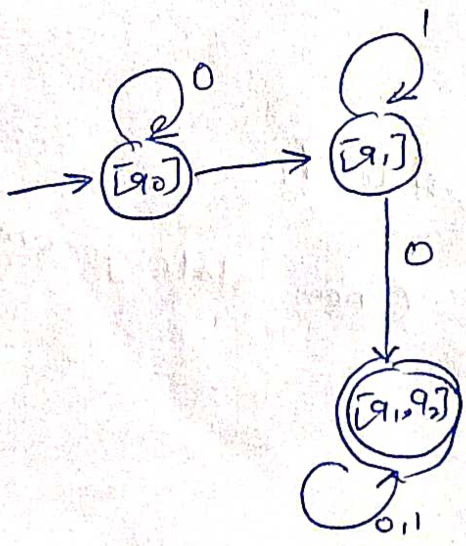
$$\delta'([q_1, q_2], 1) = \delta(q_1, 1) \cup \delta(q_2, 1)$$
$$= \{q_1\} \cup \{q_1, q_2\}$$
$$= \{q_1, q_2\}$$
$$= [q_1, q_2]$$

The state $[q_1, q_2]$ is the final state because it contains q_2 i.e. final state.

State	0	1
$\rightarrow [q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1, q_2]$	$[q_1]$
$[q_2]$	$[q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_1, q_2]$	$[q_1, q_2]$



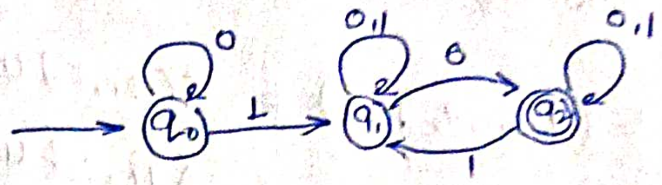
$\Downarrow$ $[q_2]$ is unreachable $\therefore$ Eliminated.



—: NFA to DFA conversion :—

→ Let $M = (Q, \Sigma, \delta, q_0, F)$ is an NFA which accepts the Language $L(M)$. There should be equivalent DFA denoted by $M' = (Q', \Sigma', \delta', q_0', F')$ show that $L(M) = L(M')$.

→ Steps of conversion :—.



Step①: Initially $Q' = \phi$

Step②:- Add q_0 of NFA to Q'. Then find the transitions from the start state.

Step③:- In Q', find the possible set of states for each input symbol. If this set of states is not in Q', then add it to Q'.

Step④:- In DFA final state will be set of all states which contain F (final states of NFA).

δ.	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$\{q_1, q_2\}$	q_1
q_2	q_2	$\{q_1, q_2\}$

∴ Now we will obtain δ' transition from state q_0.

$\delta'([q_0], 0) = [q_0]$
$\delta'([q_0], 1) = [q_1]$

The δ' transition for state q_1.

$\delta'([q_1], 0) = [q_1, q_2]$ (new state generated).
$\delta'([q_1], 1) = [q_1]$

The δ' transition for state q_2.

$$\delta'([q_2],0) = [q_2]$$
$$\delta'([q_2],1) = [q_1, q_2]$$

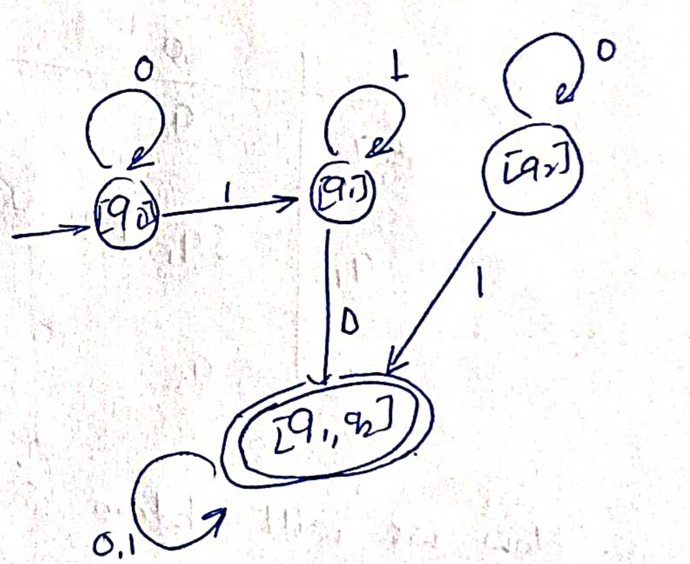
Now we will obtain δ' transition on $[q_1, q_2]$.

$$\delta'([q_1, q_2], 0) = \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \{q_1, q_2\} \cup \{q_2\}$$
$$= [q_1, q_2]$$

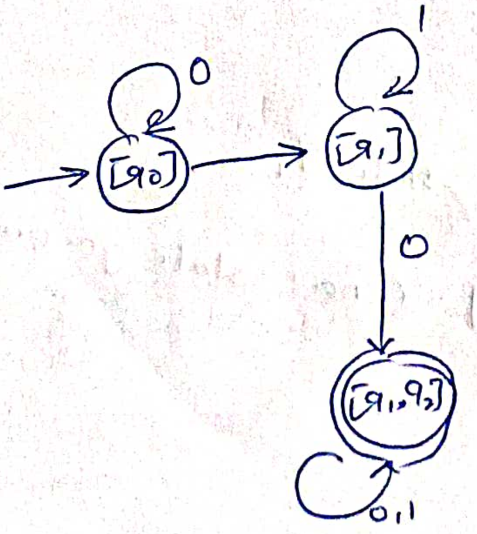
$$\delta'([q_1, q_2], 1) = \delta(q_1, 1) \cup \delta(q_1, 1)$$
$$= \{q_1\} \cup \{q_1, q_2\}$$
$$= \{q_1, q_2\}$$
$$= [q_1, q_2]$$

The state $[q_1, q_2]$ is the final state because it contains q_2 i.e. final state.

stat	0	1
$\rightarrow [q_0]$	$[q_0]$	$[q_1]$
$[q_1]$	$[q_1, q_2]$	$[q_1]$
$[q_2]$	$[q_2]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_1, q_2]$	$[q_1, q_2]$



$\Downarrow$ $[q_2]$ is unreachable $\therefore$ Eliminated.



Equivalance of NFA and DFA :-

→ Difination :- Two ~~Two~~ finite automata M_1 and M_2 (deterministic and non-deterministic) are equivalent if and only if

$$L(M_1) = L(M_2)$$

→ A DFA is just a special type of NFA.

→ Every NFA has an equivalent DFA.

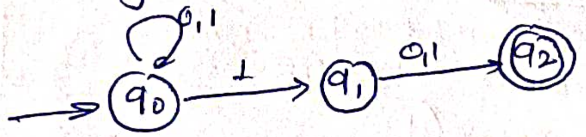
- They recognize the same language i.e., a NFA recognizing language $L(N)$ has an equivalent DFA recognizing $L(N)$.

→ The difference between a NFA and a DFA

- For NFA, the codomain of δ is 2^Q.
- For NFA, δ allow "ϵ-transition".

Conversion of NFA to DFA

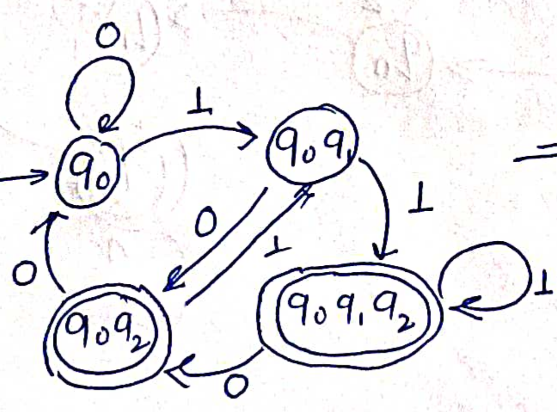
E.g.:- NFA of all binary strings in which 2nd last bit is 1.



δ	0	1
→ q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	—	—

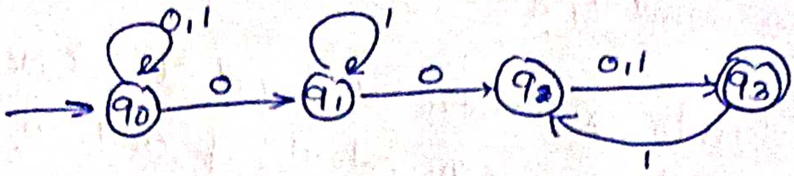
⇒

δ	0	1
→ q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_0q_1	q_0, q_2	q_0, q_1, q_2
q_0q_2	q_0	q_0q_1
$q_0q_1q_2$	q_0q_2	q_0q_1, q_2



DFA

Eg:-

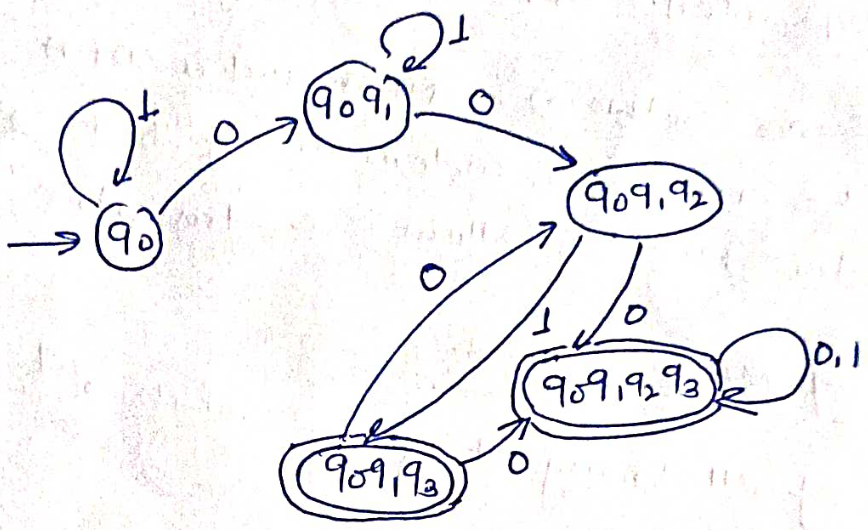


NFA.

δ	0	1
$\rightarrow q_0$	$q_0 q_1$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

$\Rightarrow$

δ	0	1
$\rightarrow q_0$	$q_0 q_1$	q_0
$q_0 q_1$	$q_0 q_1 q_2$	$q_0 q_1$
$q_0 q_1 q_2$	$q_0 q_1 q_2 q_3$	$q_0 q_1 q_3$
$q_0 q_1 q_2 q_3$	$q_0 q_1 q_2 q_3$	$q_0 q_1 q_2 q_3$
$q_0 q_1 q_3$	$q_0 q_1 q_2 q_3$	$q_0 q_1 q_2$

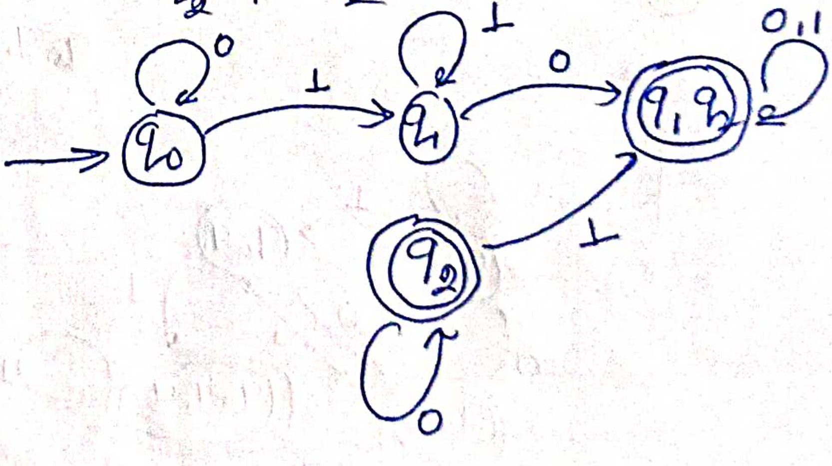


Eg:- Convert the given NFA to DFA.

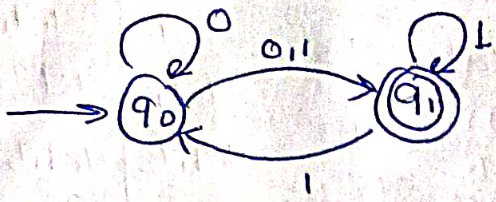
δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1, q_2	q_1
q_2	q_2	$q_1 q_2$

$\Rightarrow$

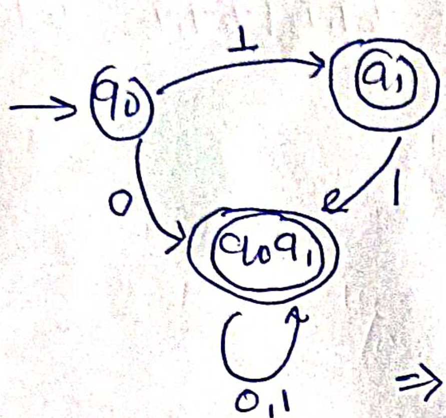
δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	$q_1 q_2$	q_1
$q_1 q_2$	$q_1 q_2$	q_1, q_2
q_2	q_2	$q_1 q_2$



Eg:- Convert the following NFA to DFA.

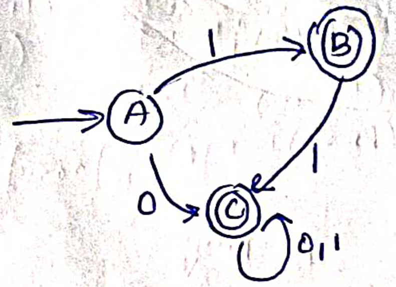


δ	0	1
→q_0	q_0q_1	q_1
q_1	ϕ	q_0q_1

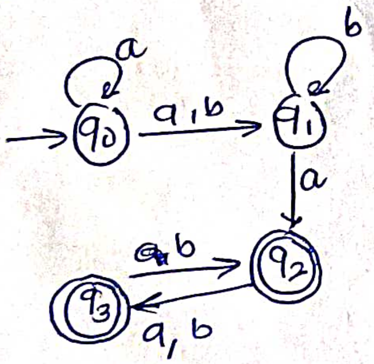


δ	0	1
→q_0	q_0q_1	q_1
q_1	ϕ	q_0q_1
q_0q_1	q_0q_1	q_0q_1

⇒ Let $A = [q_0]$, $B = [q_1]$, $C = [q_0q_1]$



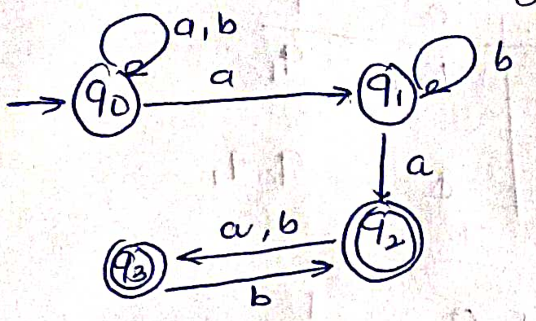
Eg:- Convert the following NFA to DFA.



δ	a	b
→q_0	q_0q_1	q_1
q_1	q_2	q_1
q_2	q_3	q_3
q_3	-	q_2

δ	a	b
→q_0	q_0q_1	q_1
q_1	q_2	q_1
q_0q_1	$q_0q_1q_2$	q_1
$q_0q_1q_2$	$q_0q_1q_2q_3$	q_1q_3
$q_0q_1q_2q_3$	$q_0q_1q_2q_3$	$q_1q_2q_3$
q_1q_3	q_2	q_1q_2
$q_1q_2q_3$	q_2q_3	$q_1q_2q_3$
q_2	q_3	q_3
q_1q_2	q_2q_3	q_1q_3
q_3	-	q_2
q_2q_3	q_3	q_2q_3

Eg:- Convert the following NFA to DFA.



δ	a	b
$\rightarrow q_0$	q_0q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

δ	a	b
$\rightarrow q_0$	q_0q_1	q_0
q_0q_1	$q_0q_1q_2$	q_0q_1
$q_0q_1q_2$	$q_0q_1q_2q_3$	$q_0q_1q_3$
$q_0q_1q_2q_3$	$q_0q_1q_2q_3$	$q_0q_1q_2q_3$
$q_0q_1q_3$	$q_0q_1q_2$	$q_0q_1q_2$