Unit - I

Finite Automata & Regular Expression

* Define following terms :

1. String :- A string is finite sequence of character or alphabet placed one after other. The length of the string is the no. of character in the string. A string is usually denoted be s, w or z and length of string is denoted be |s|, |w| or |z|. There are special type of string, known as 'Empty String'. It is denoted be 'ε' (epsilon) |ε|=0

example → s = 110110 s = abba

$\boxed{|s| = 6}$ $\boxed{|s| = 4}$

2. Prefix of String :- A string obtained by deleting leading symbol of a string is known as prefix of a string.

3. Sufix of string :- A string obtained by deleting trailing symbol of a string is known as sufix of a string.

4. Substring :- A string obtained by deleting sufix and prefix of a string is known as substring.

example → Consider, a string w = abbaba. find prefix, sufix and substring of a string.

→ let,

w = abbaba

$\boxed{|w| = 6}$

W = ε abbaba ε

prefix :- abbaba, bbaba, baba, aba, ba, a, ϵ

sufix :- abbaba, abbab, abba, abb, ab, a, ϵ

substring :- abbaba, bbaba, ba, a, abbab, baba, abba, aba, abb, ab, ϵ

* W = 1101101

→ let,

W = 1101101

$|w| = 7$

∴ W = ϵ 1101101 ϵ

prefix :- 1101101, 101101, 01101, 1101, 101, 01, 0, ϵ

sufix :- 1101101, 110110, 11011, 1101, 110, 11, 0, ϵ

Substring :- 1101101, 101101, 110110, 01101, 11011, 1101, 101, 110, 01, 11, 0, ϵ

* Deterministic Finite Automata (DFA) *

Que. Explain DFA with suitable example.

→ A finite automata consist of finite set of states & set of transition from one state to other state. That occur an input symbol choosen from an alphabet 'Σ' (input symbol) for each input symbol there is one transition from each state. q_0 is initial state from which automata start some states are design as finite state.

A DFA is define by five(5) tuple -

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

where,

Q = finite set of internal states.

Σ = finite set of input alphabet.

q_0 = initial state.

F = finite set of final state.

δ = Mapping Function.

$$\boxed{\therefore \delta = Q \times \Sigma \rightarrow Q}$$

In DFA design there are two important points -

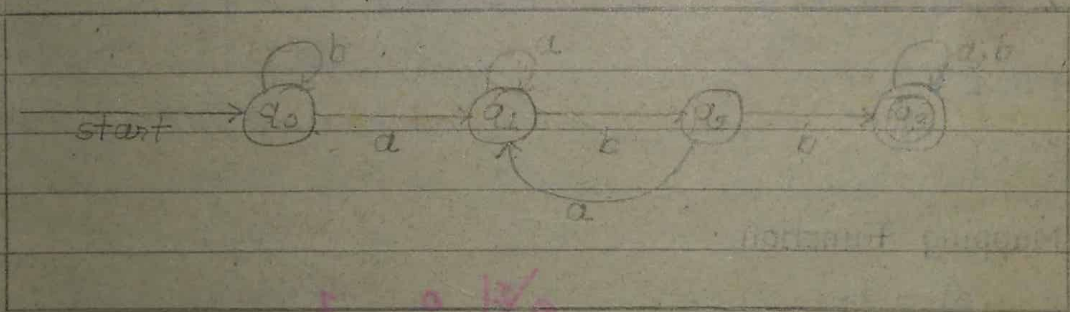
(i) Each state must contain all input symbol path.

(ii) No symbol can have more than one path for same input symbol.

Example (1) Design a DFA which accept substring abb over $\Sigma = \{a, b\}$

$\longrightarrow$ let,

$s = abb$ $\quad \Sigma = \{a, b\}$



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

$Q = \{q_0, q_1, q_2, q_3\}$ $\qquad F = \{q_3\}$

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

$q_0 = \{q_0\}$

δ = Mapping function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_1$

$\delta(q_2, b) = q_2$

$\delta(q_3, a) = q_3$

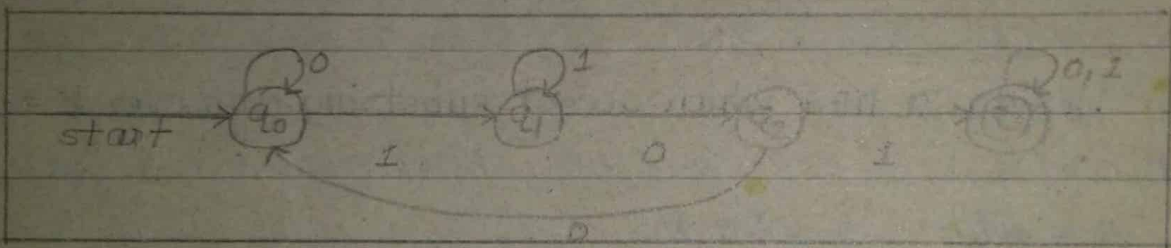
$\delta(q_3, b) = q_3$

$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_2
q_3	q_3	q_3

② $S = 101$, $\Sigma = (0, 1)$

→ let,

$S = 101$ & $\Sigma(0,1)$



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

$F = \{q_3\}$

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

$q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_1$

$\delta(q_2, 0) = q_0$

$\delta(q_2, 1) = q_3$

$\delta(q_3, 0) = q_3$

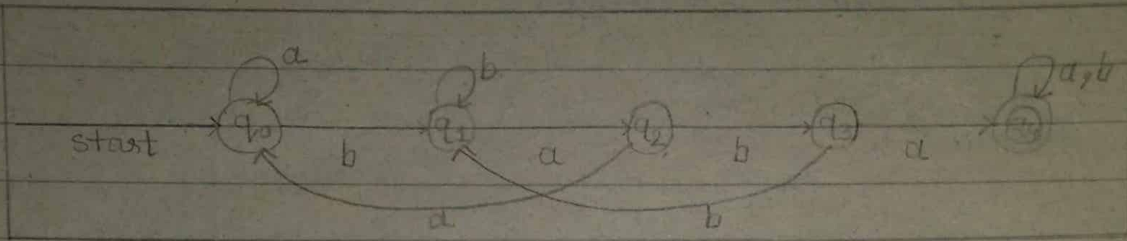
$\delta(q_3, 1) = q_3$

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3	q_3	q_3

③ $s = baba$, $\Sigma(a,b)$

⟶ let,

$s = baba$ & $\Sigma(a,b)$



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

$F = \{q_4\}$

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

$q_0 = \{q_0\}$

δ = Mapping function

$\delta(q_0, a) = q_0$
$\delta(q_0, b) = q_1$
$\delta(q_1, a) = q_2$
$\delta(q_1, b) = q_1$
$\delta(q_2, a) = q_0$
$\delta(q_2, b) = q_3$
$\delta(q_3, a) = q_4$
$\delta(q_3, b) = q_1$
$\delta(q_4, a) = q_4$
$\delta(q_4, b) = q_4$

$Q \backslash \Sigma$	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3	q_4	q_1
q_4	q_4	q_4

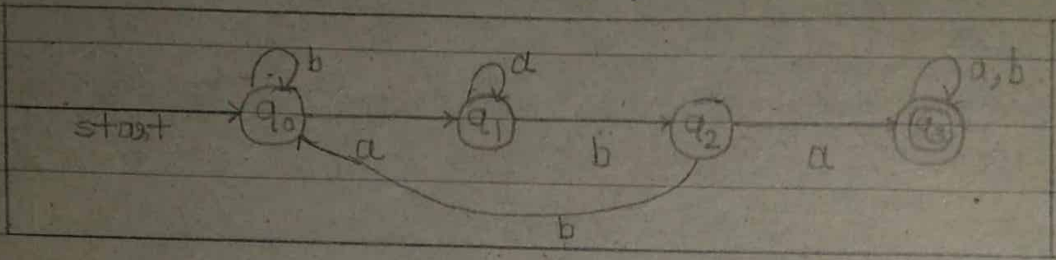
9/8/13 8

Que. Design a DFA which accept substring :

① aba

⟶ let,

$s = aba$ & $\Sigma = \{a, b\}$



$Q = \{q_0, q_1, q_2, q_3\}$ $\Sigma = \{a, b\}$

$q_0 = \{q_0\}$

$F = \{q_3\}$

δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_3$

$\delta(q_2, b) = q_0$

$\delta(q_3, a) = q_3$

$\delta(q_3, b) = q_3$

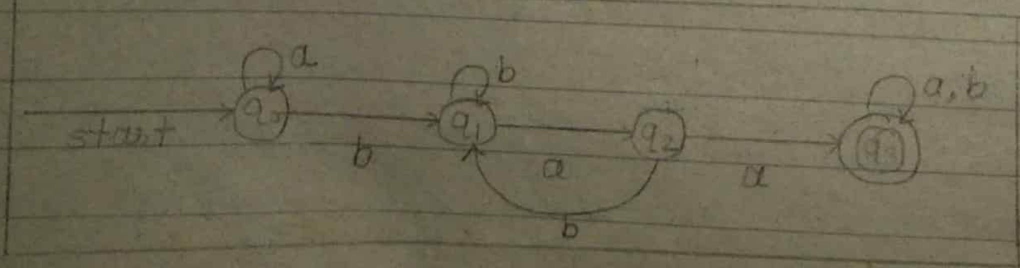
Transition Table

Q \ Σ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
q_3	q_3	q_3

② baa

⟶ let,

$s = baa$ & $\Sigma = \{a, b\}$



$Q = \{q_0, q_1, q_2, q_3\}$ $\quad\quad$ $\Sigma = \{a, b\}$

$q_0 = \{q_0\}$

$F = \{q_3\}$

δ = Mapping Function

$\delta(q_0, a) = q_0$

$\delta(q_0, b) = q_1$

$\delta(q_1, a) = q_2$

$\delta(q_1, b) = q_1$

$\delta(q_2, a) = q_3$

$\delta(q_2, b) = q_1$

$\delta(q_3, a) = q_3$

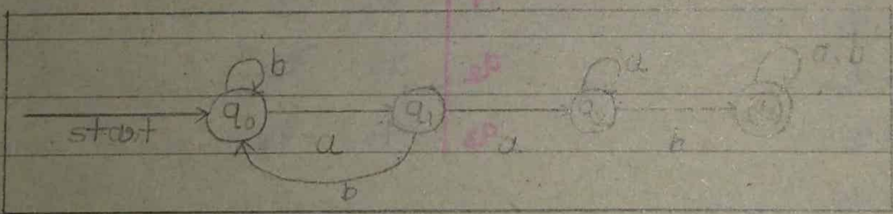
$\delta(q_3, b) = q_3$

Q \ Σ	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_3	q_1
q_3	q_3	q_3

③ aab

→ let,

$s = aab$ & $\Sigma = \{a, b\}$



$Q = \{q_0, q_1, q_2, q_3\}$ $\quad\quad$ $F = \{q_3\}$

$\Sigma = \{a, b\}$ $\quad\quad$ $q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_2$

$\delta(q_1, b) = q_0$

$\delta(q_2, a) = q_2$

$\delta(q_2, b) = q_3$

$\delta(q_3, a) = q_3$

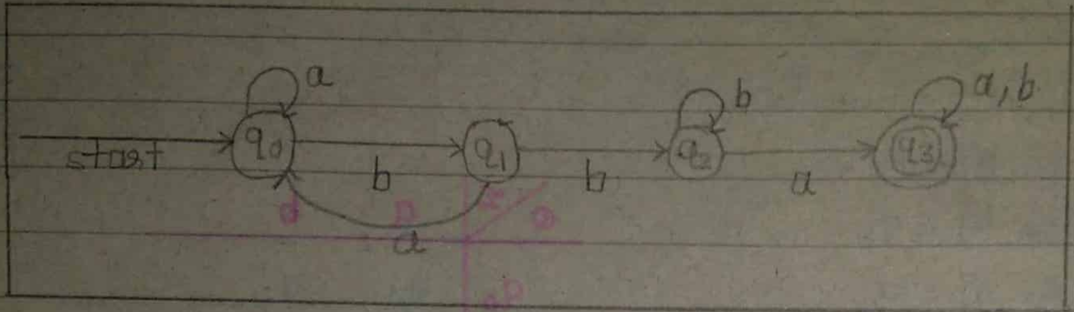
$\delta(q_3, b) = q_3$

Q \ Σ	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
q_3	q_3	q_3

④ bba

→ let,

$s = bba$ & $\Sigma = \{a, b\}$



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

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

$q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, a) = q_0$

$\delta(q_0, b) = q_1$

$\delta(q_1, a) = q_0$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_3$

$\delta(q_2, b) = q_2$

$\delta(q_3, a) = q_3$

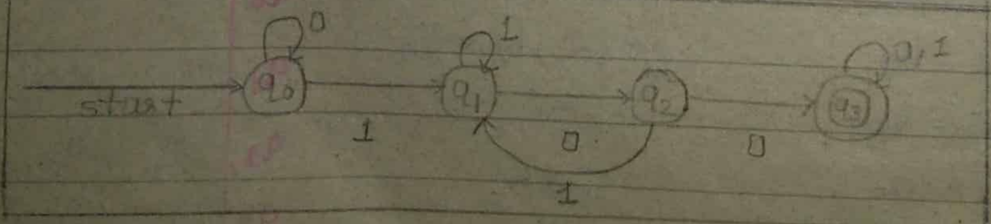
$\delta(q_3, b) = q_3$

Q \ Σ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_3	q_2
q_3	q_3	q_3

⑤ 100

→ let,

$s = 100$ & $\Sigma = \{0, 1\}$



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

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

$q_0 = \{q_0\}$

δ = Mapping function

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_1$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_1$

$\delta(q_3, 0) = q_3$

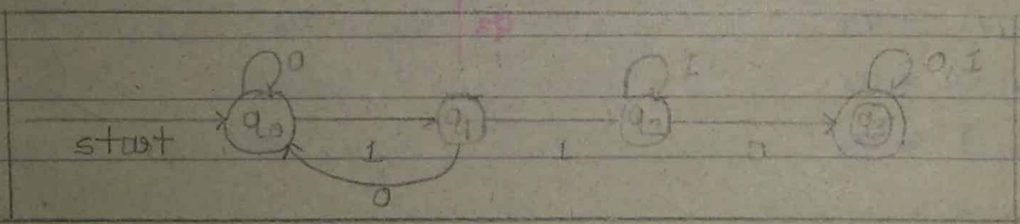
$\delta(q_3, 1) = q_3$

$Q \backslash \Sigma$	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_3	q_1
q_3	q_3	q_3

⑥ 110

→ let,

$s = 110$ & $\Sigma = \{0, 1\}$



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

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

$q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_2$

$\delta(q_3, 0) = q_3$

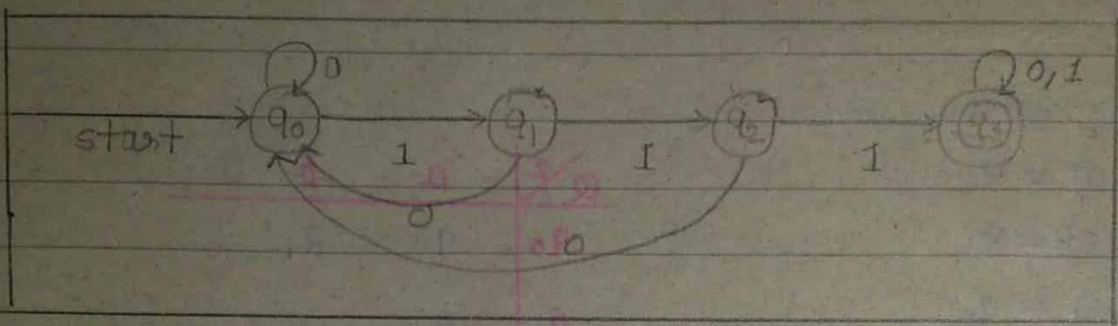
$\delta(q_3, 1) = q_3$

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_3	q_2
q_3	q_3	q_3

⑦ 111

→ let,

$s = 111$ & $\Sigma = (0,1)$



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

$\Sigma = \{0, 1\}$ $q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_0$

$\delta(q_2, 1) = q_3$

$\delta(q_3, 0) = q_3$

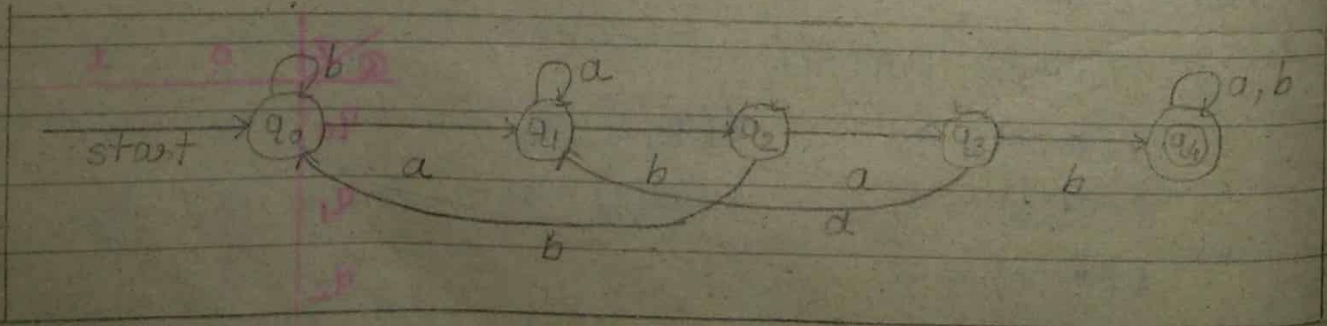
$\delta(q_3, 1) = q_3$

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q_3

⑧ abab

→ let,

$s = abab$ & $\Sigma = \{a, b\}$



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

$\Sigma = \{a, b\}$ $q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_3$

$\delta(q_2, b) = q_0$

$\delta(q_3, a) = q_1$

$\delta(q_3, b) = q_4$

$\delta(q_4, a) = q_4$

$\delta(q_4, b) = q_4$

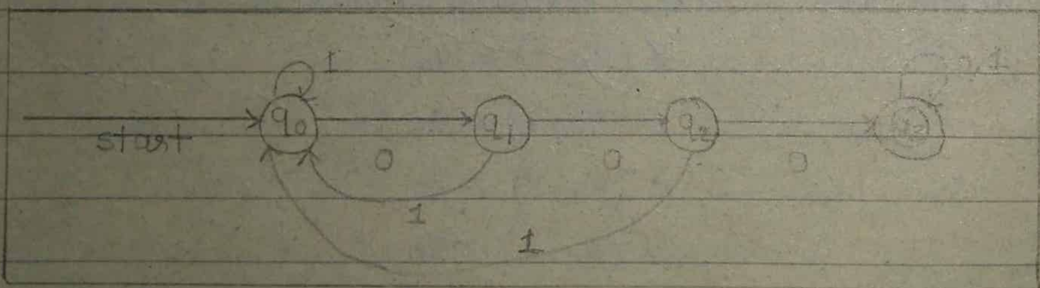
$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
q_3	q_1	q_4
q_4	q_4	q_4

S-09
W-08
S-04

Que. Design a DFA which contain three consecutive 0's over $\Sigma(0,1)$

→ let,

$S = 000$ & $\Sigma = (0,1)$



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

$\Sigma = \{0, 1\}$ $q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_0$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_0$

$\delta(q_3, 0) = q_3$

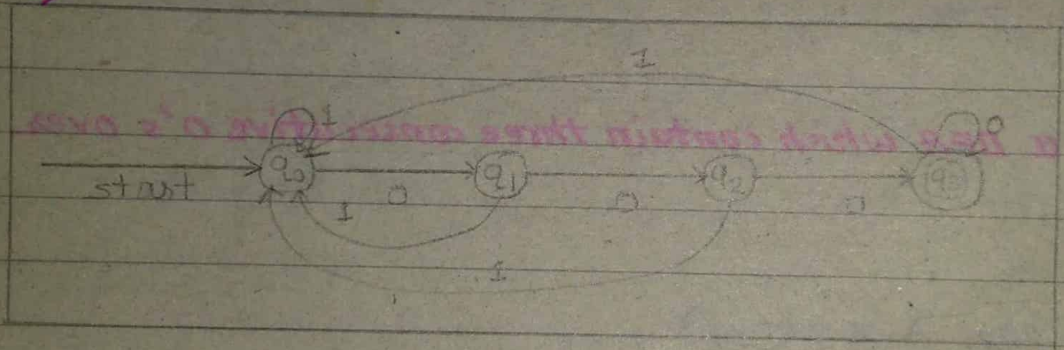
$\delta(q_3, 1) = q_3$

Q \ Σ	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_3

S-04

*Que. Design a DFA which end with three consecutive 0's over $\Sigma = (0, 1)$

→



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

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

$F = \{q_3\}$

$q_0 = \{q_0\}$

Q \ Σ	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_0

δ = Mapping Function

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_0$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_0$

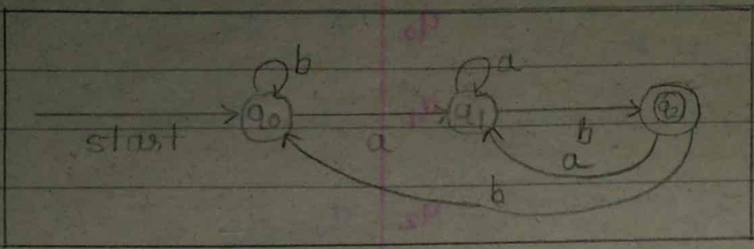
$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_0$

S-11

*** Que. Design a DFA which end with ab over $\Sigma = \{a, b\}$**

→



$Q = \{q_0, q_1, q_2\}$ $\qquad F = \{q_2\}$

$\Sigma = \{a, b\}$ $\qquad q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_1$

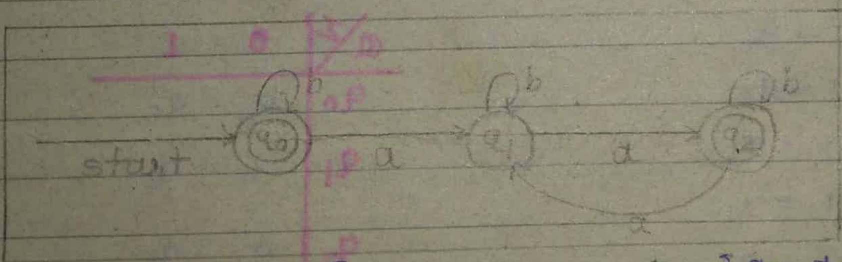
$\delta(q_2, b) = q_0$

$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

0, 2, 4, 6, 8, 10, 12, 16 ...

Que. Design a DFA which contain even no. of a's over $\Sigma = \{a, b\}$

→



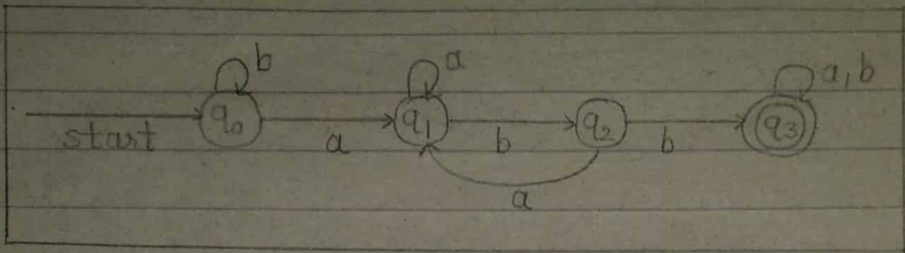
$Q = \{q_0, q_1, q_2\}$ $\qquad F = \{q_0, q_2\}$

$\Sigma = \{a, b\}$ $\qquad q_0 = \{q_0\}$

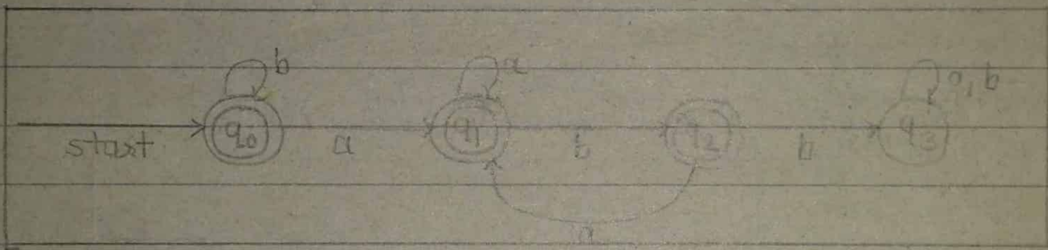
* Complement of DFA *

Que. Design a DFA which does not content substsing abb over $\Sigma = \{a, b\}$

⟶ Step 1 : Design a DFA which contain substsing abb over $\Sigma = \{a, b\}$



step 2 : To obtained its complement convert all final state to non-final state and all non-final state to final state i.e. interchanging all final and non-final state.



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

$\Sigma = \{a, b\}$.

$q_0 = \{q_0\}$

$F = \{q_0, q_1, q_2\}$

δ (Mapping Function)

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

$\delta(q_2, a) = q_1$

$\delta(q_2, b) = q_3$

$\delta(q_3, a) = q_3$

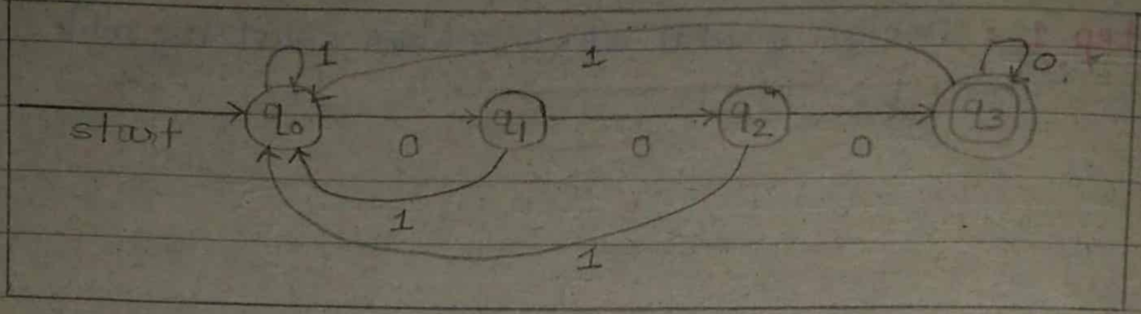
$\delta(q_3, b) = q_3$

$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_3	q_3

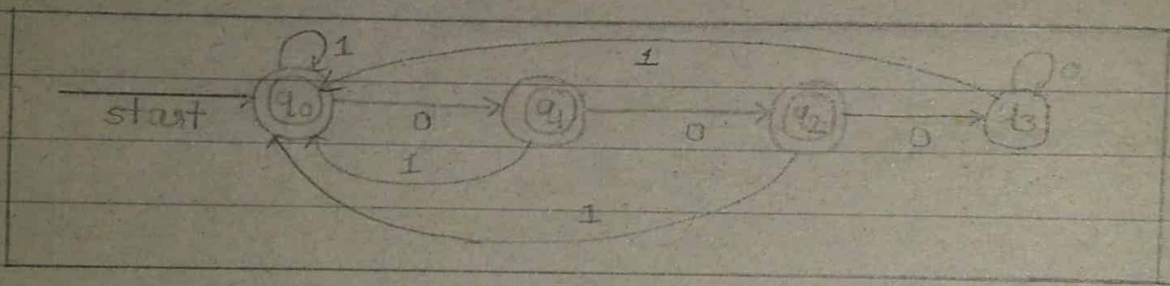
W-09

Que. Design a DFA which does not content three consecutive 0's over $\Sigma = \{0, 1\}$

→ **step 1:** Design a DFA which content substring 000 over Σ



step 2: To obtained its complement convert all final state to non-final state and all non-final state to final state i.e. interchanging all final and non-final state.



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

$F = \{q_0, q_1, q_2\}$

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

δ = Mapping Function

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_0$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_0$

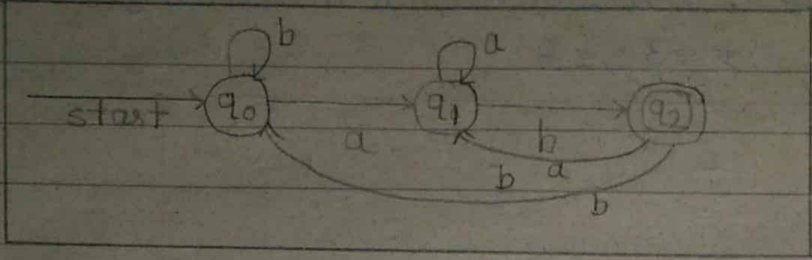
$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_0$

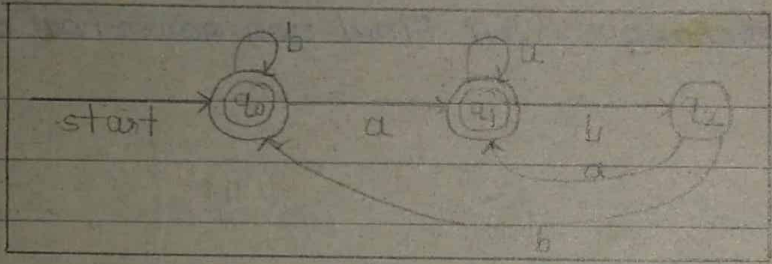
$Q \backslash \Sigma$	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_0

Que. Design a DFA which does not end with ab over $\Sigma = \{a, b\}$

→ step 1: Design a DFA which content substring ab over $\Sigma \{a, b\}$



step 2: To obtained its complement convert all final state to non-final state and all non-final state to final state i.e. interchanging all final and non-final state.



$Q = \{q_0, q_1, q_2\}$ $\quad$ $F = \{q_0, q_1\}$

$\Sigma = \{a, b\}$ $\quad$ $q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_1$

$\delta(q_1, b) = q_2$

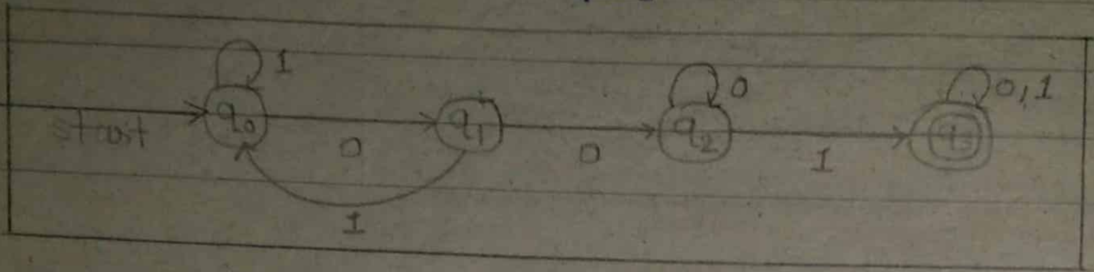
$\delta(q_2, a) = q_1$

$\delta(q_2, b) = q_0$

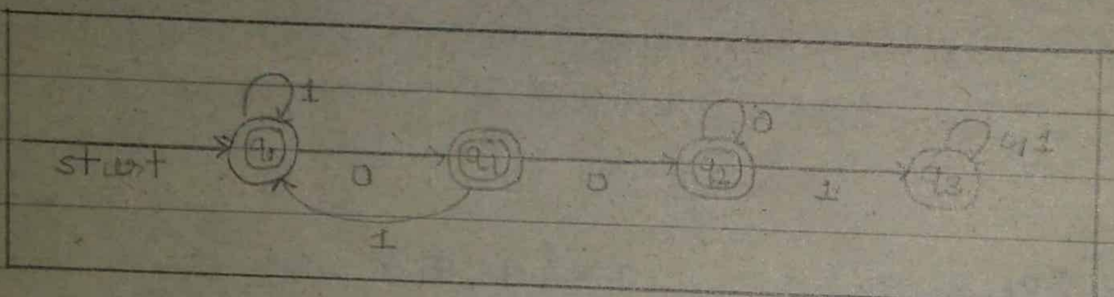
$Q \backslash \Sigma$	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

Que. Design a DFA which does not contain substring 001 over $\Sigma = \{0,1\}$

→ step 1: Design a DFA which content substring 001 over $\Sigma = \{0,1\}$



step 2: To obtained its complement convert all final state to non-final state and all non-final state to final state i.e. interchanging all final and non-final state.



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

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

δ = Mapping Function

$F = \{q_0, q_1, q_2\}$

$q_0 = \{q_0\}$

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_2$

$\delta(q_1, 1) = q_0$

$\delta(q_2, 0) = q_2$

$\delta(q_2, 1) = q_3$

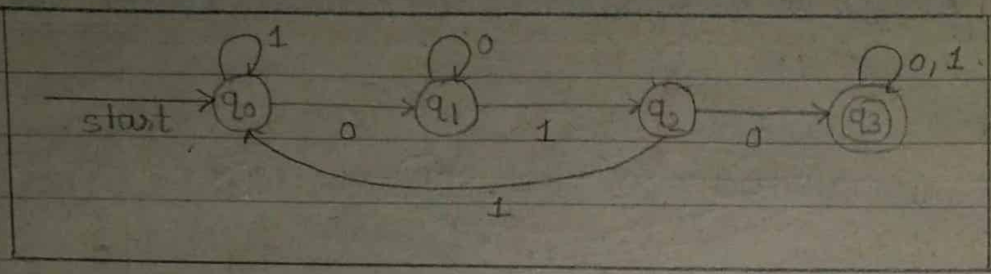
$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_3$

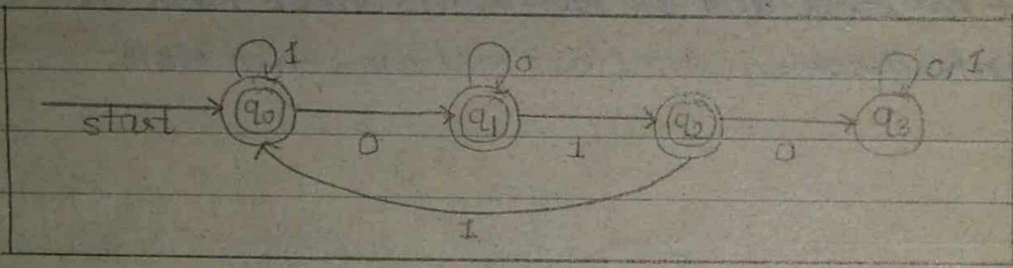
Q \ Σ	0	1
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
q_3	q_3	q_3

② 010 over $\Sigma = \{0,1\}$

→ step 1 :- Design a DFA which content substring 010 over $\Sigma = \{0,1\}$



step 2 :- To obtained its complement convert all final state to non-final state and all non-final state to final state i.e. interchanging all final and non-final state.



$Q = \{q_0, q_1, q_2, q_3\}$ $\quad F = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$ $\quad q_0 = \{q_0\}$

δ = Mapping Function

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_0$

$\delta(q_1, 0) = q_1$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_3$

$\delta(q_2, 1) = q_0$

$\delta(q_3, 0) = q_3$

$\delta(q_3, 1) = q_3$

Q \ Σ	0	1
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
q_3	q_3	q_3

W-08 S-08 S-06 S-09

** Non-Deterministic Finite Automata (NFA) *

Que. Define NFA with suitable example.

→ If a finite automata model is modified to allow 0, 1 or more transition for a input symbol on a particular state then that modified model is known as NFA.

A NFA is defined by 5 tuple.

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

where,

Q = finite set of internal state

Σ = finite set of input alphabet

q_0 = initial state

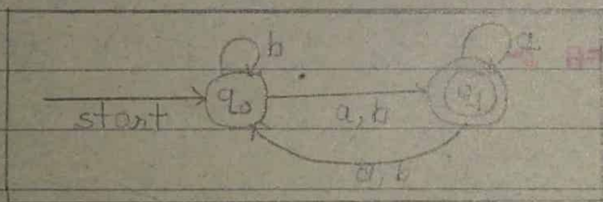
F = finite set of final state

δ = Mapping Function

$\boxed{\delta = Q \times \epsilon \rightarrow 2^Q}$

where Q is a power set of Q.

Example ①



δ = Mapping Function

$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0 q_1$

$\delta(q_1, a) = q_1 q_0$

$\delta(q_1, b) = q_0$

$Q \backslash \Sigma$	a	b
q_0	q_1	$q_0 q_1$
q_1	$q_0 q_1$	q_0

The only difference bet^n DFA & NFA is in Mapping function i.e. δ

* Conversion of NFA to DFA *

*S-04

**Que. Construct a deterministic finite automata equivalent to NFA.

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

where

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

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

$\delta(q_1, 0) = \phi$

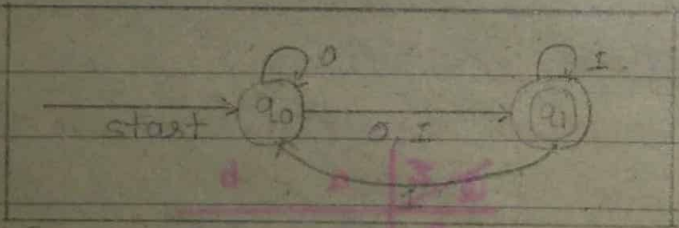
$\delta(q_1, 1) = \{q_0 q_1\}$

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

* Transition Table for NFA :-

Q \ Σ	0	1
q_0	$q_0 q_1$	q_1
q_1	ϕ	$q_0 q_1$

* Diagram for NFA :-



δ(M.F.)

$\delta(q_0, 0) = q_0 q_1$ New state

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = \phi$

$\delta(q_1, 1) = q_0 q_1$ New state

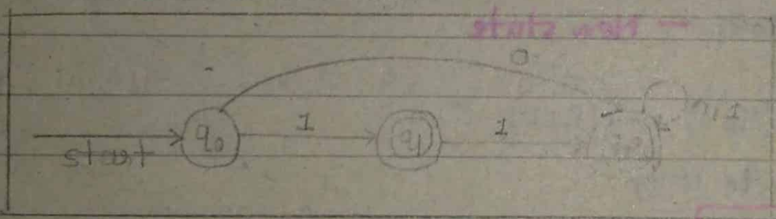
$$\delta'((q_0 q_1) 0) = \delta(q_0 0) \cup \delta(q_1 0)$$
$$= q_0 q_1 \cup \phi$$
$$\boxed{\delta'((q_0 q_1) 0) = q_0 q_1}$$

$$\delta'((q_0 q_1) 1) = \delta(q_0 1) \cup (q_1, 1)$$
$$= q_1 \cup q_0 q_1$$
$$\boxed{\delta'((q_0 q_1) 1) = q_0 q_1}$$

* Transition table for DFA :-

$Q \backslash \Sigma$	0	1
q_0	$q_0 q_1$	q_1
q_1	ϕ	$q_0 q_1$
$q_0 q_1$	$q_0 q_1$	$q_0 q_1$

* Diagram for DFA :-



W-06 **Que.** Construct a deterministic automata equivalent to NFA
$$M = (\{q_0 q_1\}, \{0, 1\}, \delta, q_0, q_0\})$$
where δ is

$Q \backslash \Sigma$	0	1
q_0	q_0	q_1
q_1	q_1	$q_0 q_1$

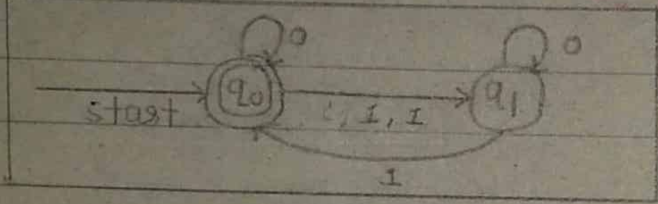
$\longrightarrow \quad M = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_0\})$

* **T. T. for NFA :-**

Q \ Σ	0	1
q_0	q_0	q_1
q_1	q_1	$q_0 q_1$

$(q_0, 0) = q_0$
$(q_0, 1) = q_1$
$(q_1, 0) = q_1$
$(q_1, 1) = q_0 q_1$

* **Diagram for NFA :-**



δ (M.F.)
$\delta(q_0, 0) = q_0$
$\delta(q_0, 1) = q_1$
$\delta(q_1, 0) = q_1$
$\delta(q_1, 1) = q_0 q_1$ — New state

$\delta'((q_0 q_1) 0) = \delta(q_0\, 0) \cup \delta(q_1, 0)$
$= q_0 \cup q_1$

$\boxed{\delta'((q_0 q_1) 0) = q_0 q_1}$

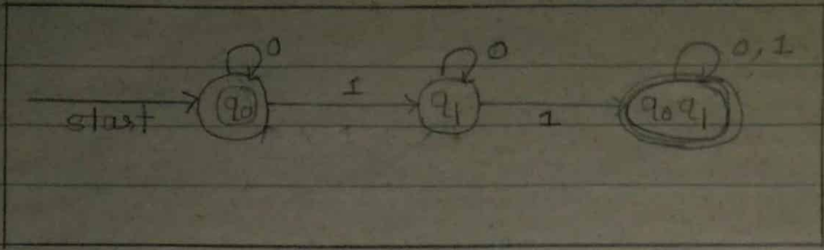
$\delta'((q_0 q_1) 1) = \delta(q_0\, 1) \cup \delta(q_1\, 1)$
$= q_1 \cup q_0 q_1$

$\boxed{\delta'((q_0 q_1) 1) = q_0 q_1}$

* **T. T. for DFA :-**

Q \ Σ	0	1
q_0	q_0	q_1
q_1	q_1	$q_0 q_1$
$q_0 q_1$	$q_0 q_1$	$q_0 q_1$

* Diagram for DFA :-



Que. Find Deterministic acceptor equivalent W-06

$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

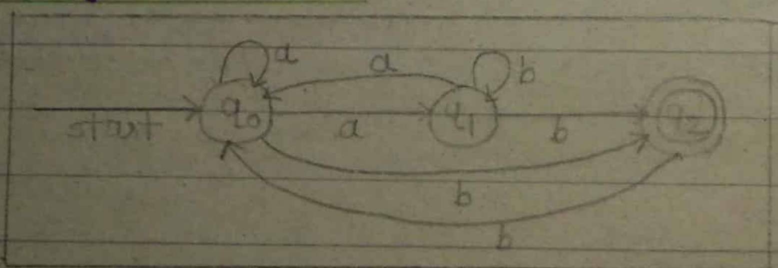
where T.T. is

Q \ Σ	a	b
q_0	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2	–	$q_0 q_1$

→ $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

Q \ Σ	a	b
q_0	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2	—	$q_0 q_1$

Diagram for NFA :-



M.F. (δ)

$\delta(q_0, a) = q_0 q_1$ New state

$\delta(q_0, b) = q_2$

$\delta(q_1, a) = q_0$

$\delta(q_1, b) = q_1$

$\delta(q_2, a) = \phi$

$\delta(q_2, b) = q_0 q_1$ New state

$\delta'((q_0 q_1) a) = \delta(q_0 a) \cup \delta(q_1 a)$

$= q_0 q_1 \cup q_0$

$\boxed{\delta'((q_0 q_1) a) = q_0 q_1}$

$\delta'((q_0 q_1) b) = \delta(q_0 b) \cup \delta(q_1 b)$

$= q_2 \cup q_1$

$\boxed{\delta'((q_0 q_1) b) = q_1 q_2}$ New state

$\delta'((q_1 q_2) a) = \delta(q_1 a) \cup \delta(q_2 a)$

$= q_0 \cup \phi$

$\boxed{\delta'((q_1 q_2) a) = q_0}$

$\delta'((q_1 q_2) b) = \delta(q_1 b) \cup \delta(q_2 b)$

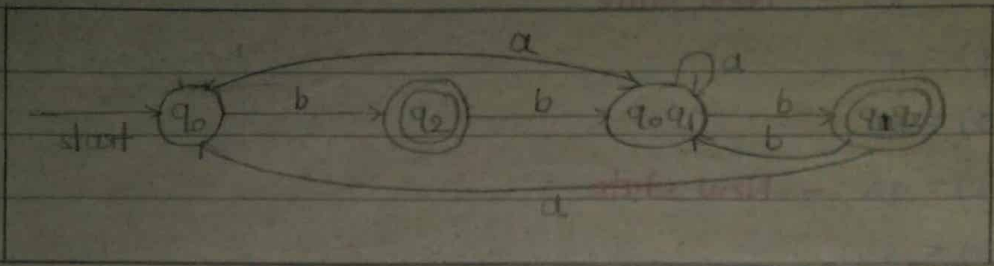
$= q_1 \cup q_0 q_1$

$\boxed{\delta'((q_1 q_2) b) = q_0 q_1}$

Transition Table for DFA :->

Q \ Σ	a	b
q_0	$q_0 q_1$	q_2
q_2	–	$q_0 q_1$
$q_0 q_1$	$q_0 q_1$	$q_1 q_2$
$q_1 q_2$	q_0	$q_0 q_1$

Diagram for DFA :→



S-05 ** Que. Construct DFA equivalent to NFA

$$M = (\{p, q, r, s\}, \{0,1\}, \delta, p, \{q, s\})$$

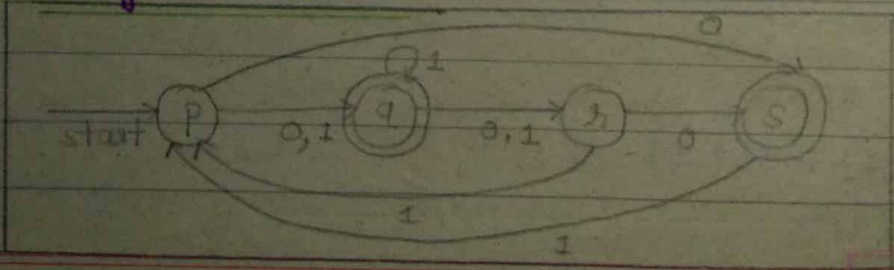
where δ is

Q \ Σ	0	1
p	q, s	q
q	r	q, r
r	s	p
s	—	p

→ $M = (\{p, q, r, s\}, \{0,1\}, \delta, p, \{q, s\})$

Q \ Σ	0	1
p	q, s	q
q	r	q, r
r	s	p
s	—	p

Diagram for NFA :→



δ (Mapping Function)

δ (p,0) = qs — New state

δ (p,1) = q

δ (q,0) = r

δ (q,1) = qr — New state

δ (r,0) = s

δ (r,1) = p

δ (s,0) = φ

δ (s,1) = p

δ' ((qs 0)) = δ(q,0) ∪ δ(s,0)

= r ∪ φ

$\boxed{\delta'((qs\,0)) = r}$

δ' (qs 1) = δ(q,1) ∪ δ(s,1)

= qr ∪ p

$\boxed{\delta'((qs,1)) = pqr}$ New state

δ' ((pqr) 0) = δ(p,0) ∪ δ(q,0) ∪ δ(r,0)

= qs ∪ r ∪ s

$\boxed{(\delta'(pqr\,0)) = qsr}$ New state

δ' ((pqr) 1) = δ(p,1) ∪ δ(q,1) ∪ (r,1))

= q ∪ qr ∪ p

$\boxed{\delta'(pqr,1) = pqr}$

δ' ((qr) 0) = δ(q,0) ∪ δ(r,0)

= r ∪ s

$\boxed{\delta'(qr,0) = rs}$

δ' ((qr) 1) = δ(q,1) ∪ δ(r,1)

= qr ∪ p

$\boxed{\delta'(qr,1) = pqr}$

δ' ((rs) 0) = δ(r 0) ∪ δ(s 0)

= s ∪ φ

$\boxed{\delta'((rs)0) = s}$

δ' ((rs) 1) = δ(r 1) ∪ δ(s 1)

= p ∪ p

$\boxed{\delta'((rs)1) = p}$

δ' ((qrs) 0) = δ(q 0) ∪ δ(r 0) ∪ δ(s 0)

= r ∪ s ∪ φ

$\boxed{\delta'((qrs)0) = rs}$

δ' ((qrs) 1) = δ(q 1) ∪ δ(r 1) ∪ δ(s 1)

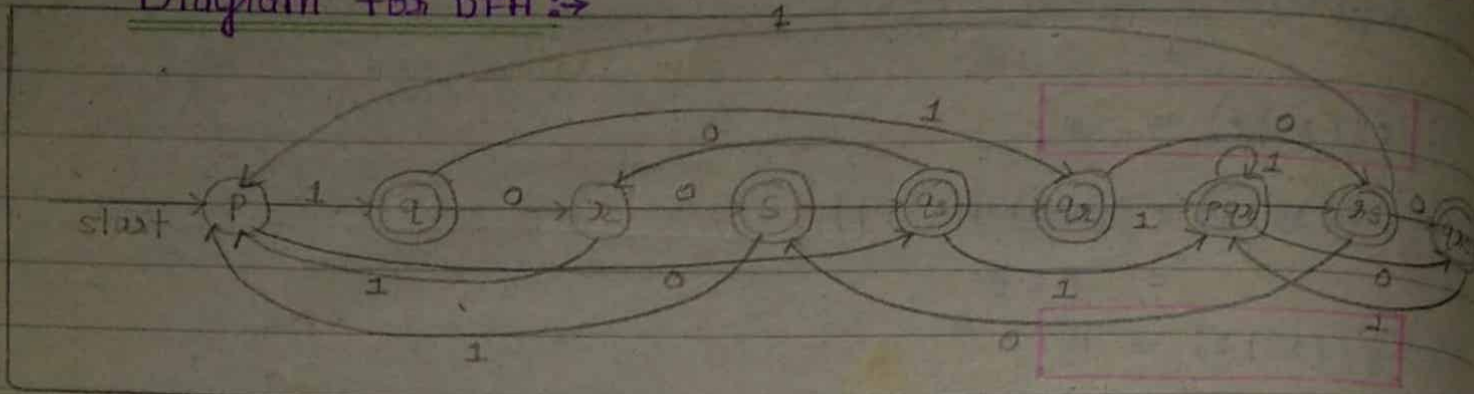
= qr ∪ p ∪ p

$\boxed{\delta'(qrs\,1) = pqr}$

Transition for DFA :→

Q \ Σ	0	1
p	qs	q
q	r	qr
r	s	p
s	—	p
qs	r	pqr
qr	rs	pqr
pqr	qrs	pqr
rs	s	p
qrs	rs	pqr

Diagram for DFA :-

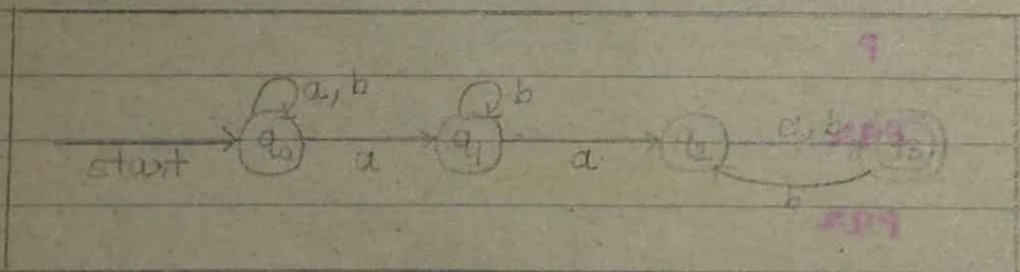


Que.

$Q \backslash \Sigma$	a	b
$\rightarrow q_0$	$q_0 q_1$	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	ϕ	q_2

$\longrightarrow$ $M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \{q_3\})$

Diagram for NFA :-



*W-07 * NFA with ϵ-moves *

Que. Define NFA with ϵ-moves.

→ If a finite automata model is modified to allow transition from one state to another state without consuming any of the input symbol then that modified model is known as NFA with ϵ-moves.

NFA with ϵ-moves is defined by 5 tuple.

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

where,

Q = finite set of internal state

Σ = finite set of input alphabet

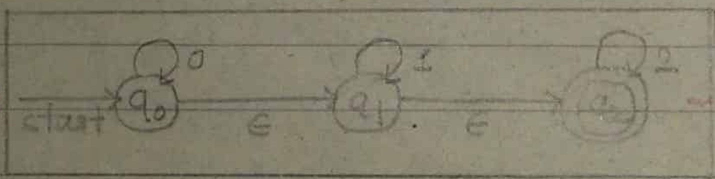
q_0 = initial state.

F = finite set of final state

δ = Mapping function.

$$\boxed{Q \times (\Sigma \cup \epsilon) \rightarrow 2^Q}$$

where, 2^Q is a power set of Q.

example → 

This a NFA with ϵ-moves as there is a transition from state q_0 to q_1 without consuming any of the i/p symbol. Similarly, from state q_1 to q_2 without consuming any of the i/p symbol. Such type of automata is known as NFA with ϵ-moves.

ϵ-closure of any state is a state itself & all the path labelled with ϵ (epsilon).

$\therefore$ ϵ-closure $q_0 = q_0 q_1 q_2$

ϵ-closure $q_1 = q_1 q_2$

ϵ-closure $q_2 = q_2$

from above example-

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

$\Sigma = \{0, 1, 2\}$

$q_0 = \{q_0\}$

$F = \{q_2\}$

δ (M.F.)

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = \phi$

$\delta(q_0, 2) = \phi$

$\delta(q_1, 0) = \phi$

$\delta(q_1, 1) = q_1$

$\delta(q_1, 2) = \phi$

$\delta(q_2, 0) = \phi$

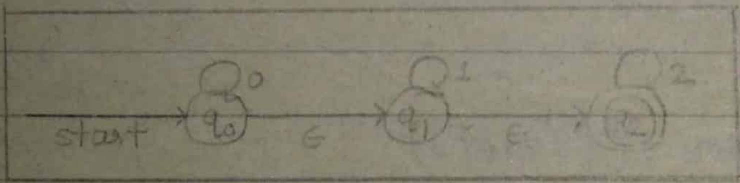
$\delta(q_2, 1) = \phi$

$\delta(q_2, 2) = q_2$

Q \ Σ	0	1	2
q_0	q_0	ϕ	ϕ
q_1	ϕ	q_1	ϕ
q_2	ϕ	ϕ	q_2

V.V.imp *** Que. Obtain NFA from NFA with ϵ-moves.

S-10 (b) → Consider, NFA with ϵ-moves.



Transition table for ϵ-moves :-

Q \ Σ	0	1	2
q_0	q_0	–	–
q_1	–	q_1	–
q_2	–	–	q_2

ϵ - closure $q_0 = q_0\, q_1\, q_2$

ϵ - closure $q_1 = q_1\, q_2$

ϵ - closure $q_2 = q_2$

$$\boxed{\hat{\delta}(q, a) = \epsilon\text{-closure }\delta(\hat{\delta}(q\ \epsilon)\ a)}$$

$\hat{\delta}(q_0, 0) = \epsilon\text{-closure }\delta(\hat{\delta}(q_0\ \epsilon)\ 0)$

$= \epsilon\text{-closure }\delta((q_0\, q_1, q_2)\, 0)$

$= \epsilon\text{-closure }\delta(q_0\ 0) \cup \delta(q_1, 0) \cup \delta(q_2\ 0)$

$= \epsilon\text{-closure }q_0 \cup \phi \cup \phi$

$= \cancel{q_0}\text{ closure }q_0$

$$\boxed{\hat{\delta}(q_0, 0) = q_0\, q_1\, q_2}$$

$\hat{\delta}(q_0, 1) = \epsilon\text{-closure }\delta(\hat{\delta}(q_0\ \epsilon)\ 1)$

$= \epsilon\text{-closure }\delta(q_0\, q_1, q_2\ 1)$

$= \epsilon\text{-closure }\delta(q_0\ 1) \cup \delta(q_1\ 1) \cup \delta(q_2\ 1)$

$= \epsilon\text{-closure }\phi \cup q_1\ \phi \cup$

$= \epsilon\text{-closure }q_1$

$$\boxed{\hat{\delta}(q_0, 1) = q_1\, q_2}$$

$\hat{\delta}(q_0, 2) = \epsilon\text{-closure }\delta(\hat{\delta}(q_0\ \epsilon)\ 2)$

$= \epsilon\text{-closure }\delta(q_0\, q_1, q_2\ 2)$

$= \epsilon\text{-closure }\delta(q_0\ 2) \cup \delta(q_1\ 2) \cup \delta(q_2\ 2)$

$= \epsilon\text{-closure }\phi \cup \phi \cup q_2$

$$\boxed{\hat{\delta}(q_0, 2) = q_2}$$

$\hat{\delta}(q_1, 0) = \epsilon\text{-closure }\delta(\hat{\delta}(q_1\ \epsilon)\ 0)$

$= \epsilon\text{-closure }\delta(q_1\, q_2\ 0)$

$= \epsilon\text{-closure }\delta(q_1, 0) \cup \delta(q_2\ 0)$

$= \epsilon\text{-closure }\phi \cup \phi$

$= \epsilon\text{-closure }\phi$

$$\boxed{\hat{\delta}(q_1, 0) = \phi}$$

$\hat{\delta}(q_1, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_1 \epsilon) 1)$
$= \epsilon\text{-closure } \delta(q_1 q_2\ 1)$
$= \epsilon\text{-closure } \delta(q_1 1) \cup \delta(q_2 1)$
$= \epsilon\text{-closure } q_1 \cup \phi$
$= \epsilon\text{-closure } q_1$

$\boxed{\hat{\delta}(q_1, 1) = q_1 q_2}$

$\hat{\delta}(q_1, 2) = \epsilon\text{-closure } \delta(\hat{\delta}(q_1 \epsilon) 2)$
$= \epsilon\text{-closure } \delta(q_1 q_2\ 2)$
$= \epsilon\text{-closure } \delta(q_1 2) \cup \delta(q_2, 2)$
$= \epsilon\text{-closure } \phi \cup q_2$
$= \epsilon\text{-closure } q_2$

$\boxed{\hat{\delta}(q_1, 2) = q_2}$

$\hat{\delta}(q_2, 0) = \epsilon\text{-closure } \delta(\hat{\delta}(q_2 \epsilon) 0)$
$= \epsilon\text{-closure } \delta(q_2\ 0)$
$= \epsilon\text{-closure } \phi$

$\boxed{\hat{\delta}(q_2, 0) = \phi}$

$\hat{\delta}(q_2, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_2 \epsilon) 1)$
$= \epsilon\text{-closure } \delta(q_2, 1)$
$= \epsilon\text{-closure } \phi$

$\boxed{\hat{\delta}(q_2, 1) = \phi}$

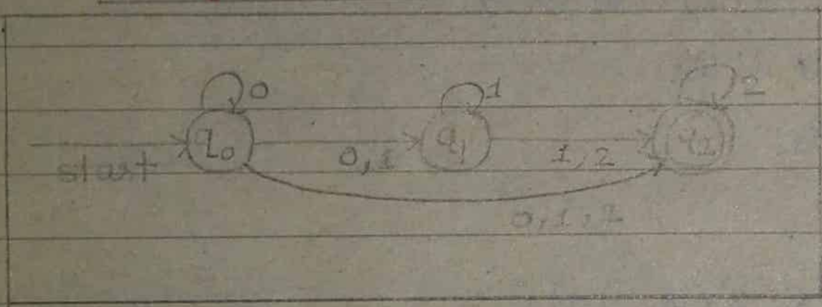
$\hat{\delta}(q_2, 2) = \epsilon\text{-closure } \delta(\hat{\delta}(q_2, \epsilon) 2)$
$= \epsilon\text{-closure } \delta(q_2, 2)$
$= \epsilon\text{-closure } q_2$

$\boxed{\hat{\delta}(q_2, 2) = q_2}$

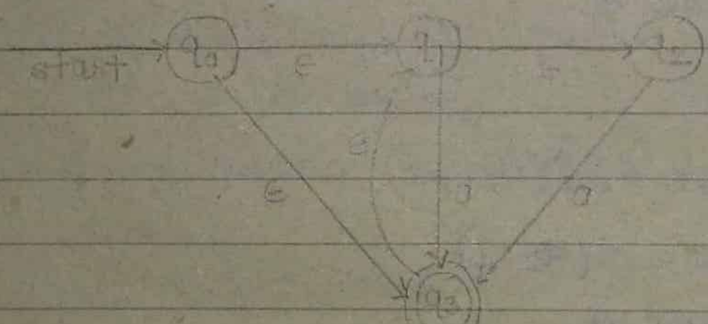
Transition Table for NFA :->

Q \ Σ	0	1	2
q_0	$q_0 q_1 q_2$	$q_1 q_2$	q_2
q_1	ϕ	$q_1 q_2$	q_2
q_2	ϕ	ϕ	q_2

Diagram for NFA :->



Que.



→ Transition table for ϵ-moves :-

Q \ Σ	0
q_0	–
q_1	q_3
q_2	q_3
q_3	–

ϵ-closure of $q_0 = q_0 q_1 q_2 q_3$
ϵ-closure of $q_1 = q_1 q_2$
ϵ-closure of $q_2 = q_2$
ϵ-closure of $q_3 = q_3 q_1 q_2$

$\grave{\delta}(q_0, 0) = \epsilon\text{-closure } \delta(\grave{\delta}(q_0, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_0 q_1 q_2 q_3) 0)$
$= \epsilon\text{-closure } \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)$
$= \epsilon\text{-closure } \phi \cup q_3 \cup q_3 \cup \phi$
$= \epsilon\text{-closure } q_3$
$\boxed{\grave{\delta}(q_0, 0) = q_1 q_2 q_3}$

$\grave{\delta}(q_1, 0) = \epsilon\text{-closure } \delta(\grave{\delta}(q_1, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_1 q_2) 0)$
$= \epsilon\text{-closure } \delta(q_1, 0) \cup \delta(q_2, 0)$
$= \epsilon\text{-closure } q_3 \cup q_3$
$= \epsilon\text{-closure } q_3$
$\boxed{\grave{\delta}(q_1, 0) = q_2 q_1 q_3}$

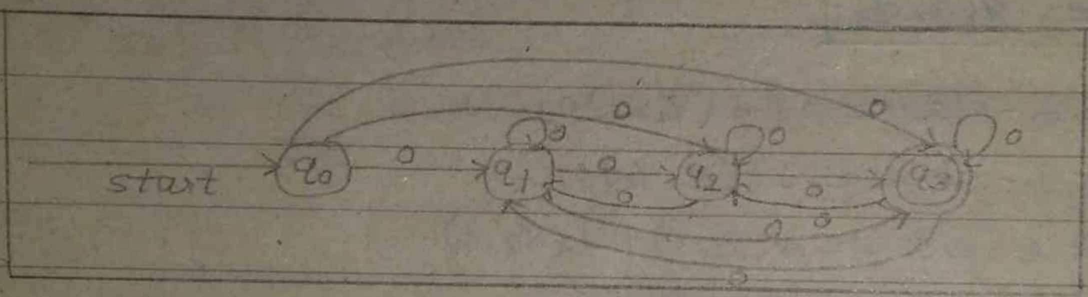
$\grave{\delta}(q_2, 0) = \epsilon\text{-closure } \delta(\grave{\delta}(q_2, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_2) 0)$
$= \epsilon\text{-closure } q_3$
$\boxed{\grave{\delta}(q_2, 0) = q_1 q_2 q_3}$

$\grave{\delta}(q_3, 0) = \epsilon\text{-closure } \delta(\grave{\delta}(q_3, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_1 q_2 q_3) 0)$
$= \epsilon\text{-closure } \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)$
$= \epsilon\text{-closure } q_3 \cup q_3 \cup \phi$
$= \epsilon\text{-closure } q_3$
$\boxed{\grave{\delta}(q_3, 0) = q_1 q_2 q_3}$

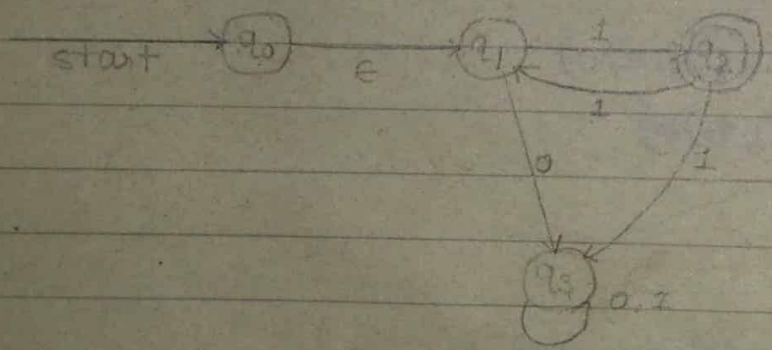
Transition Table for NFA :-

Q \ Σ	0
q_0	$q_1\ q_2\ q_3$
q_1	$q_1\ q_2\ q_3$
q_2	$q_1\ q_2\ q_3$
q_3	$q_1\ q_2\ q_3$

Diagram for NFA :-



Que.



Transition Table for ∈-moves :-

Q \ Σ	0	1
q_0	–	–
q_1	q_3	q_2
q_2	–	$q_3\ q_1$
q_3	q_3	q_3

ϵ-closure $q_0 = q_0 q_1$
ϵ-closure $q_1 = q_1$
ϵ-closure $q_2 = q_2$
ϵ-closure $q_3 = q_3$

$\hat{\delta}(q_0, 0) = \epsilon\text{-closure } \delta(\hat{\delta}(q_0, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_0 q_1) 0)$
$= \epsilon\text{-closure } \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \epsilon\text{-closure } \phi \cup q_3$
$= \epsilon\text{-closure } q_3$

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

$\hat{\delta}(q_0, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_0, \epsilon) 1)$
$= \epsilon\text{-closure } \delta((q_0 q_1) 1)$
$= \epsilon\text{-closure } \delta(q_0, 1) \cup \delta(q_1, 1)$
$= \epsilon\text{-closure } \phi \cup q_2$
$= \epsilon\text{-closure } q_2$

$\boxed{\hat{\delta}(q_0, 1) = q_2}$

$\hat{\delta}(q_1, 0) = \epsilon\text{-closure } \delta(\hat{\delta}(q_1, \epsilon) 0)$
$= \epsilon\text{-closure } \delta((q_1, 0))$
$= \epsilon\text{-closure } q_3$

$\boxed{\hat{\delta}(q_1, 0) = q_3}$

$\hat{\delta}(q_1, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_1, \epsilon) 1)$
$= \epsilon\text{-closure } \delta(q_1, 1)$
$= \epsilon\text{-closure } q_2$

$\boxed{\hat{\delta}(q_1, 1) = q_2}$

$\delta'(q_2, 0) = \epsilon\text{-closure } \delta(\hat{\delta}'(q_2, \epsilon) 0)$
$= \epsilon\text{-closure } \delta(q_2, 0)$
$= \epsilon\text{-closure } \phi$

$\boxed{\hat{\delta}(q_2, 0) = \phi}$

$\hat{\delta}(q_2, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_2, \epsilon) 1)$
$= \epsilon\text{-closure } \delta(q_2, 1)$
$= \epsilon\text{-closure } q_1 q_3$
$= \epsilon\text{-closure } q_1 \cup q_3$

$\boxed{\hat{\delta}(q_2, 1) = q_1 q_3}$

$\delta'(q_3, 0) = \epsilon\text{-closure } \delta(\hat{\delta}'(q_3, \epsilon) 0)$
$= \epsilon\text{-closure } \delta(q_3, 0)$
$= \epsilon\text{-closure } q_3$

$\boxed{\hat{\delta}(q_3, 0) = q_3}$

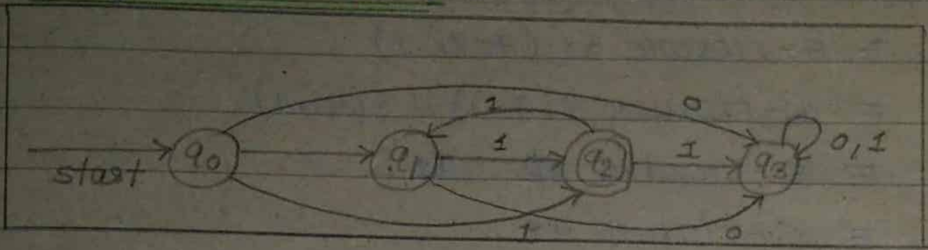
$\hat{\delta}(q_3, 1) = \epsilon\text{-closure } \delta(\hat{\delta}(q_3, \epsilon) 1)$
$= \epsilon\text{-closure } \delta(q_3, 1)$
$= \epsilon\text{-closure } q_3$

$\boxed{\hat{\delta}(q_3, 1) = q_3}$

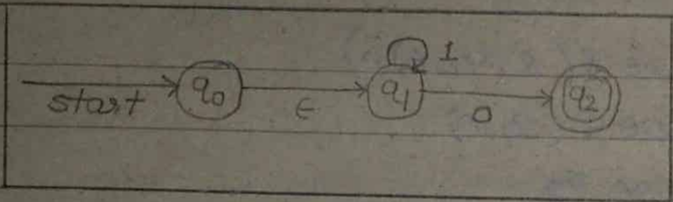
T. T. for NFA :>

Q \ Σ	0	1
q_0	q_3	q_2
q_1	q_3	q_2
q_2	ϕ	$q_1 q_3$
q_3	q_3	q_3

Diagram for NFA :-



Que.



T. T. for ε-moves :-

Q \ Σ	0	1
q_0	–	–
q_1	q_2	q_1
q_2	–	–

ϵ – closure $q_0 = q_0 q_1$

ϵ – closure $q_1 = q_1$

ϵ – closure $q_2 = q_2$

$$\hat{\delta}(q_0, 0) = \epsilon\text{-closure } \delta(\hat{\delta}(q_0, \epsilon) 0)$$
$$= \epsilon\text{-closure } \delta((q_0 q_1), 0)$$
$$= \epsilon\text{-closure } \delta(q_0, 0) \cup \delta(q_1, 0)$$
$$= \epsilon\text{-closure } \phi \cup q_2$$
$$= \epsilon\text{-closure } q_2$$

$$\boxed{\hat{\delta}(q_0, 0) = q_2}$$

* Regular Expression *

S-09

Que. Define Regular Expression.

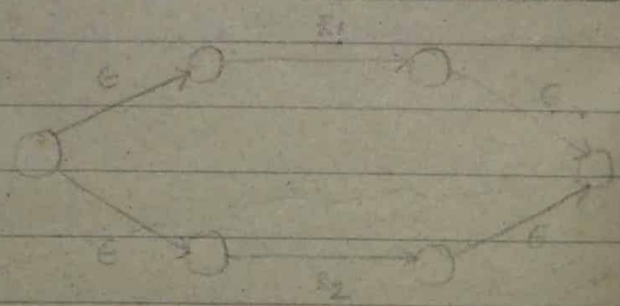
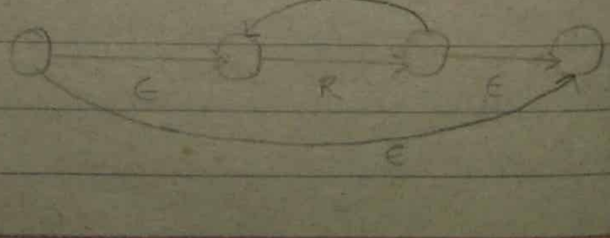
→ Regular expression over Σ (input symbol) is define as follows-

(i) If R_1 & R_2 are a regular expression then $R_1 \cup R_2$ & $R_1 + R_2$ is also a regular expression.

(ii) If R_1 & R_2 are regular expression then $R_1 \cap R_2$ & $R_1 \cdot R_2$ is also a regular expression.

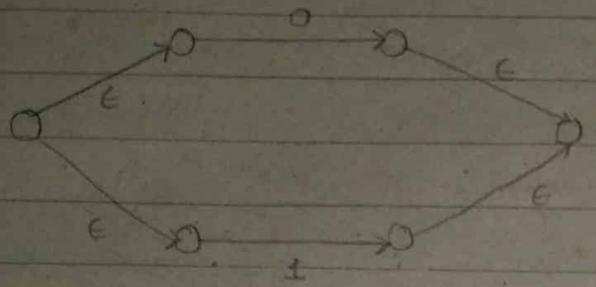
(iii) If R is a regular expression then R^* (R-closure) is also a regular expression.

* Conversion of Regular Expression to NFA with ϵ-moves

Regular Expression	NFA ϵ-moves
(1) $R_1 + R_2$, $R_1 \cup R_2$	
(2) $R_1 \cdot R_2$, $R_1 \cap R_2$	
(3) R^*	

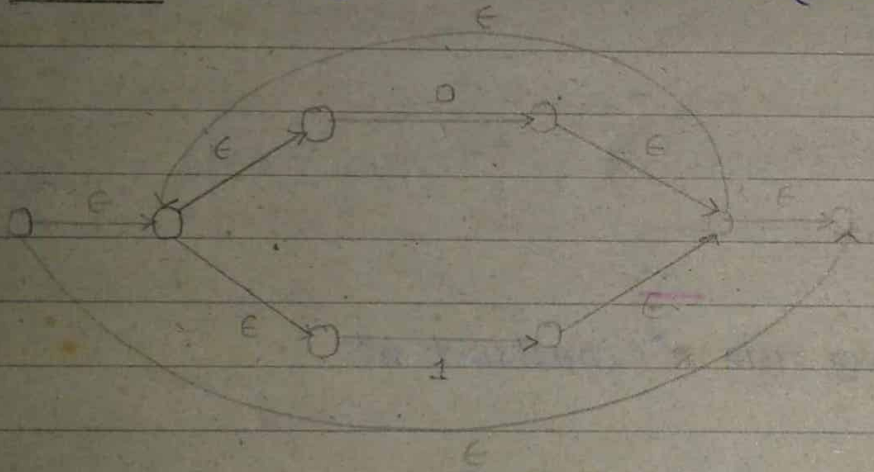
Que. Construct finite automata for a regular expression.
$(0+1)^*.0$

→ Step 1 :- Using rule R_1+R_2 construct $(0+1)$



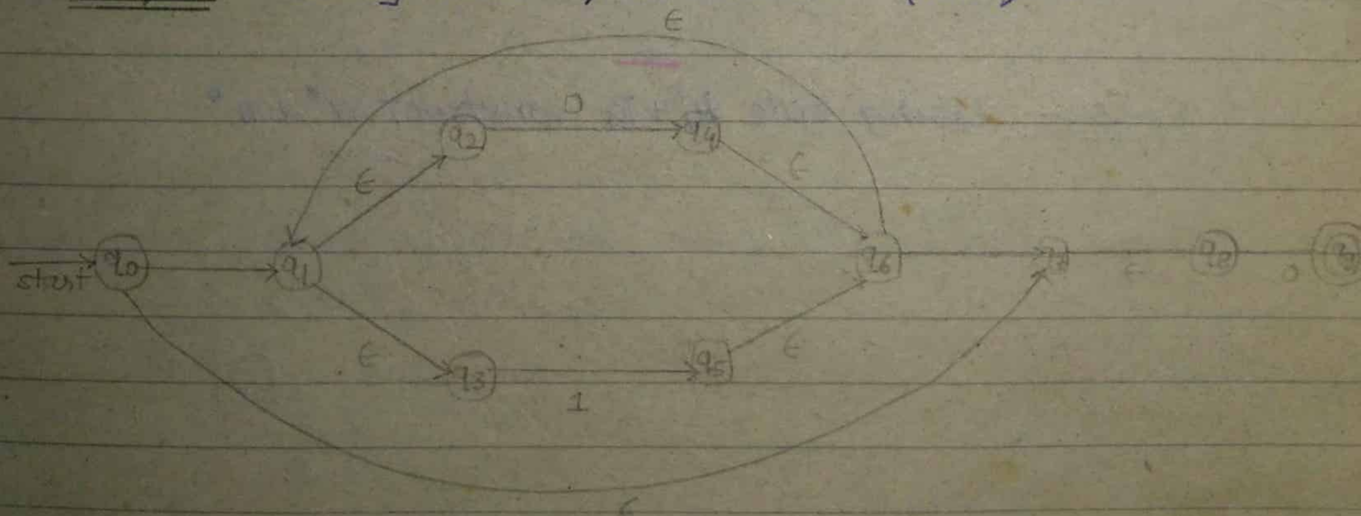
$(0+1)$

Step 2 :- Using rule R^* construct $(0+1)^*$



$(0+1)^*$

Step 3 :- Using rule $R_1.R_2$ construct $(0+1)^*.0$



$(0+1)^*.0$

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

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$

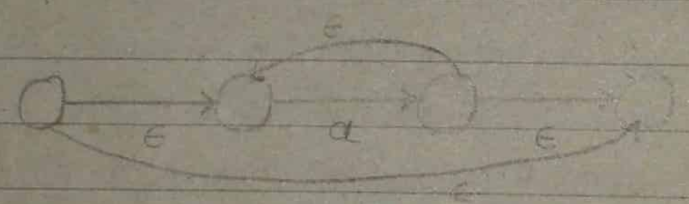
$\Sigma = \{0, 1\}$

$q_0 = \{q_0\}$

$F = \{q_9\}$

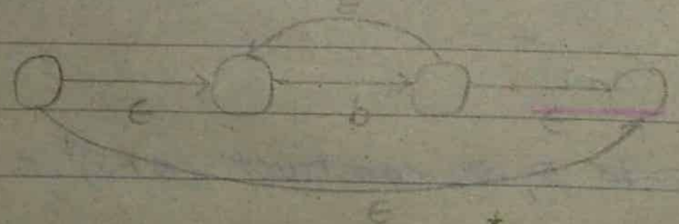
Que. $a^* + b^*$

→ step 1 :- Using rule R^* construct a^*



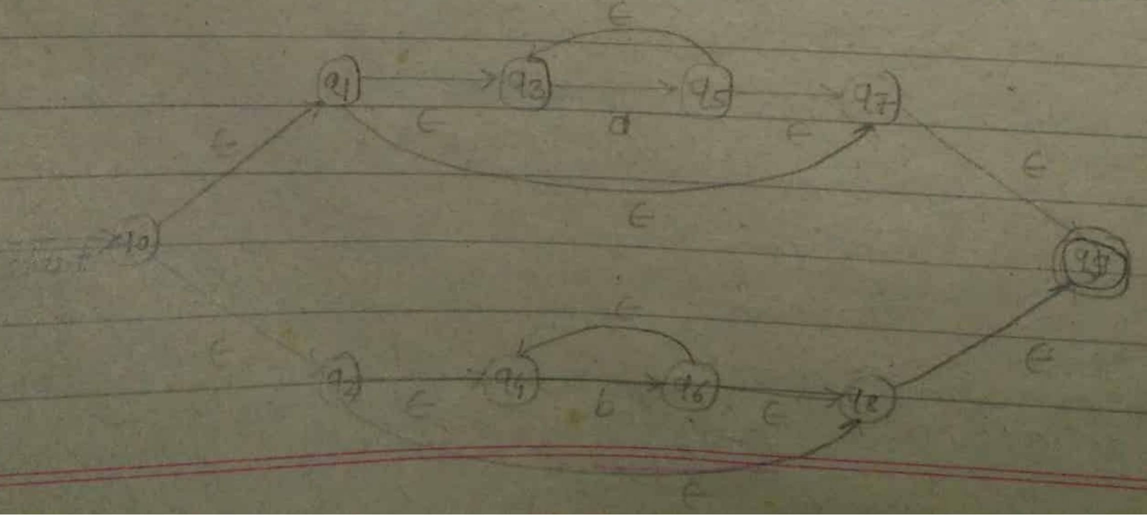
a^*

step 2 :- Using rule R^* construct b^*



b^*

step 3 :- Using rule $R_1 + R_2$ construct $a^* + b^*$



$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$

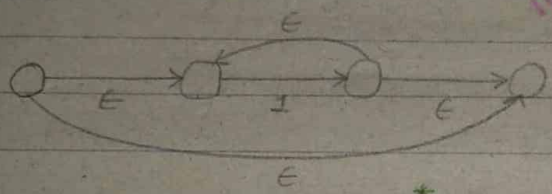
$\Sigma = \{a, b\}$

$q_0 = \{q_0\}$

$F = \{q_9\}$

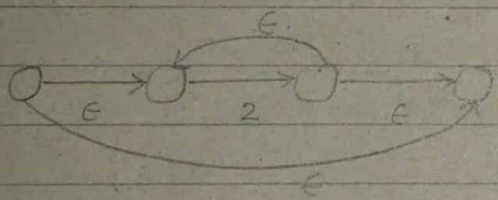
Que. $1^* 2^* 3^*$

→ Step 1 :- Using rule R^* construct 1^*



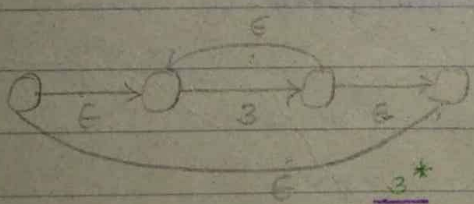
1^*

Step 2 :- Using rule R^* construct 2^*



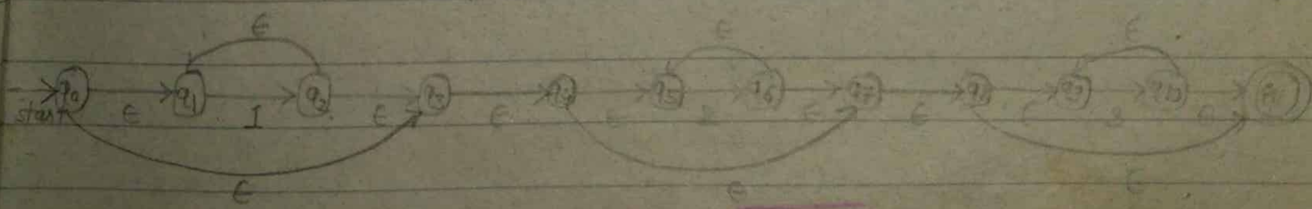
2^*

Step 3 :- Using rule R^* construct 3^*



3^*

Step 4 :- Using rule $R_1 \cdot R_2$ construct $1^* 2^* 3^*$



$1^* 2^* 3^*$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$

$\Sigma = \{1, 2, 3\}$

$q_0 = \{q_0\}$

$F = \{q_{11}\}$

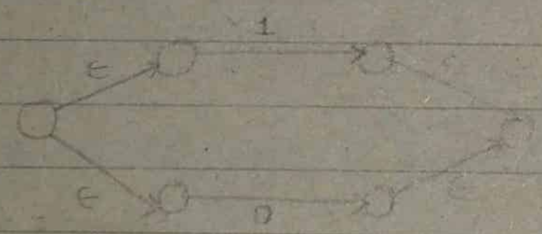
Que. Construct NFA equivalent to regular expression.

W-07 ① $1(1+0)^*$

W-06 S-05 ② $(ab \cup aab)^*$

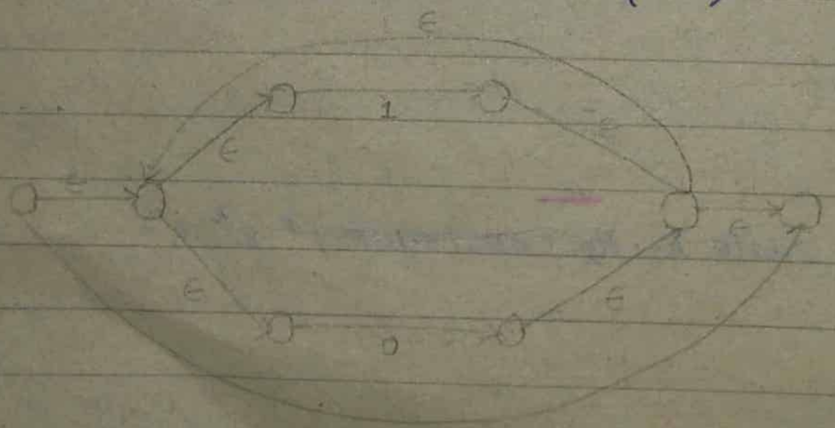
→ ① $1(1+0)^*$

step 1 :- Using rule R_1+R_2 construct $(1+0)$



$(1+0)$

step 2 :- Using rule R^* construct $(1+0)^*$

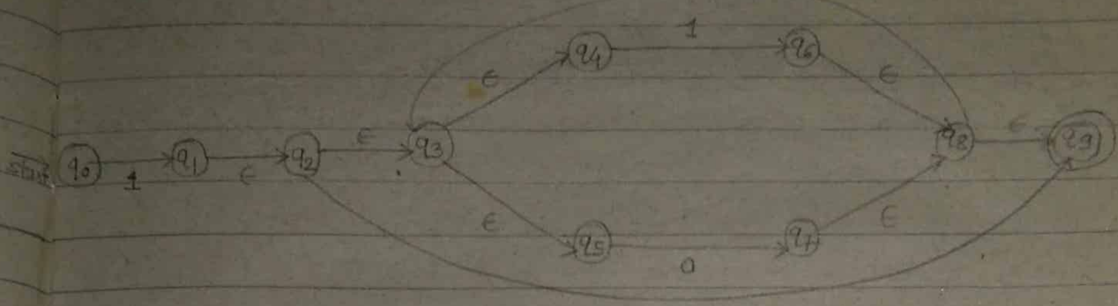


$(1+0)^*$

o, q11}

ession.

Step 3 :- Using rule $R_1 \cdot R_2$ construct $1.(1+0)^*$



$1.(1+0)^*$

② $(ab + aab)^*$

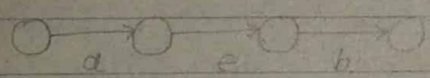
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9\}$

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

$q_0 = \{q_0\}$

$F = \{q_9\}$.

Step 1 :- Using rule $R_1 \cdot R_2$ construct ab.

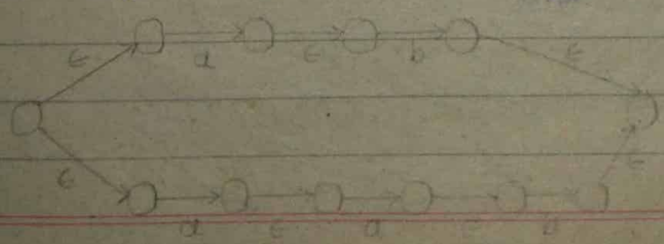


ab

Step 2 :- Using rule $R_1 \cdot R_2$ construct aab.

aab

Step 3 :- Using rule $R_1 + R_2$ construct $(ab + aab)$



ab+aab