Introduction

1) Symbol : basic entity which is not yet defined.
eg : A, ABCD, 1, 2, ...

2) Alphabet : Symbol which cannot be further divided.
eg : { A, B ... Z }
{ a, b ... z }
{ 0, ... , 9 }

3) String : Finite set of alphabets or string

4) Null string : string with length
$|\epsilon| = 0$

5) Prefix : set of alphabets at the start of word

$x = \{a, b, c\}$
prefix : { a, ab, abc, ϵ }.

6) Suffix : Set of alphabets at the end of the word
suffix : { a, ab, abc, ϵ }

7) Language : set of strings which contain conditions.

$X = \{a, b, c\}$
(condition specified)

$L = \{ab, bc, ac, \epsilon\}$

8) Closure kleene's (kleene's star) *

+ → positive closure (~~zero~~ one or more occurences)

* → kleene's closure (zero or more occurences)

$L^+ = \{a, b, c, ab, bc, ac \ldots\}$

$L^* = \{\epsilon, a, b, c, ab, bc, ac \ldots\}$

Example :

1) $\Sigma = \{a, b\}$.
string containing only a's
$L = \{a, aa, aaa, \ldots\}$

2) String starting with a and ending with b
(inbetween i can have any combination of a and b or only a or only b)
$L = \{ab, aab, abab, aaabbb \ldots\}$

3) string of even length string $\{a, b\}$

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

* you can put ϵ in this string because ϵ is null = 0.

$|\epsilon| = 0$

4) string having even no of a' and odd no of b's.

~~a~~ ~~ab~~

$L = \{aab, baa \ldots\}$

closure : $S = \{a, b\}$

1) $S^* = \{\epsilon, a, b, ab\}$

$S^+ = \{a, b, ab\}$

2) $S = \{00, 11\}$

$S^* = \{\epsilon, 00, 11, 0011, 1100, \ldots\}$

$S^+ = \{00, 11, 0011, 1111, 0000 \ldots\}$

3) $S = \{aa, b\}$

condition $\Rightarrow$ length = 2.

Kleene's closure

$S^* = \{aa, bb\}$.

4) condition $\Rightarrow$ length less than 4 or 4

$S = \{aa, b\}$

$L = \{\epsilon, aa, b, aab, baa, aaaa, bbaa, aabb, bb, bbb, baab, bbbb\}$

FINITE AUTOMATA

FSM

= finite state machine

on off

⇒ anything which is triggered to change its state.

$(q_0) \rightarrow (q_1)$ Transition diag.

FSM	FA

start from q_s and then q_0.

An automation in which set Q contains only finite no of elements is FSM.

A FSM is formally defined as a 5-tuple format :

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

- Q → set of states
- Σ → Alphabet
- δ → Transition
- q_s → start state
- F → final state.

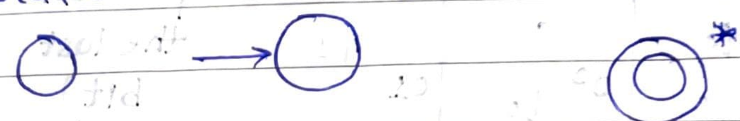
★ FA is mathematical model of FSM

1) Type I (ending with)

Design a finite automata that takes input 101 over i/p set $\Sigma = \{0, 1\}$

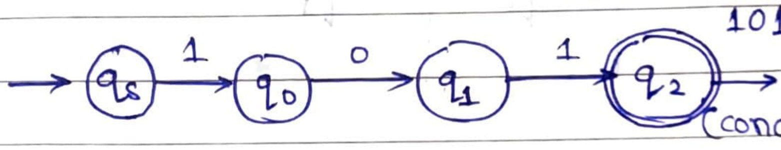
starting
start·

solution : State



end state.

Transition diagram :



Accept state (concentric circles)

eg: 0101 1101 not accepted

2) Design FSM in which i/p is valid it ends in 100 over $\Sigma = \{0,1\}$

(left to right)

Step 1) List out no. of states in Q
$(Q, \Sigma, \delta, q_s, F)$ set of states

logic : $q_s \rightarrow$ start state
$q_0 \rightarrow$ ending in 0
$q_1 \rightarrow$ ending in 1
$q_2 \rightarrow$ ending in 10
$q_3 \rightarrow$ ending in 100

Step 2) $\Sigma = \{0,1\}$
$\delta = Q \times \Sigma \rightarrow Q$ (Total transitions = $\underset{q_s, q_0, q_1, q_2, q_3}{5} \times \underset{0,1}{2} = 10$)

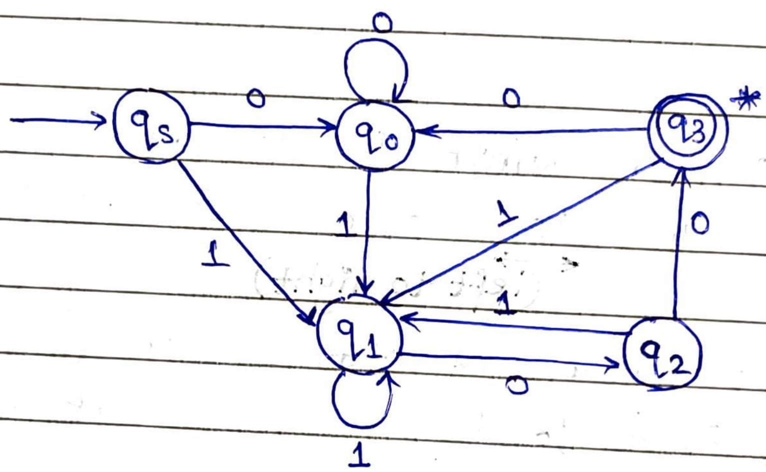
Step 3) Transition ~~state~~ table

Q	states \ i/p	0	1
→	q_s	q_0	q_1
0	q_0	q_0	q_1
1	q_1	q_2	q_1
10	q_2	q_3	q_1
100	q_3*	[illegible]	q_1

Q	states \ i/p Σ	0	1
→	q_s	q_0	q_1
0	q_0	00 q_0	01 q_1
1	q_1	10 q_2	11 q_1
10	q_2	100 q_3	101 q_1
100	q_3^*	1000 q_0	1001 q_1

(focus on the last bit)

Step 4] Transition diagram :



Step 5] Machine Function table

$Q \times \Sigma \longrightarrow O$ (y/n)

Q \ Σ	0	1
q_s	n	n
q_0	n	n
q_1	n	n
q_2	y	n
q_3	n	n.

★ skip this step if the question is finite autom

← perform this step if question is desi FSM

★ It will have y only where 100 is a word.

step 6] Take Example and verify

$\delta(q_s, 0100)$ (Accepted state)

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

Transition :

$\vdash \delta(\delta(q_s, 0), 100)$
$\vdash \delta(q_0, 100)$
$\vdash \delta(\delta(q_0, 1), 00)$
$\vdash \delta(q_1, 00)$
$\vdash \delta(\delta(q_1, 0), 0)$
$\vdash \delta(q_2, 0)$
$\vdash \delta(q_2, 0)$
$\vdash q_3$ (Accept)

$\delta(q_s, 0101)$ (Not Accepted state)

$\vdash \delta(\delta(q_s, 0), 101)$
$\vdash \delta(q_0, 101)$
$\vdash \delta(\delta(q_0, 1), 01)$
$\vdash \delta(q_1, 01)$
$\vdash \delta(\delta(q_1, 0), 1)$
$\vdash \delta(q_2, 1)$
$\vdash q_1$ (Reject)

3) Design a FSM which ends at aab over $\Sigma = \{a, b\}$
left to right

step 1) $Q = \{q_s, q_0, q_1, q_2, q_3\}$

$q_s \rightarrow$ start state
$q_0 \rightarrow$ ending at a
$q_1 \rightarrow$ ending at b
$q_2 \rightarrow$ ending at aa
$q_3 \rightarrow$ ending at aab.

Step 2) $\Sigma = \{a, b\}$

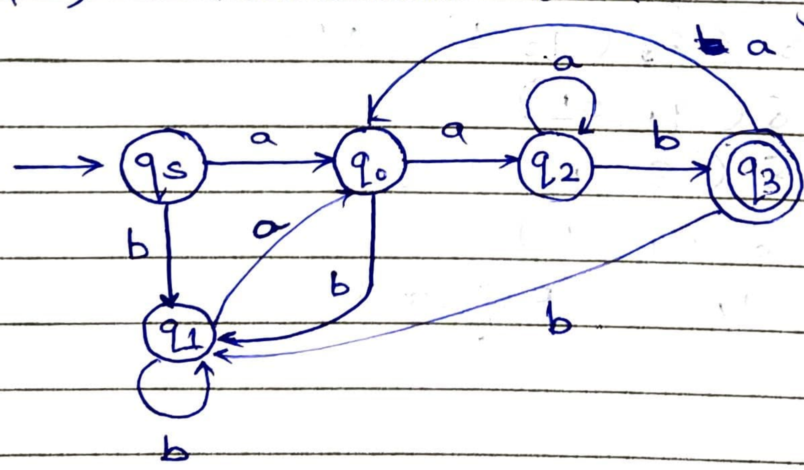
$\delta = Q \times \Sigma \longrightarrow Q$

$= 5 \times 2 \Longrightarrow 10$

Step 3) Transition table

	$Q \backslash \Sigma$	a	b
	q_s	a q_0	b q_1
a	q_0	aa q_2	ab q_1
b	q_1	ba q_0	bb q_1
aa	q_2	aaa q_2	aab q_3
aab	q_3	aaba q_0	aabb q_1

Step 4) Transition ~~table~~ diagram:



Step 5) Machine Function Table

$Q \times \Sigma \longrightarrow O$ (Y/N)

	Q \ Σ	a	b
	q_s	n	n
a	q_0	n	n
b	q_1	n	n
aa	q_2	n	Y
aab	q_3	n	n

Step 6) Example and verify

$\delta(q_s, aaab)$
$\vdash \delta(\delta(q_s, a), aab)$
$\vdash \delta(q_0, aab)$
$\vdash \delta(\delta(q_0, a), ab)$
$\vdash \delta(q_2, ab)$
$\vdash \delta(\delta(q_2, a), b)$
$\vdash \delta(q_2, b)$
$\vdash \boxed{q_3}$. (Accept)

$\delta(q_s, aaba)$
$\vdash \delta(\delta(q_s, a), aba)$
$\vdash \delta(q_0, aba)$
$\vdash \delta(\delta(q_0, a), ba)$
$\vdash \delta(q_2, ba)$
$\vdash \delta(\delta(q_2, b), a)$
$\vdash \delta(q_3, a)$
$\vdash \boxed{q_0}$ (Reject)

Type II] OR

Q1) Design FA over the i/p $\Sigma = \{a, b\}$ ending with abb or baa.

Solution: Step 1) $Q(q_s, q_0, q_1, q_2, q_3, q_4, q_5)$

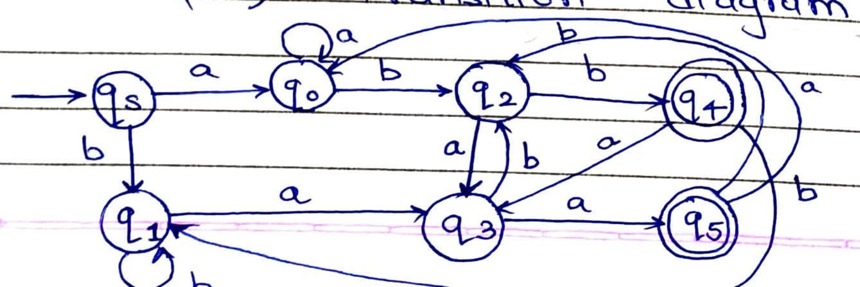
$q_s \rightarrow$ start state
$q_0 \rightarrow$ ~~starting~~ ending with a
$q_1 \rightarrow$ ~~starting~~ with b
$q_2 \rightarrow$ ~~starting~~ with ab
$q_3 \rightarrow$ ~~starting~~ with ba
$q_4 \rightarrow$ ~~starting~~ with abb
$q_5 \rightarrow$ ~~starting~~ with baa

Step 2) $\delta = Q \times \Sigma = 7 \times 2 = 14$.

Step 3) Transition table

	Q \ Σ	a	b
→	q_s	a q_0	b q_1
a	q_0	aa q_0	ab q_2
b	q_1	ba q_3	bb q_1
ab	q_2	aba q_3	abb q_4
ba	q_3	baa q_5	bab q_2
abb	q_4	abba q_3	abbb q_1
baa	q_5	baaa q_0	baab q_2

Step 4) Transition diagram



Step 5) Example.

Q2) $\Sigma = \{0,1\}$ ends in 00 or 11.

Soln : Step 1) $Q = \{q_s, q_0, q_1, q_2, q_3\}$

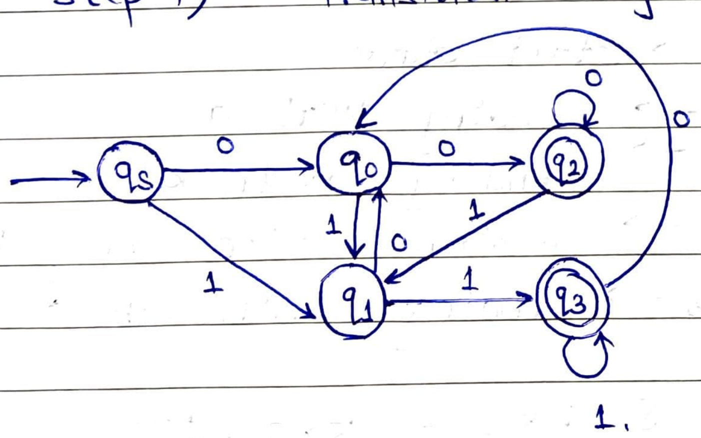
$q_s \longrightarrow$ start state
$q_0 \longrightarrow$ ~~start~~ ending in 0
$q_1 \longrightarrow$ ending in 1
$q_2 \longrightarrow$ ending in 00
$q_3 \longrightarrow$ ending in 11

Step 2) $\delta = Q \times \Sigma = 5 \times 2 = 10$

Step 3) Transition table

	Q \ Σ	0	1
	$\rightarrow q_s$	0 q_0	1 q_1
0	q_0	00 q_2	01 q_1
1	q_1	10 q_0	11 q_3
00	q_2	000 q_2	001 q_1
11	q_3	110 q_0	111 q_3

Step 4) Transition diagram



→

Step 5) Example 010011.

$\delta(q_s, 010011)$
$\vdash \delta(\delta(q_s, 0), 10011)$
$\vdash \delta(q_0, 10011)$
$\vdash \delta(\delta(q_0, 1), 0011)$
$\vdash \delta(q_1, 0011)$
$\vdash \delta(\delta(q_1, 0), 011)$
$\vdash \delta(q_0, 011)$
$\vdash \delta(\delta(q_0, 0), 11)$
$\vdash \delta(q_2, 11)$
$\vdash \delta(\delta(q_2, 1), 1)$
$\vdash \delta(q_1, 1)$
$\vdash \boxed{q_3}$ (Accept).

Q3) 2^{nd} last symbol is a over $\Sigma = \{a, b\}$

Solution:

step 1) $Q = \{q_s, q_0, q_1, q_2, q_3\}$

$q_s \rightarrow$ start state
$q_0 \rightarrow$ ending with a
$q_1 \rightarrow$ ending with b.
$q_2 \rightarrow$ ending with aa
$q_3 \rightarrow$ ending with ab

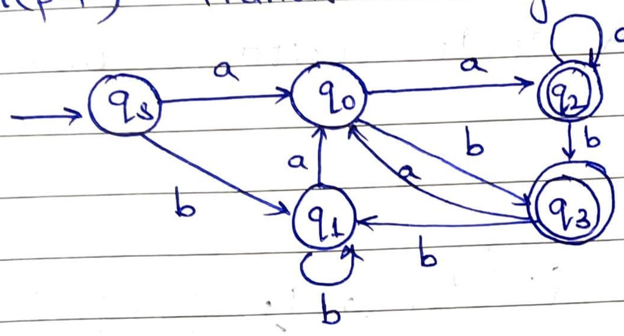
} both are valid.

step 2) $\delta = Q \times \Sigma = 5 \times 2 = 10$.

Step 3) Transition table.

	Q \ Σ	a		b	
→	q_s	a	q_0	b	q_1
a	q_0	aa	q_2	ab	q_3
b	q_1	ba	q_0	bb	q_1
aa	* q_2	aaa	q_2	aab	q_3
ab	* q_3	aba	q_0	abb	q_1.

step4) Transition diagram



Type III] (contains)

Q1) Design FA if it contains string 011 over i/p $\Sigma = \{0,1\}$

step 1) $Q = \{q_s, q_0, q_1, q_2, \overset{*}{q_3}\}$

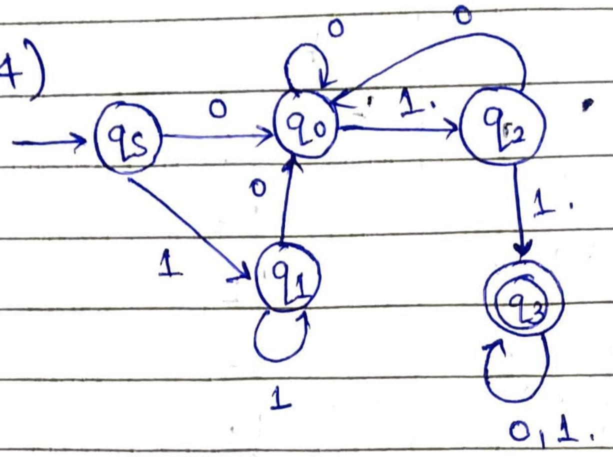
$q_0 \downarrow 0$, $q_1 \downarrow 1$, $q_2 \downarrow 01$, $q_3 \downarrow 011$

step3)

	Q \ Σ	0		1	
	q_s	0	q_0	1	q_1
0	q_0	00	q_0		q_2
1	q_1		q_0		q_1
01	q_2		q_0		q_3
011	q_3 *		~~q_0~~ × q_3		~~q_1~~ × q_3.

× no need for transition so remain in same state

Step 4)



Q2) Design a FA if it contains bba over $\Sigma = \{a, b\}$

<u>Solution:</u>

Step 1] $Q = \{q_s, q_0, q_1, q_2, q_3\}$

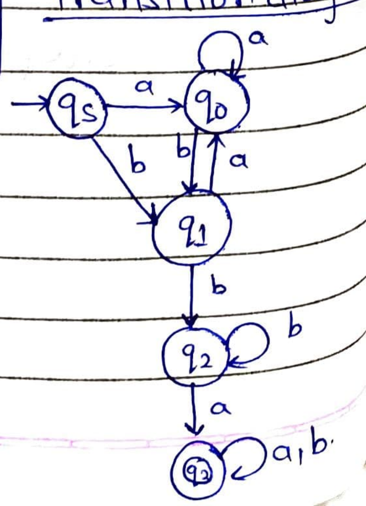
$q_s \rightarrow$ start state
$q_0 \rightarrow$ ~~ending~~ contains in a
$q_1 \rightarrow$ ending in b
$q_2 \rightarrow$ ending in bb
$q_3 \rightarrow$ ending in bba

Step 2] $\delta = Q \times \Sigma = 5 \times 2 = 10$

Step 3] Transition table

	Q \ Σ	a	b
→	q_s	a q_0	b q_1
a	q_0	aa q_0	ab q_1
b	q_1	ba q_0	bb q_2
bb	q_2	bba q_3	bbb q_2
bba	q_3	bbaa ~~q_0~~ × q_3	bbab q_3

4] Transition diagram



Type IV : [Does not contain]

Q1) Design FA that does not contain bbb over i/p $\Sigma = \{a, b\}$

Soln :
I) ending with bbb
II) contains bbb
III) does not contain bbb

Step I) $Q = \{q_s, q_0, q_1, q_2, q_3\}$

$q_s \rightarrow$ start state
$q_0 \rightarrow$ ending with ~~q_0~~ a
$q_1 \rightarrow$ ending with ~~q_1~~ b
$q_2 \rightarrow$ ending in bb
$q_3 \rightarrow$ ending in bbb

Step II) $\delta = Q \times \Sigma$

Step III) Transition table if it contains bbb.

II contains.

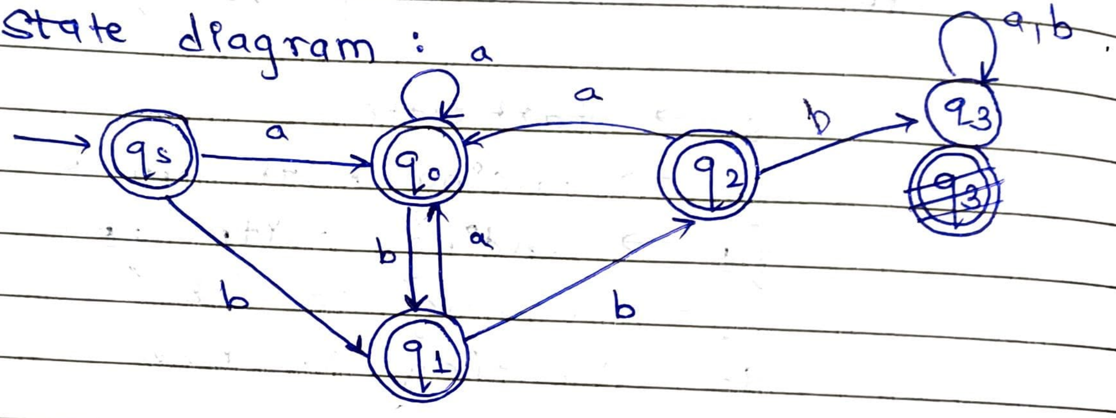
	Q \ Σ	a	b
$\rightarrow$	q_s	a q_0	b q_1
a	q_0	aa q_0	ab q_1
b	q_1	ba q_0	bb q_2
bb	q_2	bba q_0	bbb q_3
bbb	q_3*	bbba q_3	bbbb q_3

Step IV] Transition table for does not contain bbb.

whichever is non-final state becomes final state.

Q \ Σ	a	b
* q_s	q_0	q_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

State diagram :



Example: abbbabb

Q2) Does not contain 1011 over $\Sigma = \{0, 1\}$

Solution: I) $Q = \{q_s, q_0, q_1, q_2, q_3, q_4\}$

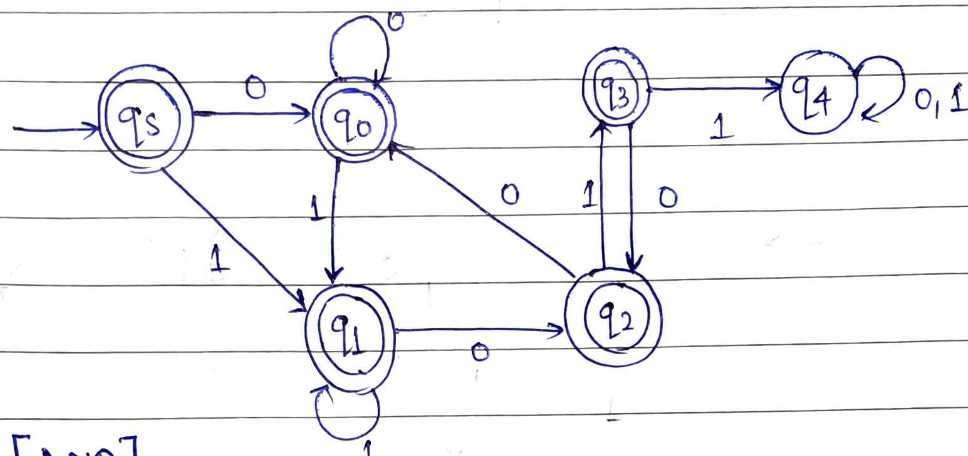
ending in 0 1 10 101 1011

II) $\delta = Q \times \Sigma = 6 \times 2 = 12$

III) Transition table if it contains 1011

	Q \ Σ	0	1
→	q_s	0 q_0	1 q_1
0	q_0	00 q_0	01 q_1
1	q_1	10 q_2	11 q_1
10	q_2	100 q_0	101 q_3.
101	q_3	1010 q_2	1011 q_4.
1011	q_4	10110 q_4	10111 q_4

III)



Type V [AND]

Q1) Design FA in which i/p is valid if it contains even no of 0's and odd no. of 1's over $\Sigma = \{0, 1\}$

Solution : 1) $q_s \rightarrow$ starting state

e e $q_0 \rightarrow$ even no of 0's & even no of 1's

* e o $q_1 \rightarrow$ even no of 0's & odd no of 1s'

o e $q_2 \rightarrow$ odd no of 0's & even no of 1s'

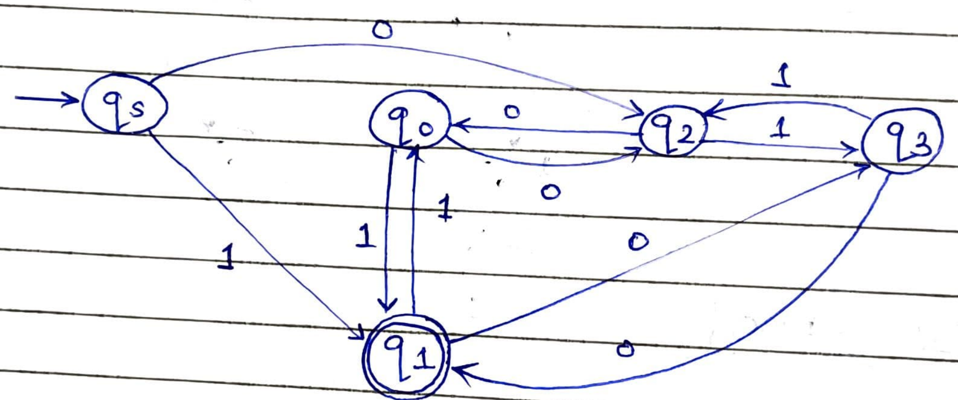
o o $q_3 \rightarrow$ odd no of 0's & odd no of 1s'

2)

3)

	Q \ Σ	0	1
	q_s	q_2	q_1
ee	q_0	q_2	q_1
eo	* q_1	q_3	q_0
oe	q_2	q_0	q_3
oo	q_3	q_1	q_2

4) <u>State diagram:</u>



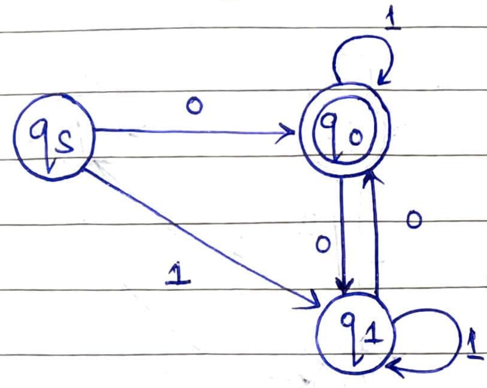
Q2) Odd no of 0's and any no. of 1's over i/p $\Sigma = \{0,1\}$

<u>Solution:</u> 1) $q_s \rightarrow$ starting state
$q_0 \rightarrow$ odd no of ~~1's~~ 0's
$q_1 \rightarrow$ even no of ~~1's~~ 0's

2)

Q \ Σ	0	1
q_s	q_0	q_1
* q_0	q_1	q_0
q_1	q_0	q_1

3) State Diagram :



Type VI : Divisible :

Q1) Design a FSM to check if given decimal no is divisible by 3.

Solution : $\Sigma = \{0, \ldots 9\}$

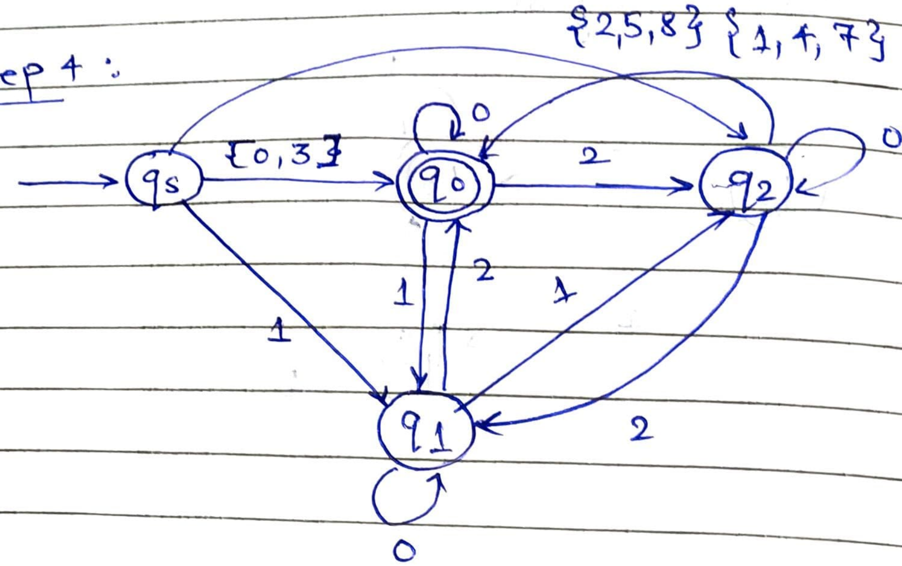
outputs : 0, 1, 2

Step 1 : $q_s \rightarrow$ start state
$q_0 \rightarrow$ remainder 0
$q_1 \rightarrow$ remainder 1
$q_2 \rightarrow$ remainder 2

Step 3 :

Q \ Σ	{0,3,6,9}	{1,4,7}	{2,5,8}
$\rightarrow q_s$	q_0	q_1	q_2
* q_0	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

Step 4 :



Step 5: Machine Function Table

Step 6 : Example (i) 1 2 1

↑

$q_1(21)$

$q_0(1)$

$= q_1$

Type VII : (Starts with)

Q1) Design a FSM which starts with 3 consequtive a's $\Sigma = \{a, b\}$

Step 1 : $q_s \rightarrow$ start state

$q_0 \rightarrow$ first a

$q_1 \rightarrow$ 2 a's

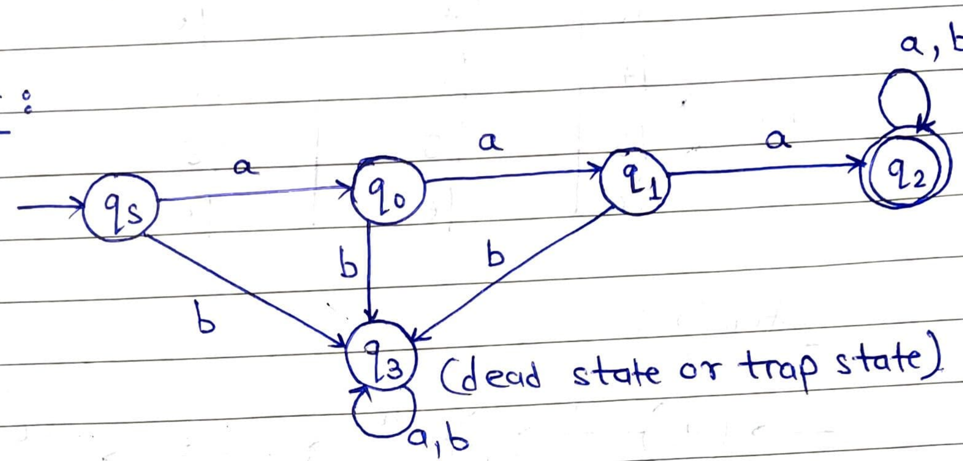
$* q_2 \rightarrow$ 3 a's

$q_3 \rightarrow$ trap state

Step3 :

Q \ Σ		a	b
	q_s	q_0	q_3
a	q_0	q_1	q_3
aa	q_1	q_2	q_3
aaa	$*q_2$	q_2	q_2
b	q_3	q_3	q_3

Step 4 :



Step 5 : MFT

Q2) Design FSM over the i/p $\Sigma = \{a, b\}$ if it starts with aab or baa

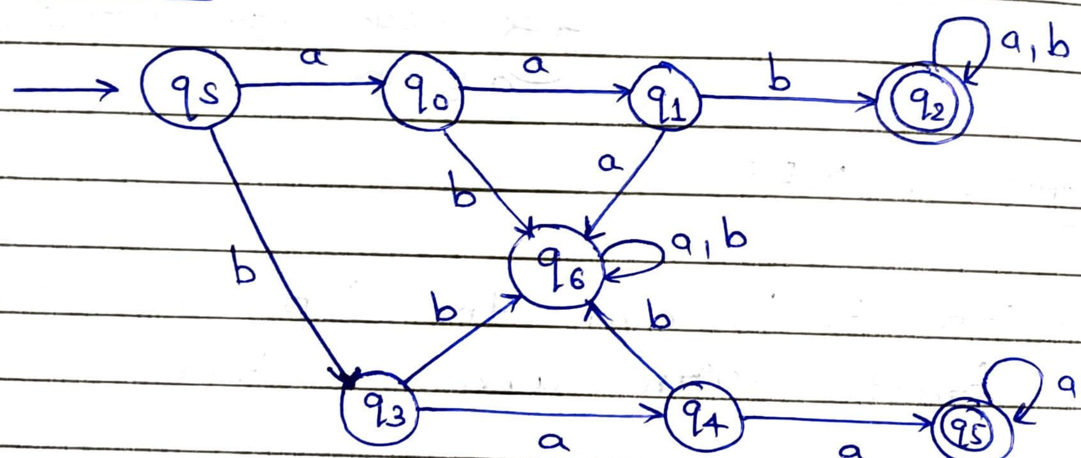
step1 :

	q_s	→	start state
a	q_0	→	starts with a
aa	q_1	→	starts with aa
aab	$*q_2$	→	starts with aab
b	q_3	→	starts with b
ba	q_4	→	starts with ba
baa	$*q_5$	→	starts with baa
	q_6	→	trap state.

Step 2 :

	Q \ Σ	a	b
→	q_s	q_0	q_3
a	q_0	q_1	q_6
aa	q_1	q_6	q_2
aab	$*q_2$	q_2	q_2
b	q_3	q_4	q_6
ba	q_4	q_5	q_6
baa	$*q_5$	q_5	q_5
	q_6	q_6	q_6

Step 3 :



VI] Divisible :

Q) Divisibility by 5

$\Sigma = \{0, \ldots 9\}$

outputs : 0, 1, 2, 3, 4

Q \ Σ	{0,5}	{1,2,3,4,6,7,8,9}
q_s	q_0	q_1
* q_0	q_0	q_1
q_1	q_0	q_1

Type VIII

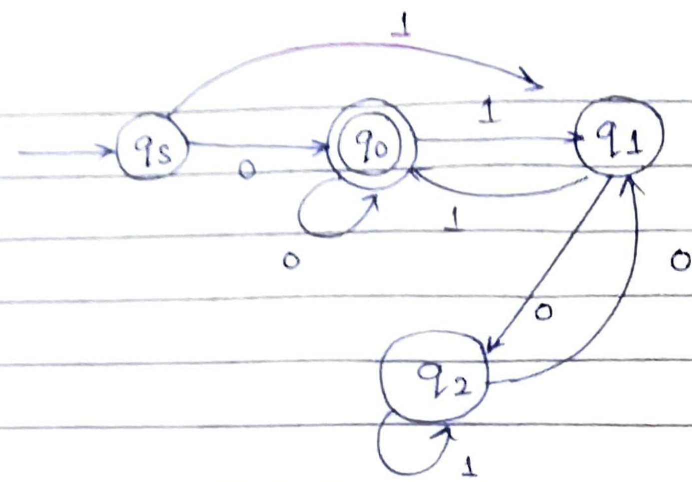
Q) Design an FSM for a binary number which is divisible by 3. $\Sigma = \{0,1\}$

Solution :

$q_s \rightarrow$ start state
$q_0 \rightarrow$ remainder 0
$q_1 \rightarrow$ remainder 1
$q_2 \rightarrow$ remainder 2

0 $\longrightarrow (2R+0)\%3$
1 $\longrightarrow (2R+1)\%3$

rem		Q \ Σ	0	1
		$\rightarrow q_s$	q_0	q_1
0	* 00	q_0	q_0	q_1
1	01	q_1	q_2	q_0
2	10	q_2	q_1	q_2

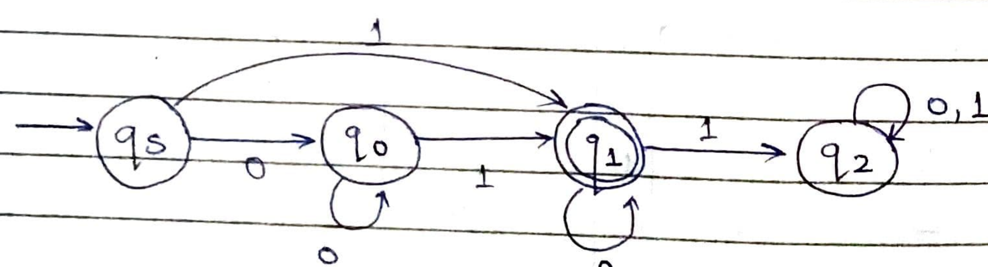


Q] Design FA that accepts set of strings with exactly 1 (one) over $\Sigma = \{0, 1\}$

Solution: $L = \{1, 01, 10, 0010, 1000, \ldots\}$

$q_s \rightarrow$ start state
$q_0 \rightarrow$ zero ones
$q_1 \rightarrow$ exactly one ~~st~~ one.
$q_2 \rightarrow$ trap state

$Q \backslash \Sigma$	0	1
$\rightarrow q_s$	q_0	q_1
q_0	q_0	q_1
$* q_1$	q_1	q_2
q_2	q_2	q_2

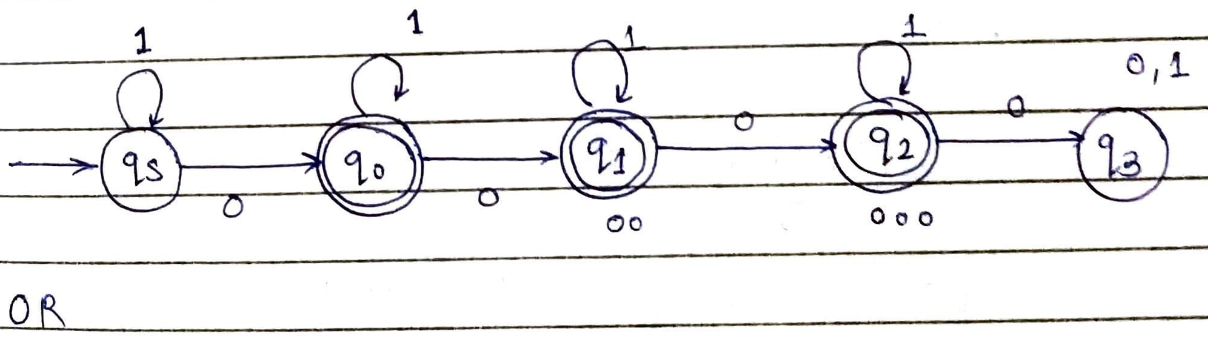


Q) Design FA for string which accepts set of all string over $\Sigma = \{0, 1\}$ with no more than 3 0's.

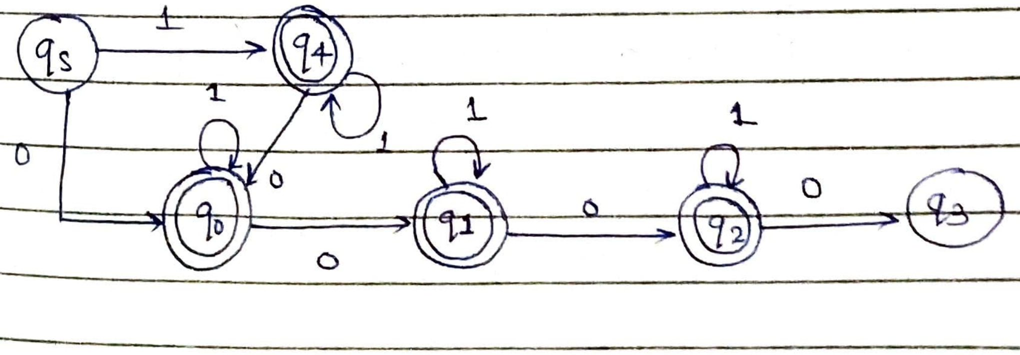
$L = \{0, 1, 00, 11, 100, 001, 1010, 1100, \ldots\}$

$q_s \rightarrow$ start state
$q_0 \rightarrow$ 1 0's
$q_1 \rightarrow$ 2 0's
$q_2 \rightarrow$ 3 0's
$q_3 \rightarrow$ trap state

Q \ Σ	0	1
$\rightarrow q_s$	q_0	q_1
$* q_0$	q_1	q_0
$* q_1$	q_2	q_1
$* q_2$	q_3	q_2
q_3	q_3	q_3



OR



Regular Expressions

R.E	to specify
ϵ	$\{\epsilon\}$
a	$\{a\}$
R|S Union	$L_R \cup L_S$
Concatenate (R).(S)	$L_R . L_S$
R	L(R)

R.E	Regular Language
R	L(R)
a+b ← a|b	$\{a\} \cup \{b\} = \{a, b\}$
a.b	$\{a\}.\{b\} = \{ab\}$
Kleen's closure ← a^*	$\{\epsilon, a, aa, aaa, \ldots\}$
positive closure ← a^+	$\{a, aa, aaa, \ldots\}$
$(ab)^*$	$\{\epsilon, ab, abab, ababab \ldots\}$
$(a+b)^*$	$\{\epsilon, a, b, aa, bb, ab, ba, bab, \ldots\}$

Q1) L_1 = the set of all strings of a's and b's ending in aa.

$(a+b)^* . aa$

$L = \{\epsilon, a, b, \ldots\} . aa.$

$L = \{aa, aaa, baa, \ldots\}$

Q2) L1 = set of strings which are starting with 0 and ending with 1.

$L = \{01, 000111, 0101, \ldots\}$

$\Downarrow$ R.E

$0(0+1)^* 1$

Q3) $L = \{\epsilon, aa, aaaa, aaaaaa, \ldots\}$

$\Downarrow$

$(aa)^*$.

1) $(A+B)^* = (A^* B^*)$

$= (A^* + B^*)^*$.

Q1) Write a regular expression for the set of strings of 0's and 1's containing 101 as substring.

$\Rightarrow (0|1)^* 101 (0|1)^*$.

Q2) Strings starting with ab and ending with ba.

$\Rightarrow ab(a+b)^* ba + aba$

Q3) Set of strings that starts with different letter over x and y.

$\Rightarrow x(x+y)^* + y(x+y)^*$

Q4) 3^{rd} char from right end of string is always a $\Sigma = \{a, b\}$

$\Rightarrow (a|b)^* a (a|b)(a|b).$

10^{th} char $\rightarrow (a|b)^* a (a|b)^9$

Q5) Find a regular expression corresponding to given sets over i/p $\Sigma = \{a, b\}$

(a) Set of all strings containing exactly 2a's
$\Rightarrow b^* a b^* a b^*$

(b) Set of all strings containing atleast 2a's
$\Rightarrow (a+b)^* a (a+b)^* a (a+b)^*$

(c) Set of all strings containing atmost 2a's
$\Rightarrow \underbrace{b^* a b^* a b^*}_{2 \rightarrow a} + \underbrace{b^* a b^*}_{1 \rightarrow a} + \underbrace{b^*}_{0 \rightarrow a}$

(d) Set of all strings containing the substring aa

$\rightarrow (a+b)^* aa (a+b)^*$

Q6) Write R.E for $\Sigma = \{0, 1\}$ whose tenth symbol from the right end is 1.

$\Rightarrow (0+1)^* 1 (0+1)^9$

Q7) Write RE for $L = \{a^n b^m \mid n \geq 4, m \leq 3\}$

$\Rightarrow$ $r = a^4 a^* (\epsilon + b + b^2 + b^3)$

Q8) Write the RE for the language.

$L = \{w \mid |w| \bmod 3 = 0, w \in (a,b)^*\}$

$RE = ((a+b)^3)^*$.

Q9) Write RE for the language $L = \{w \in (a,b)^* \mid n_a(w) \bmod 3 = 0\}$

$n_a(w) \bmod 3 = 0$ means, no of a's in a string should be 0, 3, 6, 9..

$\therefore$

$r = \{b^* a b^* a b^* a b^*\}^*$;

Q10) Find a RE consisting of all strings over $\{a,b\}$ starting with any no. of a's, followed by one or more b's, followed by one or more a's, followed by a single b, followed by any no of a's, followed by b and ending in any string of a's and b's

$\Rightarrow$ $a^* b^+ a^+ b a^* b (a+b)^*$

Q11) Write RE for the $L = \{a^n b^m \mid (n+m) \text{ is even}\}$

Soln: $(n+m)$ is even is 2 cases.

case 1: n and m both are even

case 2: n and m both are odd

let r_1 is r.e for case 1

$r_1 = (aa)^* (bb)^*$

Let r_2 is re for case 2

$r_2 = (aa)^* a (bb)^* b$.

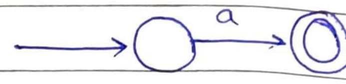
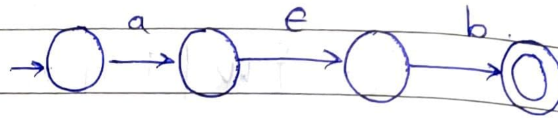
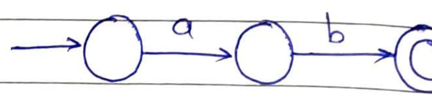
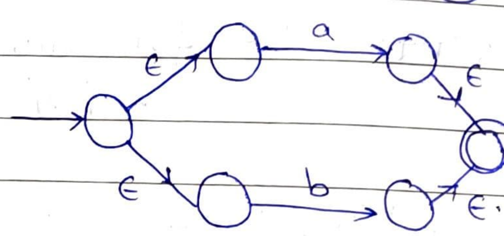
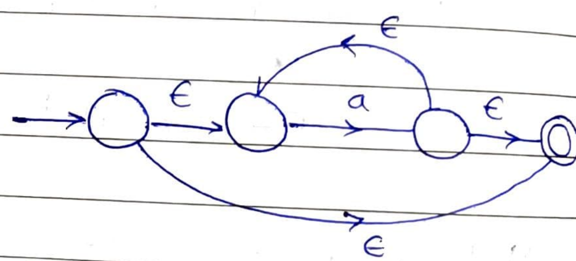
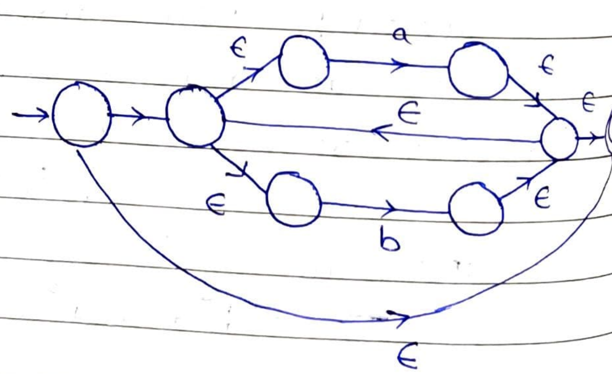
Regular Expression

$r = r_1 + r_2 = (aa)^*(bb)^* + (aa)^* a (bb)^* b$

Q 12) Write re such that it will contain no occurrence of double letter over i/p $\Sigma = \{a, b\}$

$(\epsilon + b)(ab)^*(\epsilon + a).$

Basic Rules for constructing NFA from RE (5M)

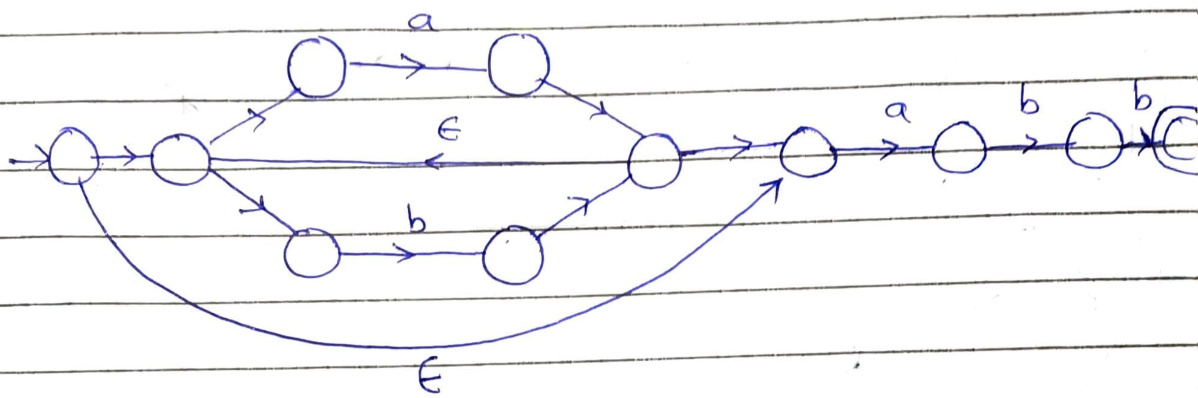
Regular Expression	NFA with ε moves
a	
a.b	 or 
a+b {a, b}	
a^* {ε, a, aa, aaa}	
$(a+b)^*$ {ε, a, b, ab, abab..}	

Q1) $(a+b)^*abb$

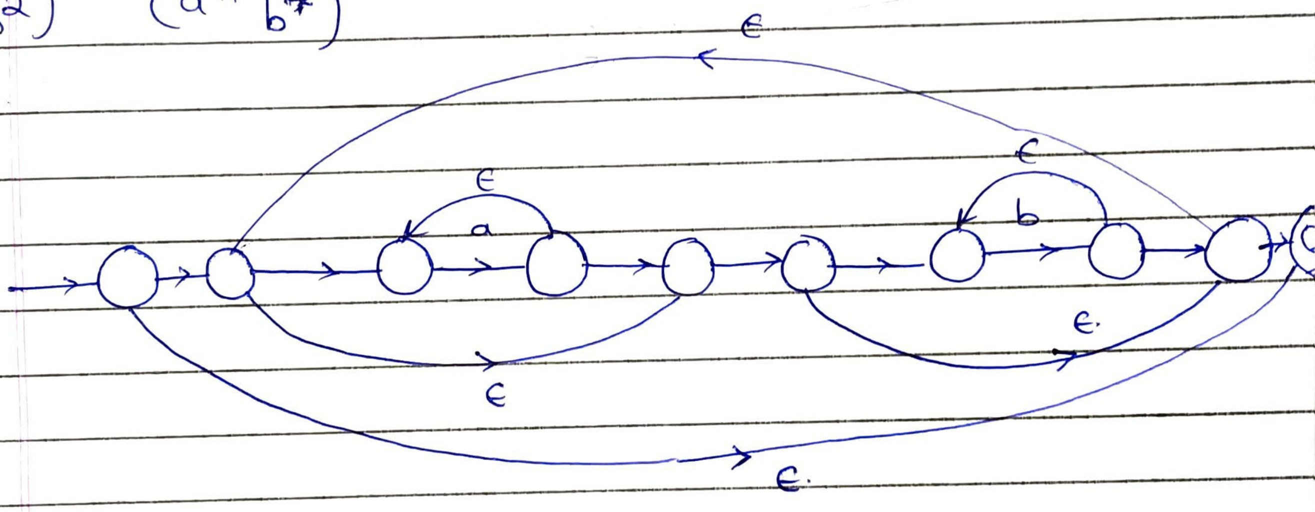
Convert the given RE into NFA

→

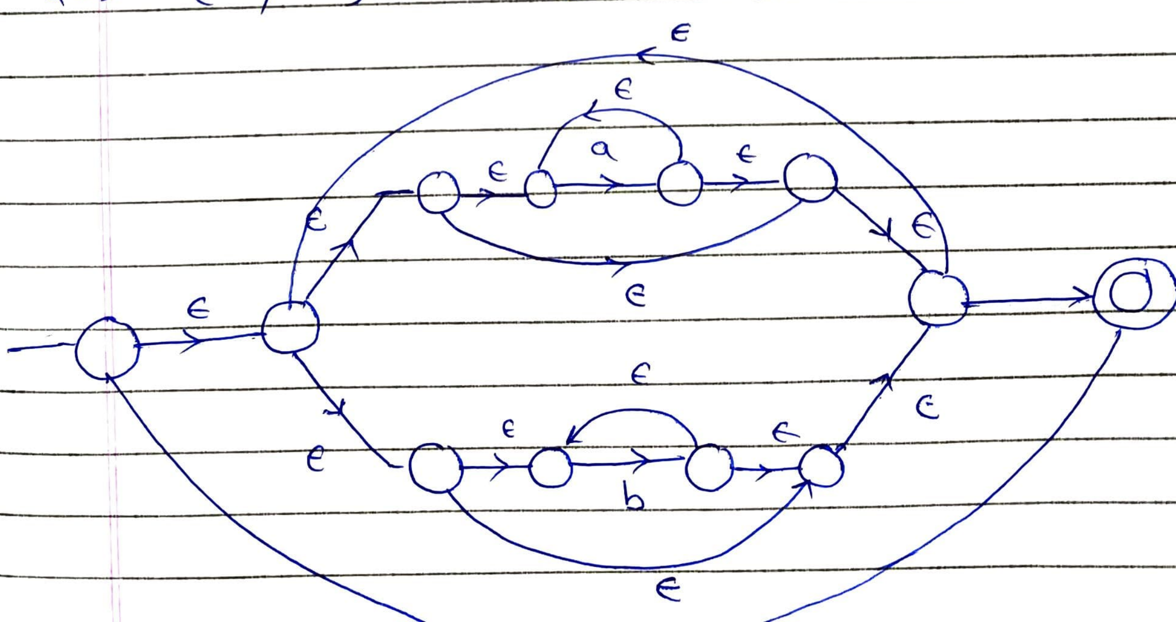
NFA is Non Deterministic Finite Automata.
It is defined by a quintuple.



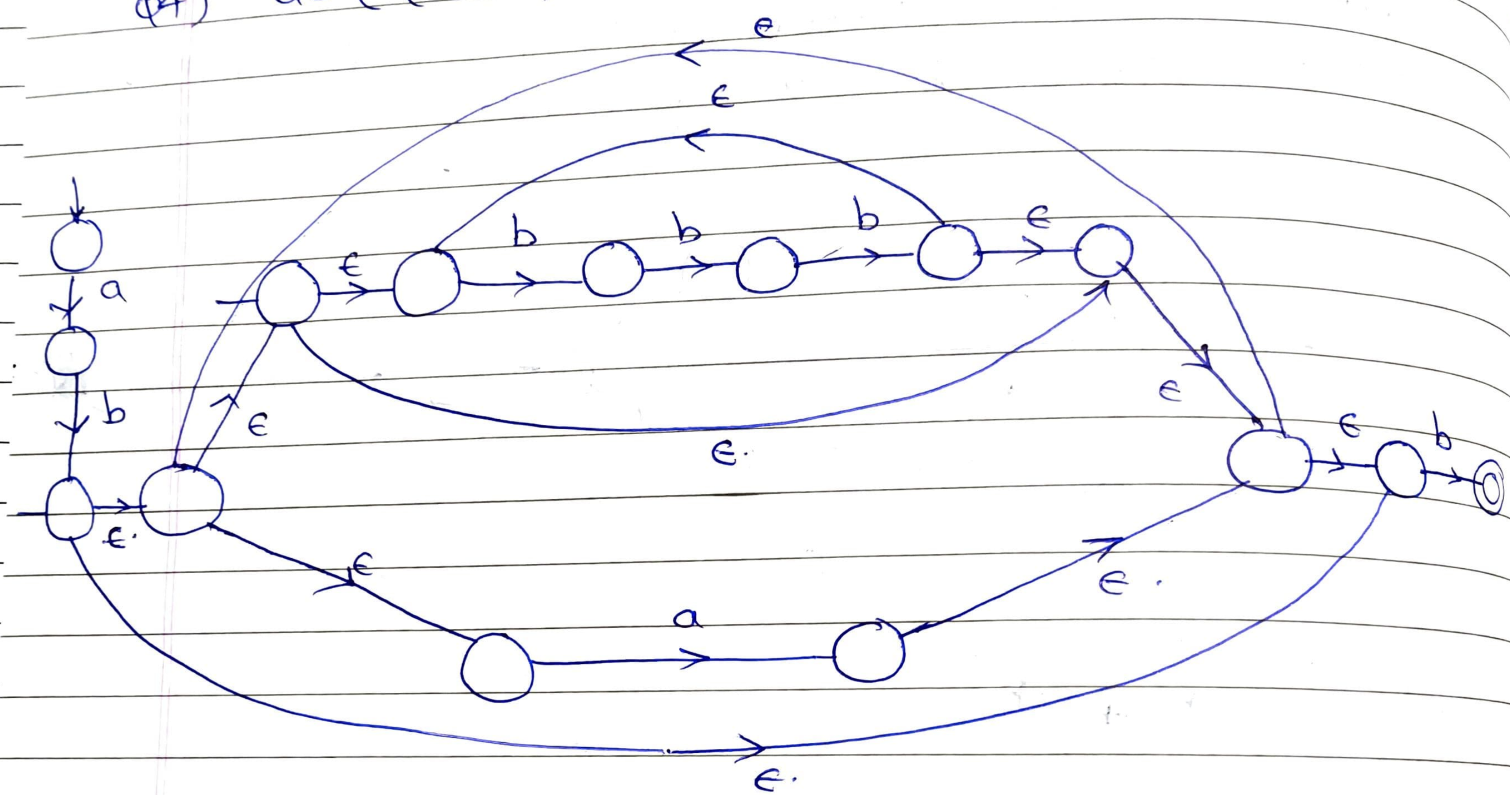
Q2) $(a^* b^*)^*$.



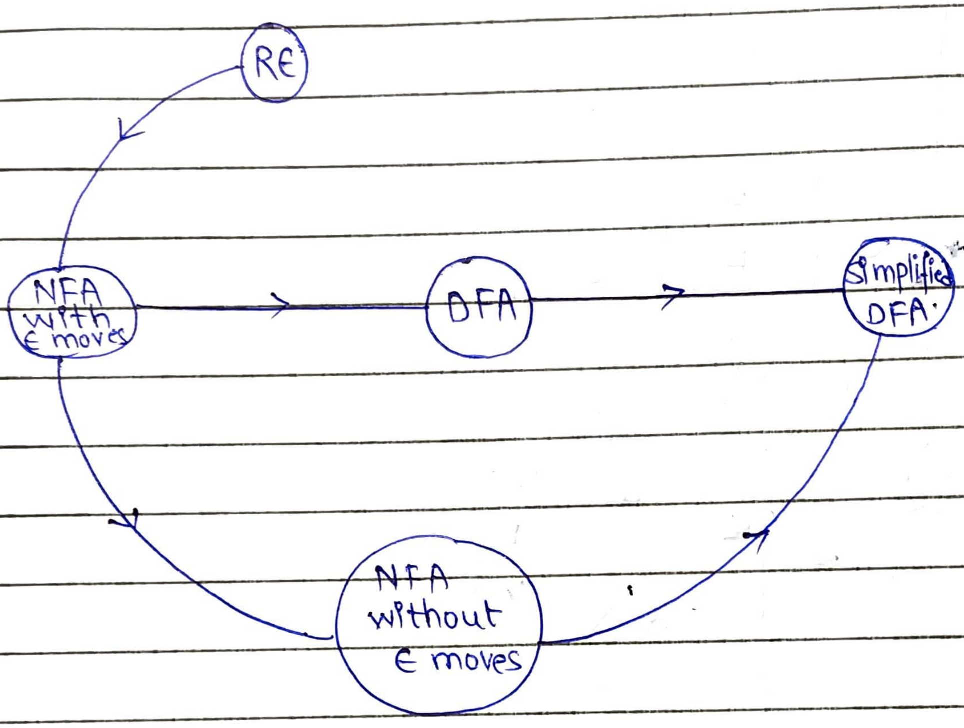
Q3) $(a^*/b^*)^* = (a^* + b^*)^*$.



Q4) $ab((bbb)^* + a)^* b$.

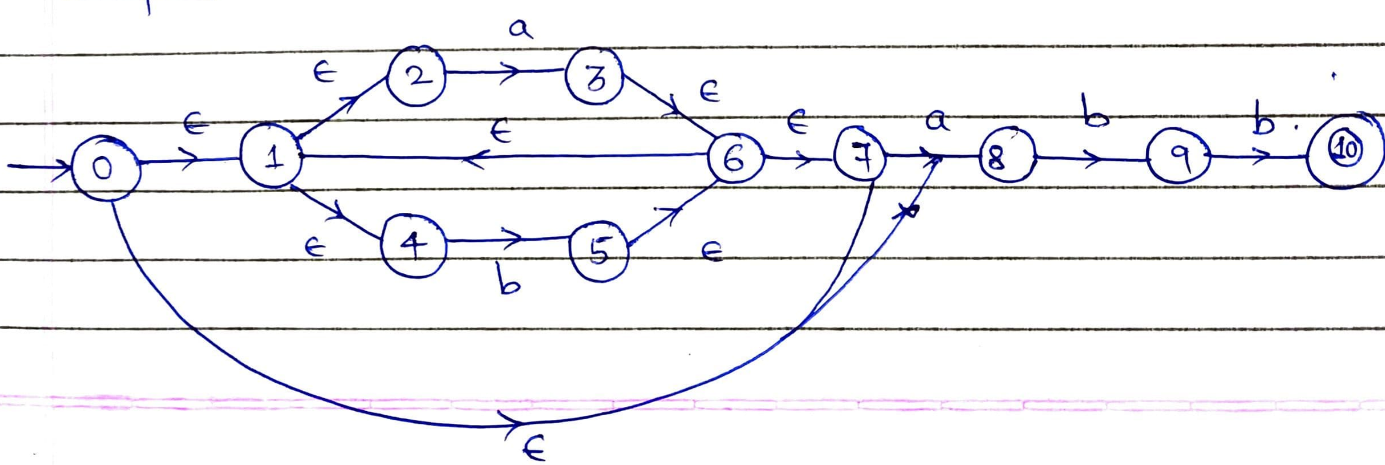


CONSTRUCTION OF DFA



Q1) construct a DFA for $(a+b)^*abb$.

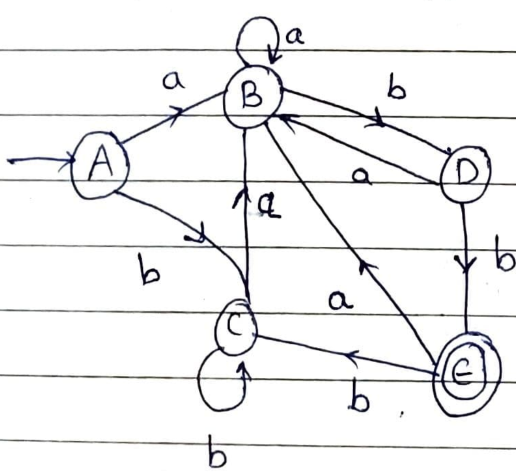
Step I] RE $\longrightarrow$ NFA with ∈ moves.



x		y = ∈ closure (x)	δ(y, a)		δ(y, b)	
A	{0}	{0, 1, 2, 4, 7}	{3, 8}	B	{5}	C
B	{3, 8}	{3, 6, 7, 1, 2, 4, 8}	{3, 8}	B	{5, 9}	D
C	{5}	{5, 6, 7, 1, 2, 4}	{8, 3}	B	{5}	C
D	{5, 9}	{5, 6, 1, 7, 2, 4, 9}	{3, 8}	B	{5, 10}	E
E	{5, 10}	{5, 6, 7, 1, 2, 4, 10}	{3, 8}	B	{5}	C

*

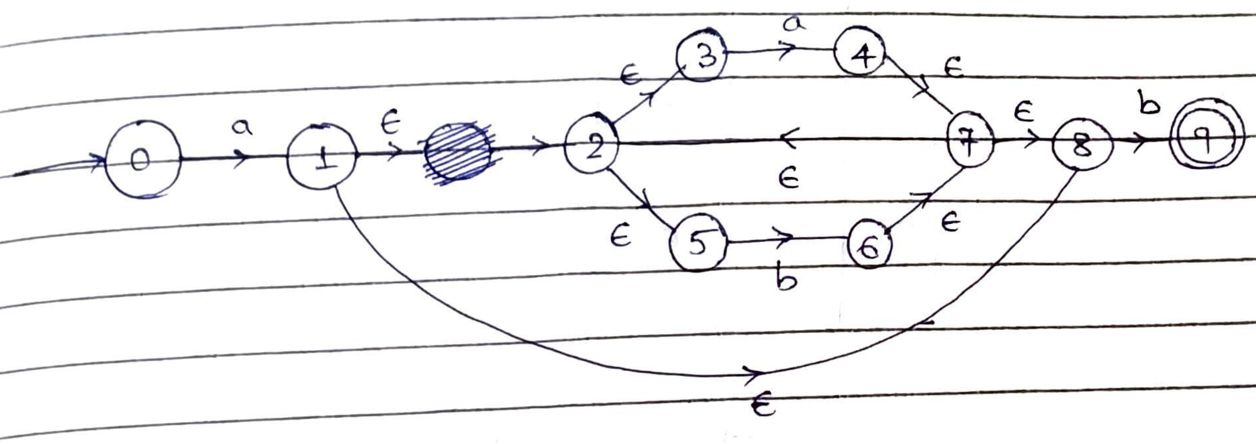
	Q \ ∈	a	b
→	A	B	C
	B	B	D
	C	B	C
	D	B	E
*	E	B	C



Q2] Construct a DFA for a string starting with a and ending with b.

Soln: SI] RE = $a(a+b)^* b$.

Step II] RE → NFA with ε moves.



Step III] Table

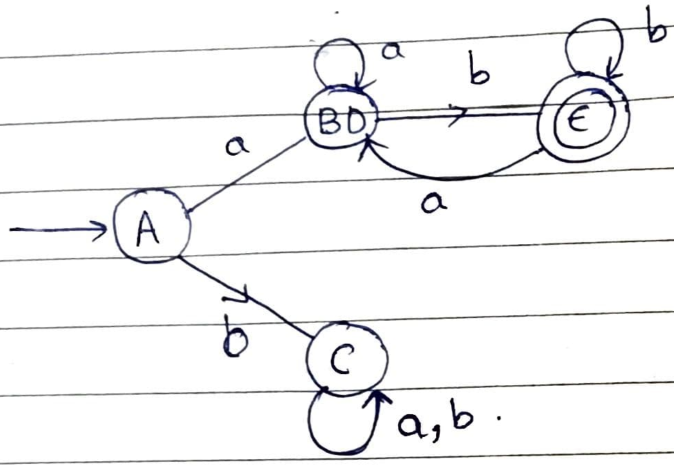
	x	$y = \epsilon\ closure(x)$	$\delta(y,a)$	$\delta(y,b)$
A	{0}	{0}	{1} B	{ } C
B	{1}	{1, 2, 3, 5, 8}	{4} D	{6, 9} E
C	{ ~~4~~ }	{ ~~7, 2, 8~~ }	{ } C	{ } C
D	{4}	{4, 7, 2, 8, 3, 5}	{4} D	{6, 9} E
E	{6, 9}	{6, 9, 7, 8, 2, 3, 5}	{4} D	{6, 9} E

Step IV]

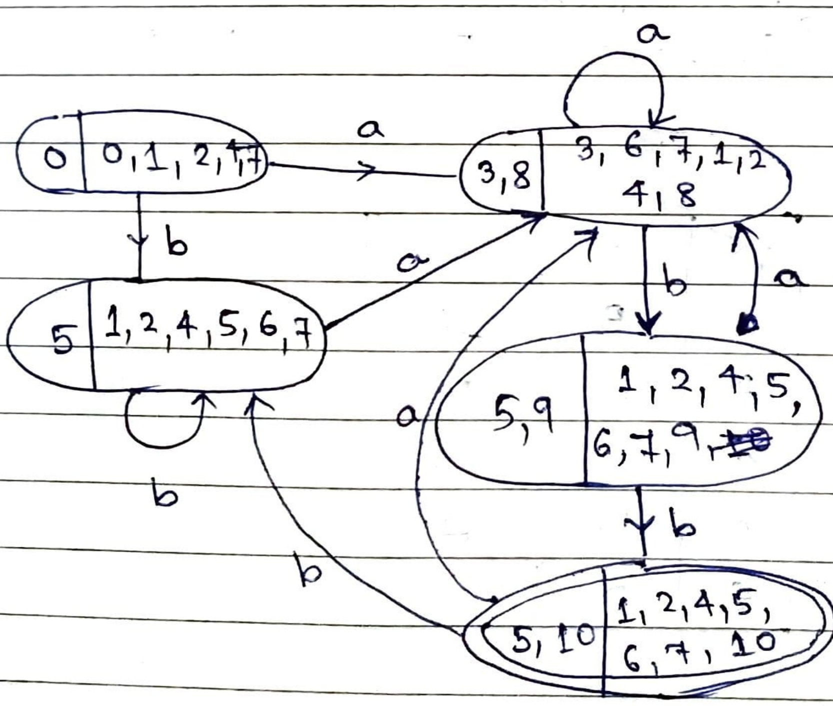
	Q \ Σ	a	b
	A	B	C
	B	D	E
	C	C	C
	D	D	E
*	E	D	E

If you have similar transitions, combine those states.

Q \ Σ	a	b
A	BD	C
BD	BD	E
C	C	C
E	BD	E



Q1) Construct DFA for $(a+b)^* abb$.



Q2) Construct DFA $\{\underbrace{\{p, q, r, s\}}_{Q}, \underbrace{\{0, 1\}}_{\Sigma}, \underset{\delta}{\delta}, \underset{q_0}{p}, \underset{f}{s}\}$

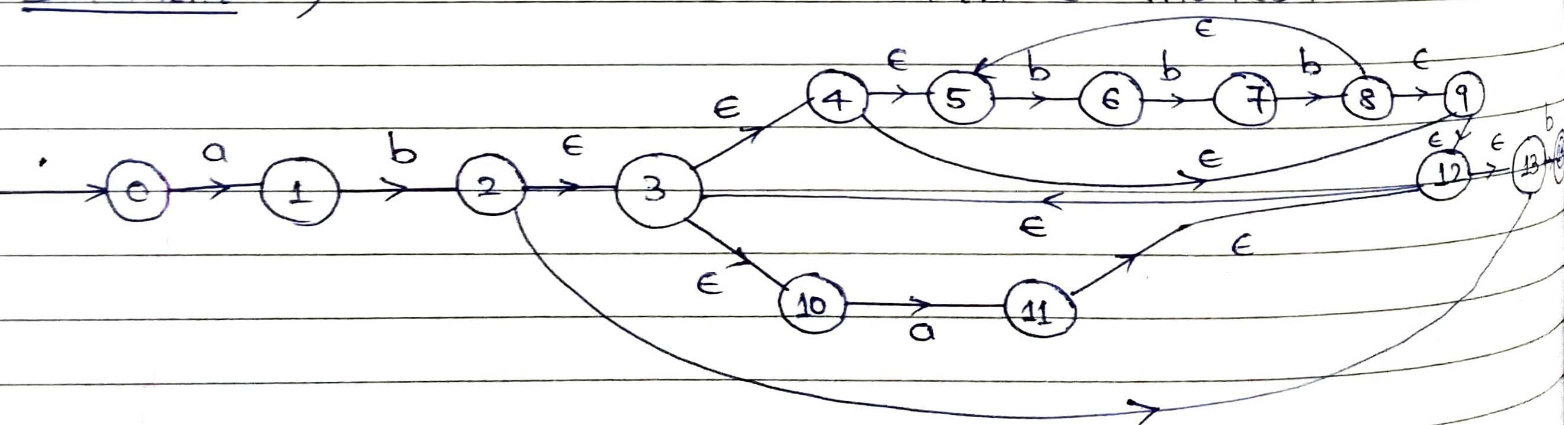
	$Q \backslash \Sigma$	0	1
$\rightarrow$	p	p, q	p
	q	r	r
	r	s	—
*	s	s	s

	0	1
A	B	A
B	C	D
C	EF	D
D	EF	A
EF*	EF	GH
GH*	EF	GH.

	0	1
→ A	B	A
B	C	D
C	EFGH	D
D	EFGH	A
* EFGH	EFGH	EFGH

Q) Construct a minimized DFA
$R = ab((bbb)^* + a)^* b.$

Solution: I) RE ⟶ NFA with ∈ moves.



Step II] Table

	x	y ε closure (x)	δ(y,a)	δ(y,b)
A	{0}	{0}	{1} B	{ } C
B	{1}	{1}	{ } C	{2} D.
C	{ }	{ }	{ } C	{ } C
D	{2}	{2,3,4,5,9,12,13,10}	{11} E	{6,14} F
E	{11}	{11,12,13,3,4,5,9,10}	{11} E	{6,14} F
F	{6,14}	{6,14}	{ } C	{7} G
G	{7}	{7}	{ } C	{8} H
H	{8}	{8,9,12,~~3~~ 10,5,4,13}	{11} E	{6,14} F

Step III]

Q \ ε	a	b
A	B	C
B	C	D
C	C	C
D	E	F
E	E	F
F	C	G
G	C	H
H	E	F

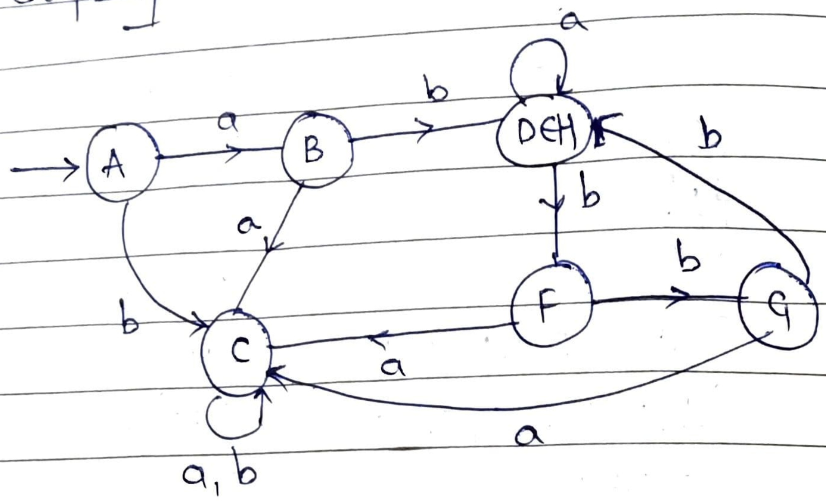
⇒

Q \ ε	a	b
A	B	C
B	C	DE
C	C	C
DE	DE	F
F	C ~~DE~~	G
G	C	H
H	C	F

⇓

Q \ ε	a	b
A	B	C
B	C	DEH
C	C	C
DEH	DEH	F
F	C	G
G	C	DEH

Step IV]

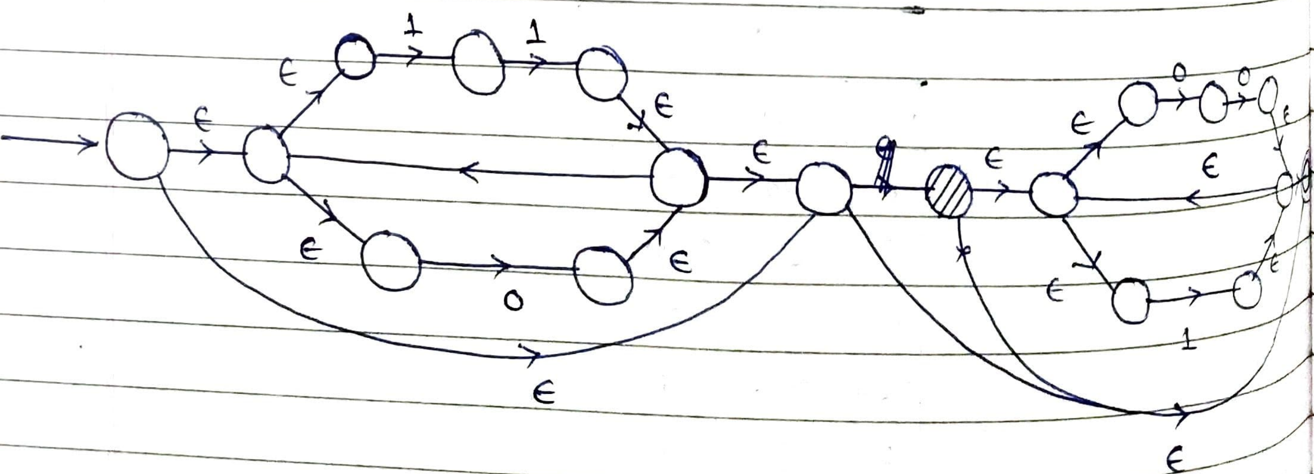


Q2) construct a NFA with ϵ moves [5M]

(i) $(11+0)^*(00+1)^*$

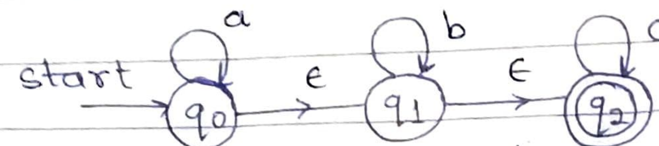
(ii) $a^* + (ab+a)^*$

(i) $(11+0)^*(00+1)^*$



★ NFA with ϵ moves to NFA without ϵ moves:

Q1)



Step I] Find the ϵ closure for q_0, q_1, q_2

Q	ϵ closure
q_0	$q_1\ q_2\ q_0$
q_1	q_1, q_2
q_2	q_2

Step II] $\delta(\{q_0,q_1,q_2\},a)$ $= \epsilon$ closure $\delta(\{q_0,q_1,q_2\},a)$

$= \epsilon$ closure $\delta(q_0,a) \cup \delta(q_1,a) \cup \delta(q_2,a)$

$= \epsilon$ closure $\delta(q_0 \cup \phi \cup \phi)$

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

similarly

$\delta(\{q_0,q_1,q_2\},b) = \epsilon$ closure $\delta(\{q_0,q_1,q_2\},b)$

$= \epsilon$ closure $\delta((q_0,b) \cup (q_1,b) \cup (q_2,b))$

$= \epsilon$ closure $\delta(\phi \cup q_1 \cup \phi)$

$= \epsilon$ closure $\delta(q_1)$

$= \{q_1,q_2\}$

$\delta(\{q_0,q_1,q_2\},c) = \epsilon$ closure $\delta(\{q_0,q_1,q_2\},c)$

$= \epsilon$ closure $\delta((q_0,c) \cup (q_1,c) \cup (q_2,c))$

$= \epsilon$ closure $\delta(\phi \cup \phi \cup q_2)$

$= \{q_2\}$

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

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

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

$\delta(\{q_2\}, a) = \epsilon \text{ closure } \delta(\{q_2\}, a)$
$= \epsilon \text{ closure } \phi$
$= \phi$

$\delta(\{q_2\}, b) = \epsilon \text{ closure } \delta(\{q_2\}, b)$
$= \phi$

$\delta(\{q_2\}, c) = \epsilon \text{ closure } \delta(\{q_2\}, c)$
$= \cancel{\phi}\ \{q_2\}$

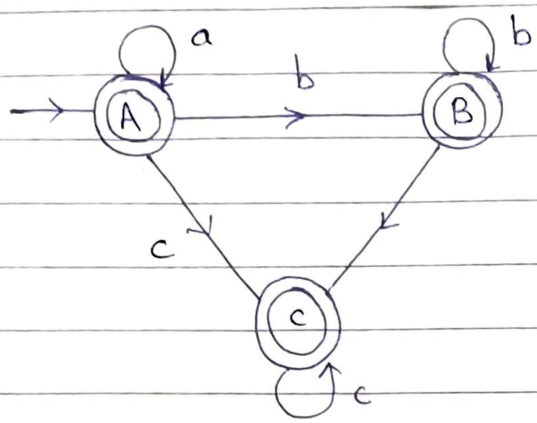
III

state \ Σ	a	b	c
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_1, q_2\}$	$\{\phi\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_2\}$	$\{\phi\}$	$\{\phi\}$	$\{q_2\}$

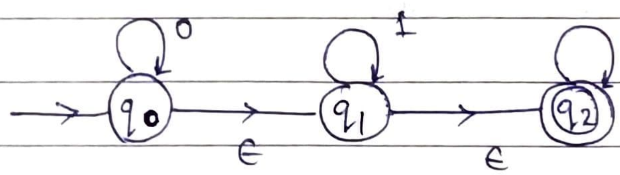
IV]

q_0, q_1, q_2

	a	b	c
A	A	B	C
B	ϕ	B	C
C	ϕ	ϕ	C



Q2)



Step I] Find the ϵ closure for q_0, q_1, q_2

Q	ϵ closure
q_0	q_0, q_1, q_2
q_1	q_1, q_2
q_2	q_2

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

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

$= \epsilon \text{ closure } \delta(q_0 \cup \phi \cup q_2)$

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

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

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

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

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

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

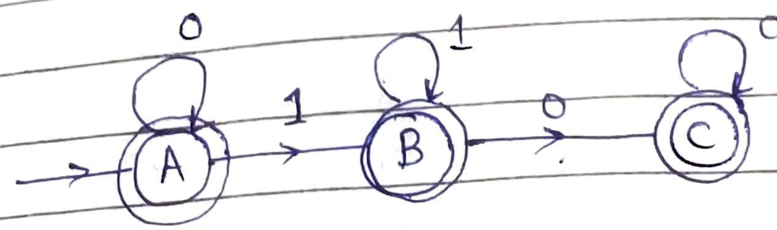
III]

State \ Σ	0	1
$\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$
$\{q_1,q_2\}$	$\{q_2\}$	$\{q_1,q_2\}$
$\{q_2\}$	$\{q_2\}$	ϕ

IV]

	0	1
A	A	B
B	C	B
C	C	ϕ

V]



Finite Automata with o/p

with o/p — without o/p

with o/p: Moore m/c, Mealy m/c.

without o/p: NFA, DFA

1) output $\longrightarrow$ present state and present i/p
$\downarrow$
Mealy m/c

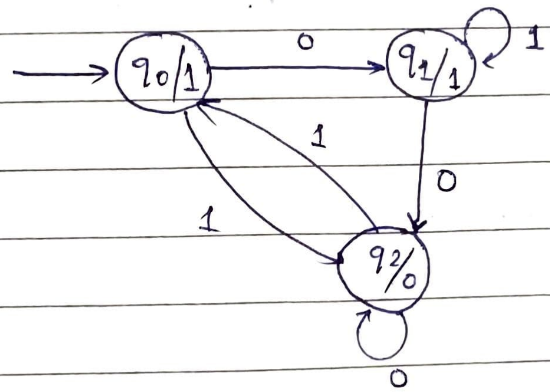
2) output $\longrightarrow$ present state
$\downarrow$
Moore m/c.

★ **Moore m/c :**

$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

- Q → total no. of states
- Σ → i/p
- Δ → o/p alphabet
- δ → transition
- λ → o/p mapping function
- q_0 → initial state

example :



NOTE: All states can be merged if all states have same transition and o/p symbol associated with same state.

Q1) Design a Moore machine to output:
i) A if it ends in 101
ii) B if it ends in 110
iii) C otherwise over $\Sigma = \{0,1\}$

Soln: I) $q_0 \rightarrow$ ending in 0
$q_1 \rightarrow$ ending in 1
$q_2 \rightarrow$ ending in 10
$q_3 \rightarrow$ ending in 101
$q_4 \rightarrow$ ending in 11
$q_5 \rightarrow$ ending in 110.

II) Transition table

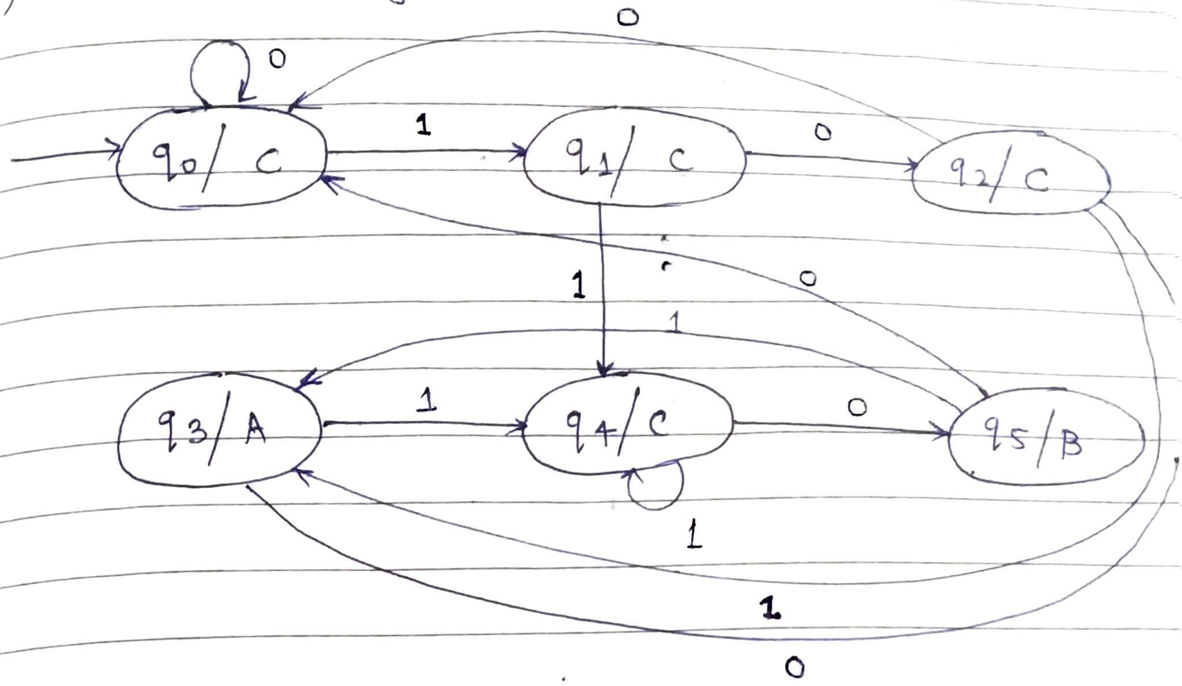
	Q \ Σ	0	1
0	q_0	q_0	q_1
1	q_1	q_2	q_4
10	q_2	q_0	q_3
* 101	q_3	q_2	q_4
11	q_4	q_5	q_4
* 110	q_5	q_0	q_3

$\lambda = \Delta$

λ	Δ
$\lambda(q_0)$	C
$\lambda(q_1)$	C
$\lambda(q_2)$	C
$\lambda(q_3) \rightarrow$	A
$\lambda(q_4)$	C
$\lambda(q_5) \rightarrow$	B

$q_5 \Rightarrow$ same as q_0
merge with q_0

IV) Transition diagram :



V) Example :

0101

CCCCA

$$\rightarrow q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3$$

Step I) $(Q, \Sigma, \lambda, \Delta, \delta, q_0)$

$\{q_0 \cdots q_5\}$

$\{0,1\}$

$\delta = Q \times \Sigma$

$\{A, B, C\}$

Q2) Design a moore m/c to output A, if i/p doesn't contain double letter otherwise o/p R over i/p {a,b}

Soln:

Ends	Q \ Σ	a	b
	q_s	q_0	q_1
a	q_0	q_2	q_1
b	q_1	q_0	q_3
	q_2	q_2	q_1
~~aa~~ bb	q_3	q_0	q_3

contains

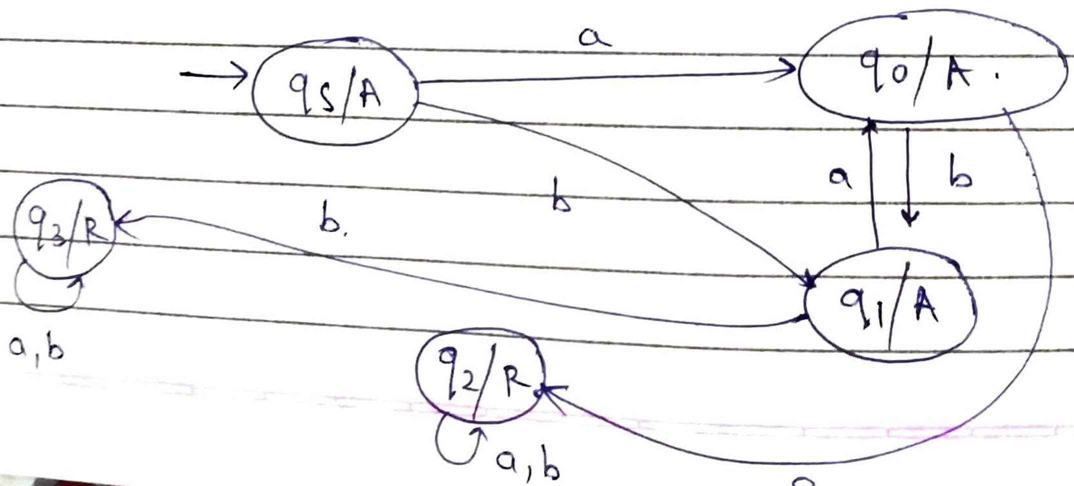
Q \ Σ	a	b
q_0	q_2	q_1
q_1	q_0	q_3
q_2	q_2	q_2
q_3	q_3	q_3

does not contain

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

$\Longrightarrow$

λ	Δ
~~$\lambda(q_s)$~~	A
$\lambda(q_0)$	A
$\lambda(q_1)$	A
$\lambda(q_2)$	R
$\lambda(q_3)$	R

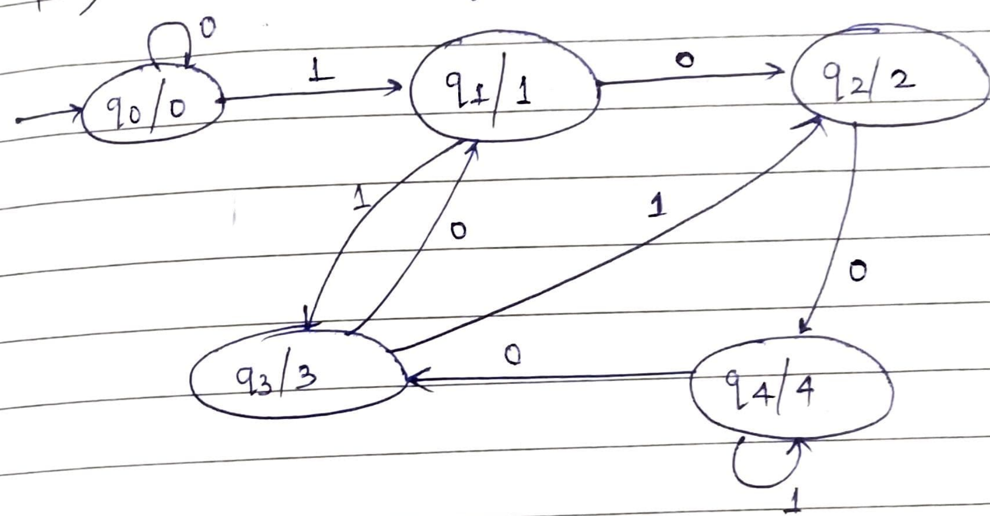


Q3) Design a Moore m/c for residue mod 5 over $\Sigma = \{0, 1\}$

Soln: Step III

	$Q \backslash \Sigma$	0	1		λ	Δ
$\rightarrow$	q_s	q_0	q_1			
0	q_0	q_0	q_1		$\lambda(q_0)$	0
01	q_1	q_2	q_3	$\Rightarrow$	$\lambda(q_1)$	1
10	q_2	q_4	q_0		$\lambda(q_2)$	2
11	q_3	q_1	q_2		$\lambda(q_3)$	3
100	q_4	q_3	q_4		$\lambda(q_4)$	4

Step 4) Transistion diag.



V) Example: 1000

1 0 0 0 i/p

q_0 q_1 q_2 q_4 q_3 states

0 1 2 4 3

* Mealy m/c :

$(Q, \Sigma, \delta, \lambda, \Delta, q_0)$

Moore
n
$n+1$

Mealy
n i/p
n o/p.

Q1) Design Mealy m/c to output A if it ends in 101 else B over $\Sigma = \{0, 1\}$

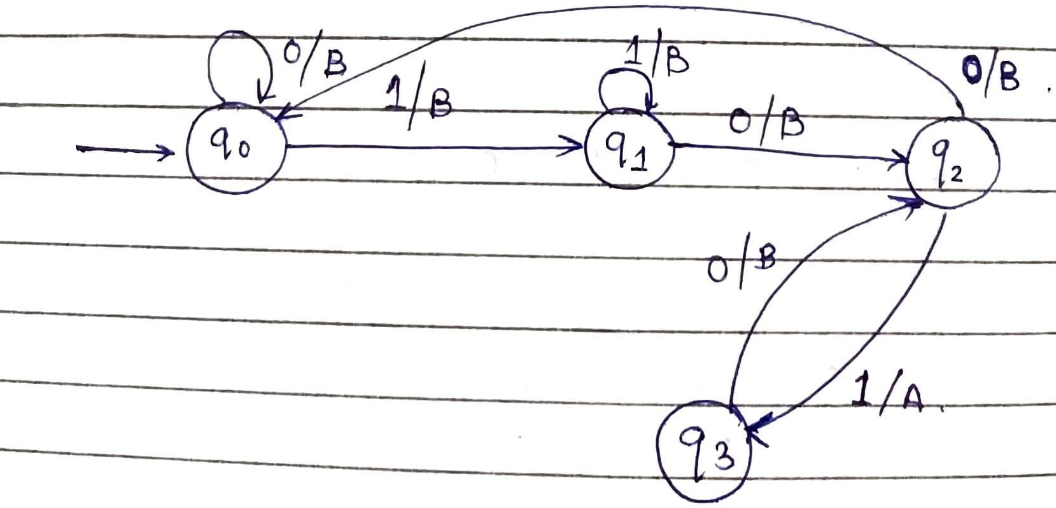
Soln: III] Transition table

$\lambda = Q \times \Sigma \rightarrow \Delta$

	Q \ Σ	0	1
	$\rightarrow q_s$	q_0	q_1
0	q_0	q_0	q_1
1	q_1	q_2	q_1
10	q_2	q_0	q_3
101	q_3	q_2	q_1

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

IV] Transition diagram :



V) Example : 0101

$\rightarrow q_0 \xrightarrow{0/B} q_0 \xrightarrow{1/B} q_1 \xrightarrow{0/B} q_2 \xrightarrow{1/A} q_3$

Q2) Design a Mealy m/c to output same characters as input except when 'i' is followed by 'e' then e should change u over $\Sigma = \{a, e, i, o, u\}$

OR

(to change each occurence of i,e to i,u)

ie → iu.

Soln:

$Q \backslash \Sigma$	a	e	i	o	u
→ q_5	q_0	q_1	q_2	q_3	q_4
a q_0	q_0	q_1	q_2	q_3	q_4
e q_1	q_0	q_1	q_2	q_3	q_4
i q_2	q_0	q_5	q_2	q_3	q_4
o q_3	q_0	q_1	q_2	q_3	q_4
u q_4	q_0	q_1	q_2	q_3	q_4
ie q_5	q_0	q_1	q_2	q_3	q_4

op table

$Q \backslash \Sigma$	a	e	i	o	u
→ q_5	a	e	i	o	u
q_0	a	e	i	..	.
:	a	e	i	..	.
q_2		$\boxed{u}$			
:	a	e	i	.	.
:	a	e	.		.
q_5	a	e	.		.

After Merging

Q \ Σ	a	e	i	o	u
q_0	a	e	i	o	u
q_2	a	u	i	o	u

Q) Design a Mealy m/c to implement binary adder.

Soln : Binary adder :

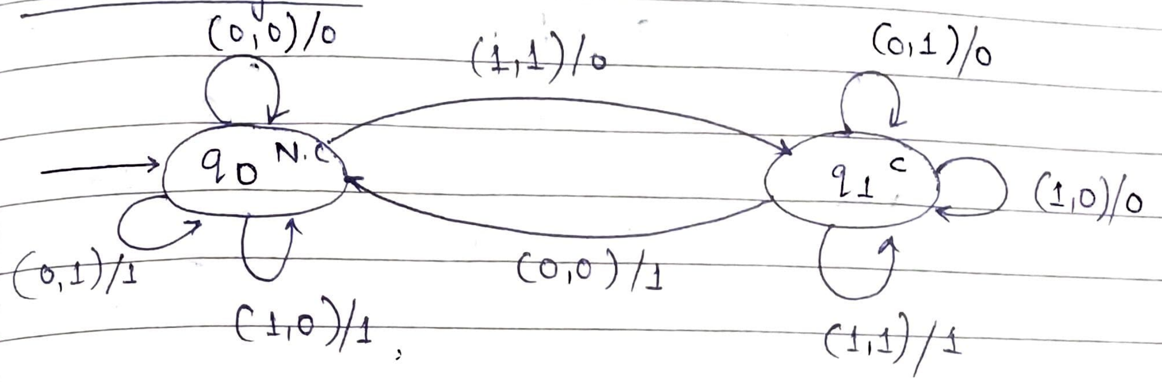
i/p 1	i/p 2	A.	carry	
0	0	0	0	N.C
0	1	1	0	
1	0	1	0	
1	1	0	1	C

	q \ Σ	(0,0)	(0,1)	(1,0)	(1,1)	
	→ q_s	q_0	q_0	q_0	q_1	merge
N.C ←	q_0	q_0	q_0	q_0	q_1	
C. ←	q_1	q_0	q_1	q_1	q_1	

δ :

Q \ Σ	(0,0)	(0,1)	(1,0)	(1,1)
NC q_0	q_0	q_0	q_0	q_1
C q_1	q_0	q_1	q_1	q_1

State diagram :



$\lambda = Q \times \Sigma \longrightarrow \Delta$

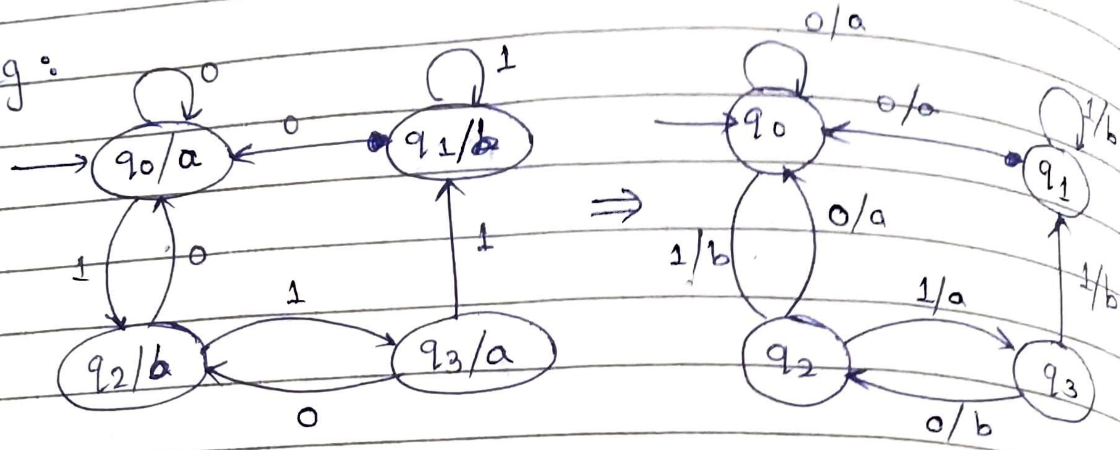
Q \ Σ	(0,0)	(0,1)	(1,0)	(1,1)
q_0	0	1	1	0
q_1	1	0	0	1

10 M

★ CONVERSION OF MOORE M/C to MEALY MACHINE M/C.

I) Assign o/p symbol associated with the state to all of its incoming transitions

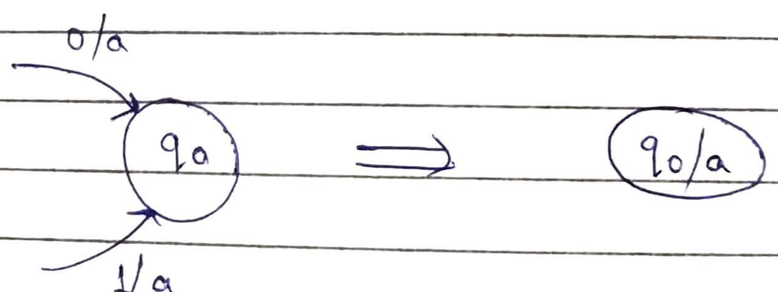
eg :



★ CONVERSION OF MEALY M/C to MOORE M/C:

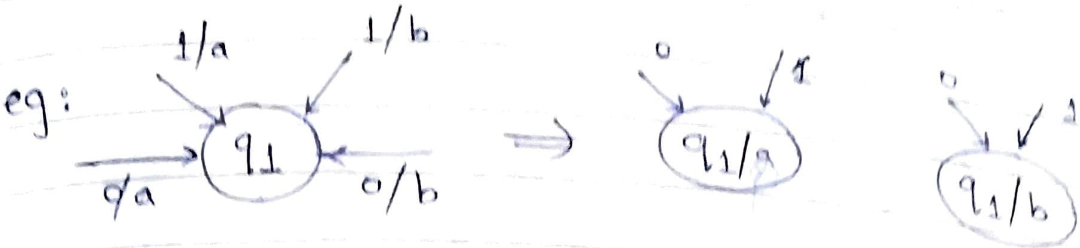
(1) If o/p symbol associated with incoming transition to a state are same then assign that symbol to that state.

eg :

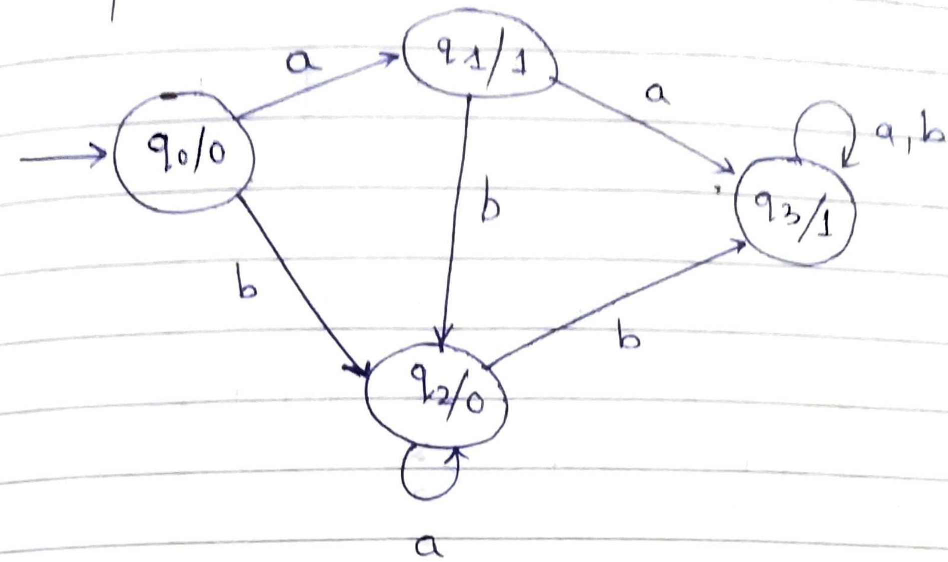


(2) If o/p symbol associated with the incoming transition are not same then split the state as many as o/p symbol with each state producing diff o/p symbol.

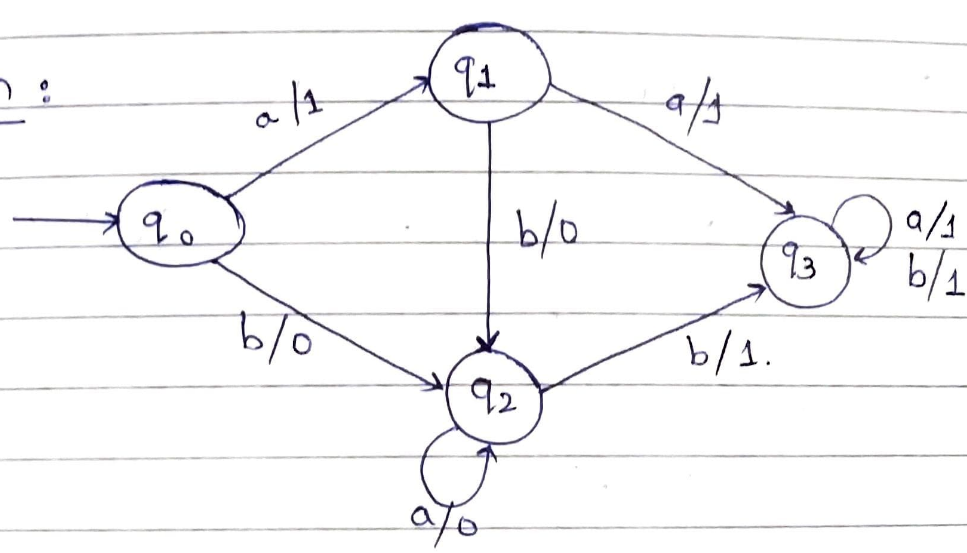
eg:



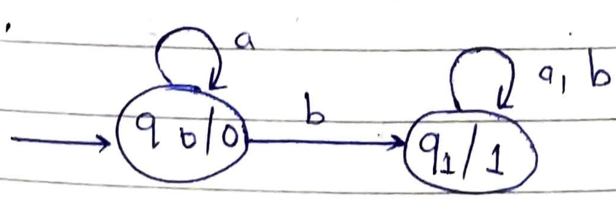
Q1) convert given moore m/c into equivalent mealy m/c.



Solution :



Q2) construct a Mealy m/c for the given moore m/c.



Solution :

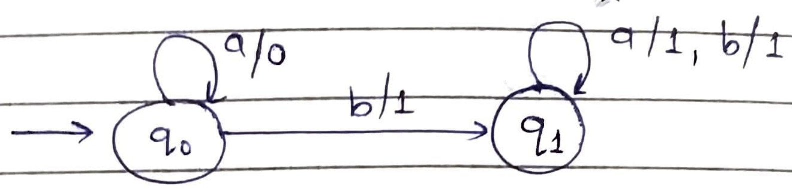
Let Moore m/c be

$M_o = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$Q = \{q_0, q_1\}$ $\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

present state	i/p a	b	o/p
→ q_0	q_0	q_1	0
q_1	q_1	q_1	1

Corresponding Mealy m/c



The required mealy m/c is

$M_e = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

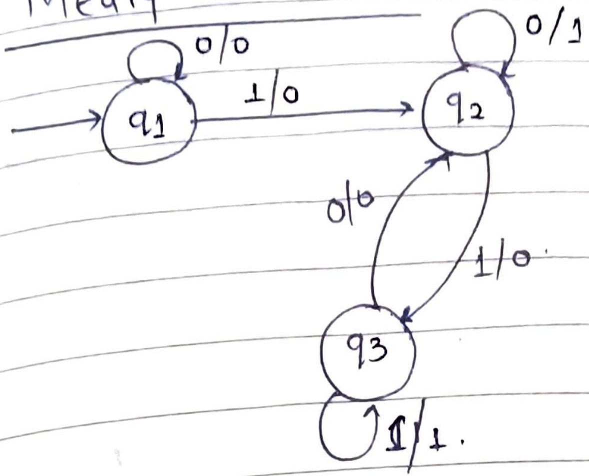
$Q = \{q_0, q_1\}$

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

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

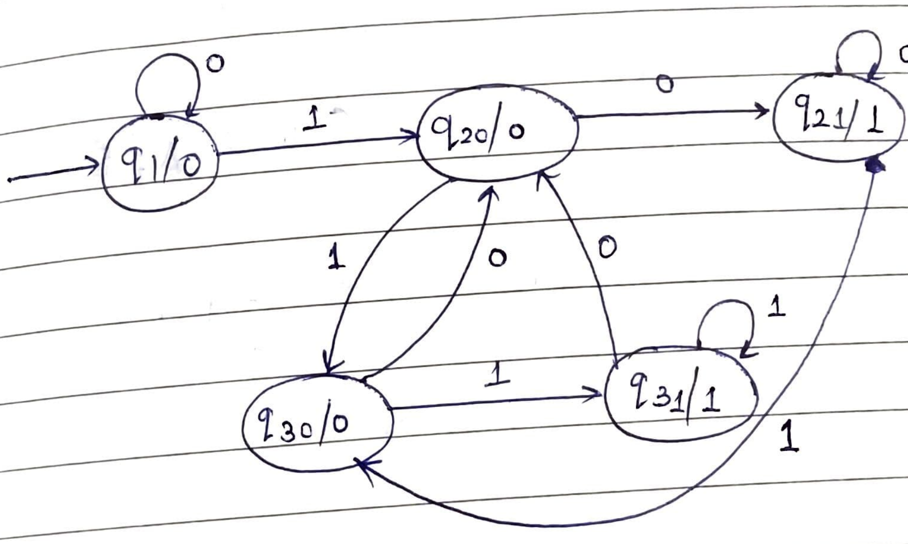
Q \ Σ	a: state	a: o/p	b: state	b: o/p
→ q_0	q_0	0	q_1	1
q_1	q_1	1	q_1	1

Q) Mealy to Moore



Soln:

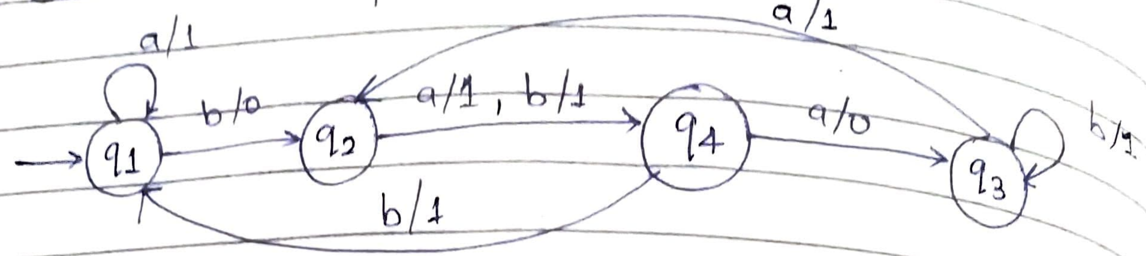
Q \ Σ	0: State	0: o/p	1: State	1: o/p
q_1	q_1	0	q_2	0
q_2	q_2	1	q_3	0
q_3	q_2	0	q_3	1



Transition table:

Q \ Σ	0	1	o/p
q_1	q_1	q_{20}	0
q_{20}	q_{21}	q_{30}	0
q_{21}	q_{21}	q_{30}	1
q_{30}	q_{20}	q_{31}	0
q_{31}	q_{20}	q_{31}	1

Q) Convert mealy to moore m/c :



For mealy machine :

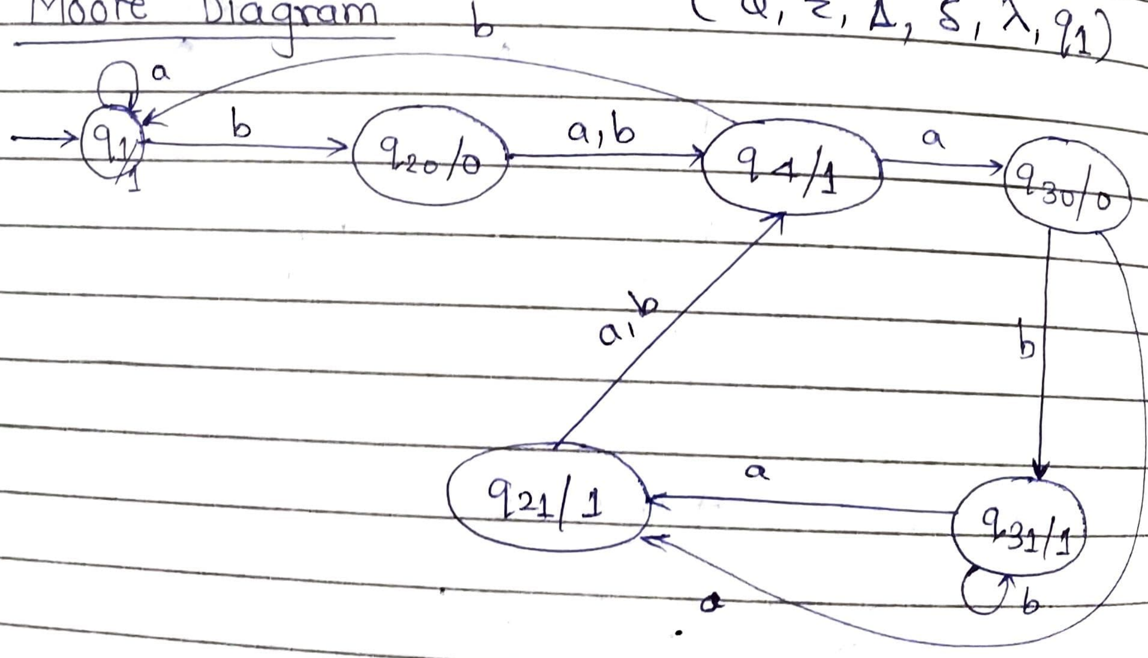
Tuple = $(Q, \Sigma, \Delta, \delta, \lambda, q_1)$

$Q = \{q_1, q_2, q_3, q_4\}$ $\Sigma = \{a, b\}$ $\Delta = \{0, 1\}$

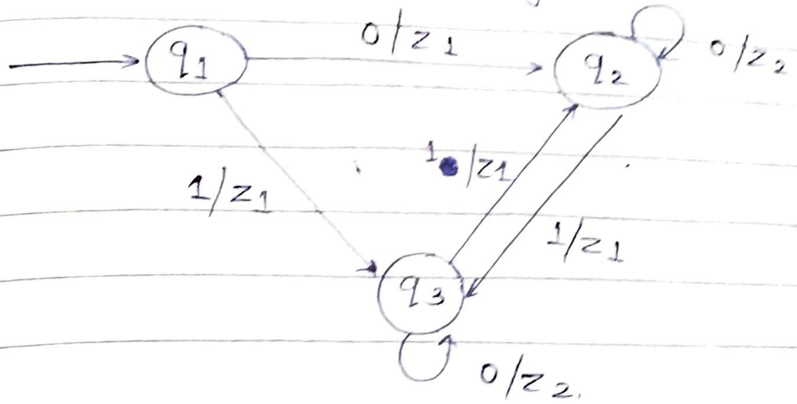
Transition table :

$Q \backslash \Sigma$	0		1	
	State	o/p	State	o/p
q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

Moore Diagram $(Q, \Sigma, \Delta, \delta, \lambda, q_1)$



Q5) ~~Eo~~ Draw corresponding moore machine.

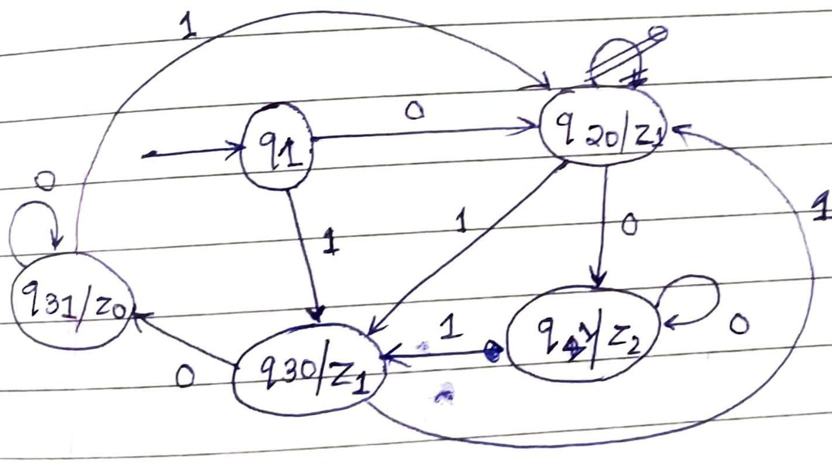


Δ = {z1, z2}

Step I)

	0		1	
	State	o/p	State	o/p
→ q_1	q_2	z_1	q_3	z_1
q_2	q_2	z_2	q_3	z_1
q_3	q_2	z_1	q_2	z_2

Moore Diagram

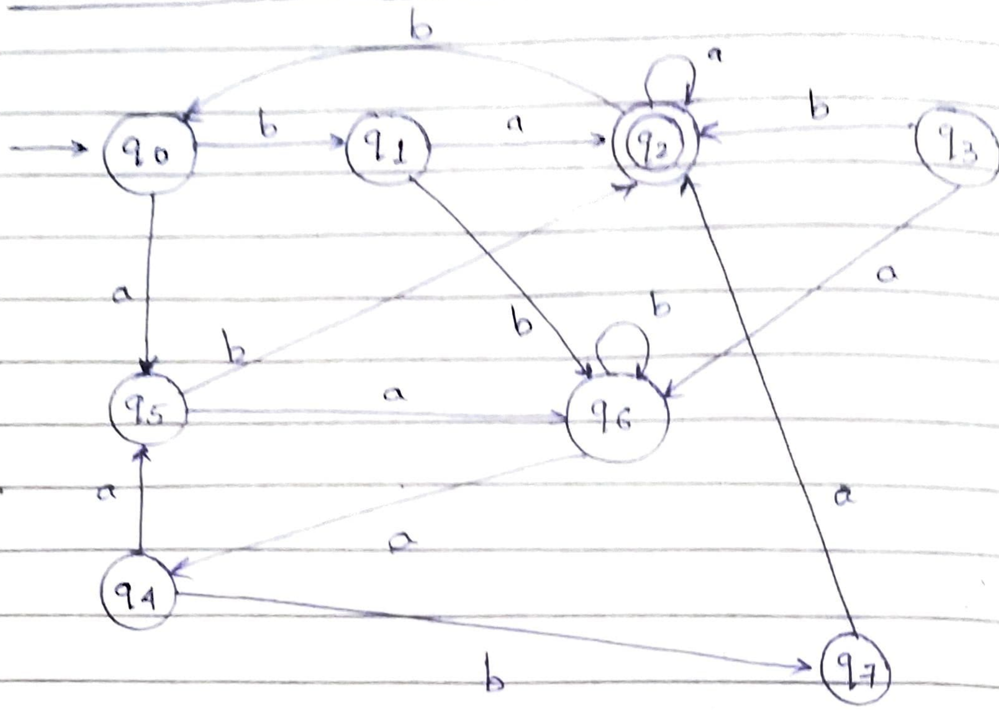


Δ = {z1, z2}

Transition table:

q \ Σ	0	1	o/p
q_1	q_{20}	q_{30}	z_1 or z_2
q_{20}	q_{21}	q_{30}	z_1
q_{21}	q_{21}	q_{30}	z_2
q_{30}	q_{31}	q_{20}	z_1
q_{31}	q_{31}	q_{20}	z_2

* DFA minimization (Alternative method):



Soln: I)

state \ i/p	a	b
→ q_0	q_5	q_1
q_1	q_2	q_6
q_2	q_2	q_0
q_3	q_6	q_2
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

Step II)

Non-final

	a	b
q_0	q_5	q_1
q_1	q_2	q_6
q_3	q_6	q_2
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

Final.

q_2	q_2	q_0

Step III]

	a	b
q_0	q_5	q_1, q_7
q_1, q_7	q_2	q_6
q_3	q_6	q_2
q_4	q_5	q_1, q_7
q_5	q_6	q_2
q_6	q_4	q_6
~~q_7~~	~~q_2~~	~~q_6~~

Final

	a	b
q_2	q_2	q_0, q_4

Step IV]

q_0, q_4	q_3, q_5	q_1, q_7
q_1, q_7	q_2	q_6
q_3, q_5	q_6	q_2
q_6	q_0, q_4	q_6

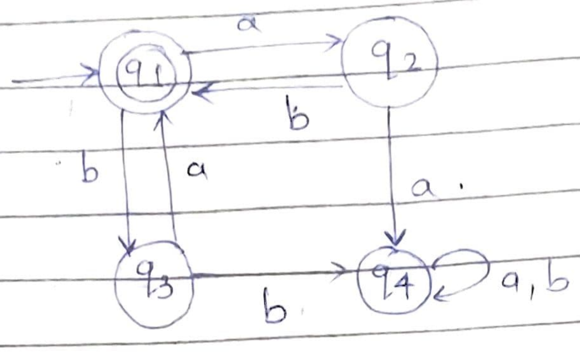
Step V]

Q \ Σ	a	b
→ $\{q_0, q_4\}$		
$\{q_1.\ \}$		
$\{q_3, q_5\}$		
$\{q_6\}$		
$\{q_2\}$		

Arden's Theorem

(No proof) (Only statement)

Q1) Find regular expression for the given DFA.



Step I)

since q_1 is the initial state.

$q_1 = q_2 b + q_3 a + \epsilon$

$q_2 = q_1 a$

$q_3 = q_1 b$

× $q_4 = q_2 a + q_3 b + q_4 a + q_4 b$

(q_4 is redudant since q_4 has not outgoing edges through which we can reach q_1).

Substitute: $q_1 = q_2 b + q_3 a + \epsilon$

$q_1 = q_1 a b + q_1 b a + \epsilon$

$q_1 = q_1 (ab + ba) + \epsilon$.

Arden's theorem :

P and Q ← RE's,

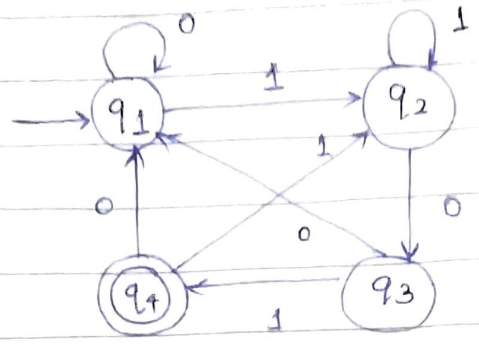
⇓

$R = Q + RP \Rightarrow R = QP^*$

$$\underset{R}{q_1} = \underset{R}{q_1} \underset{P}{(ab+ba)} + \underset{Q}{\epsilon}$$

$$\boxed{R = \epsilon (ab+ba)^*}$$

Q3) Find RE for the given DFA.



initial st...

Soln : I) $q_1 = q_1 0 + q_3 0 + q_4 0 + \epsilon$

$q_2 = q_1 1 + q_2 1 + q_4 1$

$q_3 = q_2 0$

$\boxed{q_4 = q_3 1} \longrightarrow (i)$

$q_4 = q_3 1$

$q_4 = q_2 0\,1.$

$q_4 = q_2 0\,1$

$q_2 = q_1 1 + q_2 1 + q_2 0 1 1$

$\underset{R}{q_2} = \underset{Q}{q_1 1} + \underset{R}{q_2} \underset{(P)}{(1 + 0 1 1)}$

R = Q + ...

⇓

R = Q...

$q_2 = q_1 [1 (1 + 0 1 1)^*] \longrightarrow (ii)$

$q_1 = q_1 0 + q_3 0 + q_4 0 + \epsilon$

$q_1 = q_1 0 + q_2 0 0 + q_3 1 0 + \epsilon$

$q_1 = q_1 0 + q_2 0 0 + q_2 0 1 0 + \epsilon$

$q_1 = q_1 0 + q_2 (0 0 + 0 1 0) + \epsilon$

$q_1 = q_1 0 + q_1 (1 (1 + 0 1 1)^*) (0 0 + 0 1 \ldots$
$+ \epsilon.$

$\underset{R}{q_1} = \underset{R}{q_1} \underbrace{(0 + (1 (1 + 0 1 1)^*)(0 0 + 0 1 0))}_{P} \ldots$

$$q_1 = \epsilon\,(0 + 1(1+011)^*(00+010))^*$$

$$q_4 = q_3 1 = q_2 01 = q_1(1(1+011)^*)01$$

$$q_4 = (0 + 1(1+011)^*(00+010))^*\,(1(1+011)^*)01$$

(Final state)

Theory Question Numerical – (5-8M)

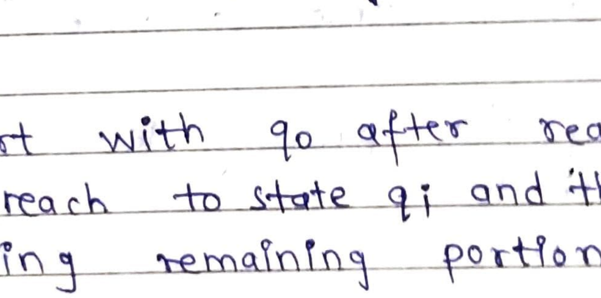
PUMPING LEMMA FOR REGULAR SET:

Let n be the regular set then there is constant n such that if z is any word in L. and $|z| \geq n$ we may write.

$z = uvw$ in such a way that $|uv| \leq n$, $|v| \geq 1$

for all $i \geq 0$ uv^iw is in L

$$z = a_1, a_2, \ldots, a_m$$



Start with q0 after reading some portion we reach to state qi and then after reading remaining portion we reach qm, final state.

Suppose DFA has n i/p states then by Pigeon hole principle $q_i = q_j$ because there cannot be n+1 states.

here $u = a_1 \ldots a_i$
$v = a_{i+1} \ldots a_j$ (no. of times pumping)
$w = a_{j+1} \ldots a_m$ such that $z = uvw$

Application of pumping lemma:

It is used to prove that a given language is not regular.

Steps:

I) Initially assume that the given language is regular.

II) $z = uvw$ where $|uv| \leq n$ and $|v| > 0$

III) Find suitable integer i such that $uv^iw \notin L$. This contradicts our assumption. Hence L is not regular.

Q) Prove that language $L = \{a^n b^n \mid n \geq 1\}$ is not regular.

Step I) Let us assume that language is regular.

Step II) According to pumping lemma there are n such that,

$z = uvw$ where $|uv| \leq n$ and $|v| \geq 0$

$a^n b^n$ $n = 4$

aaaabbbb

$u = aa$
$v = aa$
$w = bbbb$

$z = \underbrace{uv}_{\leq n} w$

$\underbrace{aaaa}_{uv}$ $w = bbbb$.

$|v| \geq 1$.

$uv^i w = aa(aa)^i bbbb$.

$i = 1$ $= aaaabbbb$ ✓

$i = 2$ $= aaaaabbbb$. ✗

It proves that the assumption is wrong hence it is not a regular expression.

Step III)

$Z = UVW = \cancel{a^S b^S}$	$= a^n b^n$	aaabbb
$UV = \cancel{a^S}$	$= a^n$	aaa
$W = \cancel{b^S}$	$= b^n$	bbb
$V = \cancel{a}$	$= a^{s}$	$3-1=2$ aa
$U = \cancel{a^{s-q}}$	$= a^{n-s}$	a
$W = \cancel{b}$	$= b^n$	bbb

$uv^i w = a^{n-s} (a^s)^i b^n$

for $i = 1$ $n = 4$
$s = 2$ $= a^{4-2}(a^2)^i b^4$
$= aaaabbbb$ ✓

for $i = 2$ $n = 4$
$s = 2$ $= a$
$= aaaaabbb$. ✗

It proves that the given language is not regular.

Q2) Prove that for $L = \{a^n b^n c^n \mid n \geq 1\}$ is not regular.

Soln] I) Let us assume that the language is regular.

II) According to the pumping lemma there are n such that

$Z = UVW$ where $|UV| \leq n$ and $|V| \geq 1$

III] $Z = UVW = a^n b^n c^n$

$UV = a^n$

$W = b^n c^n$

$U = a^{n-s}$

$V = a^s$

$W = b^n c^n$

$n=2$

aabbcc

aa

bbcc

$a^{2-1} = a$

$a^1 = a$

bbcc

$$UV^iW = (a^s)^i (a^{n-s}) b^n c^n$$

$$= a^{is+n-s} b^n c^n$$

$$= a^{n+s(-1+i)} b^n c^n$$

for $i=1$ $= a^{n+0} b^n c^n = a^n b^n c^n = Z$

for $i=2$ $= a^{n++s} b^n c^n = a^{n+s} b^n c^n \neq Z$

Hence it proves that the given language is not regular.

Q3) Prove that for $L = \{a^n b a^n \mid n \geq 1\}$ is not regular.

Soln) I) let us assume that the language is regular.

II) According to the pumping lemma there are n such that

$z = uvw$ where $|uv| \leq n$ and $|v| \geq 1$.

$n = 2$

III) $z = uvw = a^n b a^n$

$uv = a^n$

$w = ba^n$

$u = a^{n-s}$

$v = a^s$

$w = ba^n$

aabaa

aa

baa

a

a

baa

$$uv^iw = (a^s)^i (a^{n-s}) ba^n$$
$$= a^{is} a^{n-s} ba^n$$
$$= a^{is+n-s+n} b$$
$$= a^{s(i-1)+2n} b.$$

for $i = 1$ $\quad n = 2 \quad s = 1$ $\quad = a^{1(1-1)+2(2)} b$

$- a^4 b = a^2 b a^2 = z$

for $i = 2$ $\quad n = 2 \quad s = 1$ $\quad = a^{1(2-1)+4} b$

$= a^5 b \neq z$

Hence the given language is not regular.

Q4) Prove that for $L = \{a^{n!} \mid n \geq 0\}$ is not regular.

I) Let us assume that the language is not regular.

II) According to the pumping lemma there are n such that

$z = uvw$, where $|uv| \leq n$ and $|v| \geq 1$

III) $z = uvw = a^{n!} = a^{n(n-1)!}$

$uv = a^n$

$w = a^{(n-1)!}$

$u = a^{n-s}$

$v = a^s$

$w = a^{(n-1)!}$

$n=3$

$a^{3!}$

$a^{2!}$

$a^{3-1=2}$

a

$a^{(n-1)!}$

$z = u(v^i)w = a^{n-s} (a^s)^i a^{(n-1)!}$

$z = a^{n-s} a^{is} a^{(n-1)!}$

for $i=1$ $n=3$ $= a^{3-1} a^1 a^{2!}$

$s=1$ $= a^{2+1} . a^{2!} = a^3 . a^{2!} = a^{3!}$

$= z$

for $i=2$ $n=3$ $= a^{3-1} a^2 a^{2!}$

$s=1$ $= a^4 . a^{2!} \neq z.$

Hence the given language is not regular.

Q5) Prove that for $L = \{a^n b^l a^k \mid k \geq n+l\}$ is not regular.

I) Let us assume that the language is not regular.

II) According to the pumping lemma there are n such that

$z = uvw$, where $|uv| \leq n$ and $|v| \geq 1$

III) $z = uvw = a^n b^l a^k$

$uv = a^n$

$w = b^l a^k$

$u = a^{n-s}$

$v = a^s$

$w = b^l a^k$

$n = 3$

$aaab^l a^k$

$b^l a^k$

$a^{3-1} = a^2 = aa$

a

$b^l a^k$

$uv^i w = (a^{n-s})(a^s)^i b^l a^k$

$= a^{n-s} a^{is} b^l a^k$

$= a^{n-s+is} b^l a^k$

$= a^{n+s(i-1)} b^l a^k$

for $i = 1$ $\quad n = 3$ $\quad s = 1$

$= a^{3+1(1-1)} b^l a^k$

$= a^3 b^l a^k = a^n b^l a^k$

$= z$

for $i = 2$ $\quad n = 3$ $\quad s = 1$

$= a^{3+1(1)} b^l a^k$

$= a^4 b^l a^k$

$= a^4 b^l a^k \neq z$

Hence the given language is not regular.

Grammars

* Grammar is defined by quadtuple (V, T, P, S)
i.e $G = (V, T, P, S)$

where ;
V : Finite set of variables (non terminals)
T : Finite set of terminals (terminal symbols)
P : Finite set of productions [rules] that represent the language.
S : start symbol $S \in V$

Variables (Non Terminals) :

Variables are those symbols that take part in the derivation of a sentence but are not the part of derived sentence.

Terminals :

Terminals are those symbols that are the part of derived sentence.

Production rules :

Production is of the form
$\alpha \rightarrow \beta$
where α and β are in some sentential form [sequential form consists of combination of terminals and non terminals]

Start Symbol [Start Variable] :

Derivation of a sentence should start from start variable production.

Example :

- Let G be a grammar defined by a quadruple (V, T, P, S)

  V = {S, A}

  T = {0, 1}

  S = Start symbol

  P

  $S \to Ab$

  $S \to a$

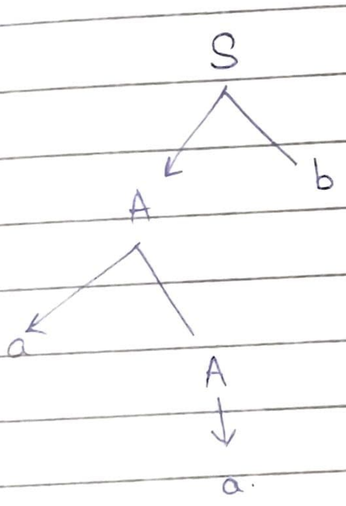
  $A \to aA$

  $A \to a$

- The above productions can be written as :

  $S \to Ab / a$

  $A \to aA / a$.



★ context free grammar (CFG) :

A grammar is a CFG if all the productions are of the form

$A \to \alpha$

where,

A → Variable (Non terminal)

α : Sentential form

Sentential form is a combination of terminals and nonterminals.

★ Derivation :

It is a process to derive a string i.e to determine whether the given grammar can derive the given sentence using any combination of production rules starting from the start variable production.

There are two types of derivations

(i) Leftmost Derivation

(ii) Rightmost Derivation

(1) Leftmost Derivation (LMD): Derivation is said to be LMD if at every step we select and replace the leftmost variable by its production rule.

(2) Rightmost Derivation (RMD): Derivation is said to be RMD if at every step we select and replace the rightmost variable by its production rule.

Q1) Let $G = (\{S\}, \{a, b, +, *\}, P, S)$ where

$P : S \rightarrow S + S / S * S / a / b$

Find (i) Leftmost derivation

(ii) Rightmost derivation for string $a + a * b$.

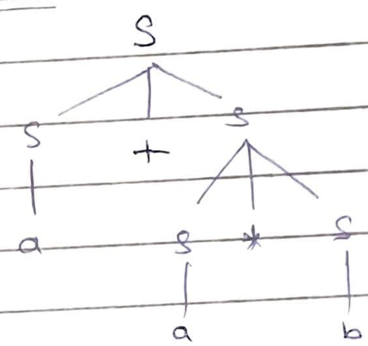
Soln : Leftmost derivation :

	Production rule
$S \to S+S$	$[S \to S+S]$
$\to a+S$	$[S \to a]$
$\to a+S*S$	$[S \to S*S]$
$\to a+a*S$	$[S \to a]$
$\to a+a*b$	$[S \to b]$

Rightmost derivation :

$S \to S+S$	$[S \to S+S]$
$\to S+S*S$	$[S \to S*S]$
$\to S+S*b$	$[S \to b]$
$\to S+a*b$	$[S \to a]$
$\to a+a*b$	$[S \to a]$

Derivation Tree :



Q2) Let G be the grammar whose productions are

$S \to 0B / 1A$

$A \to 0 / 0S / 1AA$

$B \to 1 / 1S / 0BB$

For the string '00110101' find

(1) Leftmost Derivation

(2) Right Derivation

Leftmost Derivation : Production

$S \rightarrow 0B$	
$S \rightarrow 00BB$	[$B \rightarrow 0BB$]
$S \rightarrow 001B$	[$B \rightarrow 1$]
$S \rightarrow 0011S$	[$B \rightarrow 1S$]
$S \rightarrow 00110B$	[$S \rightarrow 0B$]
$S \rightarrow 001101S$	[$B \rightarrow 1S$]
$S \rightarrow 0011010B$	[$S \rightarrow 0B$]
$S \rightarrow 001101011$	[$B \rightarrow 1$]

Rightmost Derivation : Production

$S \rightarrow 0B$	[$S \rightarrow 0B$]
$S \rightarrow 00BB$	[$B \rightarrow 0BB$]
$S \rightarrow 00B1$	[$B \rightarrow 1$]
	[

Normal Forms (of CFG) : (8-10 M)

1) Chomsky Normal Form (CNF)
2) Greibach Normal Form (GNF)

1) Chomsky Normal Form (CNF) :

In CNF, we have restriction on the length of RHS and the nature of symbols

(Notheory)

in the RHS of productions.

A CFG is in CNF if every production is of the form:

Non-terminal → string of exactly two non-terminals or

Non-terminal → one terminal.

CONVERSION TO CHOMSKY NORMAL FORM:

Step I) Simplify the given grammar G by eliminating null productions, useless variables and unit productions.

Step II) Add to the solution the productions which are already in CNF.

Step III) For the productions, not in CNF.

(a) Replace the terminals by some variables.

(b) Limit the number of variables on RHS to 2.

★ ELIMINATION OF USELESS VARIABLES:

A variable 'A' is said to be useful if

$S \Rightarrow \alpha X \beta \Rightarrow W$ (S=start symbol)

where 'W' is the sentence, $\alpha X \beta$ is in sentential form (SaAa) and 'S' is a start symbol.

The production is said to be useful if it involves useless variables.

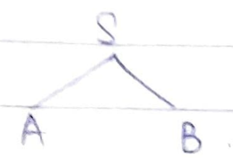
There are 2 types :
1) Non Generating
2) Non Reachable

1) Non Generating :

$S \rightarrow AB$

$C \rightarrow ab$

$D \rightarrow ef$



2) Non Reachable :

$S \rightarrow CA$

$C \rightarrow a$

$A \rightarrow bA$

The production is $\boxed{\epsilon \rightarrow e}$ Non reachable.

ELIMINATING PROCEDURE :

Given $G = (N, T, P, S)$ then define $\rightarrow$

$G' = \{ N', T', P', S') \}$ be the CFG

with no useless variables where

$L(G') = L(G)$

Step 1 : Initialize P' to P.

Step 2 : Find useless variables

a) cannot derive any sentence.

b) It is not reachable from S.

Step 3 : Eliminate all the production which involve useless variable production.

Q1) Elimate the useless variable.
$S \to aSb \mid a \mid aA$

Soln: Let $G = \{\{S, A\}, \{a, b\}, P, S\}$

P': $S \to aSb \mid a \mid aA$.
∵ A is non-deriving
Hence A is useless.
$S \to aSb \mid a$

Q2) $S \to aSbb \mid b \mid aAB$
$B \to bB \mid d$

Soln: Let $G = \{\{S, A, B\}, \{a, b\}, P, S\}$

P': $S \to aSbb \mid b \mid aAB$, $B \to bB \mid d$
∵ A is non deriving
Hence A is useless.
∴ $S \to aSbb \mid b$
$B \to bB \mid d$

Hence B is not reachable.
$\boxed{S \to aSbb \mid b}$

Q3) $S \to aSb \mid a \mid bAB$
$A \to bA \mid aB \mid c$
$B \to bB \mid dB$.

Soln: Let $G = \{\{S, A, B\}, \{a, b, c\}, P, S\}$

P': $S \to aSb \mid a \mid bAB$,
$A \to bA \mid aB \mid c$,
$B \to bB \mid dB$

$\because$ B is non ~~reachable~~ deriving hence B is useless

$\therefore S \to asb \mid a$

$A \to bA \mid c$

A is non reachable

$\therefore \boxed{S \to asb \mid a}$

✯ ELIMINATION OF UNIT PRODUCTION

A production of the form $A' \to B'$ where A' and B' are the variables is called unit production.

Given $G = (V, T, P, S)$ ~~wh~~ then $G' = (V', T', P', S')$

Let the CFG with no unit production ~~#~~ such that $L(G') = L(G)$

Step I) : Initialize P' to P.

Step II) : Find unit production.

For y : $A \to B$

Step III) : Add the unit production to A.

Step IV) : Eliminate all the unit production.

Q1) $S \to asb \mid a \mid A$

$A \to aA \mid b$.

Soln : Let $G = (V, T, P, S)$

$G' = \{ \{A, S\}, \{a, b\}, P', S' \}$.

old production	new production
$A \to b$	$A \to b$
$A \to aA$	$A \to aA$
$S \to aSb$	$S \to aSb$
$S \to a$	$S \to a$
$S \to A$	$S \to aA / b$.

P' : $A \to b / aA$

$S \to \underbrace{aSb / a / A}_{\text{old production}} / \underbrace{aA / b}_{\text{new production}}$.

Q2) $S \to aSb \mid b \mid A$

$A \to aA \mid B$

$B \to b \mid cB$

Soln : Let $G = (V, T, P, S)$

$G' = \{ (A, B, S), (a, b, c), P', S')$

old	new
$B \to b$	$B \to b$
$B \to cB$	$B \to cB$
$A \to aA$	$A \to aA$
$A \to B$	$A \to b \mid cB$
$S \to aSb$	$S \to aSb$
$S \to b$	$S \to b$
$S \to A$	$S \to aA \mid B$.

P' : $B \rightarrow b \mid cB$
$A \rightarrow aA \mid B \mid b \mid cB$
$S \rightarrow aSb \mid b \mid A \mid aA \mid B \mid b \mid cB$

Remove unit production :

$$\boxed{\begin{aligned} &B \rightarrow b \mid cB \\ &A \rightarrow aA \mid b \mid cB \\ &S \rightarrow aSb \mid b \mid aA \mid cB \end{aligned}}$$

ELIMINATION OF NULL PRODUCTION :

of variable x is nullable if $x \rightarrow \epsilon$

eliminating procedure :

Given $G = (V, T, P, S)$:
then $G' = \{V', T', P', S'\}$ be the CFG that does not contain any null production such that $\boxed{L(G') = L(G) - \epsilon}$

Step I) : Initialize P' to P.
Step II) : Find nullable variable.
Say $x \rightarrow \alpha$ is a production which contains some nullable variable then add to x the production obtained by deleting all possible subset of nullable variables from α (not present on x)

Step III) Delete Null production.
If $A \rightarrow B \mid \epsilon$ and $B \rightarrow a$. Hence 'A' is nullable substitute ϵ production to all productions of A.

Q1) $S \to aSa \mid bSb \mid \epsilon$

Soln : Step 1] Let $G = (VTPS)$
$G' = \{\{S\}, \{a, b), P', S'\}$

Step 2]

old	new
$S \to aSa$	$S \to aSa$
$S \to bSb$	$S \to bSa$
$S \to \epsilon$	$S \to aa$
	$S \to bb$

Q2) $S \to a \mid Xb \mid aYa$
$X \to Y \mid \epsilon$
$Y \to b \mid X$

old	new	
$Y \to b$	$Y \to b$	
$Y \to X$.	$Y \to X$	$Y \to b \mid \epsilon$
$Y \to \epsilon$	$Y \to \epsilon$	
$X \to Y$	$X \to Y$	$X \to b \mid \epsilon$
$X \to \epsilon$	$X \to \epsilon$	
$S \to a$	$S \to a$	
$S \to Xb$	$S \to b$	
$S \to aYa$	$S \to aa$.	

$Y \Rightarrow b \mid X \mid \cancel{\epsilon}$
$X \to \cancel{Y} \mid \cancel{\epsilon} \mid b$
$S \to a \mid Xb \mid b \mid aYa \mid aa$

$S \rightarrow a \mid Xb \mid aYa \mid b \mid aa$

$X \rightarrow b$

$Y \rightarrow b.$

conversion :

Q1) $S \rightarrow aB \mid bA$

$A \rightarrow a \mid aS \mid bAA$

$B \rightarrow b \mid bSA \mid aBBB$

Find the grammar in CNF equivalent to

Soln : StepI) The grammar is already simplified as there is no unit production, ϵ production and no useless variable.

StepII) old	new	
$A \rightarrow a$	$A \rightarrow a$	
$B \rightarrow b$	$B \rightarrow b$	
$S \rightarrow aB$	$S \rightarrow \cancel{C_bA}\ aB$	$(C_a \rightarrow a)$
$S \rightarrow bA$	$S \rightarrow C_bA$	$(C_b \rightarrow b)$
$A \rightarrow aS$	$A \rightarrow C_aS$	$(\because C_a \rightarrow a)$
$A \rightarrow bAA$	$A \rightarrow C_bAA$	
	$\rightarrow C_1 A$	$(\because C_bA \rightarrow C_1)$
$B \rightarrow bSA$	$B \rightarrow C_bSA$	
	$\rightarrow C_2 A$	$(\because C_bS \rightarrow C_2)$
$B \rightarrow aBBB$	$B \rightarrow C_aBBB$	
	$\rightarrow C_3 \cancel{C_4} BB$	$C_aB \rightarrow C_3$
	$\rightarrow C_3C_4$	$BB \rightarrow C_4$

So, all the productions in the new production are in CNF form. Hence

the required grammar is
(V, T, P, S)

$= \{\{S, A, B, C_a, C_b, C_1, C_2, C_3, C_4\}, \{a, b\}, P, S\}$

$$\boxed{\begin{array}{lll} S \to C_a B \mid C_b A & & \\ A \to a \mid C_a S \mid C_1 A & & \\ B \to b \mid C_2 A \mid C_3 C_4 & & \\ C_a \to a & C_1 \to C_b A & C_3 \to C_a B \\ C_b \to b & C_2 \to C_b S & C_4 \to BB \end{array}}$$

Q2) $A \to aBa \mid bBb$
$B \to aB \mid bB \mid \epsilon$.

Step I) Remove NULL production

old	new
$A \to aBa$	$A \to$ ~~aC_ba~~ ~~$(C_b \to B)$~~

Q3) $S \rightarrow bA \mid aB$
$A \rightarrow bAA \mid aS \mid a$
$B \rightarrow aBB \mid bS \mid b.$

Step I) The grammar is already simplified as there is no unit production, ϵ production and no useless variable.

Step II)

old	new	
$S \rightarrow bA$	$S \rightarrow C_b A$	$(C_b \rightarrow$
$S \rightarrow aB$	$S \rightarrow C_a B$	$(C_a \rightarrow$
$A \rightarrow bAA$	$A \rightarrow C_b AA$	$(C_b \rightarrow$
	$A \rightarrow C_b D$	$D \rightarrow$
$A \rightarrow aS$	$A \rightarrow C_a S$	
$B \rightarrow aBB$	$B \rightarrow C_a BB$	
$B \rightarrow bS$	$B \rightarrow C_b S$	
$B \rightarrow b$	$B \rightarrow b$	
$A \rightarrow a$	$A \rightarrow a.$	
$B \rightarrow aBB$	$B \rightarrow C_a E$	$(BB \rightarrow$

$S \rightarrow C_b A \mid C_a B$
$A \rightarrow C_b D \mid C_a S \mid a$
$B \rightarrow C_a E \mid C_b S \mid b$
$C_a \rightarrow a$
$C_b \rightarrow b$
$D \rightarrow AA$
$E \rightarrow BB$

So all the productions in the new production are in CNF form. Hence the required grammar is (V, T, P, S)

$= \{\{S, A, B, C_a, C_b, D, E\}, \{a, b\}, P, S\}$

Q4) $S \rightarrow aAbB$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid b$.

Soln : Step I) The grammar is already simplified as there is no unit production, ϵ production and no useless variable.

Step II)

old	new	
$S \rightarrow aAbB$	$S \rightarrow C_a A C_b B$	$(C_a \rightarrow a)$ $(C_b \rightarrow b)$
$A \rightarrow aA$	$A \rightarrow C_a A$	
$A \rightarrow a$	$A \rightarrow a$	
$B \rightarrow bB$	$B \rightarrow C_b B$	
$B \rightarrow b$	$B \rightarrow b$	
$S \rightarrow aAbB$	$S \rightarrow C_a D B$ $S \rightarrow C_a E$	$(D \rightarrow A C_b)$ $(E \rightarrow DB)$

CNF :

$S \rightarrow C_a C$
$A \rightarrow C_a A \mid a$
$B \rightarrow C_b B \mid b$
$C_a \rightarrow a$
$C_b \rightarrow b$
$D \rightarrow A C_b$
$C \rightarrow DB$

So all the productions in the new production are in CNF form. Hence the required grammar is (V, T, P, S)

$= \{ \{S, A, B, C_a, C_b, D, C\}, \{a, b\}, P, S\}$

Q5) $S \rightarrow AB \mid aB$
$A \rightarrow aab \mid \epsilon$
$B \rightarrow bbA$

Soln : step I) Eliminate ϵ production :

$S \rightarrow AB \mid aB \mid B$
$A \rightarrow aab$
$B \rightarrow bbA \mid bb$

Eliminate unit production :
$S \rightarrow AB \mid aB \mid bbA \mid bb$
$A \rightarrow aab$
$B \rightarrow bbA \mid bb$
Check if in CNF or not

old	new	
$S \to AB$	$S \to AB$	
$S \to aB$	$S \to \cancel{aC_b}\ C_aB$	$(C_a \to a)$
$S \to bbA$	$S \to C_bC_bA$	$(C_b \to b)$
$S \to bbA$	$S \to C_bD$	$(C_bA \to D)$
$S \to bb$	$S \to C_bC_b$	
$A \to aab$	$A \to C_aC_aC_b$	
$A \to aab$	$A \to C_aE$	$(E \to C_aC_b)$
$B \to bbA$	$B \to C_bC_bA$	$(C_b \to b)$
	$B \to C_bD$	
$B \to bb$	$B \to C_bC_b$	

<u>CNF</u> :

$$\boxed{\begin{array}{l} S \to AB \mid C_aB \mid C_bD \mid C_bC_b \\ A \to C_aE \\ B \to C_bD \mid C_bC_b \\ C_a \to a \\ C_b \to b \\ D \to C_bA \\ E \to C_aC_b \end{array}}$$

So all the productions in the new production are in CNF form. Hence the required grammar is (V, T, P, S)

$= \{\{S, A, B, C_a, C_b, D, E\}, \{a, b\}, P, S\}$.

☆ GNF (Greibach Normal Form)

① eliminate unit / useless and ϵ productions

② $A \to a\alpha$

↳ string of non-terminals (possible to be empty)

Q1) $S \to ab \mid aS \mid aaS$

old	new	
$S \to ab$	$S \to aC_b$	$(C_b \to b)$
$S \to aS$	$S \to aS$	
$S \to aaS$	$S \to aC_aS$	$(C_a \to a)$

GNF :

$$\boxed{\begin{array}{l} S \to aC_b \mid aS \mid aC_aS \\ C_a \to a \\ C_b \to b \end{array}}$$

Q2)

$S \to abaSa \mid aba$

old	new	
$S \to abaSa$	$S \to abASA$	$(A \to a)$
	$S \to aBASA$	$(B \to b)$
$S \to aba$	$S \to abA$	$(A \to a)$
	$S \to aBA$	$(B \to b)$

GNF :

$$\boxed{\begin{array}{l} S \to aBASA \mid aBA \\ A \to a \\ B \to b \end{array}}$$

Q3) $S \to AB$
$A \to BS \mid a$
$B \to SA \mid b$

soln :

$$\begin{array}{ccc} A_1 & A_2 & A_3 \\ \downarrow & \downarrow & \downarrow \\ S & A & B \end{array}$$

($A_1 A_2 A_3$ are used since they were forming cycles).

1) $A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 \mid a$
$A_3 \to A_1 A_2 \mid b$

II) check if above rules are in GNF or not RHS production should start with higher numbered variables.

$A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 \mid a$
$A_3 \to b$

III) Substitute $A_1 \to A_2 A_3$

$A_3 \to \underbrace{A_2 A_3}_{A_1} A_2$

$A_3 \to A_2 A_3 A_2 \mid b$
$\downarrow$
$A_3 \to \underline{A_3 A_1} A_3 A_2 \mid b \mid a A_3 A_2$

Left Recursion :

$$\boxed{A \rightarrow A\alpha \mid \beta}$$

After removing left recursion

$$\boxed{\begin{array}{l} A \rightarrow \beta A' \\ A' \rightarrow \alpha A' \mid \epsilon \end{array}} \Rightarrow \boxed{\begin{array}{l} A \rightarrow \beta A' \mid \beta \\ A' \rightarrow \alpha A' \mid \alpha \end{array}}$$

$$A_3 \Rightarrow \underbrace{A_3 A_1 A_3 A_2}_{A\alpha} \mid \underbrace{b}_{\beta_1} \mid \underbrace{a A_3 A_2}_{\beta_2}.$$

$$A_3 \rightarrow bA' \mid aA_3A_2A' \mid b \mid aA_3A_2.$$

$$A' \rightarrow A_1A_3A_2A' \mid A_1A_3A_2$$

Using A_3 which is already in GNF rewrite A_2.

$$A_2 \rightarrow A_3A_1 \mid a$$

$$A_2 \rightarrow bA'A_1 \mid aA_3A_2A'A_1 \mid bA_1 \mid aA_3A_2A_1 \mid a.$$

Using A_2 rewrite A_1.

$$A_1 \rightarrow A_2A_3$$

$$A_1 \rightarrow bA'A_1A_3 \mid aA_3A_2A'A_1A_3 \mid bA_1A_3 \mid aA_3A_2A$$

aA_3

Using A_1 rewrite A'

A_1

A_2

A_3

$A' \rightarrow bA'A_1A_3A_3A_2A' \mid aA_3A_2A'A_1A_3A_3A_2A' \mid$
$bA_1A_3A_3A_2A' \mid aA_3A_2A_1A_3A_3A_2A' \mid aA_3A_3A_2A' \mid$
$bA'A_1A_3A_3A_2 \mid aA_3A_2A'A_1A_3A_3A_2 \mid bA_1A_3A_3A_2 \mid$
$aA_3A_2A_1A_3A_3A_2 \mid aA_3A_3A_2.$

PUSH DOWN AUTOMATA

Finite Automata can not recognize all context free languages. FA have finite memories, whereas recognition of context free languages may require storing an unbounded amount of information.

A pushdown automata is a system which is mathematically defined as follows :

$$M = (Q, \Sigma, \Gamma, \delta, Z, F)$$

Where,

Q : Finite set of states

Σ : Input alphabet

Γ : Finite set of stack symbols (stack alphabet)

Also called as push down alphabet

δ : It is a transition function which maps,

$Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow$ subsets of $(Q \times \Gamma^*)$

q_0 : Initial state (start state)

$q_0 \in Q$

Z : Stack start symbol

(It indicates bottom of the stack)

$Z \in \Gamma$

F : Finite set of final states

$F \subseteq Q$

Q) Design a push down automata

$L = \{a^n b^n \mid n \geq 1\}$

Step I) Defn

Step II) ~~Push~~ Logic :

push 1X for each a in stack

and pop 1X for each b in stack

Step III) $Q \rightarrow \{q_0, q_1, q_f\}$

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

$\Gamma \rightarrow \{Z_0, X\}$

$\delta \rightarrow$

$Z_0 \rightarrow Z_0$

$F \rightarrow q_f$

Step IV) Transition Table

$(q_0, a, Z_0) \rightarrow (q_0, XZ_0)$

$(q_0, a, X) \rightarrow (q_0, XX)$

$(q_0, a, X) \rightarrow (q_0, XXX)$

$(q_0, b, X) \rightarrow (q_1, XX)$

$(q_1, b, X) \rightarrow (q_1, X)$

$(q_1, b, X) \rightarrow (q_1, \epsilon)$

$(q_1, \epsilon, Z_0) \rightarrow (q_f, Z_0)$