$(Q, \Sigma, \delta, q_0, \cancel{F}) \longrightarrow$ FA

$(Q, \Sigma, \Delta, \delta, \lambda, q_0) \longrightarrow$ Moore m/c

Q → total no. of states

Σ → input

Δ → output alphabet

δ → transition

λ → output maping

q_0 → initial state

$(Q \quad \Sigma \quad \delta \quad q_0 \quad \cancel{F}) \text{ — FA}$

$(Q \quad \Sigma \quad \Delta \quad \delta \quad \lambda \quad q_0) \text{ — Moore machine}$

total no of states — input — output alphabet — transition — output maping — initial state

Input string n , Output string n+1

~~State which have transition~~

Q.

→ All state can be merged if all states have same transition & o/p symbol associated with same state

Page No.
Date

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

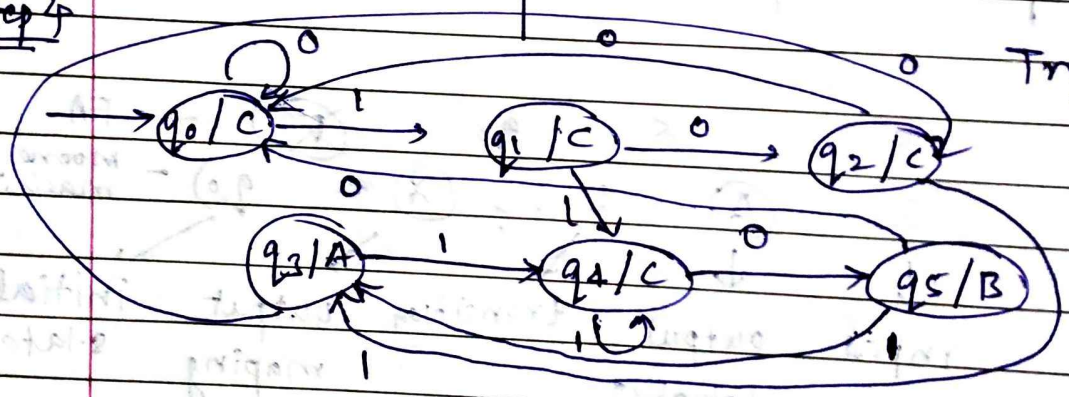
Step 1: Transition Table

	Q \ Σ	0	1
	q_5	q_0	q_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
~~$\lambda(q_5)$~~	~~C~~
$\lambda(q_0)$	C
$\lambda(q_1)$	C
$\lambda(q_2)$	C
$\lambda(q_3)$	A
$\lambda(q_4)$	C
$\lambda(q_5)$	B

Step 4



Transition Diagram

Step 5 Example 0111 → n=4

$\rightarrow q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_4$

0 1 0 1 1

for this sum

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

C C C C A → output

Step 1 $(Q, \Sigma, \lambda, \Delta, \delta, q_0)$

Q: $q_0 \cdots q_5$; Σ: $\{0,1\}$; λ: $\{0,1\}$; Δ: $\{A, B, C\}$; $\delta = Q \times \Sigma$; q_0: initial state

Step 2 ~~$\delta = Q \times Q \rightarrow Q$~~

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

Page No.
Date

Q2 Design a Moore m/c to output A, if i/p does not contain double letter otherwise output R over i/p {a, b}

Step 3)

Ends

δ \ Σ	a	b
q_s	q_0	q_1
q_0	q_2	q_1
q_1	q_0	q_3
q_2	q_2	$q_1 \rightarrow q_2$
q_3	q_0	$q_3 \rightarrow q_3$

Contains

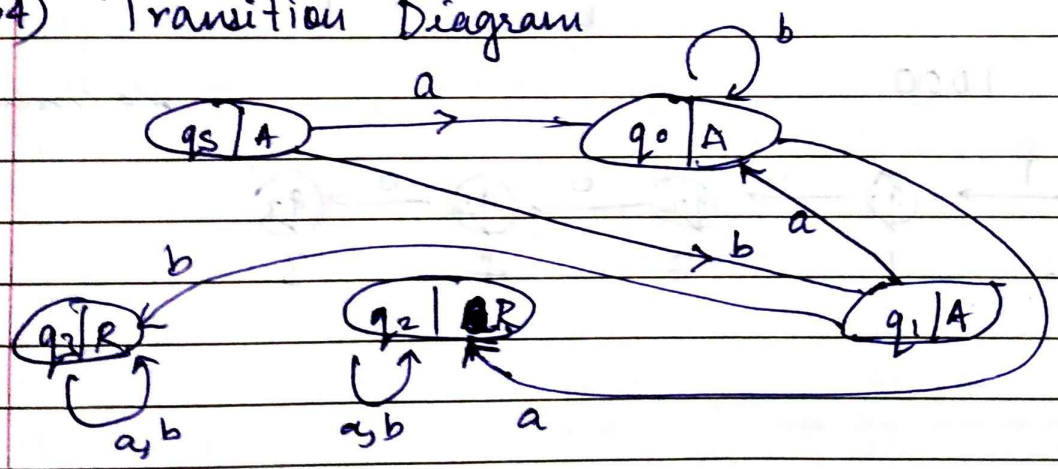
δ \ Σ	a	b
q_s	q_0	q_1
q_0	q_2	q_1
q_1	q_0	q_3
q_2	q_2	q_2
q_3	q_3	q_3

Does Not Contain

δ \ Σ	a	b
q_s	q_0	q_1
q_0	q_2	q_0
q_1	q_0	q_3
q_2	q_2	q_2
q_3	q_3	q_3

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

Step 4) Transition Diagram



Page No.
Date

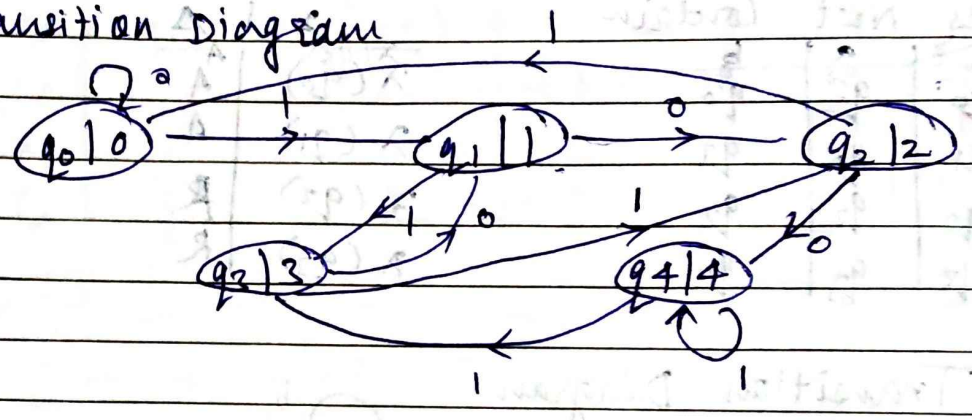
Q3 Design a Moore m/c for residue mod 5 ones $\Sigma = \{0,1\}$

Step 3

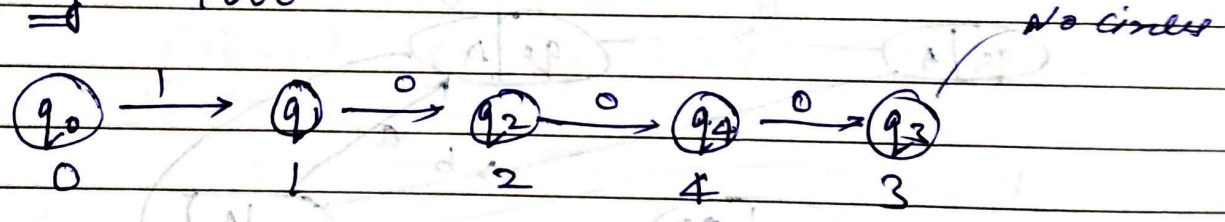
	δ \ Σ	0	1	λ	Δ
→	q_3	q_0	q_1	q_3	
0	q_0	q_0	q_1	q_0	0
01	q_1	q_2	q_3	q_1	1
10	q_2	q_4	q_0	q_2	2
11	q_3	q_4	q_2	q_3	3
100	q_4	q_3	q_4	q_4	4

output mapping func table

Step 4) Transition Diagram



Eg 1000



MEALY MACHINE

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

Input : n ; Output : n+1

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

Step 2 Transit. table $\lambda = Q \times \Sigma \rightarrow \Delta$

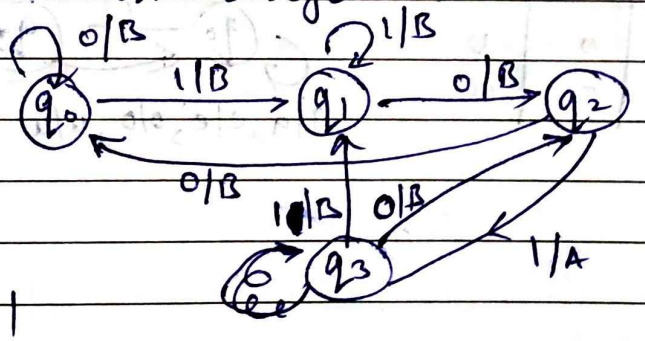
Q \ Σ	0	1
→ q_0	q_0	q_1
0 q_0	q_0	q_1
1 q_1	q_2	q_1
10 q_2	q_0	q_3
101 q_3	q_2	q_1

merge

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

Step 4 Transition Diagram :-



Step 5 0101

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

Output : BBBA

Page No.
Date

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

OR

(to change each occurence of ie to iu)

ie → iu

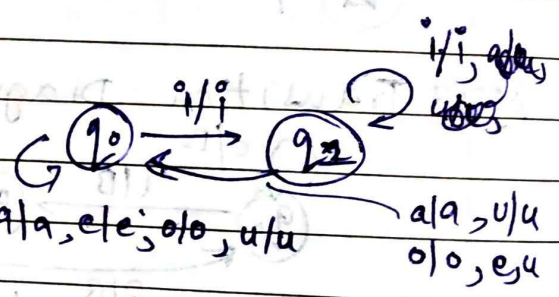
	$Q \backslash \Sigma$	a	e	i	o	u
→	q_s	q_0	q_1	q_2	q_3	q_4
a	q_0	q_0	q_1	q_2	q_3	q_4
e	q_1	q_0	q_1	q_2	q_3	q_4
i	q_2	q_0	q_1	q_2	q_3	q_4
o	q_3	q_0	q_1	q_2	q_3	q_4
u	q_4	q_0	q_1	q_2	q_3	q_4
ie	q_5	q_0	q_1	q_2	q_3	q_4

output table

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

After merging

$Q \backslash \Sigma$	a	e	i	o	u
q_0	a	e	i	o	u
q_1	a	u	i	o	u



Conversion 6 steps
(B) ~~Steps~~, (C) Tuple
Table
Diagram

Page No.
Date

Q3) Design a Mealy m/c to implement binary adder

Solution 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.

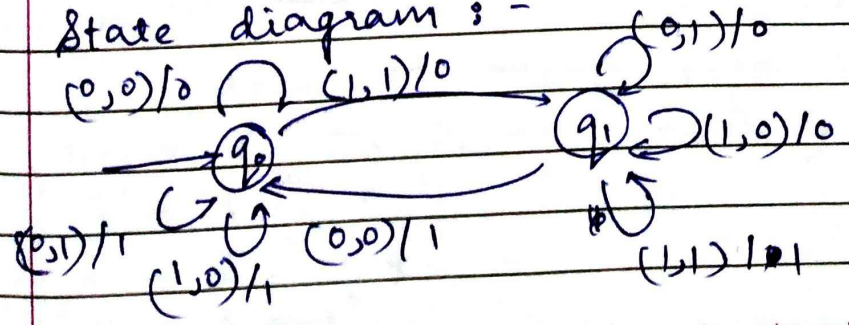
$Q \backslash \Sigma$	(0,0)	(0,1)	(1,0)	(1,1)	
→ q_0	q_0	q_0	q_0	q_1	merge
NC ← q_0	q_0	q_0	q_0	q_1	
C ← q_1	q_0	q_1	q_1	q_1	

δ	(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

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

$Q \backslash \Sigma$	(0,0)	(0,1)	(1,0)	(1,1)
q_0	0	1	1	0
q_1	1	0	0	1

State diagram :-



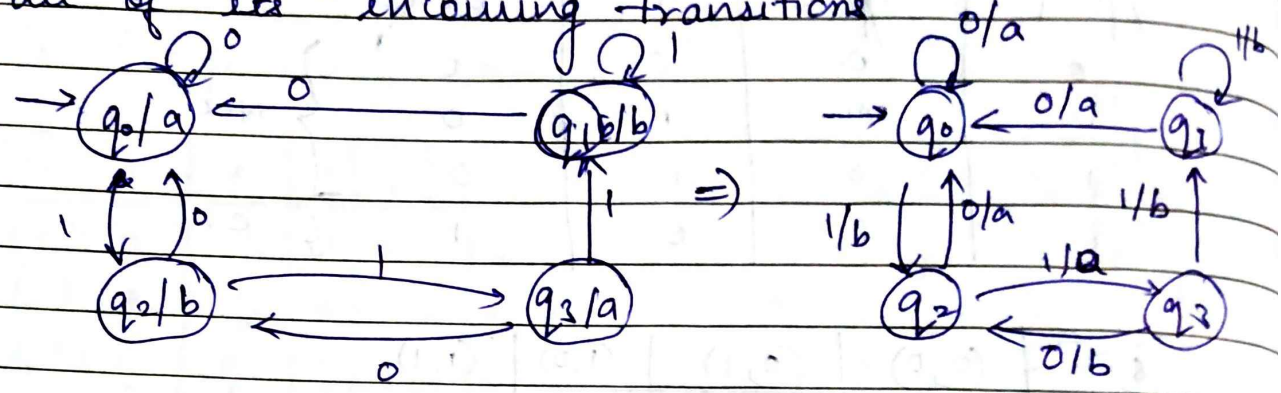
Mostly Conversion : Mealy → Moore

Page No.
Date

CONVERSION OF MOORE m/c TO MEALY m/c

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

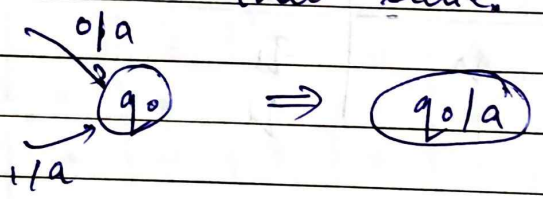
Eg:



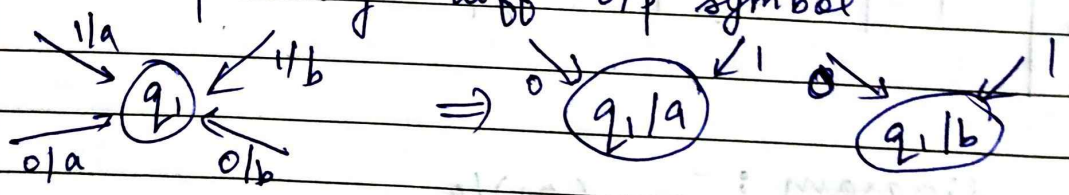
CONVESION OF MEALY m/c TO MOORE m/c

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

Eg:

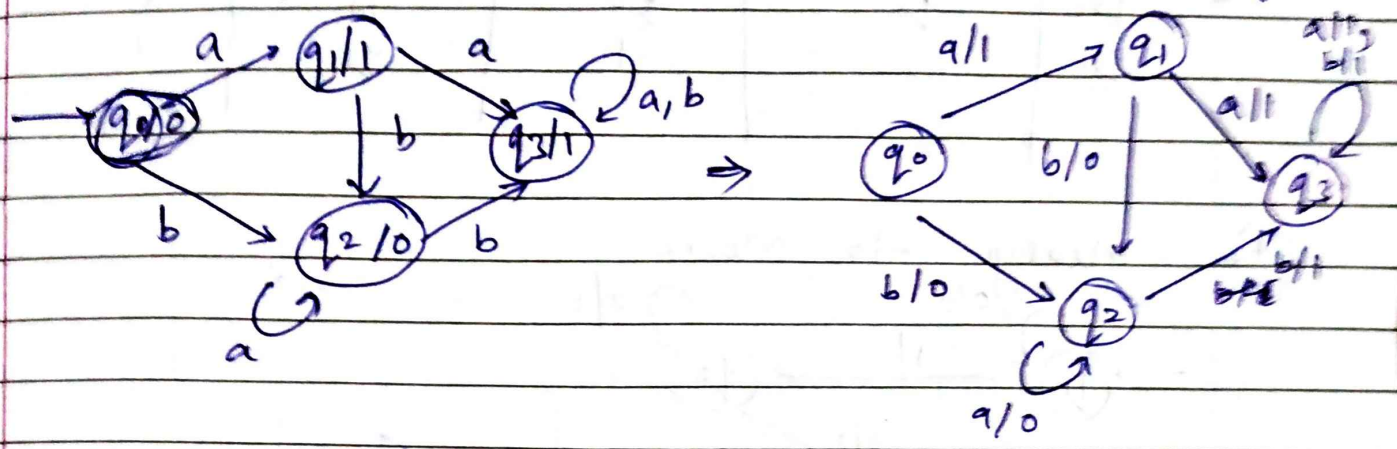


(2) If o/p symbol associated with the incoming transition are not same then split the state as many as o/p ~~syt~~ symbol with each state producing diff o/p symbol

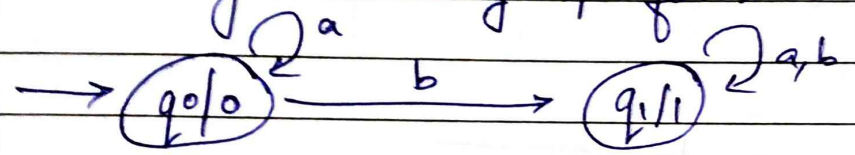


Page No.
Date

Q1) Convert given moore m/c into equivalent mealy m/c



Q2) Convert given Mealy m/c for the given Moore m/c



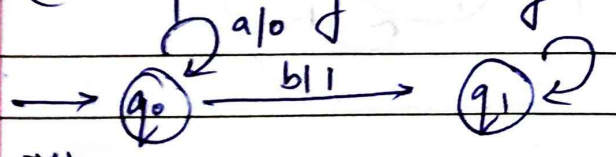
Let Moore m/c be

$M_0 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

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

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

Corresponding Mealy m/c



~~The~~

The required mealy m/c is

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

$Q = \{q_0, q_1\}$

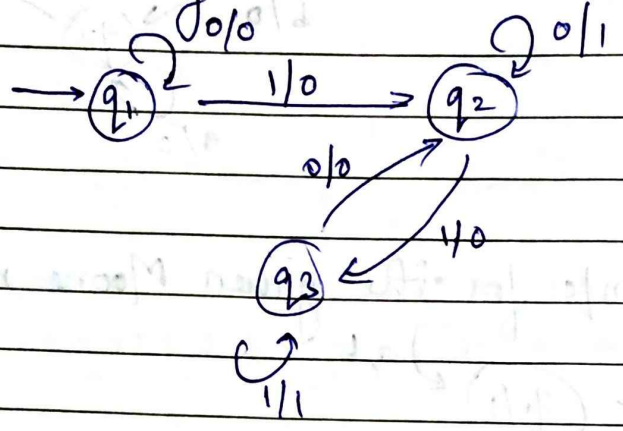
$\Sigma = \{a, b\}$

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

whenever you are not able to decide any value you can assume any value from $\Delta = \{0, 1\}$

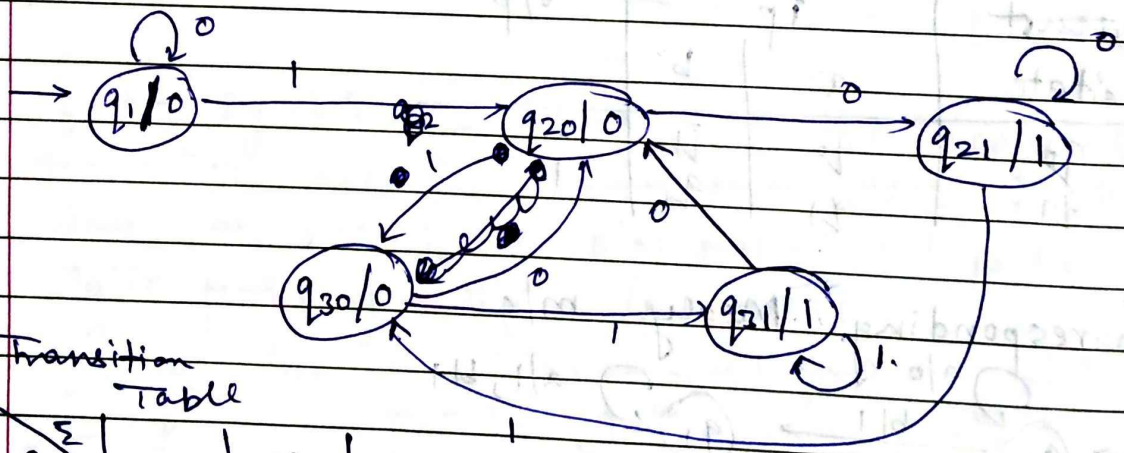
Q \ Σ	a		b	
	state	o/p	state	o/p
→ q_0	q_0	0	q_1	1
q_1	q_1	1	q_1	1

Q4) Mealy to Moore



Sol^n

Q \ Σ	0		1	
	State	o/p	State	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



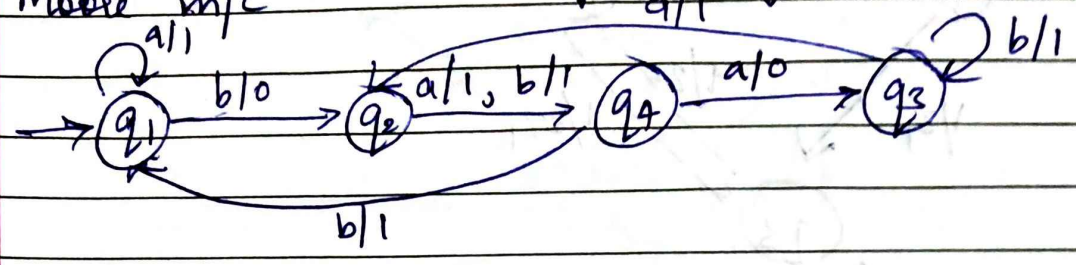
Transition Table

Q \ Σ	0	1	o/p
q_1	q_0	q_{20}	0
q_{20}	q_{21}	q_{30}	0
q_{21}	q_{21}	q_{30}	1
q_{30}	q_{20}	q_{21}	0
q_{31}	q_{20}	q_{31}	1

Page No.
Date

Mealy → Moore

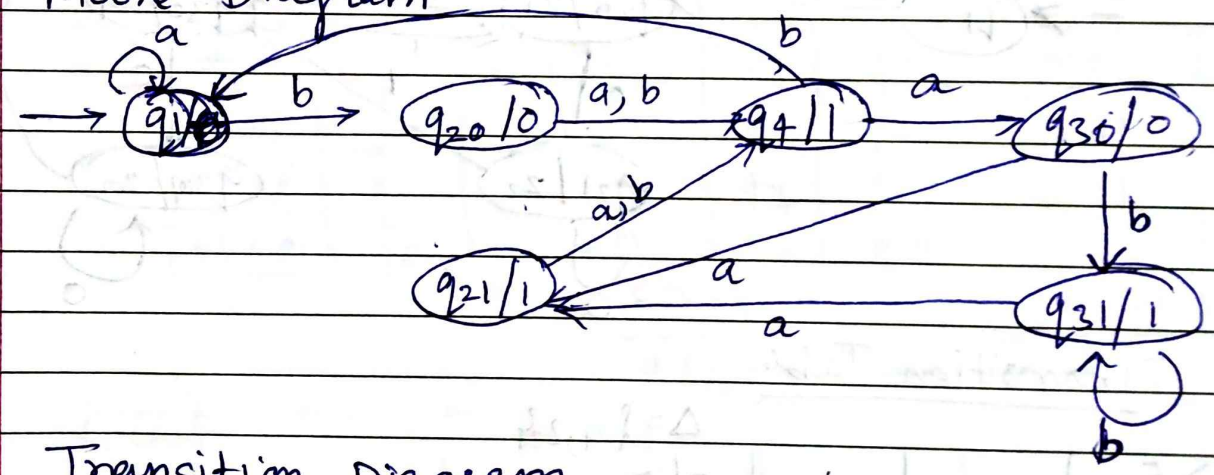
Q Convert the following Mealy m/c into equivalent Moore m/c



Transition table for above Mealy m/c

Q \ Σ	a: State	a: o/p	b: State	b: 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

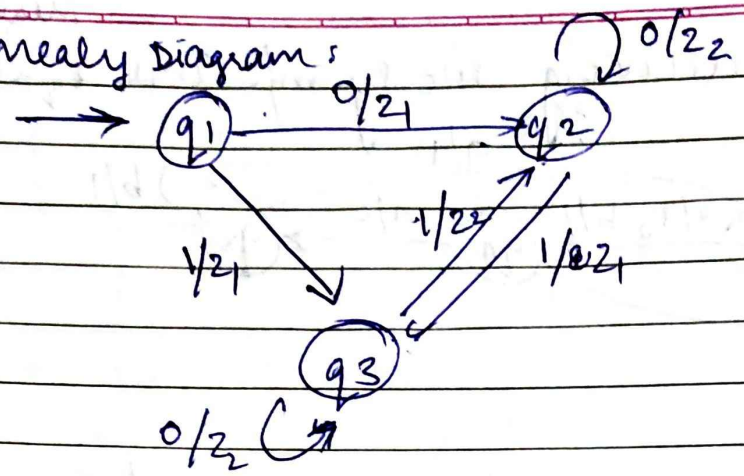


Transition Diagram

Q \ Σ	a	b	o/p
q_1	q_1	q_{20}	1
q_{20}			0
q_{21}			1
q_{30}			0
q_{31}			1
q_4			1

Page No.
Date

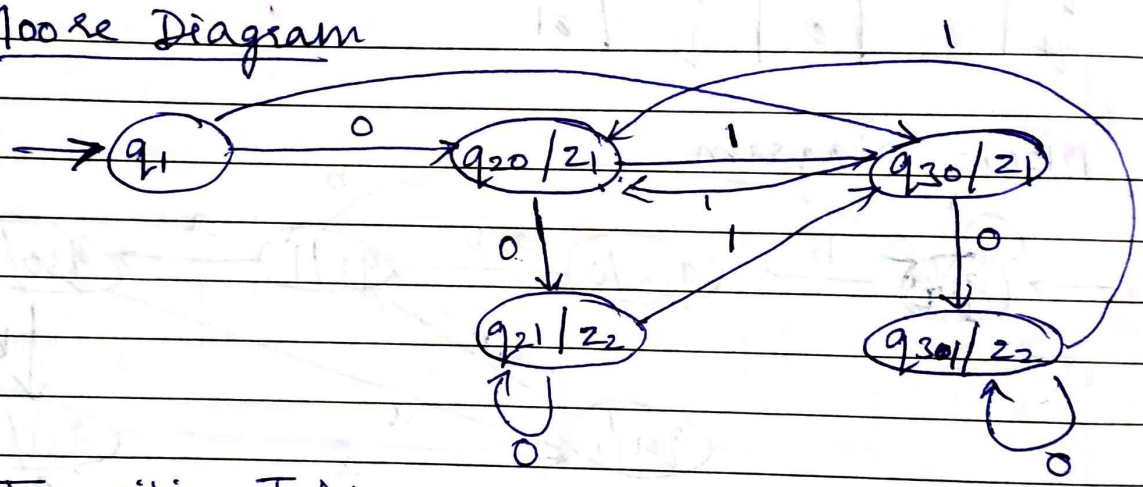
Q Mealy Diagram:



Step 1

q \ Σ	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_2	z_1
q_3	q_3	z_2	q_2	z_1

Moore Diagram

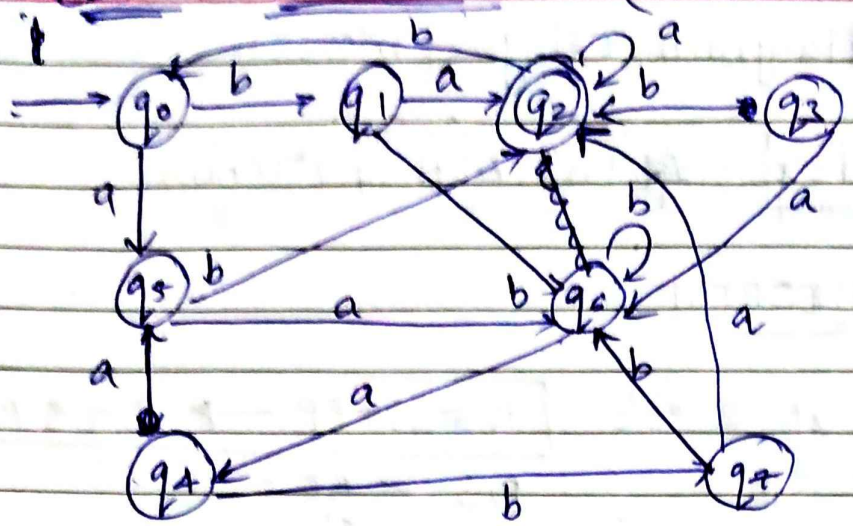


Transition Table

$\Delta = \{z_1, z_2\}$

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

DFA Minimization (Alternative Method)



Solution

Step 1

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

Non Final

Step 2

	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 3

Non final

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

Step 4

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

Q \ Σ	a	b
$\{q_0, q_4\}$	$\{q_3, q_5\}$	$\{q_1, q_7\}$
$\{q_1, q_7\}$	$\{q_2\}$	$\{q_6\}$
$\{q_3, q_5\}$	$\{q_6\}$	$\{q_3, q_5\}$
$\{q_6\}$	$\{q_0, q_4\}$	$\{q_6\}$
$\{q_2\}$	$\{q_2\}$	$\{q_0, q_4\}$

18/09/2020

Page No.
Date

1) Transition diagram using ϵ moves
2) NFA to DFA
3) Algebraic ~~method~~ Theorem Method – ~~Arden's Method~~ Arden's Theorem

ARDEN'S THEOREM

(i) Equivalent of 2 RE's $\boxed{R = Q + RP \rightarrow R = Qp^*}$

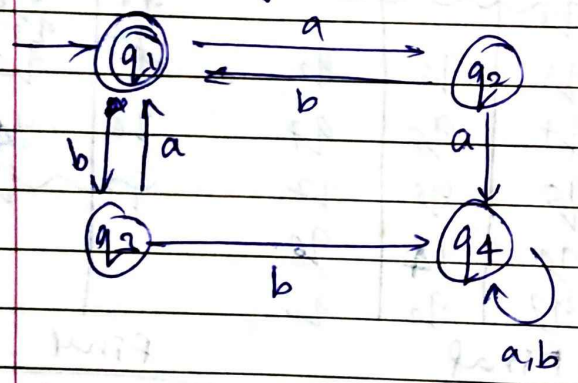
~~(ii)~~ DFA $\xrightarrow{\text{find}}$ RE

$P, Q \rightarrow$ RE's
$R \rightarrow$ State

Assumptions:

1) Transition diagram shouldn't have ϵ moves
2) It should have only one ~~start~~ initial state

Q1) Find regular exp. for the given DFA



Step 1

$q_1 = q_2 b + q_3 a + \epsilon$
$q_2 = q_1 a$
$q_3 = q_1 b$
$q_4 = q_2 a + q_3 b + q_4 a + q_4 b$

Not reachable ~~from~~ start state
Not outgoing edges
(Redundant State)

$q_1 = q_2 b + q_3 b + \epsilon$
$q_1 = q_1 ab + q_1 ba + \epsilon$
$q_1 = q_1 (ab + ba) + \epsilon$
$R = RP \quad Q$

~~Arden's th~~

P &

$q_1 = \epsilon (ab + ba)^*$
$q_1 = (ab + ba)^*$

Arden's thm

P & Q $\leftarrow$ RE's

$\Downarrow$

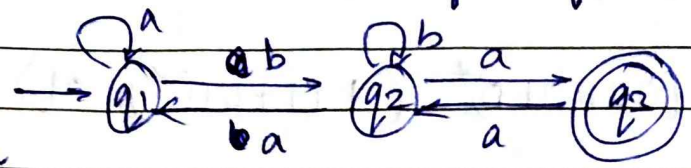
$\boxed{R = Q + RP} \Rightarrow \boxed{R = Q(P^*)}$

State

Note: ~~Fo~~ final RE should be wrt. final state

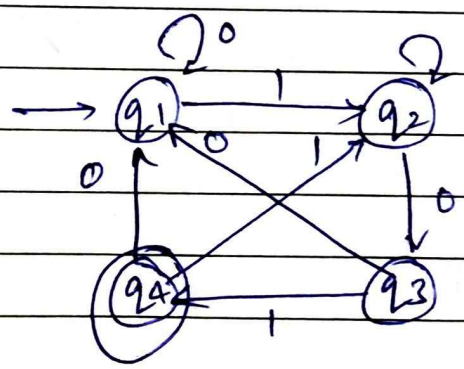
Page No.
Date

Q2) Write down RE ~~from~~ for the given DFA



Sol^n

Q3)



Step 1

initial state

$q_1 = q_1 0 + q_2 0 + q_4 0 + \epsilon$

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

$q_3 = q_2 0$

$q_4 = q_3 1$

$q_4 = q_3 1 = q_2 01$

From ① ~~$= (q_1 1 + q_2 1 + q_4 1)01$~~

$q_2 = q_1 1 + q_2 1 + q_2 011$

$q_2 = q_1 1 + q_2 (1 + 011)$

R Q R (P)

$q_2 = q_1 1 (1+011)^*$

$q_1 = q_1 0 + q_2 0 + q_4 0 + \epsilon$

$= q_1 0 + q_2 00 + q_2 010 + \epsilon$

$= q_1 0 + q_2 (00 + 010) + \epsilon$

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

~~$q_1 = q$~~

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

R = R P + Q

R = QP*

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

18/9/20

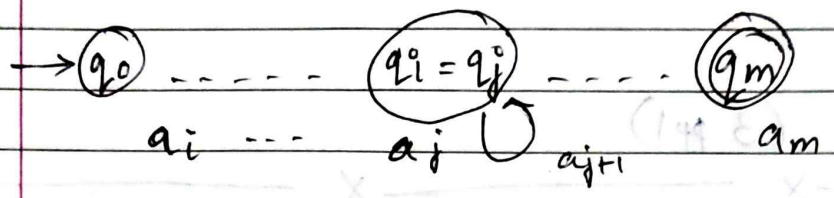
Pumping Lemma for Regular Set

Page No.
Date

Theory Q : 8m
Numerical : 5/8m

Let n be regular set then there ~~are~~ is constant n such that if z is any word in L & $|z| > n$ 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 q_0.
- After reading from some position we reach the state q_i & then reading remaining portion we ~~get~~ reach q_m, final state.
- Suppose DFA n i/p states then by Pigen 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 $z = uvw$

→ ~~Function~~ Use of Pumping Lemma

It is used to proove that a given language is not regular

→ Steps to Solve Numerical

1) Initially assume L (given language) is regular
2) Write $z = uvw$ where $|uv| \leq n$ & $|v| \geq 1$
3) Find suitable integer i such that $uv^iw \in L$. This contradicts our assumption

Hence L is not regular.

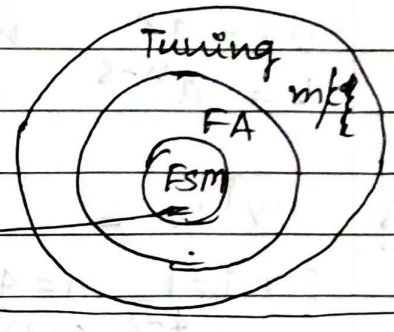
Any language accepted by finite automata is regular.

Page No.
Date

Q Prove ~~that the~~ Language $L = \{a^n b^n \mid n \geq 1\}$ is not regular

Step 1 Let us assume that language is regular.

For IRREGULAR, we need to store in memory
For REGULAR, we do not need memory



Step 2 According to pumping lemma there are n such that $z = uvw$ where $|uv| \leq n$ $|v| \geq 1$

$a^n b^n$ aaabbb ⇒

o/p Equal or more

$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_4 \xrightarrow{b} q_5 \xrightarrow{b} q_6$

n=3

$z = \underbrace{aaa}_{u}\ \underbrace{b}_{v}\ \underbrace{bb}_{w}$ $|uv| \leq n$ $v \geq 1$

$uv = aaa$ $v = a$ $w = bbb$

n=4

$z = aaaa\ bbbb$ $|uv| \leq n$ – aaaa $v \geq 1 \rightarrow a$ or aa, aaa

Eg1 $u = aaa$ $v = a$ $w = bbbb$ $w = bbbb$

$uv^i w = \{aaa\}(a)^i\ bbbb$

$i = 1$ aaaabbbb ✓ same o/p

$i = 2$ aaaaa bbbb X

it proves that given language is not regular

Eg2 $u = aa$ $v = aa$ $w = bbbb$

$uv^i w = aaa\ (aa)^i\ bbbb$

$i = 1$ = aaaaa bbbb ✓ Same o/p

$i = 2$ = aaaaaa bbbb X

Not regular

Page No.
Date

Step2 $z = uvw = a^s b^s$ $(s=n)$ 3

$uv = a^s$ $w = b^s$ aaabbbccc

$v = a^q$ $v = a^{s-q}$ $w = b^s$

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

$uv = a^n$ $w = b^n$

$u = a^{n-s}$ $v = a^s$ $w = b^n$

Step3 $uv^i w = a^{n-s} (a^s)^i b^n$

For $i=1$ $n=4$ $s=2$ Same

$a^{4-2} (a^2)^1 b^4 = aa\ aabbbb$ ✓

For $i=2$ $n=4$ $s=2$

$a^{4-2} (a^2)^2 b^4 = aaaaaabbbb$ X

This proves that given expression is not regular.

Q2) Given language $L = \{a^n b^n c^n \mid n \geq 1\}$ is not regular

Step1 Let us assume the language is regular

Step2 According to pumping lemma there are n such that $z = uvw$ where $|uv| \leq n$ $|v| \geq 1$

$z = uvw = a^n b^n c^n$

$uv = a^n$ $w = b^n c^n$

$u = a^{n-s}$ $v = a^s$ $w = b^n c^n$

Step3 $uv^i w = a^{n-s} (a^s)^i b^n c^n$

For $i=1$ $n=3$ $s=2$

$a^{3-2} (a^2)^1 b^3 c^3 = aaabbbccc$ ✓

For $i=2$ $n=3$ $s=2$

$a^{3-2} (a^2)^2 b^3 c^3 = aaaaabbbccc$ X

This proves that given expression is not regular.
Hence Proved

$a^{n!} \mid n \geq 0$

$a^n b^l a^k \quad k \geq n+l$

Page No.
Date

Q3) Given language $L = \{a^n b a^n \mid n \geq 1\}$ is not regular

Step1 Let us assume the language is regular

Step2 According to pumping lemma there are n such that $z = uvw$ where $|uv| \leq n$ $v \geq 1$

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

$uv = a^n \qquad w = ba^n$

$u = a^{n-s} \quad v = a^s \quad w = ba^n$

Step3 $uv^iw = a^{n-s}(a^s)^i ba^n$

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

$a^{3-2}(a^2)^1 ba^3 = aaabaaa$ ✓

For $i = 2 \quad n = 3 \quad s = 2$

$a^{3-2}(a^2)^2 ba^3 = aaaaabaaa$ ✗

The given language is not regular.
Hence Proved.

Q4) Given language $L = \{a^{n!} \mid n \geq 1\}$ is not regular

Step1 Assume L is not regular. Let m be a pumping lemma constant.

Step2 We don'not know the value of m, but whatever it is, we can always ~~leave~~ choose $n = m$

Let $n = m$

Let $w = a^{m!}$ [m: no. of states

Step3 Select substring 'v' near the begining of the string we can select 'v' only from m symbols as we have only m number of states and length of string $\geq m$

Grammar

A grammar is defined by quadruple (V, T, P, S)
i.e. $G = (V, T, P, S)$
where
V: Finite set of variable [Non terminals] ~~{A, B, C}~~
T: Finite set of terminals [terminal symbols]
P: Finite set of productions [rules] that represent the ~~S i.e.~~ recursive defination language
S: Start symbol $S \in V$

- Variables [Non terminals]: (Capital letters)
Variables ~~are~~ are those symbols that take part in the derivation of a sentence but are not ther part of derived sentence

- Terminals:
~~Those ther~~ Terminals are those symbols that are the part of derived sentences.

- Production rules:
Production is of the form
$\alpha \longrightarrow \beta$
where α and β are in some sequential 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.

Page No.
Date

Example

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

~~S e's~~ $V = \{S, A\}$ $T = \{0, 1\}$ S = start symbol

P $S \longrightarrow Ab$

$S \longrightarrow a$

$A \longrightarrow aA$

$A \longrightarrow a$

The above production can be written as

$S \longrightarrow Ab \mid a$ $A \longrightarrow aA \mid a$

→ **Context Free Grammer (CFG) :**

Defination

A grammar is a CFG if all the productions are of

$A \longrightarrow \alpha$

L R

where

A : Variable [Non terminal] single capital

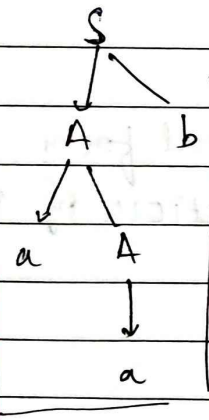
α : Sequential form combo of terminals & non-terminal symbols

combo of cap & small

of Production

LHS – One terminal

RHS – Combination of terminal & non-terminal



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 type of derivations:

1) Leftmost derivation 2) Rightmost derivation

Page No.
Date

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

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

Q Let $G = (\{S\}, \{a, b, +, *\}, P, S)$ 5–8 marks

$\{S\}$ → Terminal; $\{a, b, +, *\}$ → non-terminal; P → Production rule; S → Start Prod.

Solⁿ (i) Leftmost derivation $a+a*b$

We start with a production of a start symbol

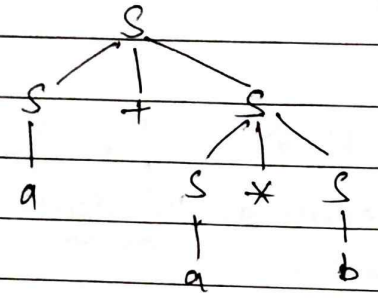
	Production Rule
$S \rightarrow S+S$	$[S \rightarrow S+S]$
$\rightarrow a+S$	$[S \rightarrow a]$
$\rightarrow a+S*S$	$[S \rightarrow S*S]$
$\rightarrow a+a*S$	$[S \rightarrow a]$
$\rightarrow a+a*b$	$[S \rightarrow b]$

(ii) Rightmost derivation for $a+a*b$ ~~a*b~~ ~~a*a+b~~ $a*a+b$
Here we apply productions to rightmost non-terminals

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

Page No.
Date

→ Derivation tree (Parse-tree)

It is a graphical representation of the derivation process. The derivation in CFG can be

Its derivation tree