TOC (Theory of Computation) {PANDEY OR MISHRA

__/__/__

Alan Turing

Automaton is defined as a system where energy, materials and information are transformed, transmitted and used for performing some functions without the direct participation of human being.

Attributes:

1. finite no. of Inputs. (finite time)
2. finite no. of Outputs. (finite time)
3. States
4. State Relation: The next state of an automaton at any instant of time is determined by the present state and present input.
5. Output Relation: The output is related to either state only or to both input and state.

➪ Description of finite Automaton:

A finite automaton can be represented by five tuple ie, $(Q, \Sigma, \delta, q_0, F)$

$Q \rightarrow$ finite non-empty set of states. $Q = \{q_0, q_1, \ldots\ldots q_5\}$

$\Sigma \rightarrow$ finite non-empty set of inputs called input alphabet.

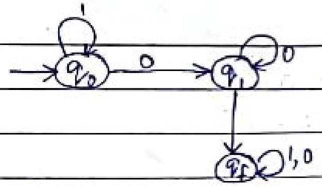
$\delta \rightarrow$ is the fn which maps $\boxed{Q \times \Sigma \rightarrow Q}$ and is usually called transition function

$q_0 \rightarrow$ Initial state and $q_0 \in Q$

$F \rightarrow F \subseteq Q$ is the set of final states.

Scanned by CamScanner

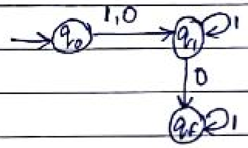
__/__/____



{0,1}

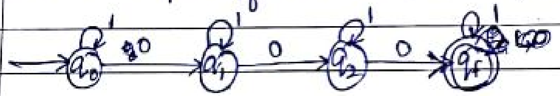
011

110



101

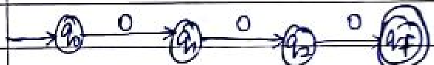
Q Construct a finite automata that accepts the set of strings where the no. of zeroes in every string is multiple of 3 over the alphabet {0,1}



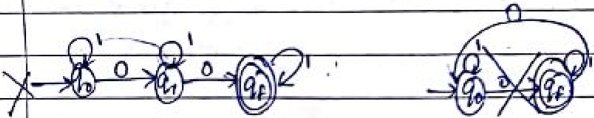
1000

0100

0001

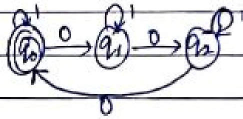


000

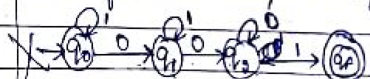


$3 \times 0 = 0$

$3 \times 1 = 3$



Q Design a finite automata which accepts the lang in which 2nd symbol of input is Zero and fourth input symbol is 1.



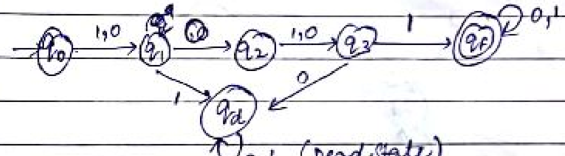
0001

1011

1001 0001

__/__/____

1001



(Dead State)

Q $L = \{ (01)^i \, 1^{2j} \mid i \geq 1, j \geq 1 \}$

0101

for $i = j = 1$

$L = (01)^1 \, 1^{2(1)}$

$= 0111$

for $i = j = 2$

$L = (01)^2 \, 1^{2(2)}$

$= 01011111$

for $i = j = 3$

$L = (01)^3 \, 1^{2(3)}$

$= 010101111111$

for $i = j = 4$

$L = (01)^4 \, 1^{2(4)}$

$= 0101010111111111$

$L = (01)^5 \, 1^{2(5)}$

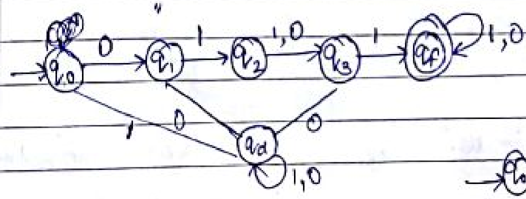
0111

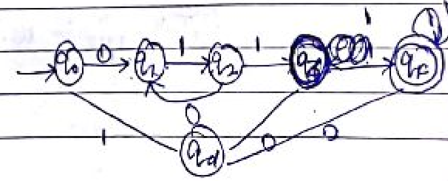
01011111

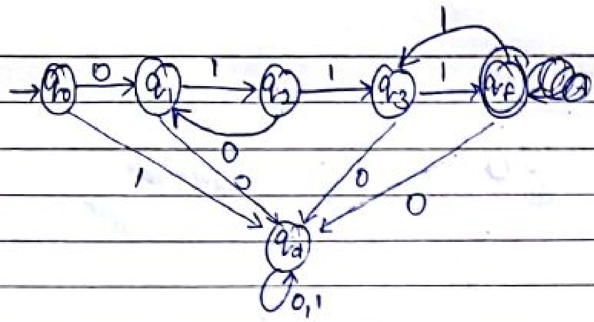
010101111111

0101010111111111

01010101011111111111

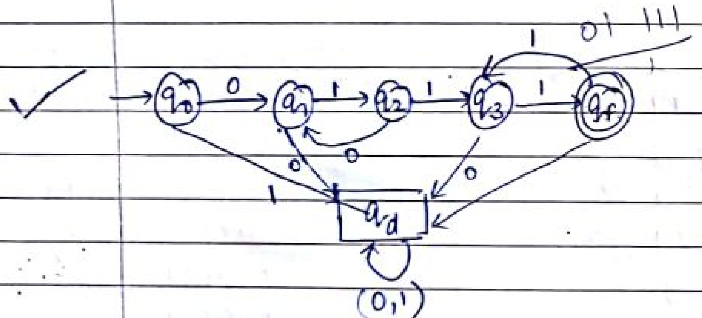






0101 1111

111



Q $L = \{ w \in (a,b)^* \mid n_b(w) \bmod 3 > 1 \}$

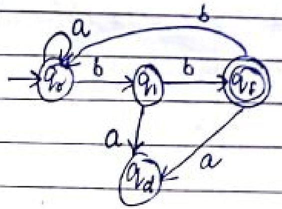
$\{a,b\}^* = (a,b)^*$

abb

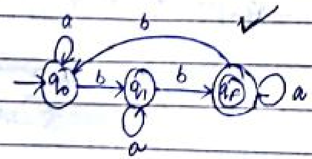
abbbbb

abbbbbbbb

abbbbbbbbbbb



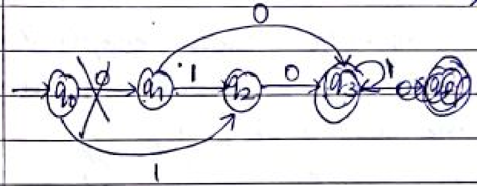
OR



Q Design a finite automata over the alphabet {0,1} which accepts set of string which either start with 01 ~~and~~ or end with 01

Sol

01 -- 01



0100

0001

001

010

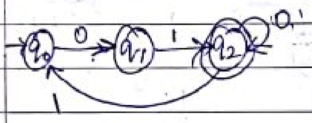
001

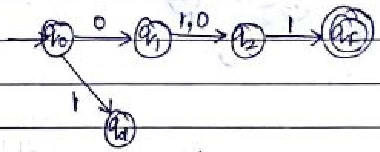
011

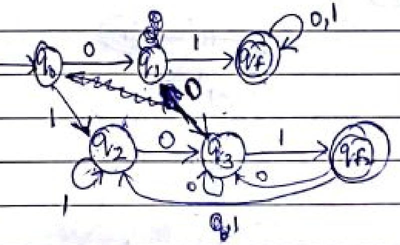
101

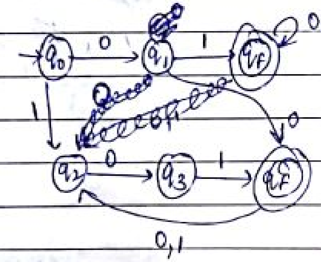
001

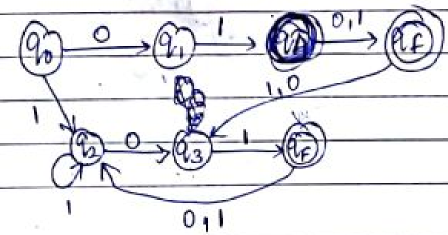
010

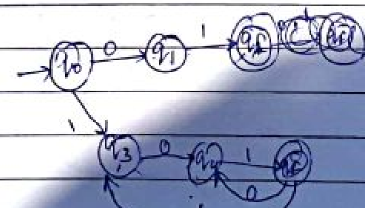


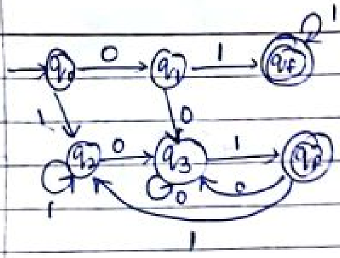






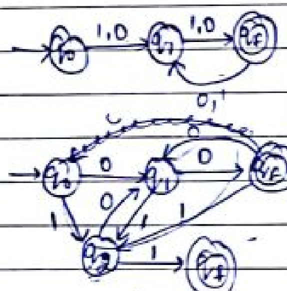


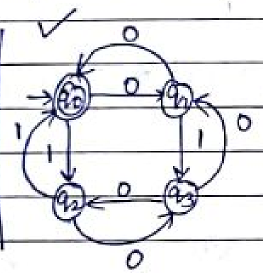




Ques Design a finite automata which accepts the strings having both an even no. of zeroes and an even no. of 1's over the alphabet {0,1}

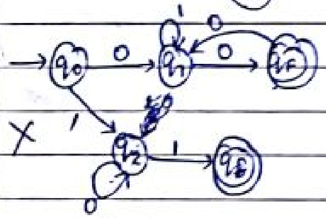
Sol

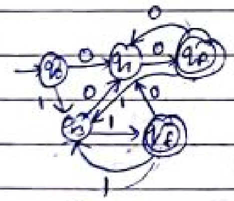




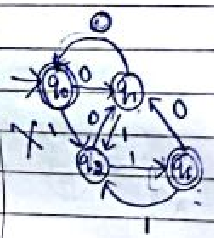
00
0000
000 000

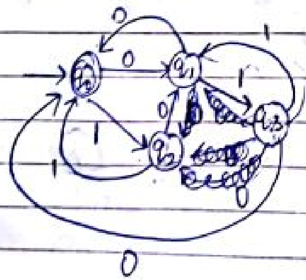
11
1111
0101





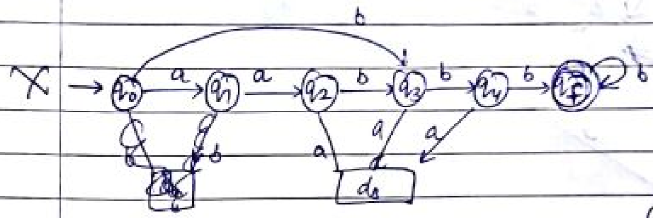
00
0011
0101
1100
1001





Q $L = \{ w : n_a(w) = 2, n_b(w) \geq 2, w \in (a,b)^* \}$

bbba

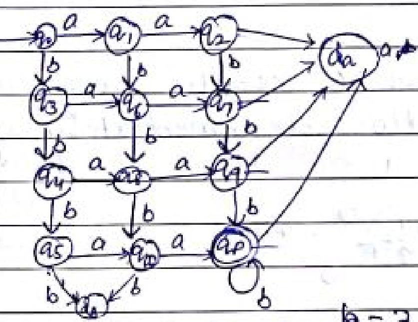


aa bbbb

bb aa bbbb

~~bbb~~ a

bb b aaa bbb

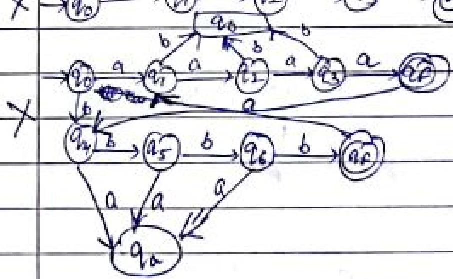


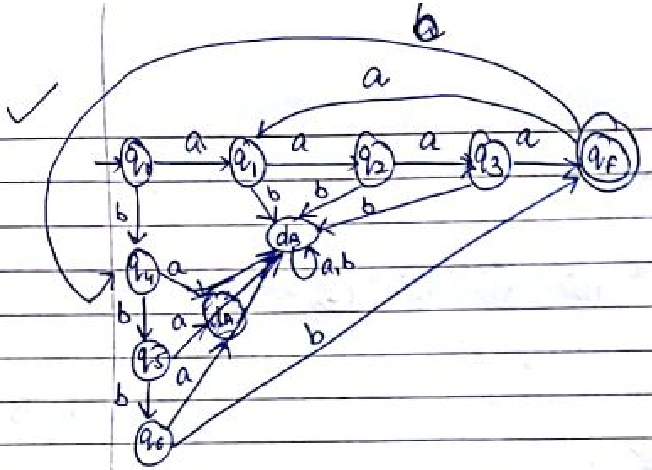
a=2 b=7

aaa_ aa_bbb

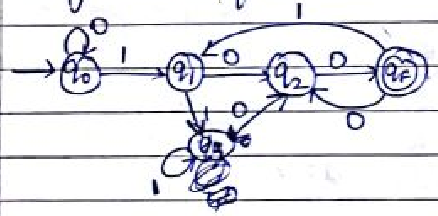
Q Design a DFA over the alphabet a, b such that every string accepted by automation contains no runs of length less than 4

aaaabbbb





Q Design a DFA which accepts strings in which every 00 is followed immediately by a 1



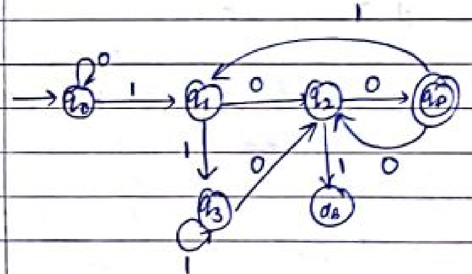
100100

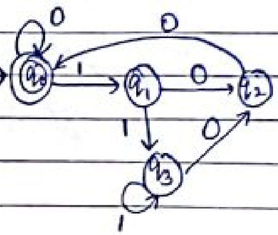
~~001~~ 0100

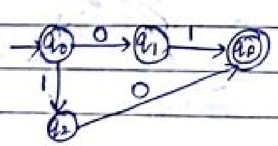
01 00

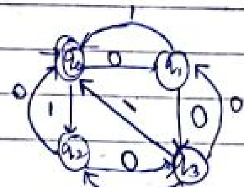
~~10~~ 0100

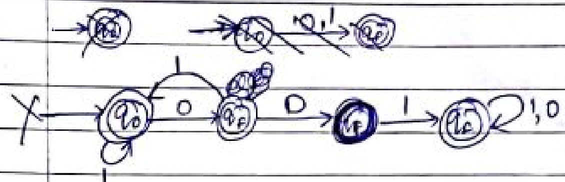
0101

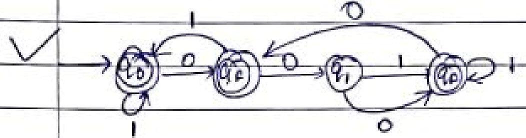








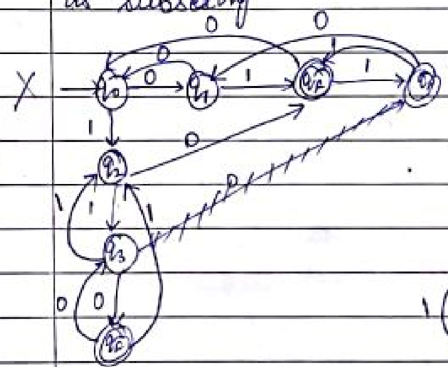


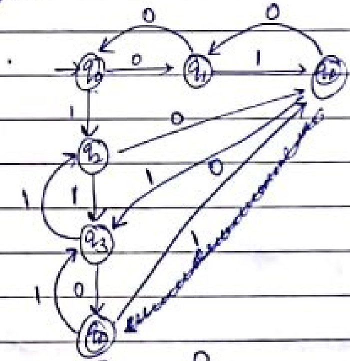


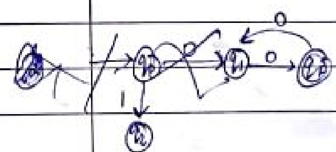
Ex Design DFA which accepts language which contains leftmost symbol differ from rightmost symbol.

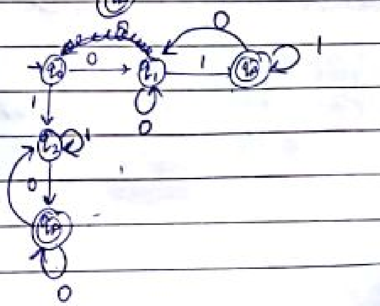
Ex Design DFA which accepts set of strings such that every string contains 00 as a substring but not 000 as substring

01
10
001
011
110
100









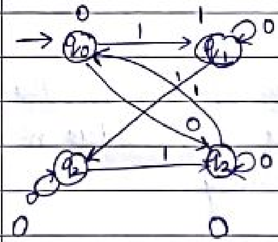
2.

Moore Machine

The mathematical formulation of Moore machine can be defined as a six ~~ky~~tuple machine i.e,

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

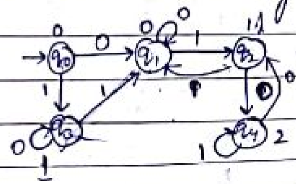
$Q \rightarrow$ Non-empty finite set of states in M_0.
$\Sigma \rightarrow$ Non-empty finite set of input symbols
Δ (upper) $\rightarrow$ Non-empty finite set of outputs.
δ (lower) $\rightarrow$ Transition function ~~which~~ which takes two arguments (input state and input symbol)
$q_0 \rightarrow$ Initial state
$\lambda \rightarrow$ It is the mapping function which maps Q to output.



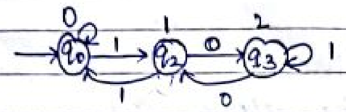
q_0 0111
output 00010

Q Design a Moore M/C which determine the residue ~~[3]~~ mod 3 for each binary string treated as binary integer.

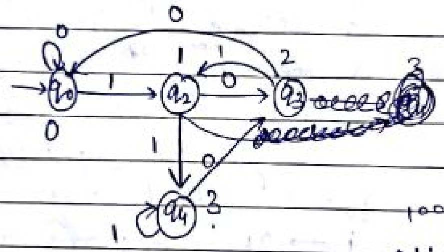
001
010
011
100
111



0	0000	% 3	=	0
1	0001	% 3	=	1
2	0010	"		2
3	0011	"		0
4	0100	"		1
5	0101	"		2
6	0110	"		0
7	0111	"		1
8	1000	"		2



0	0000	% 4	= 0
1	0001	% 4	= 1
2	0010	"	= 2
3	0011	"	= 3
4	0100	"	= 0
5	0101	"	= 1
6	0110	"	= 2
7	0111	"	= 3
8	1000	"	= 0



1001

0110

1011

Q Give the moore m/c for the input from the input alphabet {0,1,2} which prints the residue module 5 of the input treated as ternary.

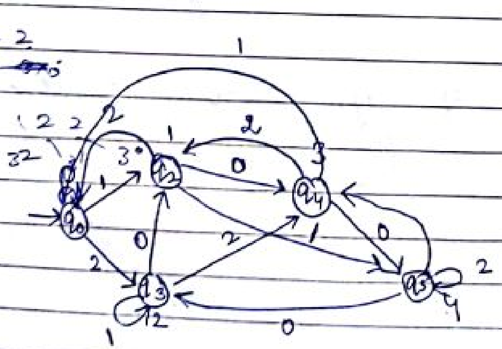
1001

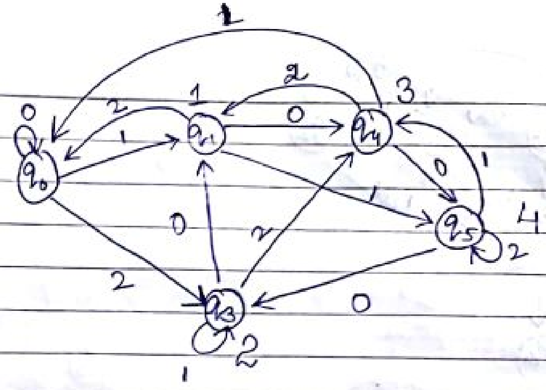
27 9 3 1

0000 = 0
0001 = 1
0002 = 2
0010 = 3
0011 = 4
0012 = 0
0020 = 1
0021 = 2
0022 = 3
0100 = 4
0101 = 0
0102 = 1
0110 = 2
0111 = 3
0112 = 4

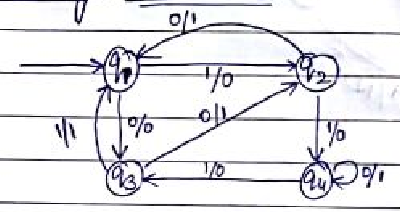
012

0120 = 0
0121 = 1
0122 = 2
0200 = 3
0201 = 4
0202 = 0





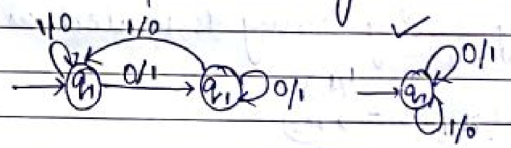
Mealy Machine



	a = 0		a = 1	
	State	o/p	State	o/p
q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

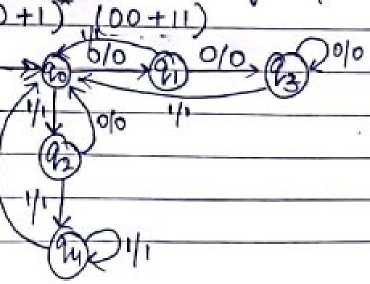
Q Design a Mealy M/c which prints 1's complement of the input bitstring over the alphabet {0,1}

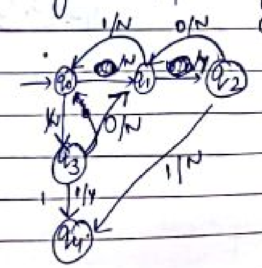
00 = 11
10 = 01



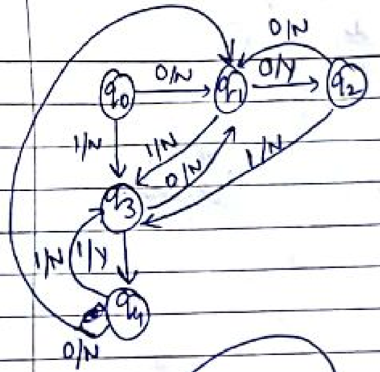
Q Construct a mealy m/c for regular expression $(0+1)^*(00+11)$

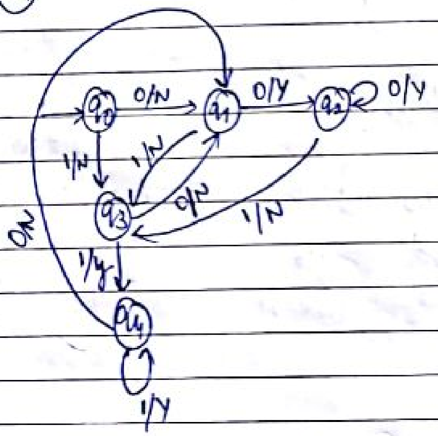
00 = 0
0+1 = 1



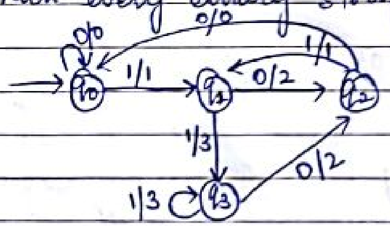


Scanned by CamScanner

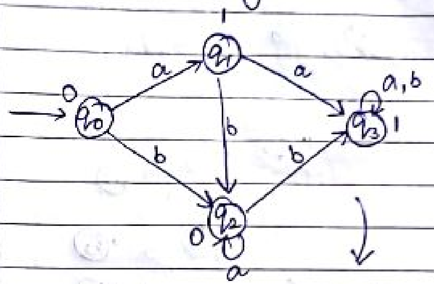


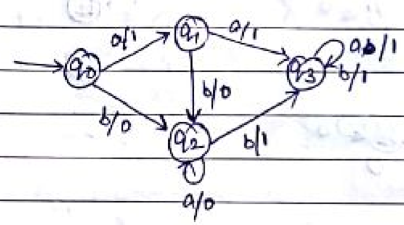


Q Construct a mealy m/c which residue mod 4 in which every binary string is treated as binary integer

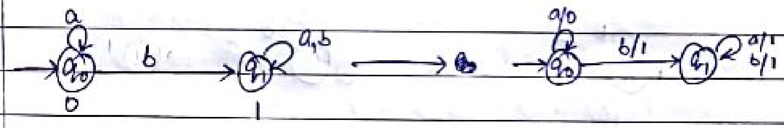


Moore to Mealy :

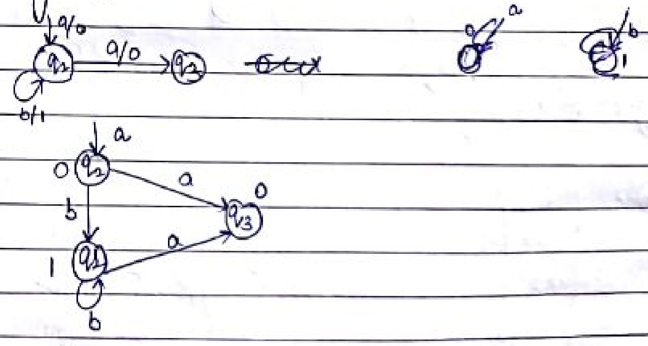


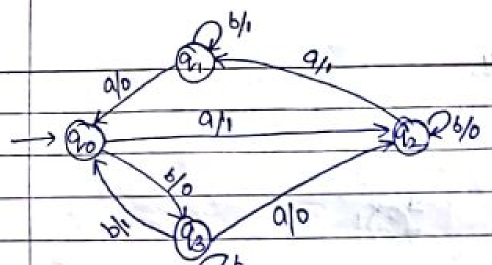


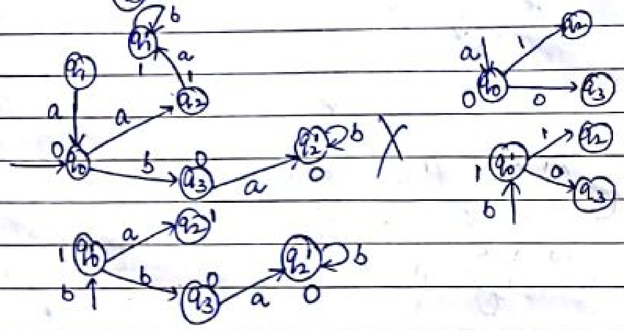
Q

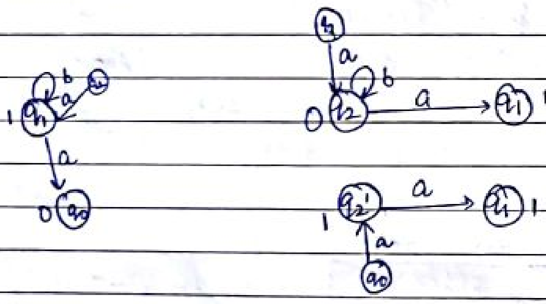


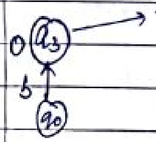
Mealy to Moore

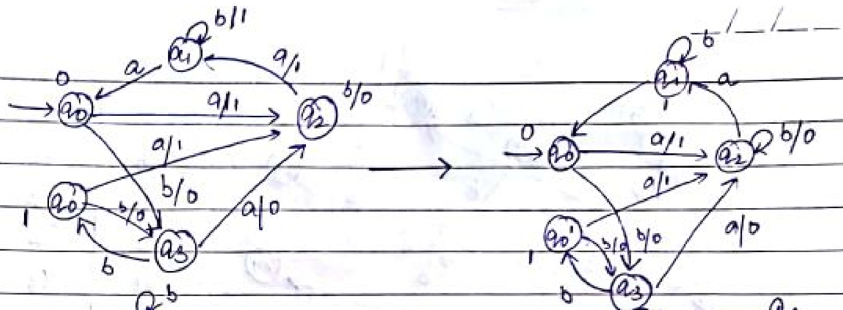


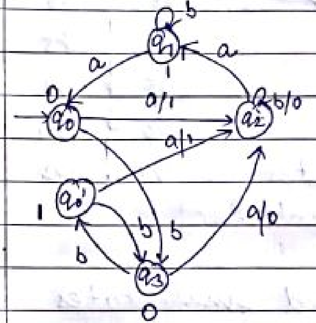


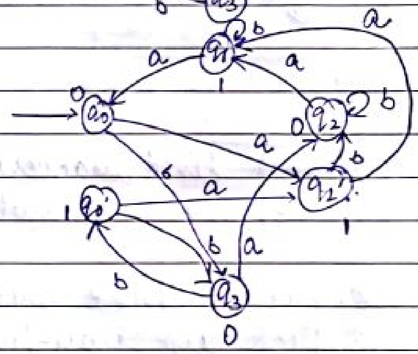






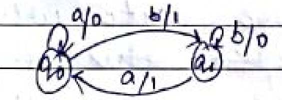


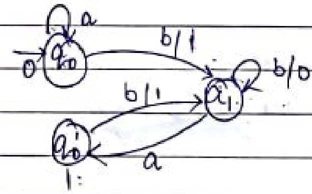


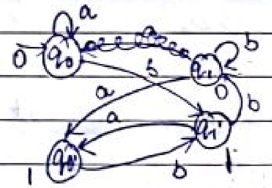


ba
10

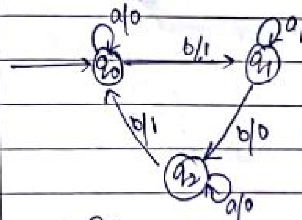
1

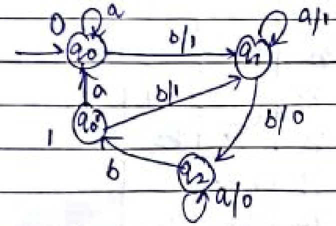


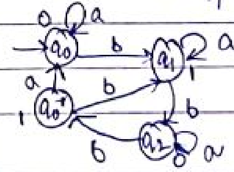




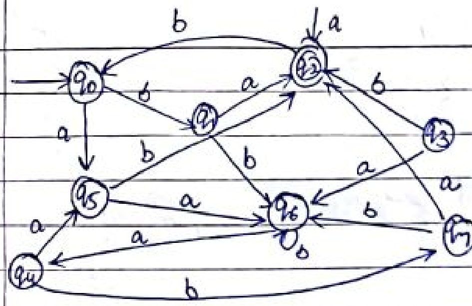
2







MINIMIZATION



	a	b
$\rightarrow q_0$	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

Steps :

1. First find inaccessible states i.e, which are not in table.
2. Now, divide initial and final states
3. Pick up non-final states and compares with each other, those states which are similare, delete it. Repeat this step until all states formed are not similar to each other.

1. Delete q_3 bcz it is not in table

	a	b
$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
$* q_2$	q_2	q_0
q_4	q_5	q_7
q_5	q_6	q_2
q_6	q_4	q_6
q_7	q_2	q_6

2 Now divide final & Non-final state

Non final

$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
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

	a	b
q_2	q_2	q_0

3 Compare the states which are similare and delete one. q_1 & q_7 are same delete q_7

	a	b
$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
q_4	q_5	~~q_7~~ q_1
q_5	q_6	q_2
q_6	q_4	q_6

final

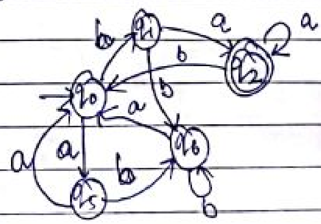
	a	b
q_2	q_2	q_0

4 Now again compare states & delete which are similar q_0 & q_4 are similare delete q_4

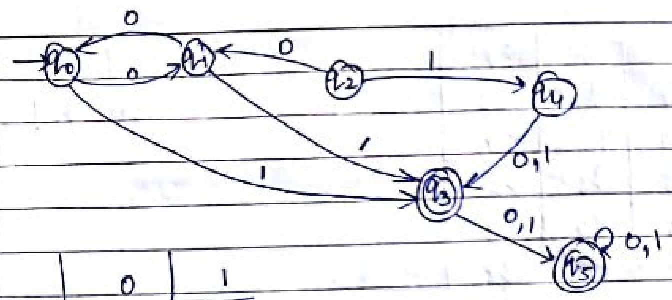
	a	b
$\rightarrow q_0$	q_5	q_1
q_1	q_2	q_6
q_5	q_6	q_2
q_6	q_0	q_6

final

	a	b
q_2	q_2	q_0



Q



	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3
~~q_2~~	~~q_1~~	~~q_4~~
* q_3	q_5	q_5
~~q_4~~	~~q_3~~	~~q_3~~
* q_5	q_5	q_5

Delete q_2
Delete q_4

Non-final

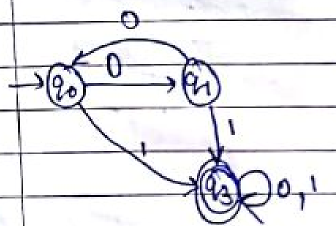
	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_0	q_3

final

	0	1
q_3	q_5	q_5
~~q_5~~	~~q_5~~	~~q_5~~

final

q_3	0	1
q_3	q_3	q_3



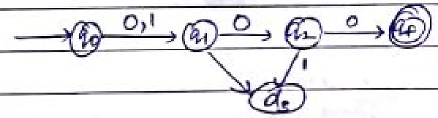
__/__/__

Regular Expressions:
Regular Expressions are useful for representing certain sets of strings, or certain subsets over sigma.

Regular Set: Any set represented by a regular expression is called a regular set.

Q Write a RE where for a language which accepts set of all strings of zeroes & ones ending in 00.

$(0+1)^*00$



Q Zeroes & 1's begining with zero and ending with 1

Solⁿ $0(0+1)^*1$

Q Give RE for language which accepts all strings begin & end with a & in b/w any word using b.

Solⁿ ~~$a(a+b)^*b(a+b)^*a$~~ $\quad a\ b^*\ a$ ✗

$[a b^* a + a]$

Q RE for a's & b's whose 5th symbol from right end is a.

$(a+b)^*abbbb$ ✗ $\quad (a+b)^*a(a+b)^4$ ✓

Scanned by CamScanner

Q RE to denote lang. which accept Strings any no. of a's is followed by any no. of b's followed by any no. of c

Soln $a^*b^*c^*$

Q2 which have atleast one a followed by atleast one b followed by atleast c

Soln $aa^*bb^*cc^*$ $\downarrow$ $a^+b^+c^+$

$aa^* \to a^+$

Chomsky

1. Type 0 (Phrase Structured Grammer):
In type 0, the right hand side rules are free from any restriction i.e, $\alpha \to \beta$ and $\alpha \neq \epsilon$, where $\alpha, \beta \to$ are in substantial form {$\epsilon \to$ Null}
eg: $AB \to aS$
$SB \to Sb$
$S \to \epsilon$
$ASB \to aASB$
In this, there is no terminal in LHS i.e, a, b

2. Type 1 (Restricted or Context Sensitive Grammer)
- For each production of the form $\alpha \to \beta$, the length of β is atleast as much as length of α except for $\alpha \to \epsilon$
- The $\alpha \to \epsilon$ is allowed only if the start symbol α doesn't appear RHS of any production.

- The term context sensitive is used bcz the grammer has production of the form
$\alpha_1 A \alpha_2 \to \alpha_1 B \alpha_2$
where the replacement of Non terminal A by B is allowed only if α_1 precceds A and α_2 succeds A, α_1 & α_2 may or may not be empty.

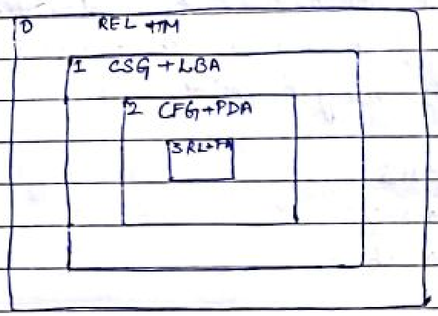
eg: $A \to AB$
$AB \to AC$
$AC \to ab$

3. Type 2 (Context free Grammer):
The only allowed type of production is $A \to \alpha$ where $A \to$ Non-terminal & $\alpha \to$ substantial form
- The LHS of production thus contains only one Non-terminal. eg: $S \to aSa|bSb|a|b$
aSa
aaSaa
aabSbaa

4. Type 3 (Regular
- The LHS of each production should contain only one non-terminal.
- The RHS can contain atmost one non-terminal which is allowed to appear as rightmost or leftmost symbol. These Languages

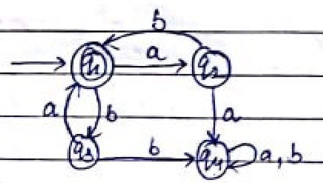
Scanned by CamScanner

The Regular lang. are denoted by simpler expression called Regular Expression.



Type 0 → Recursive Enumerable Language : Turning M/c.
Type 1 → Context Sensitive Grammer : Linear Bounded Automata
Type 2 → Context free Grammer : Push Down Automata
Type 3 → Regular Language : finite Automata (DFA & NDFA)

Q



ab
ba

$\cancel{(a+b)}(a+b) = (a+b)^2$

Mathematical Method

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

$q_2 \to q_1 a$

$q_3 \to q_1 b$

$q_4 \to q_2 a + q_3 b + q_4 a + q_4 b$

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

use q_2 & q_3

$q_1 = q_1 ab + q_1 ba + \epsilon$

$q_1 = q_1(ab+ba) + \epsilon$ { compare with 1 }

Arden's Theorem

$\boxed{R = Q + RP}$

$R = Q + RP$

$R = Q + (Q+RP)P$

$R = Q + QP + RP^2$

$R = Q + QP + (Q+RP)P^2$

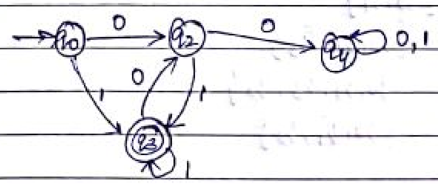
$R = Q + QP + QP^2 + QP^3 \ldots$

$R = Q(\epsilon + P + P^2 + \ldots)$

$R = QP^*$ — (1)

$\therefore \boxed{q_1 = \epsilon(ab+ba)^*}$ or $q_1 = (ab+ba)^*$

Q



$q_0 = \epsilon$

$q_2 = q_0 0 + q_3 0$

$q_3 = q_0 1 + q_2 1 + q_3 1$ $= \epsilon 1 + (q_0 0 + q_3 0)1 + q_3 0 = \epsilon 1 + q_0 01 + q_3 01 + q_3 0$

$q_3 = q_0 01 + q_3(01+0)$

$q_4 = q_2 0 + q_4 0 + q_4 1$

$q_3 = q_0 1 + (q_0 0 + q_3 0)1 + q_3 1$

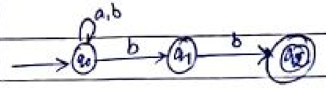
$= q_0 1 + q_0 01 + q_3 01 + q_3 1$

$= q_0\cancel{(1+01)} + q_0(01+1)q_3$

$\Rightarrow \cancel{(1+01)}(01+1)^* = (01+1)^*$

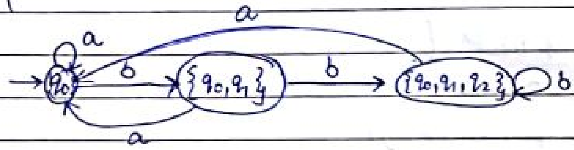
NFA to DFA

Q

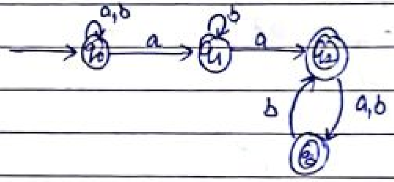


	a	b
$\rightarrow q_0$	q_0	q_0, q_1
q_1	-	q_2
q_2	-	-

	a	b
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0\}$	$\{q_0, q_1, q_2\}$

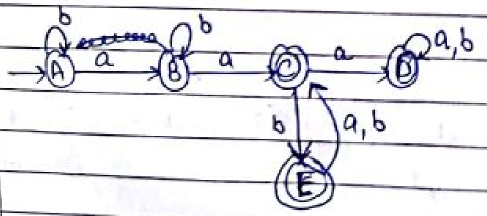


Q



	a	b
q_0	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	-	q_2

	a	b
A $\rightarrow q_0$	B $\{q_0, q_1\}$	A $\{q_0\}$
B $\{q_0, q_1\}$	C $\{q_0, q_1, q_2\}$	B $\{q_0, q_1\}$
C $\{q_0, q_1, q_2\}$	D $\{q_0, q_1, q_2, q_3\}$	E $\{q_0, q_1, q_3\}$
D $\{q_0, q_1, q_2, q_3\}$	D $\{q_0, q_1, q_2, q_3\}$	D $\{q_0, q_1, q_3, q_2\}$
E $\{q_0, q_1, q_3\}$	C $\{q_0, q_1, q_2\}$	C $\{q_0, q_1, q_2\}$

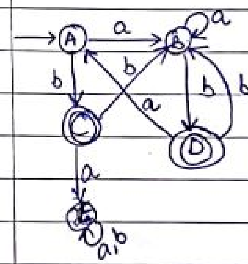


Q

	a	b
$\rightarrow q_0$	q_0, q_1	q_2
q_1	q_0	q_1
$* q_2$	-	q_0, q_1

$q_0, q_1, q_0.$

	a	b
A $\rightarrow q_0$	B $\{q_0, q_1\}$	C $\{q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	D $\{q_2, q_1\}$
$\{q_2\}$	$\{\phi\} \rightarrow$ E	$\{q_0, q_1\}$
$\{q_2, q_1\}$	$\{q_0\}$	$\{q_0, q_1\}$



Q Write a CFG for pallindrome no.'s (binary)

Soln
$A \to SAS$
$S \to 0S0/1S1/1/0/\epsilon$
$A|B|C$

Q $0^* 1 (0+1)^*$ $A = 0A/C$ 00101 0101

$0^* = 00$ $S \; B = A1C$ 001

$1 = 1$ $C = 0C/1C/\epsilon$

$(0+1)^* = 01, 1$

~~$0S1 0 1S/0/01/10/\epsilon$~~

A1B a → Non-terminal
A → Terminal

~~$S = 0ST/0$~~

$T = 1U$

$U = 01/10/\epsilon$

$\boxed{S = 0ST/1U/01/10/\epsilon/0}$ ×

~~$S \to 0ST/0/1/\epsilon$~~
~~$T \to 1U$~~
~~$U \to \epsilon/01/10$~~

✓ $S \to A1B$
$A \to 0A/\epsilon$
$B \to 0B/1B/\epsilon$

Q Write a CFG which generates having equal no.'s of a's & b's

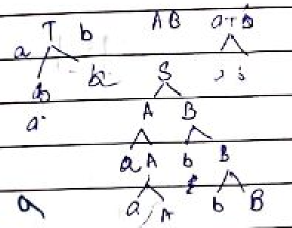
Soln ~~$S \to aSb/aSbS/bSaS/\epsilon$~~ $a_n = b_n$
$S \to aTb/\epsilon$ $S \to AB/\epsilon$
$T \to aU$ $A \to aA/ba$
$U \to bS$ $B \to A$

$S \rightarrow aTbU / bTaU / aUbT / bUaT / aT / bT / \epsilon$

$T \rightarrow aT$

$U \rightarrow b$

aabb



$S \rightarrow AB / BA / ATB / BTA / \epsilon$

$A \rightarrow aA$

$B \rightarrow bB$

$T \rightarrow ab / ba / \epsilon$

aabg

abbaba

$S \rightarrow AB / BA / ATA / BTB / ATB / BTA / \epsilon$

$A \rightarrow aA / \epsilon / bA$

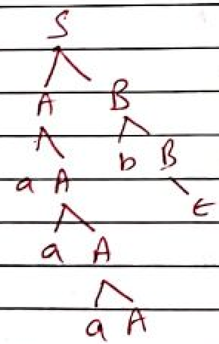
$B \rightarrow bB / \epsilon$

$T \rightarrow ab / A / B / ba / \epsilon$

a	b
b	a

abbaba

aAb



bbaa

abab

baba.

A aababb

$S = \epsilon / SASBS / SBSAS$

$A = a$

$B = b$

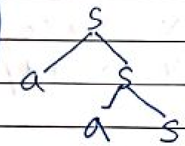
ASBS

BSAS

$S = ASBS / BSAS / \epsilon$

$A = a$

$B = b$



Q. Design a CFG that generates set of all strings with exactly one a.

Solⁿ ~~$A \rightarrow BaB$~~

$B \rightarrow bB / \epsilon$

$A \rightarrow BaB$

Q. $\{a^n b^m c^m d^n \mid n \geq 1, m \geq 1\}$

abcd

aabbccdd

aaabbbcccddd

$S \rightarrow aAbBcCdD$

$A \rightarrow aA / \epsilon$

$B \rightarrow bB / \epsilon$

$C \rightarrow cC / \epsilon$

$D \rightarrow dD / \epsilon$

abcd

$S \rightarrow aAbcMd$

$A \rightarrow adA / \epsilon$

$M \rightarrow bcM / \epsilon$

abcd

aabbccdd

$S \rightarrow adAbcM$

$S \rightarrow aSd \,/\, aAd$

$A \rightarrow bAC \,/\, bc$

$(ab)^n$ ab aabb

Q $L = \{a^n b^n c^m d^m \mid n \geq 1, m \geq 1\}$

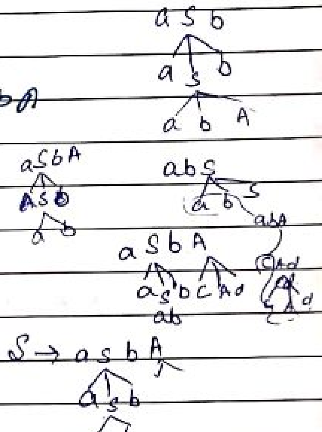
$S \rightarrow aSbA \,/\, abA \,/\, aSb$

$A \rightarrow cAd \,/\, cd$

$S \rightarrow XY$

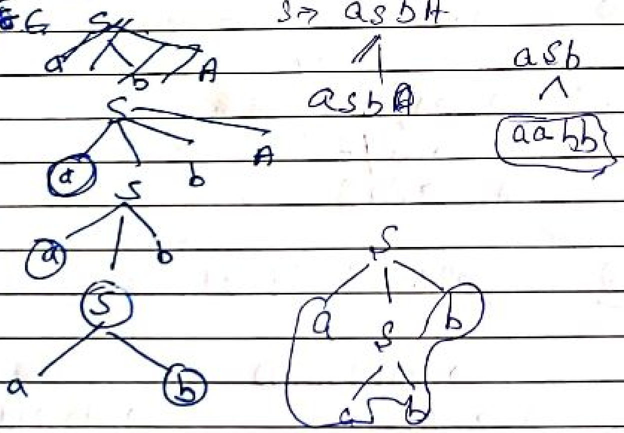
$X \rightarrow aXb \,/\, ab$

$Y \rightarrow cYd \,/\, cd$



$S \rightarrow aSbcAd \,/\, aSb \,/\, ab$

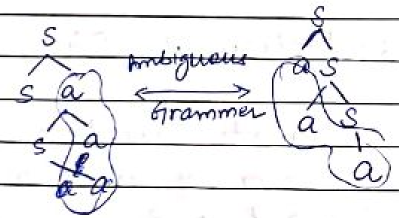
$A \rightarrow cAd \,/\, cd \,/\, \epsilon$



aaba

AMBIGUOUS GRAMMER

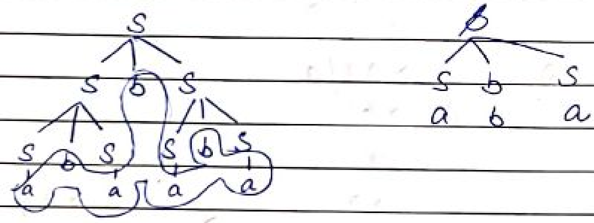
Q $S \rightarrow aS \,/\, Sa \,/\, a$



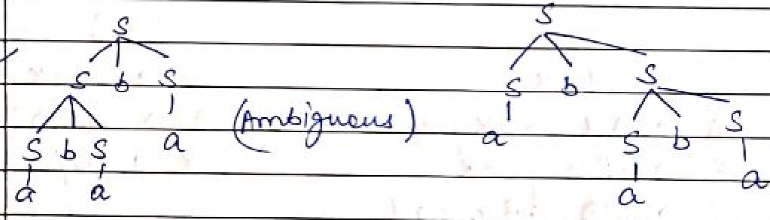
Ambiguous Grammer

aaa

Q $S \rightarrow SbS \,/\, a$



abababa



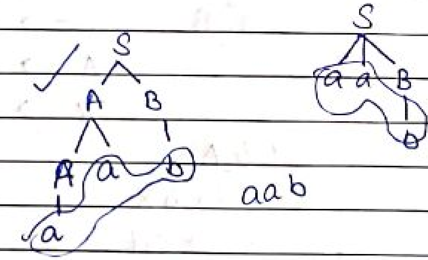
(Ambiguous)

ababa ababa

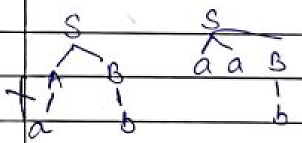
Q $S \rightarrow AB \,/\, aaB$

$A \rightarrow a \,/\, Aa$

$B \rightarrow b$



aab



Reduction of CFG

1. Eliminate useless symbol
2. Removal of unit production
3. Removal of ϵ

Q

$S \rightarrow AB / a$
$A \rightarrow b$

Elimate useless symbol

$S \rightarrow a$
$A \rightarrow b$ $\longrightarrow$ $\boxed{S \rightarrow a}$

Q

$S \rightarrow aB / bX$
$A \rightarrow BAd / bSX / a$
$B \rightarrow aSB / bBX$
$X \rightarrow SBD / aBx / ad$

$\longrightarrow$ Delete B

$S \rightarrow aB / bX$
$A \rightarrow BAd / bSX / a$
$X \rightarrow SBD / aBx / ad$

$S \rightarrow aB / bX$
aSB

Delete All that have B

$S \rightarrow bX$
$A \rightarrow bSX / a$
$X \rightarrow ad$

Delete A

$S \rightarrow bX$
$X \rightarrow ad$

Q

$A \rightarrow xyz / Xyzz$
$X \rightarrow Xz / xYz$
$Y \rightarrow yYy / Xz$
$Z \rightarrow Zy / z$

Delete X & Y

$A \rightarrow xyz$
$Z \rightarrow Zy / z$

Delete Z

$A \rightarrow xyz$

Q

$S \rightarrow aC / SB$
$A \rightarrow bSCa$
$B \rightarrow aSB / bBC$
$C \rightarrow aBC / ad$

Delete B

$S \rightarrow aC$
$A \rightarrow bSCa$
$C \rightarrow ad$

Delete A

$S \rightarrow aC$
$C \rightarrow ad$

2. Removal of Unit Production

$NT \longrightarrow NT$ (ie, Non-terminal produces Non-terminal)

Q

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow C / b$
$C \rightarrow D$
$D \rightarrow E$
$E \rightarrow a$

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow \cancel{C}a / b$
$C \rightarrow \cancel{D}\ a$
$D \rightarrow \cancel{E}\ a$
$E \rightarrow a$

X

$S \rightarrow AB$
$A \rightarrow a$
$B \rightarrow a/b$

Q $S \rightarrow A/bb$
$A \rightarrow B/b$
$B \rightarrow S/a$

$A \rightarrow B/b$
$A \rightarrow S/b/a$
$S \rightarrow S/a/b/bb$

$S \rightarrow B/b/bb$
$B \rightarrow S/a$
$\{S \rightarrow a/b/bb$ ✓$\}$ Ans

Q $I \rightarrow a/b/Ia/Ib/I_0/I_1$
$F \rightarrow I/(E)$
$T \rightarrow F/T*F$
$E \rightarrow T/E+T$

3. Removal of ϵ
$S \rightarrow aA$
$A \rightarrow b/\epsilon$
Sol $S \rightarrow aA/a$ & $A \rightarrow b$

Q $S \rightarrow aSa | bSb | \epsilon$
Sol^n $S \rightarrow aSa | bSb | aa | bb$

Q $S \rightarrow a | Xb | aYa$
$X \rightarrow Y | \epsilon$
$Y \rightarrow b | X$

Sol^n $S \rightarrow a | b | Xb | aYa | aa$
$X \rightarrow Y$
$Y \rightarrow b / X$

Q $S \rightarrow ABAC \{c / bc | abaC | baC | abC\}$
$A \rightarrow aA | \epsilon$
$B \rightarrow bB | \epsilon$
$C \rightarrow c$

Sol^n $S \rightarrow ABAC | c | BC |$ AAC
$A \rightarrow aA | a$
$B \rightarrow bB | b$
$C \rightarrow c$

✓ $S \rightarrow ABAC | BAC | C | AAC | ABC | BC | AC$
$A \rightarrow aA | a$
$B \rightarrow bB | a$
$C \rightarrow c$

Q $(a+b)^* bb (a+b)^*$
Sol $S \rightarrow AbbA$
$A \rightarrow aA | bA | \epsilon$

__/__/____

$S \to AbbA \mid bb \mid Abb \mid bbA$
$A \to aA \mid bA \mid a \mid b$

⇨ Chomsky Normal form (CNF)

- If a CFG has only production of the form
1. Non terminal → string of exactly two nonterminal or of the form
2. Non-terminal → one terminal

i.e, $S \to AA$
$S \to a$

eg: $S \to bA \mid aB$
$A \to bAA \mid aS \mid a$
$B \to aBB \mid bS \mid b$

Sol^n $S \to CA \mid DB$
$C \to b$
$D \to a$
$A \to \cancel{CA}X \mid DS \mid a$
$B \to \cancel{DB}DY \mid CS \mid b$
$X \to AA$
$Y \to BB$

eg $S \to 1A \mid 0B$
$A \to 1AA \mid 0S \mid 0$
$B \to 0BB \mid 1$

__/__/____

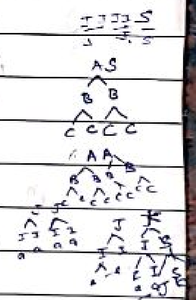
Sol^n $S \to XA \mid YB$
$A \to XC \mid YS \mid 0$
$B \to YD \mid 1$
$X \to 1$
$Y \to 0$
$C \to AA$
$D \to BB$

Q (Eg 7.11 Pandey) Design a CFG for a language $L = \{ a^{4n} \mid n \geq 1 \}$ and Convert that CFG into CNF

Sol^n $L = a^4$ if $n = 1$ aaaa
$= a^8$ if $n = 2$ aaaa aaaa

$S \to aaaa \mid aaaaS$	$S \to JJ \mid IK$
$S \to II \mid IJ$	$I \to a$
$I \to aa$	$J \to II$
$J \to aaS$	$K \to JS$

OR ~~S → XASY~~



Greibach Normal form (GNF)

$A_i \longrightarrow a\gamma$

$A_i \longrightarrow A_g\gamma$ where, $g > i$ and $\gamma \in V_n^*$

Q. $S \to AB \mid BC$
$A \to aB \mid bA \mid a$
$B \to bB \mid cC \mid b$
$C \to c$

Sol^n $S \to aBB \mid bAB \mid aB \mid bBC \mid cCC \mid bC$
$A \to aB \mid bA \mid a$
$B \to bB \mid cC \mid b$
$C \to c$

Q. $S \to AB$
$A \to aA \mid bB \mid b$
$B \to b$

Sol^n $S \to aAB \mid bBB \mid bB$
$A \to aA \mid bB \mid b$
$B \to b$

Q. $S \to aAS \mid a$
$A \to SbA \mid SS \mid ba$

$AS \to B$
$aB \mid a$
$A \to SbA$
$bA \to C$
$S \to aBC$
$bD \to$
$D \to R$

Submission - 11 April

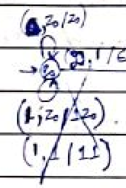
Assignment

1. Convert the Grammer to GNF
$A_1 \to A_2A_3$ $A_2 \to A_3A_1 \mid b$ $A_3 \to A_1A_2 \mid a$

2. Design a PDA M to accept $L = \{0^n 1^m 2^m \mid n, m \geq 1\}$

3. Design a CFG $L = \{0^m 1^n \mid m \neq n\}$

$A_1 \to A_3A_3A_1 \mid bA_3$
$A_2 \to aA_1 \mid b \mid A_4A_2A_1$



3. $S \to 0A1B \mid 0A11B \mid 0 \mid 1$
$A \to 0A \mid \epsilon$
$B \to 1B \mid \epsilon$

$S \to A \mid B$
$A \to 0A1 \mid A1 \mid 1$
$B \to 0B1 \mid 0B \mid 0$

1. $A_3 \to A_1A_2 \mid a$ $A_1 \to A_2A_3$ $A_2 \to A_3A_1 \mid b$
Put A_1
$A_3 \to A_2A_3A_2 \mid a$
Put A_2
$A_3 \to A_3\underbrace{A_1A_3A_2}_{\alpha} \mid \underbrace{bA_3A_2}_{\beta} \mid \underbrace{a}_{\gamma}$

$A_3 \to bA_3A_2Z \mid aZ \mid bA_3A_2 \mid a$
$Z \to A_1A_3A_2Z \mid A_1A_3A_2$
$A_2 \to bA_3A_2ZA_1 \mid aZA_1 \mid bA_3A_2A_1 \mid aA_1 \mid b$
$A_1 \to bA_3A_2ZA_1A_3 \mid aZA_1A_3 \mid bA_3A_2A_1A_3 \mid aA_1A_3 \mid bA_3$

2.

_/_/_

PDA (Push Down Automata)

$((q,a,z),(p,y))$

This represents that PDA is in state q with z on the top of stack may read 'a' from input tape replace z by y on the top of stack and enters state p.

$((q,a,\epsilon)(p,a))$ → push

$((q,a,z),(p,\epsilon))$ → pop

Q Design a PDA which accepts the language
$L = \{w \in \{a,b\}^* \mid w$ has equal no. of a's & b's $\}$
$L = a^n b^n$

~~((q,a,z)e(q,a))~~
1. $((s,a,\epsilon),(q,c))$
2. $((s,b,\epsilon),(q,c))$
3. $((q,a,c),(q,ac))$
4. $((q,b,c),(q,bc))$
5. $((q,a,b),(q,\epsilon))$
6. $((q,b,a),(q,\epsilon))$
7. $((q,\#,c),(f,\epsilon))$
8. $((q,a,a),(q,aa))$
9. $((q,b,b),(q,bb))$

_/_/_

S.No.	State	unread input	stack	Transition use
1	S	ababbbaa	ε	1
	q	babbbaa		

Ques $L = \{a^n b^{2n} : n > 0\}$

abb aabbbb
aaa bbbb

1. $((S,a,\epsilon),(q,c))$
2. $((q,a,c),(q,ac))$
3. $((q,a,a),(q,aa))$
4. $((q,b,a),(q,\epsilon))$
5. $((q,b,c),(q,c))$
6. $((q,\#,c),(f,\epsilon))$

aabbbb
abb

a; s; st;

__/__/____

$0^n 1^m 2^m \quad n, m \geq 1$

$0 \; 1 \; 1 \; 2 \; 2$

$(q_0, 0, \epsilon), (q_1, 0)$
$(q_1, 0, 0), (q_1, 0)$
$(q_1, 1, 1), (q_1, 1)$

__/__/____

Turing Machine

$TM = \{Q, \Sigma, \Gamma, \delta, q_0, h\}$

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ finite set of non-blank info. symbols

$\Gamma \rightarrow$ set of tape symbols including the blank symbol.

$\delta \rightarrow$ Next move partial function

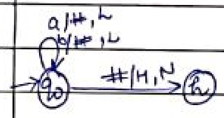
$$(Q \times F) = (Q \times \Gamma \times (L, R, N))$$

$q_0 \rightarrow$ Initial state

$h \rightarrow$ halt state

Q. Design a turing m/c which acts as an eraser i.e, it erases all non-blank symbols on the tape where the sequence of all non-blank symbols doesn't contain any blank symbol # in between

Sol^n



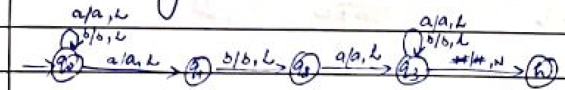
$(q_0, a) \longrightarrow (q_0, \#, L)$

$(q_0, b) \longrightarrow (q_0, \#, L)$

$(q_0, \#) \longrightarrow (h, \#, N)$

Q Design a turing m/c that accepts the language of all string which contains aba as a substring

Sol^n

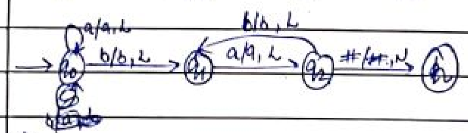


#	a	a	b	a	a

aba

ababa

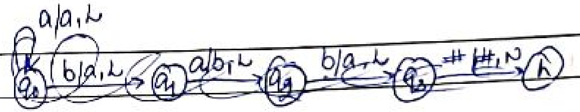
a/a



abab

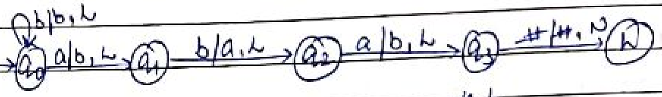
bababa

ababa /___

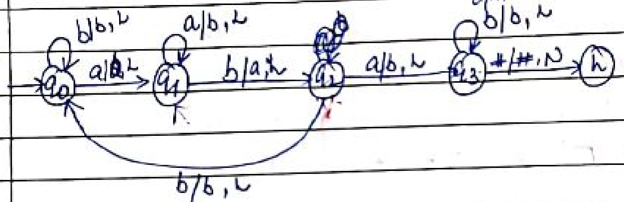


bbababb

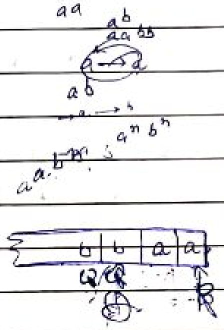
abbabb



abba



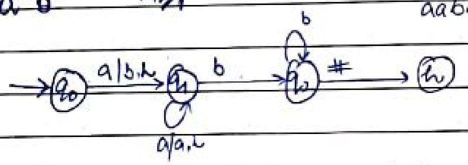
abbaba

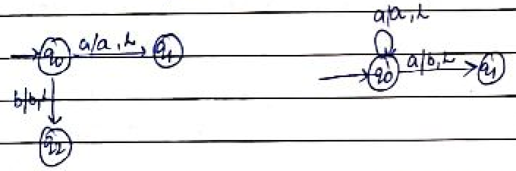


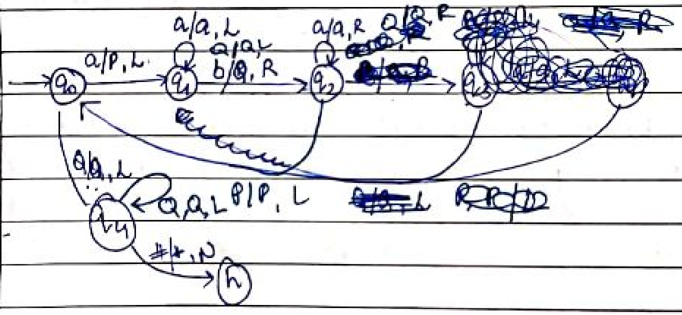
Q $a^n b^n$ $n \geq 1$

ab

aabb







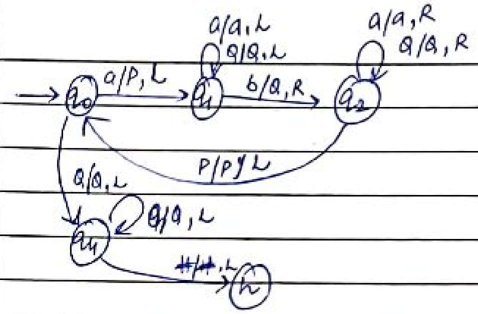
__/__/__

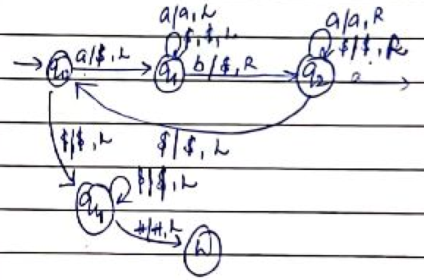
abab

abab

bbaa

abab





babbaa

bb baaa

q0

q1 b b b a a $

q2 b b $ a a $

q0 b b $ a $ $

q1 b $ $ a $ $

$ a

a/a, R $/$ R