① Peter Linz (PHI)
② Hopcroft (Pearson)

PAGE NO.
DATE: / /

Theory of Computation

Automata

(~~Check~~ set of rules to ~~for~~ ~~understand~~ compute)

- → Finite automata → simplest, min. processi
- → Push Down automata → (FA + stack)
- → Linear Bounded automata (FA + finite length tape
- → Turing Machine (FA + infinite length tape)

→ if not by turing machine then problem can not be computed.

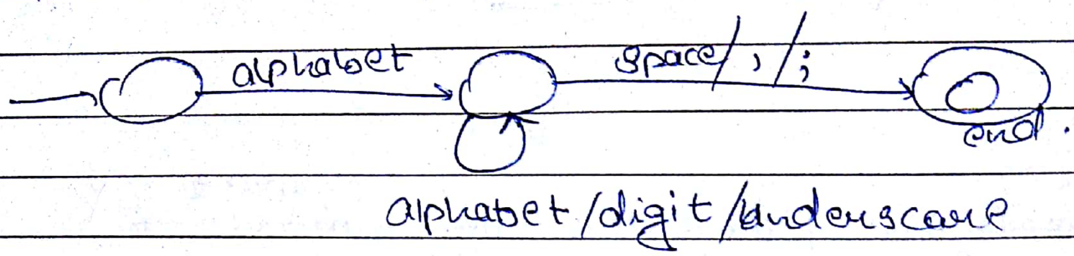
FINITE AUTOMATA

- is 5-tuple $(Q, \Sigma, \delta, q_0, F)$

Q → set of state
Σ → Input symbol
δ → transition function
q_0 → initial state
F → set of final states

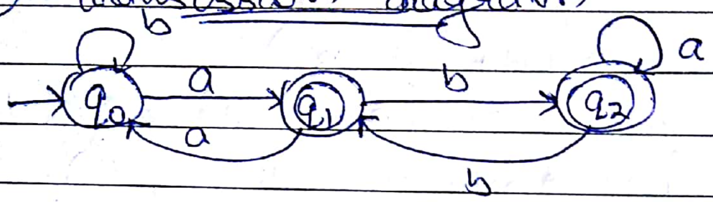
PAGE NO. | DATE : / /

to check identifier



→ ~~States represent~~

① transmission diagram



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

q_0

$F = \{q_1, q_2\}$

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

Initial	→
Final	○

② transition table

	a	b
→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_2	q_1

or ③ transition functions

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

$\delta(q_0, b) \rightarrow q_0$

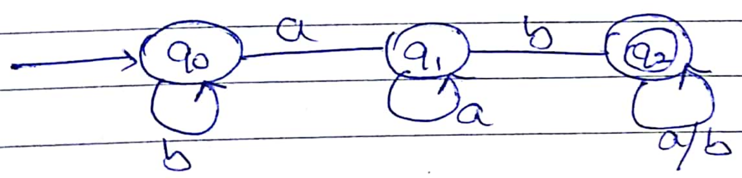
$\delta(q_1, b) - q_2$
$\delta(q_1, a) - q_0$
$\delta(q_2, a) - q_2$
$\delta(q_2, b) - q_1$

Q. Design a FA over (a, b) that accepts all string containing ab as substring

Deterministic FA (DFA)

→ move for all symbol
→ only one move per symbol
→ move not defined are dead moves

ba	baab	abab	aaa	bbb
×	✓	✓	×	×



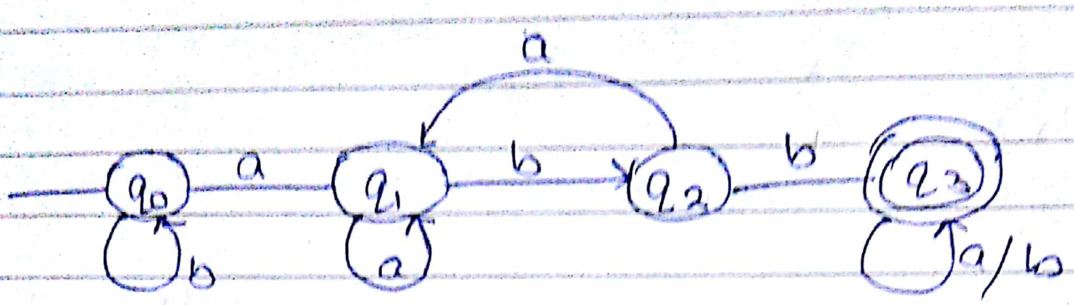
Acceptance in DFA is defined by final state
– String aba

$(q_0, aba) \vdash (q_1, ba) \vdash (q_2, a) \vdash (q_2, \lambda)$ ✓

String baa

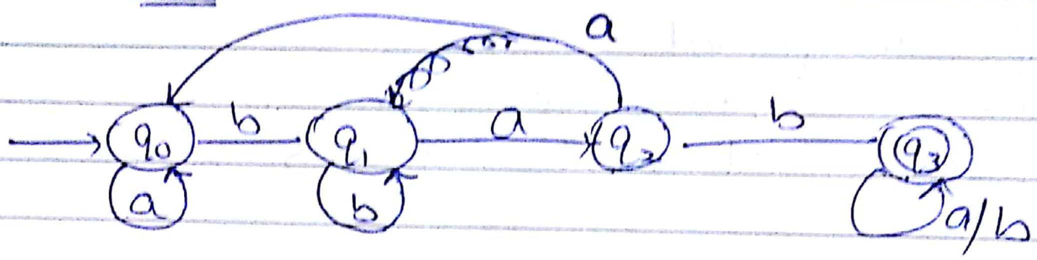
$(q_0, baa) \vdash (q_0, aa) \vdash (q_1, a) \vdash (q_1, \lambda)$ ×

Q abb as substring



ab a

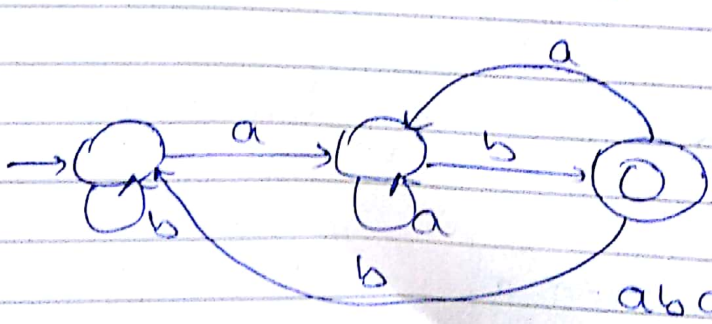
Q bab



bab

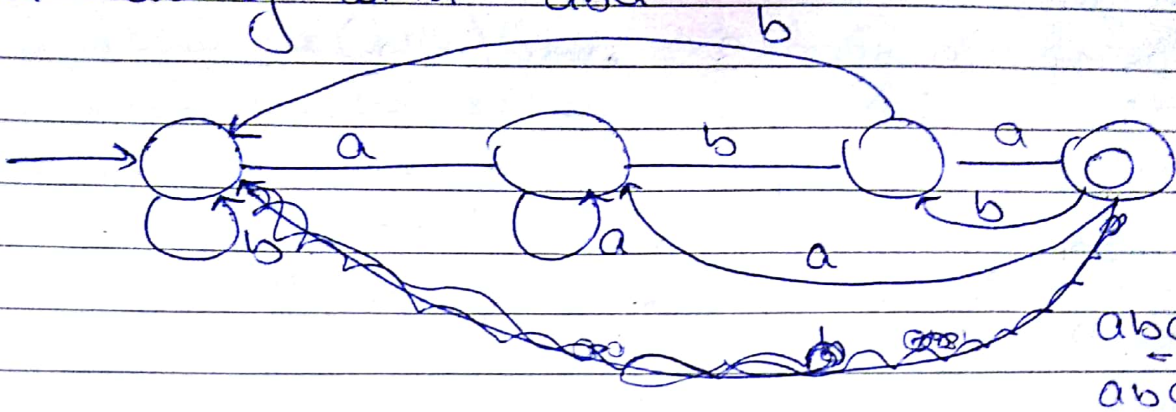
Q. ending with ab

babb ✗ bbab ✓ ab ✓



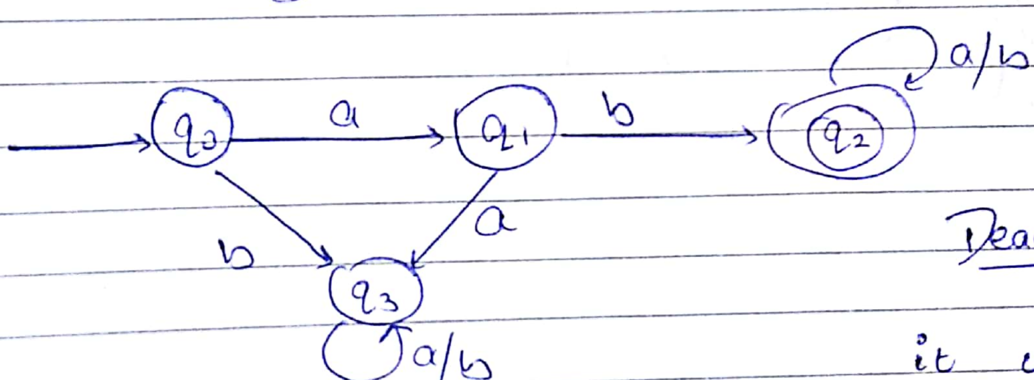
aba
abb

Q ending with aba



abab a
aba aba

Q. Starting with ab



Dead state

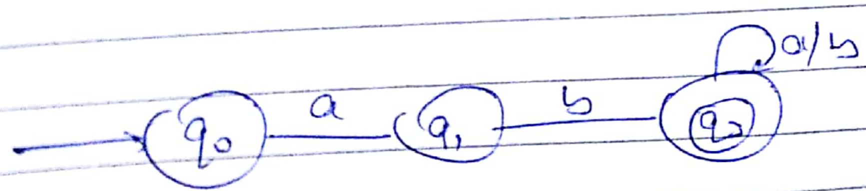
it is state which has no path outgoing to other state

$(q_d, a) \vdash (q_d \ \forall a \in \Sigma)$

$(q_0, aab) \vdash (q_1, ab) \vdash (q_3, b) \vdash (q_3, \wedge)$

dead

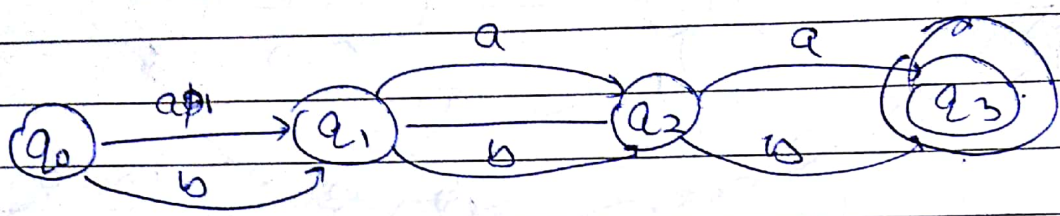
X



aab
baa
aba

PAGE NO.
DATE : / /

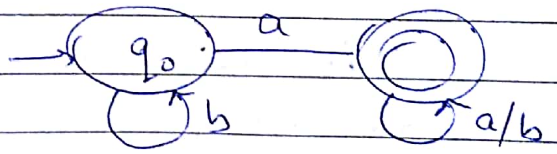
Q. Design an automata which accepts all strings of length 3 over (a, b)



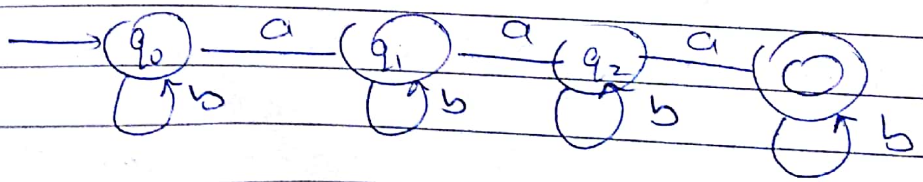
Q. draw DFA that accept string that contains

(1) atleast one a
(2) exactly 3 a
(3) exactly 2 a and 1 b

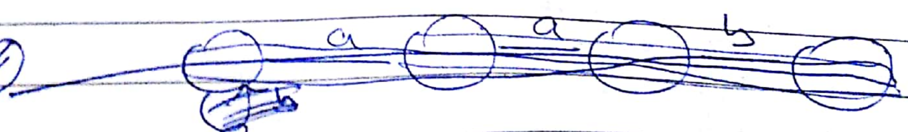
(1)



(2)



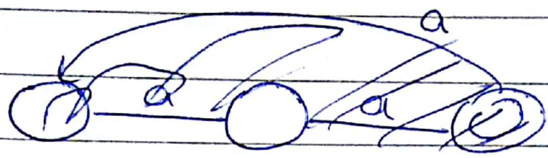
(3)



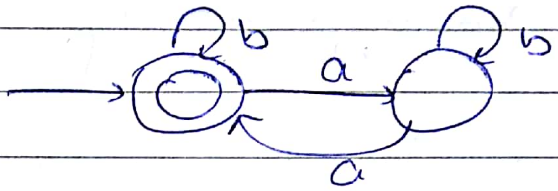
Q. $\Sigma = \{a, b\}$

Draw a FA that accept all string with even no. of a.

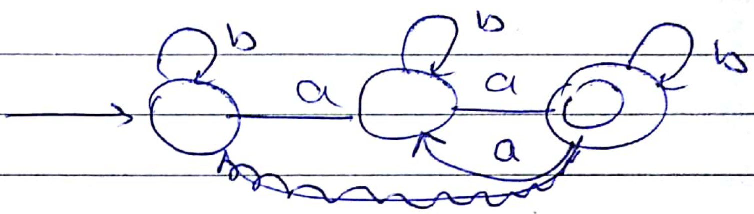
~~$n_a \geq 0$~~



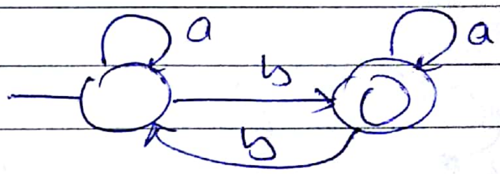
$n_a \geq 0$

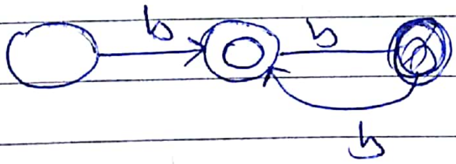


$n_a > 0$



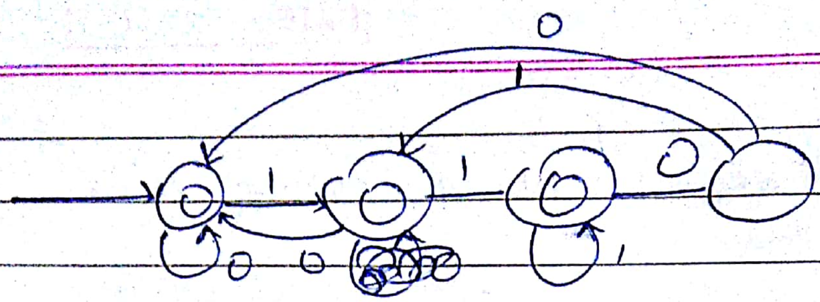
Q. no. of b are odd



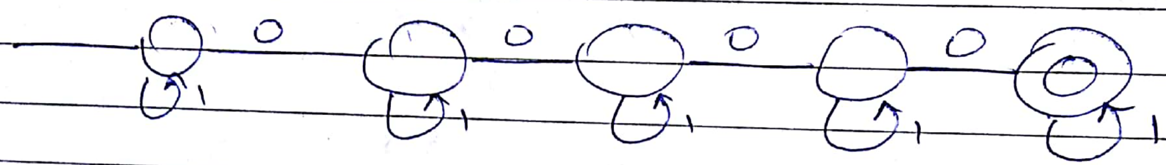


Q all string that does not ~~contain~~ ending 110 as substring

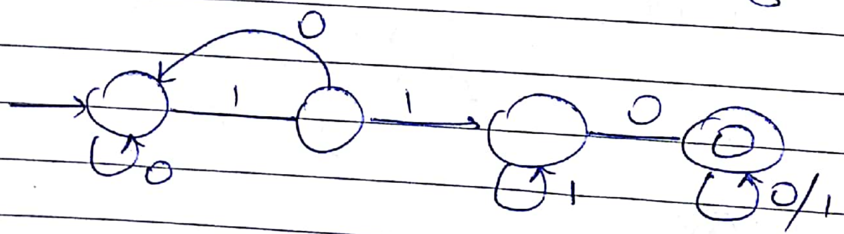
complement (non final as final & final as non final)



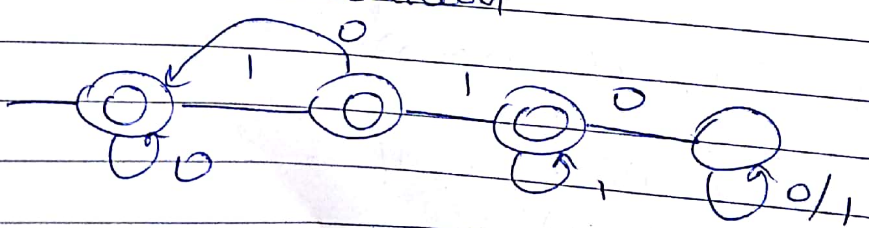
Q. all string containg exactly 4 0's



Q. Contains 110 as string

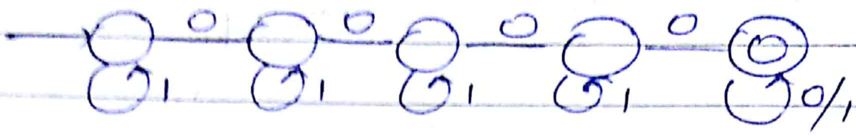


does not contain

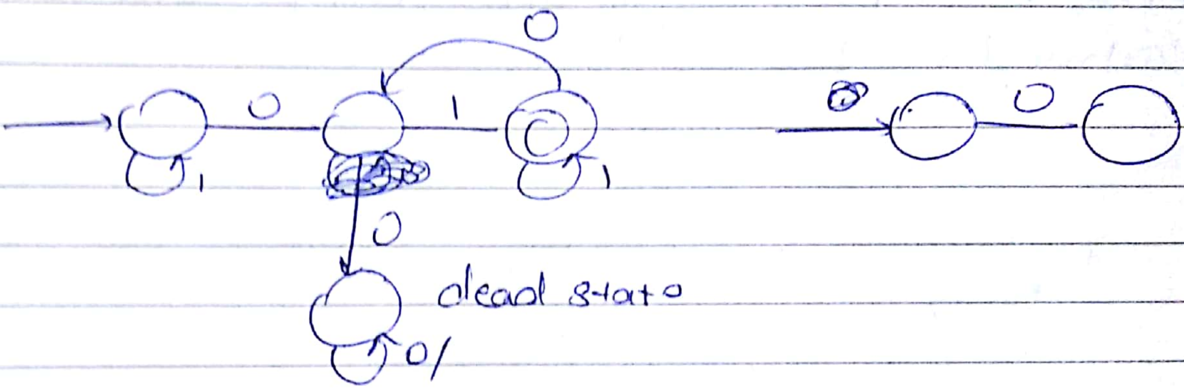


Scanned by CamScanner

Q. atleast 4 zeroes

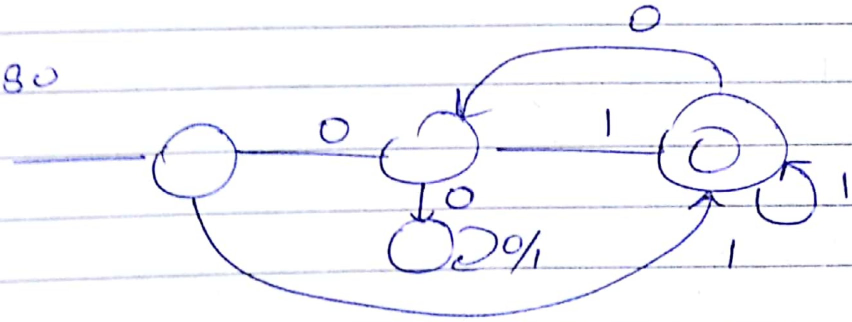


Q. Each occurance of 0 is followed by 1

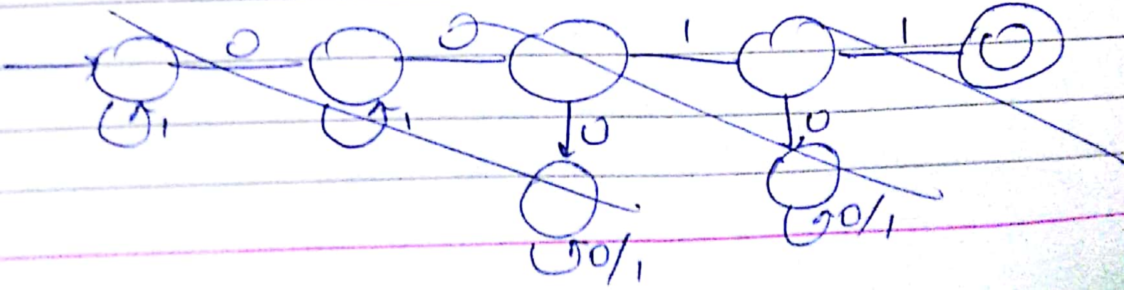


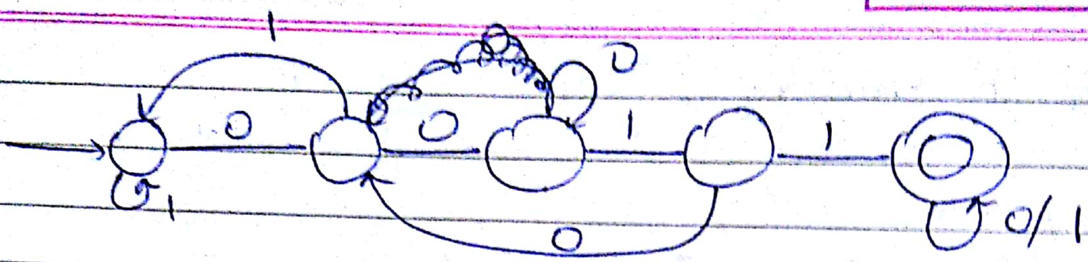
but if 1111 comes it is acceptable either make initial as final or

so

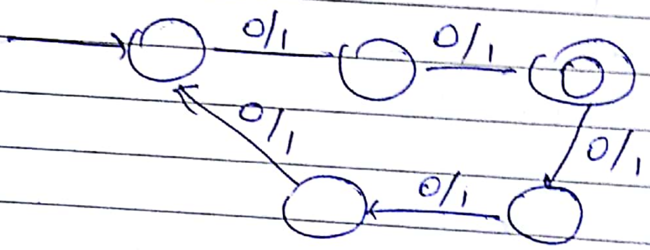


Q. FA accept string that contains (at least one) two consecutive 0 followed by two consecutive 1

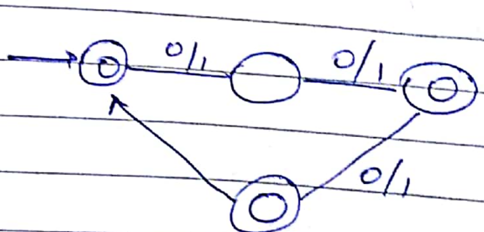




Q. FA that accept string on {0,1} where $|w| \bmod 5 = 2$

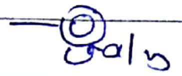


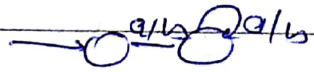
Q. $|w| \bmod 4 \neq 1$



Q.1 what will minimum number of state in FA to accept a string of length atleast n

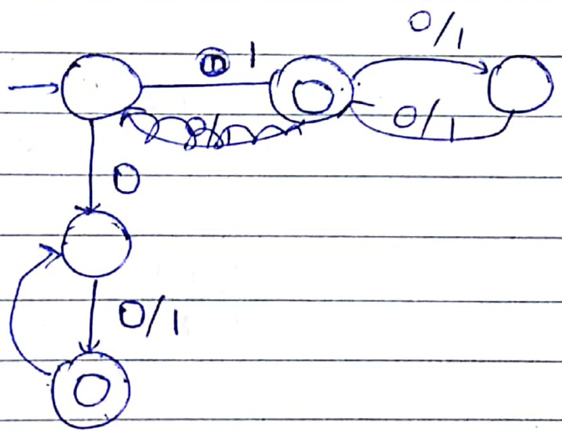
(A) n (B) n+1 (C) n-1 (D) NOT

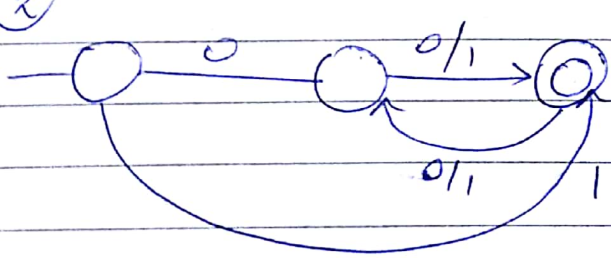
n=0 

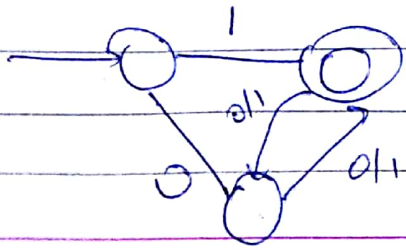
n=1 

Q2. Draw a FA over {0,1} if at start with 0 has even length and if start with 1 has odd length.

Ans:

① 

② 

③ 

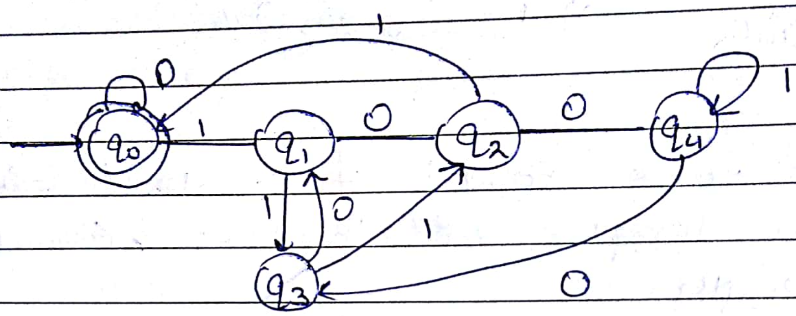
Scanned by CamScanner

5

110 10

PAGE NO.
DATE : / /

Q. Design a FA accept all binary string divisible by 5



10 2 %5 = 2
11 3 %5 = 3

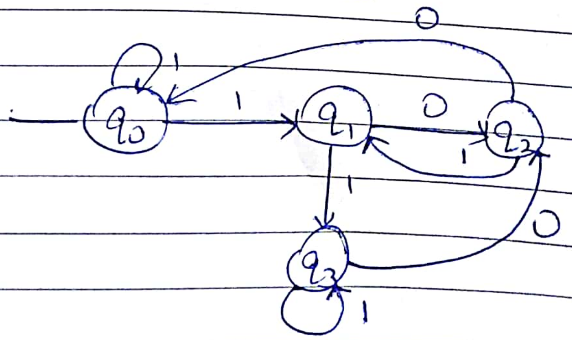
q_0 - remainder 0
q_1 - " 1
q_2 - " 2
q_3 - " 3
q_4 - " 4

100 4%5 = 4
101 5 %5 = 5

110 6 %5 = 1
111 7 %5 = 2

1000 8%5 = 3
1001 9%5 = 4

Q. divisible by 4



10
11

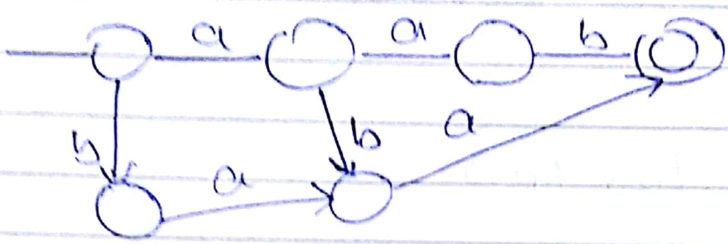
100
1

110
111
111

PAGE NO.
DATE : / /

10
100
1000

Q. exactly 2 a & 1 b



Q.1 all binary string having decimal values as 2^n.

Q2 number of a % 3 ≥ 1 $n_a \% 3 \geq 1$

Q3. equal occurrence of 01 and 10

Q4.

	a	b
$\rightarrow q_0$	$q_1 q_2$	q_2
q_1	q_2	$q_1 q_3$
q_2	q_0	$q_1 q_0$
q_3	q_2	$q_1 q_2$

Q5.

	0	1	2
$\rightarrow q_0$	$q_1 q_4$	q_4	$q_2 q_3$
q_1		q_4	
q_2	q_2		q_3
q_3		$q_4 q_1$	
q_4	q_1	q_2	$q_1 q_2$

PAGE NO.
DATE : / /

Non-Deterministic finite Automata

→ not confurmed with every state

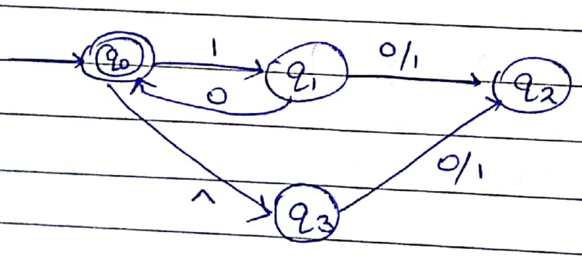
DFA $(q_i, a) \vdash q_j$

NDFA $(q_i, a) \vdash \{q_j, q_k, q_p\}$

5 tubles $(Q, \Sigma, q_0, \delta, F)$

$\delta : Q \times (\Sigma \cup \wedge) \vdash 2^Q$

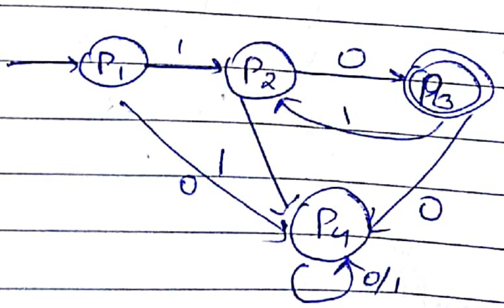
Example :-



$(10)^x \quad x > 0$

NDFA version

FA - 1

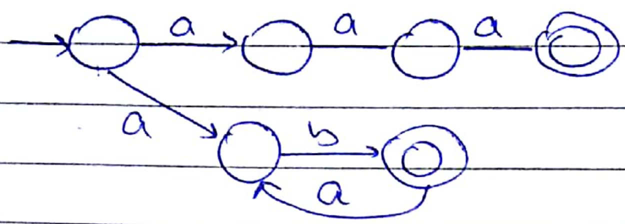


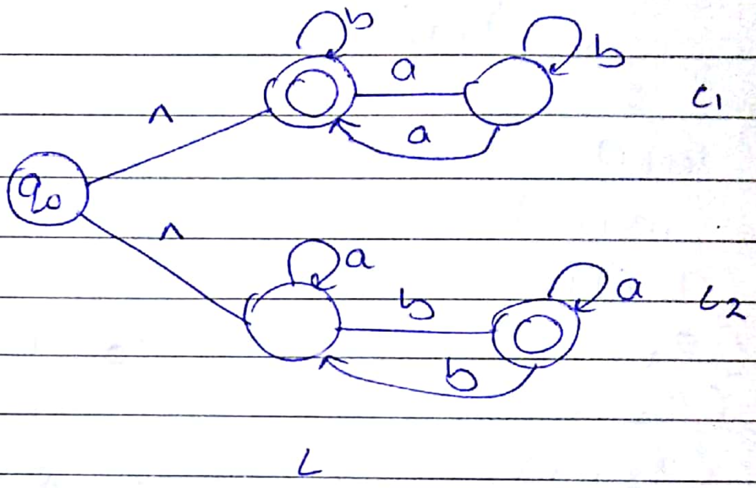
$(10)^x \quad x > 0$

DFA version

FA - 1

$a^3 \cup (ab)^x$





* $L = L_1 \cup L_2$

L_1 :- even a

L_2 :- odd b

just combine both of them

→ more flexible

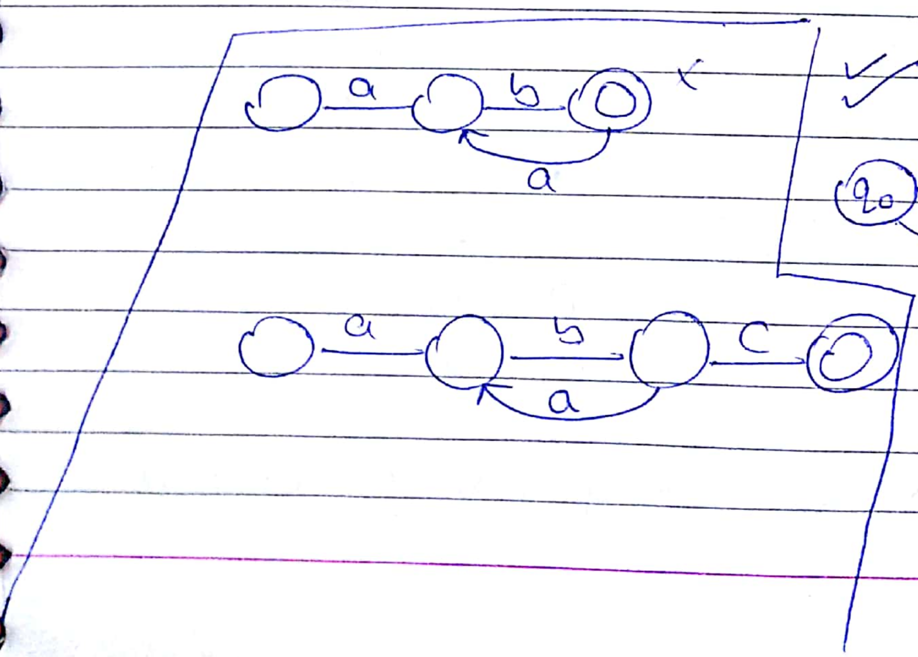
Q. Draw a NFA

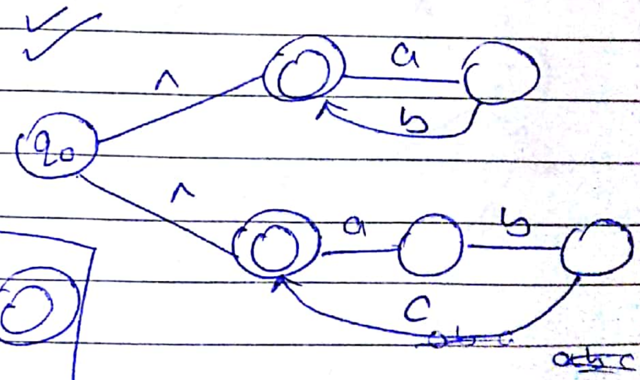
$L = \{ab, abc\}^x$, $x > 0$

abab ✓ abcax

abcab ✓

abcabc ✓

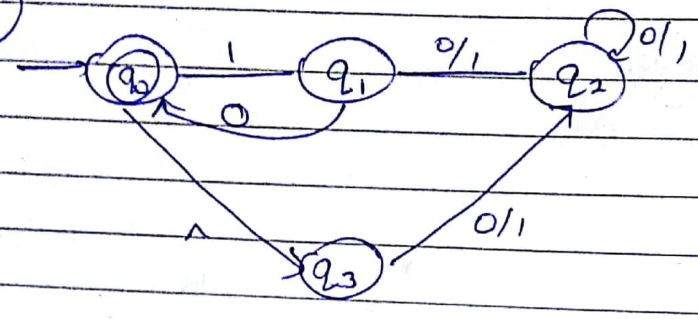




PAGE NO.
DATE : / /

* Acceptance of String

(1)



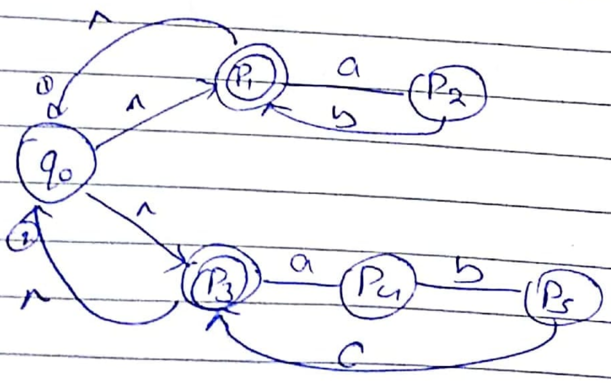
1010

$(q_0, 1010)$
$\vdash (q_1, 010)$
$\vdash (q_2, 10)$
$\vdash (q_2, 0)$
$\vdash (q_2, \Lambda)$

$(q_0, 10)$
$\vdash (q_1, 0)$
$\vdash (q_2, \Lambda)$ (q_0, Λ) ✓

out of 3 one is correct so string is accepted.

(2)



$(q_0, abab)$

$P_1 (abab)$
P_2, bab
P_1, ab
P_2, b
P_1, Λ

for abcab → more flexibility (1) & (2)

Convert NDFA to DFA

Let $M = (Q, \Sigma, q_0, F, \delta)$ be NDFA
and $M' = (Q', \Sigma', q_0', F', \delta')$ be corresponding DFA

→ $\Sigma' = \Sigma$

→ State of DFA are defined as $[q_1, q_2 \ldots q_i]$ where $q_1, q_2 \ldots q_i$ are states of NDFA

ex. NDFA $Q = \{q_0, q_1, q_2\}$

$$
\begin{array}{llll}
DFA = & [q_0] & [q_0, q_1] & [q_1, q_2] \\
 & [q_1] & [q_0, q_2] & \\
 & [q_2] & [q_0, q_1, q_2] &
\end{array}
$$

→ $q_0' = [q_0]$

→ F' = ~~is~~ state $[q_0, q_1, \ldots q_n]$ ~~where~~ which contains at least one final state of NDFA

→ $\delta'([q_0, q_1 \ldots q_i], a) \vdash [q_0 q_1 \ldots q_i]$
iff $\delta(q_0, a) \cup \delta(q_1, a) \cup \ldots \delta(q_i, a)$
$= \{q_1, q_2 \ldots q_i\}$

Q.

NDFA

	a	b
→q_0	q_1	$q_0 q_1$
q_1	q_0	q_1

DFA

	a	b
→[q_0]	[q_1]	[$q_0 q_1$]
[q_1]	[q_0]	[q_1]
[q_0, q_1]	[$q_1 q_0$]	[$q_0 q_1$]

Q.

NDFA

	a	b
→q_0	$q_0 q_1$	q_2
q_1	q_0	q_1
q_2		$q_0 q_1$

DFA

	a	b
→[q_0]	[$q_0 q_1$]	[q_2]
[$q_0 q_1$]	[$q_0 q_1$]	[$q_2 q_1$]
[q_2]		[$q_0 q_1$]
[$q_2 q_1$]	[q_0]	[$q_0 q_1$]

Q

NDFA

	a	b
→q_0	$q_1 q_2$	q_2
q_1	q_2	$q_1 q_3$
q_2	q_0	$q_1 q_0$
q_3	q_2	$q_1 q_2$

DFA

	a	b
→[q_0]	[$q_1 q_2$]	[q_2]
[$q_1 q_2$]	[$q_2 q_0$]	[$q_1 q_3 q_0$]
[q_2]	[q_0]	[$q_1 q_0$]
[$q_2 q_0$]	[$q_0 q_1 q_2$]	[$q_1 q_0 q_2$]
[$q_1 q_0 q_3$]	[$q_1 q_2$]	[$q_2 q_1 q_3$]
[$q_1 q_0$]	[$q_2 q_1$]	[$q_1 q_3 q_2$]
[$q_0 q_1 q_2$]	[$q_1 q_2 q_0$]	[$q_2 q_1 q_3 q_0$]
[$q_1 q_2 q_3$]	[$q_0 q_2$]	[$q_1 q_3 q_0 q_2$]
[$q_0 q_1 q_2 q_3$]	[$q_1 q_2 q_0$]	[$q_2 q_1 q_3 q_0$]
~~[$q_0 q_2$]~~		

NDFA

	0	1	2
$\rightarrow q_0$	$q_1 q_4$	q_4	$q_2 q_3$
q_1		q_4	
q_2	q_2		q_3
q_3		$q_4 q_1$	
q_4	q_1	q_2	$q_1 q_2$

DFA

	0	1	2
$[q_0]$	$[q_1 q_4]$	$[q_4]$	$[q_2 q_3]$
$[q_1 q_4]$	$[q_1]$	$[q_4 q_2]$	$[q_1 q_2]$
$[q_4]$	$[q_1]$	$[q_2]$	$[q_1 q_2]$
$[q_2 q_3]$	$[q_2]$	$[q_4 q_1]$	$[q_3]$
$[q_1]$		$[q_4]$	
$[q_4 q_2]$	$[q_1 q_2]$	$[q_2]$	$[q_1 q_2 q_3]$
$[q_1 q_2]$	$[q_2]$	$[q_4]$	$[q_3]$
$[q_2]$	$[q_2]$	$\cancel{[q_4]}$	$[q_3]$
$[q_3]$		$[q_4 q_1]$	
$[q_1 q_2 q_3]$	$[q_2]$	$[q_4 q_1]$	$[q_3]$

PAGE NO.
DATE : / /

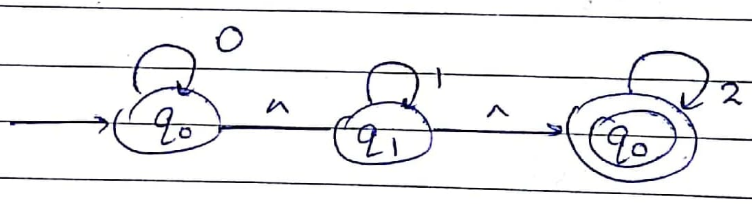
* NFA - $\wedge$ move
  ↓
  NFA
  ↓
  DFA
  ↓
  minimize DFA

*. Removal of $\wedge$-move

$q_1 \rightarrow q_2 \qquad \delta(q_1, \wedge) \rightarrow q_2$

then following steps are performed to remove same

1. Duplicate all moves from q_2 to q_1 also
2. if q_2 is FS make q_1 as FS



	0	1	2	$\wedge$
q_0	q_0		[illegible]	$\cancel{q_1}$
q_1		q_1	[illegible]	$\cancel{q_2}$
q_2			q_2	

copy

copy

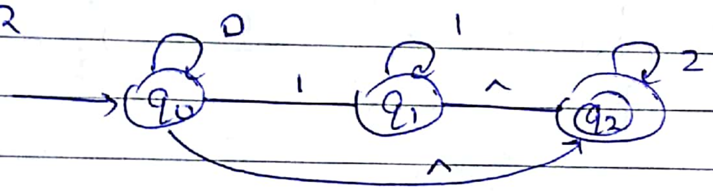
① $q_0 \xrightarrow{\wedge} q_1$

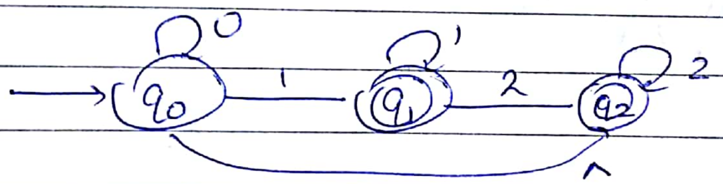
② $q_1 \xrightarrow{\wedge} q_2$

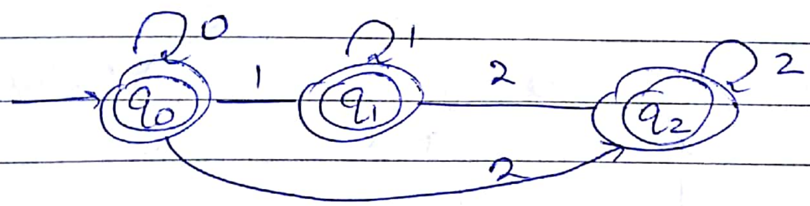
③ $q_0 \rightarrow q_2$

	0	1	2	^
→ q_0	q_0	q_1	q_2	$\cancel{q_2}$
q_1		q_1	q_2	
q_2			q_2	

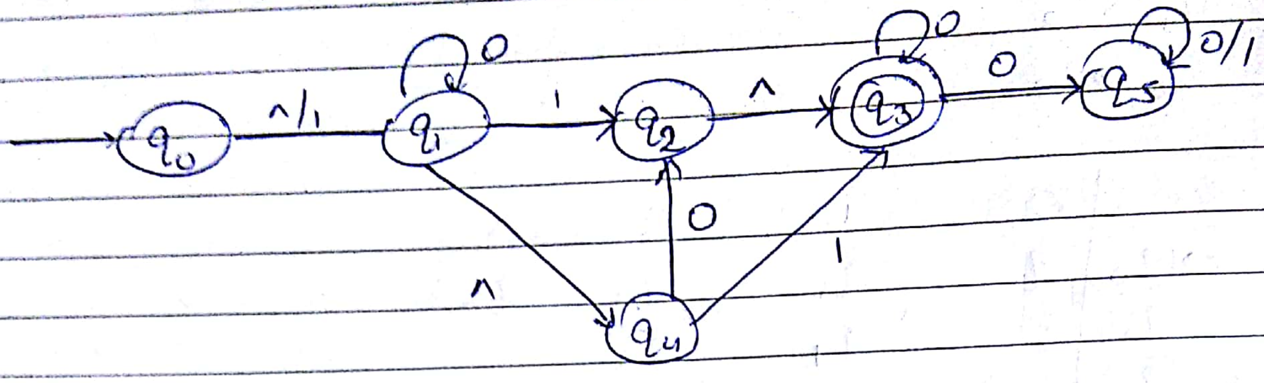
OR

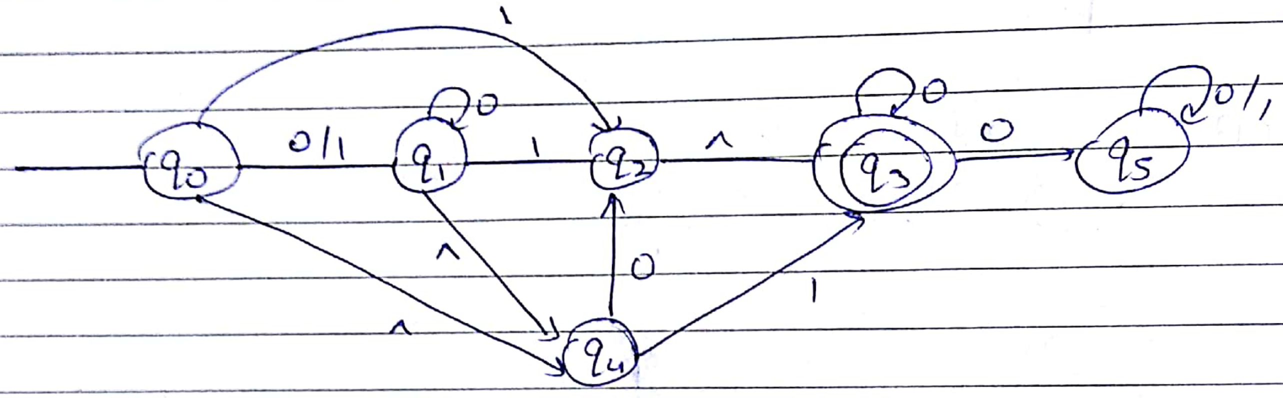


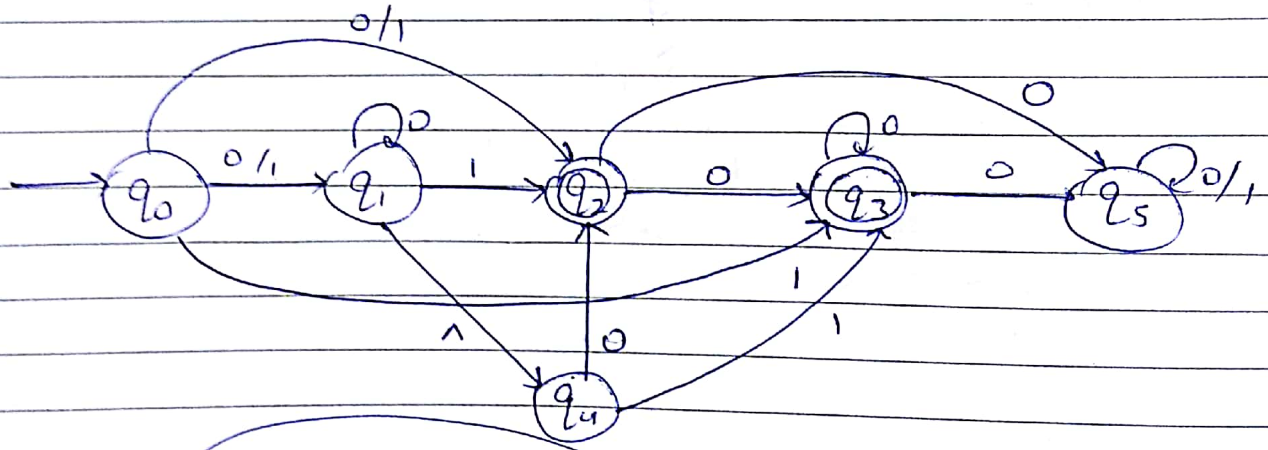


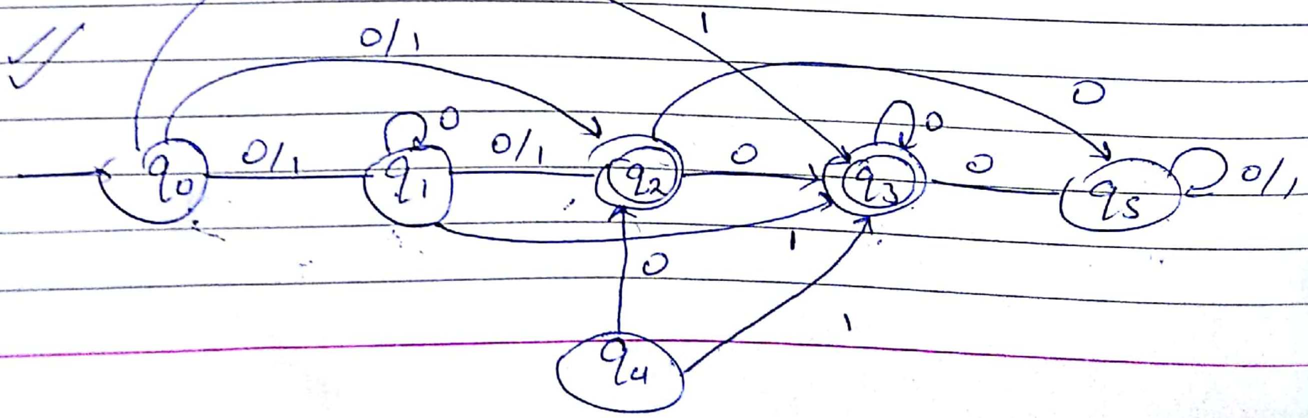


Q:

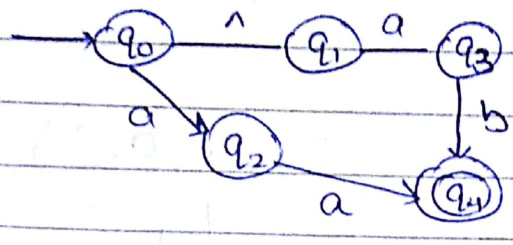








*



ab

club the ^ moves as one

(q_0, q_1)

$\downarrow a$

$(q_2\ q_3)$

$\downarrow b$

(q_4)

$\Downarrow$ FS ✓ String is accepted (no need to backtrack)

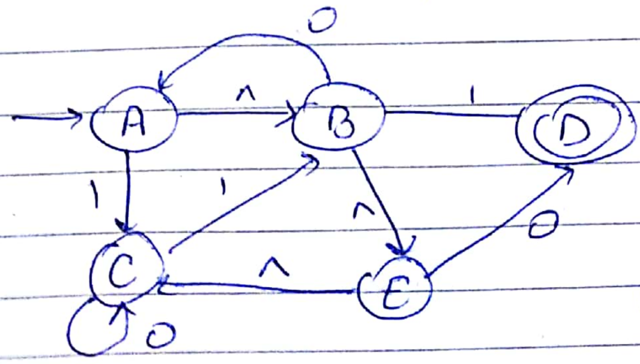
→ Λ-closure(q)

① $q \in \Lambda\text{-closure}(q)$

② if $\delta(q, \Lambda) \rightarrow p$ then $p \in \Lambda\text{-close}(q)$

③ $\forall\ r \in \Lambda\text{-closure}(q)$ if $\delta(r, \Lambda) \rightarrow s$ then $s \in \Lambda\text{-closure}(q)$

④ Repeat step 3 till no new state is added in Λ-closure(q)



$\Lambda\text{-closure}(A) = (ABEC)$

$\delta(A, 01) \rightarrow \Lambda\text{-closure}(A) = (ABEC)$

$\downarrow 0$

$(ACDBE) \xleftarrow{\Lambda\text{-clo}} (ACD)$

$\downarrow 1$

$(CBD) \xrightarrow{\Lambda\text{-clos}} \boxed{(CBDE)}$

$\delta(B, 110)$

$\Lambda\text{-closure}(B) \quad (BEC)$

$\downarrow 1$

$(DB) \rightarrow \Lambda\text{-closure} \rightarrow (DBEC)$

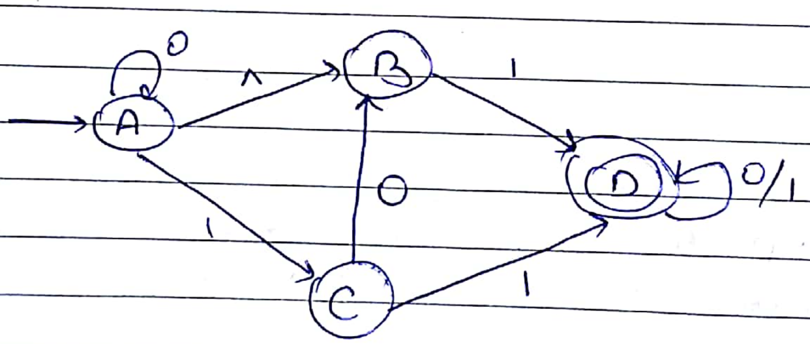
$\downarrow 1$

$(DBEC) \xleftarrow{\Lambda\text{-closure}} (DB)$

$\downarrow 0$

$(ADC) \xrightarrow{\Lambda \text{ closure}} \boxed{ADCBE}$

Q.



1. find $\delta(A, 10)$
2. Remove Λ-move
3. Convert NFA to DFA

① $\delta(A,10) \longrightarrow \Lambda\text{-Clos}(A)$ (AB)

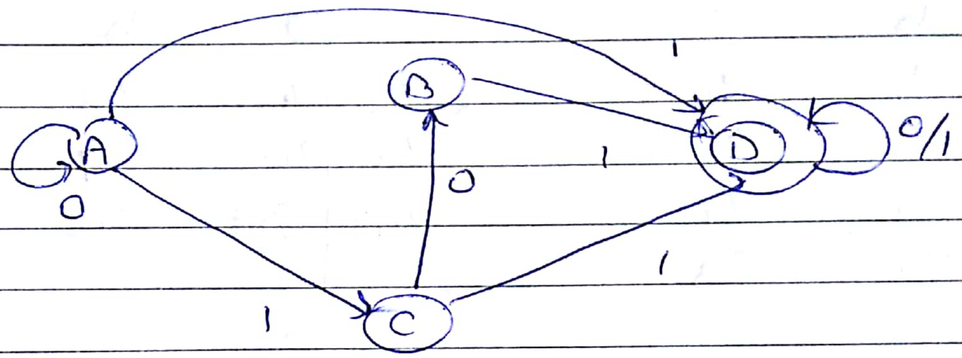
$\downarrow 1$

(CD) $\xrightarrow{\Lambda\text{-clo}}$ (CD)

$\downarrow 0$

(BD) $\xrightarrow{\Lambda\text{-clo}}$ [BD]

②



③

NFA

	0	1
→A	A	CD
B		D
C	B	D
D	D	D

DFA

	0	1
→[A]	[A]	[CD]
[CD]	[BD]	[D]
[BD]	[D]	[D]
[D]	[D]	[D]

Minimization of DFA

- state equivalence

two states q_1 & q_2 are termed as equivalent state iff $\delta(q_1, a) = F$ then $\delta(q_2, a) \in F$ $\forall a \in \Sigma$
& if $\delta(q_1, a) \in NF$ then $\delta(q_2, a) \in NF$

	a	b	c
$\rightarrow q_1$	q_2	q_1	q_4
q_2	q_2	q_1	q_4
q_3	q_2	q_4	q_4
q_4	q_2	q_1	q_3

$\Rightarrow$

	a	b	c
q_{12}	q_{12}	q_{12}	q_4
q_3	q_{12}	q_4	q_4
q_4	q_{12}	q_{12}	q_3

$F \neq NF$

* method
  1. Remove all dead and unreachable state
  2. Divide set Q into two sets P_1 and P_2 consisting of final states
  3. Now for each state q_i in P_i check for state equivalence. If the state in set are not equivalent split the set
  4. Repeat step3 till all the states in set are equivalent

PAGE NO.
DATE : / /

	a	b
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

X

(1) $q_0\ q_1\ q_5\ q_2\ q_6\ q_4\ q_7$

→ q_3 is unreachable → remove this

→ if final state is unreachable ⇒ Automata accepts nothing $L = \phi$

(2) $P_1(q_0, q_1, q_4, q_5, q_6, q_7)$ $P_2(q_2)$

$P_3(q_0, q_4, q_6)$ $P_4(q_1, q_5, q_7)$

$P_5(q_0, q_4)$ $P_6(q_6)$ $P_7(q_1, q_7)$ $P_8(q_5)$

	P_3	P_4
$q_0\ a \rightarrow P_1$	$(q_0, a) - P_4$	$q_1, a \rightarrow P_6$
$q_0\ b \rightarrow P_1$	$(q_0\ b) - P_4$	$q_1, b \rightarrow P_2$
$q_1\ a \rightarrow P_1$	$(q_4, a) - P_4$	$q_5, a \rightarrow P_2$
$q_1\ b \rightarrow P_2$	$(q_4, b) - P_4$	$q_5, b \rightarrow P_6$
$q_4\ a \rightarrow P_1$	$(q_6, a) - P_3$	$q_7, a - P_6$
$q_4\ b \rightarrow P_1$	$(q_6, b) - P_3$	$q_7, b - P_2$
$q_5\ a \rightarrow P_2$		
$q_5\ b \rightarrow P_1$		

P_5

$q_0, a \rightarrow P_7$ $q_4, a \rightarrow P_7$

$q_0, b \rightarrow P_8$ $q_4, b \rightarrow P_8$

P_7

$q_1\ a \rightarrow P_6$ $q_7, a \rightarrow P_6$

$q_1, b \rightarrow P_2$ $q_7, b \rightarrow P_2$

$q_7\ a \rightarrow P_1$ $q_7\ b\ P_1$

	a	b
$\rightarrow P_5$	P_7	P_8
P_6	P_6	P_5
P_7	P_6	P_2
P_8	P_2	P_6
P_2	P_5	P_2

Q.

	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6	q_5	q_6
q_7	q_6	q_3

× unreachable

① $q_0\ q_1\ q_2\ q_3$

② $P_1(q_0, q_1, q_2)$ $\qquad$ $P_2(q_3)$

$P_3(q_0, q_1)$ $\qquad$ $P_4(q_2)$

$P_5(q_0)$ $\qquad$ $P_6(q_1)$

PAGE NO.
DATE : / /

P_3

$q_0\, a \to P_1$	$q_0\, a \to P_3$
$q_0\, b \to P_1$	$q_0\, b \to P_3$
$q_1\, a \to P_1$	$q_1\, a \to P_3$
$q_1\, b \to P_1$	$q_1\, b \to P_4$
$q_2\, a \to P_2$	
$q_2\, b \to P_1$	

	a	b
$\to P_5$	P_6	P_5
P_6	P_5	P_4
P_4	P_2	P_6
P_2	P_2	P_5

Q.

	a	b
$\to q_0$	q_0	$q_0 q_2$
q_1	q_3	$q_2 q_3$
q_2	q_4	-
q_3	q_4	-
q_4	-	-

NFA
↓
DFA

① $q_0\ q_2\ q_4$

	a	b
$[q_0]$	$[q_0]$	$[q_0 q_2]$
$[q_0 q_2]$	$[q_0 q_4]$	$[q_0 q_2]$
$[q_0 q_4]$	$[q_0]$	$[q_0 q_2]$

Scanned by CamScanner

	a	b
R_1	R_1	R_2
R_2	R_3	R_2
R_3	R_1	R_2

$P_1(R_1, R_2)$ $\quad$ $P_2(R_3)$

$R_1\,a \rightarrow R_1$ $\quad$ $R_2\,a \rightarrow R_3$

$R_1\,b \rightarrow R_3$ $\quad$ $R_2\,b \rightarrow R_2$

R_1 $\quad$ R_2

Regular Expression

→ it is an algebric way to represent language accepted by FA

→ it contains 3 symbols
($+$, $\cdot$, $*$)

$+$ → union

$\cdot$ → concetenation

$*$ → Kleene clausure (any no. of times repeatation)

a^* → {0 or more occurance of a}

$a^* = \{\wedge, a, aa, \ldots\}$

$a+b$

$a \cdot b$

$(a+b)^* = \{\wedge, a, b, ab, aa, ba, bb, bab, aaa, \ldots$

$(a+b)(a+b)(a+b)\ldots$

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

(1) L = accept all string containing abb as substring

$$\underbrace{(a+b)^* \; abb \; (a+b)^*}_{\text{not concerned}}$$

(2) L = String starting with abb

$abb(a+b)^*$

(3) L = String containg even no. of a

$b^*(a.a)^* b^*$ × here aba is not accepta

$(b^* a b^* a b^*)^*$ not include only b

$b^*(b^* a b^* a b^*)^* b^*$

or

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

(4) L = all string containg at least 3 a's

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

or

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

or

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

PAGE NO.
DATE : / /

(5) L = exactly 3 a's

$b^* a\, b^* a\, b^* a\, b^*$

(6) L = containing atleast a pair of consecutive a's

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

(7) L = containing no pair of consecutive a's

$(ab+b)^* a$ or ~~$(b+ab)^* + a$~~

→ not able to end with a

~~(8) L =~~

* $a \cdot b + c$

$ab+c$ — $\{ab, c\}$

$a(b+c)$ — $\{ab, ac\}$

precedence need to be defined

Star	*
Concatenation	.
union	+

Q. $L = a^{2n} b^{2m+1} abb \quad n>0 \quad m>0$

$r1 : (aa)^* (bb)^* b\, abb$ => not L (coz $n, m > 0$)

$r2 : aa(aa)^* \, bb(bb)^* b\, abb$

$r3 : (aa)^+ (bb)^+ b\, abb$

] equivalent

positive closure
↑

$a^+ = \{a, aa, aaa, \ldots\}$

$a^* = \{\wedge, a, aa, \ldots\}$

$\cancel{\wedge + a}$ $\boxed{\wedge + a^+ = a^*}$

Kleene Theorem

→ for every regular expression there exist a FA to accept

* Language
↓
R.E
↓
$\wedge$, NFA
↓
NFA
↓
DFA
↓
minimize DFA

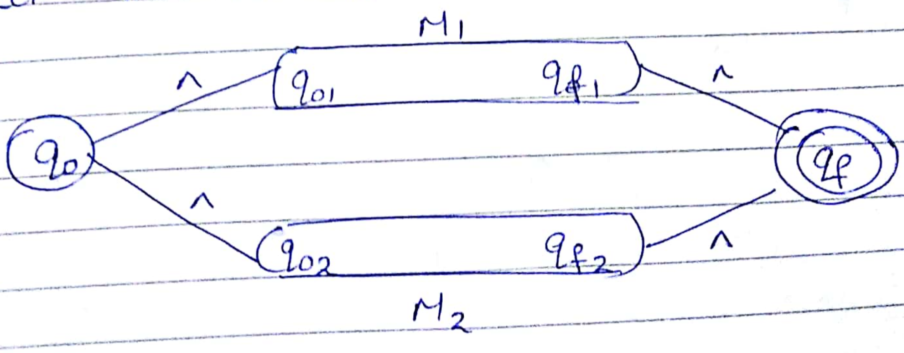
Proof: $R = \phi \to q_0 \quad q_1$

$R = \wedge \qquad \to q_0 \xrightarrow{\wedge} q_1$

$R = a \qquad \to q_0 \xrightarrow{a} q_1$

R_1 & R_2 are two R.E. & corresponding FA for them are $M_1(Q_1, q_{01}, \Sigma_1, \delta_1, q_{f_1})$ & $M_2(Q_2, q_{02}, \Sigma_2, \delta_2, q_{f_2})$

(i) Let $R = R_1 + R_2$



for R => $M(Q, \Sigma, \delta, q_0, q_f)$

$Q = Q_1 \cup Q_2 \cup \{q_0, q_f\}$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\delta = \delta_1 \cup \delta_2 \cup$

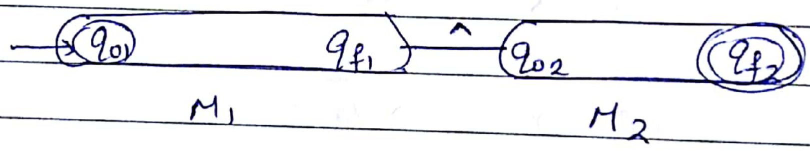
$(q_0, \wedge) \rightarrow \{q_{01}, q_{02}\}$

$\cup\ (q_{f_1}, \wedge) \rightarrow q_f$

$\cup\ (q_{f_2}, \wedge) \rightarrow q_f$

hence for + union FA exist

(ii) let $R = R_1 . R_2$



$M(Q, \Sigma, \delta, q_0, q_f)$

$Q = Q_1 \cup Q_2$

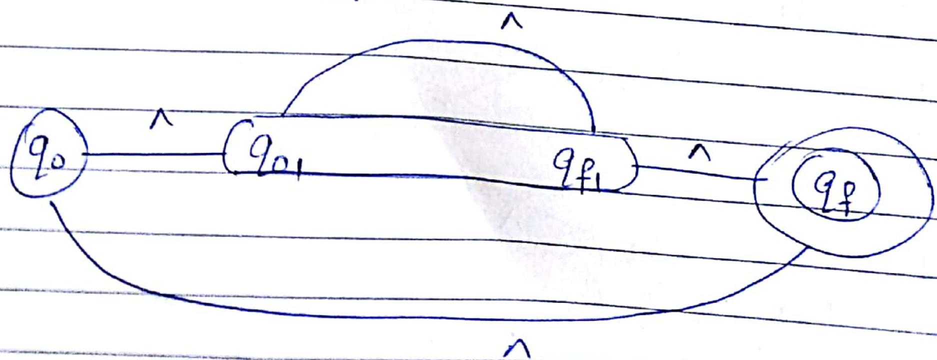
$\Sigma = \Sigma_1 \cup \Sigma_2$

$q_0 = q_{01}$

$q_f = q_{f_2}$

$\delta = \delta_1 \cup \delta_2 \cup (q_{f_1}, \wedge) \rightarrow q_{02}$

(iii) let $R = R_1^*$



PAGE NO.
DATE : / /

$Q = Q_1 \cup \{q_0, q_f\}$

q_0

q_f

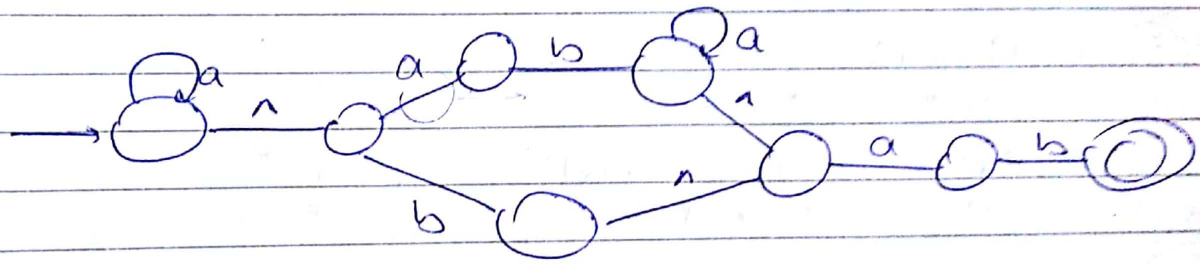
$\Sigma = \Sigma_1$

$\delta = \delta_1 \cup (q_0, \wedge) \rightarrow \{q_{01}, q_f\} \cup (q_{f1}, \wedge) \rightarrow \{q_f, q_{01}\}$

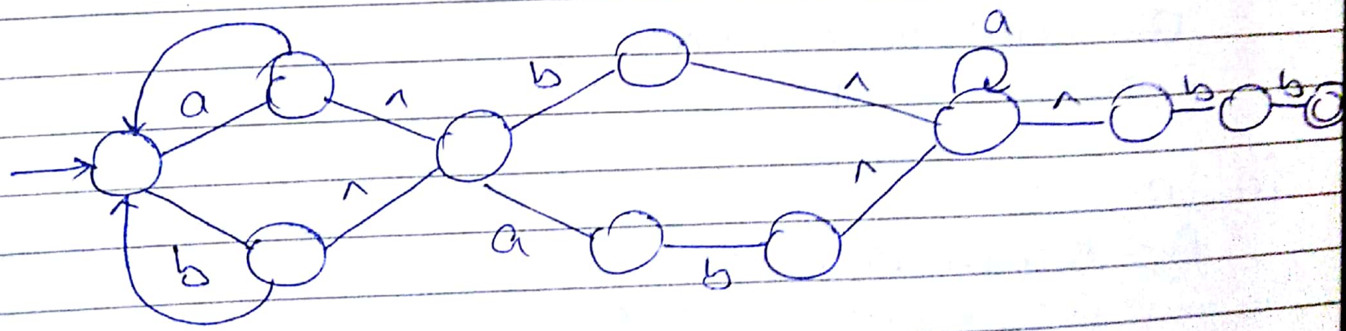
~~so hence can accept RE can form FA~~

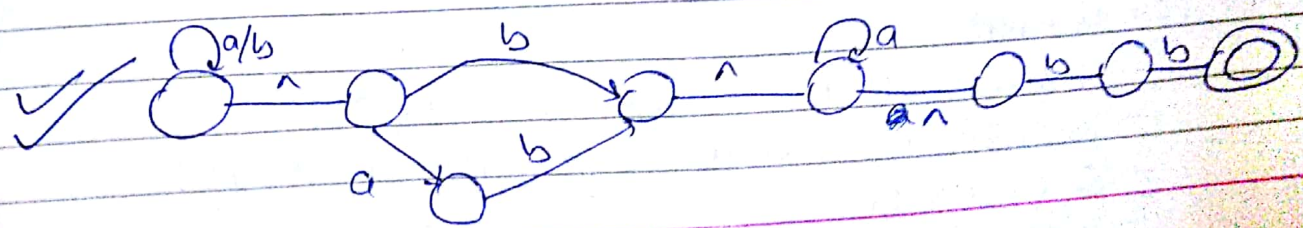
Q. R.E $\Rightarrow$ $a^* (b + aba^*) ab$

R_1 R_2 R_3



Q. RE $\Rightarrow$ $(a+b)^* (b+ab) a^* bb$





DATE : / /

Finite Automata to Regular Expression

→ Identities

1. $\phi + R = R$
2. $\phi . R = \phi$
3. $\Lambda . R = R$
4. $\Lambda^* = \Lambda$
5. $R^* R = R R^* = R^+$
6. $R^* R^* = R^*$
7. $R + R = R$
8. $\Lambda + R R^* = R^*$
9. $(PQ)^* P = PQPQ \ldots PQP = P(QP)^*$
10. $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
11. $(P+Q)R = PR + QR$

* Arden Theorem

$R = Q + RP$ and $P \neq \Lambda$

then $R = QP^*$

Let $R = QP^*$

$R = Q + QP^* P$

$= Q(\Lambda + PP^*) = QP^*$ $\{8\}$

Let $R = Q + RP$

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

$= Q + QP + (Q + RP)P^2 = Q + QP + QP^2 + RP^3$

$= \underbrace{(Q + QP + QP^2 + \ldots + QP^i)}_{A} + \underbrace{RP^{i+1}}_{B}$

PAGE NO.
DATE : / /

Let $w \in R$ & $|w| = i$

$\therefore w \in (Q + QP + \cdots + QP^i)$

or $w \in RP^{i+1}$

$P \neq \Lambda$ so $|RP^{i+1}| \geq i+1$

therefore $w \notin RP^{i+1}$

$w \in Q + QP + QP^2 + \cdots QP^i$

So

$R = Q + QP + QP^2 + \cdots + QP^i$

$R = Q(\Lambda + P + P^2 + \cdots + P^i)$

~~$R \in Q$~~ $\boxed{R = QP^*}$

* **Method :** FA to RE

Write equation for each state $q_i \in Q$, describing its reachability from other states

$$q_1 = \alpha_{11} q_1 + \alpha_{12} q_2 + \alpha_{13} q_3 + \cdots + \alpha_{1n} q_n$$
$$q_2 = \alpha_{21} q_1 + \alpha_{22} q_2 + \cdots + \alpha_{2n} q_n$$
$$\vdots$$
$$q_n = \alpha_{n1} q_1 + \alpha_{n2} q_2 + \cdots + \alpha_{nn} q_n$$

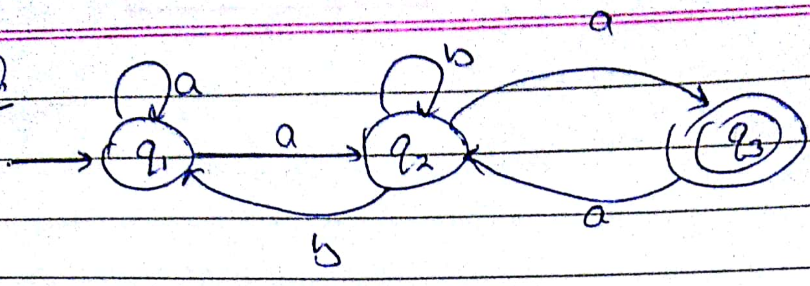
Apply identities & arden theorem & generate

$$q_f = q_0 \alpha_{ij}$$

Remove q_0. This is the final answer

Scanned by CamScanner

Q.



$$q_1 = q_1 a + q_2 b \quad \text{—} \; (1)$$
$$q_2 = q_1 a + q_2 b + q_3 a \quad \text{—} \; (2)$$
$$q_3 = q_2 a \quad \text{—} \; (3)$$

from (2) and (3)

$$q_2 = q_1 a + q_2 b + q_2 aa$$
$$q_2 = q_1 a + q_2 (b + aa)$$
$$R = Q + RP$$
$$R = QP^* \quad \Rightarrow \quad q_2 = q_1 a (b + aa)^* \quad \text{—} \; (4)$$

from (1) and (4)

$$q_1 = q_1 a + q_1 a (b+aa)^* b$$
$$q_1 = q_1 (a + a(b+aa)^* b) + \Lambda \qquad \{R + \Lambda = R\}$$
$$R = RP \qquad + Q$$
$$R = QP^* \quad \Rightarrow \quad q_1 = \Lambda (a + a(b+aa)^* b)^*$$
$$q_1 = (a + a(b+aa)^* b)^* \quad \text{—} \; (5) \qquad \{R \cdot \Lambda = R\}$$

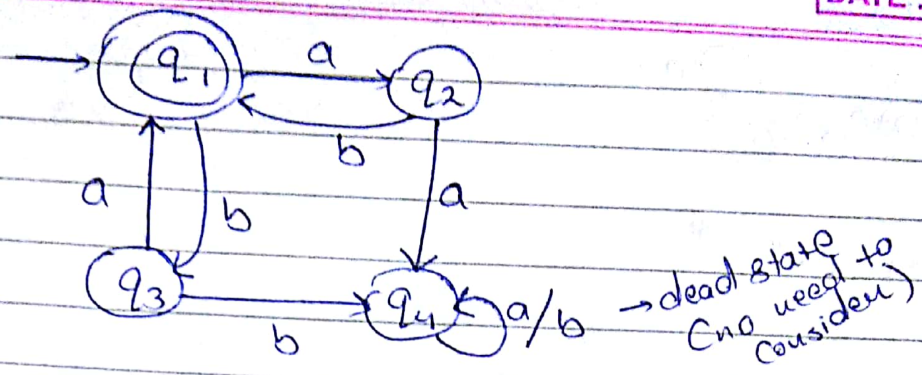
from (3) and (4)

$$q_3 = q_1 a (b+aa)^* a$$
$$q_3 = (a + a(b+aa)^* b)^* a (b+aa)^* a$$

Ans.

PAGE NO.
DATE : / /

Q.



$q_1 = q_2 b + q_3 a$ — (1)

$q_2 = q_1 a$ — (2)

$q_3 = q_1 b$ — (3)

$q_4 = q_2 a + q_3 b + q_4 (a+b)$ — (4)

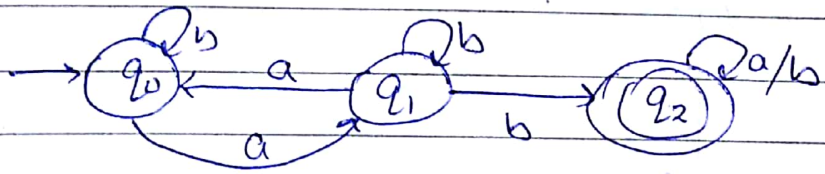
$q_1 = q_1 ab + q_1 ba$

$q_1 = q_1 (ab+ba) + \Lambda$

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

$q_1 = \cancel{ab^*} (ab+ba)^*$

Q.



$q_0 = q_0 b + q_1 a$ — (1)

$q_1 = q_0 a + q_1 b$ — (2)

$q_2 = q_1 b + q_2 (a+b)$ — (3)

$R = QP + RP$

$q_2 = q_1 b (a+b)^*$

$q_1 = q_0 a + q_1 b$ $\qquad q_0 = q_1 a b^*$

$q_1 = q_0 a b^*$ $\qquad q_0 = q_0 ab^* ab^* + \Lambda$

$\Rightarrow q_2 = q_0 ab^* b(a+b)^*$ $\qquad q_0 = (ab^* ab^*)^*$

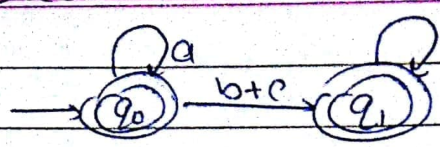
$q_2 = (ab^* ab^*)^* ab^* b(a+b)^*$

Scanned by CamScanner

PAGE NO.
DATE : / /

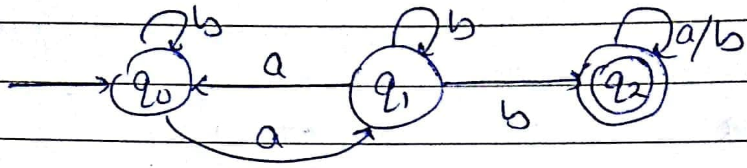
Direct Method

Q.



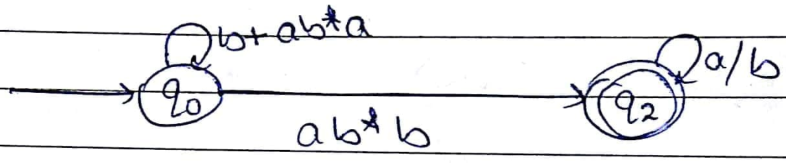
$(a^* + (b+c)a^*)^*$

Q



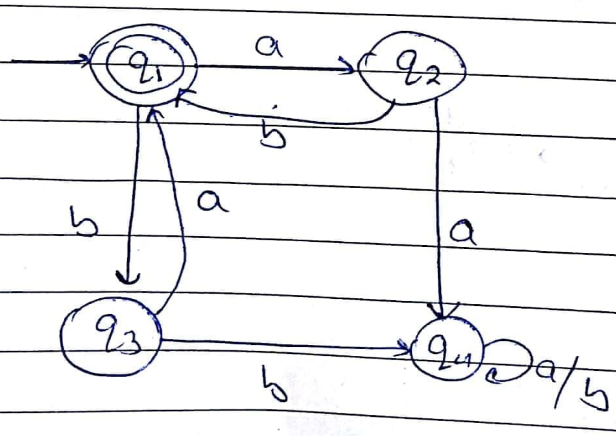
remove non-final states

=>

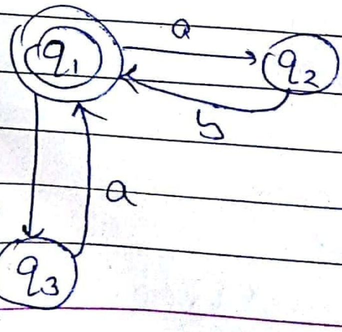


RE : $(b+ab^*a)^* ab^*b(a+b)^*$

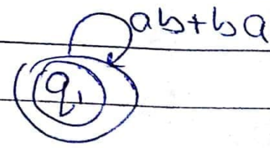
Q.



remove dead state

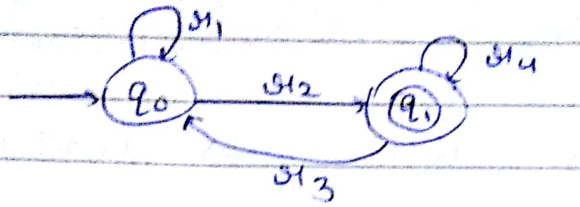


=>



RE : $(ab+ba)^*$

Q.



$r_1^* \; r_2(r_4^* + r_3 r_1^* r_2)^*$

Closure Properties of Regular Language

L_1 & L_2 are R.L.

1. union $L = L_1 \cup L_2$
2. Concatenation $L = L_1 . L_2$
3. Kleene Closure $L = L_1^*$
4. Complement

$L = \overline{L_1}$

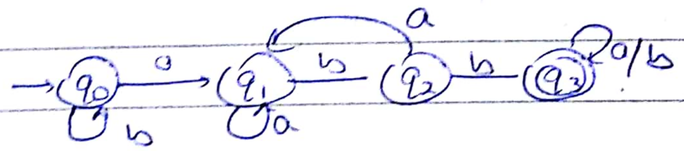
$L_1 = (a+b)^* \, abb \, (a+b)^*$

Procedure

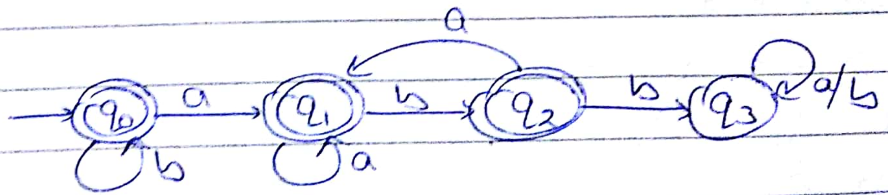
→ make all non-F.S. as F.S

→ make all F.S. as non-F.S.

L_1



L



$\Lambda + a(a^* + ba) + aa^*b$

PAGE NO.
DATE : / /

⑤ Reversal / Transpose

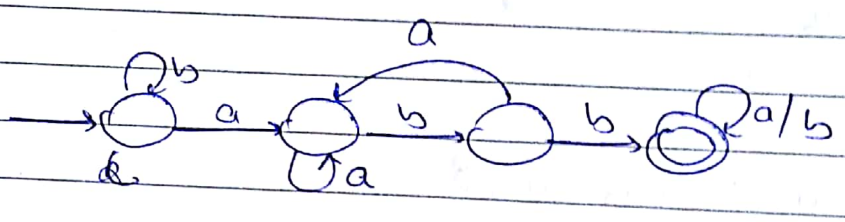
$L^R = L^T$

if abb substring ⇒ bba as sub.

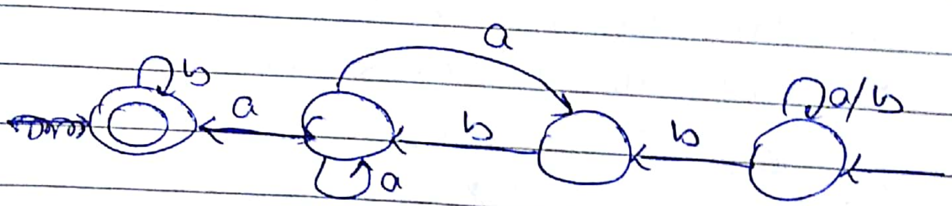
Procedure

→ make initial state as final state
→ reverse all the moves
→ make f.s. as I.s., if we are having more than one f.s. then introduce a new I.s. & null move from this state to all f.s.

L_1



L^T



not a DFA

⑥ Intersection

$L = L_1 \cap L_2 = \overline{\bar{L_1} \cup \bar{L_2}}$

$L_1 : M_1 (q_{01}, q_{f_1}, \Sigma, \delta_1, Q_1)$

$L_2 : M_2 (q_{02}, q_{f_2}, \Sigma, \delta_2, Q_2)$

$L : M (q_0, q_f, \Sigma, \delta, Q_1 \times Q_2)$

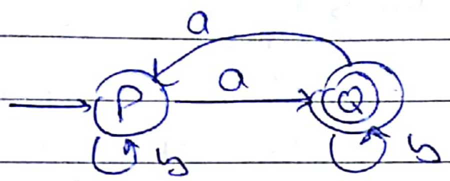
$q_0 = [q_{01}, q_{02}]$

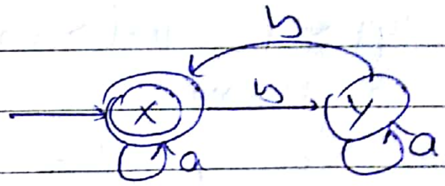
$q_f = [q_{f_1}, q_{f_2}]$

$$\delta([q_{i_1}, q_{i_2}], a) = [\delta_1(q_{i_1}, a), \delta_2(q_{i_2}, a)]$$

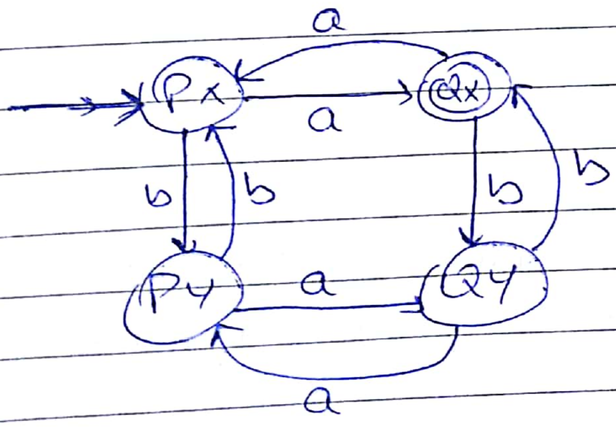
Q $\Sigma = \{a, b\}$

F.A. with odd a and even b





	a	b
→ Px	Qx	Py
Qx	Px	Qy
Py	Qy	Px
Qy	Py	Qx

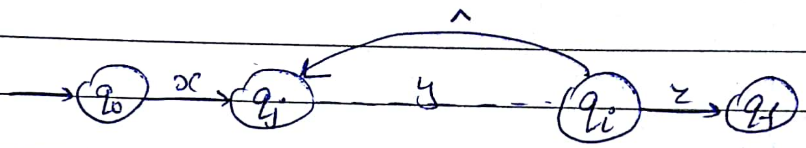


⑦ Set Difference

$L = L_1 - L_2 = L_1 \cap \overline{L_2}$

Pumping Lemma

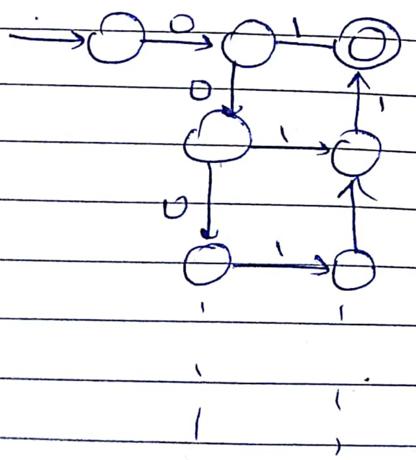
→ if L is regular then each string $\omega \in L$ can be written as xyz i.e. $\omega = xyz$
where $|y| \geq 1$, $|\omega| \geq m$ → no. of states {length of string should be more than no. of states for states to be repeated} (pigeon-hole principle)
and $|xy| \leq m$
then $\forall i \geq 0$ $\boxed{xy^i z \in L}$



there must y which repeats because string is infinite with finite states

Q. $L = 0^x 1^x$ $x > 0$

$x = 1$ $\omega = 01$
$x = 2$ $\omega = 0011$



no. of states are infinite
=> not a regular language

if $0 < x < 5$ => finite => regular

PAGE NO.
DATE : / /

Q. $L = 0^n 1^n \quad n > 0 \qquad |Q| = k \qquad k \geq m$

$\omega = xyz \quad = 0^m 1^m \qquad |\omega| = 2m > k$

Let $y = 0 \qquad |y| \geq 1$

$\omega = \underset{x}{0^{m-1}} \underset{y}{0} \underset{z}{1^m} \qquad |xy| \leq k$

$i = 0 \quad \omega = 0^{m-1} 1^m \notin L$

$i = 1 \quad \omega = 0^{m-1} 0 1^m = 0^m 1^m \in L$

$i = 2 \quad \omega = 0^{m-1} 0^2 1^m = 0^{m+1} 1^m \notin L$

$\therefore$ not regular

$y = 1 \qquad \omega = \underset{x}{0^m} \underset{y}{1} \underset{z}{1^{m-1}}$

$i = 2 \qquad 0^m 1^2 1^{m-1} \notin L$

$y = 01$

$\omega = \underset{x}{\underline{0^{m-1}}} \; \underset{y}{\underline{01}} \; \underset{z}{\underline{1^{m-1}}}$

$i = 2 \qquad 0^{m-1} (01)^2 y^{m-1}$

$0^{m-1} 0101 y^{m-1} = 0^m 1 0 1^m \notin L$

Q. $L = 0^{2n} \qquad n \geq 0$

Let $y = 00 \qquad \omega = \underset{x}{\underline{\wedge}} \; \underset{y}{\underline{00}} \; \underset{z}{\underline{0^{2(n-1)}}}$

$i = 0 \qquad \omega = 0^{2(n-1)} \in L$

$i = 1 \qquad \omega = 00\, 0^{2(n-1)} \in L$

$i = 2 \qquad \omega = (00)^2\, 0^{2(n-1)} \in L$

$\therefore \forall i \; xy^iz \in L$

$\Rightarrow$ regular

5 4

10

PAGE NO.
DATE : / /

Regular Grammar

(4 tuple)

$G(V_N, \Sigma, P, S)$

$V_N \rightarrow$ Non terminal

$\Sigma \rightarrow$ terminal

$P \rightarrow$ Production

$S \rightarrow$ Start symbol

$S \rightarrow NVN$

$N \rightarrow ram$

$N \rightarrow apple$

$N \rightarrow Book$

$V \rightarrow eat$

$V \rightarrow ate$

or

$N \rightarrow ram \mid apple \mid Book$

ram eat apple

$S \rightarrow NVN$

ram V N

ram eat apple

→ A grammar G is regular if all production are of form

$A \rightarrow aB$
$A \rightarrow a$
} Right linear

or

$A \rightarrow Ba$
$A \rightarrow a$
} left linear

Q. $S \rightarrow abA$

$A \rightarrow aA \mid a$

Right Linear

$S \rightarrow abA$

$\rightarrow aba$

$S \rightarrow abA$

$\rightarrow abaA$

$\rightarrow abaaA$

$\Rightarrow L = abaa^{+}$

aba

Sa
Aaa
abaa

Sa
Saa
Aaaa
ab

PAGE NO.
DATE : / /

Q. $S \rightarrow Aa \mid Sa$ Left Linear
$A \rightarrow ab$

$S \rightarrow Aa$
$\rightarrow aba$

$S \rightarrow Sa$
$\rightarrow Aaa$
$\rightarrow abaa$

$S \rightarrow Sa$
$\rightarrow Saa$
$\rightarrow Aaaa$
$\rightarrow abaaa$

$L = abaa^*$

Q. $S \rightarrow aA \mid a$
$A \rightarrow baB$
$B \rightarrow bB \mid b$

$S \rightarrow aA$
$\rightarrow abaB$
$\rightarrow abab$

ababB
ababb

$S \rightarrow a$

$L = (ababb^* + a)$

Q. $S \rightarrow aA$
$A \rightarrow bB \mid b$
$B \rightarrow aB$

$S \rightarrow ab$

$S \rightarrow abB$
$S \rightarrow abaB$
$\rightarrow abaaB$

$L = aba^*$

PAGE NO.
DATE : / /

Q. $S \to aA \mid a$
$A \to Ab \mid b \mid Aa$

$S \to a$

$S \to aA$
$\to ab$

$S \to aA$
$\to aAb$
$\to abb$
$\to aAbb$

$S \to aA$
$\to aAa$
$\to aba$
$\to aAba$
$\to aAaba$

aAa
~~abb aAabb~~
~~aAabb~~
ababb

ababa

~~$ab^* + ababb^*$~~

$S \to aA$
$S \to aAb$ $\quad aAba$
$aAbb$ $\quad aAaa$

any combination of a & b

$\Rightarrow S \to aA(a+b)^*$

to terminate $A \to b$

$\Rightarrow S \to ab(a+b)^*$

$\Rightarrow L = a + ab(a+b)^*$

Q. $S \to aA$
$A \to Bb \mid b$
$B \to aA$

$S \to ab$

$S \to aBb$
$\to aaAb$
$aabb$

$S \to aaAb$
$aaBbb$
$aaaAbb$

~~$a^* b^*$~~

$L = a^n b^n \quad n \geq 1$ (not regular)

Scanned by CamScanner

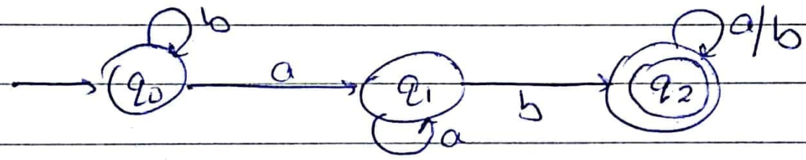
→ if some production are left linear & right linear then grammar may or may not be regular but if either left or right ⇒ always regular

Finite Automata to regular grammar

1. $\delta(q_i, a) \to q_J$ & $q_J \notin F$ $\quad A_i \to aA_J$

2. $\delta(q_i, a) \to q_J$ & $q_J \in F$ $\Rightarrow$ $A_i \to aA_J \mid a$

Q.



$A_0 \to bA_0$

$A_0 \to aA_1$

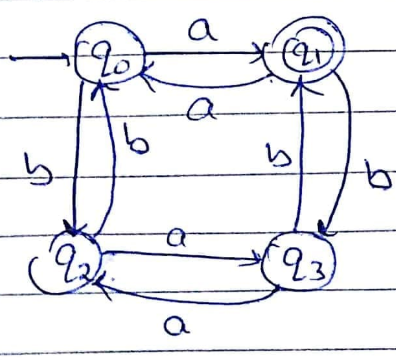
$A_1 \to aA_1$

$A_1 \to bA_2 \mid b$

$A_2 \to aA_2 \mid a$

$A_2 \to bA_2 \mid b$

Q.



$A_0 \to aA_1 \mid a$

$A_0 \to bA_2$

$A_1 \to aA_0$

$A_1 \to bA_3$

$A_2 \to aA_3$

$A_2 \to bA_0$

$A_3 \to aA_2$

$A_3 \to bA_1 \mid b$

Regular Grammar to FA

① $A_1 \to aA_2$ then $\delta(q_1, a) \to q_2$

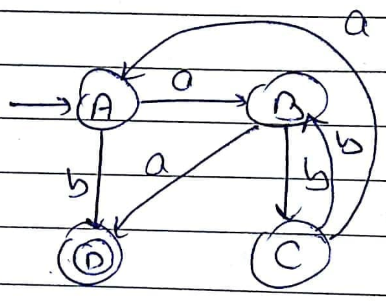
② $A_1 \to a$ then $\delta(q_1, a) \to q_f$

where $q_f \in F$

Q $A \to aB \mid b$

$B \to bC \mid a$

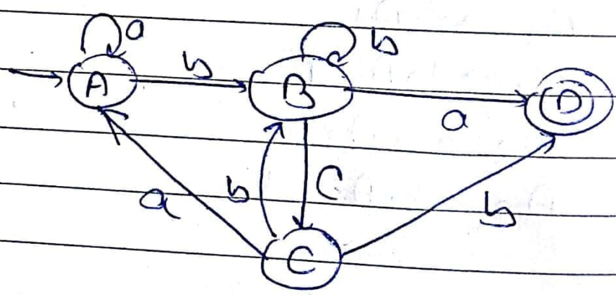
$C \to aA \mid bB$

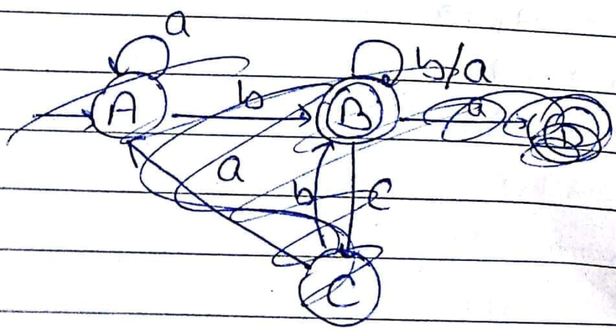


Q. $A \to aA \mid bB$

$B \to bB \mid cC \mid a$

$C \to aA \mid bB \mid b$





Context free grammar

→ A grammar $G(V_N, \Sigma, P, S)$ is CFG if all the production are of form

$A \rightarrow \alpha$ where $\alpha \in (V_N \cup \Sigma)^*$

↳ set of terminal & non-terminal of any combination

$S \rightarrow AB$

$A \rightarrow aaA$

$A \rightarrow \wedge$

$B \rightarrow Bb$

$B \rightarrow \wedge$

String <u>aaab</u>

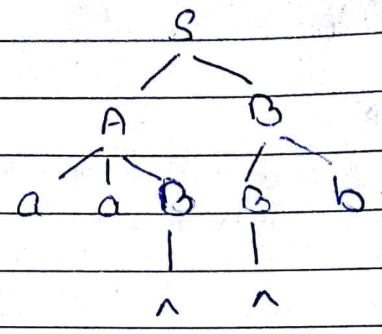
Derivation — Left most / Right most

(checking whether string belongs to the grammar or not)

(aaab)	Left most (aab)	Right most (aab)
$S \rightarrow AB$	$S \rightarrow \underline{A}B$	$S \rightarrow A\underline{B}$
$S \rightarrow aaAB$	$S \rightarrow aaAB$ $(A \rightarrow aaA)$	$S \rightarrow A\underline{B}b$
$S \rightarrow aaaaAB$	$S \rightarrow aaB$ $(A \rightarrow \wedge)$	$S \rightarrow \underline{A}b$
× does not belong to CFG	$S \rightarrow aaBb$ $(B \rightarrow Bb)$	$S \rightarrow aa\underline{A}b$
	$S \rightarrow aab$ $(B \rightarrow \wedge)$ ✓	$S \rightarrow aab$ ✓

PAGE NO.
DATE : / /

Derivation Tree



Q. $S \rightarrow abSCB$

$S \rightarrow \wedge$

$B \rightarrow bB \mid b$

abSCB

String ababcbcbb

$S \rightarrow abScB$

$S \rightarrow ababScBcB$

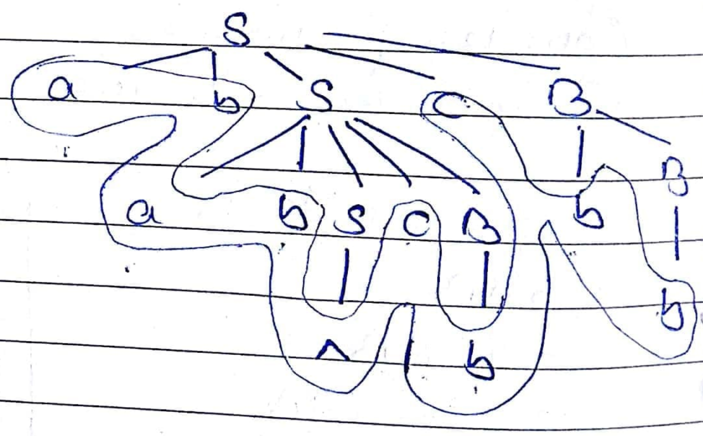
$S \rightarrow ababcBcB$

~~S → ababcbB~~

$S \rightarrow ababcbcB$

$S \rightarrow ababcbcbB$

$S \rightarrow ababcbcbb$ ✓

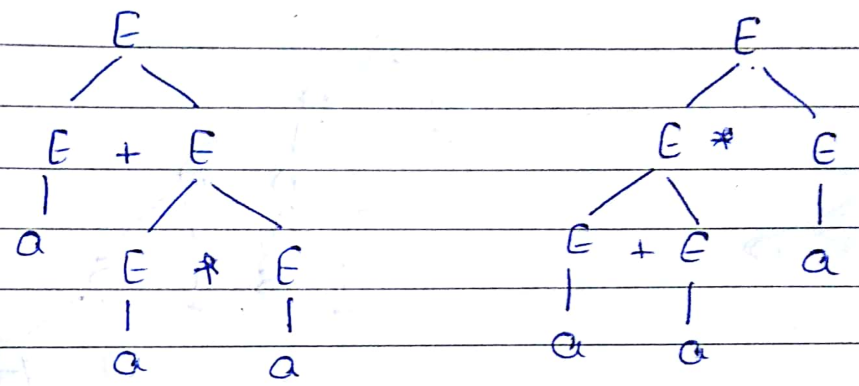


Yield of tree
→ when string is accepted.

Ambiguous Grammar

$E \to E+E \mid E*E \mid a$

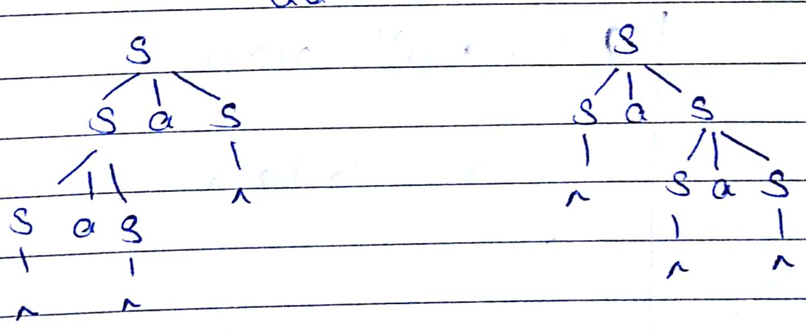
String $a+a*a$



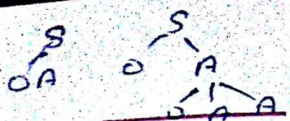
→ if for one string there is more than one derivation tree than grammar is ambiguous.

eg: $S \to SaS \mid \wedge$

aa



if at same level same non terminal on both the sides with null move than ambiguous

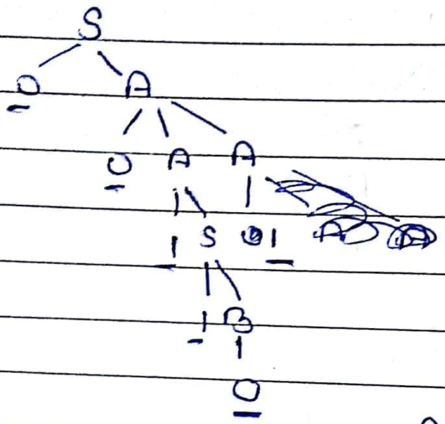


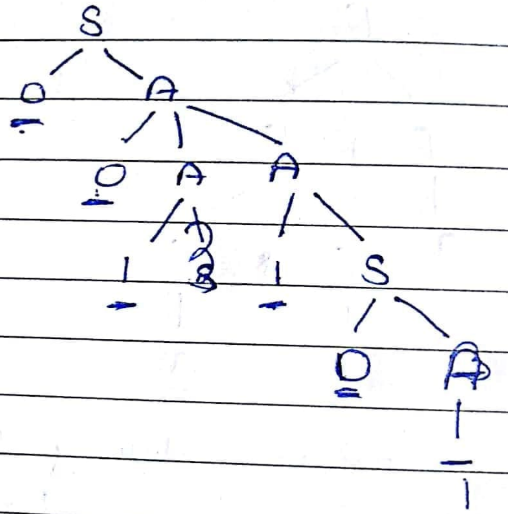
Q. $S \to 0A \mid 1B$

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

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

001101



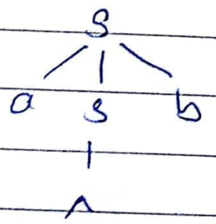


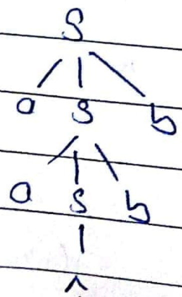
Ambiguous

Generating Grammar

Q. $L = a^n b^n \quad n \geq 0$

$S \to aSb \mid \wedge$





Q $L = a^n b^n \quad n > 0$

$S \to aSb \mid ab$

Q. $L = ww^R$ $\quad w \in \{a, b\}^*$

$S \rightarrow aSa \mid bSb \mid \wedge$ — even length of palindrom

abba
bbaabb

$S \rightarrow aSa$
$S \rightarrow abSba$
$S \rightarrow abba$

a starting → a end] aSa

Q. $L = wcw^R$
$S \rightarrow aSa \mid bSb \mid c$

Q. L = all string having equal no. of a & b

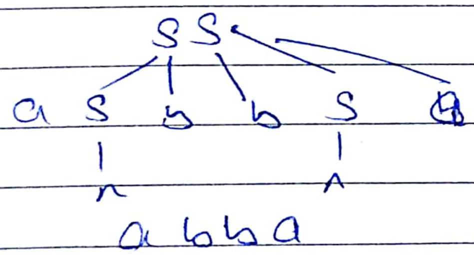
$S \rightarrow aSb \mid bSa \mid \wedge$

Cannot be generated

Q. String → aabaabbb

~~$S \rightarrow aSb \mid bSa \mid \wedge$~~
$S \rightarrow aSb \mid bSa \mid \wedge \mid SS$

a S b
a a S b b
a a b S b b b
a a b a S b b b b



Q. L = all string having unequal no. of a & b

Q. $L = a^n b^m \quad n > m$

Scanned by CamScanner

$a^* \to \quad S \to aS | \wedge$

PAGE NO.
DATE : / /

Q. $L = a^n b^m \quad n \geq m$

$S \to aSb | A$
$A \to aA | a$

Q $L_1 = a^n b^{2m}$

$a^* (bb)^*$

$S \to AB$
$A \to aA | \wedge$
$B \to bbB | \wedge$

$L_2 = a^n b^{2n} \quad n \geq 0$

$S \to aSbb | \wedge$

Q. $L = \underline{a^n b^n} \underline{c^m d^m} \quad n, m > 0$

$S \to AB$
$A \to aAb | ab$
$B \to cBd | cd$

Q. $L = \underline{a^m b^{2m}} \underline{c^n} \quad m, n \geq 0$

$S \to AC$
$A \to aAbb | \wedge$
$C \to cC | \wedge$

Q. $L = a^n b^{m+n} c^m \quad m, n \geq 0$ ~~$a^n b^n b^m c^m$~~
$= \underline{a^n b^n} \underline{b^m c^m}$

~~$aBbc$~~

$S \to AB$
$A \to aAb | \wedge$
$B \to bBc | \wedge$

Scanned by CamScanner

PAGE NO.
DATE : / /

Q. $L = a^n b^{2n} c^n$

Q. $L = a^n b^m c^m d^n$

$S \to aSd \mid A$
$A \to bAc \mid \wedge$

Q $L = a^n b^m c^n d^m$

CFG ✗

intersecting

Q. $L = a^n b^m c^k \quad n = m$ or $m \le k$

$L_1 \quad \underline{n = m} \quad a^n b^n c^*$

$L_2 \quad \underline{m \le k} \quad a^* b^m c^{m+x}$

$L = L_1 \cup L_2$
$S \to S_1 \mid S_2$
$S_1 \to AC$
$A \to aAb \mid \wedge$
$C \to cC \mid \wedge$

$S_2 \to BD$
$B \to aB \mid \wedge$
$D \to bDc \mid c$

Q. $L = w(a,b)$ where $n_a^{(w)} \ne n_b^{(w)}$

$S \to aSb \mid bSa \mid \cancel{SS} \mid A \mid B$
$A \to aA \mid a$
$B \to aB \mid b$

$S \to aSb \mid bSa \mid abS \mid baS \mid Sab \mid Sba \mid A \mid B$
$A \to aA \mid a \quad B \to aB \mid b$

Scanned by CamScanner

Q. $L = a^n \underbrace{w w^R} b^n$

$S \rightarrow aSb \mid A$
$A \rightarrow aAa \mid bAb \mid \wedge$

Closure properties of CFL

Let L_1 and L_2 be CFL
L_1 : $G_1 (S_1, \Sigma_1, V_1, P_1)$
L_2 : $G_2 (S_2, \Sigma_2, V_2, P_2)$

① $L = L_1 \cup L_2$

$S \rightarrow S_1 \mid S_2$

$L_1 = a^n b^n c^m \quad m, n \geq 0$
$L_2 = a^n b^m c^m \quad m, n \geq 0$

$S_1 \rightarrow AC$	$S_2 \rightarrow PB$
$A \rightarrow aAb \mid \wedge$	$P \rightarrow aP \mid \wedge$
$C \rightarrow \wedge C \mid \wedge$	$B \rightarrow bBc \mid \wedge$

② concatenation

$L = L_1 . L_2$

$ab \; a^2b^2 \; a^3b^3c^4 \quad b^4c^4 \; bc \; ab^2c^2$

$S \rightarrow S_1 S_2$

③ Kleene closure

$L = L_1^*$

$abc \quad a^2b^2 \quad abc^4 \quad abc^9$

$S \rightarrow SS_1 \mid \wedge$

PAGE NO.
DATE : / /

(4) Reversal (Transpose)

$L = L_1^R$

$L_1 = a^n b^n c^m \quad n, m \geq 0$

$L = c^m b^n a^n \quad n, m \geq 0$

$\downarrow$

$G(V, \Sigma, S, P)$

$\underline{P = P^T}$

L_1

$S_1 \rightarrow AC$

$A \rightarrow aAb \mid \wedge$

$C \rightarrow cC \mid \wedge$

L

$S \rightarrow CA$

$A \rightarrow bAa \mid \wedge$

$C \rightarrow Cc \mid \wedge$

(5) Intersection

eg: $L = L_1 \cap L_2$

$L \Rightarrow a^n b^n c^n \quad n \geq 0$

intersecting $\Rightarrow$ not CFL

eg: $L_1 = a^n \underline{b^m c^m} d^k \quad n, m, k \geq 0$ $\qquad b = c$

$L_2 = \underline{a^n b^p c^q d^n} \quad n, p, q \geq 0$ $\qquad a = d$

$L = L_1 \cap L_2 \Rightarrow a^n b^m c^m d^n$ ✓ CFL

So for intersection we have to go language by tag language.

$\wedge \rightarrow$ is a termi...

PAGE NO.
DATE : / /

(6) Compliment

$$\overline{\bar{L_1} \cup \bar{L_2}} = L_1 \cap L_2$$

Let us assume CFL are closed under complement

$L_1 / L_2 \rightarrow CFL$

$\bar{L_1} / \bar{L_2} \rightarrow CFL$

$\bar{L_1} \cup \bar{L_2} \rightarrow CFL$

$\overline{\bar{L_1} \cup \bar{L_2}} \rightarrow CFL$

$\Rightarrow L_1 \cap L_2 \rightarrow$ must be CFL

but intersection is not CFL

$\Rightarrow$ Our assumption is ~~not~~ wrong $\Rightarrow$ compliment is not CFL

Simplification of CFL

→1) Elimination of useless symbol

↳ non generating non terminal
↳ non reachable

(1) Non Generating NT

W_i = Set of all NT that are generating

$W_i = \{A \mid A \rightarrow \alpha \text{ \& } \alpha \in \Sigma^*\}$

$W_{i+1} = W_i \cup \{A \mid A \rightarrow \alpha \in P \text{ and } \alpha \in (\Sigma \cup W_i)^*\}$

$W_i = W_{i+1}$

$V_N' = W_i$

P' = all production from W_i

Scanned by CamScanner

PAGE NO.
DATE : / /

eg: $C \to d$
$S \to AB | BC$
$A \to aA$
$C \to dC$
$B \to bC | SA$
$E \to b | c$

$\omega_1 = \{C, E\}$

$\omega_2 = \{C, E\} \cup \{B\} = \{E, C, B\}$

$\omega_3 = \{E, C, B\} \cup \{S\} = \{E, C, B, S\}$

now remove productions not having E, C, B, S of non terminal

$\Rightarrow$ $C \to d | dC$
$S \to BC$
$B \to bC$
$E \to b | C$

eg: $S \to AB | CaD$
$A \to Bc | Aa | b$
$B \to CB$
$C \to Ac | d$
$D \to SA$
$G \to a$
$H \to aG$

$\omega_1 = \{A, C, G\}$
$\omega_2 = \{A, C, G\} \cup \{H\}$
$\omega_3 = \underline{\{A, C, G, H\}} \cup \{\ \}$

DATE: / /

~~A→BC~~ $A \to Aa \mid b$

~~BCD~~ $C \to Ac \mid d$

$G \to a$

$H \to aG$

grammar is empty $\Rightarrow$ coz S is not present

(2) Elimination of Non-Reachable symbol

$w_i = \{S\}$

$w_{i+1} = w_i \cup \{A \mid B \to A \text{ and } B \in w_i\}$

$V_N' = V_N \cap w_i$, $\Sigma' = \Sigma \cap w_i$

P' = all production in P that include Σ, $V_N \in w_i$

$S \to aA \mid b$

$A \to ac \mid b \mid e$

$C \to Be$

$B \to cB \mid d$

$E \to a \mid b$

$w_1 = \{S\}$

$w_2 = \{S\} \cup \{A, a, b\} = \{S, A, a, b\}$

$w_3 = \{S, A, a, b, e, C\}$

$w_4 = \{S, A, a, b, e, C, B\}$

$w_5 = \{S, A, a, b, e, C, B, c, d\}$

$V_N' = V_N \cap w_i$

$= \{S, A, B, C, E\} \cap w_i$

$= \{S, A, B, C\}$

$\Sigma' = \Sigma \cap w_i$

$= \{a, b, c, d, e\} \cap w_i$

$= \{a, b, c, d, e\}$

PAGE NO.
DATE : / /

$P' \Rightarrow$ $S \to aA \mid b$
$A \to aCb \mid e$
$C \to Be$
$B \to cB \mid d$

Q. $S \to aS \mid bA$
$A \to eB \mid d$
$B \to cB \mid cC$
$C \to cC$
$E \to eF \mid d$
$F \to eE \mid g$

$\cancel{W_1 = \{S\}}$
$\cancel{W_2 = \{S, a, b,}$

$W_1 = \{A, E, F\}$
$W_2 = \{A, E, F\} \cup \{S\}$
$\cancel{W_3 =}$

$S \to aS \mid bA$
$A \to d$
$E \to eF \mid d$
$F \to eE \mid g$

$W_1 = \{S\}$
$W_2 = \{S\} \cup \{a, b, A\}$
$W_3 = \{S, a, b, A, d\}$
$\cancel{W_4 = d}$

P' $\boxed{\begin{array}{l} S \to aS \mid bA \\ A \to d \end{array}}$

Scanned by CamScanner