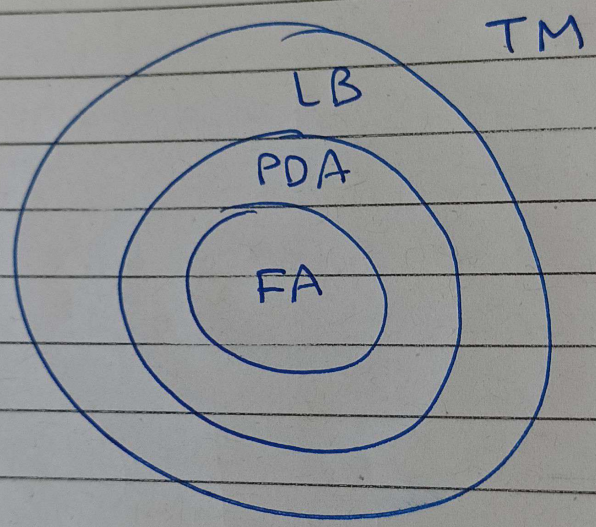
Page No.
Date :

① finite automata	(Regular lang)
② push down	context free
③ linear bounded	context sensitive
④ Turing machine	recursively enumeration (free grammar)



TOC

Date :

Symbols

(Σ) Alphabets $\subset$ Symbols

0 - 9

A - Z

a - z

Image

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

String = collection of alphabets.

S :- a, aa, ab, ba, bb, aaa...

{ Sentence :- collection of words
Language :- collection of sentence
" " strings }

eg of lang :-

L_1 = strings of length 2 → $\{aa, bb, ab, ba\}$

→ finite & ∞ language

L_2 = all strings ends with a → ∞
(a, aa, aaa, ...)

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

$|\Sigma|$ = power ≡ length of string.

$\epsilon = |\Sigma|^0$ → all strings of len = 0
$|\Sigma|^1$ → " " " " = 1

ϵ ≡ length of '•' string

$|\Sigma|^2 = \{aa, ab, ba, bb\}$

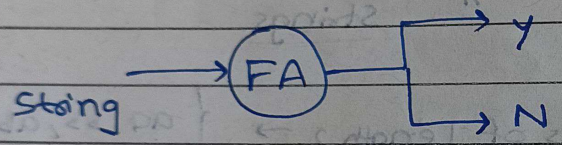
$|\Sigma|^* \rightarrow |\Sigma|^0 \cup |\Sigma|^1 \cup |\Sigma|^2 \ldots$

universal language

$= \{ \underbrace{\epsilon}_{0}, \underbrace{a}_{1}, b, \underbrace{ab, aa, bb, ba}_{2} \ldots \}$

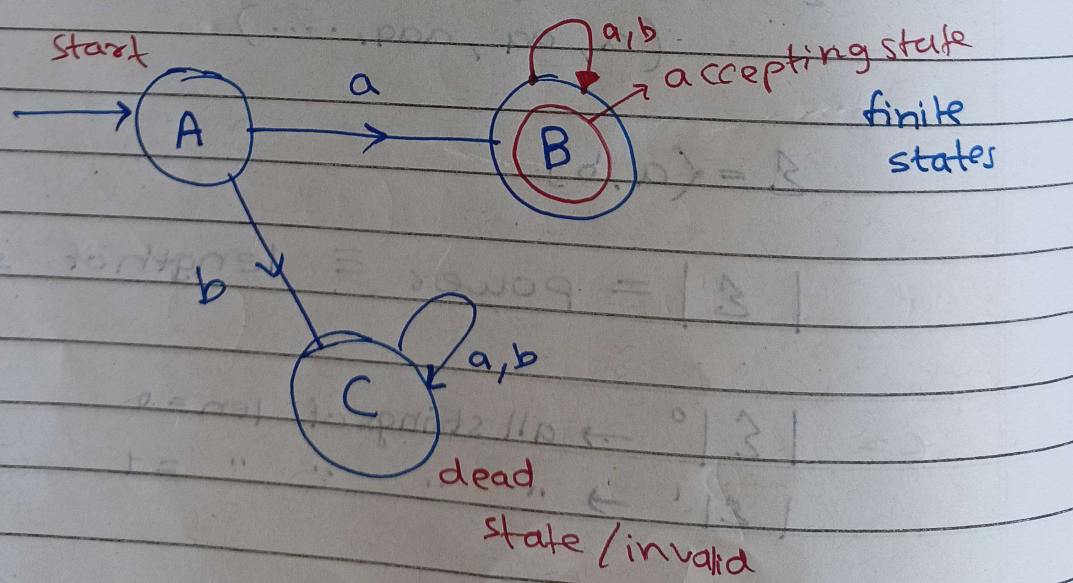
→ Compiler is prerequisite of TOC

Q How to achieve string validation in finite time?



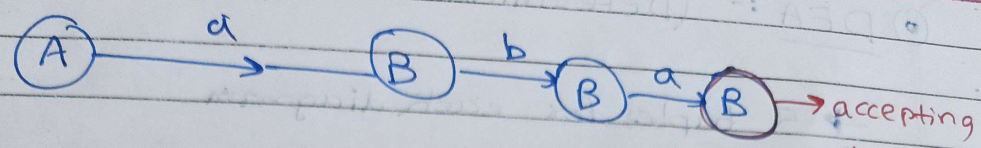
(1) $\Sigma = (a, b)$

L = all strings starts with 'a'



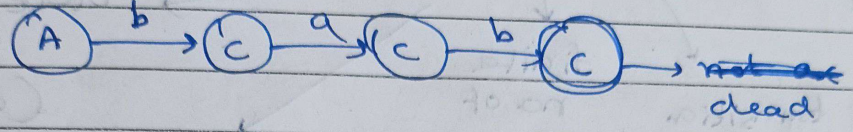
Page No.
Date :

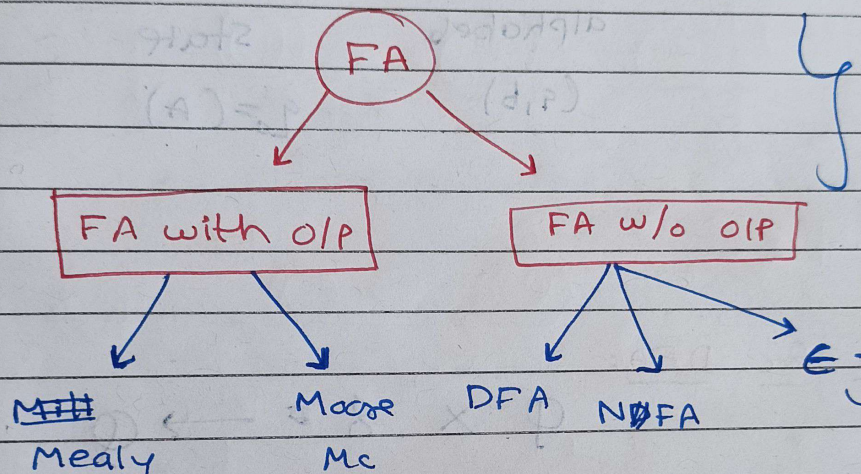
~~abb~~



valid : invalid

b ab





NFA $\xrightarrow{\text{converter}}$ DFA

(easy to understand)

DFA :- (Deterministic Finite Automata)

→ 5 tuples for each diagram

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

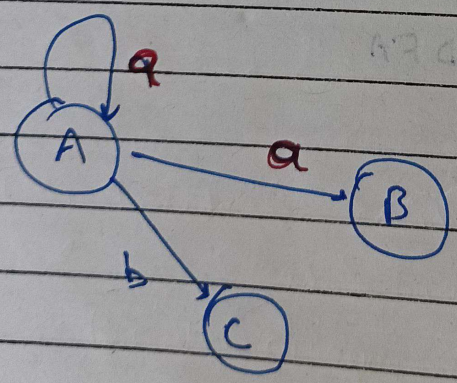
- Q → finite number of states (A, B, C)
- δ → transition function
- Σ → finite no. of I/P alphabets (a, b)
- q_0 → start state $q_0 = (A)$
- F → accepting state (atleast one exists) (eg: B)

δ for DFA :-

$$Q \times \Sigma \rightarrow Q$$

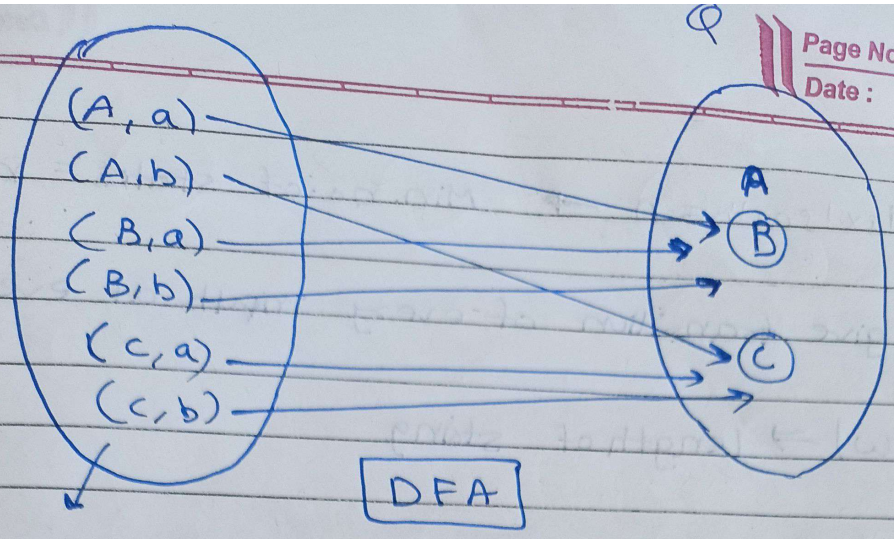
cross से कटा आएँगे

→ 1 state (using unique symb) to another state



(NFA) (ambiguous)

$$\{A, B, C\} \times \{a, b\} \rightarrow Q$$



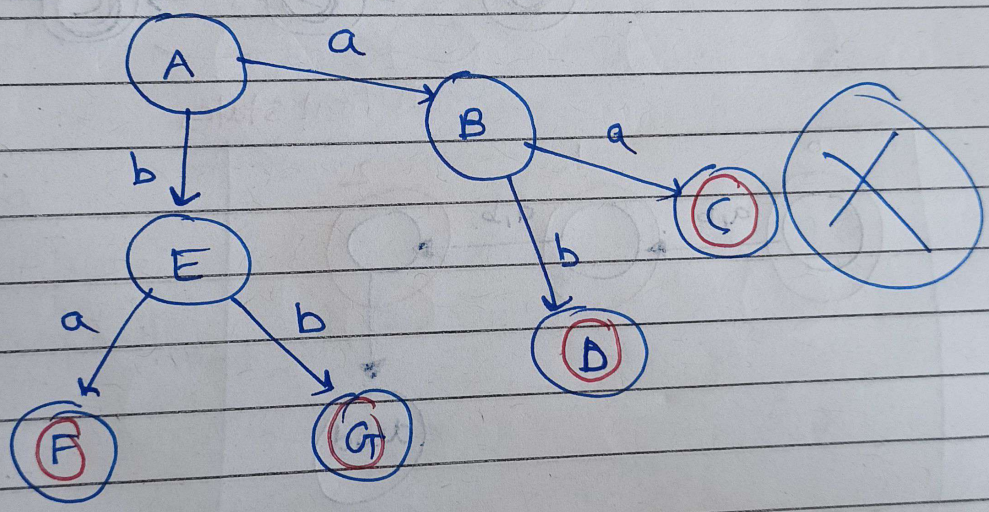
DFA

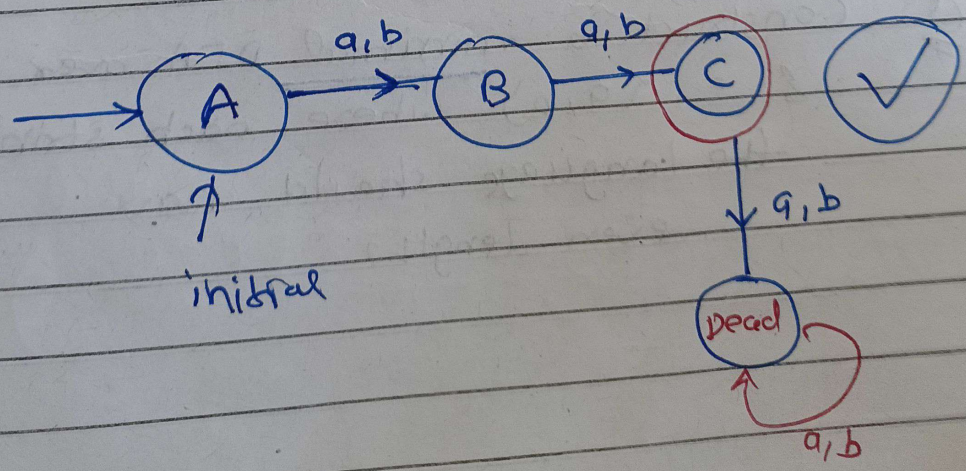
any one will not have more than one image.

Q

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

L = all strings of len 2





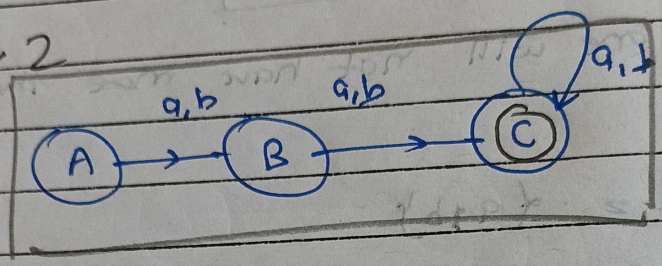
Date :

Max length = n → Min. no. of states = n+1

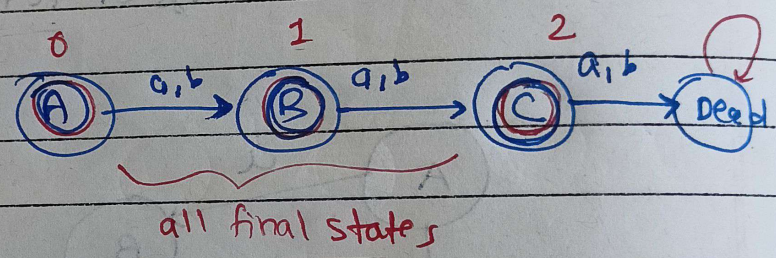
→ give transition of every inputs at every stage

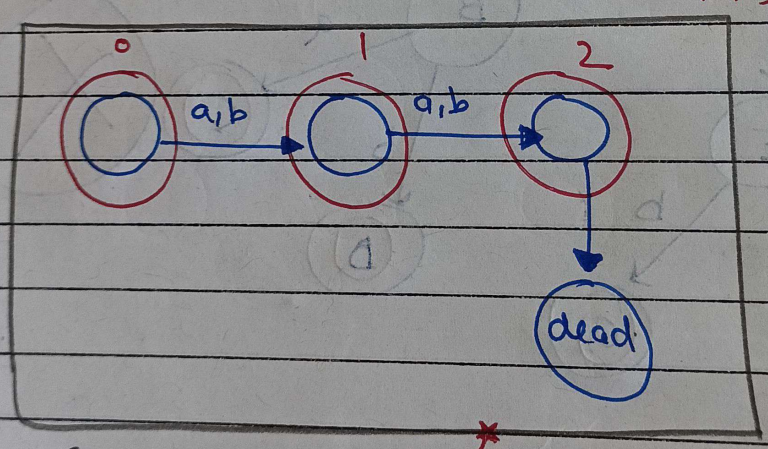
|w| ⇒ length of string.

Q. |w| ≥ 2

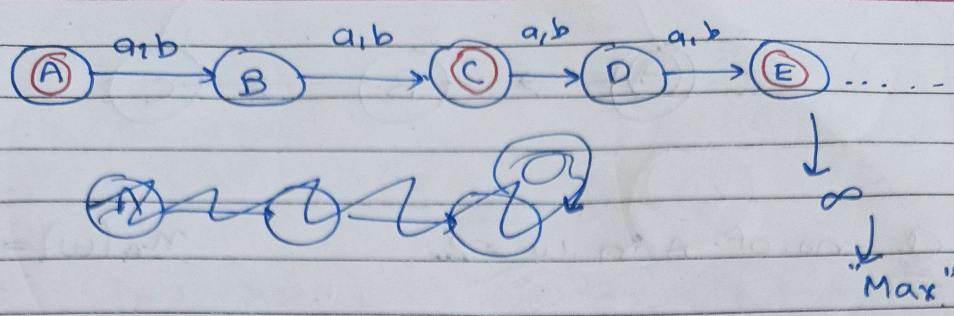


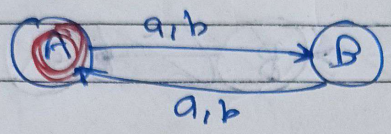
Q |w| ≤ 2



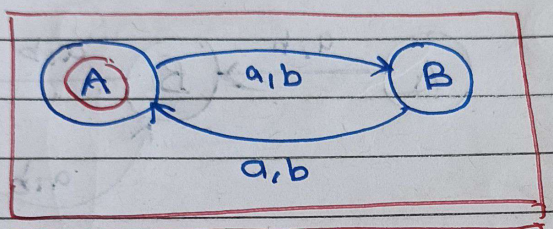


Q. Construct a minimal DFA over $\Sigma = \{a,b\}$ where each string of the language should have even length?

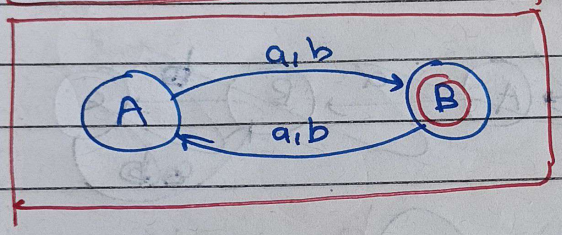




(Minimal)

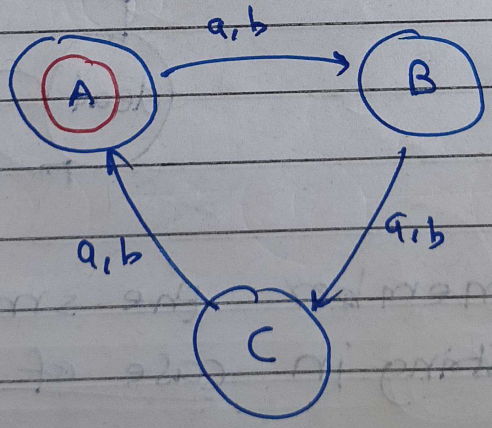


even



odd

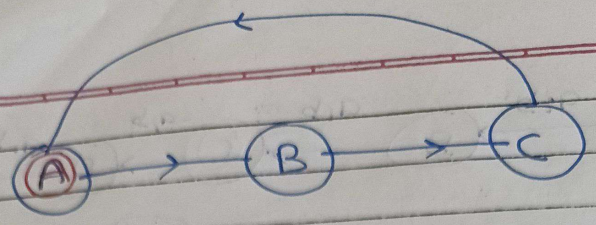
Q. $|w|_{mod\ 3} = 0$.



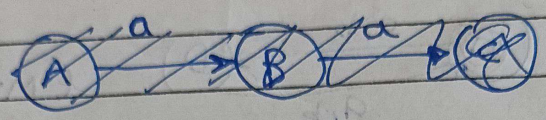
$|w| \bmod k = m$

k → no. of states

m → final state

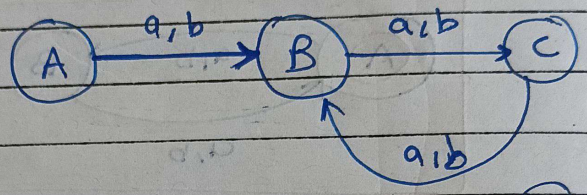


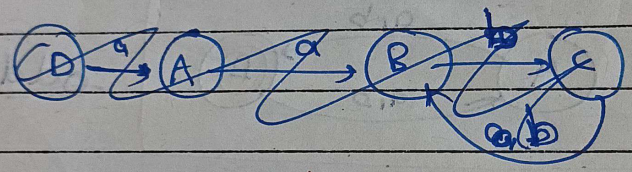
Q no. of A in $\omega = 2$. $n_A(\omega) = 2$

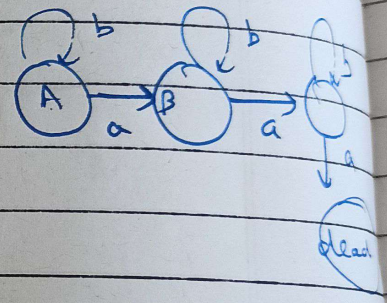


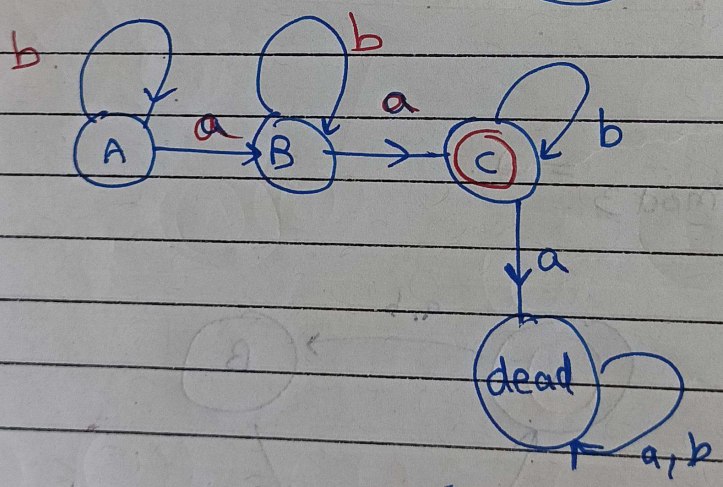
aa
aab
aba

aabb...







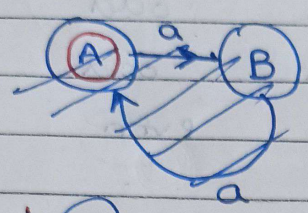


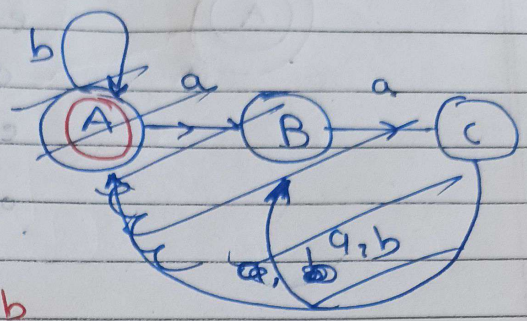
Remember the smallest string in case of Ω

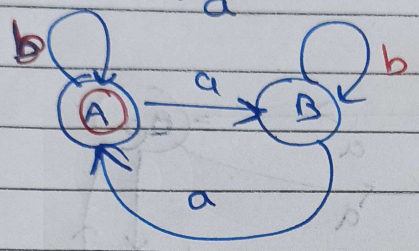
Page No.
Date :

Q. $(n_A(w) \% 2 == 0)$

{}, aa, aaaa

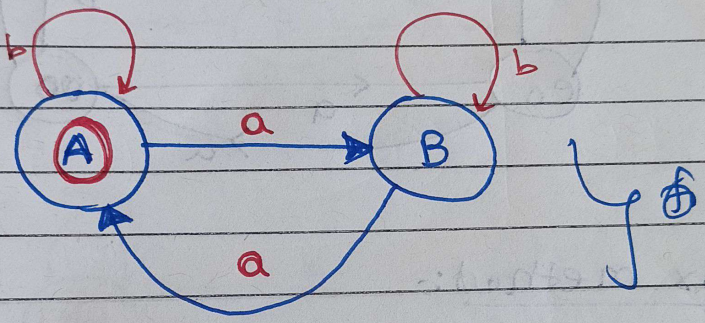




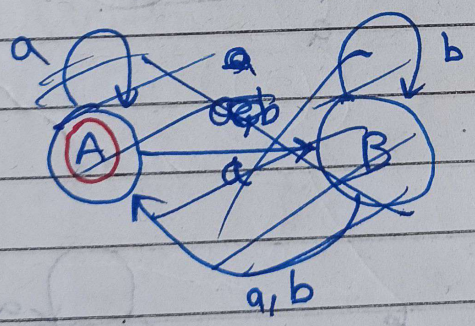


b, a baaabbbbbb a

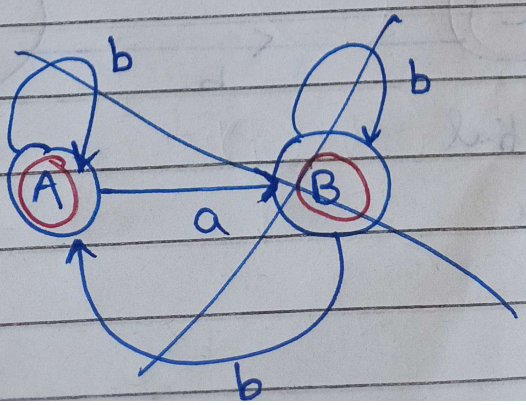
→



Q. No. of A in w %2==0 & No. of B in w %2==0.



abb



~~ba~~

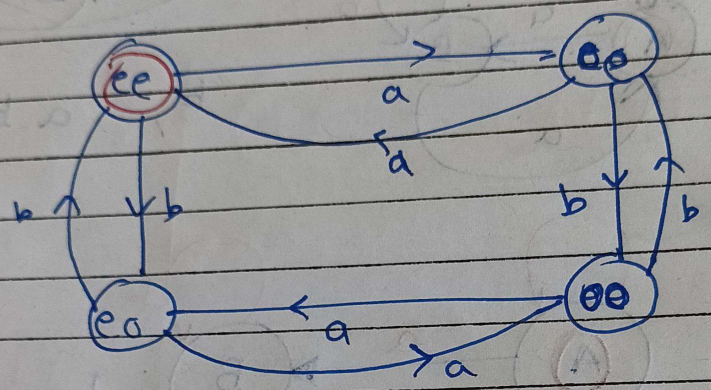
~~ab~~

b a

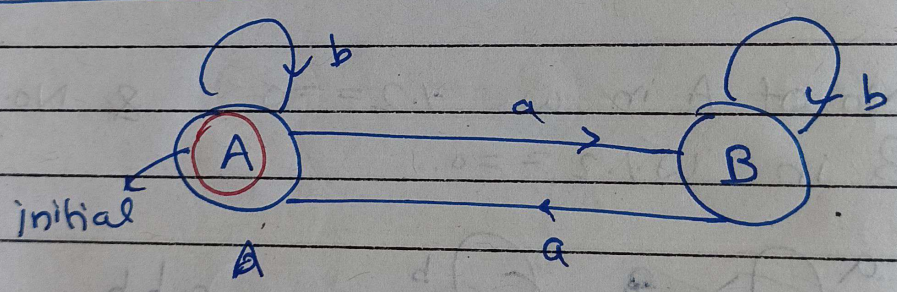
Date :

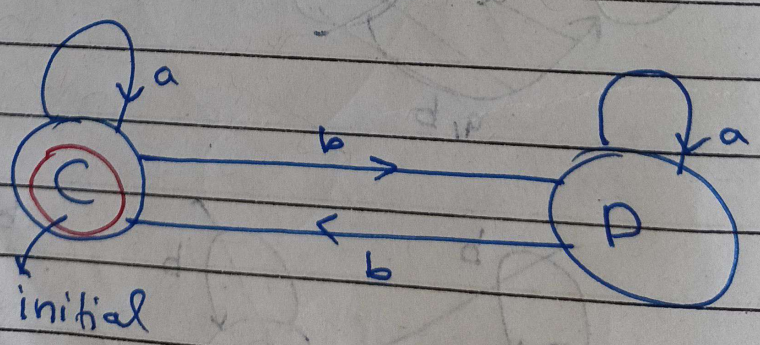
0, ~~aa~~, ab, ba, aa, bb,

A	even	even
	odd	odd
	even	odd
	odd	even

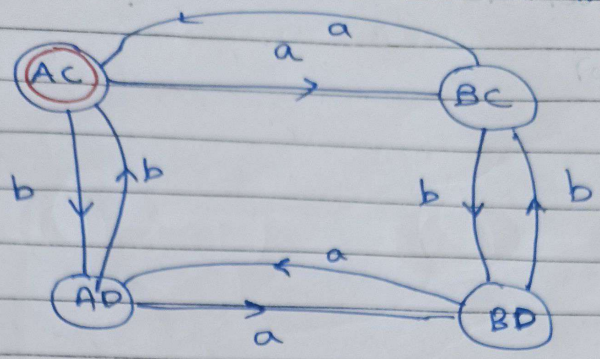


Matrix method:-

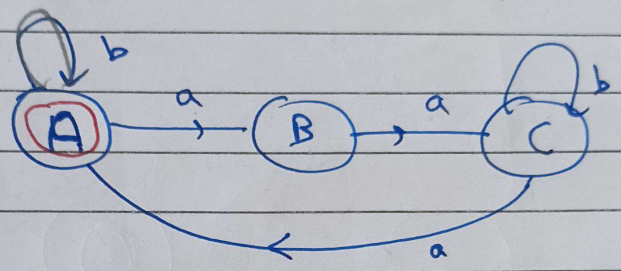




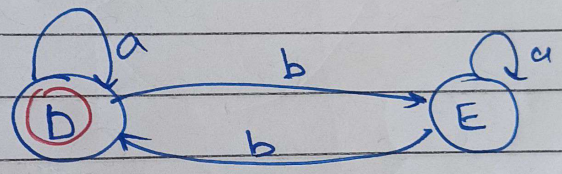
After combination, AC will be initial & final also would be AC.

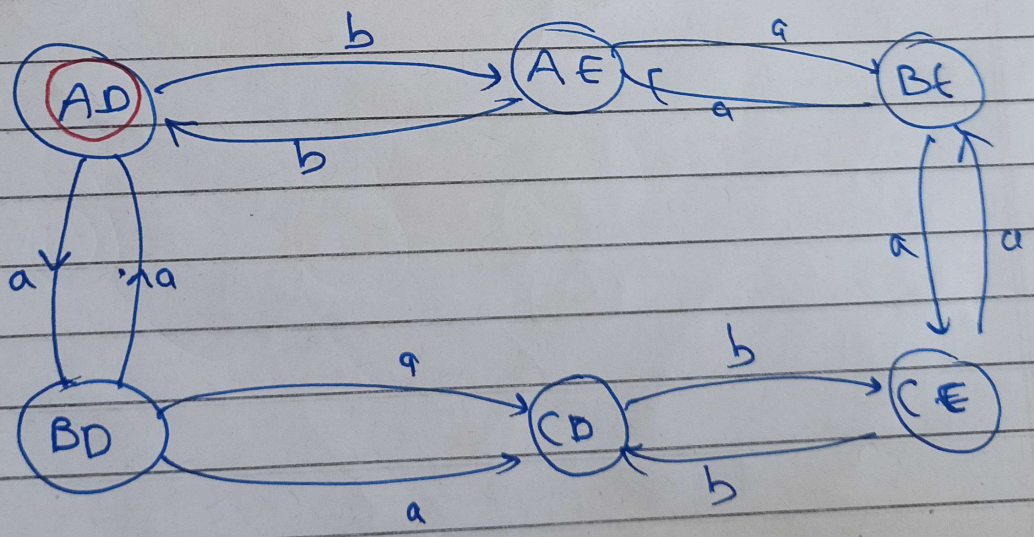


$n_a(w) \% \text{mod } 3 == 0$ && $n_b(w) \% \text{mod } 2 == 0$.



{A B C} x {D E}
= {AD, AE, BD, BE, CD, CE}





$n_a(w) \bmod 3 = 0$

$n_b(w)$

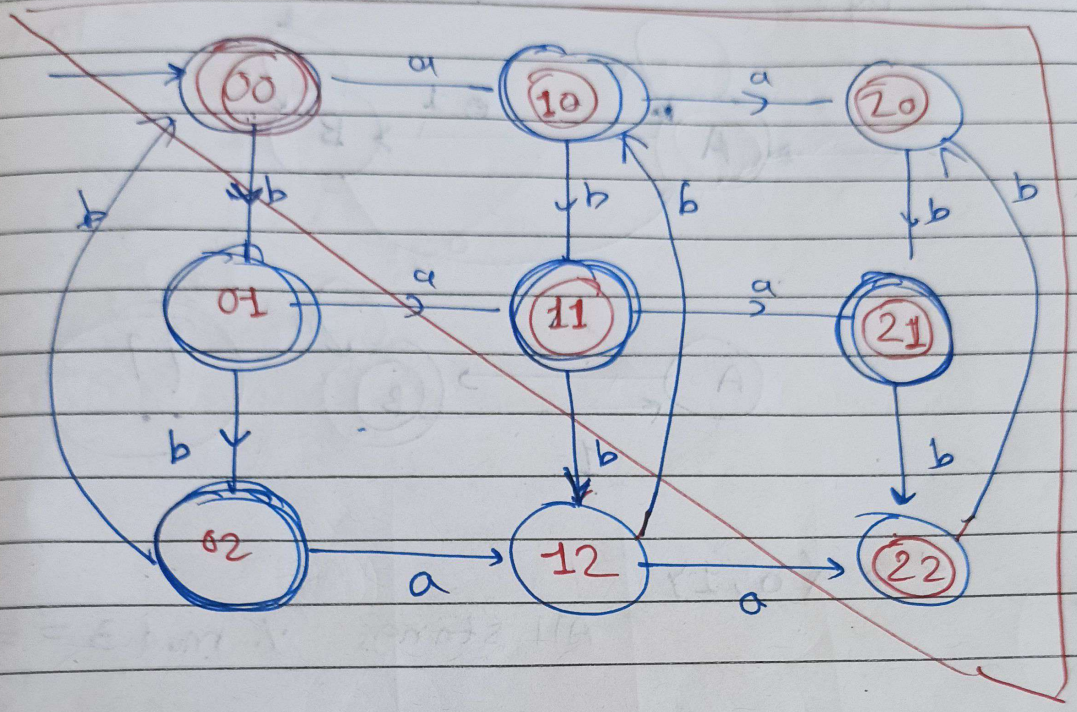
a, 1
a
aa, b
aaaa

2, 1 1, 0

Page No.
Date :

Q. $[n_a(w) \geq n_b(w)] \bmod 3$

aaaab
4%3 = 1
2%3 = 2

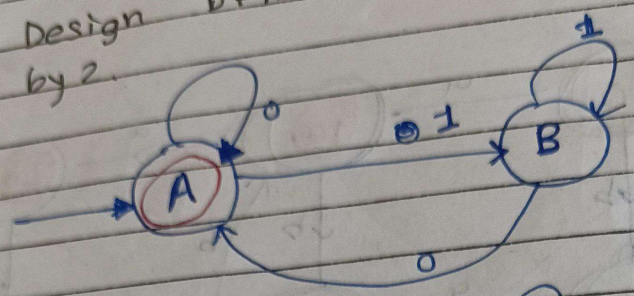


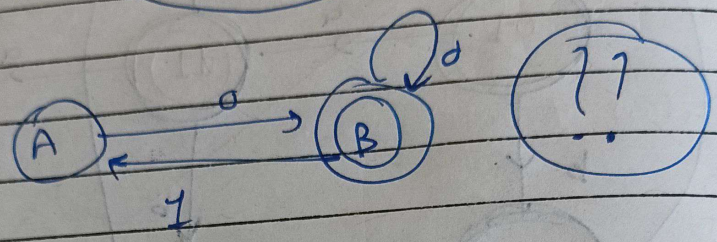
epsilon ≡ eve...

Page No.
Date :

Q. $\Sigma = \{0,1\}$
Design DFA with binary strings divisible by 2.

last digit = 0

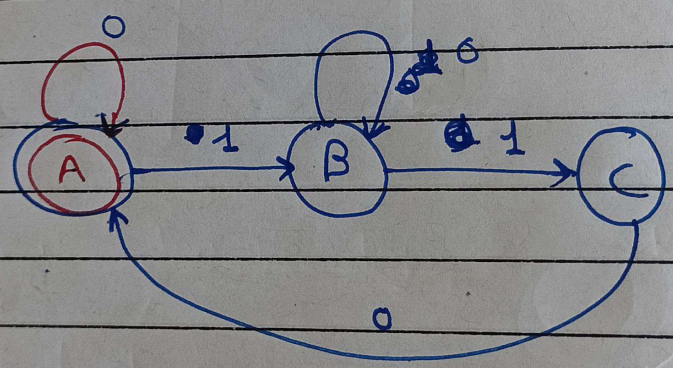




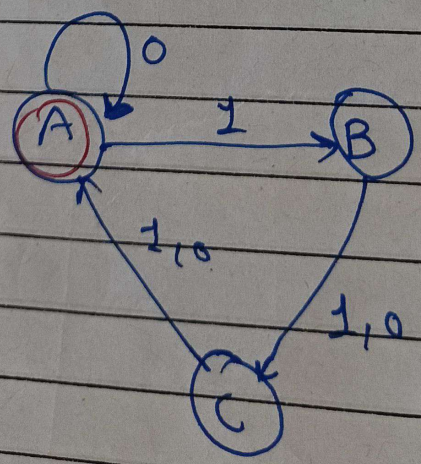
Q. $\Sigma = \{0,1\}$
All strings % mod 3 == 0.

% 3 ⇒ 0, 1, 2 → 3 states

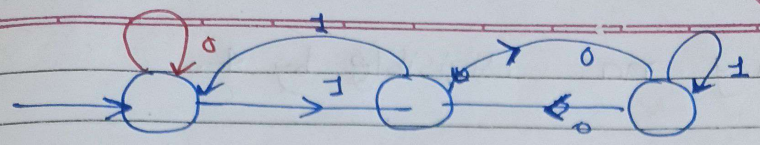
000 →
011 → 3
110 → 6
1001 → 9



000
011
110
1001
1100
⋮



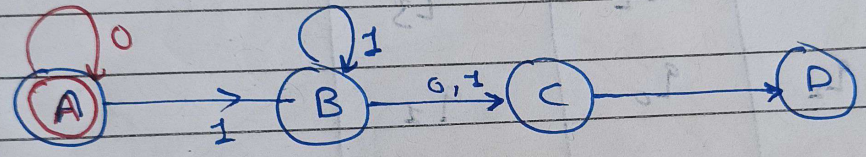
0
110
100



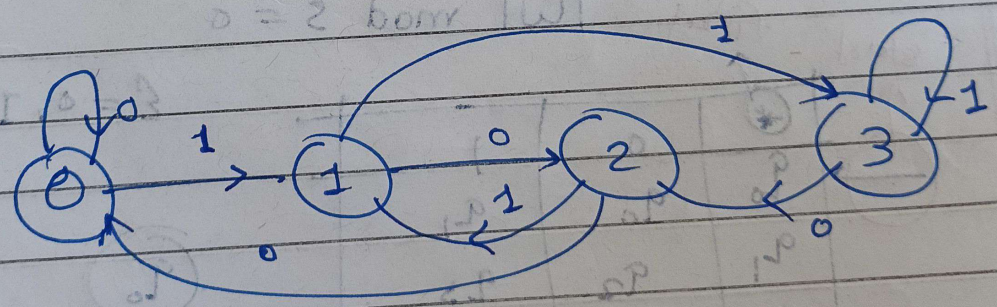
0	1	10
11	100	101
110	111	1000
1001	1010	~~1001~~ 1011
⋮	⋮	⋮

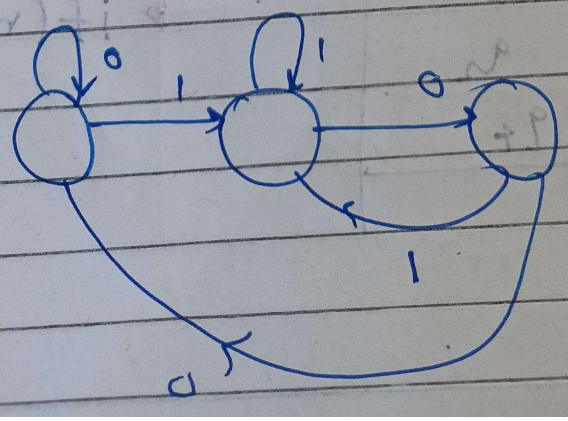
Q. divisible by 4 :-

0	0000
4	0100
8	1000
12	1100
16	



0	1	10	11
100	101	110	111
1000	1001		
1100			

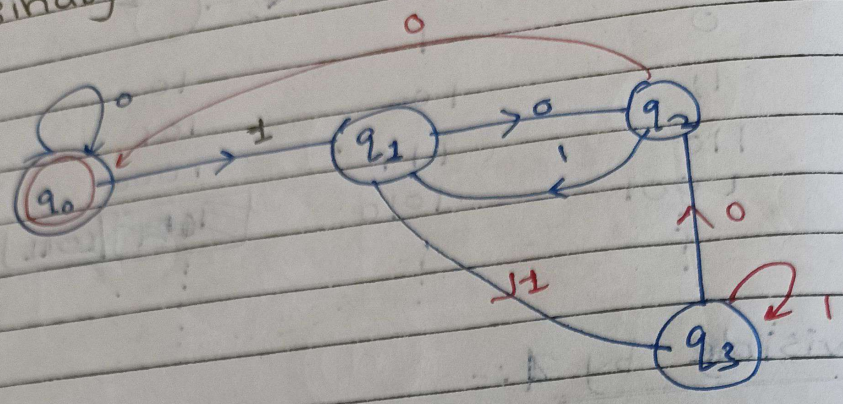




13-1

Binary no. divisible by 4.

Transition



(Transition Table)

	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

Binary no. divisible by 5: Round Robin

$|w| \bmod 5 = 0$

$\Sigma = 0, 1$

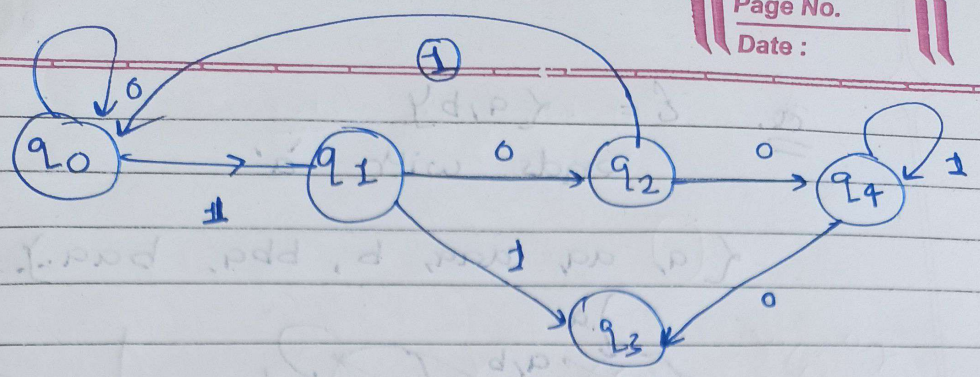
start

final

*	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

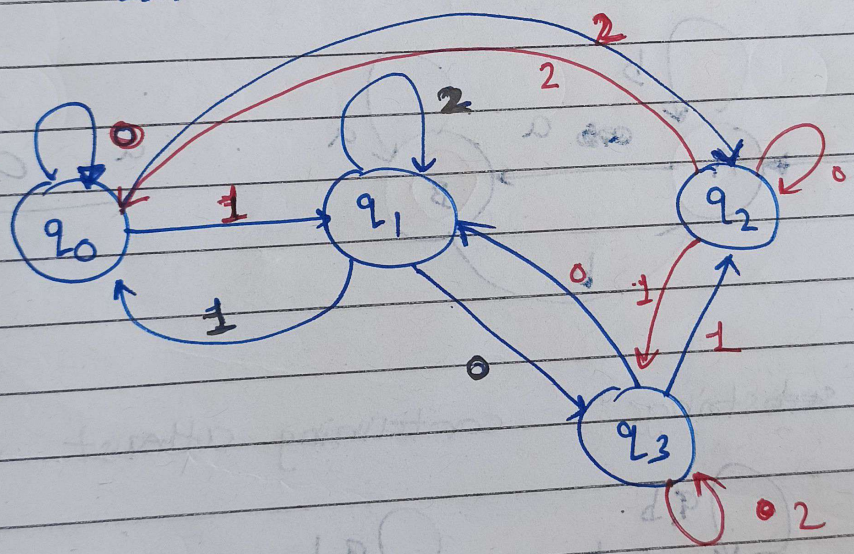
q_0

$\Rightarrow$ if (mod i == 0) * i



Q. $|w| \bmod 4 = 0$ $\quad \Sigma = \{0, 1, 2\}$

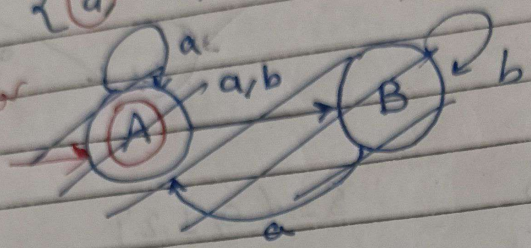
	0	1	2
q_0	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3

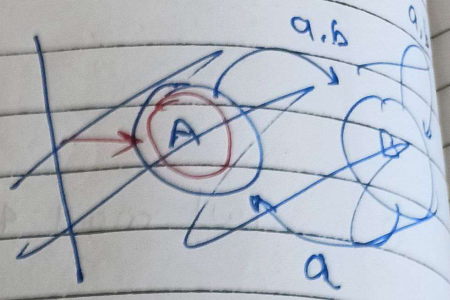


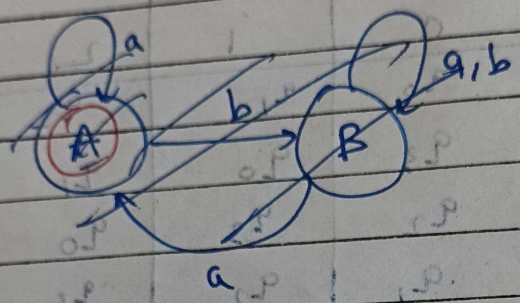
Q. $\Sigma = \{a, b\}$

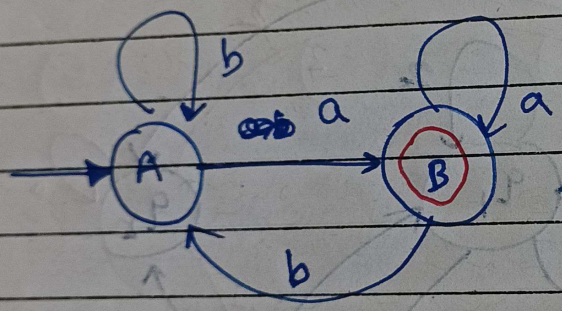
ends with 'a'

{a, aa, aaa, b, bba, baa...}



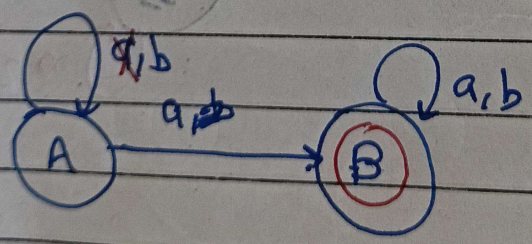


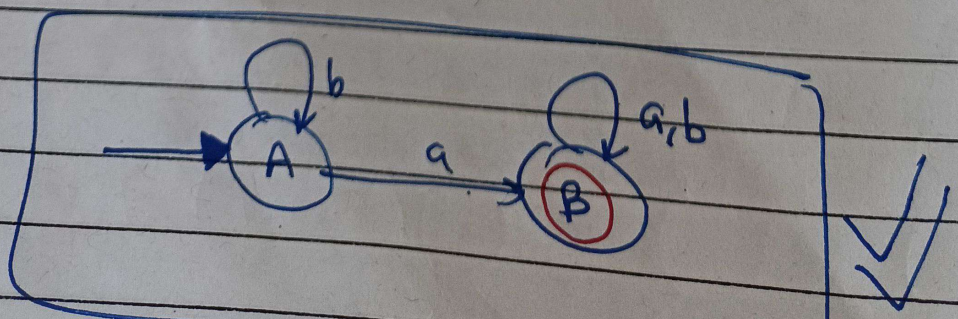




a, aa, aabba,

→ substrings containing atleast one 'a'.

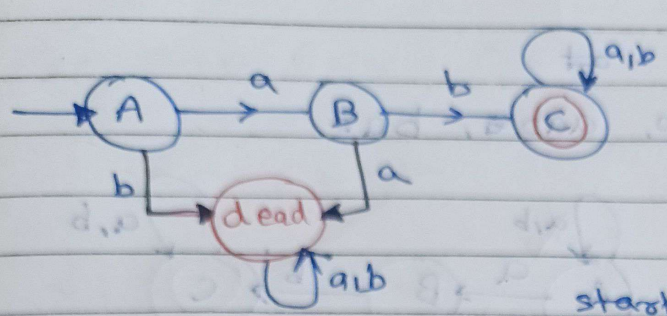




Page No.
Date :

Starts with a,b :- (ab)

aba , abb,
abbb , abaa



starts with b
↓
dead

starts with aa
↓
dead

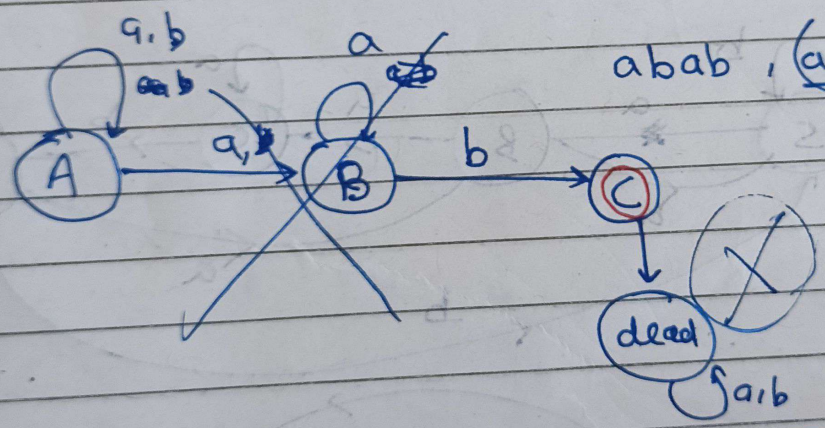
ab → len2

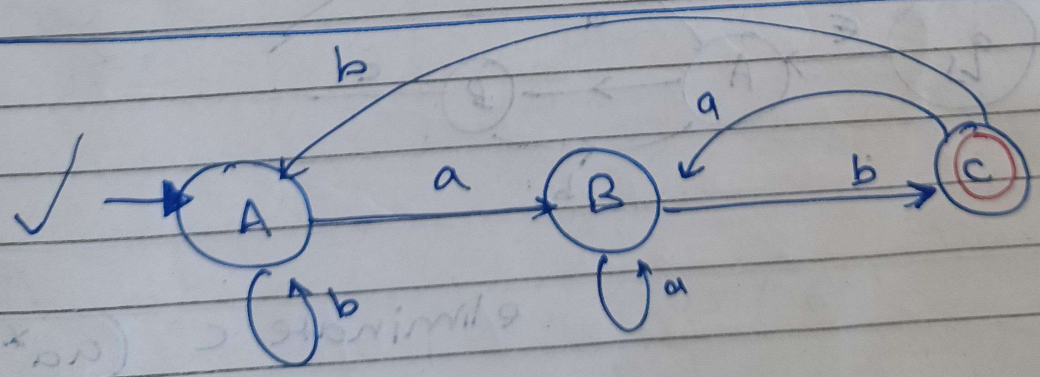
min req state = len+1
= 3

ends with a,b

~~aba~~ aab
ab , aab, bab

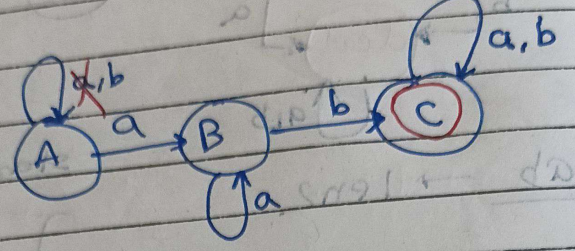
abab , (aabab), (bbab)



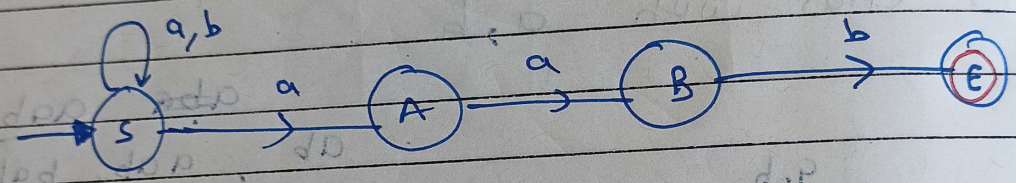


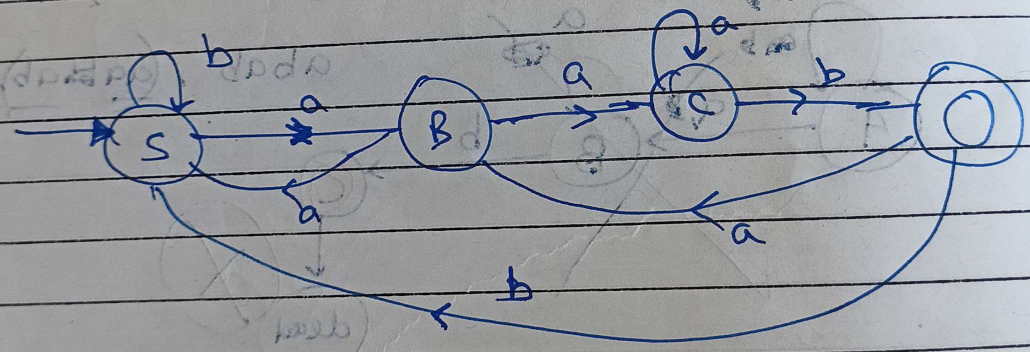
Q. Substring as a, b

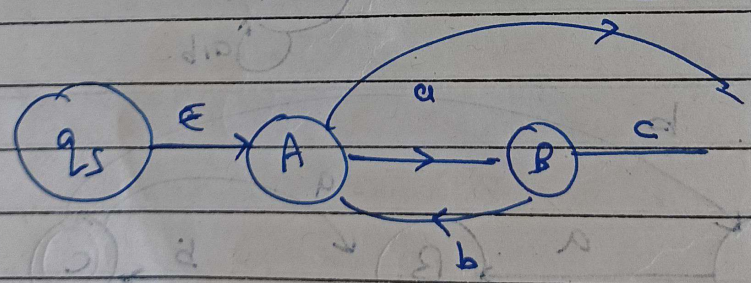
ab, aab, bab



① NFA for ends with a a b.







eliminate c (aa^*b)

$aaa\,a^*b + b.(b+ab)^*\,a\,aa^*b$

ans $= aaa^*b\,(aaa\,a^*b + b\,(a+ab)^*$

$E \rightarrow E+id \;/\; id.$

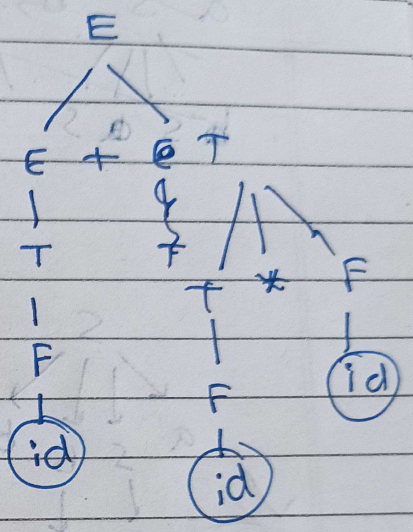
$E \rightarrow E+E \;/\; E*E \;/\; id$

$E \rightarrow E+T/T$

$T \rightarrow T*F/F$

$F \rightarrow id$

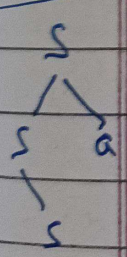
id+id+id

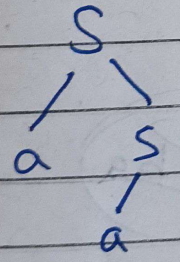


Priority (level-wise)
w/o solving T, we can't add.

Q $S \rightarrow as \;/\; Sa \;/\; a$

$w = aa$





a

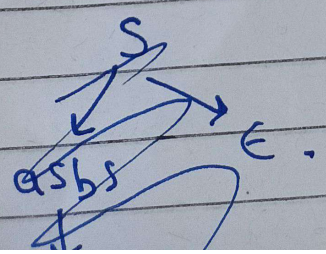
$S \rightarrow as \rightarrow a(a)$

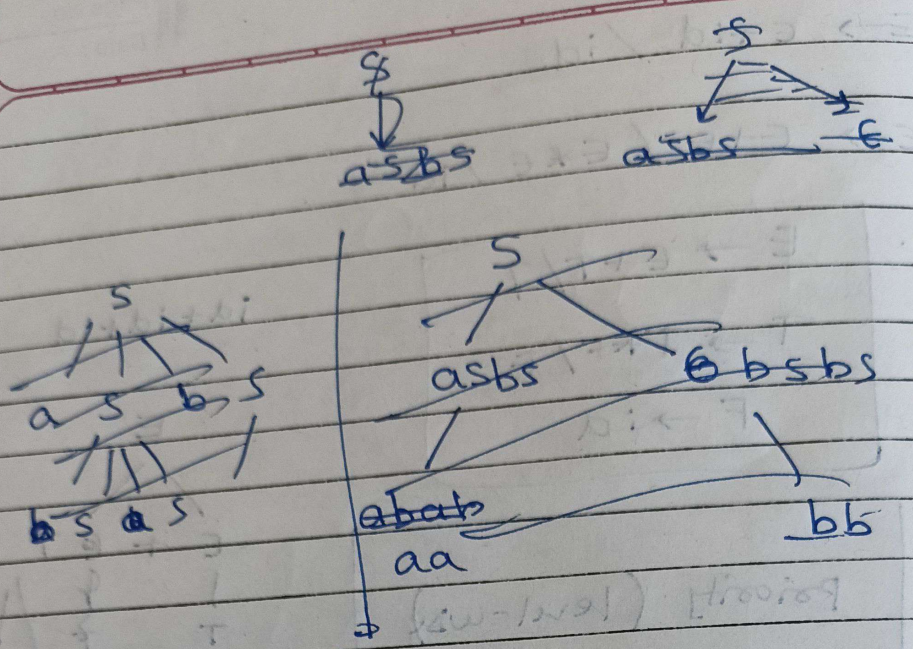
$S \rightarrow Sa \rightarrow (a)a$

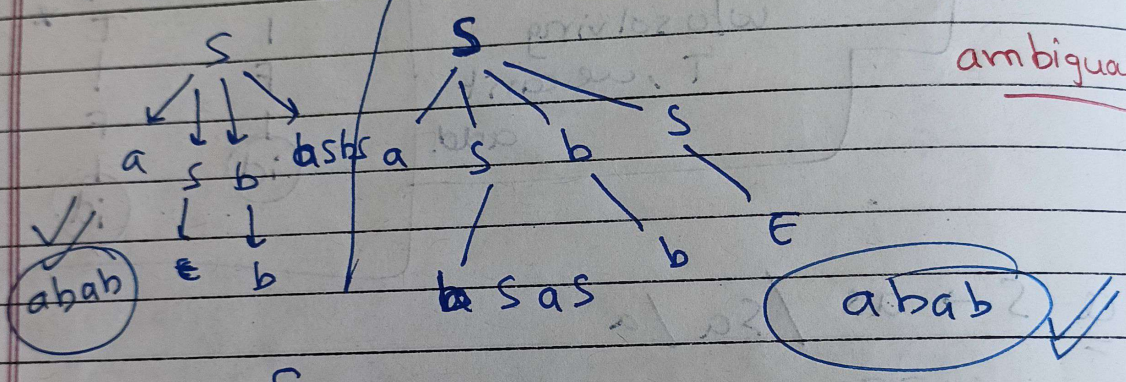
ambi

② $S \rightarrow asbs \;/\; bsas \;/\; \epsilon$

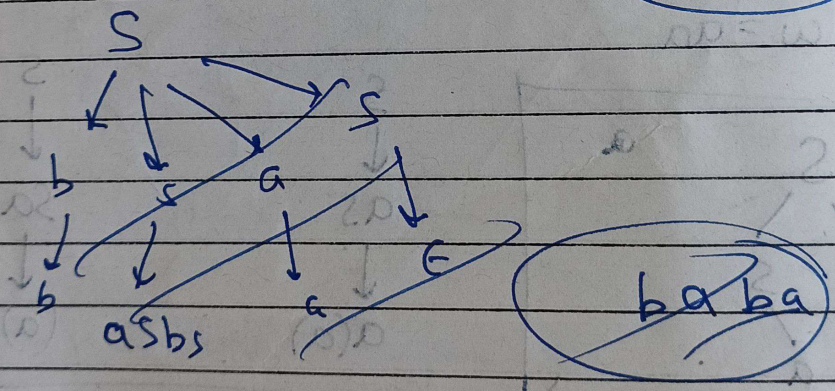
$w = abab$







ambiguous

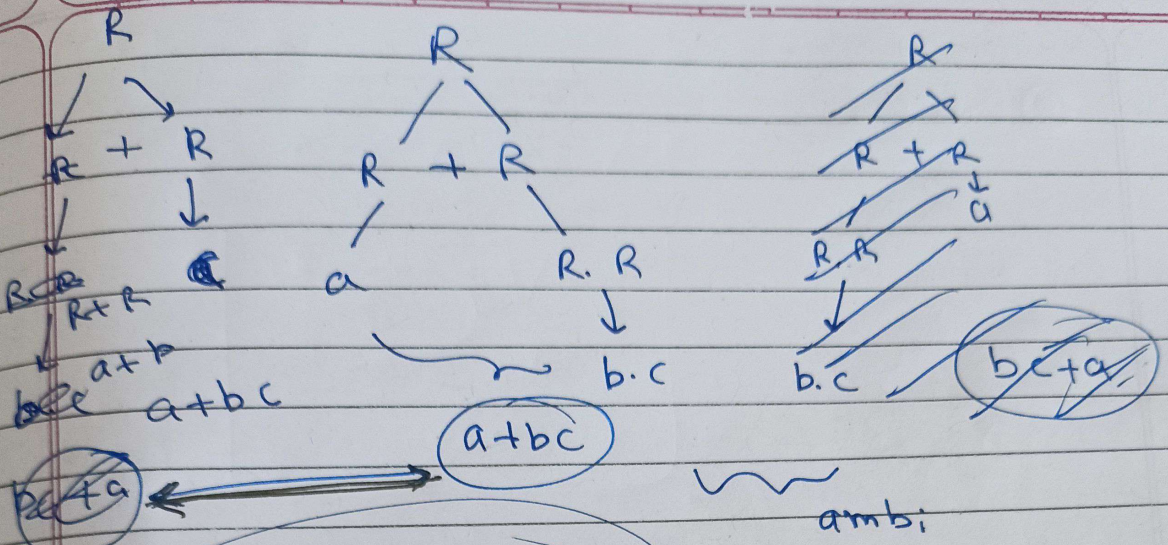


Q. $R \rightarrow R+R$

$R \rightarrow R.R$

$R \rightarrow R^* / a/b/c$

$w = a+bc$



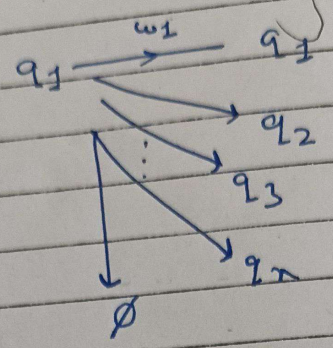
$R \rightarrow R + T / T$

$S \rightarrow S.T$

$T \rightarrow T^* / a / b / c$

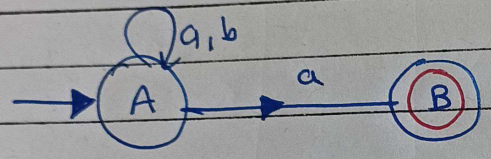
20-1

NFA:-



Q1) $\Sigma = \{a, b\}$, ends with 'a'.

$(a \mid b)^* \underset{a}{\text{———}}$

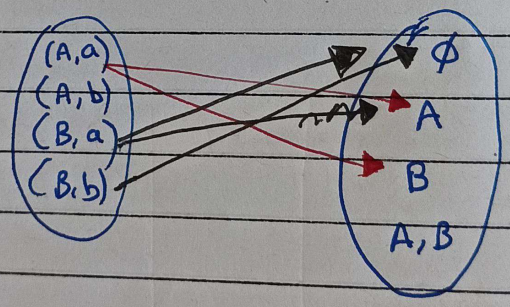


$Q = \{A, B\}$

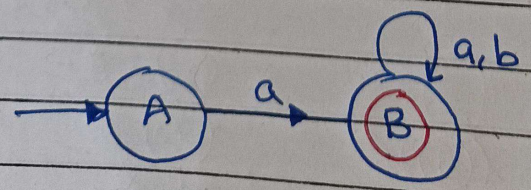
$\Sigma = \{a, b\}$

$\delta : Q \times \Sigma \longrightarrow 2^Q$

Transition function



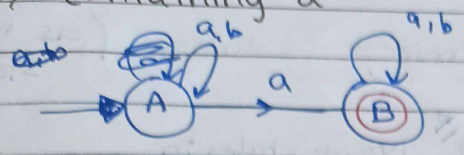
Q.2) starts with a.



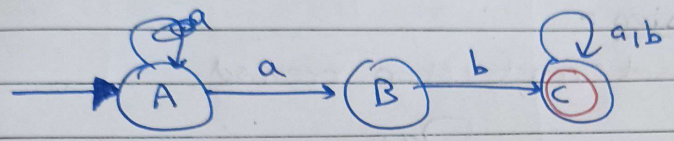
if no path, by default dead / not found

Page No.
Date :

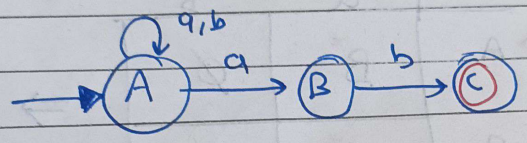
Q.3) containing 'a'



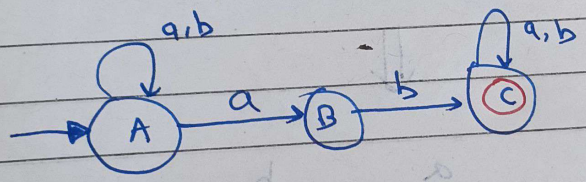
Q.4) starts with a,b



ends with a,b



containing a,b

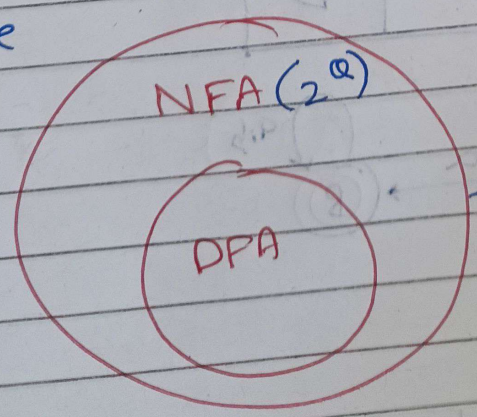


Conversion of NFA → DFA :-

→ which is powerful ? NFA or DFA

Both are equal since both are inter-convertible

in sense of



→ Every NFA is not DFA.
→ " DFA is NFA

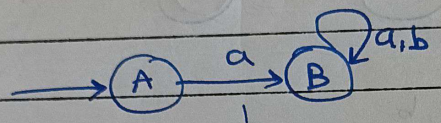
Page No.
Date :

→ NFA is most optimized solution (as minimum states as possible)

No. of states in NFA ≤ No. of states in DFA.

Conversion :-

① Subset construction method



	a	b
→ A	B*	∅
B	B*	B*

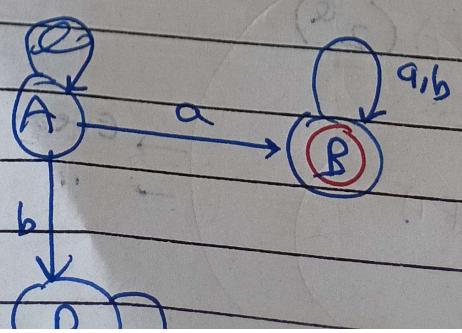
⇒ convert to DFA

⇓

	a	b
→ A	B*	D
B	B*	B*
D	D	D

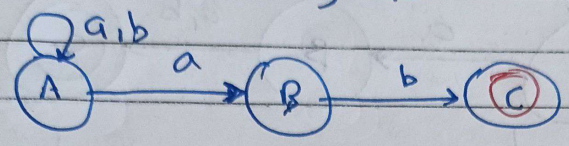
A→b in NFA is ∅, write Dead(D) in DFA

Add on to vertical table as many as new state come in.



Page No.
Date :

~~ends with~~ ~~containing~~ ends with a,b

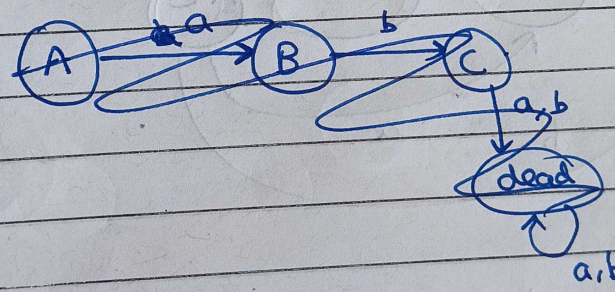


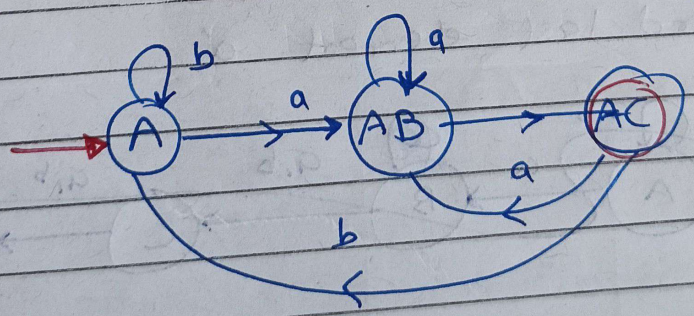
	a	b
→ A	A, B	A
B	∅	C
*C	∅	∅

A → b = B
A → a = A
B → a = ∅
AB

	a	b
→ A	A,B	A
B	D	C
C	D	D
D	D	D

	a	b
A	AB	A
AB	AB	AC
AC	AB	BA
~~BC~~		

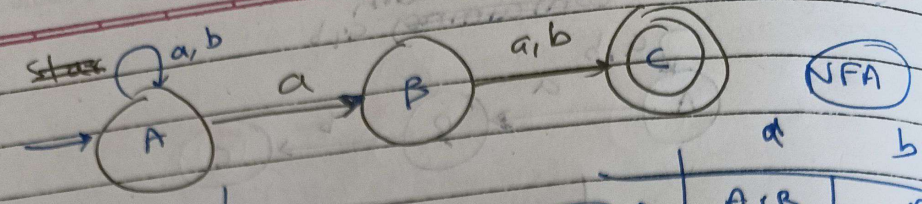




$A \to a = A$
$A \to b = \cancel{B}$

Page No.
Date :

iii) ~~Start~~



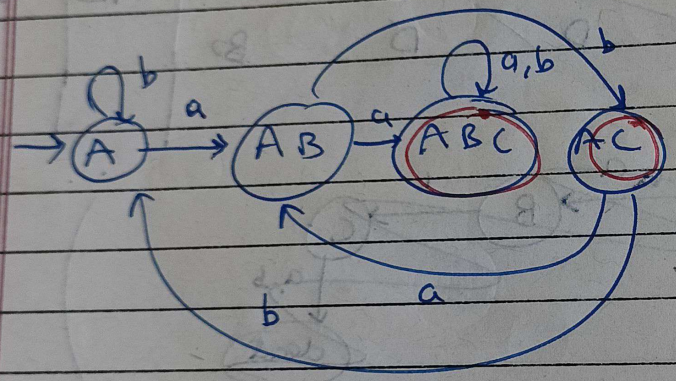
	a	b
A	~~A~~B	A
AB	ABC	AC
~~B~~		
ABC	ABC	~~ABC~~ AC
AC	AB	A

	a	b
A	A,B	A
B	C	C
C	$\emptyset$	$\emptyset$

$AB \to a$
$A \to a = B$
$A \to a = A$
$B \to a = C$
A

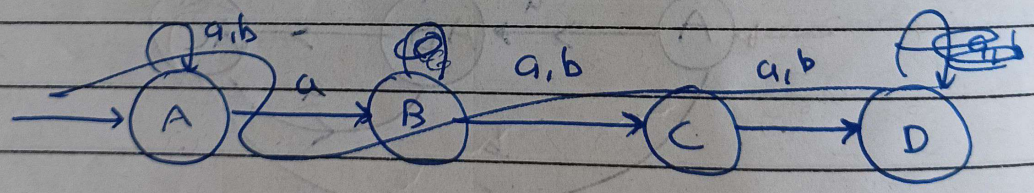
$AC \to a$
$A \to a$

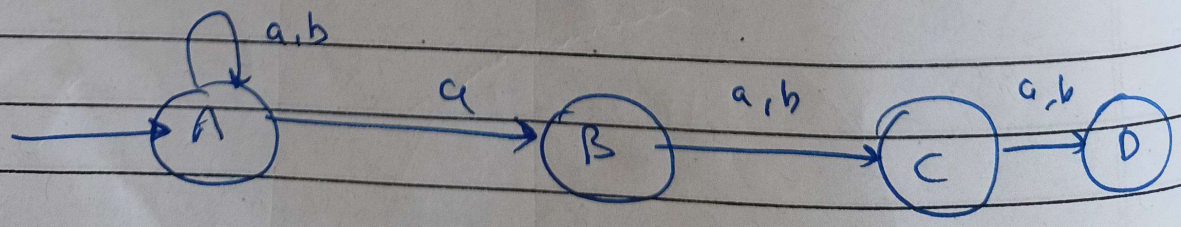
$AB \to b$
$A \to b = \cancel{A}$
$B \to b = \cancel{C}$
AC



23-1

iv) Third last element 'a'





	a	b
A	A, B	A
B	C	C
C	D	D
D	$\emptyset$	$\emptyset$

	a	b
A	AB	A
AB	ABC	AC
ABC	ABCD	ACD
AC	ABD	AD
AD	AB	A
ABD	ABC	AC
ABCD	ABCD	ACD
ACD	ABD	AD

A , AB, AC, , AD, ABD, ABCD, ABC, ACD

AB → a

A → a = AB

D →

AB → b = A

A →

ABC → a

ABCD

ABC → b

ACD

23-1

Minimisation of DFA :- (Reduce no. of states).

Let (p, q) are two states. They are equivalent ~~p → w~~ if transition $\delta(p, w) \in$ Final

$\Rightarrow \delta(q, w) \in$ Final

$\delta(p, w) \in$ Final $\Rightarrow \delta(q, w) \in$ Final

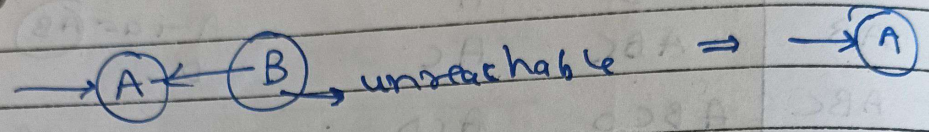
$\delta(p, w) \in$ Final $\not\Rightarrow \delta(q, w) \notin$ Final

$|w| = 0 \rightarrow 0\text{-eq}^n$

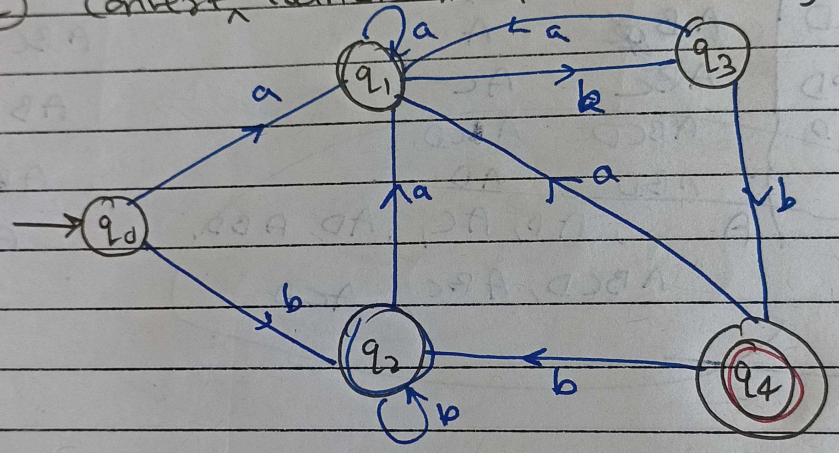
$|w| = 1 \rightarrow 1\text{-eq}^n$

$|w| = 2 \rightarrow 2\text{-eq}^n$

(1) Remove the unreachable state from start



(2) Convert to transition table & apply 0, 1, 2 equivalence



Step 1) → from state, we can reach all (None is unreachable)

Step-2) →

	a	b
q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4 *
q_4	q_1	q_2

Page No.
Date :

step-3) find 0 equivalent $\{q_0 q_1 q_2 q_3\}$ $\{q_4\}$

÷ 2 ~~states~~ partions, (final clubbed & non-final clubbed

1 equivalent

$\{q_4\}$

$(p, q) \Rightarrow (q_0, q_1)$ $\{q_4\}$

$\{q_0 \to a = q_1\}$ NF ✓ $\{q_0 \to b = q_2\}$ NF ✓

$\{q_1 \to a = q_1\}$ NF ✓ $\{q_1 \to b = q_3\}$ NF ✓

allowed

~~$\{q_0 q_1 q_2\}\{q_3\}\{q_4\}$~~

check for q_2 now

If states do not

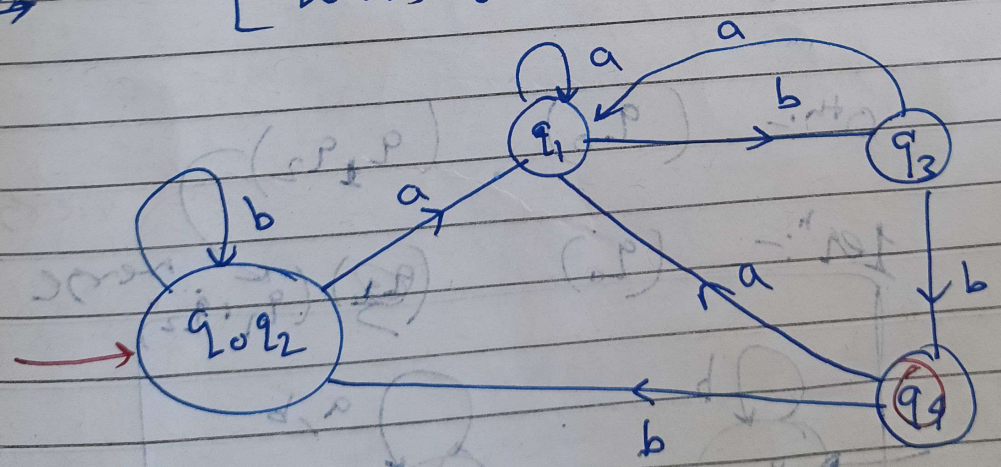
1 eqn :- $\{q_0 q_1 q_2\}\{q_3\}\{q_4\}$ (3 states)

2nd eqn :- $[q_0 q_2]$ $[q_1]$ $[q_3]$ $[q_4]$

Now a diff state

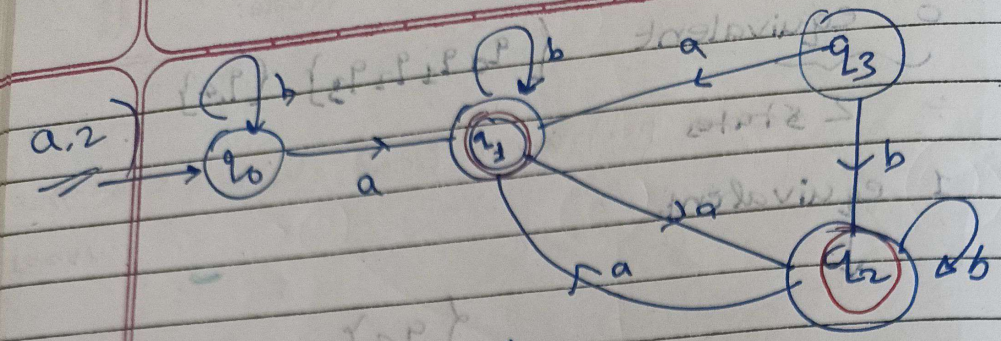
3rd eqn :- $[q_0 q_2]$ $[q_1]$ $[q_3]$ $[q_4]$

Same states



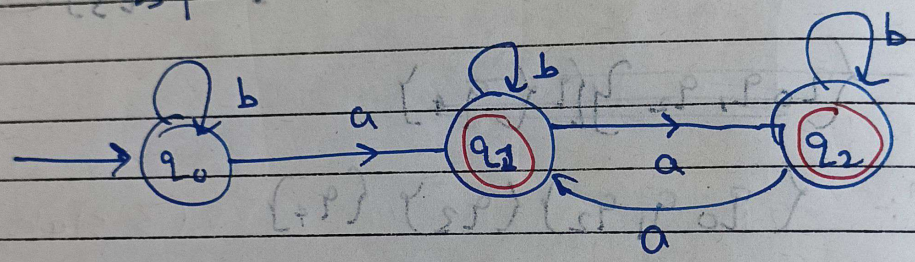
Page No.
Date :

a.2



	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2
q_3		

→ Deleting q_3 as unreachable



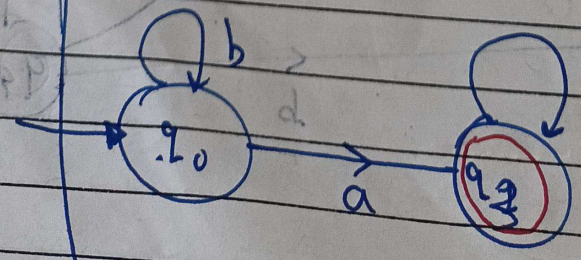
	a	b
q_0	q_1	q_0
* q_1	q_2	q_1
* q_2	q_1	q_2

0th:- (q_0) $(q_1 q_2)$

1st:- (q_0) (q_1, q_2)

merge

merge possible

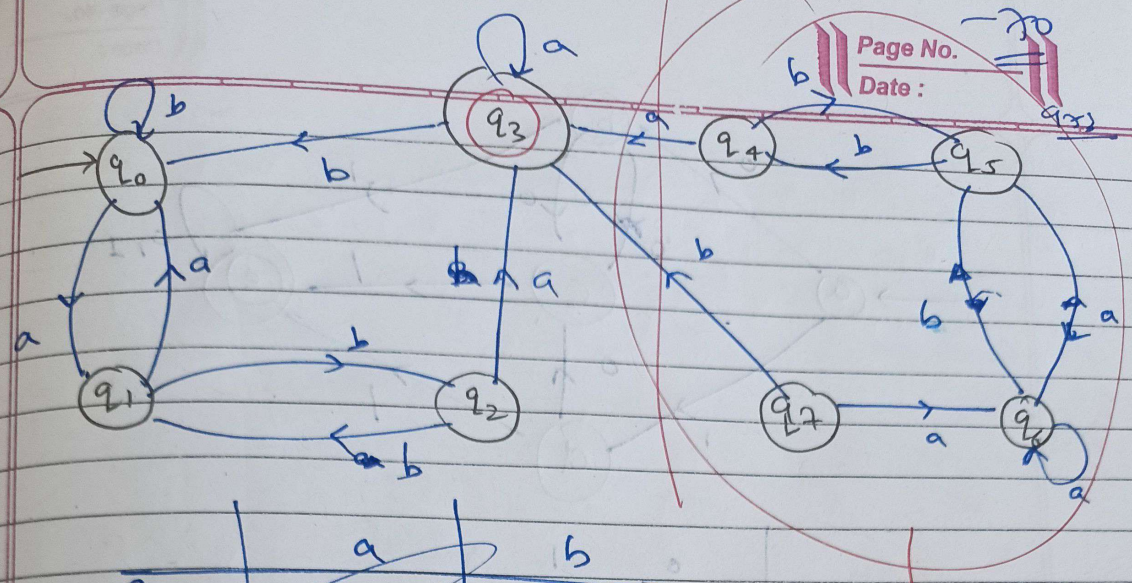


q_3

can be merged

Page No.
Date :

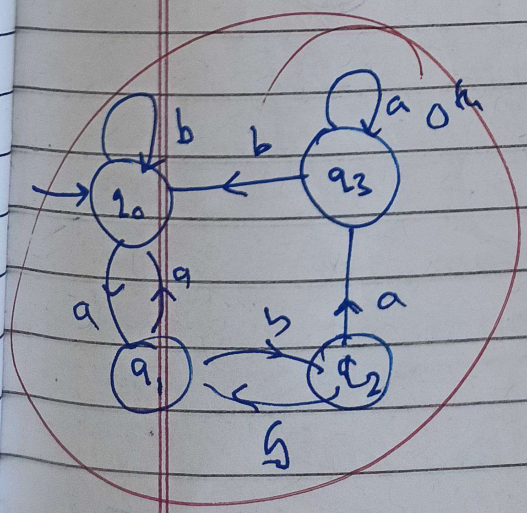
Q.



Unreachable

	a	b
→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3 ~~q_1~~	q_1
* q_3	q_3	q_0
q_4	q_3	q_5
q_5	q_6	q_4
q_6		
q_7		

	a	b
→ q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
→ * q_3	q_3	q_0

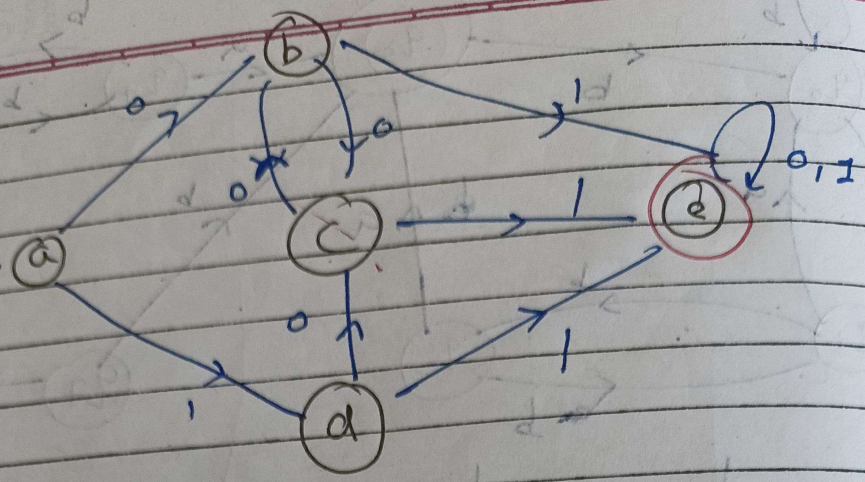


$\{q_0 q_1 q_2\}$ $\{q_3\}$

$\{q_0 q_1\}\{q_2\}$ $\{q_3\}$

4th states

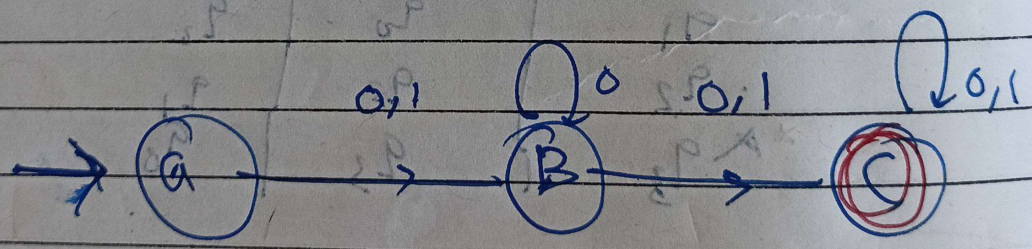
$\{q_0\}\{q_1\}\{q_2\}\{q_3\}$



	0	1
→ a	b	d
b	c	e
c	b	e
d	c	e
← e	e	e

0th $\{a, b, c, d\}$ $\{e\}$

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



Page No.
Date :

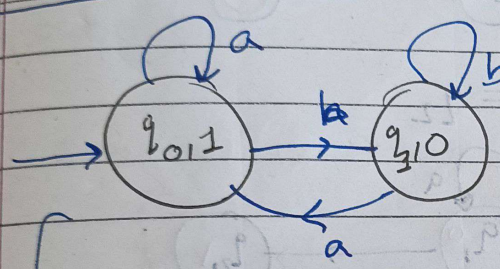
Milley – More :- Finite atutomata with output

Moore — Mealy

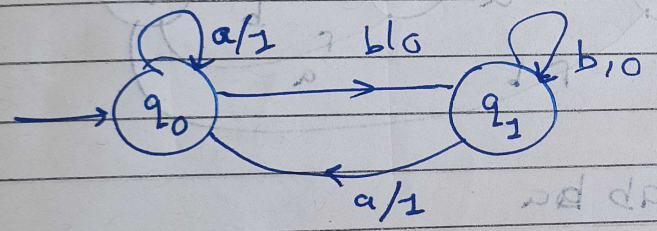
$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$

- Q → States
- Σ → I/P (a,b)
- δ → Transition
- q_0 → start
- Δ → output alphabet (c,d)
- λ → o/p function

no final state

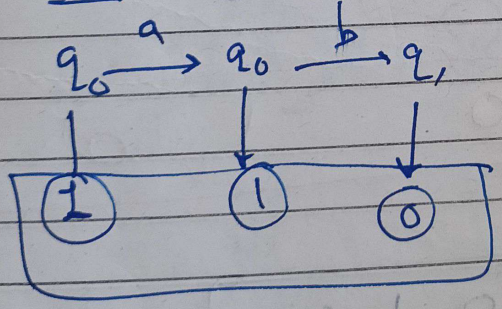


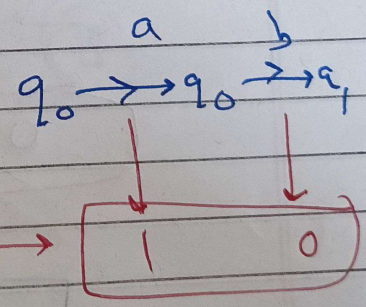
$\lambda : Q \to \Delta$



$\lambda :- Q \times \Sigma \to \Delta$

ab





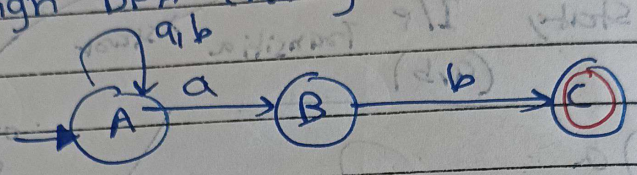
In moore machine → n outputs

In miley " → n+1 "

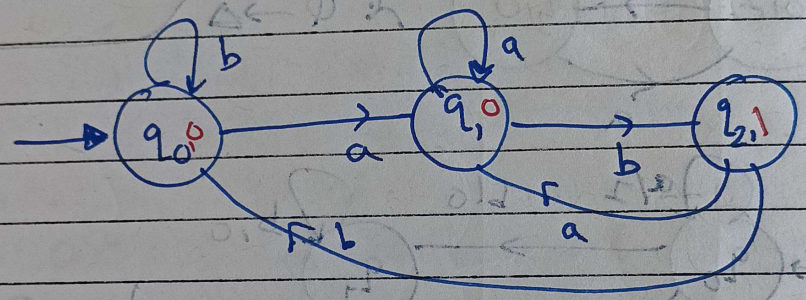
Both will be DFA

Q. Construct a moore machine that ~~consist~~ takes set of all str of $\Sigma = \{a, b\}$ as input & print 1 as output for every occurences of a, b.

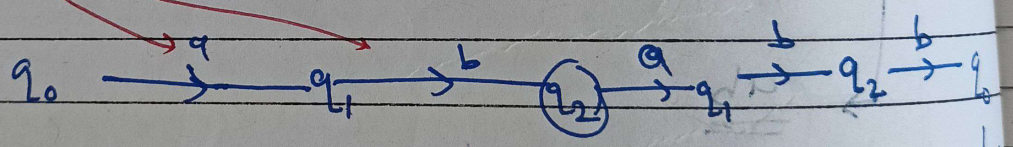
Design DFA ending with a, b.



Assume $C \equiv q_2$



ab ab ba

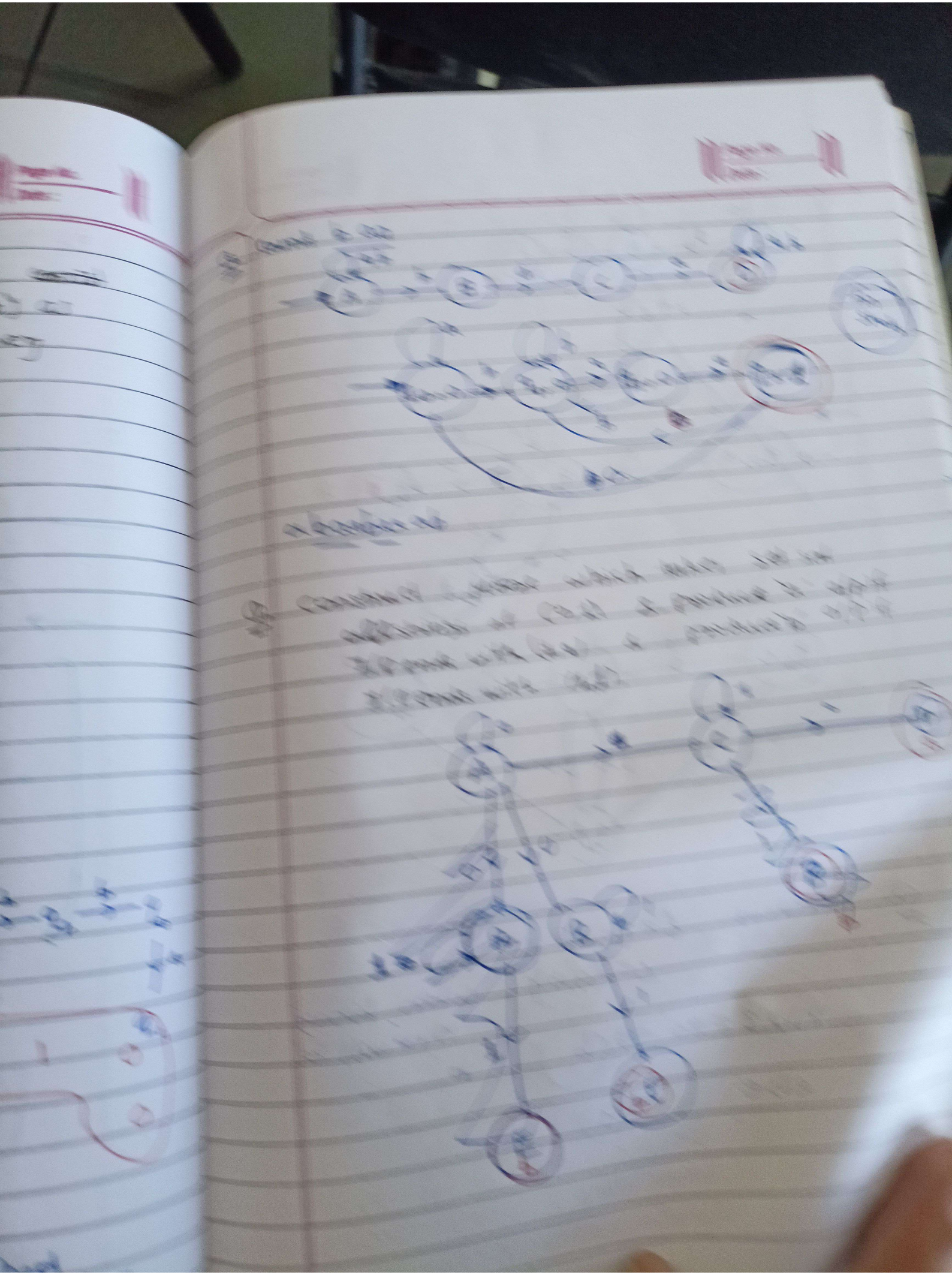


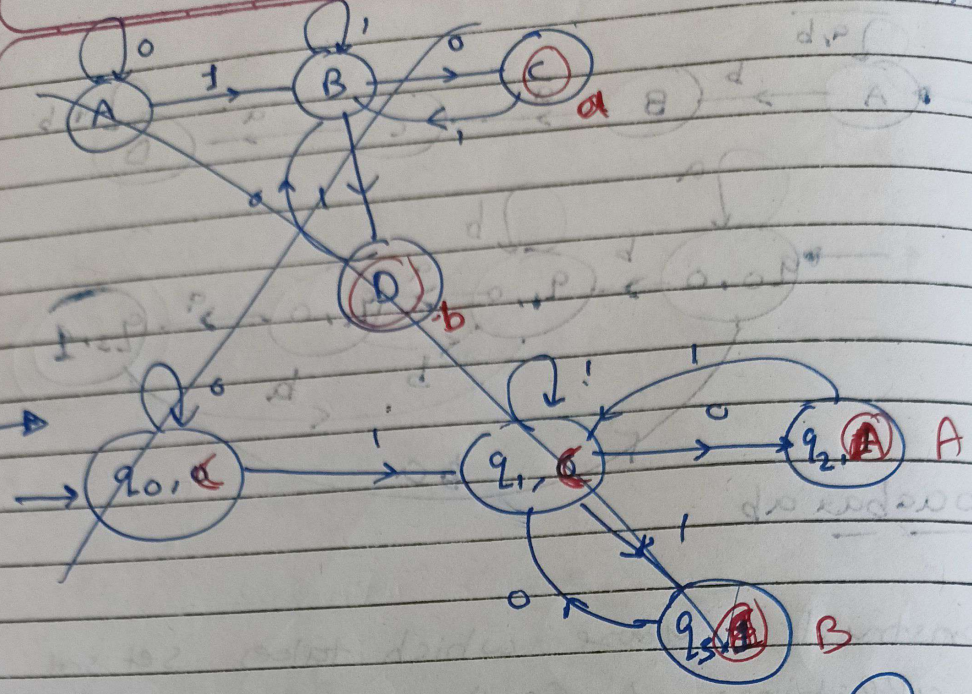
0 0 1 0 1 q_1 0 0

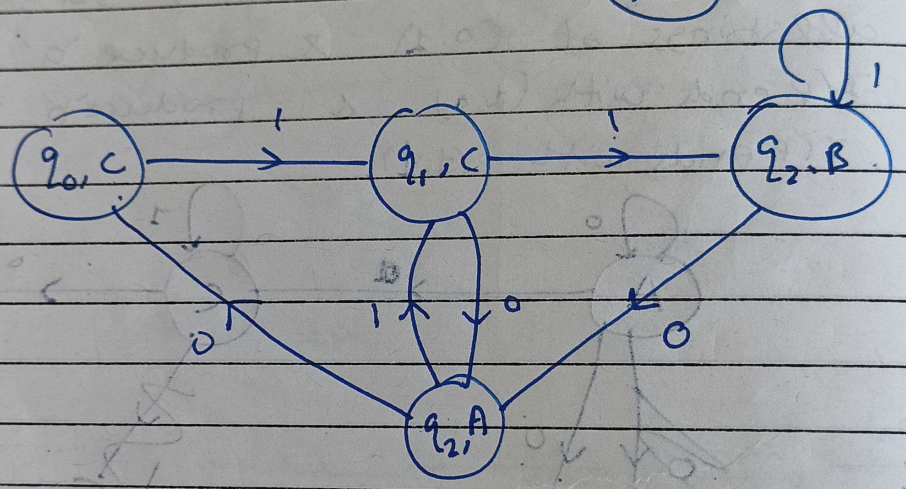
$\boxed{0\ 0\ 1\ 0\ 1\ 0\ 0}$

↓

Two 1's ⇒ input had 2 "AB"







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

$\delta = \{q, n\}$ when binary num will be

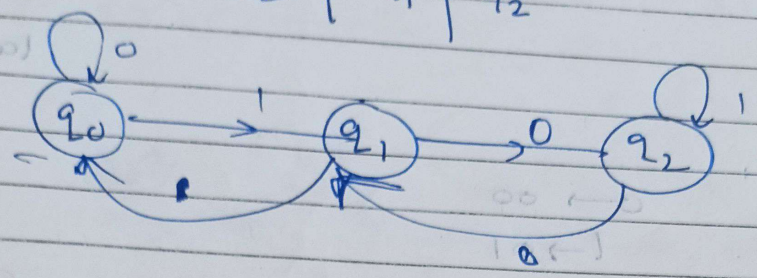
$n \% 3 == 0 \Rightarrow b$

else $\Rightarrow a$

Page No.
Date :

OR

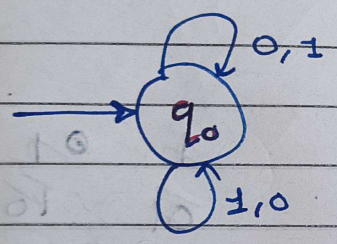
	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2



Mealy Machine :-

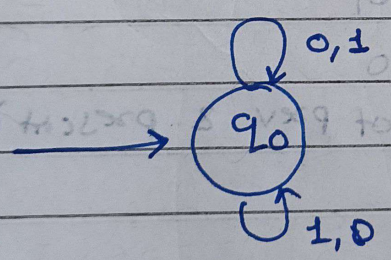
$\lambda \longrightarrow Q \times \Sigma \longrightarrow \Delta$

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



at every state, diff input ⇒ diff output (1's complement)

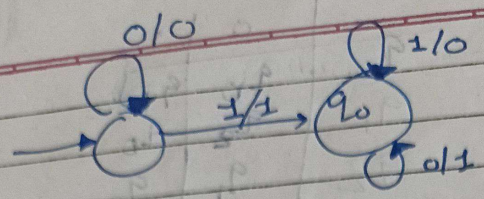
2's complement



LSB ⟶ MSB

	1010
1's	0101
2's	+1
	0110

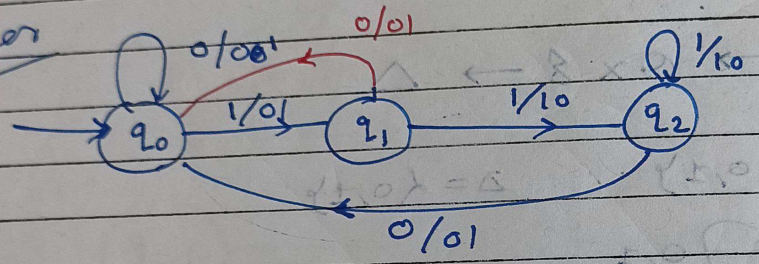
	0011
	1100
2'nd	~~1111~~
	+1
	1101



till 1st one from LSB → MSB, write same & later take 1's complement

0 → 00
1 → 01
1 → 10
1 → 10

Answer



⟹ find question?

~~1 1 0~~
~~01 10 01~~

0	1	1	0
↓	↓	↓	↓
00	01	10	01
0	1	3	1

~~1 1~~ ~~01~~ ~~01~~ ~~10~~

(a) 11 → 01
(b) ~~10 → 00~~
(c) ~~sum of prev & present bit~~
(d) ~~none~~

1	0	1
↓	↓	↓
01	01	01

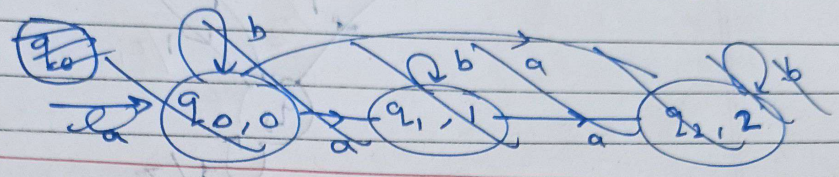
Page No.
Date :

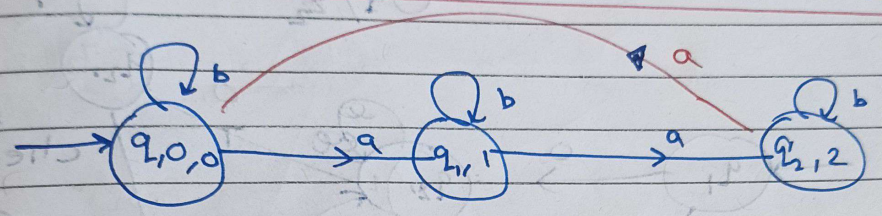
→ Conversion of Moore → Mealy.

→ Design Moore mc counting number of a's % 3.

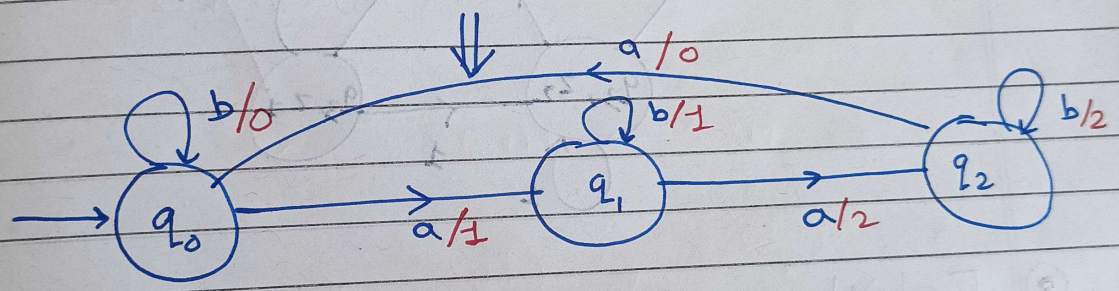
$\Sigma = \{a, b\}$ → o/p states

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

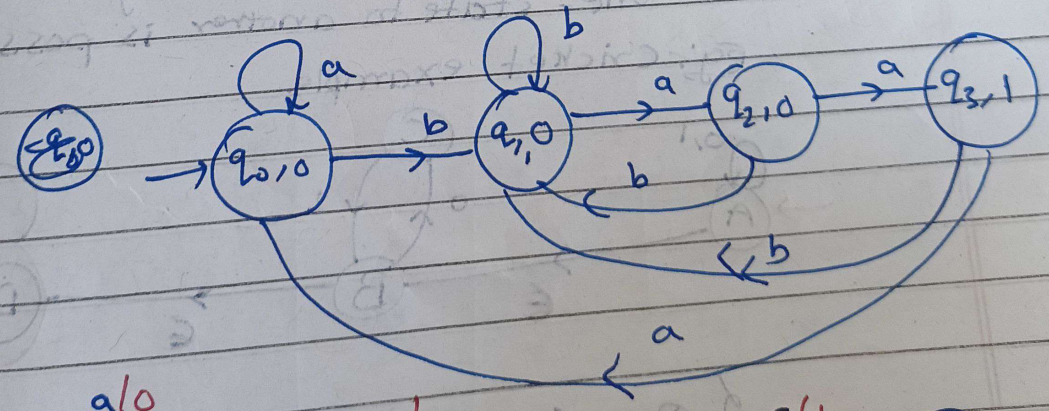


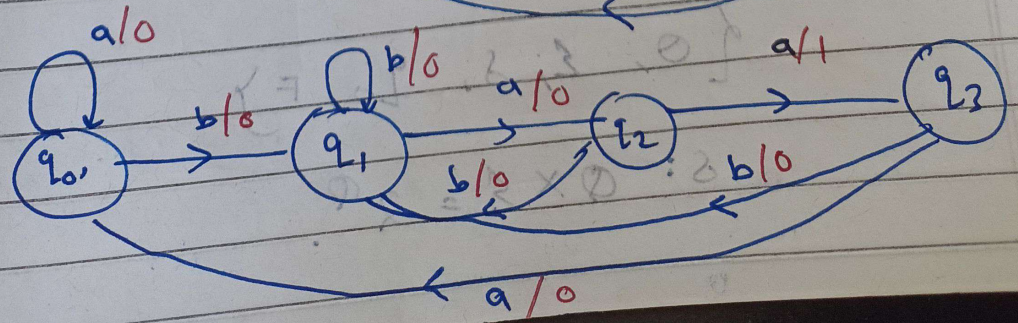


Moore → Milley (No. of states = will be same)



→ Counter of baa



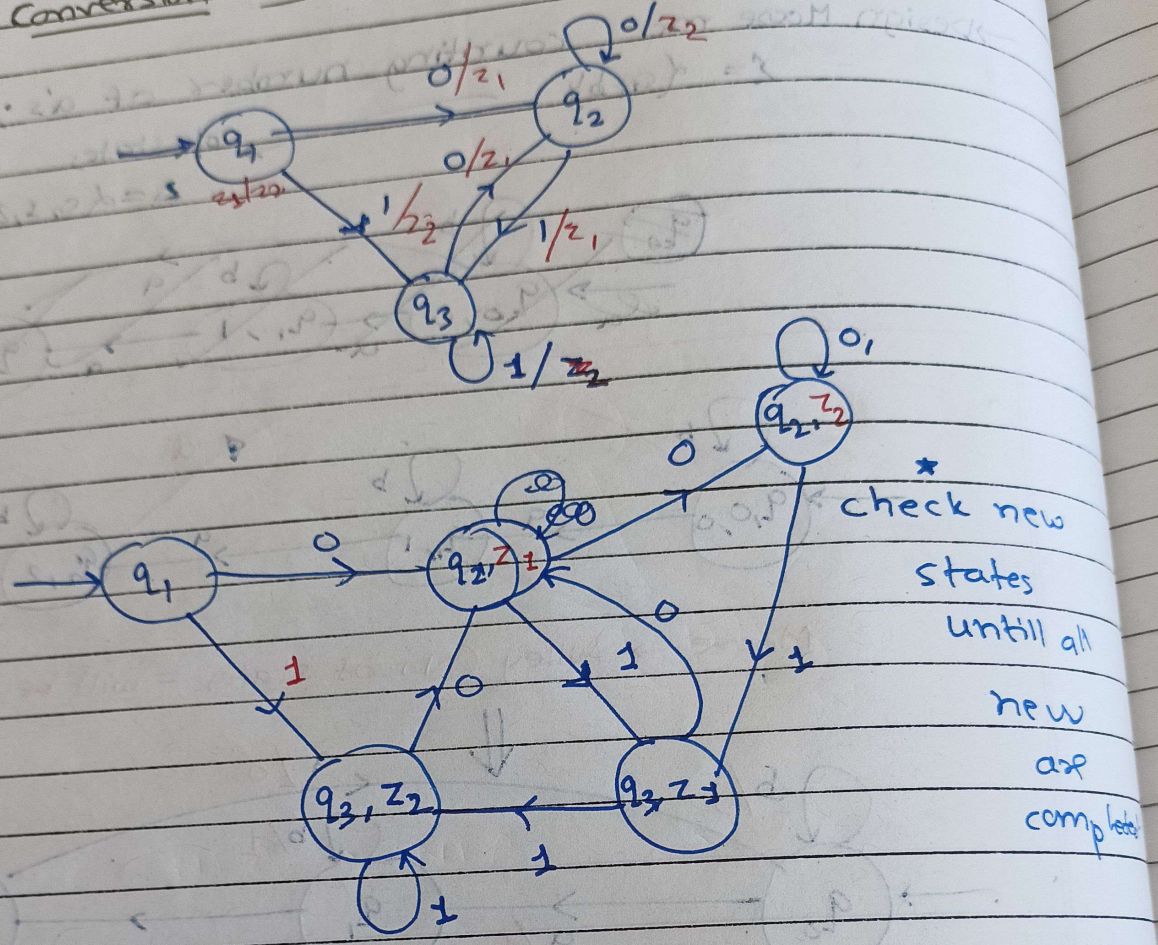


NFA → DFA ≡ Milley → Moore

→ Milley is more optimised.

Page No.
Date :

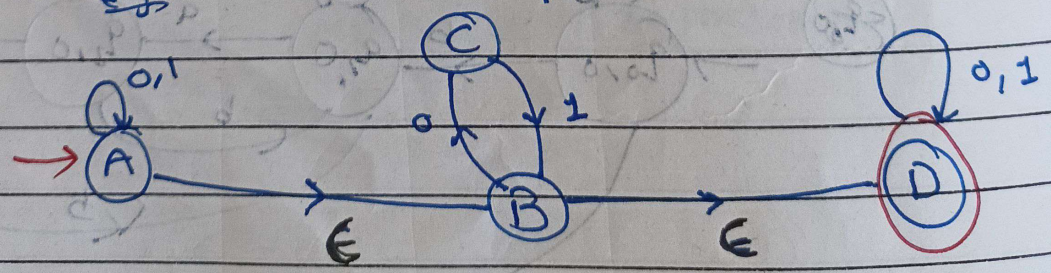
⇒ Conversion from Milley to Moore:-



Ephsilon NFA :-

→ w/o any input, switching from one-state to another is possible.

eg:- Cricket example.



$\{Q, \Sigma, \delta, q_0, F\}$

$\delta : Q \times \Sigma = 2^Q$

In epsilon NFA, $\delta: Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$

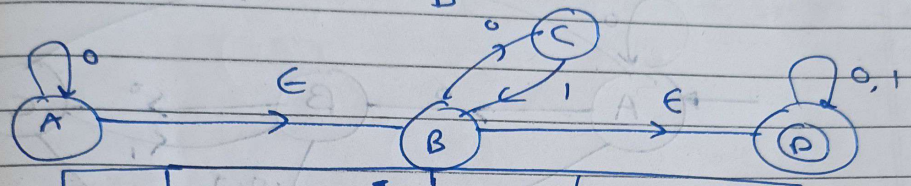
ε-closure (A) :- $= \{A, B, D\}$

$A \xrightarrow{\epsilon} A$

$A \xrightarrow{\epsilon} B$

$B \xrightarrow{\epsilon} D$

A, B, D → all are closure of epsilon



	0	1
A	A, B, C, D	D
B	C, D	D
C	∅	B, D
D	D	D

C → B → D

A(0) → find A(ε) → state x → find x(0) → find x(ε) → union

A(1) → ϵ^* 1 ϵ^*

A → ε → A → ∅ → ∅ ∅

→ B → ∅ → ∅ ∅

→ D → D → D D

Take union

D

B(0) → B D

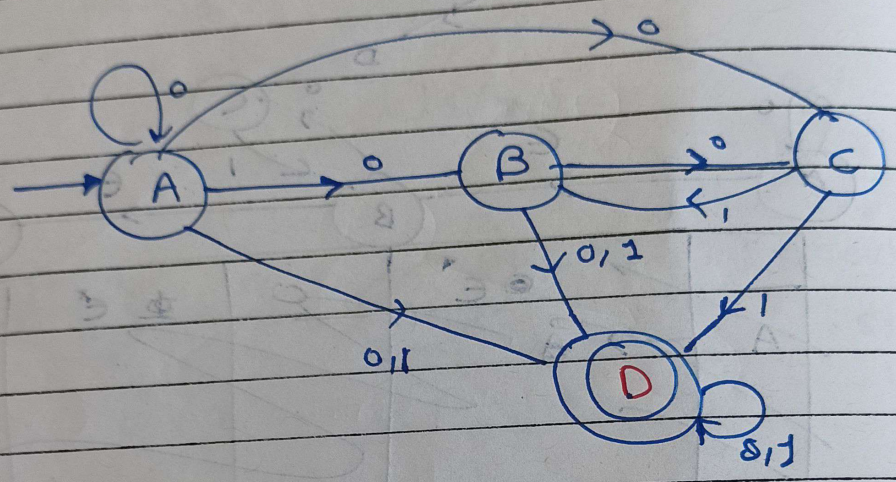
ϵ^* 1 ϵ^*

B C ∅ C

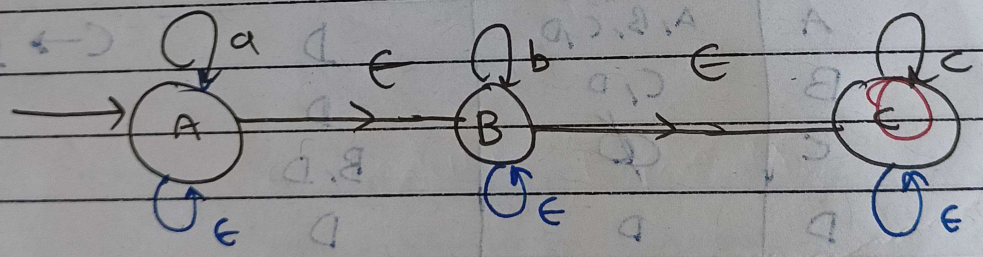
D D ∅ D D

C, D D

	0	1
A	A, B, C, D	D
B	C, D	D
C	∅	B, D
D	D	D



∈-NFA ⟶ NFA



	a	b	c
A	A, B, C	B, C	C
B	∅	B, C	C
C	∅	∅	C

Page No.
Date :

A

ϵ^*	a	ϵ^*
A	A	A, B, C
B	ϕ	ϕ
C	ϕ	ϕ

$\boxed{A, B, C}$

ϵ^*	b	ϵ^*
A	ϕ	ϕ
B	B	B, C
C	ϕ	ϕ

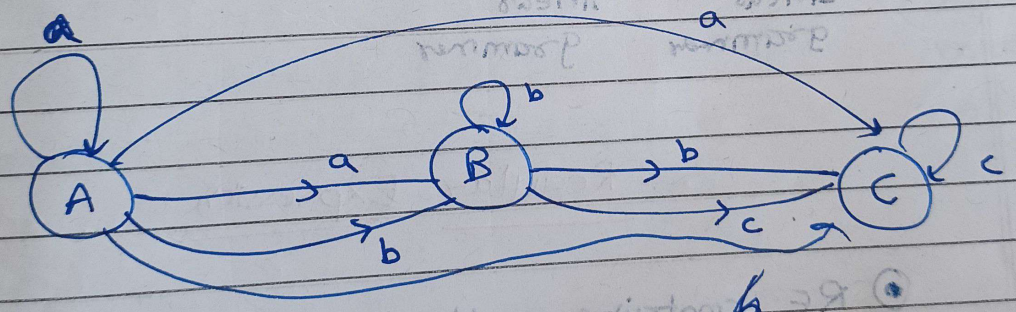
ϵ^*	c	ϵ^*
A	ϕ	ϕ
B	ϕ	ϕ
C	c	c

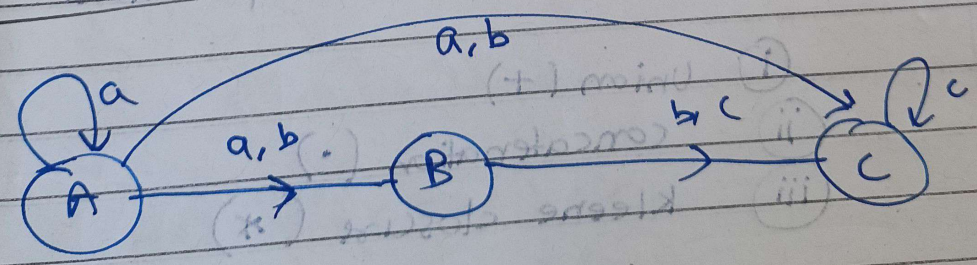
B

ϵ^*	a	b	ϵ^*	c
B	ϕ	B	ϕ	ϕ
C	ϕ	ϕ	ϕ	c

C

ϵ^*	a	b	c	ϵ^*
c	ϕ	ϕ	c	c





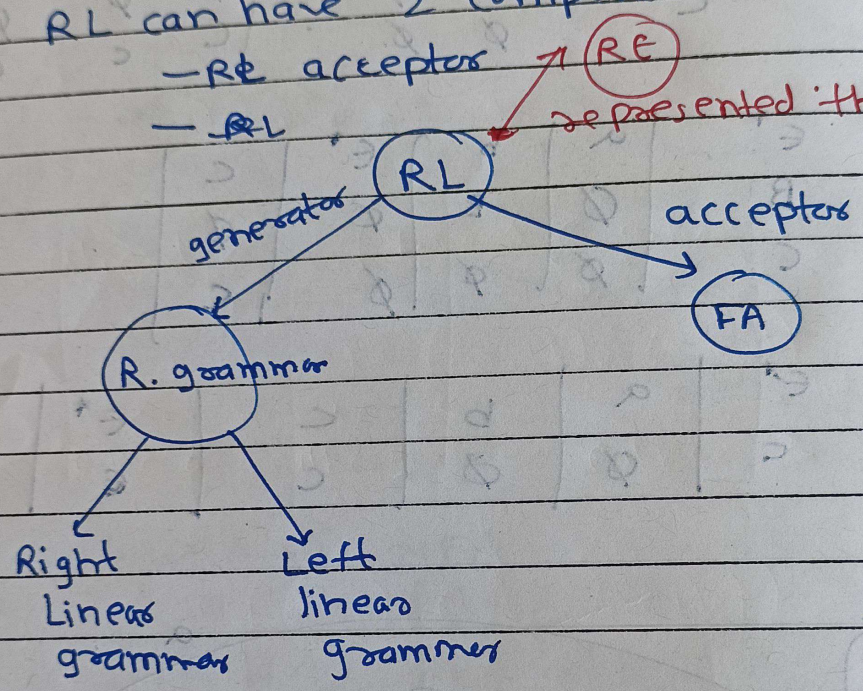
Page No.
Date :

Q. → FA doesn't have any memory element (Drawback)

Regular Grammar & Lang:

Regex :- Regular expression (Pattern matching)

⇒ RL can have 2 components
- RE acceptor
- RL



Regular Expression

RE contains three operators:

(i) Union (+)
(ii) concatenation (·)
(iii) Kleene closure (*)

$\epsilon^* x = x$ $\quad$ $\phi^* x = \phi$

empty ~~empty~~ null

Page No.
Date :

$\Rightarrow \Sigma = \{a, b\}$ $\quad$ $\epsilon, \phi, \{a\}, \{b\}$ $\quad$ singletons

primite expression

$a+b$

$a.b$

a^*

$a.\phi \rightarrow \phi$

$a.\epsilon \rightarrow a$

$a^* = \{\epsilon, a, aa, aaaa, aaaa \ldots\}$

$a^* \Rightarrow$ (a repeats)

$(a+b)^* = \{\epsilon, a, aa, aaa, \ldots b, bb, bbb \ldots$
$abaab \ldots bb \ldots\}$

any number of a or b.

3-2

$a^3 = aaa$ $\quad$ $a^* = a$ (any no. of times) $\rightarrow$ समझाने के लिए (no notation exist like this)

$(a+b)^* \rightarrow$ either a or b.

~~(a+b~~

$a.b$ = a followed by b.

Q $\Sigma = \{a, b\}$ each string len == 2.

$L = \{aa, ab, bb, ba\}$.

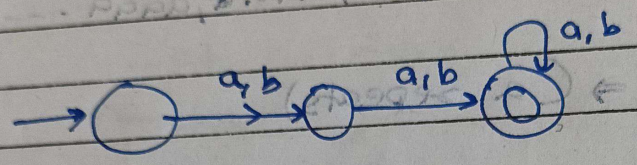
$RE = aa + ab + ba + bb$
$= a(a+b) + b(a+b)$
$= \underbrace{\boxed{(a+b).(a+b)}}_{\text{string of exactly 2}}$

Page No.
Date :

Q. ⇒ string len 3 = $(a+b).(a+b).(a+b)$

Q. → Size is atleast 2.

$$\underbrace{(a+b).(a+b)}_{\text{exactly 2}}.\underbrace{(a+b)^*}_{0 \text{ or } 1 \text{ or } 2 \ldots \ldots \text{ or } m}$$



Q. atmost 2. $(L = \{\underbrace{\epsilon}_{0}, \underbrace{a, b}_{1}, \underbrace{aa, ab, ba, bb}_{2}\})$

$$\underbrace{\epsilon}_{\text{either } 0} + \underbrace{(a+b)}_{\text{either } 1} + \underbrace{((a+b).(a+b))}_{\text{either } 2}$$

$$\cancel{\epsilon + (a+b)(\epsilon + (a+b))}$$

$$\cancel{\Rightarrow \epsilon(\epsilon + (a+b)) + (a+b)}$$

$$\epsilon^2 + (a+b)\epsilon + (a+b).(a+b)$$

$$\Rightarrow \boxed{x^2 + (a+b)x + ab}$$

$$\downarrow$$

$$(x+a)(x+b)$$

$$\Rightarrow \boxed{(\epsilon + a + b)(\epsilon + a + b)}$$

SJF H.

Page No.
Date :

Q. even length of str

ϵ, aa, ab
ba, bb

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

a^*

a^+ $\boxed{a^+ = a^* - \epsilon}$

Q. for odd length

~~$\epsilon + (a+b) + [(a+b)^* (a+b) (a+b)^*]$~~

1+ even

$\Rightarrow (a+b) \cdot [(a+b) \cdot (a+b)]^*$

Q. length % 3 == 0

$[\underbrace{(a+b) \cdot (a+b) \cdot (a+b)}_{\text{length 3 ka multiple}}]^*$

Q. length % 3 == 2

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

Q. No. of a's == 2, b can be any.

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

$b^* \cdot (a) \cdot b^* \cdot (a) \, b^*$

Q. atleast 2 ā

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

$b^* \; a^+ \; b^* a^+ \; b^*$

Q atmost 2 ā

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

$\Rightarrow \boxed{b^* (\epsilon + a) a + ab^* ab^*)}$

$\Rightarrow \boxed{b^* (\epsilon + a (\epsilon + b^* a b^*))}$

$\Downarrow$

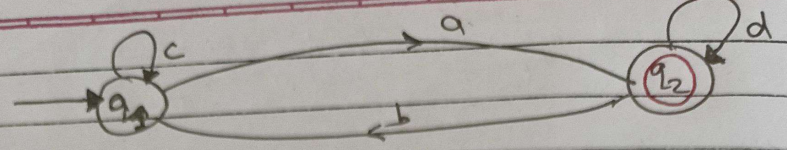
$\boxed{b^* (\epsilon + a) \; b^* (\epsilon + a) \; b^*}$

Q. no. of A should be even

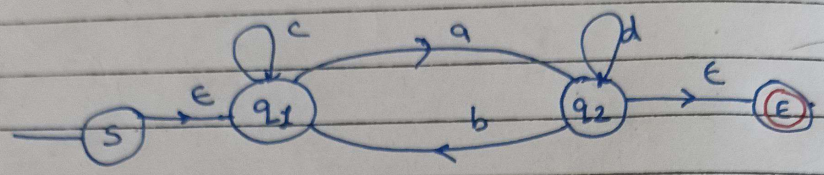
$b^* (aa)^* \; b^* (aa)^* \; b^*$ ✗ babba

$\boxed{[b^* a \; b^* a \; b^*]^* + b^*}$ $b^* \cancel{(ab)^*}$

$\rightarrow b^* (ab^* a)^*$

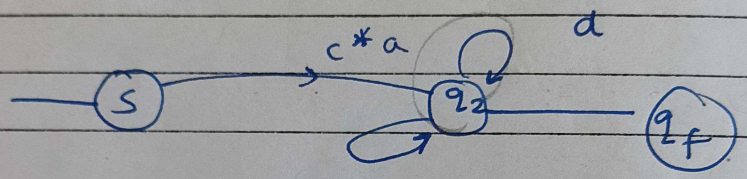


Q. Convert to REGEX:-



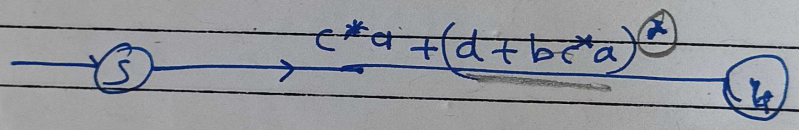
⇓

Removing q_1



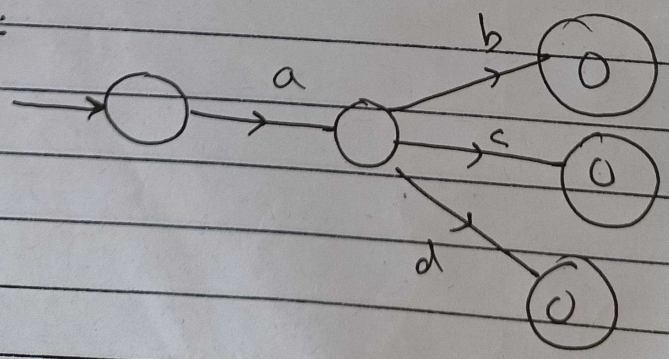
↓

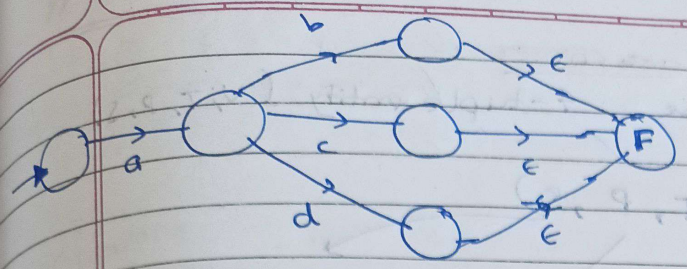
$(d+bc^*a)$



Regex $\boxed{c^*a(d+bc^*a)^*}$

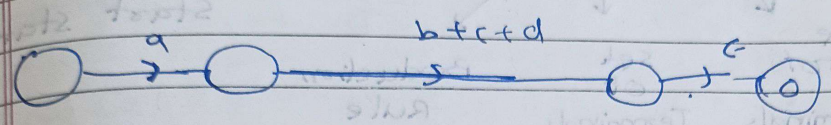
Q.



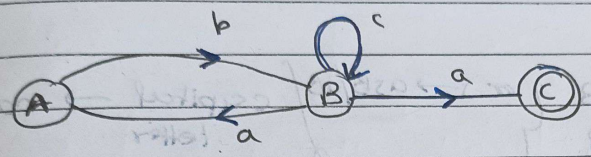


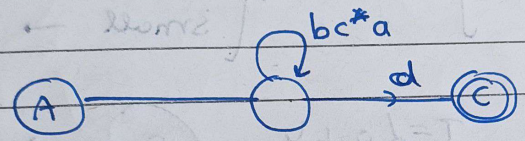
Regex =

$\boxed{a.(b+c+d)}$



Q.

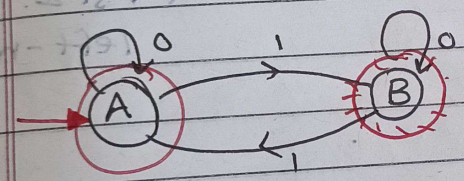


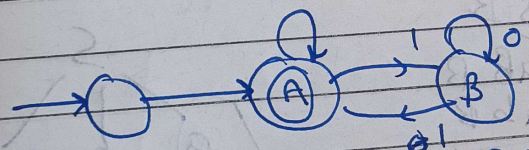


$\cancel{(bc^*a)^*d}$

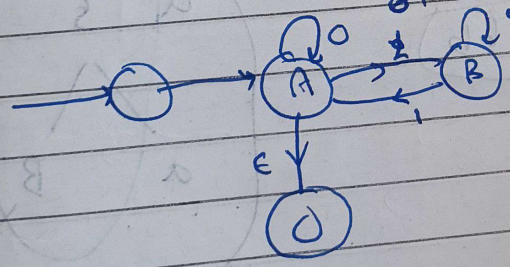
$\boxed{b(c+ab)^* d}$

Q.





$(0^* + 1\,0^*\,1)^*$



$\Rightarrow \boxed{(0 + 10^*1)^*}$

Introduction to grammer :-

Grammers are 4-tuple entity (V, T, P, S)

(V, T, P, S)

- V → finite set of non-terminals
- T → set of Terminals
- P → Production Rule
- S → Start state.

P:
- S → aSB
- S → aB
- B → b

or S → aSB/aB

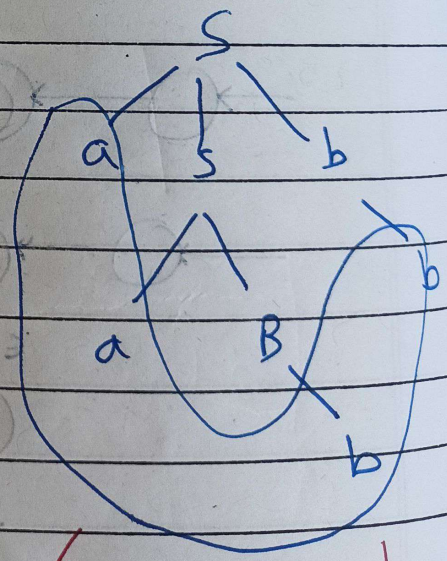
capital letter → non terminal

small → terminal

{V = {S, B}} T = {a, b} S → start state (first & left-most)

Derivation

S → aSB
aaBB
aabB
aabb



★ Syntax analysis / Parse-Tree

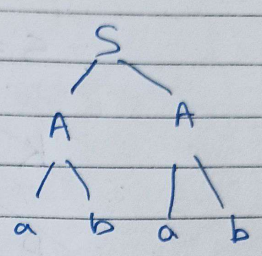
$a^n b^n$ $n=3$ aaa bbb
$n \geq 1$ $n = \text{LOL}$

Page No.
Date :

Q. $\Sigma = \{a, b\}$ length = 2 L = aa, bb, ab, ba

RE = (a+b)(a+b)

writing grammer :- $S \rightarrow AA$
$A \rightarrow \cancel{a/b}\ a/b$



Q. $L = a^n / n \geq 0 \leftrightarrow \boxed{a^*}$

generated grammer

$\cancel{S \rightarrow A\ \epsilon / A\ aS}$ $\boxed{S \rightarrow \epsilon / aS}$

Q $(a+b)^*$

$S \rightarrow \epsilon / aS / bS$

terminal

a(as)
~~ab~~ aa(ϵS)
aS
a(as)
a(a(bS))

Q atleast 2 string len {a,b}

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

$S \rightarrow ABS$
$A \rightarrow BB / \epsilon$
$B \rightarrow a/b$

$S \rightarrow ABS$
$A \rightarrow BB / \epsilon$
$B \rightarrow a/b$

AB (ABS)

Page No.
Date :

$\boxed{13/2}$

Q. $L = a^{m+n} \cdot b^m \cdot c^n \quad / n, m \geq 1$

a^2bc

a^3b^2n

$S = aKPS$

$K = a.b$

$P = c$

$L = A\,B\,C$

$A = aA/aa/a$

$\cancel{A' = aA'/a}$

$B = bB/b$

$C = cC/c$

Q. Try string length atmost 2.

$\{\epsilon, a, b, aa, ab, ba, bb\}$

$(\epsilon + a + b) \cdot (\epsilon + a + b)$

$S = AB/\epsilon/BB$

$RE = \epsilon + (a+b)(a+b)$

$A \quad A$

$A = a/b$

$S \rightarrow AA$

$B = a/b$

$A \rightarrow a/b/\epsilon$

$S = \cancel{A} / AA / \epsilon$

$A = a/b$

or

$S = AA$

$A = a/b/\epsilon$

$S \to aRb$

$R \to aR / bR / \epsilon$

$a(a+b)^* b$

Page No.
Date :

Q string starts with 'a' and ends with 'b'.

$a(a+b)^* b$

not required since already part of it.

$S = aRb / \cancel{\epsilon}$

$R = \cancel{a/b/}\ aR / bR / \cancel{abR / baR} / \epsilon$

Q starts and ends with same symbol.

$\cancel{(a+b)^* (a+b)^* (a+b)}$ $a(a+b)^* a$, $b(a+b)^* b$.

$\cancel{S = RR^* R / \epsilon\ SR^* S}$ $+ a + b + \epsilon$

$\cancel{R = (a+b) / \epsilon}$

$\to aR / bR / \epsilon$

$S = (a+b)^*$

$\boxed{\begin{array}{l} \cancel{S = aRa / bRb / \epsilon} \\ R = \cancel{(a+b)R}\ aR / bR / \epsilon \end{array}}$ $/a / b.$

Q. even length string.

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

$\boxed{\begin{array}{l} S = AAS / \epsilon \\ \cancel{A = a/b} \end{array}}$

aaa aaabbbbbb

Page No.
Date :

$S \to aAb$

$A \to aAb / \epsilon$

***

Q. $L = a^n b^n$; $n \geq 1$

~~$a(ab)^*b$~~ $a(aAb)b$

$a^n b^n$

~~$L = \begin{cases} S = aRb \\ R = abR / \epsilon \end{cases}$~~

$a(aSRb)b$

~~$L = \begin{cases} R = \\ S = a/b/\epsilon \end{cases}$~~

~~$\{ S = ab S$ $aSRb$, $R = \epsilon \}$~~

~~$a(aSb)b$~~

~~$\to aaDb$~~

***

$S \to aAb$

$A \to aAb / \epsilon$

$\to$ ***

$S \to aSb / ab$

$\epsilon =$

$L = a^n b^n c^m ; n, m \geq 1$

~~$S \to aSbR / ab$~~

~~$R \to cR / \epsilon$~~

$S \to A.K$ ~~/ abc~~

$A \to aAb / ab / \epsilon$

$K \to cK / \epsilon$

Page No.
Date :

$L = \boxed{a^n b^n}\boxed{c^m}$

* concatenate

$L = A.B$
$A \rightarrow aAb / ab \rightarrow n$
$B \rightarrow cB / c \rightarrow m$

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

$L = A.B$
$A \rightarrow aAB / ab$
$B \rightarrow cBd / cd$

Q. even length palindrome $L = (w w^R) / w = (a+b)^*$

aa, bb, abba, baab,

$\cancel{((a+b)(a+b))^*}$

$S = aSa / bSb / \epsilon$

(aba)(aba)

odd length

$S = aSa / bSb / a / b$
(aba)(a/b)(aba)

any length

$S = aSa / bSb / a / b / \epsilon$

aaabbbaa

$S = bSb$
aa bbaa
bbaa

$aSa \rightarrow a(aSa)a$

Page No.
Date :

Q = $L = a^m b^m c^m d^n$ / $n, m \geq 1$.

~~S = A.B.D~~

~~A = aAb / ab~~

~~B = cB / c~~

~~D = dD / d~~

S = aSd / bSc / abcd

~~S = aSRD /~~

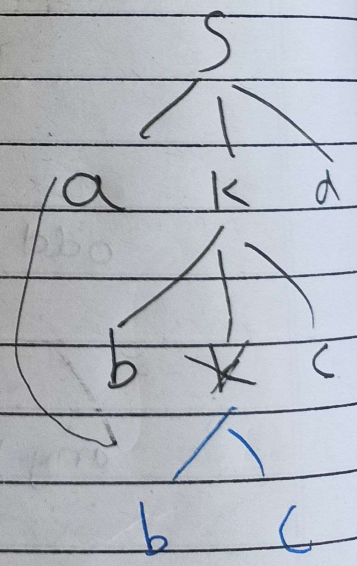
aa bbb ccc dd

R = bRc / bc.

~~S = a(SR)d / a~~

S → aSd / aKd.

K = bKc / bc



ab^2c^2d

a S d

aaSdd

aa. bKcdd

aabbKccdd

not solvable by context-free

Page No.
Date :

Q. $L = a^n b^m c^n d^m \;/\; n, m \geq 1$

$S \rightarrow$ ~~aSc / b~~ ~~aSc / bSd /~~

$S \rightarrow aSD^{\epsilon} / bSD$

Q. $L = a^n b^n c^n, \; n \geq 1$

$S \rightarrow aSc \;/\; aBc$

$B \rightarrow bB / b$

solved only by context-sensible

Classifications of Grammar :- (Chomsky)

Type 3 $\longrightarrow$ Regular Grammar
Type 2 $\longrightarrow$ Context free grammar
Type 1 $\longrightarrow$ context sensitive grammar
Type 0 $\longrightarrow$ free grammar

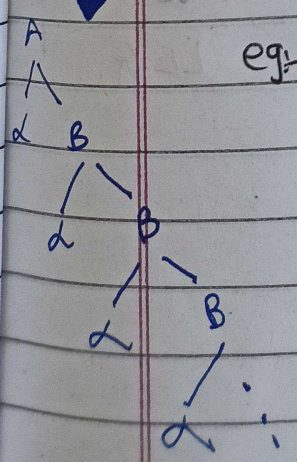
① Type 3 :-

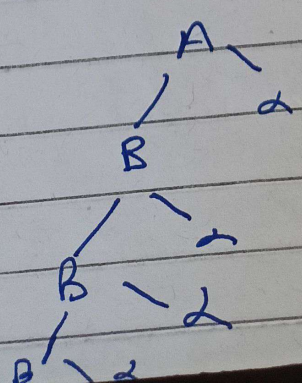
$$\boxed{A \rightarrow \alpha B \;//\; \beta}$$

right-linear

$A, B \in V$ (non-terminals)
$\alpha, B \in T$ (Terminal)

eg. $A \rightarrow aB / a$
$B \rightarrow aB / bB / a / b.$





$$\boxed{B\alpha / \beta}$$

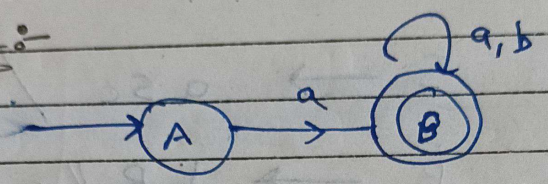
left-linear.

Both right & left linear together are not possible.

$$\left.\begin{array}{l} a \to Ba / a \\ B \to aB / a \end{array}\right\} \otimes$$

~~Regular Grammer Conversion to~~ Finite Automata :- to RGC.

Type-2 :-



$a(a+b)^*$

~~B = a/b~~

reachable to 'B' via 'a'

(RLG)

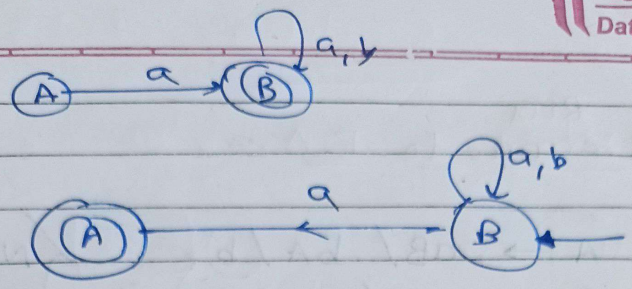
$$\boxed{\begin{array}{l} A \to aB \\ B \to aB / bB / \epsilon \end{array}}$$

Reverse grammer
change final → to non-final
change arrow dir^n
change start state

To convert to LLG.

Compliment → exchange final & non-final

Page No.
Date :



$$\left.\begin{array}{l} B \rightarrow aA \,/\, aB \,/\, bB \\ A \rightarrow \epsilon \end{array}\right\} \text{RLG}$$

$\Downarrow$ LLG

$$\boxed{\begin{array}{l} B \rightarrow Aa \,/\, Ba \,/\, Bb \\ A \rightarrow \epsilon \end{array}}$$

22-2

$L = FA$

$\Downarrow$ Reverse

L^R FA

$\downarrow$ G

RLG

L^R $\Downarrow$ Reverse

$(L^R)^R$ $\boxed{\text{LLG}}$

$L =$

RLG to LLG conversion

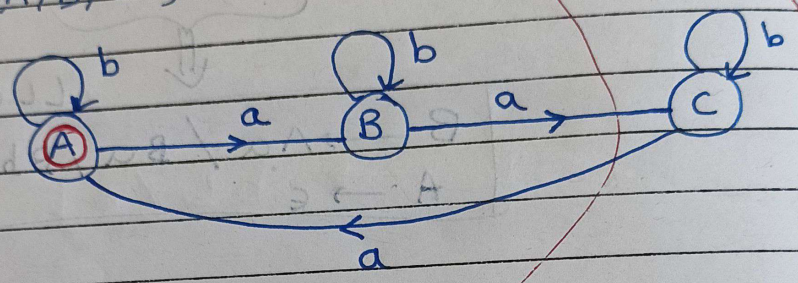
Page No.
Date :

RLG Grammar to F.A :-

$A \rightarrow aB / bA / b$
$B \rightarrow aC / bB$
$C \rightarrow aA / bC / a$

{No. of non-terminals = No. of states}

$\{A, B, C\}$



* $A \rightarrow aB / bA / \epsilon$
$B \rightarrow aC / bB$
$C \rightarrow aA / bC$

To find final state, add 'ε'.

∴ (final state)

place ε at bA = bε = b → stop

placing ε at aA = aε = a → end

⇒ ≅ 'a' mod 3 == 0.

LLG to F.A :-

L^R (1) LLG → RLG

$(FA)^R$ (2) RLG → FA

$(L^R)^R = L$ (3) FA $\xrightarrow{\text{reverse}}$ FA = L.

eg:- ends with 'a'

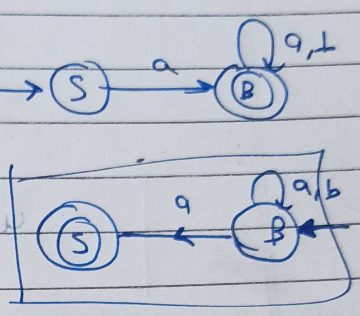
$RE = (a+b)^* a$

$S \rightarrow Aa$
$A \rightarrow Aa \,/\, Ab \,/\, \epsilon$

↓

RLG:

$S \rightarrow aA$
$A \rightarrow aA \,/\, aB \,/\, \epsilon$



Pumping Lemma:-

works on Pigeonhole Principle

L = ? How to know it is regular or not. ?

$L = a^n b^n, n \geq 1.$ or $L = a^n, n \geq 1$

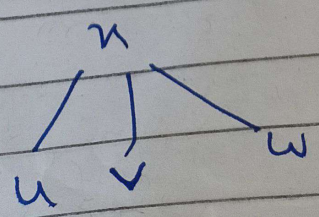
↓

→ ∞ input, finite state (each state has to accomodate atleast 1 input)

$L = a^p, b^p, p \geq 1$

① choose n length str. (aaabbb)
② Mark str = x. (x = aaabbb)
③ $|x| \geq n$.
④ divide into 3 parts (u, v, w) ~~, length of u & v < n~~

$|v| \geq 1$



Page No.
Date :

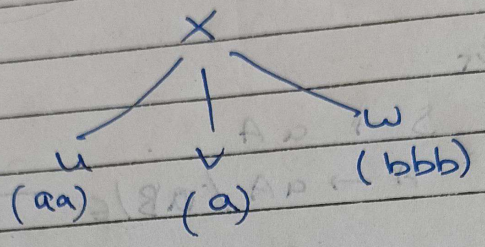
Conditions $|x| \geq n$

$uv^iw \in L$

$|v| \geq 1$

$\boxed{i \geq 0}$

$|uv| \leq n$



$u = aa, \quad v = a, \quad w = bbb$

$aa(a)^i bbb$ $\quad i=0 \Rightarrow aabbb \notin L$

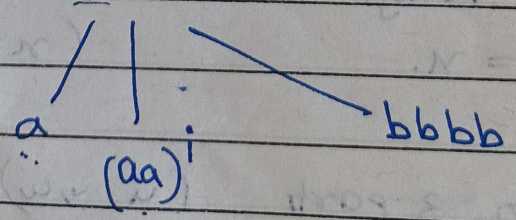
$u \quad v \quad w$

$i=2 \Rightarrow aaaabbb \notin L$

$\boxed{\therefore \text{Non-regular}}$ ← contradiction to regularity (i.e $a^p b^p$)

$L = a^n, b^m / n, m \geq 1$

$x = aaabbbb$



Page No.
Date :

~~Type-2 Grammar :~~

$\Rightarrow a^n$ / n is even

Assumption:- L is regular $\forall$ ~~$\Rightarrow$~~ $n \% 2 == 0$.

(i) $n = 6$ (aaaaaa)

(ii) $|x| = n = 6$

$x = aaaaaa$

$n = 4$

$str = a^6$

u (a) v (aa) w (aaa)

aaaaaa

u=a, v=aa, w=aaa

aa a aaa

$a(aa)^{i=1}(aaa)$ ϕ ' $\in$ L

"Regular".

$\underline{n = 4}$

aaaa

(aa) $(a)^i$ (aaa)

$i = 0$

aaaaaa

a aaaa a

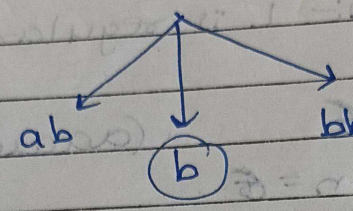
a $(aa)^i$ aa

✓

Page No.
Date :

Q. $L = ww^R \,/\, w \in (a,b)^*$

$x = abbbba \geq 3 \quad n=3$



$xv^iw \Rightarrow ab(b)^i(bba)$

$i=0 \Rightarrow abbba$

odd not working

Q. $ww^R \,/\, |w|=2,\ w \in \{a,b\}$

aa aa
ab ba
ba ab
bb bb

Q. $L = a^n b^n \,/\, n < 10^{10}$

finite $\rightarrow$ $\therefore$ regular

$L = a^{n^2} \,/\, n \geq 1$ or $L = a^n \,/\, n = \text{prime}$

non-regular

★ should be done for all possibilities of a given bifurcation.

Page No.
Date :

Type-2 Grammar :- (context-Free grammar)

$A \longrightarrow \alpha$

$\alpha \in \{ \text{Terminal} \cup \text{Non-terminal} \}^*$

eg:- $A \longrightarrow aAb \mid ab.$

$\{ A \longrightarrow aAb, \; A \longrightarrow ab \}$

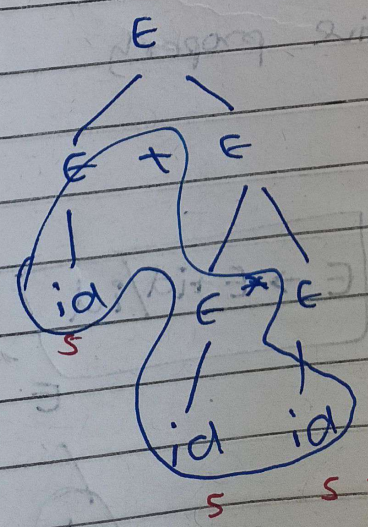
independent with prev & next

$aA \longrightarrow aa$ (context-sensitive) (type-1)

→ CFG (Context-Free)

1. maybe ambiguous
2. maybe non-deterministic
3. left recursive.

eg:- $E \longrightarrow E+E \mid E*E \mid id$



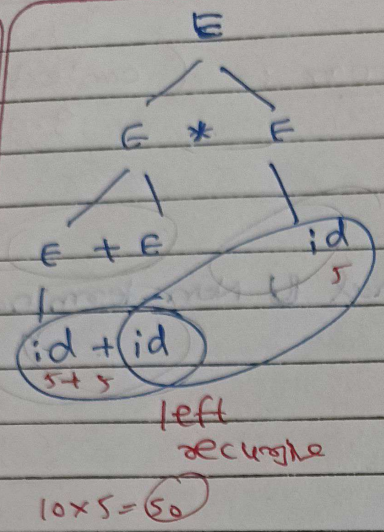
$E(id + id*id)$

right recu

$a+(b$

$5 \rightarrow 5 \times 5 + 5 = 30$

(a+b)/c



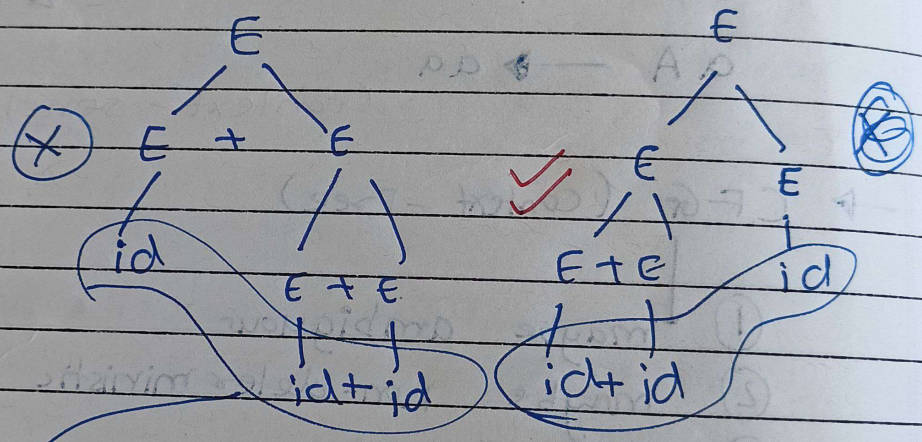
id + id*id

If 2 parse tree can be generated for same string

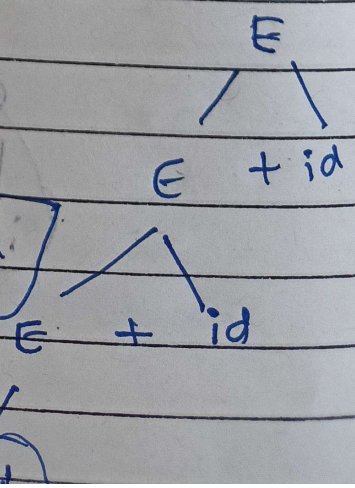
↓

⇒ Ambiguous.

eg:- id+ id+ id



{ 5+5+5 = 5+5+5 }

doesn't follow Associative property

$\boxed{E \to E+id/id}$