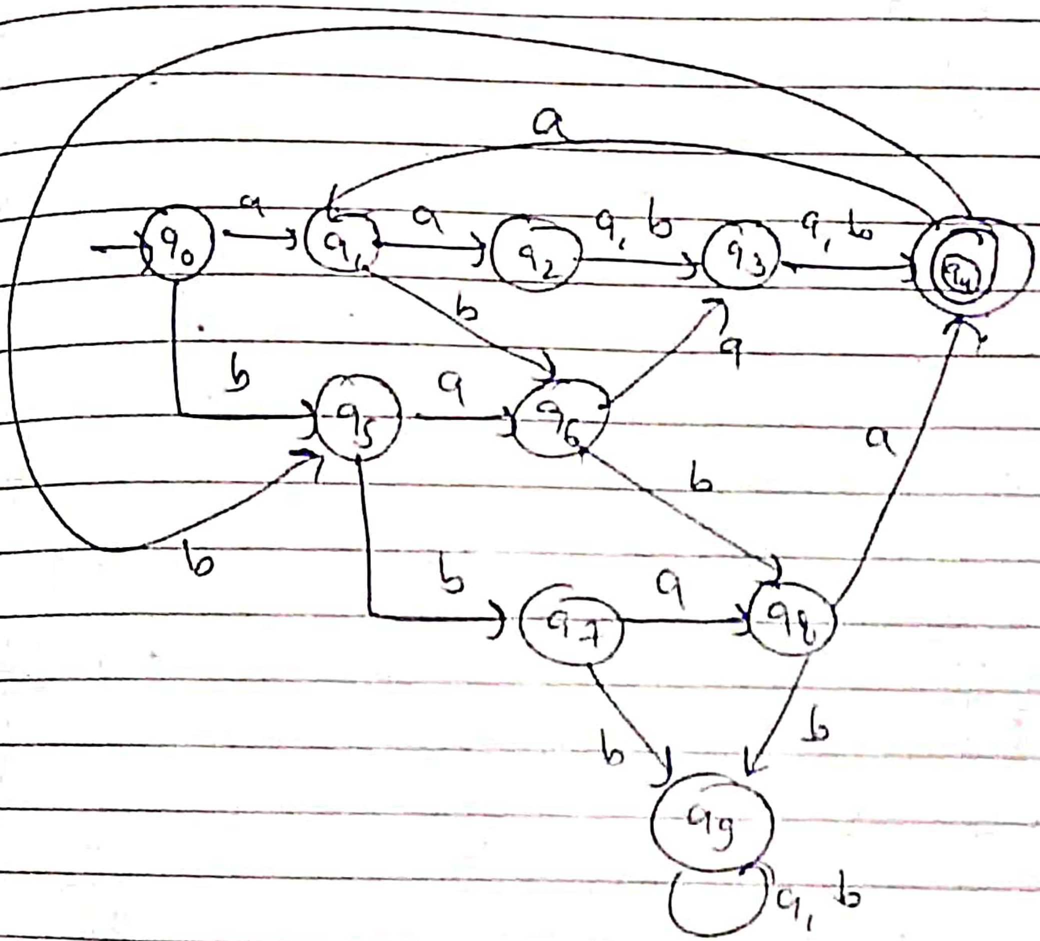
1)

1) L = {aabb, abab, abba, baab, baba, aaa, b, aaba, abaa, baaa, aaaa, ababaaaa, bababbaa .... }



Grammar

S → aA | bB

A → aC | bD

B → aF | aFS | baa | baaS

C → EE | EES

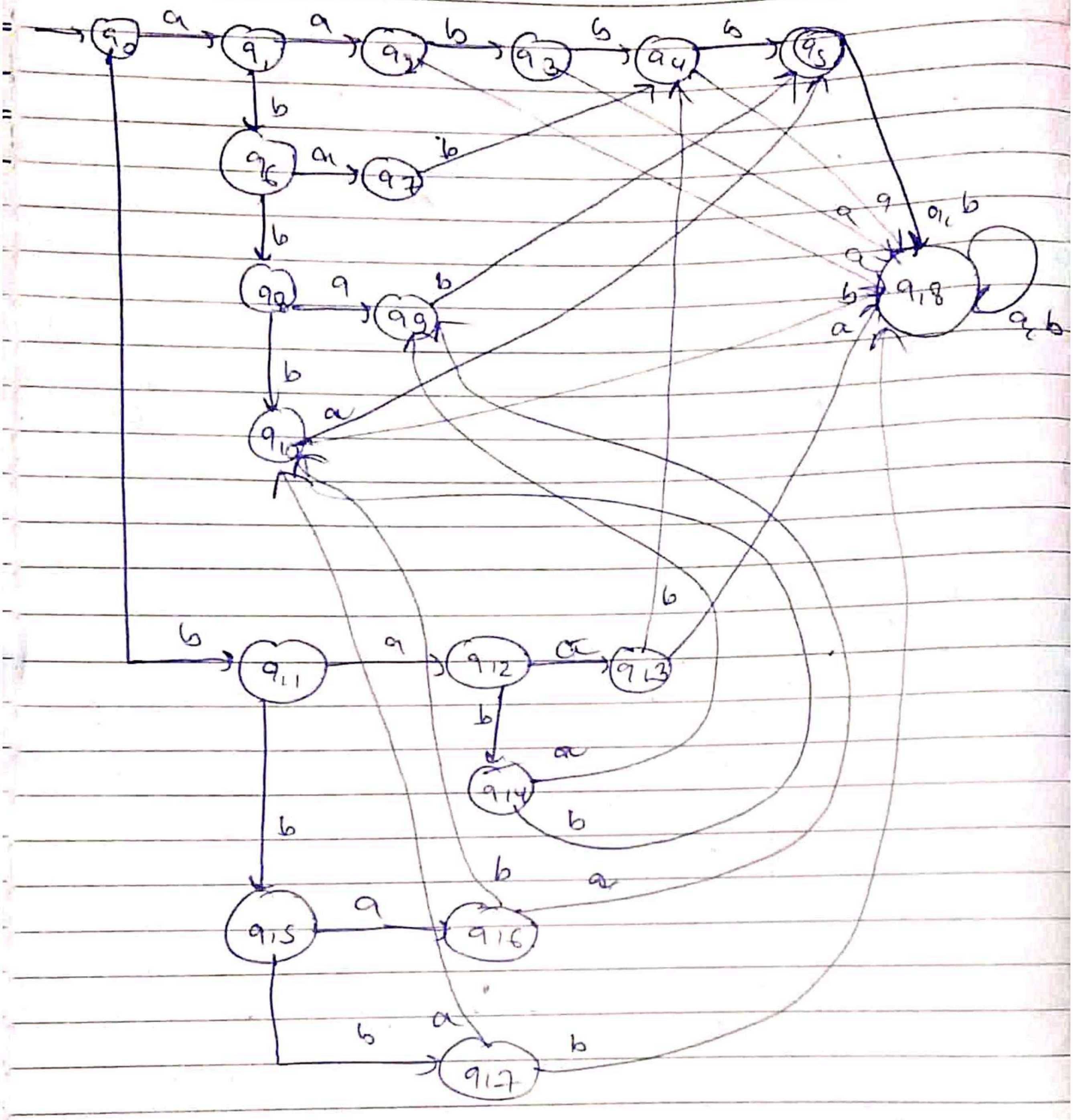
D → aE | aES

E → a | b

F → aE | ba

2)

L = {aabbb, ababb, abbba, babab, bbaab}



Grammar

S → aA | bB

A → abbb | bB

B → aE | bF

C → abb | bD

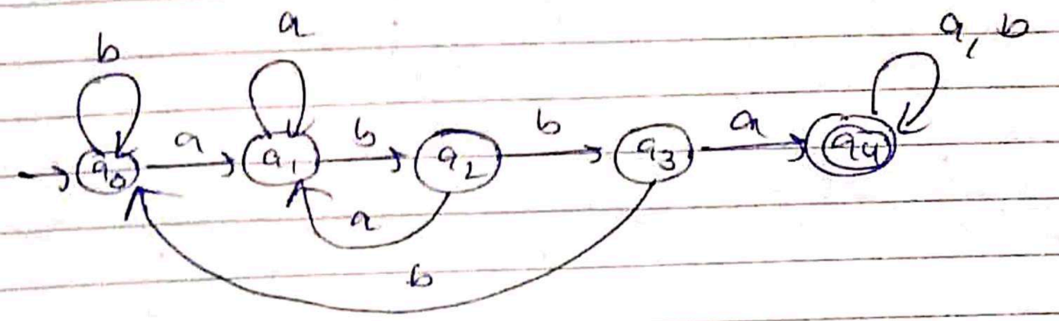
D → ab | ba

E → abb | bab

F → aD | baa

3).

$L = \{ abba, babba, abbab, aaabbab \ldots \}$

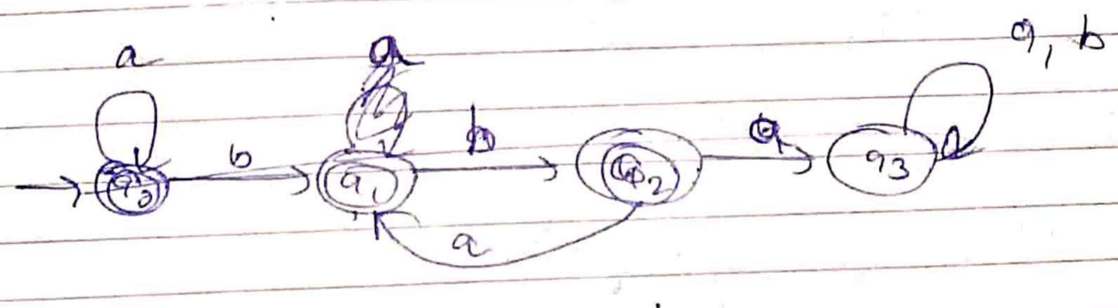


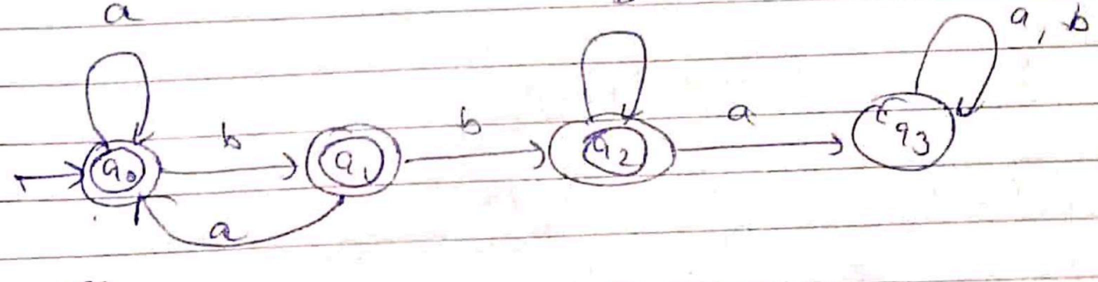
Grammer

$S \to aA \mid bB$
$B \to bB \mid aA$
$A \to aA \mid bC$
$C \to aA \mid bD$
$D \to a \mid bS$

4).

$L = \{ \lambda, a, b, ab, ba, aab, aba, bab \ldots \}$



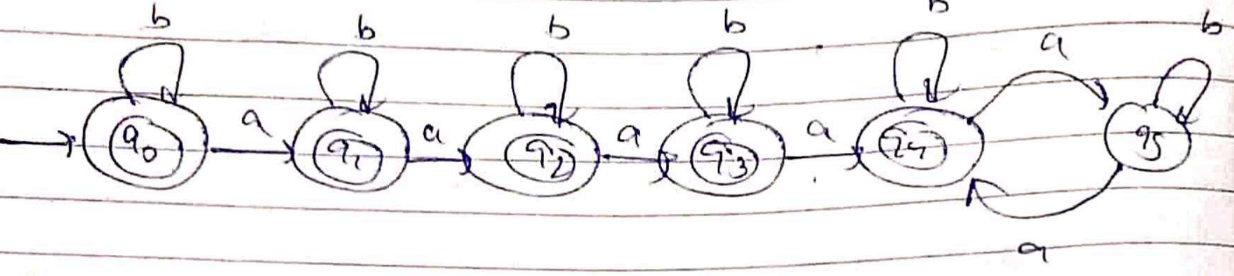


Grammar

$S \to aS \mid bA \mid \lambda$
$A \to aS \mid bB \mid \lambda$
$B \to bB \mid \lambda$

(7)

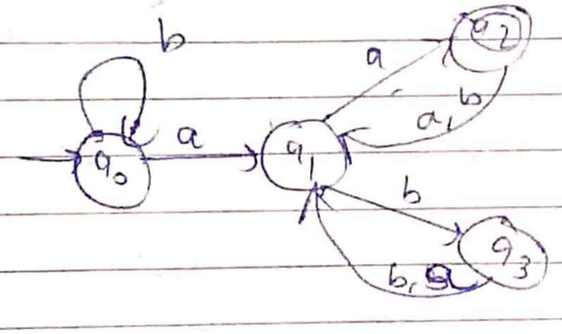
$L = \{ \lambda, a, b, ab, aab, aaab, aaaab, aaaaab, bba, bb \dots \}$



~~S → BBC~~
~~B →~~

S → AAC
A → BDBD
B → bB / ε
C → ABC / ε
D → a / ε

8). $L = \{ aa, aaba, aaaa, abba, abaa, \dots \}$



S → bS | aA

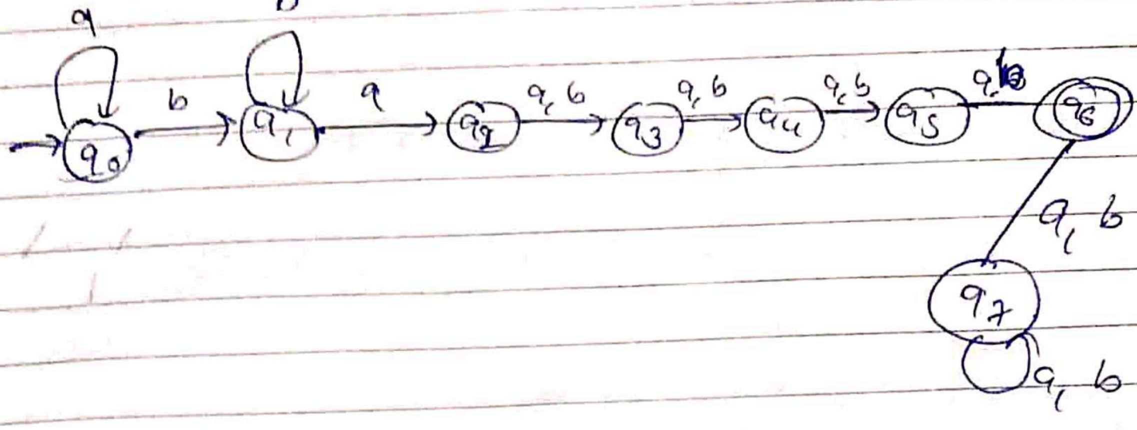
A → a | aB / bB

B → aA | bA

PAGE:
DATE: / /

Q1.

$L = \{aaaaa, abbbb, aabba, aabbb, \ldots\}$



Grammar

$S \to aS \mid bA$
$A \to bA \mid aBBBB$
$B \to a \mid b$

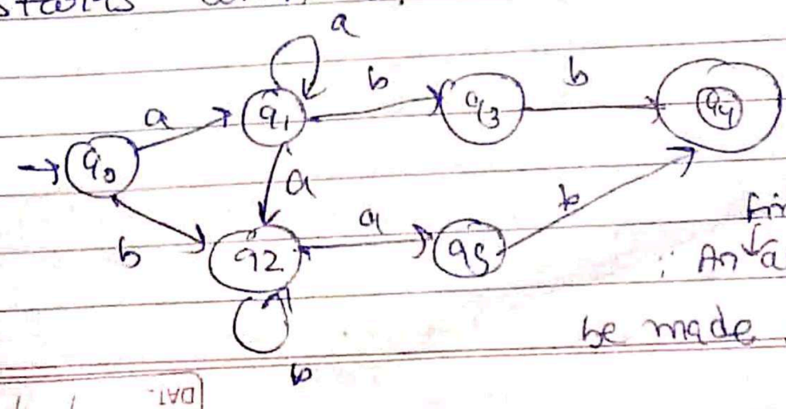
Q2)

(i) $S \to aA \mid bB$
$A \to aA \mid aB \mid bb$
$B \to bB \mid ab$

i). $L = \{abb, bab, aabb, bbab, aaab, bbbab, aaabb, aaaaaab, \ldots\}$

Either ending with ab ~~or~~ bb if starts with a and ending with bb only when starts with b.

ii).



∴ An finite automata can be made ∴ Regular (iii).

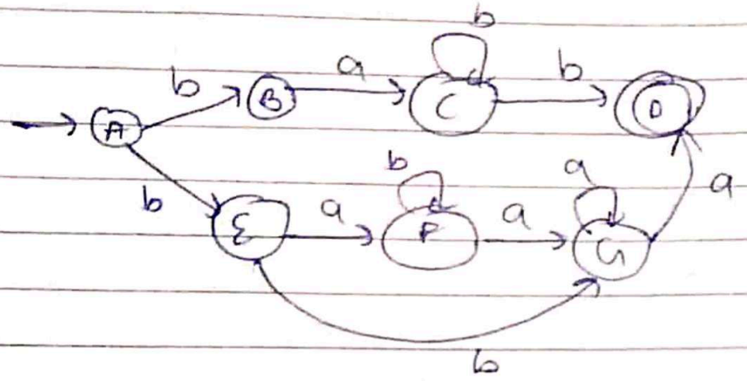
$aa^*(bb + ab^*ab) + bb^*ab$

(iv).

step 1). Reverse regular expression

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

step 2). Construct finite automata



step 3). obtain right linear grammar

$A \rightarrow bB \mid bE$
$B \rightarrow aC$
$C \rightarrow bC \mid b$
$E \rightarrow aF \mid bG$
$F \rightarrow bF \mid aG$
$G \rightarrow a$

step 4). Reverse right hand side of every product

$A \rightarrow Bb \mid Eb$
$B \rightarrow Ca$
$C \rightarrow Cb \mid b$
$E \rightarrow Fa \mid Gb$
$F \rightarrow Fb \mid Ga$
$G \rightarrow a$

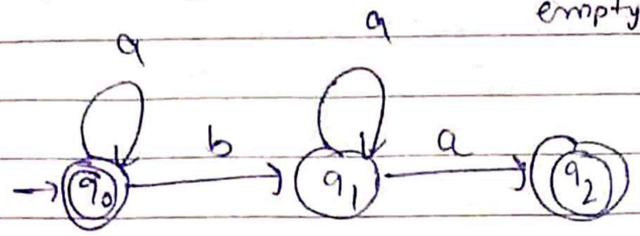
(7) $S \rightarrow aS \mid bA \mid \lambda$

$A \rightarrow aA \mid a$

i) $L = \{\lambda, a, aa, aaa, aaaa, aba, abaa, abaaa,$
$ba, baa, baaa, baaaa \ldots\}$

Either, string of a's or, empty or substring ba should be there

ii)

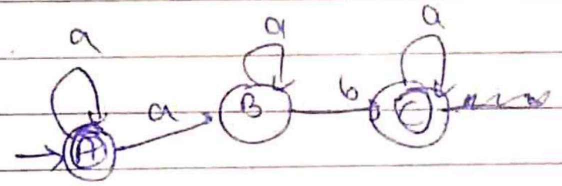


(iii) $\cancel{a^* + b\, a\, a}$ $\quad a^* + a^* b a^* a$

(iv) step 1) Reverse regular expression

$a a^* b a^* + a^*$

Step 2) Obtain finite automation



Step 3) obtain right linear grammar

$A \rightarrow aA \mid aB$
$B \rightarrow aB \mid bC$
$C \rightarrow aC \mid \lambda$

Step 4) Reverse right hand side of every production

$A \rightarrow Aa \mid Ba$
$B \rightarrow Ba \mid Cb$
$C \rightarrow Ca \mid \lambda$

Q3).

1). $(aa+b)^* + (bb+a)^*$

2). $b^* a b^* a b^* a b^*$

Q3).

Regular languages are closed under intersection, i.e if L_1 and L_2 are regular then $L_1 \cap L_2$ are also regular.

proof :-

observe that $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$.
Since regular languages are closed under union and complementation, we have

- $\overline{L_1}$ and $\overline{L_2}$ are regular.
- $\overline{L_1} \cup \overline{L_2}$ is regular.
- Hence $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ is regular

4 Let $w = s_1 s_2 \ldots s_n$ be a word over Σ. Then w^R denotes the word $s_n \ldots s_2 s_1$: the reverse of w. $\lambda^R = \lambda$.

Let L be a language. Then L^R denotes $L^R = \{w^R : w \in L\}$; called the reverse of L.

The family of regular languages is closed under reversal

proof :

- Let L be a regular, and $M = (Q, \Sigma, \delta, q_0, \{q_f\})$ be an NFA that accepts it.

- We construct $M^R = (Q, \Sigma, \delta^R, q_f, \{q_0\})$, when δ^R is δ with its orientation of arcs reversed.

- There is a path from q_0 to q_f in M if and only if there is a path from q_f to q_0 in M^R: $L(M^R) = L^R$

Q7).

Applications

- for designing of lexical analyser of computer
- for recognizing the pattern using regular expressions.
- For the designing of combination and sequential circuits using mealy and moore machines.
- Used in text editors
- Used in the implementation of spell checkers.
- Regular expressions are useful in a wide variety of text processing tasks, and more generally string processing, where the data need not be textual
- Common application of regular expressions includes data validation, data scrapping parsing, syntax highlighting system.

Q 8)

1).

$L = \{ w.w \; ; \; w \in \{a, b\}^+ \}$

$\Rightarrow L = \{aa, bb, abab, aabaab, abaaba,$
$baaba baaba \ldots \}$

let, $z = \underbrace{baa}_{u}\underbrace{baba}_{v}\underbrace{ba}_{w}$ baaba

uv^iw, when $i = 0$

then, $baaaba$

this language is not satisfying the given expression, when we are changing i $\therefore$ This language is not regular.

2). $L = \{a^n b^m : n < m\}$

$L = \{ abb, abbb, aabbb, aaabbbbb, \ldots \}$

$z \in L$

$z = a^{k-1}b^k$

$a^{k-1}b^k$

a^{k-2} a^k b^k

for $i = 1$ $uv^iw \rightarrow a^{k-1}b^k$

for $i = 2$ $uv^2w \rightarrow a^{k-2}a^{k+2}b^k \rightarrow \underbrace{a^k b^k}$

this violates the expression given.

$\therefore$ Not regular

3) $\{ww^R : w \in (a,b)^*\}$

$L = \{aa, bb, abba, aabbaa, abaaba, baaaab \ldots\}$

let, $z = \underbrace{abb}_{u}\underbrace{abb}_{v}\underbrace{abba}_{w}$

for $i=1$ $uv^1w =$ abbabba bba

for $i=2$ $uv^2w =$ abb abba abba bba

This language also not satisfying the given expression, when we are changing values of

∴ This language is not regular

4) $L = \{a^n b^m c^{n+m} ; n>=1, m>=1\}$

$L = \{abcc, aabccc, abbccc, aabbcccc, aaabcccc, abbbcccc, aabbbccccc \ldots\}$

$z \in L$

$z = a^k b^k c^{2k}$

$$a^k b^k c^{2k} \rightarrow \underbrace{a^k\, b^{k-1}}_{u}\ \underbrace{b^k}_{v}\ \underbrace{c^{2k}}_{w}$$

for $i=1 \rightarrow uv^1w$

$\hookrightarrow a^k b^{k-1} b^1 c^{2k}$

$\hookrightarrow a^k b^k c^{2k}$

for $i=2 \rightarrow uv^2w \rightarrow a^k b^{k-1} b^2 c^{2k}$

$a^k b^{k+1} c^{2k}$

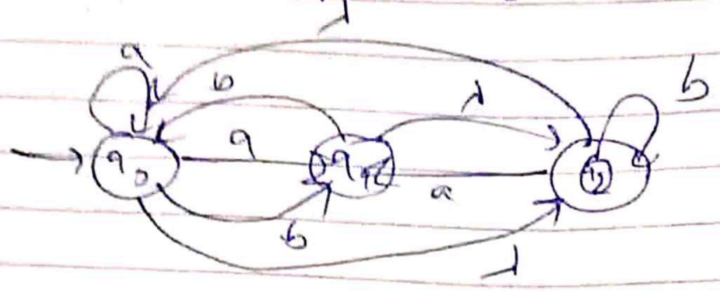
This also violates the expression given in question. ∴ This is also not regular

PAGE
DATE / /

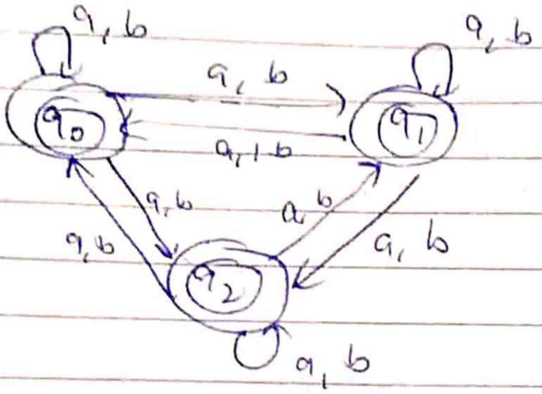
Q 5).

a)

	a	b	λ
→ q_0	q_0, q_1	q_1	q_3
q_1	–	q_0	q_2
final q_2	q_1	q_2	q_0

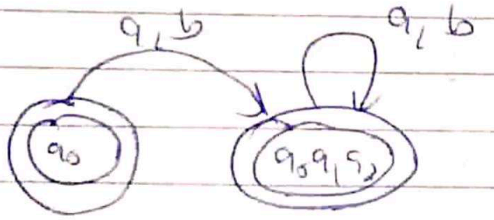


Equivalent NFA will be



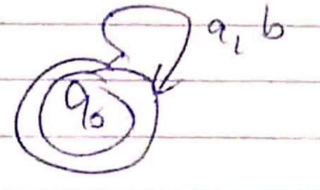
	a	b
→ q_0	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_1	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
q_2	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

DFA conversion

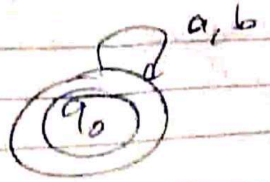


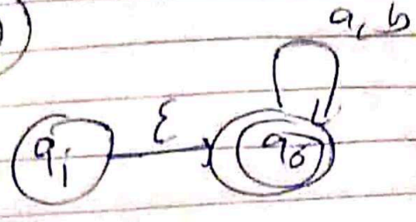
	a	b
q_0	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

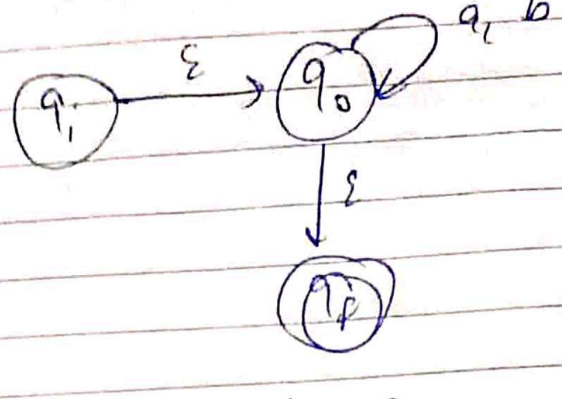
Minimal DFA



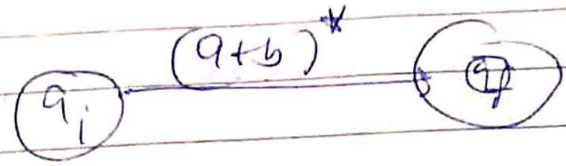
Regular expression by state elimination method

Given 

Rule 1). 



Now, elimination q_0



$\therefore$ RE :> $(a+b)^*$

Regular expression by Arden's method

$q_0 = \lambda + q_0 a + q_0 b$

$q_0 = \lambda + (a+b) q_0$

$\therefore \quad R = Q + RP$

$R = QP^*$

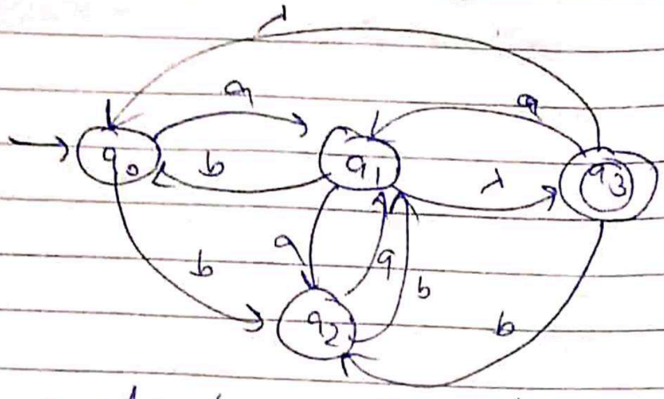
Here $Q = \lambda$, $P = (a+b)$

$\therefore \quad R = \lambda (a+b)^*$

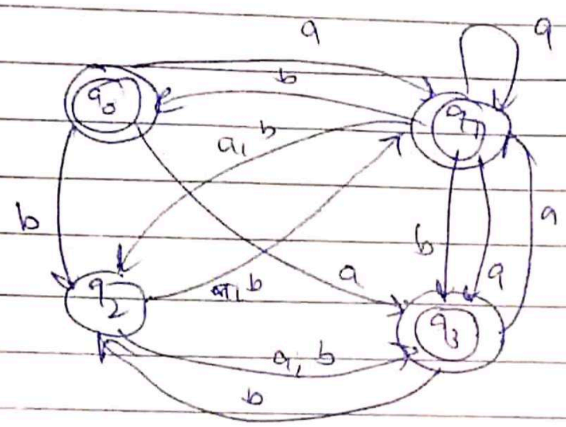
$\therefore$ RE $= (a+b)^*$

b)

	a	b	λ
→ q_0	q_1	q_2	q_3
q_1	q_2	q_0	q_3
q_2	q_1	q_1	—
q_3	q_1	q_2	—



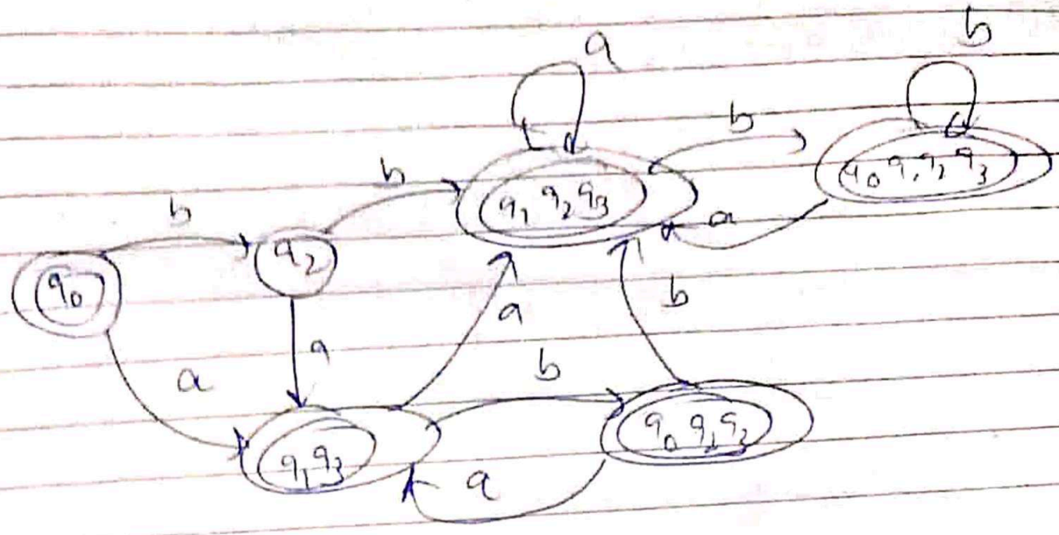
Equivalent NFA is



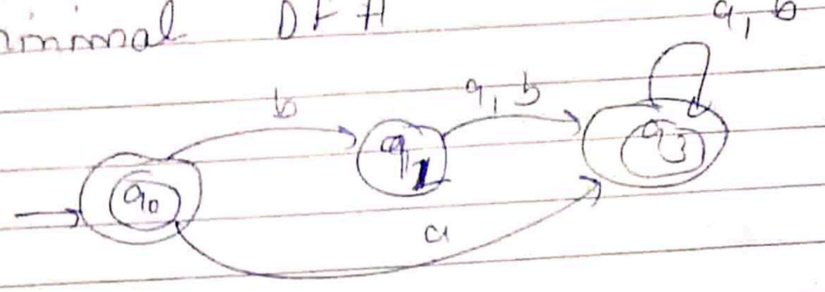
	a	b
q_0	$\{q_1, q_3\}$	q_2
q_1	$\{q_1, q_2, q_3\}$	$\{q_0\ q_2\ q_3\}$
q_2	$\{q_1, q_3\}$	$\{q_1\ q_3\}$
q_3	$\{q_1, q_3\}$	$\{q_2\}$

DFA conversion

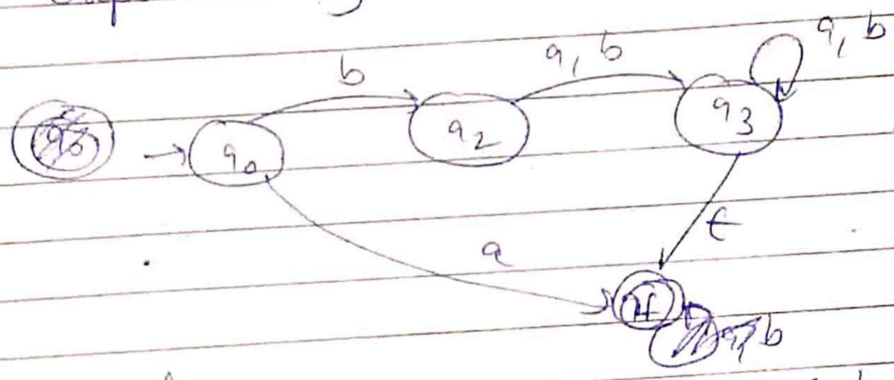
	a	b
q_0	$\{q_1, q_3\}$	$\{q_2\}$
q_2	$\{q_1\ q_3\}$	$\{q_1\ q_2\ q_3\}$
$q_1\ q_3$	$\{q_1, q_2\ q_3\}$	$\{q_0\ q_2, q_3\}$
$q_1 q_2 q_3$	$\{q_1\ q_2\ q_3\}$	$\{q_0\ q_1\ q_2\ q_3\}$
$q_0 q_2\ q_3$	$\{q_1\ q_3\}$	$\{q_1\ q_2\ q_3\}$
$q_0 q_1 q_2 q_3$	$\{q_1\ q_2\ q_3\}$	$\{q_0\ q_1\ q_2\ q_3\}$



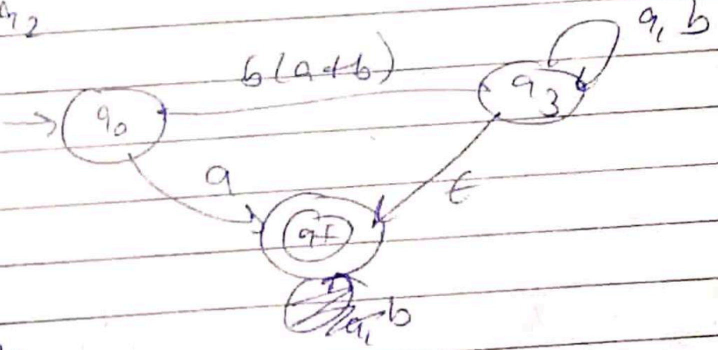
Minimal DFA



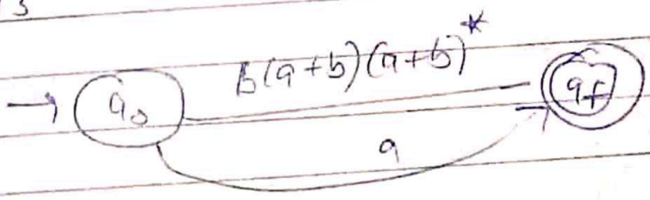
Regular expression by state elimination method



Eliminating q_2



Eliminating q_3



Eliminating q_0

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

Regular expression by Arden's method

$q_0 = \lambda$

$q_1 = q_0 b$

$q_3 = q_0 a + q_3 a + q_3 b + q_1 a + q_1 b$

$q_3 = a + (a+b) b + (a+b) q_3$

$R = Q + RP$

$R = QP^*$

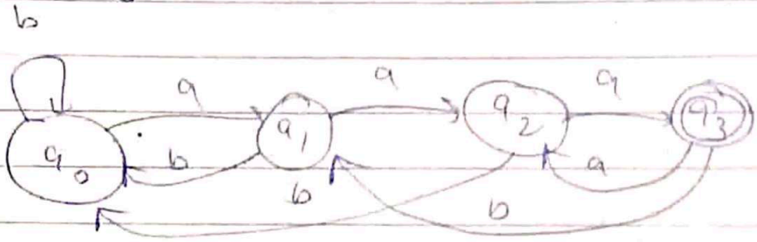
$Q = a + (a+b) b$

$P \to a+b$

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

c).

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

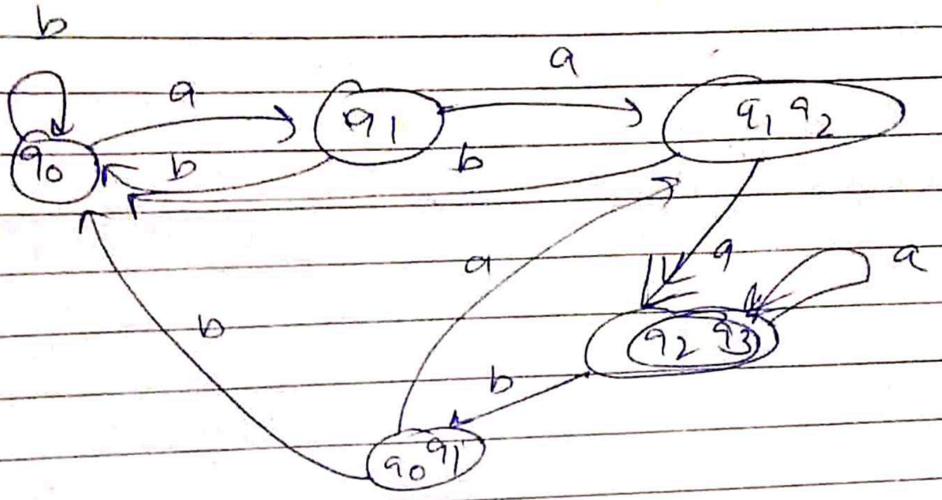


	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_2	q_1

DFA conversion

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

PAGE:
DATE:



Minimal DFA