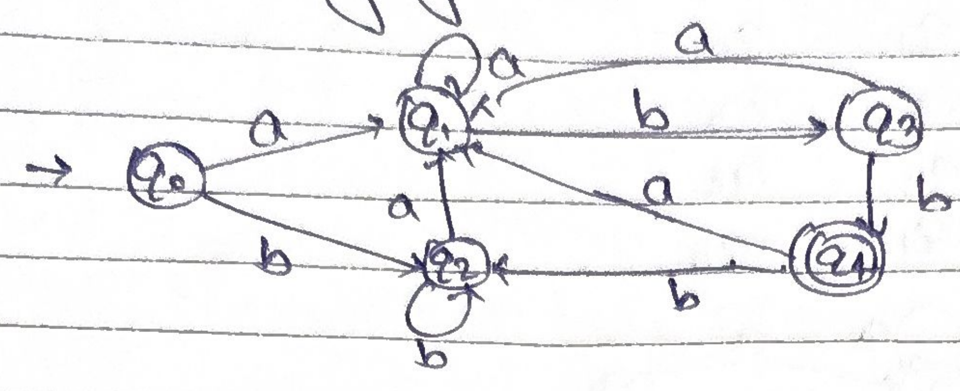
DATE
PAGE

Assignment - 2

Name :- Sachin Kumar
Roll :- 230102036

Q1) Minimization of DFA :-



State	Input 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

→ Non-final State :-

State	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

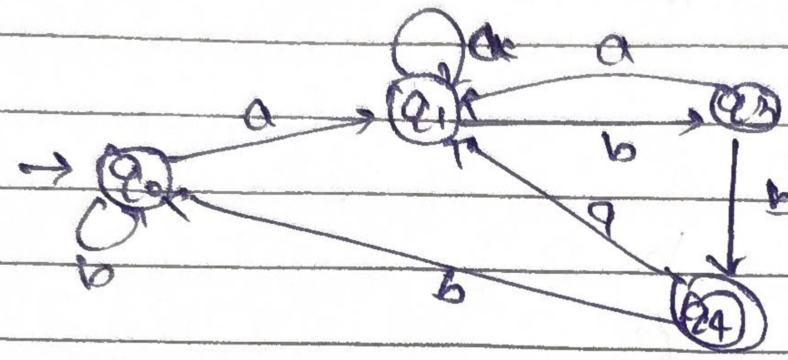
→ Here q_0 & q_2 have same outgoing So, we remove one of them.
→ q_0 is initial state so we Can't remove it.
→ there we remove q_2 & q_0 in place of q_2.

→ final State

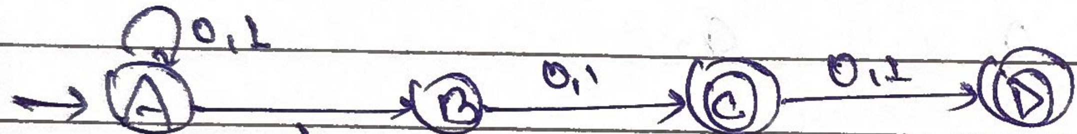
State	a	b
q_4	q_1	q_2

Combined state :-

State	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_4
q_4	q_1	q_0



Q2) NFA to RE (Regular Expression)



$A = A.0 + A.1 + \varepsilon$

$B = A.1$

$C = B.0 + B.1$

$D = C.0 + C.1$

$\underset{R}{A} = \underset{R}{A}(\underset{P}{0+1}) + \underset{Q}{\varepsilon}$

According to Arden's theorem.

$R = QP^*$

$R = \varepsilon(0+1)^*$ $\quad A = (0+1)^*$

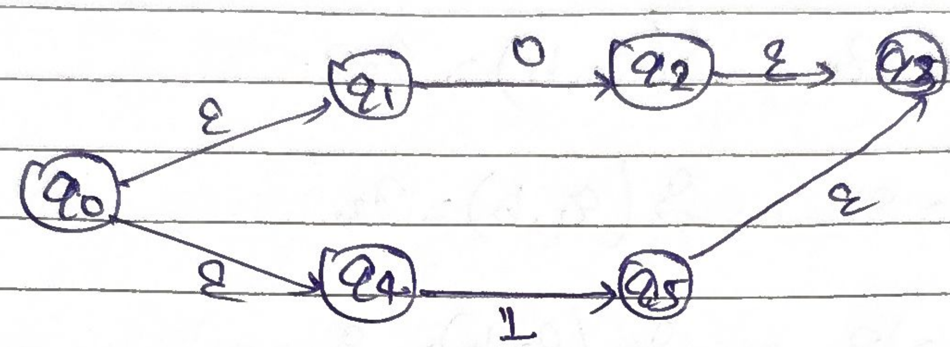
DATE
PAGE

$B = A.1$
$= (0+1)^{*} . 1$
$= (0+1)^{*} . 1$

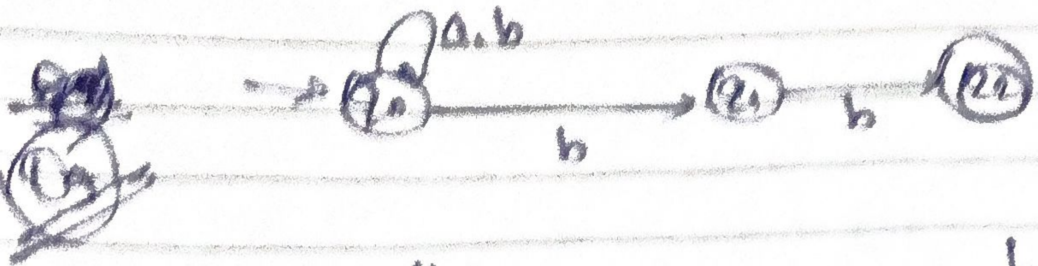
$C = B(0+1)$
$= (0+1)^{*} . 1 (0+1)$

$D = C(0+1)$
$= (0+1)^{*} . 1 (0+1) (0+1)$

Q3)



Convert $(0+1)^{*} 1 (0+1)$



state	a	b
q_0	q_0	q_0q_1
q_1	—	q_2
q_2	—	—

$\delta(q_0,a) = q_0$, $\delta(q_0,b) = q_0q_1$

$\delta(q_1,a) = \epsilon$, $\delta(q_1,b) = q_2$

$\delta(q_2,a) = \epsilon$, $\delta(q_2,b) = \epsilon$

$\delta([q_0,q_1],a) = \delta(q_0,a) \cup \delta(q_1,a)$

$= q_0 \cup \epsilon = q_0$

$\delta([q_0,q_1],b) = \delta(q_0,b) \cup \delta(q_1,b)$

$= q_0q_1 \cup q_2$

$= q_0q_1q_2$

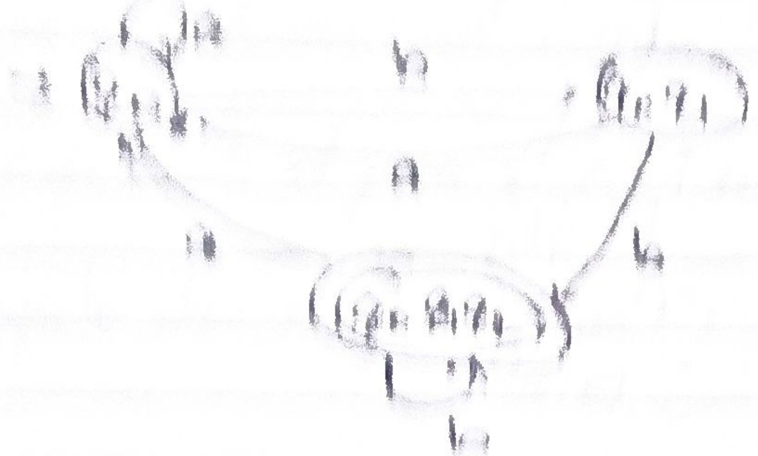
$\delta([q_0,q_1q_2],a) = \delta'((q_0q_1),a) \cup \delta(q_2,a)$

$= q_0 \cup \epsilon = q_0$

$\delta([q_0q_1q_2],b) = \delta'([q_0,q_1],b) \cup \delta(q_2,b)$

$= q_0q_1q_2 \cup \epsilon = q_0q_1q_2$

[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]



Q4). Regular language to Finite Automata.

R.E = a(b+a)* b

DATE
PAGE

Q5) DFA | NFA

→ A finite automata having a fixed number of states and each input symbol uniquely determine the next state.

→ It is finite Automata in which there exist many path from specific input from the current state to next state.

→ DFA is Hard to construct.

→ NFA is easy to construct compare to DFA.

→ Backtracking is Possible in DFA.

→ Backtracking is not always Possible.

→ It requires more space

→ it require more space and permit empty string transition

→ DFA transition function is $\delta: Q \times \Sigma = Q$

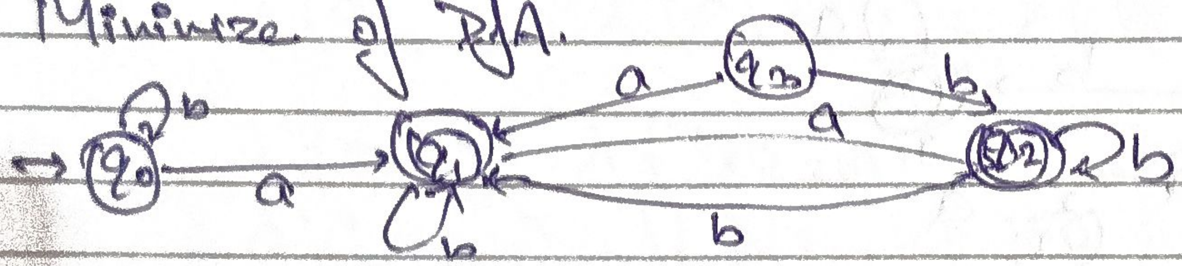
→ Transition function $\delta: Q \times \Sigma = 2Q$.

Q6. → $\Sigma = \{a, b, c\}$

→ Determine the regular expression for all strings containing exactly 'a' over Σ.

DATE
PAGE

(Q7.) Minimize of DFA.



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

→ q_3 → is unreachable state

Non-final state :-

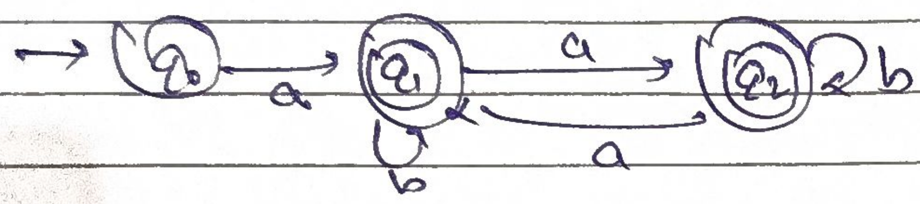
State	a	b
q_0	q_1	q_0

Final State:-

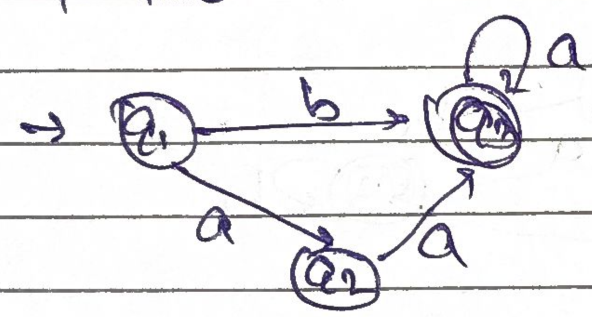
State	a	b
q_1	q_2	q_1
q_2	q_1	q_2

Combined State:-

State	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_1	q_2



Q8) DFA to RE



$q_1 = \varepsilon$

$q_2 = q_1 \cdot b + q_2 \cdot a + q_3 \cdot a$

$q_3 = q_1 \cdot b + q_1 \cdot a \cdot a + q_3 \cdot a$

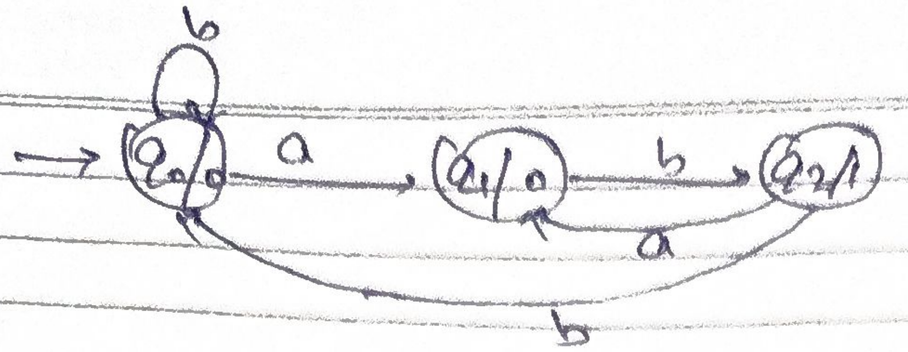
$q_3 = q_1 (b + aa) + q_3 \cdot a$

R Q R P

$R = QP^*$

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

Q10).



State	a	b	Output
q_0	q_1	q_0	0
q_1	q_1	q_2	0
q_2	q_1	q_0	1

~~State~~ ~~input~~

$$\lambda'(q_0,a) = \lambda(\delta(q_0,a))$$
$$= \lambda(q_1)$$

State = q_1
output = 0

$\lambda'(q_0,b) = \lambda(\delta(q_0,b)) = \lambda(q_0) = 0$ $\delta = q_0$, output = 0.

$\lambda'(q_1,a) = \lambda(\delta(q_1,a)) = \lambda(q_1) = 0$ $\delta = q_1$ output = 0

$\lambda'(q_1,b) = \lambda(\delta(q_1,b)) = \lambda(q_2) = 1$ $\delta = q_2$ $O = 1$

$\lambda'(q_2,a) = \lambda(\delta(q_2,a)) = \lambda(q_1) = 0$ $\delta = q_1$ $O = 0$

$\lambda'(q_2,b) = \lambda(\delta(q_2,b)) = \lambda(q_0) = 0$ $\delta = q_0$ $O = 0$

Q/Σ	Input a		Input b	
	State	Output	State	Output
q_0	q_1	0	q_0	0
q_1	q_1	0	q_2	1
q_2	q_1	0	q_0	0