8/9/

Construct DFA Equivalent to the given NFA

	0	1
p	{p, q}	p
q	r	r
r	s	-
s	s	s

$\delta'(p, 0) = \{p, q\}$

$\delta'(p, 1) = \{p\}$

$\delta'([p, q], 0) = \delta'(p, 0) \cup \delta'(q, 0)$

$\{p, q\} \cup \{r\} \Rightarrow \{p, q, r\}$

$\delta'([p, q], 1) = \delta'(p, 1) \cup \delta'(q, 1)$

$\{p\} \cup \{r\} \Rightarrow \{p, r\}$

$\rightarrow F$

$\delta'([p, q, r], 0) = \delta'(p, 0) \cup \delta'(q, 0) \cup \delta'(r, 0)$

$= \{p, q, r, s\}$

$\delta'([p, q, r], 1) = \delta'(p, 1) \cup \delta'(q, 1) \cup \delta'(r, 1)$

$= \{p\} \cup \{r\} \cup \{\emptyset\}$

$= \{p, r\}$

$\rightarrow F$

$\delta'([p, r], 0) = \delta'(p, 0) \cup \delta'(r, 0)$

$= \{p, q\} \cup \{s\}$

$\Rightarrow \{p, q, s\}$

$\delta'([p, r], 1) = \delta'(p, 1) \cup \delta'(r, 1)$

$= \{p, q\}$

$\delta'([p,q,r,s],0) = \delta'(p,0) \cup \delta'(q,0) \cup \delta'(r,0) \cup \delta'(s,0)$
$= \{p,q,r,s\}$

$\delta'([p,q,r,s],1) = \delta'(p,1) \cup \delta'(q,1) \cup \delta'(r,1) \cup \delta'(s,1)$
$= \{p,r,s\}$

$\delta'([p,q,s],0) = \delta'(p,0) \cup \delta'(q,0) \cup \delta'(s,0)$
$= \{p,q,r,s\}$

$\delta'([p,q,s]^{F},1) = \delta'(p,1) \cup \delta'(q,1) \cup \delta'(s,1)$
$= \{p,r,s\}$

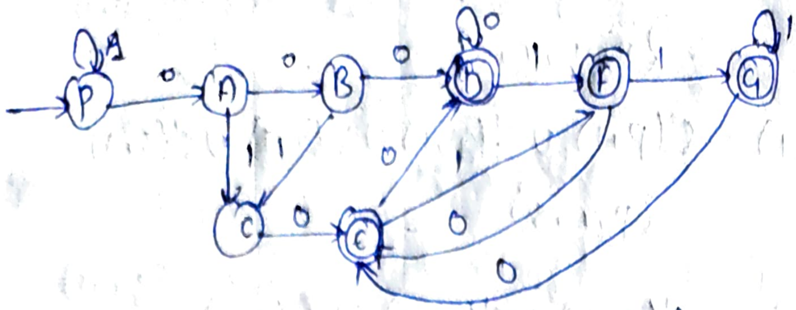
$\delta'([p,r,s],0) = \delta'(p,0) \cup \delta'(q,0) \cup \delta'(s,0)$
$= \{p,q,r,s\}$

$\delta'([p,r,s]^{F},1) = \delta'(p,1) \cup \delta'(r,1) \cup \delta'(s,1)$
$= \{p,s\}$

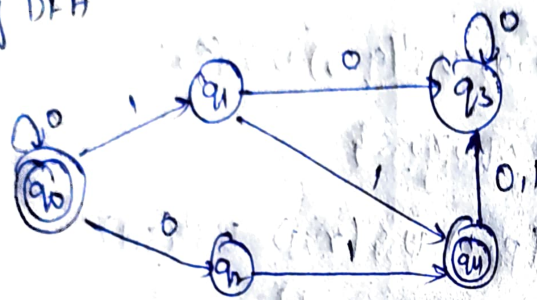
$\delta'([p,s],0) = \delta'(p,0) \cup \delta'(s,0)$
$= \{p,q,s\}$

$\delta'([p,s],1] = \delta'(p,1) \cup \delta'(s,1)$
$= \{p,s\}$

states \ i/p	0	1
p	{p,q} A	p
{p,q} A	{p,q,r} B	{p,r} C
{p,q,r} B	{p,q,r,s} D	{p,r} C
{p,r} C	{p,q,s} E	{p,s} F
{p,q,r,s} D	{p,q,r,s} D	{p,r,s} G
{p,q,s} G	{p,q,r,s} D	{p,r,s} G
{p,r,s} G	{p,q,r,s} D	{p,s} F
{p,s} F	{p,q,s} E	{p,s} F



5) Convert the follow into DFA & recognize the lang generated by DFA



	0	1
q_0	$\{q_0, q_2\}$	q_1
q_1	$\{q_3\}$	$\{q_4\}$
q_2	$\emptyset$	$\{q_4\}$
q_3	$\{q_3\}$	$\emptyset$
q_4	$\{q_3\}$	$\{q_3\}$

$\delta'(q_0, 0) = \{q_0, q_2\}$

$\delta'(q_0, 1) = \{q_1\}$

$\delta'\{q_1, 0\} = q_3$

$\delta'\{q_1, 1\} = q_4$

$\delta'([q_0, q_2], 0) = \delta'(q_0, 0) \cup \delta'(q_2, 0)$

$= \{q_0, q_2\}$

$\delta'([q_0, q_2], 1) = \delta'(q_0, 1) \cup \delta'(q_2, 1)$ $\rightarrow F$

$= \{q_1, q_4\}$

$\delta'([q_1, q_4], 0) = \delta'(q_1, 0) \cup \delta'(q_4, 0)$

$= \{q_3\}$

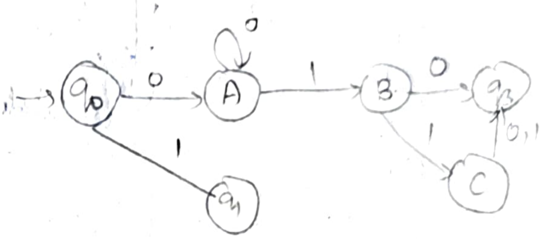
$\delta'([q_1, q_4], 1) = \delta'(q_1, 1) \cup \delta'(q_4, 1)$ $\rightarrow F$

$= \{q_4, q_5\}$

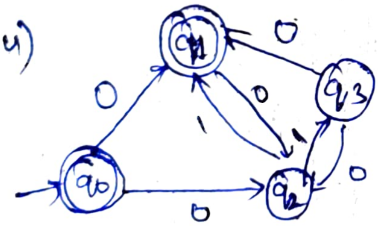
$\delta'([q_4, q_5], 0) = \{q_3\}$

$\delta'([q_4, q_5], 1) = \{q_3\}$

states \ i/p	0	1
q_0	$\{q_0, q_2\}$ A	$\{q_1\}$
$\{q_0, q_2\}$ A	$\{q_0, q_2\}$ A	$\{q_1, q_4\}$ B
$\{q_1, q_4\}$ B	q_3	q_4, q_5 C
$\{q_4, q_5\}$ C	q_3	q_3



4)



states \ i/p	0	1
q_0	$\{q_1, q_2\}$	$\emptyset$
q_1	$\{q_1, q_2\}$	$\emptyset$
q_2	$\emptyset$	$\{q_3, q_1\}$
q_3	$\{q_1, q_2\}$	$\emptyset$

$\delta'([q_0], 0) = \{q_1, q_2\} \rightarrow F$

$\delta'(q_0, 1) = \emptyset$

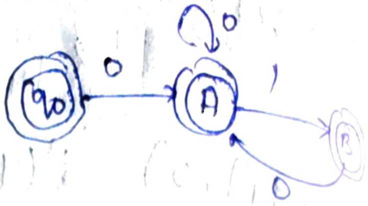
$\delta'([q_1, q_2], 0) = \{q_1, q_2\}$

$\delta'([q_1, q_2], 1) = \{q_1, q_3\}$

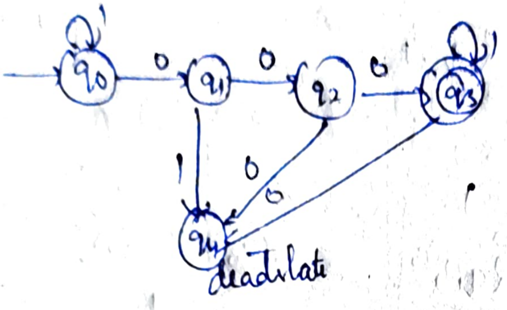
$\delta'([q_1, q_3], 0) = \{q_1, q_2\}$

$\delta'([q_1, q_3], 1) = \emptyset$

state\i/p	0	1
q_0	$\{q_1, q_2\}$	-
$\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_1, q_3\}$
$\{q_1, q_3\}$	$\{q_1, q_2\}$	-



6) Design DFA with $\Sigma = \{0,1\}$ accepts the set of all strings with 3 consecutive zero's

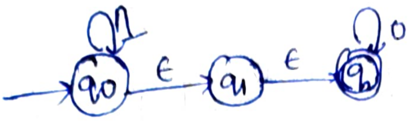


7) Design DFA such that the string doesn't contain 3 consecutive 1's

★ NFA with ε-transitions.

The ε-transitions in NFA are given in order to move from 1 state to another state without having any symbol from input set Σ.

Consider the NFA with ε as



★ Significance of NFA with ε :

The ε transitions are used to change from one state to another state. sometimes to reach to final state we do not get proper state for start state in such a case we simple want to reach to certain state which leads to final state. such a transition to that specific state should be without any input symbol Therefore we require some ε moves by which a proper state can be obtained for reaching to final state. Thus ~~on~~ ε moves plays an important role in NFA.

★ Definition of ε-closure:

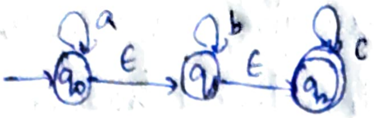
The ε-closure of P is a set of all states which are reachable from state P on ε transitions such that

→ ε-closure(p) = P when $p \in Q$.

→ If there exist ε-closure(p) = {q}, & δ(q, ε) = r then

ε-closure(P) = {q, r}

Ex: Find ε-closure for the following NFA with



$\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$

$\epsilon\text{-closure}(q_2) = \{q_2\}$

Conventioning its equivalents:

· The NFA with ε can be converted to NFA without ε & and

NFA with ε → NFA without ε → DFA

Conversion from NFA with ε to NFA without ε.

In this method we try to remove all the ε transition from given nfa

Step 1: Find out all ε transitions from each from each state

Step 2: Then obtain the δ' transitions from one state to an

Step 3: Step 2 is repeated for each i/p symbol & for each state of given nfa.

Step 4: using the resultant states the transition table for equivalent NFA without ε can be built

Rule for conversion:

$\delta'(q, a) = \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), a))$.

where $\delta'(q, \epsilon) = \epsilon\text{-closure}(q)$

Ex: Convert the given NFA with ϵ to NFA without ϵ

St 1: Find out all the ϵ transition from each state.

$\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1, q_2)$

$\epsilon\text{-closure}(q_2) = \{q_2\}$

St 2: Find out δ' transitions for each state

$\delta'(q_0, 0) = \epsilon\text{-closure}(\delta(\hat{\delta}(q_0, \epsilon), 0))$

$\epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 0))$

$= \epsilon\text{-closure}(\delta(q_0, q_1, q_2), 0))$

$= \epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\delta'(q_0, 1) = \epsilon\text{-closure}(\delta(\hat{\delta}(q_0, \epsilon), 1))$

$\epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), 1))$

$\epsilon\text{-closure}(\delta(q_1, q_2), 1)$

$= \epsilon\text{-closure}(\phi) = \phi$

$\delta'(q_1, 1) = \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), 1)$

$\epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1), 1))$

$\epsilon\text{-closure}(\delta(q_1, q_2), 1)$

$\epsilon\text{-closure}(q_1, 1) = \{q_1, q_2)$

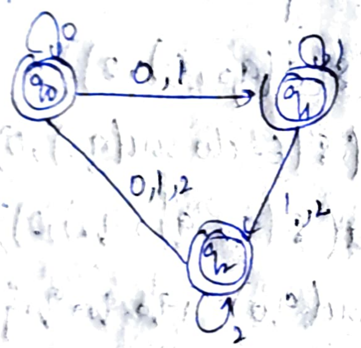
$\delta(q_1, 2) = \epsilon\text{-closure}(\delta(\hat{\delta}(q_1, \epsilon), 2))$

$\epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_1), 2))$

$\epsilon\text{-closure}(\delta(q_1, q_2), 2))$

$\epsilon\text{-closure}(q_2, 2) = \{q_2\}$

	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$



$\therefore$ q_0, q_1, q_2 is a final state because; $\epsilon\text{-closure}(q_0)$, $\epsilon\text{-closure}(q_1)$, $\epsilon\text{-closure}(q_2)$ contains final state q_2

convert the following NFA with ϵ to NFA without ϵ

ϵ-closure$(q_0) = q_0$

ϵ-closure$(q_1) = q_1, q_2$ - final state

ϵ-closure$(q_2) = q_2$ - final state

* $\delta'(q, a) = \epsilon\text{-closure}(\delta(\hat{\delta}(q, \epsilon), a))$

where $\hat{\delta}(q, \epsilon) = \epsilon\text{-closure}(q)$

$\delta'(q_0, a) = \epsilon\text{-closure}(\delta(\hat{\delta}(q_0, \epsilon), a))$

$= \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0), a))$

$= \epsilon\text{-closure}(\delta(q_0, a))$

$= \epsilon\text{-closure}(q_1)$

$= \{q_1, q_2\}$

$\delta'(q_0, b) = \phi$

$\delta'(q_1, a) = \epsilon\text{-closure}(\delta((q_1, q_2), a))$

$= \epsilon\text{-closure}(\delta(q_1, a) \cup \delta(q_2, a))$

$= \epsilon\text{-closure}(\phi \cup \phi)$

$= \{\phi\}$

$\delta'(q_1, b) = \phi \cup q_2$

$= \{q_2\}$

$\delta'(q_2, a) = \epsilon\text{-closure}(\hat{\delta}(\hat{\delta}(q_2, \epsilon), a))$

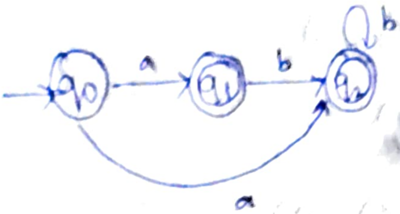
$= \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_2), a))$

$= \epsilon\text{-closure}(\delta(q_2, a))$

$= \{\phi\}$

$\delta'(q_2, b) = \{q_2\}$

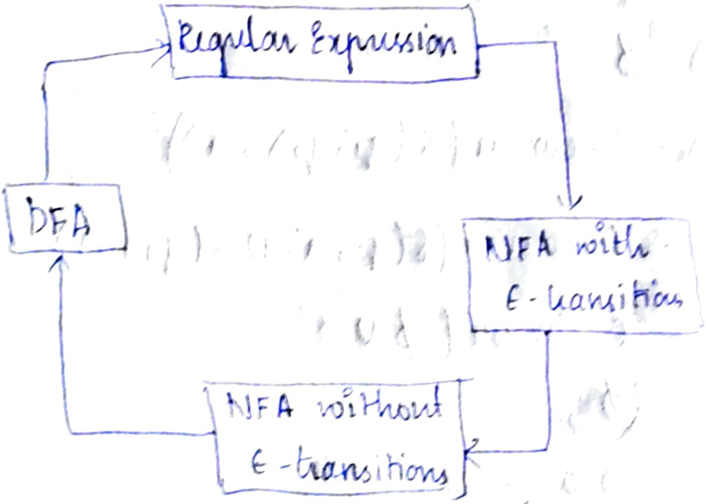
States \ I/P	a	b
q_0	$\{q_1, q_2\}$	-
q_1	-	$\{q_2\}$
q_2	-	$\{q_2\}$



★ Convert NFA with ε to DFA

NFA with ε → NFA without ε → DFA

Q) Constructing Finite automata for given Regular Exp



if r be a regular Exp then there exist a NFA with ε moves which accepts $L(r)$

Construction of NFA with ε moves

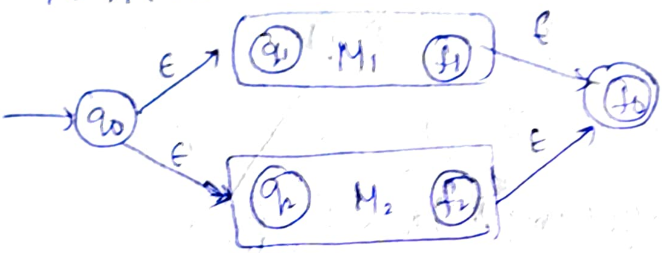
1) $r = \epsilon$

3) $r = a$

2) $r = \phi$

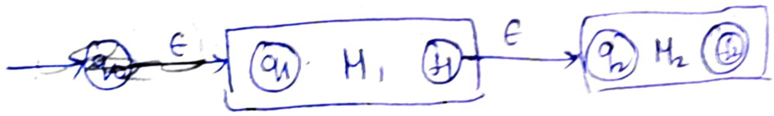
4) union

$r = r_1 + r_2$

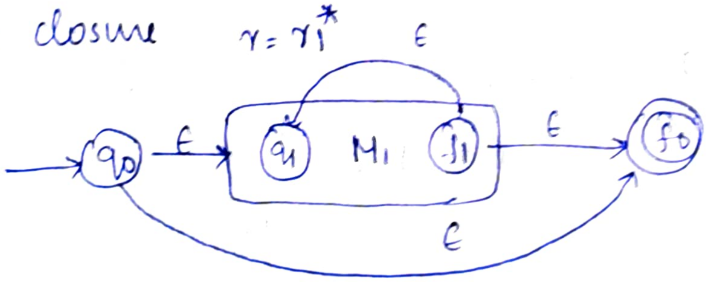


5) Concatanation

$r = r_1 \cdot r_2$



case 6) closure $r = r_1^*$

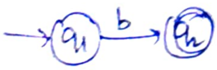


Q) Construct NFA for RE $b + ba^*$

$RE = b + ba^*$

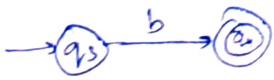
$r = \underbrace{b}_{r_1} + \underbrace{ba^*}_{r_2}$

$r_1 = b$

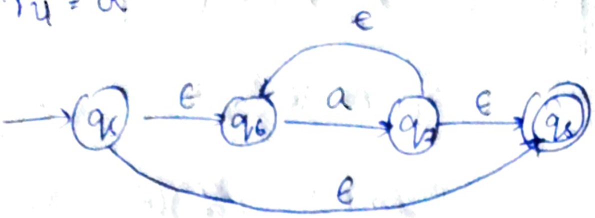


$r_2 = \underbrace{b}_{r_3}\underbrace{a^*}_{r_4}$

$r_3 = b$

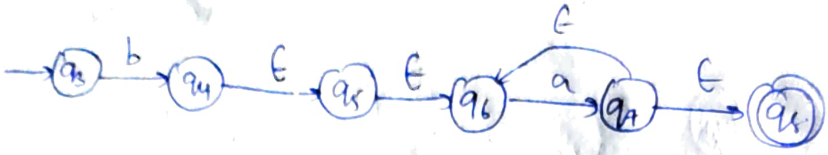


$r_4 = a^*$

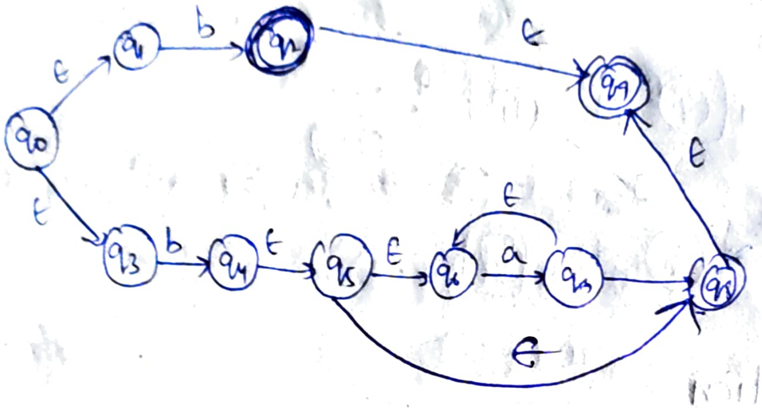


$r_2 = r_3 \cdot r_4$

$r_2 =$

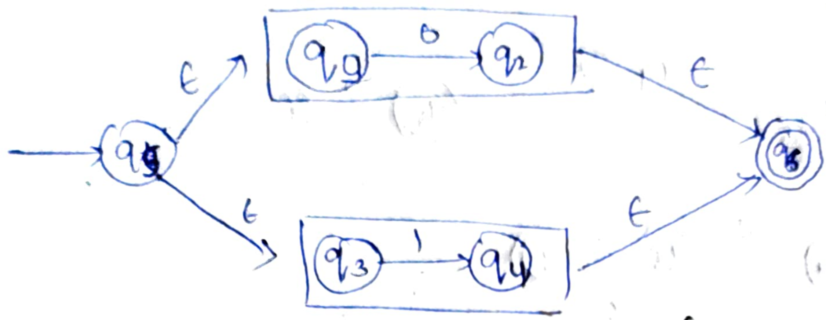


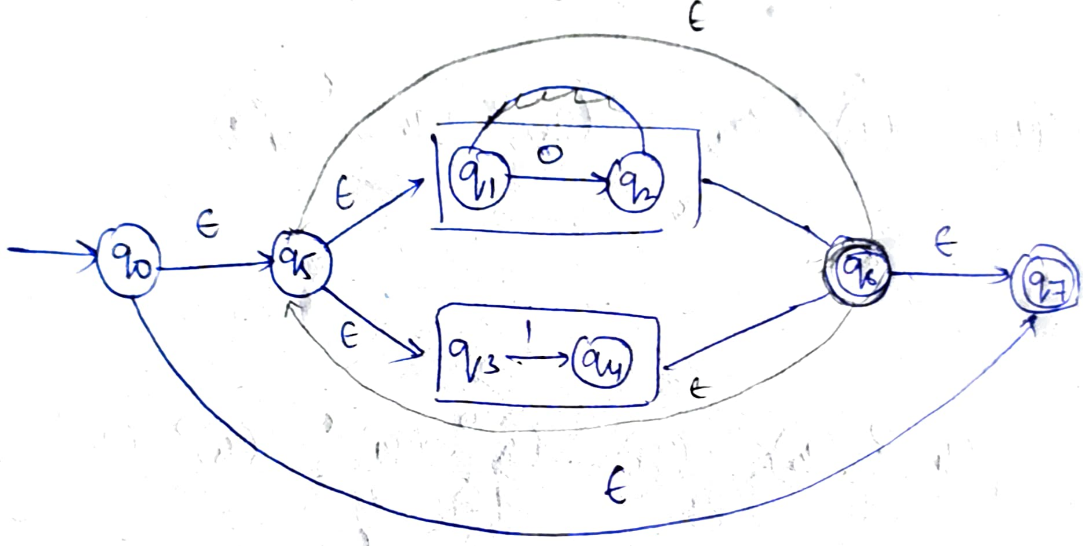
$r = r_1 + r_2$



Construct NFA with ε moves for RE $(0+1)^*$

$r_1 = 0$ $\quad$ $r_2 = 1$





Construct the NFA for the lang having odd no of 0 over the set $\Sigma = 0$

$RE = \underbrace{0}_{r_1}\underbrace{(00)^*}_{r_2}$

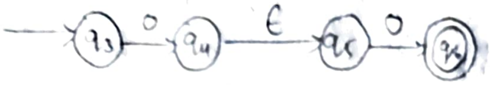
$r_1 = 0$

$r_2 = (\underbrace{0}_{r_3}\underbrace{0}_{r_4})^*$

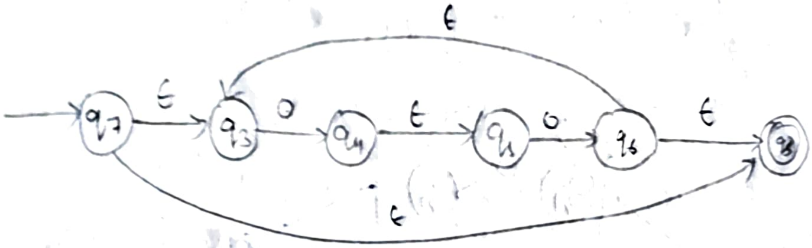
$r_3 = 0$

$r_4 = 0$

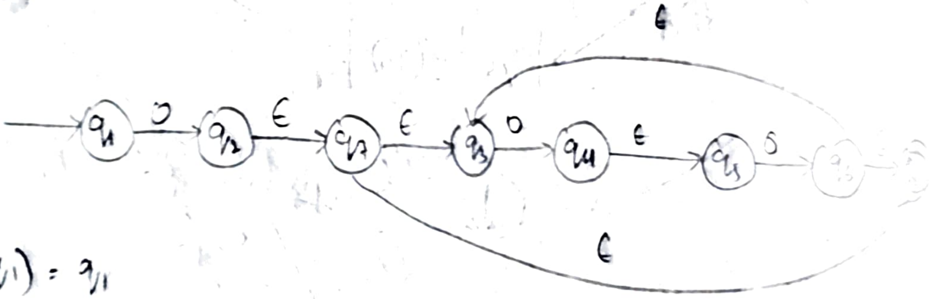
$r_2 =$



r_2^*



$r_1(r_2)^*$



$\epsilon\text{-}c(q_1) = q_1$

$\epsilon\text{-}c(q_2) = \{q_2, q_7, q_3, q_8\}$ - F

$\epsilon\text{-}c(q_3) = \{q_3\}$

$\epsilon\text{-}c(q_4) = \{q_4, q_5\}$

$\epsilon\text{-}c(q_5) = \{q_5\}$

$\epsilon\text{-}c(q_6) = \{q_6, q_8, q_3\}$ - F

$\epsilon\text{-}c(q_8) = \{q_8\}$ - F

$\epsilon\text{-}c(q_7) = \{q_7, q_3, q_8)$ - F

$\delta'(q_0, 0) = \epsilon\text{-}c(\delta(\hat{\delta}(q_0, \epsilon), 0))$

$\epsilon\text{-}c(\delta(\hat{\delta}(q_0, 0), 0))$

$\epsilon\text{-}c(\delta(q_1), 0))$

$\epsilon\text{-}c(q_2) \Rightarrow \{q_2, q_3, q_7, q_8)$

$\delta'(q_2, 0) \cdot \epsilon\text{-}c(\delta(q_2, q_3, q_7, q_8), 0))$

$\epsilon\text{-}c(\delta(q_2, 0) \cup \delta(q_3, 0) \cup \delta(q_7, 0) \cup \delta(q_8) 0))$

$\epsilon\text{-}c(\emptyset \cup q_4 \cup \emptyset \cup \emptyset)$

$\{q_4, q_5\}$

$\delta'(q_3, 0) \cdot \epsilon\text{-}c(\delta(q_3, 0))$

$\epsilon\text{-}c(q_4) = \{q_4, q_5\}$

$\delta'(q_4, 0) \cdot \epsilon\text{-}c(\delta(q_4, q_5), 0)$

$\epsilon\text{-}c(\delta(q_4, 0) \cup \delta(q_5, 0))$

$\epsilon\text{-}c(\emptyset \cup q_6)$

$\{q_3, q_6, q_8\}$

$\delta'(q_5, 0) = \epsilon\text{-}c(\delta(q_5, 0))$

$= \epsilon\text{-}c(q_6)$

$= (q_6, q_3, q_8$

$\delta'(q_6, 0) \cdot \epsilon\text{-}c(\delta(q_6, q_3, q_8), 0)$

$= \epsilon\text{-}c(\delta(q_6, 0) \cup \delta(q_3, 0) \cup \delta(q_8, 0))$

$= \epsilon\text{-}c(\emptyset \cup q_4 \cup 0)$

$\Rightarrow \{q_4, q_5)$

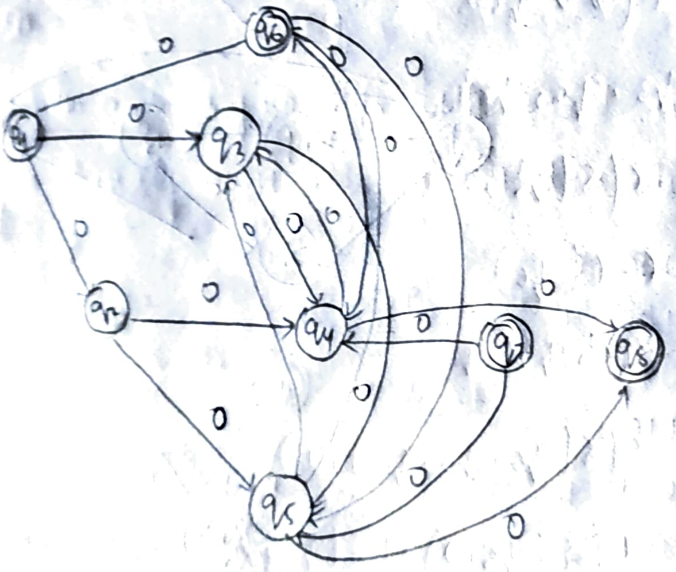
$\delta'(q_7, 0) \cdot \epsilon\text{-}c(\delta(q_3, q_7, q_8), 0)$

$\epsilon\text{-}c(\delta(q_3, 0) \cup \delta(q_7, 0) \cup \delta(q_8, 0))$

$\epsilon\text{-}c(q_4 \cup \emptyset \cup \emptyset) \Rightarrow \{q_4, q_5\}$

$\delta'(q_3, 0) = \epsilon\text{-}c(\delta(q_5, 0))$
$= \phi$

state \ V_1	0
q_1	$\{q_2, q_3, q_6, q_8\}$
q_2	$\{q_4, q_5\}$
q_3	$\{q_4, q_5\}$
q_4	$\{q_3, q_6, q_8\}$
q_5	$\{q_3, q_6, q_8\}$
q_6	$\{q_4, q_5\}$
q_7	$\{q_4, q_5\}$
q_8	—



a) $(\underbrace{01}_{r_1} + \underbrace{10}_{r_2})^*$

$r_2 : \underbrace{0}_{r_5}\underbrace{1}_{r_6}$

$r_5 = 0 \quad \to q_5 \xrightarrow{0} q_6$

$r_1 = \underbrace{0}_{r_3}\underbrace{1}_{r_4}$

$r_4 = 1$

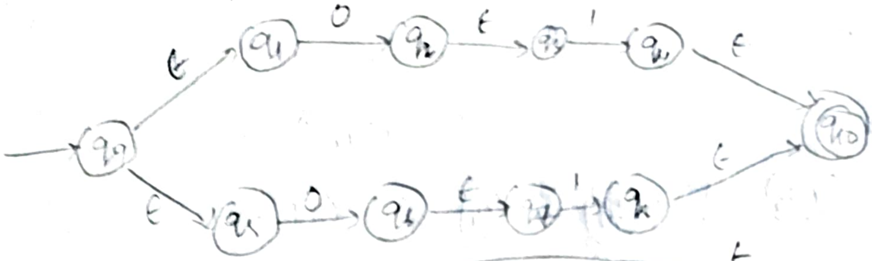
$r_6 = 1 \quad \to q_7 \xrightarrow{1} q_8$

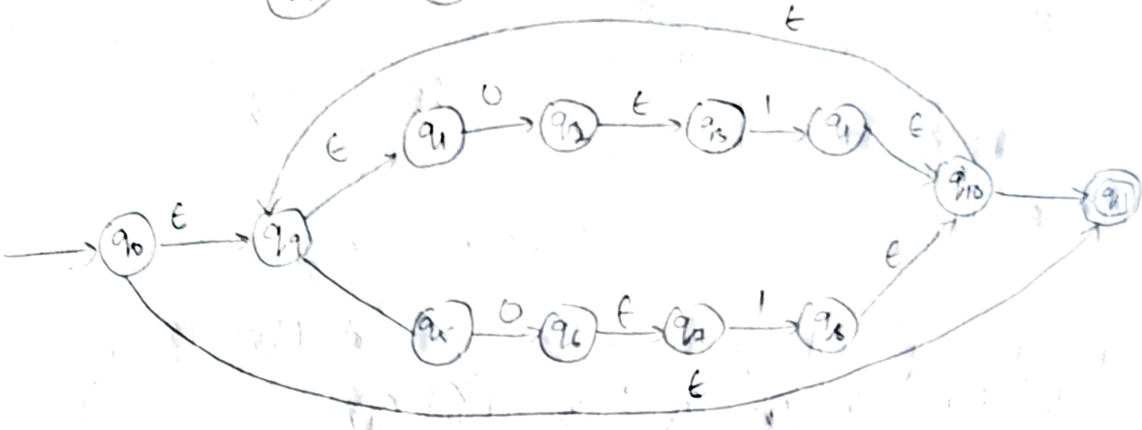
$r_3 = 0$

$\to q_1 \xrightarrow{0} q_2 \qquad \to q_3 \xrightarrow{1} q_4$

$r_2 = \ \to q_5 \xrightarrow{0} q_6 \xrightarrow{\epsilon}$

$r_1 = \ \to q_1 \xrightarrow{0} q_2 \xrightarrow{\epsilon} q_3 \xrightarrow{1} q_4$



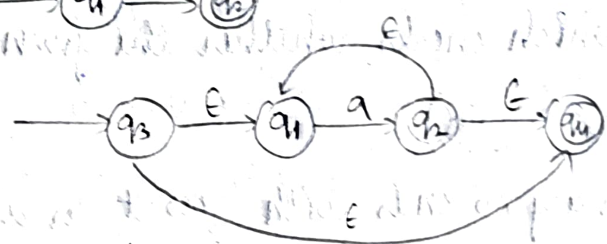


Q) RE = $a^* \, b \, (a+b)^*$

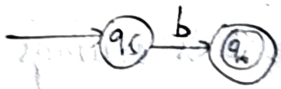
$r_1 \quad r_2 \quad r_3 \quad r_4$

$r_1 = a$

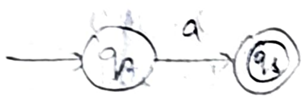
$r_1^* \Rightarrow$



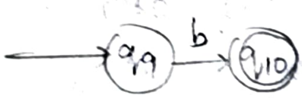
$r_2 = b$



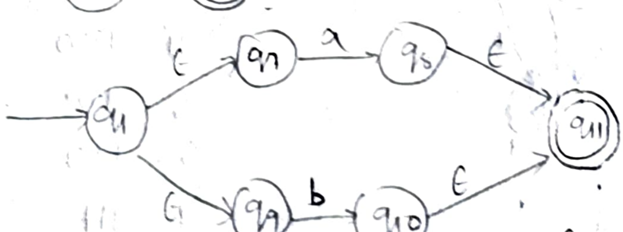
$r_3 = a$



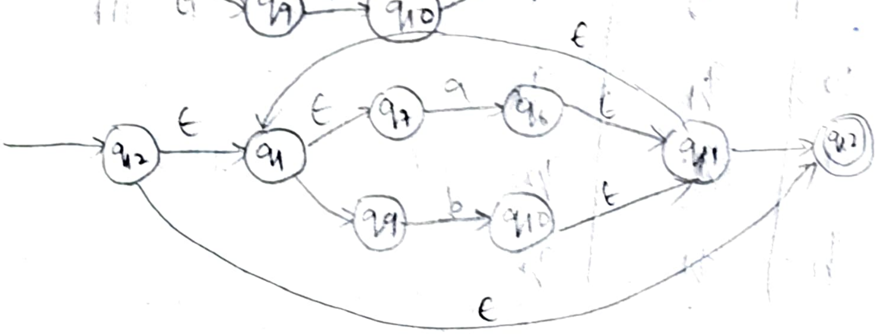
$r_4 = b$



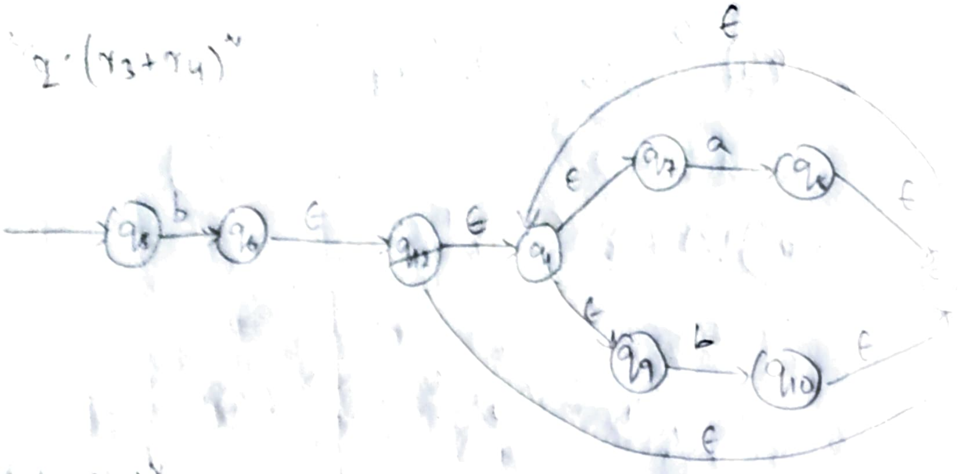
$r_3 + r_4 =$



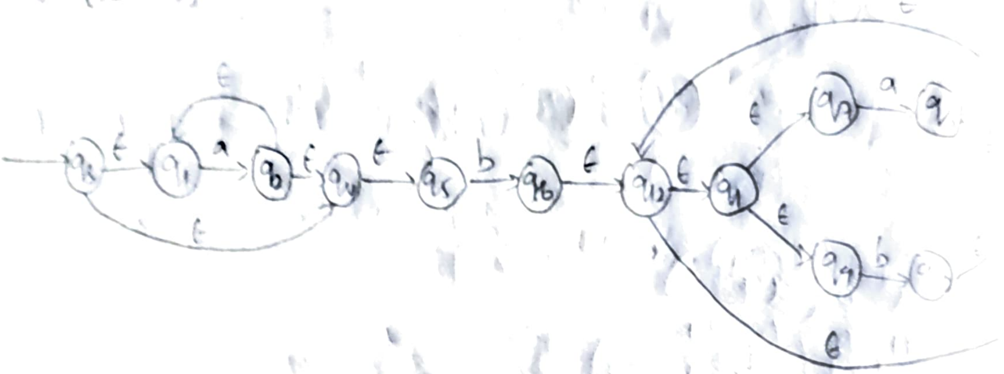
$(r_3 + r_4)^* =$



$r_2 \cdot (r_3 + r_4)^*$

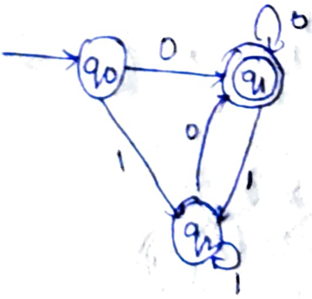


$r_1^* r_2 (r_3 + r_4)^*$



Design a DFA which checks whether the given binary num is even.

Note: when a binary no ends with zero it is also even
when a binary num ends with 1 it is always odd



0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

	0	1
q_0	q_1	q_2
q_1	q_1	q_2
q_2	q_1	q_2

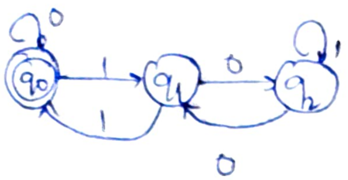
Q) design a DFA such-that binary no is div by 3 over set Σ={0,1}

$q_0 \to$ rem 0

$q_1 \to$ rem 1

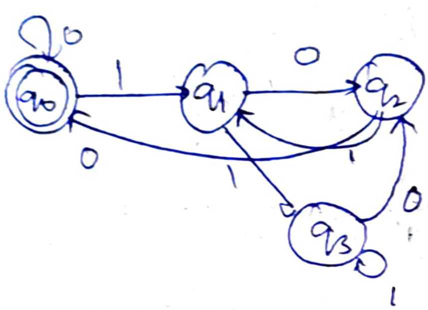
$q_2 \to$ rem 2

0 - 000 - r - 0
1 - 001 - 1
2 - 010 - 2
3 - 011 - 0
4 - 100 - 1
5 - 101 - 2
6 - 110 - 0
7 - 111 - 1
8 - 1000 - 2



	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

Q) design a DFA binary no is div by 4



$q_0 \to$ rem 0

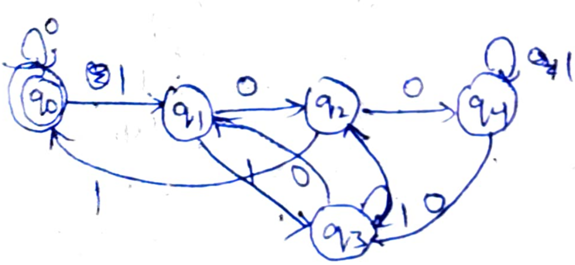
$q_1 \to$ rem 1

$q_2 \to$ rem 2

$q_3 \to$ rem 3

0 - 000 - 0
1 - 001 - 1
2 - 010 - 2
3 - 011 - 3
4 - 100 - 0
5 - 101 - 1
6 - 110 - 2
7 - 111 - 3
8 - 1000 - 0

	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



$q_0 \to$ rem 0

$q_1 \to$ rem 1

$q_2 \to$ rem 2

$q_3 \to$ rem 3

$q_4 \to$ rem 4

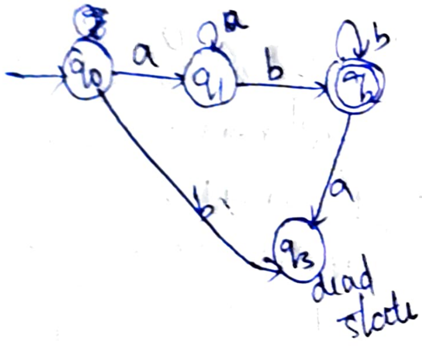
9 - 1001 - 4

10 - 1010 - 0

state \ i/p	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

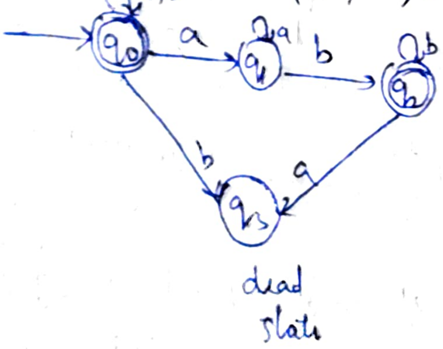
Q) design a DFA such the $L = \{a^n b^m \mid n, m \geq 1\}$

$L = \{ab, aab, abb, aaabbb, aabbb, \ldots\}$

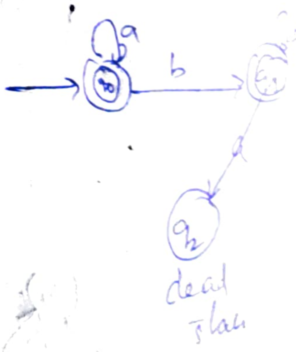


Q) $L = \{a^n b^m \mid n, m \geq 0\}$ $\Sigma = \{a, b\}$ $\Rightarrow$

$L = \{\epsilon, a, b, ab, aab, abb, aaabb$

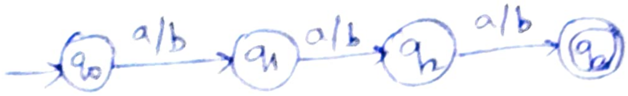


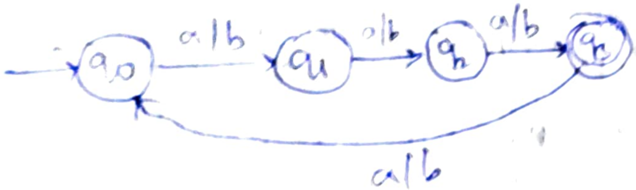
(or)



Q) Design a DFA such that string with a's & b's & length is exactly 3

Q) string with a's & b's length is div by 3

1) 

2) 

[illegible]