To understand — Theory of computation. |14/8/24|

→ (1) Mathematical models of a computer
→ (2) TOC has a tons of real-world problem

Program :- Express algorithm

Algorithm :- An algorithm is recipe

→ Every algorithm tells us how to compute a function.

function :- It is a abstract notion which tells me that here is a mapping b/w i/P & o/P domain and range. And a algorith tells to g how to get an o/P.

Eg :- is_Prime_Number → {Yes, No}

$$\text{is\_Prime\_Number} = \begin{cases} \text{Yes, if } (n) \text{ is Prime} \\ \text{No, if } (n) \text{ is not Prime} \end{cases}$$

If, we consider the class of all function then there are only some tiny function who allow them to compute according to algorithm & reset do not allow.

Basic Goal :- To identify the class of function ~~who~~ which can admit algorithm to compute them.

what is machine ⇒ Mathematical model.

Set member Problem :-

S a set

Given any a, b to decide if $a \in S$

Suppose, we show that there is no algorith to solve the set member problem of graph (f).
Then, we can conclude that there is no algorithm to compute f.

Suppose there is an algorithm to compute f for (a, b), given as input.
we compute
f(a) using algo for computing f

$b = f(a) \Leftrightarrow (a, b) \in graph(f)$

* our sets are going to be set of finite string.

symbols ⇒ 0, 1, a, b,

Alphabet : An alphabet is a finite set of symbols

Ex :- {0, 1}, {a, b, c, d ..... z} ....

Automata / Machine / Mathematical Model :-

① FSA {Finite state Automata}

② PDA {Push-Down Automata} :-

code → compiler → verify syntax

③ Turing machine :- Halting problem

To denote or define a pattern : Grammers ← Linguistics

Maths Recap :-

① set :- collection of distinct objects :-
                                  * no repetation

Eg :- $S = \{1, 2, 3\}$

size/cardinality $|S| = 3$

② cartesian Product :- consist of ordered Pair

$A = \{1, 2\}$

$B = \{a, b, c\}$

$A \times B = \{(1,a)\ (1,b)\ (1,c)\ (2,a)\ (2,b)\ (2,c)\}$

→ such that

$A \times B = \{(a,b) \mid a \in A,\ b \in B\}$

③ Relation :- subset of cartesian Product.

④ Function :-

⑤ Graph

sequence of alphabet.

(6) sequences : (A, B, C, D ..... z)
In this order is followed

(1, 1, 2, 3, 5, 8 .....) : seq. of fibonacci no.

• repetations are allowed.

Formal - language / linguistics :-

(1) Alphabet - Finite set of symbols (Σ)

$\Sigma_{eng} = \{A, B, C ..... Z, a, b, c ...... z\}$

$\Sigma_{bin} = \{0, 1\}$

$\Sigma_{dec} = \{0, 1, 2, 3, ... 9, .\}$

$\Sigma_{hindi} = \{$अ, आ, इ, ई ...$\}$

$\Sigma_{empty} = \phi$

(2) string / word :- finite sequence of alphabe

Eg :- Applied, abc, 10100110, 123.45

$\boxed{\text{Length of string = no. of alphabet in word}}$

empty string ($\epsilon, \Lambda, \lambda$) $\Rightarrow$ length = 0

Length of 123.45 = 6

$\boxed{\epsilon . s = s . \epsilon = s}$

Σ^* = set of all string (including empty string.

③ language :- subset of Σ^*

④ Grammer :- set of rules to produce valid string in a formal language

Transitional function

DFA [Detesministic Finite Automata]:-

TUPles in DFA :- DFA is represented by 5 TUPles

Q → set of finite state

Σ → I/P

δ → Transition function $[Q \times \Sigma \to Q]$

q_0 → initial state

f → Final state

FA

with O/P ↓ Mealy & Moore

↓ without O/P DFA, NFA & NFA with

languages → ① Finite language
→ ② Infinite language

States :-

○ => Initial state

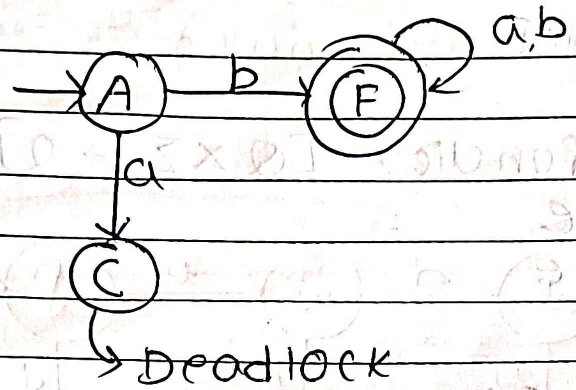
◎ => Final state

→ => Transition from one state to another

Language → starting with 'b'

→ $L = \{b, ba, bb, bab, bbb \dots\}$

Model



$Q \to \{A, C, F\}$

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

$\delta \to Q \times \Sigma = Q \Rightarrow \delta(A, b) = F$

$\delta(A, a) = C$

$q_0 \to$ Initial $= A$

$f \to$ finite state

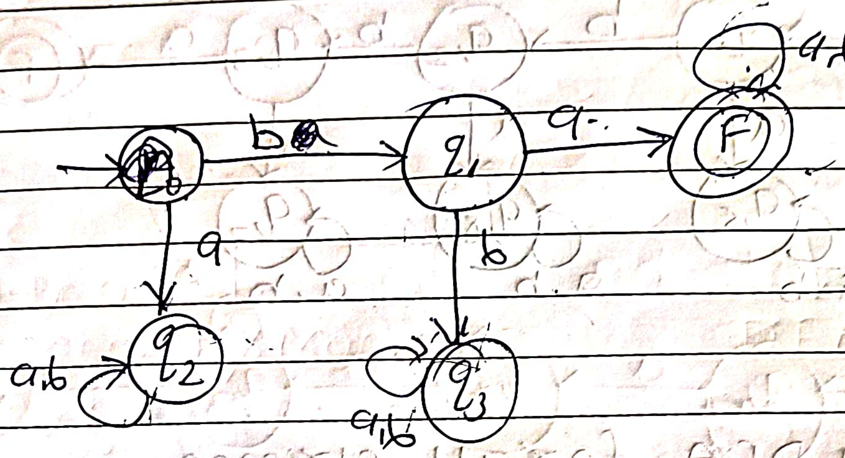
Transition	a	b
A	C	F
C	C	C
F	F	F

→ Transition table

ba

Q) Design a DFA for all string starting with 'ba' over the i/P alphabet a,b.

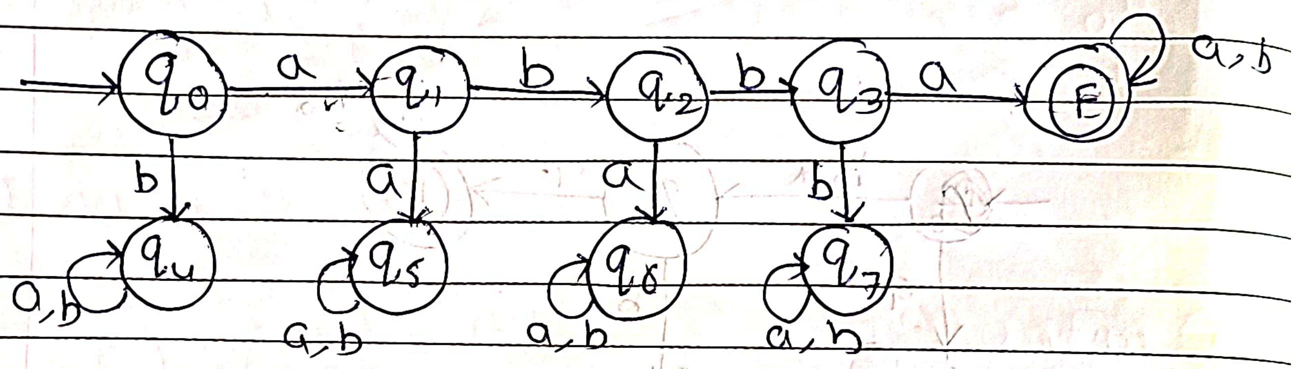
$\alpha = \{$ b, ~~b~~ ba, bab, abb, baa, bbb, baa ... $\}$



Transitions	a	b
q_0	q_2	q_1
q_1	F	q_3
q_2	q_2	q_2
q_3	q_3	q_3
F	F	F

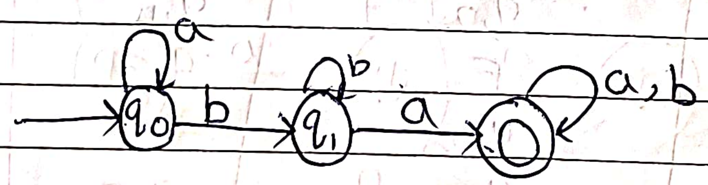
Q) a, b, b, a

$L = \{abba, aab, a, baba, abbb, abbab, abbaa \ldots\}$



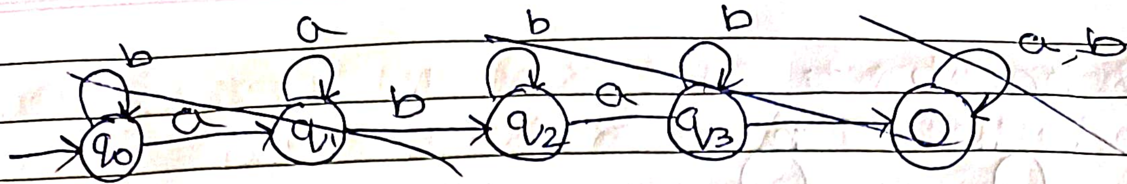
Q) Design a DFA for all strings containing ~~ab~~ba

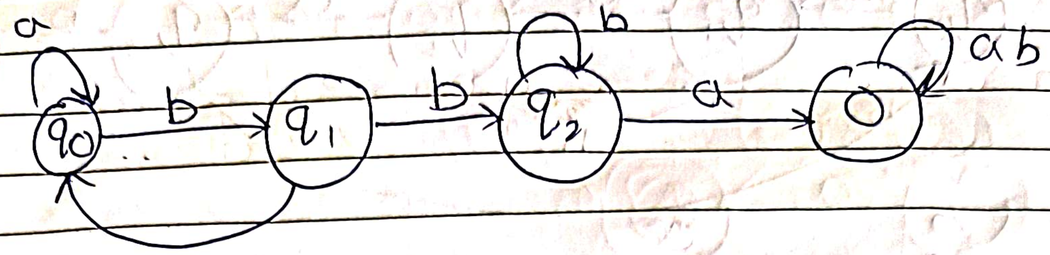
Soln: $L = \{ba, aaba, bab, baa, aabba \ldots\}$



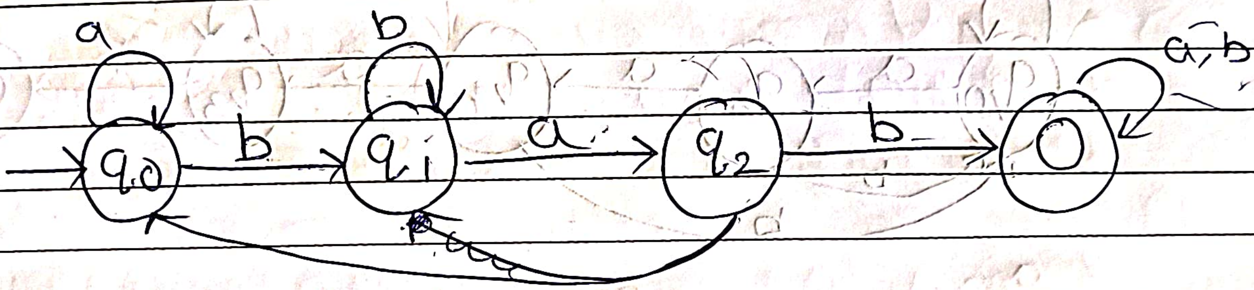
Q) Design a DFA for all strings containing aba.

Soln: $L = \{aba, baba, aaba, bbaba \ldots\}$

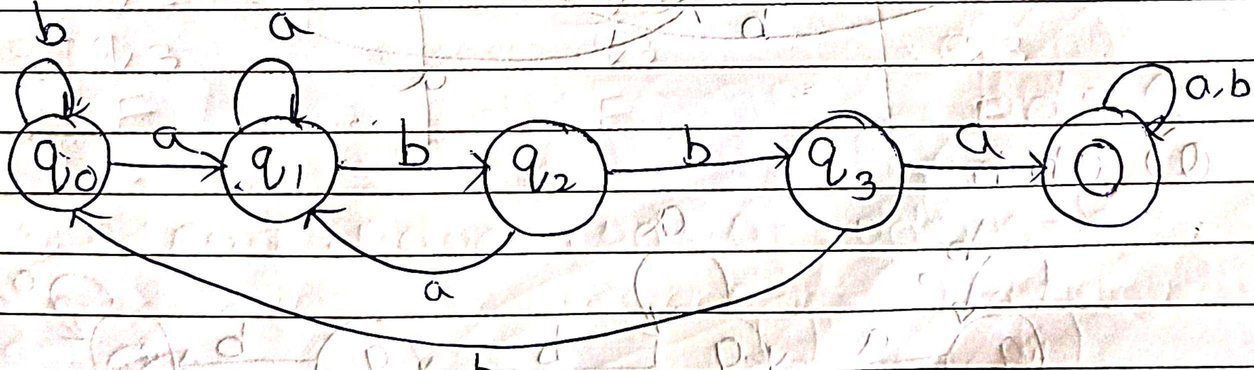

b
a
b
a
b
a
b
a, b
q_0
q_1
q_2
q_3


a
b
b
b
a
a b
q_0
q_1
q_2

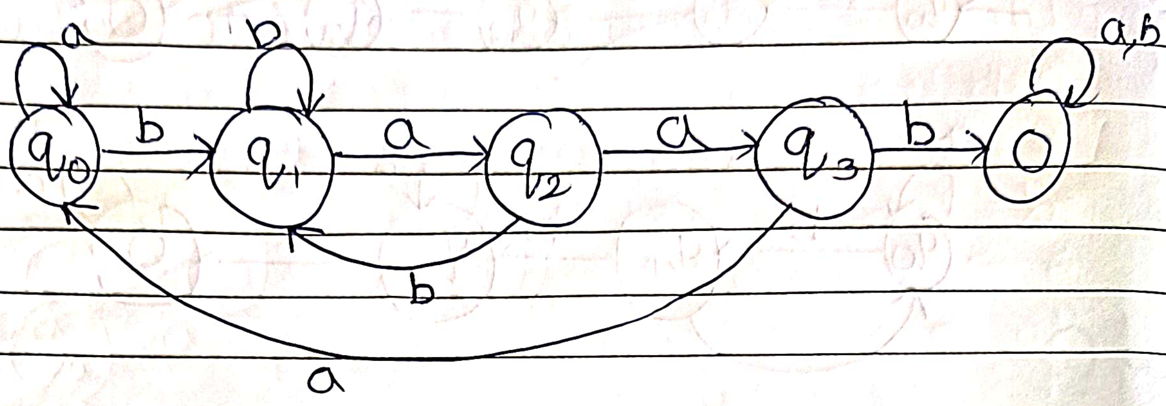
Q) bab :-


a
b
b
a
b
a, b
q_0
q_1
q_2

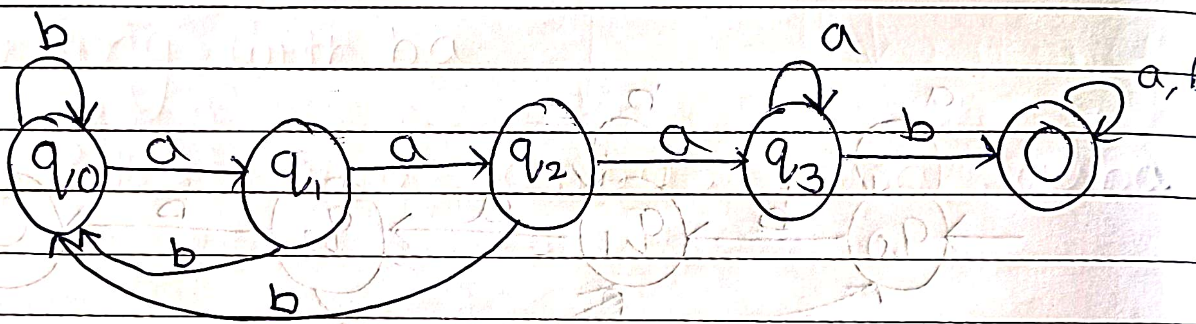
Q) abba :-


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

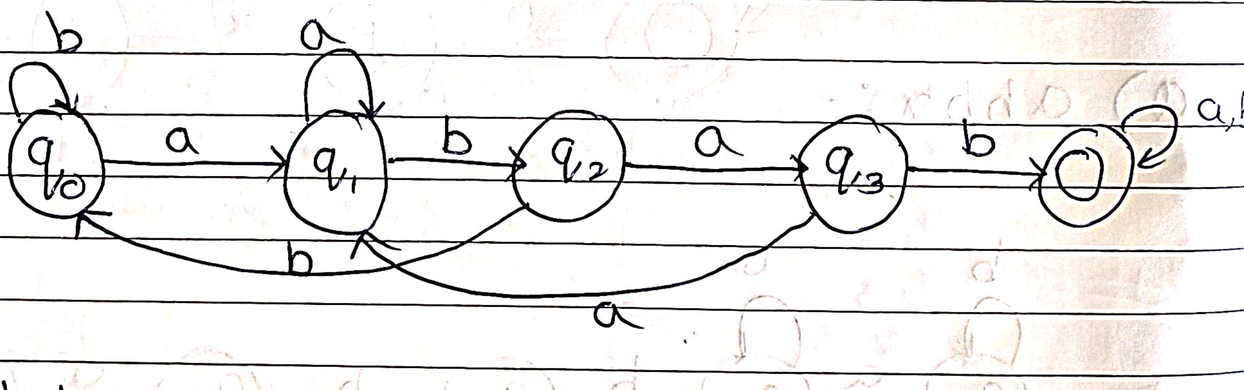
Q) baab



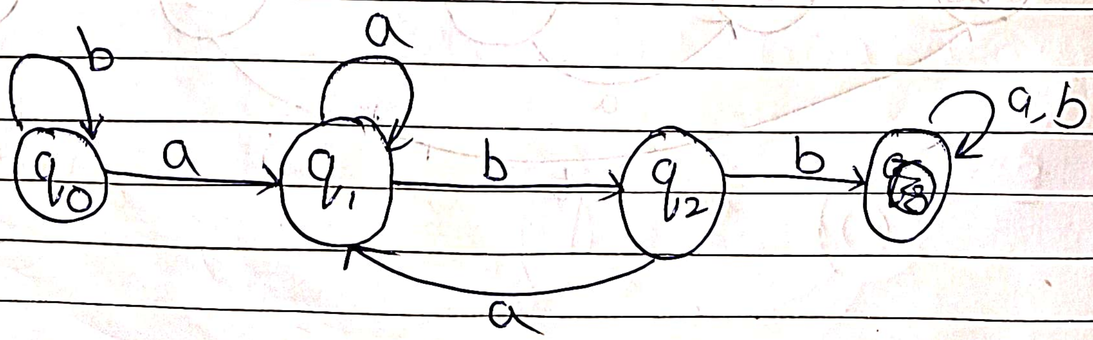
Q) aaab:



Q) abab:

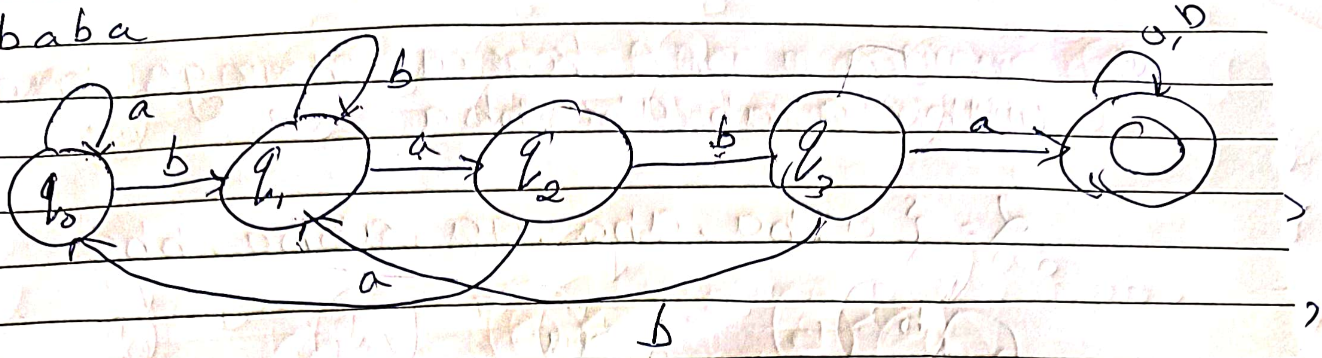


Q) abb

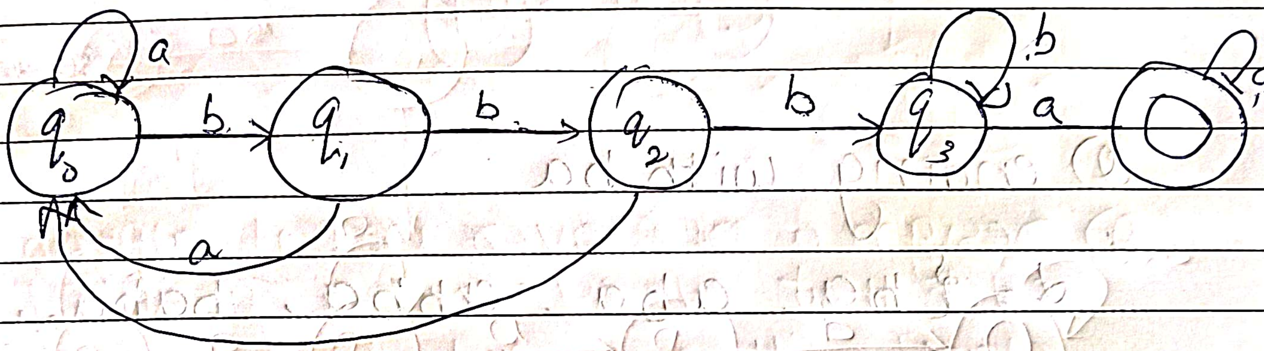


} 1 b b

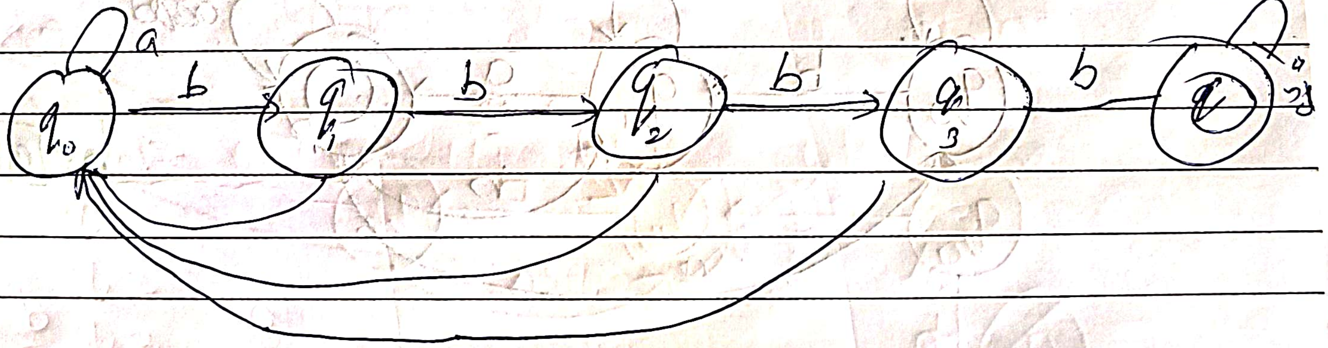
baba



Q) bbba

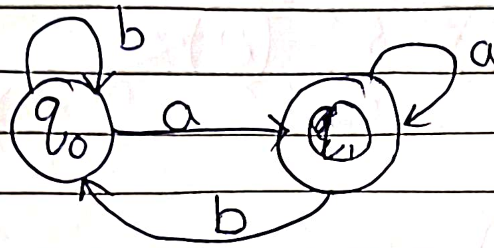


bbbb



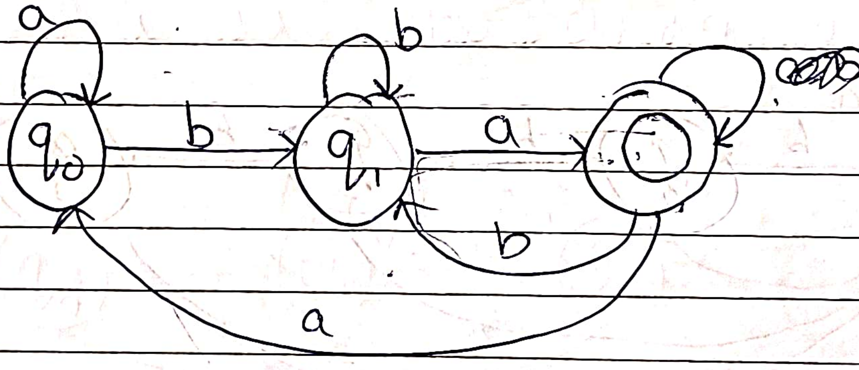
Q) Design a DFA for all strings ending with a, ab, aba, bba

$L = \{a, ba, aba, aa, abba, bba \ldots\}$



Q) ending with ba

$L = \{ba, aba, abba, baba, aaba \ldots$

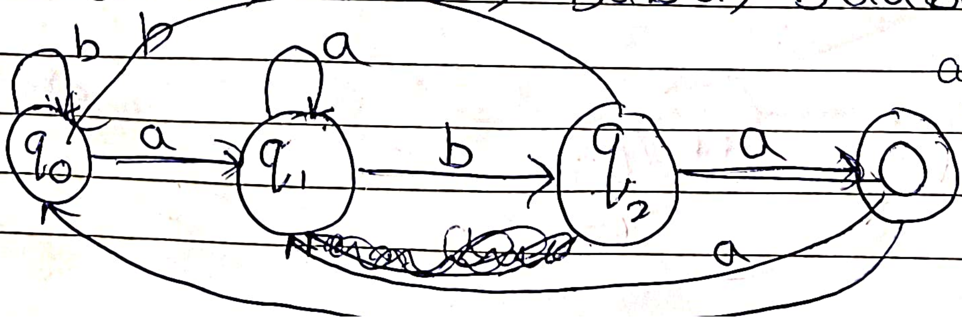


ab

ba$\boxed{a}$

Q) aba

$L = \{aba, aaba, baba, baaba \ldots\}$



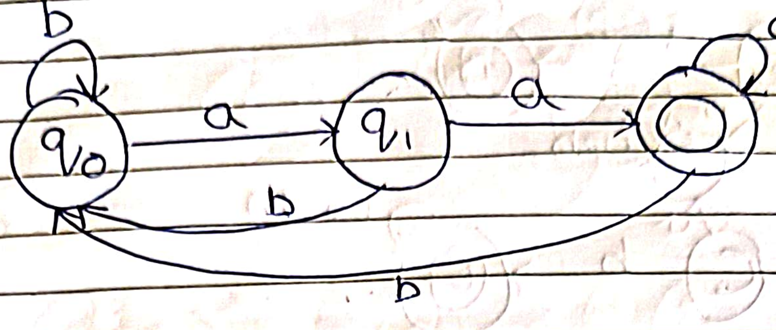
aba a

abab

ab$\boxed{b}$

Q) aa

$L = \{aa, baa, abaa, aabbaa \ldots\}$



Q) abba

$L = \{abba, aabba, babba, bbabba \ldots\}$

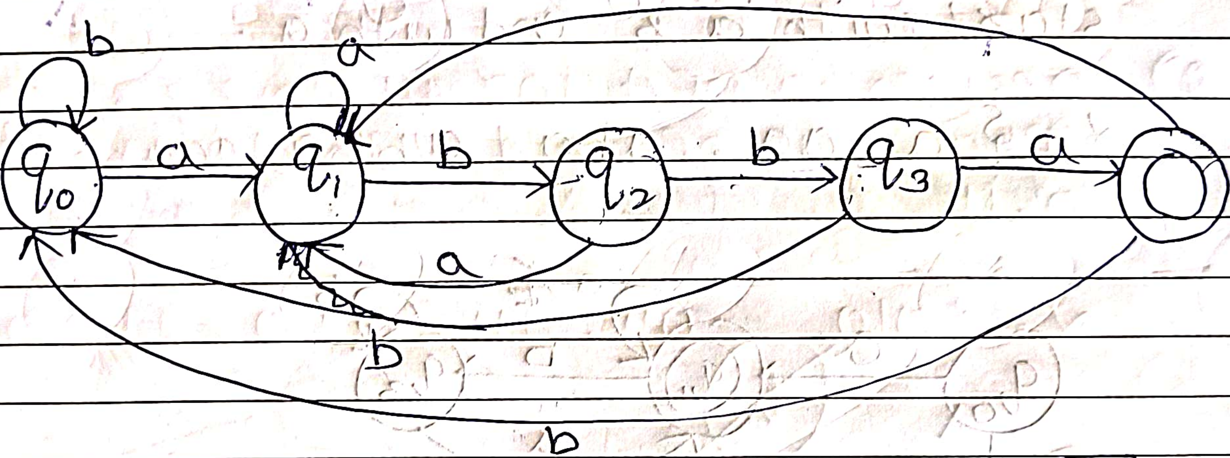


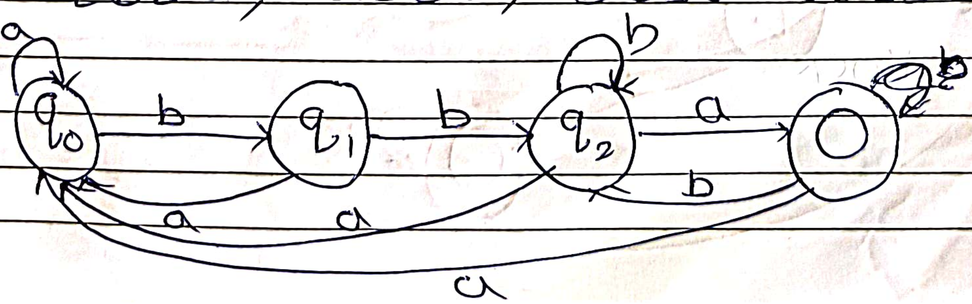
abb $\boxed{b}$

Q) bba

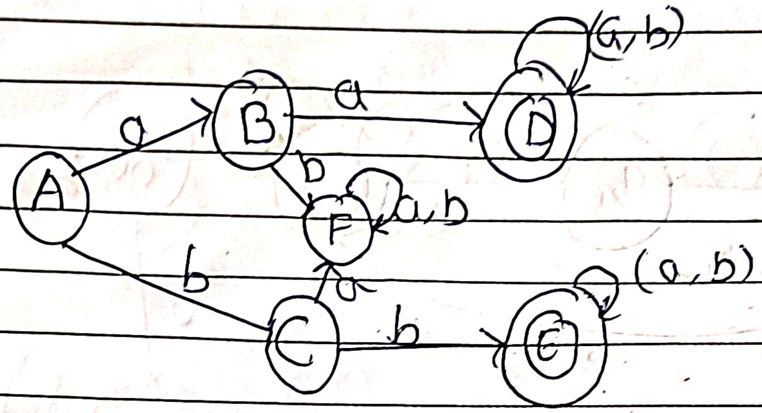
abbb

a bba a

$L = \{bba, abba, bbba\ aabba \ldots\}$

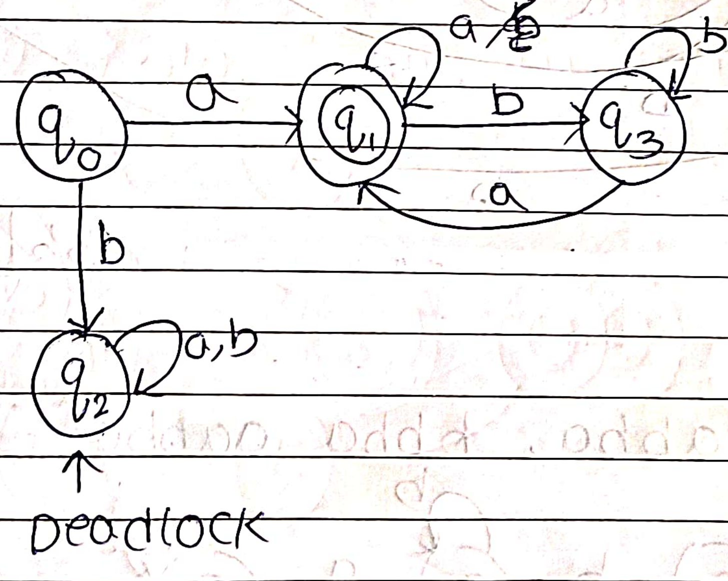


Q) Design a DFA for all strings starting either with aa or bb.



Q) Design a DFA over input alphabet a, b such that every string accepted must start with 'a' & end with a.

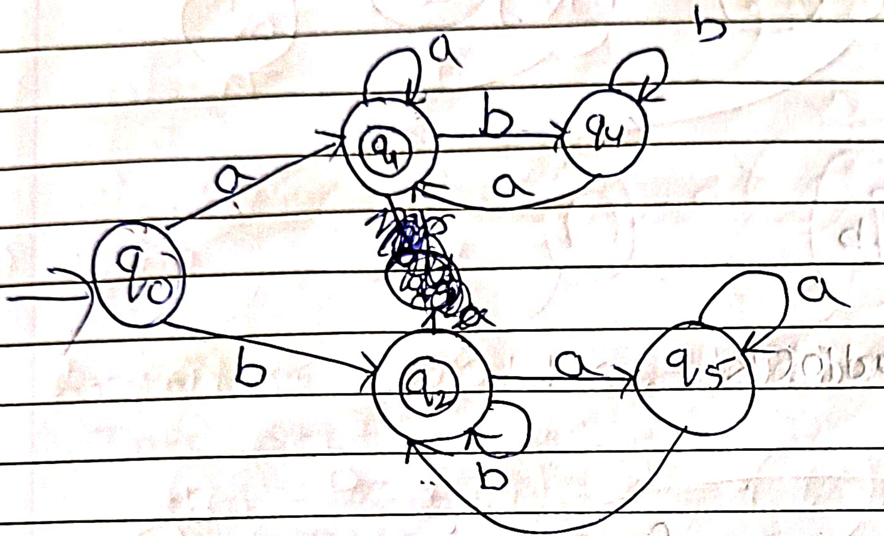
$L = \{a, aa, aba, abba, aaa \ldots\}$



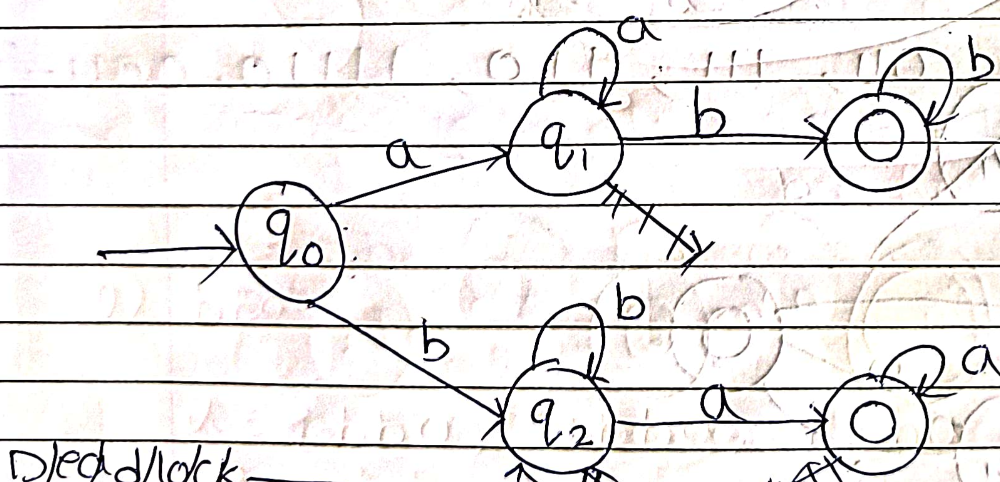
overinput
↳ IMP hai

Q) Design a over i/p alphabet a, b such that every string accepted either start or end with same symbol.

a, b
α = {aa, bb bab, aba . . . }



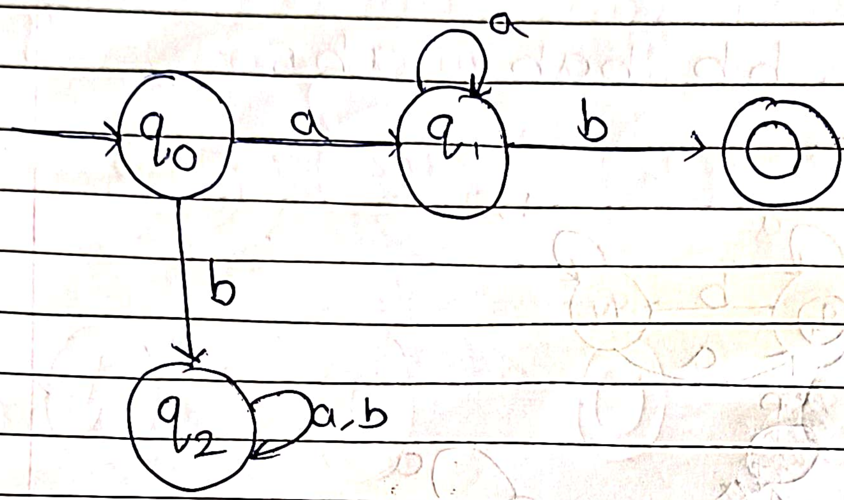
Q) DFA start with 'a' & end with b & viceversa



α = {ab, aab, ~~bab~~ aba, aaaa - - }

Start & end with the ab.

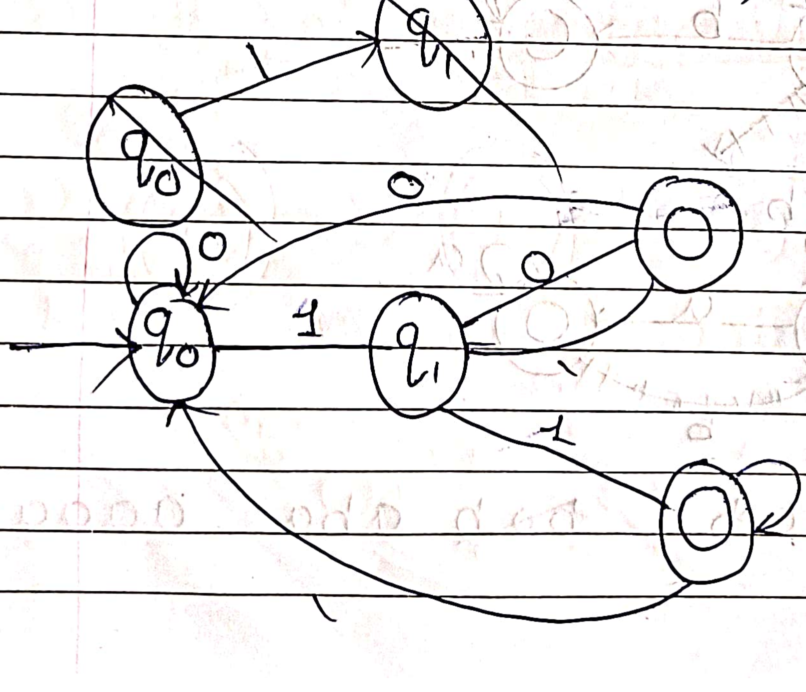
Q) Design a DFA



deadlock

Q) Design a DFA for all string over i/P alpha where end with ~~10, 11~~ either 10, 11

$\alpha = \{010, 011, 111, 110, 11110, 0011--$

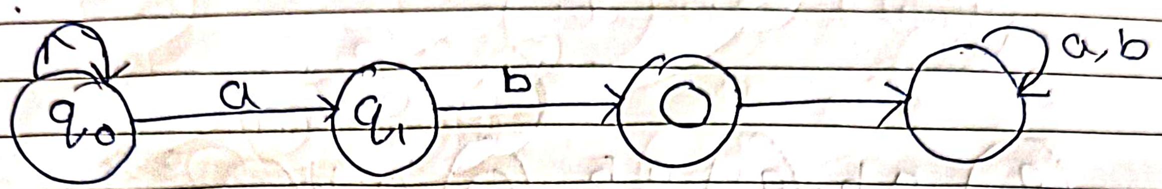


10, 11

Tanica Chandrashekhar

Q) Design a DFA for all string contain exactly length of 2 over the input alphabet a,b.

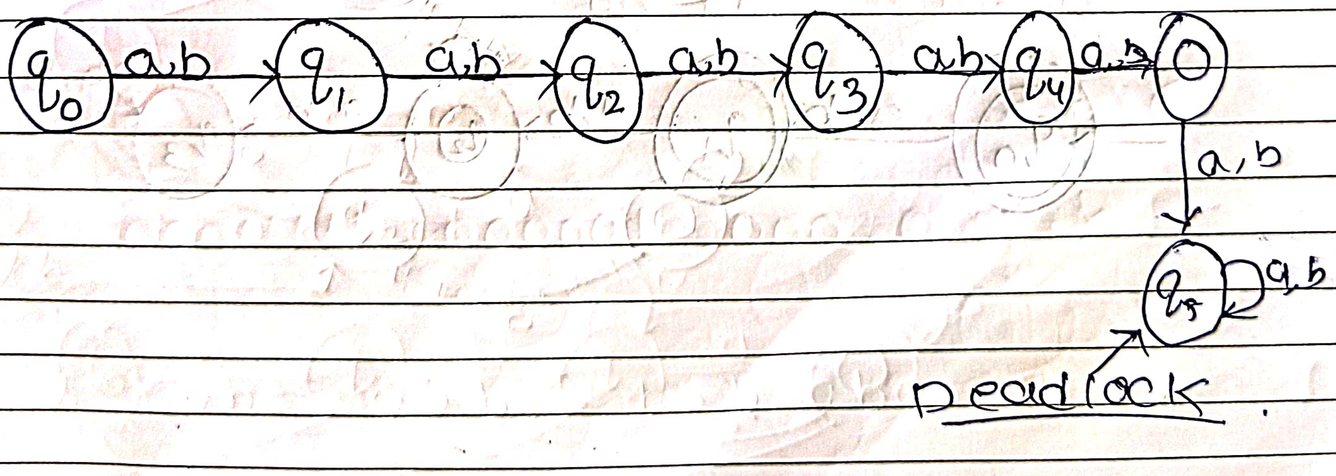
$L = \{aa, bb, ab, ba\}$



Formula→
Total state = $n+2$

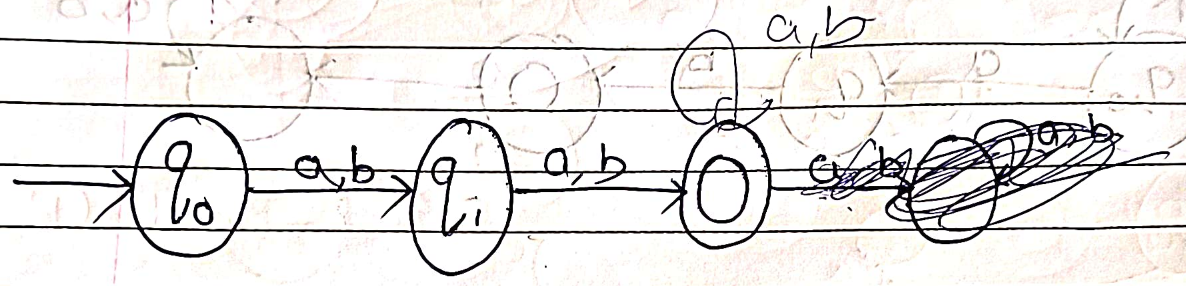
Q) length of 5

$L = \{aaaaa, aaaab, aaaba, baaaa, bbbbb, babab \ldots\}$



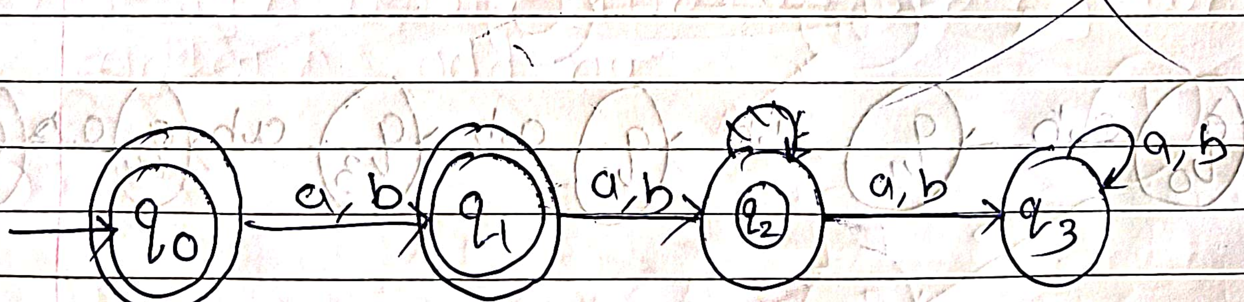
Q) atleast 2 length

$L = \{aa, aaa, abb, baa, abaa, baaab, bb, abab \ldots\}$



Q) at most 2 length

$= L = \{e, a, b, aa, ab, ba, bb$

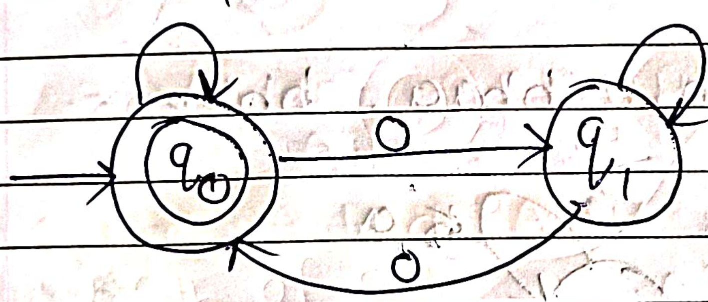


Exactly ⇒ n+2 states
atleast ⇒ n+1 "
atmost ⇒ n+2

Q) Design a DFA for all strings which contain even no. of

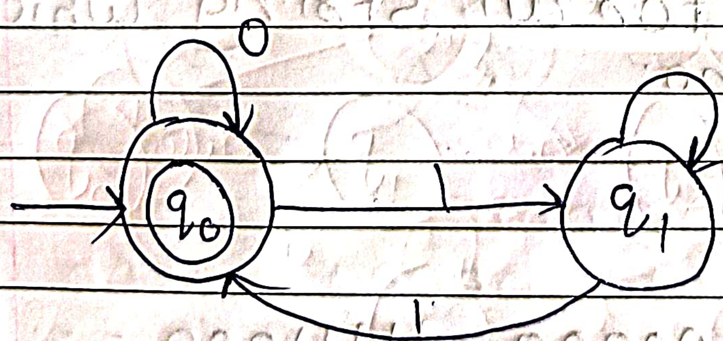
1) 0's :-

$\alpha = \{ \epsilon, 0, 00, 0011, 110011, 0000, 0000100 \ldots$



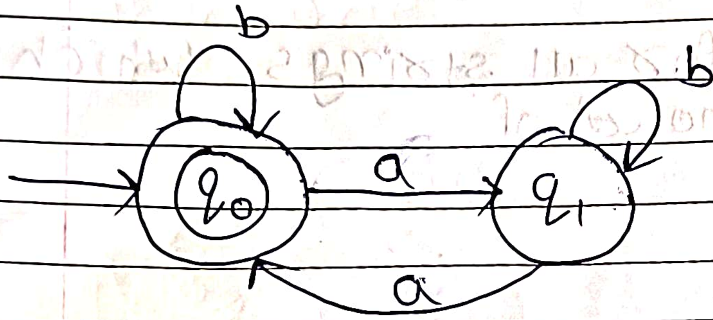
2) 1's :-

$\alpha = \{ \epsilon, 11, 1111, 110011, 111111 \ldots \}$



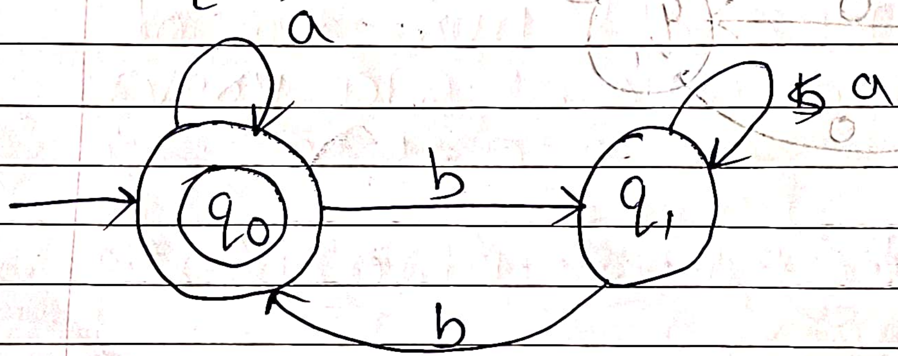
3) a's :-

$\alpha = \{\epsilon, aa, aaaa, bbaa \ldots\}$



4) b's :-

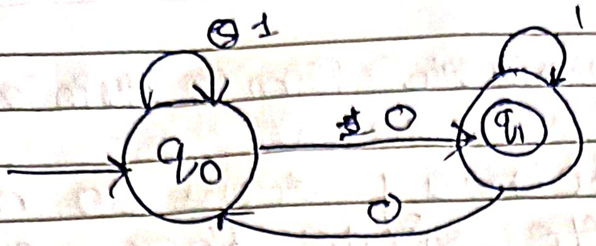
$\alpha = \{\epsilon, bb, bab, bbaa, bbbb \ldots\}$



Q) Design a DFA for all string which contain odd no.

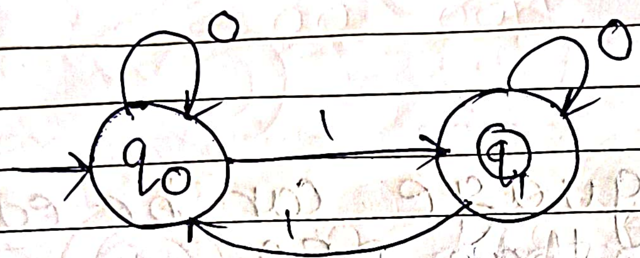
1) 0's

$\alpha = \{0, 000, 100000, 1111000 \ldots\}$



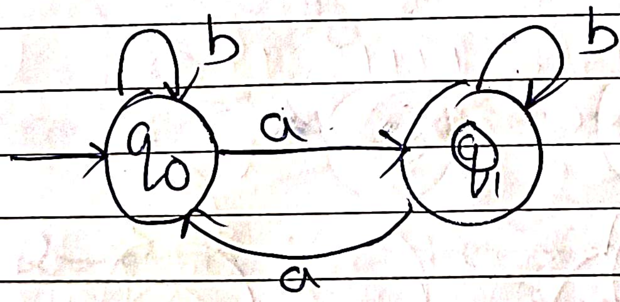
2) 1's :-

$\alpha = \{1, 11111, 111, 10101 \ldots \}$



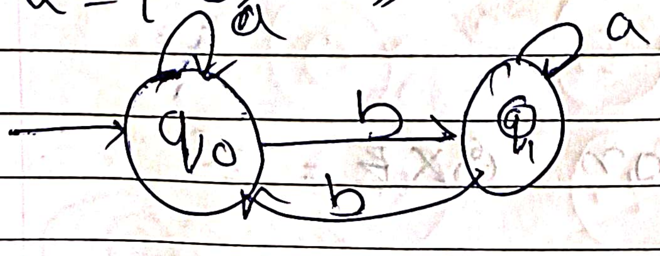
3) a's

$\alpha = \{a, aaa, ababa \text{---} \}$



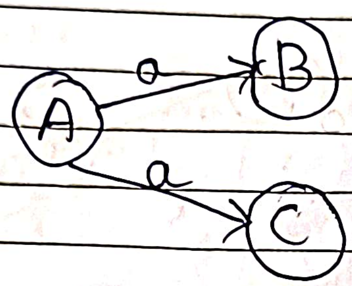
4) b's

$\alpha = \{b, bbb, babab \text{---} \}$



NFA [Non-Deterministic finite Automata].

NFA or NDFA we may lead to more than one ~~stage~~ state for a given input.



NFA for a language can be easier to construct than DFA.
Every DFA is NFA but every NFA is not a DFA.

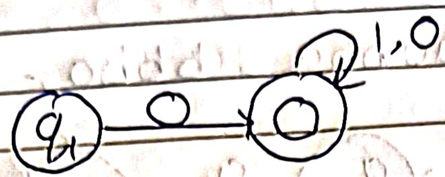
NFA also defines by '5' tuples:-

1. q
2. Σ
3. δ
4. q0
5. F

Transition function $Q \times \Sigma = 2^Q$

Q) Design a NFA over the i/p alphabet $\Sigma = 0,1$ such that every string must be start with '0'

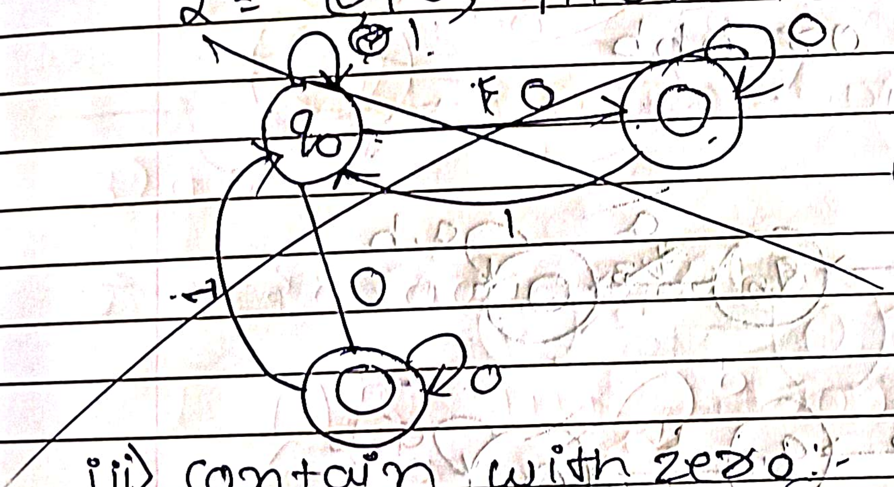
Ans ⇒

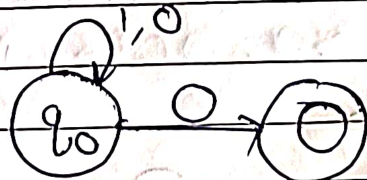


$L = \{0, 01, 00110 \ldots\}$

ii) End with zero :-

$L = \{010, 1110, 010 \ldots\}$

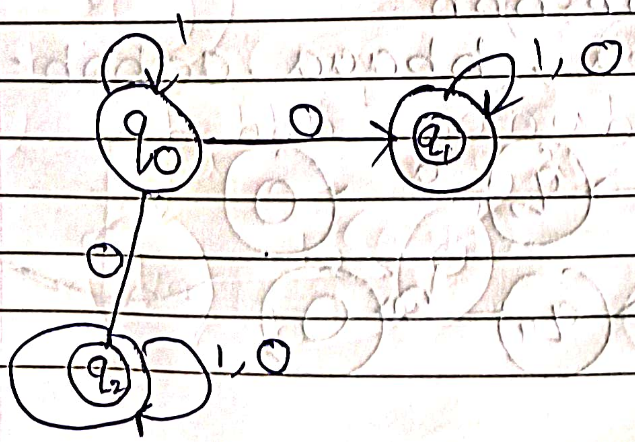


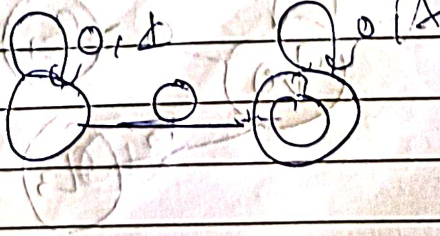


iii) contain with zero :-

$L = \{\boxed{0}, 01, 10,$

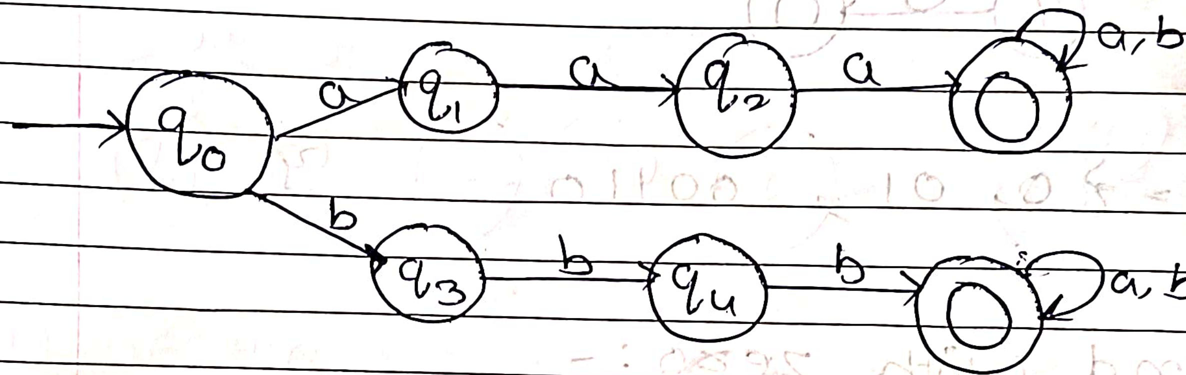
$L = \{101, 10, 01, 0011, 100111 \ldots\}$



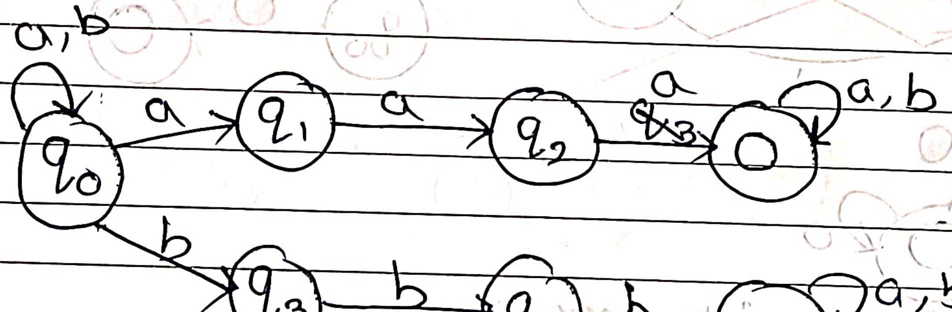


Q) Design a NFA over the i/p alphab $\Sigma = a, b$, such that every either start with aaa or bbb.

$\alpha = \{aaa, aaab, aaabba, bbba, \ldots\}$

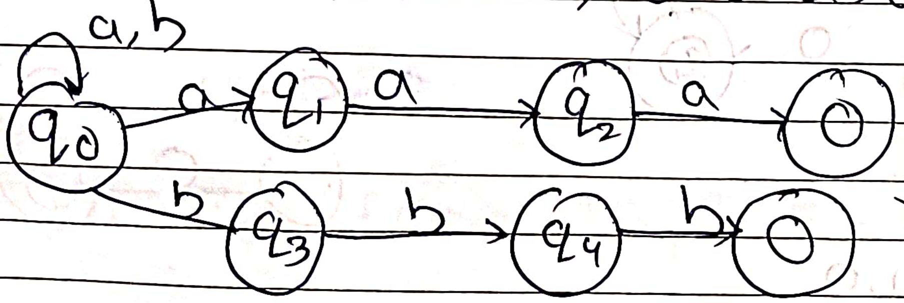


contain aaa or bbb



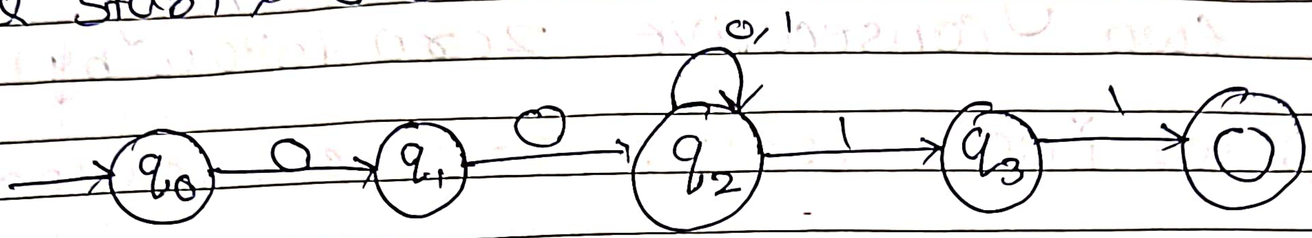
End with aaa or bbb:

$\alpha = \{baaa, aaaa, bbaaa, aabbb \ldots\}$



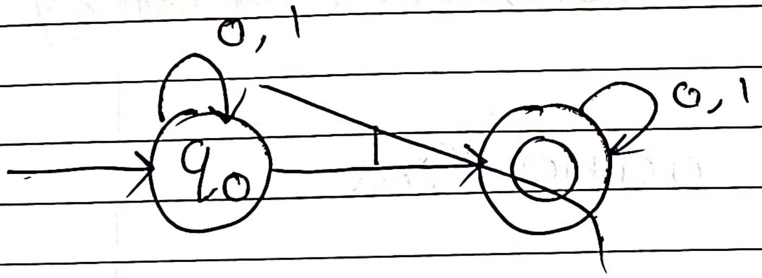
second last position '1'.

Q start & end with '0' '00' & '11'



Q) second last position '1'

α = { ~~010,~~ 00010 , 11111 , 0101010, --- }

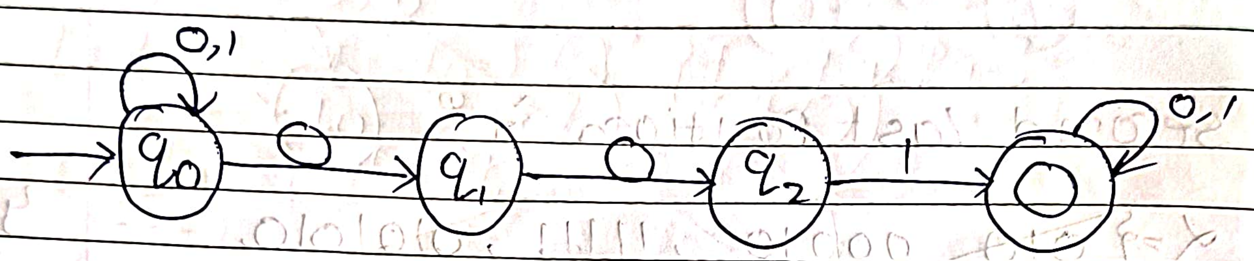


Q) ~~Desi~~

α = {10, 11, 010, 00110 11111 111... }

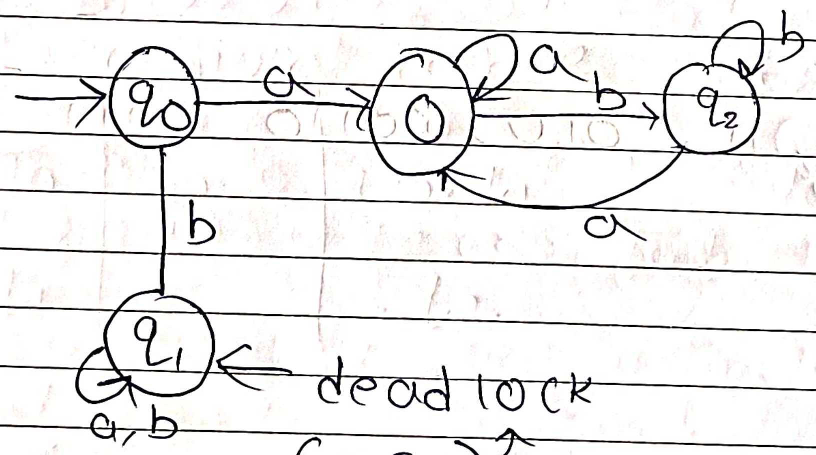
Q) Having ~~two~~ which containing two consecutive zero follow by 1

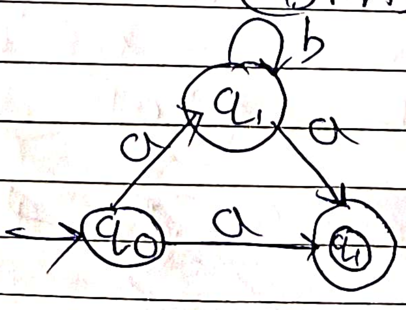
Ans $L = \{001, 10011, 000110, 10010011$



Q). Design a NFA for all string starting & ending with a.

Ans $L = \{a, aba, aaa, aaba, aa$





Transition Table (NFA):-

	a	b
q_0	$q_1 q_2$	-
q_1	-	-
q_2	$q_1 q_2$	q_2

Transition Table (DFA):-

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

$(q_1, a) \cup (q_2, b)$

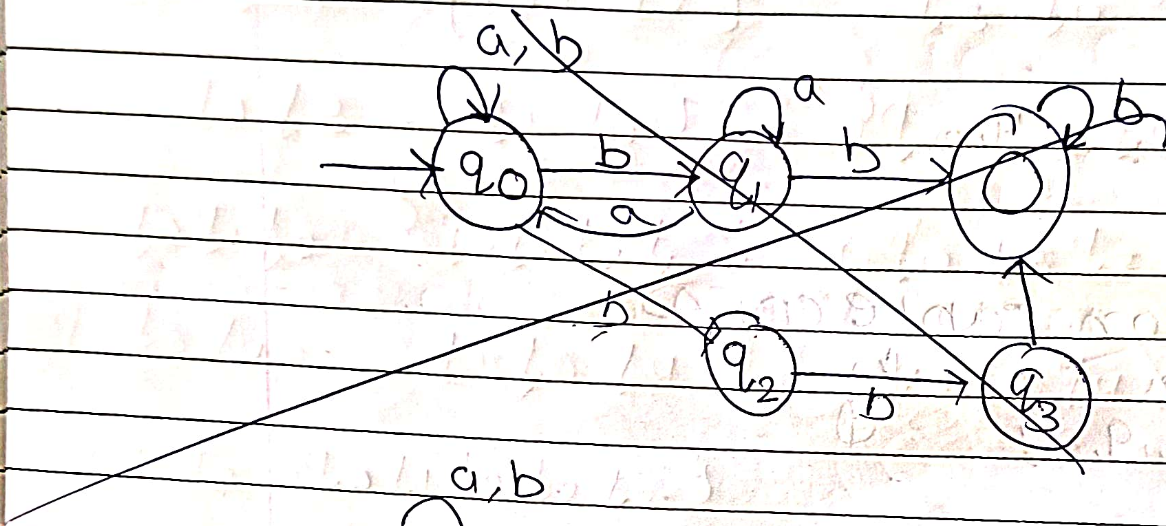
$\phi \cup q_1 q_2$

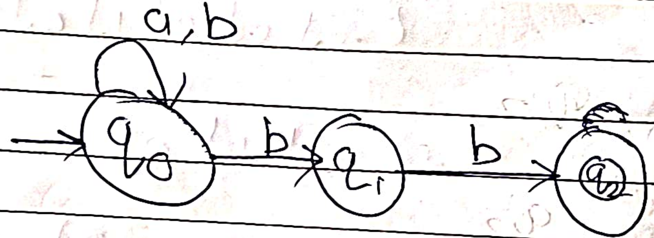
$(q_1 b) \cup (q_2 b)$

$\phi \cup q_2$

Q) Design a NFA for all string ending with bb.

Ans $L = \{abb, abb, bbb, aabb \ldots\}$





(NFA)

Transition table (NFA)

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

Transition Table (DFA)

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

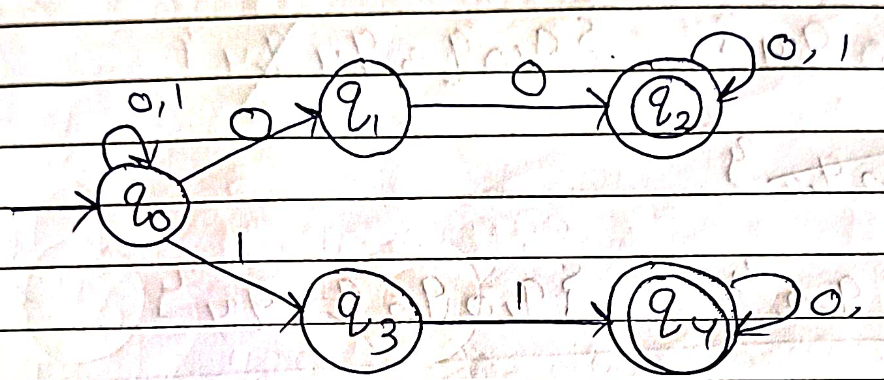
$(q_0, a) \cup (q_1, a)$
$q_0 \cup \phi$

Q_4
$(q_0, b) \cup (q_\phi, b)$
$\{q_0, q_1\} \cup \{q_2\}$

Starting and bb
→

Q) Design a NFA for a language that contains string which have at least two consecutive zeros or ones.

$L = \{00, 11, 0011,$ ~~10011~~$, 00100 \ldots\}$



Transition table (NFA)

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

$(q_0,0) \cup (q_1,0)$ | $(q_0,1) \cup (q_1,1)$
$q_0 q_1 \cup q_2$ | $q_3 q_0 \cup \phi$

Transition Table (DFA)

	0	1
q_0	$\{q_0, q_1\}$	$\{q_0, q_3\}$
q_0q_1	$\{q_0q_1q_2\}$	$\{q_3, q_0\}$
q_0q_3	$\{q_1, q_0\}$	$\{q_0q_3q_4\}$
$q_0q_1q_2$	$\{q_0q_1q_2\cancel{q_2}\}$	$\{q_0q_3q_2\}$
$q_0q_3q_4$	$\{q_1q_0q_4\}$	$\{q_0q_3q_4\cancel{q_4}\}$
$\cancel{q_0q_1q_2q_2}$	$\cancel{\{q_0q_1q_2q_2q_2\}}$	
$q_0q_3q_2$	$\{q_0q_1q_2\}$	$\{q_0q_2q_3q_4\}$

$(q_0,0) \cup (q_3,0)$ | $(q_0,1) \cup (q_3,1)$
$q_1 q_0 \cup \phi$ | $q_3 q_0 \cup q_4$

$(q_0,0) \cup (q_1,0) \cup (q_2,0)$ | $(q_0,1) \cup (q_1,1) \cup (q_2,1)$
$(q_0q_1) \cup q_2 \cup q_2$ | $q_0 q_3 \; q_2$

$(q_0,0) \cup (q_3,0) \cup (q_4,0)$ | $(q_0,1) \cup (q_3,1) \cup (q_4,1)$
$q_1 q_0 \; \phi \; q_4$ | $q_3 q_0 \; q_4 q_4$

$\cancel{(q_0,0) \cup (q_1,0) (q_2,0) \cup (q_2,0)}$
$\cancel{q_0 q_1 \; q_2 \; q_2 q_2}$

$(q_0,0) \cup (q_3,0) \cup (q_2,0)$ | $(q_0,1) \cup (q_3,1) \cup (q_2,1)$
$q_0 q_1 \; \phi \; q_2$ | $q_3 q_0 q_4 \; q_2$

$q_1 q_0 q_4$ $\quad q_0 q_1 q_2 q_4$ $\quad q_3 q_0 q_2 q_4$

$q_0 q_2 q_3 q_4$ $\quad q_0 q_1 q_2 q_4$ $\quad q_0 q_3 q_2 q_4$

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

$(q_1, 0) \cup (q_0, 0) \cup (q_4, 0)$ | $(q_0, 1) \cup (q_1, 1) \cup (q_4, 1)$
$q_2, q_0 q_1, q_1$ | $q_3 q_0 \quad q_2 \quad q_4$

$(q_0, 0)(q_2, 0)(q_3, 0)(q_4, 0)$ | $(q_0, 1)(q_2, \phi)(q_3, 0)(q_4, \phi)$
$q_0 q_1 \quad q_2 \quad \phi \quad q_1$ | $q_0 q_3 \quad q_2 \quad q_4 q_4$

$(q_0, 0)(q_1, 0)(q_2, 0)(q_4, 0)$ | $(q_0, 1)(q_1, 1)(q_2, 1)(q_4, 1)$
$q_0 q_1 \quad q_2 \quad q_2 \quad q_4$ | $\phi \quad q_3 q_0 q \quad q_2 \quad q_2$
$q_0 q_1 q_4 q_2$ | $q_0 q_2 q_3 q_4$

~~Q) string has even no. of a's & even no. of b's.~~

Mod :

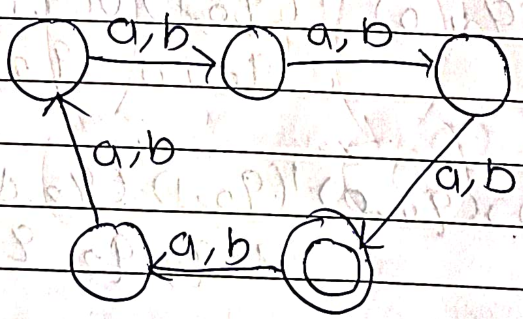
1) 3 mod 5

remainder

Remainder

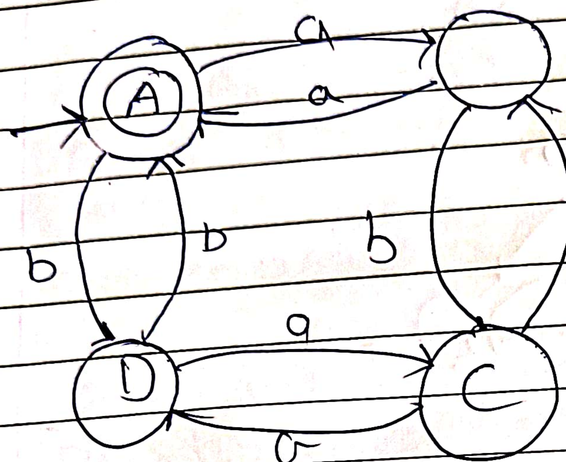
0, 1, 2, 3, 4

0	5	0
1	5	1
2	5	2
3	5	3
4	5	4
5	5	0
6	5	1
7	5	2
8	5	3
9	5	~~3~~ 4
10	5	0

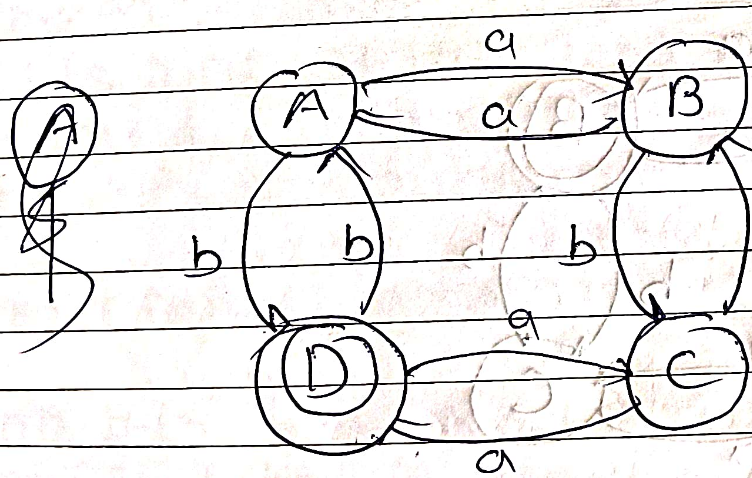


Q) String has even no. of a's & even no. of b's.

$\mathcal{L} = \{\epsilon, aa, bb, aabb, baba, aaaa, bbbb \ldots\}$



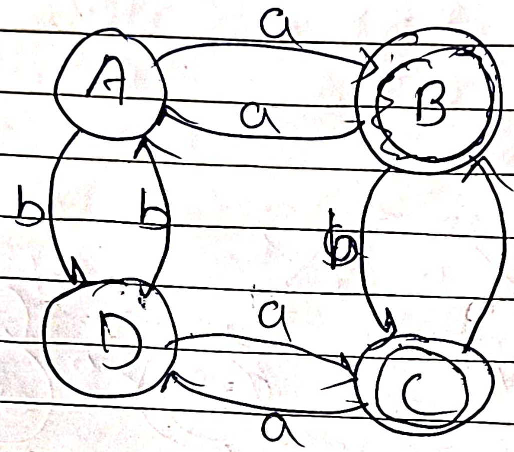
Q) even no. of a's & odd of b



$\mathcal{L} = \{aab, baa, bbbaa \ldots\}$

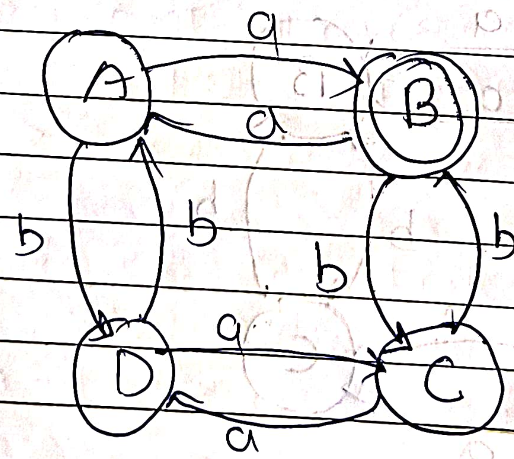
Q) Odd no. of a's & odd no of b

$\mathcal{L} = \{a, b, aaab, ababab \ldots\}$



Q) odd no. of a's & even no. of b

$\mathcal{L} = \{aaabb, a_3bb, ababa \ldots\}$



Minimization of DFA :-

→ Reduce Dead state
→ Unreachable state
→ Equal state

Non-Productive
These states do not act anything to the language accepting power to the machine

Dead State :-
It is the state from where we can't reach to the final state

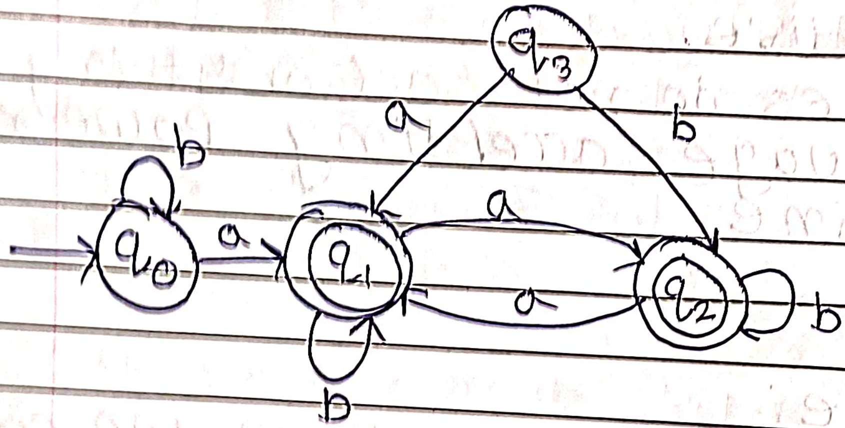
Unreachable state :-
It is that state which can not be reached from initial state by Passing any input string.

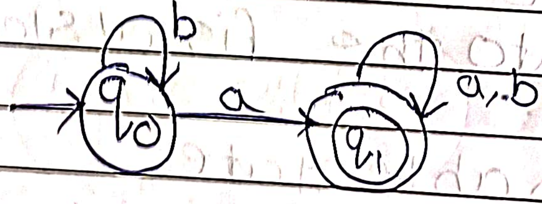
Procedure for ~~fing~~ ding equal state :-

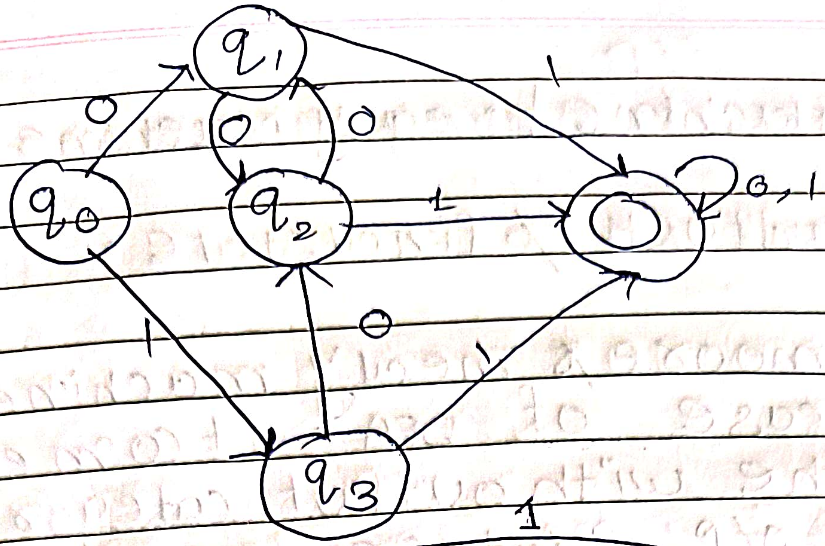
For this first of all group all the non-final ~~finite~~ state in one set and all final state in another state.

Now, on the both state individualy check whether any of the underline state of that Particular set are behaving

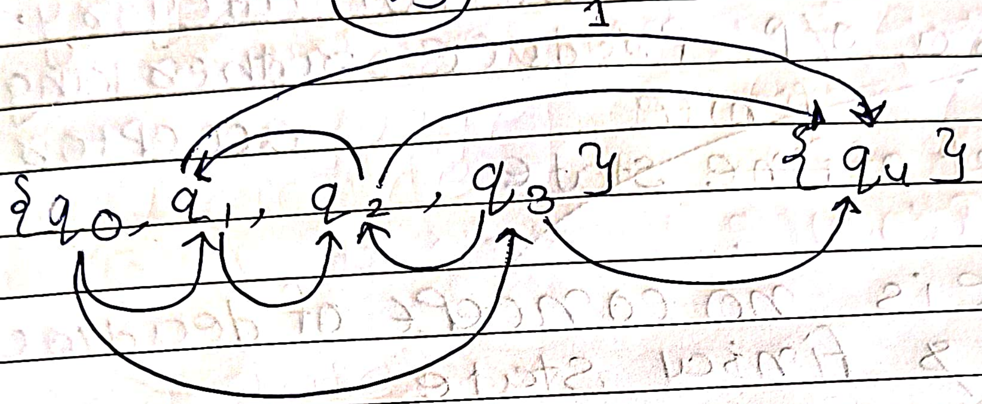
in the same way that is they are having same transition on each input alphabet.
If answer is 'yes' than these two states are equal otherwise not'.



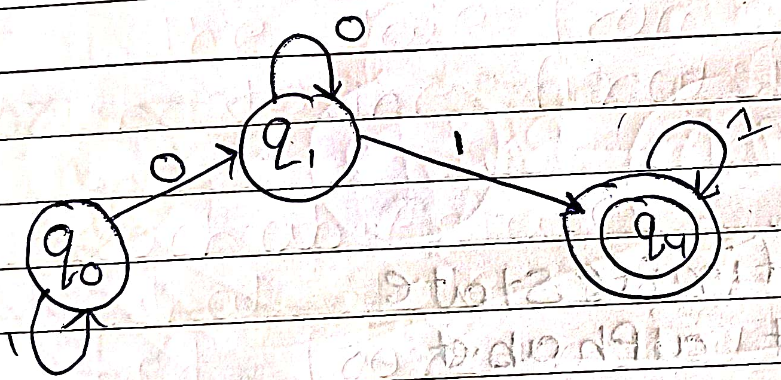




q1
q0
q2
q3
0
1
0,1



{q0, q1, q2, q3}
{q4}
1



q0
q1
q4
0
1

Moore Machine | Mealy Machine :-

→ No deadlock & final state

→ Both moore & mealy machine are special case of DFA. It comes under the with output category. Both act as a o/p produce rather language acceptor

~~Q → set of finite state~~

There is no concept of deadlock state & final state

Q → set of finite state
Σ → Input alphabet
δ → Transition function

q_0 → Initial state
Δ → output alphabet/symbol
λ → o/p function $\lambda : Q \Rightarrow \Delta$

~~Length~~

~~aaab~~ baa baa

→ q_0 0000

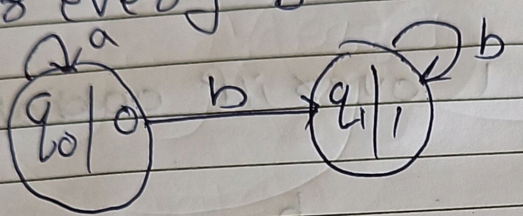
Lengthofan

aaabbaabaa
0 0 0 0 1 0 0 0 1 0 0
$q_1\ q_2\ q_2\ q_3\ q_0\ q_1\ q_2\ q_3\ q_1\ q_2$,

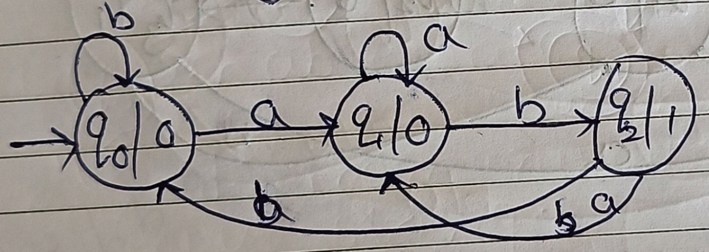
Moore :-
In moore machine if length of i/P string is 'n' than length of o/P string will be n+1.

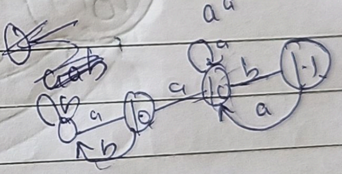
moore machines are a response for empty string (ε).

Q) For every 'b' ~~Prese~~ Print '1'



Q) For string ab Produce O/P '1'

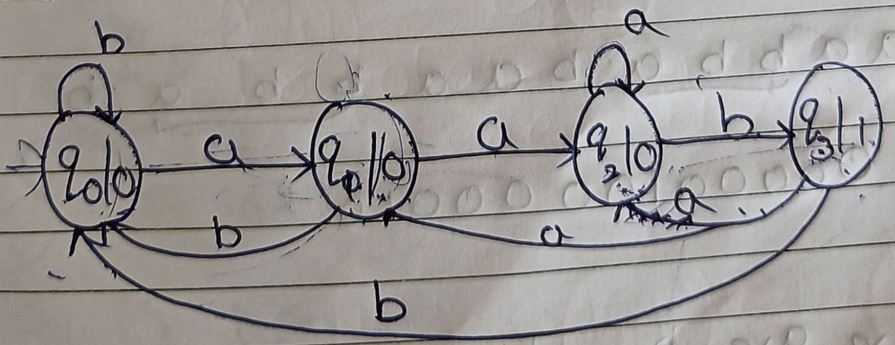




aab

aab a

Q) aab Produce 1



aab[b]

~~aagb~~

α = {aab, baab, aaab, aabaab

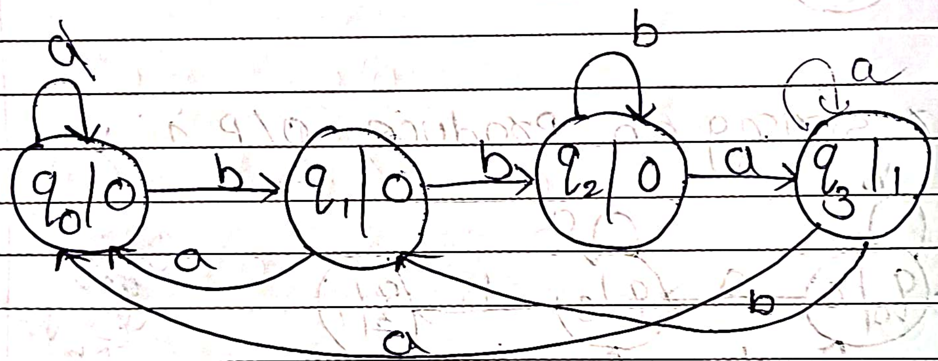
Q) If ~~length of~~ i/p string aaabbaabaa

aaabbaabaa

$\boxed{00001000100}$

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

Q) bba produce 1 over i/P aabbabaabbab



$L = \{bba, abbba, ababaabba \ldots \}$

a a b b a b a a b b a b

000001000010

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

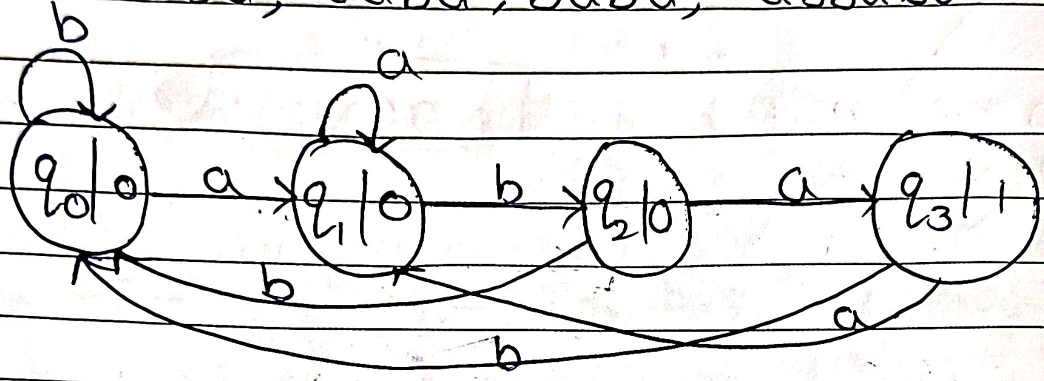
abaa

abab
abaa

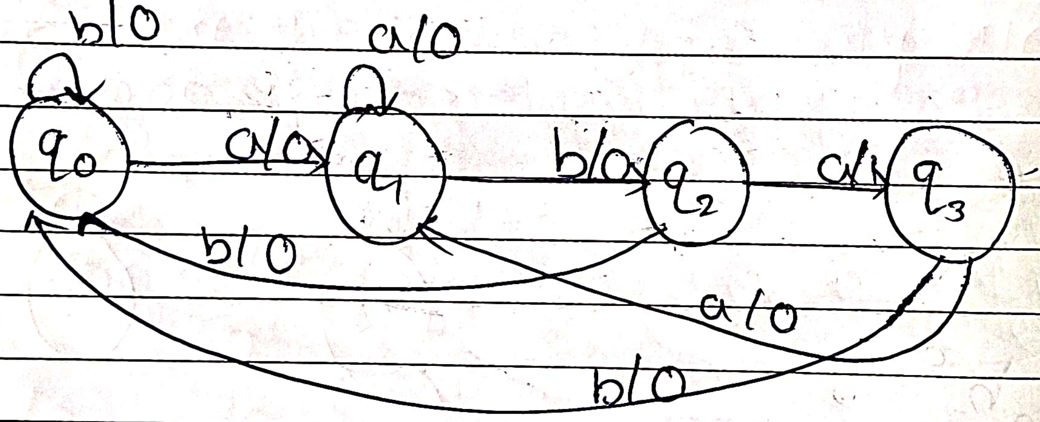
Q) aba:-

$L = \{aba, aaba, baba, abbaba \ldots\}$

Moore=

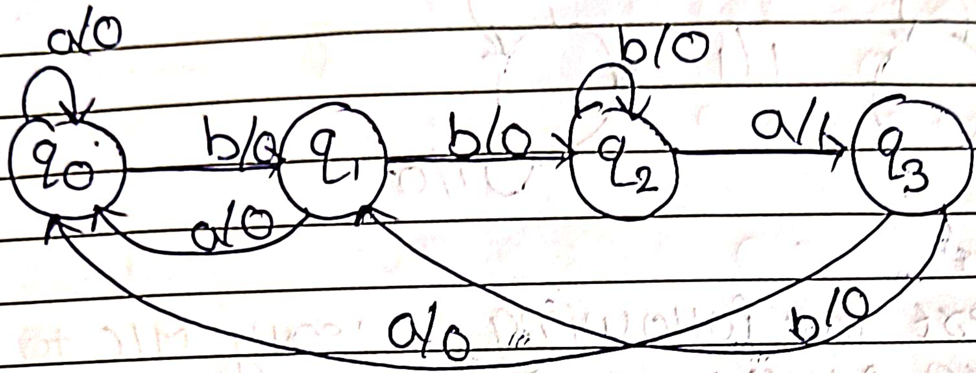


Mealy:-



Moore transition table:-

	a	b	O/P
q_0	q_0	q_1	0
q_1	q_0	q_2	0
q_2	q_3	q_2	0
q_3	q_0	q_1	1

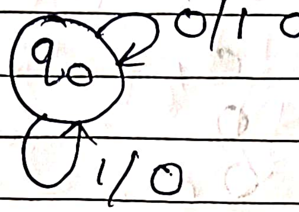


state	a	b	O/P
q_0	q_0	q_1	0
q_1	q_0	q_2	0
q_2	q_3	q_2	0
q_3	q_0	q_1	1

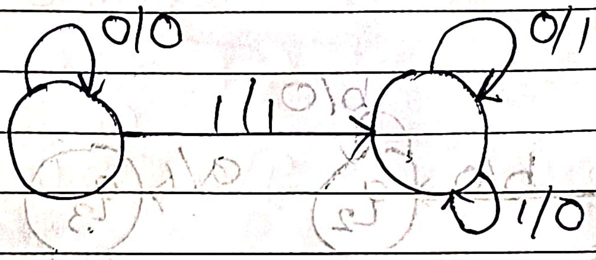
Mealy transition table:-

	a		b	
	state	O/P	state	O/P
q_0	q_0	0	q_1	0
q_1	q_0	0	q_2	0
q_2	q_3	1	q_2	0
q_3	q_0	0	q_1	0

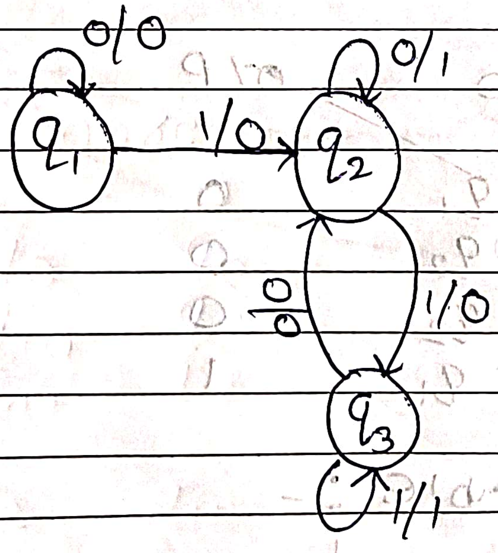
Q) 1's complement Mealy machine:



Q) 2's complement



Q) Convert the following Mealy M/C to moore Machine



~~Q~~

q_1

q_2

q_3

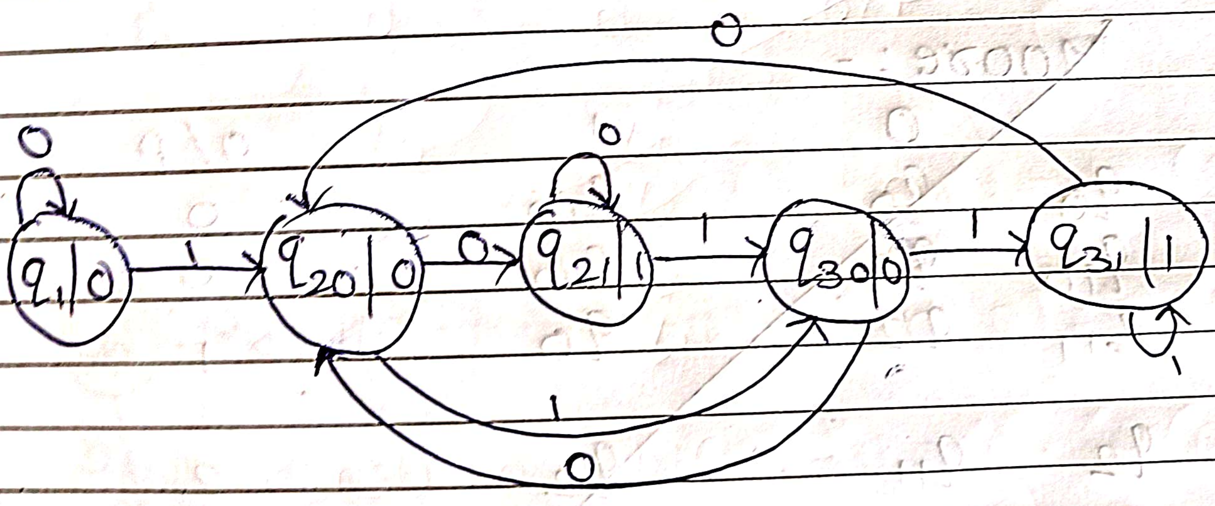
010011
+ 1

010100

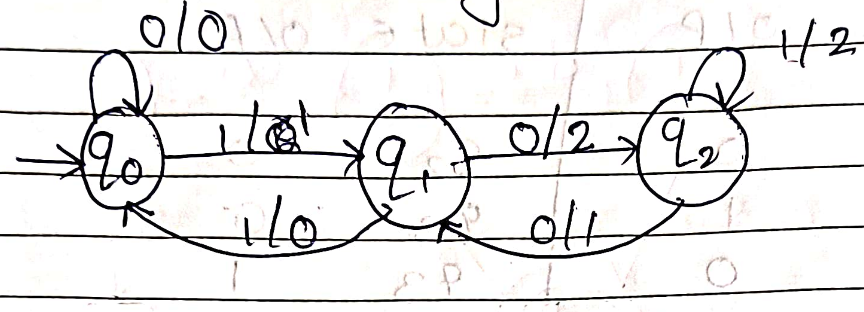
	0		1	
	state	O/P	state	O/P
q_0	q_1	0	q_2	0
q_2	q_2	1	q_3	0
q_3	q_2	0	q_3	1

<u>Moore</u>:

	0	1	O/P
q_1	q_1	q_{20}	0
q_{20}	q_{21}	q_{30}	0
q_{21}	q_{21}	q_{30}	1
q_{30}	q_{20}	q_{31}	0
q_{31}	q_{20}	q_{31}	1



Q) convert Mealy to moore

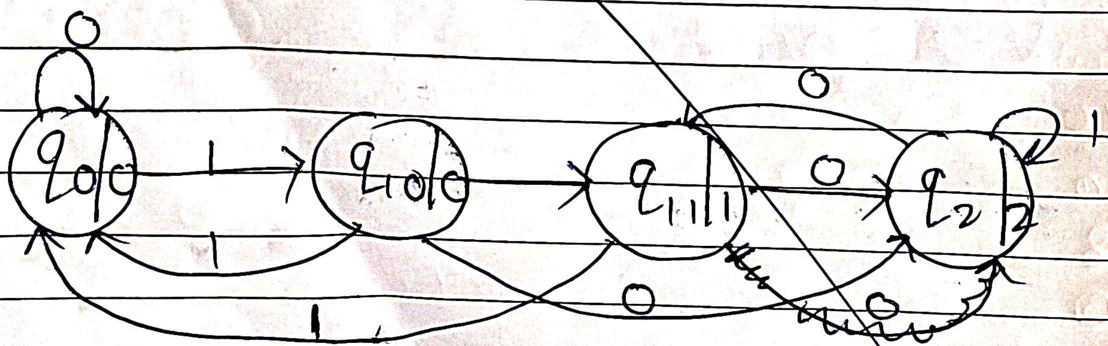


Transition table :-

	0 state	0 O/P	1 state	1 O/P
q0	q0	0	q1	1
q1	q2	2	q0	0
q2	q1	1	q2	2

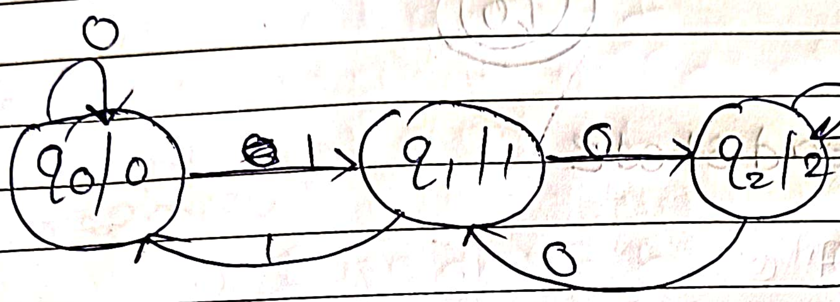
Moore :-

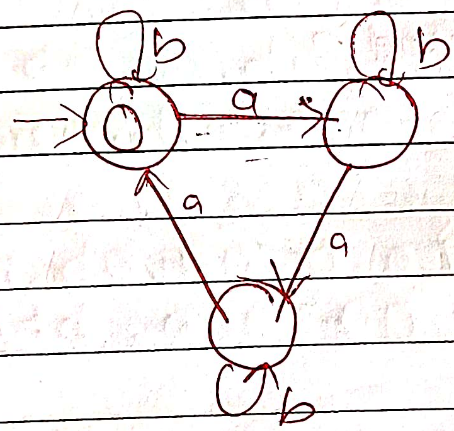
	0	1	O/P
q0	q0	q10	0
q10	q2	q0	0
q11	q2	q0	1
q2	q11	q2	2



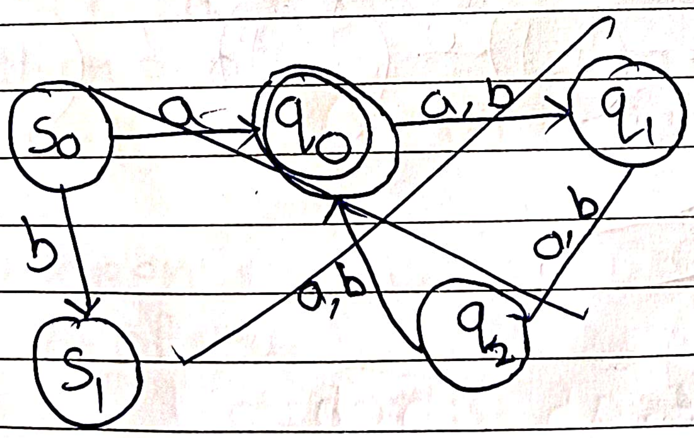
Moore

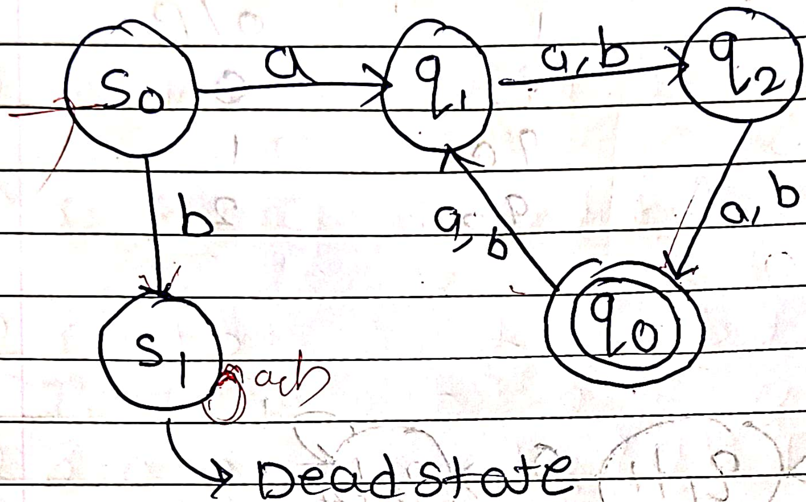
	0	1	O/P
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_1	q_2	2





0, 3, 6, 9





$|w|_a = 0 \bmod 3$

$|w|_b =$

Regular Expressions :-

① Regular expression is a convinient way to represent pattern of string.
② Regular expression is an algebric description of language.
③ It is a way of representing the regular language.
④ Regular expression is a notation to specify a language

Operation on regular expr :-

→ union (a+b) - (OR)
If a & b are two regular expr then their union is also regular expression (a+b) or (a ∪ b)

→ concatination (a·b) -
If a & b are two regular expr then their concatination will be (a·b)

→ kleen closure :- (a^*)
Set of all possible strings
Eg :- $a^* = \{\epsilon, a, aa, aaa \ldots\}$
Including 'ε'.

→ Positive closure :- (a^+)
~~set~~ Excluding 'ε'
Eg. $a^+ = \{a, aa, aaa \ldots\}$

$\epsilon = 1$

Indentities of RE :-

1) $r = \phi$, $L = \{\ \}$

2) $r = \epsilon$, $L = \{\epsilon\}$

3) $r = a+b$, $L = \{a, b\}$

4) $r = a \cdot b$ $L = \{ab\}$

5) $r = a(b+a)$ $L = \{ab, aa\}$

6) $r = (a+b)^*$ $L = \{\epsilon, ab, aa, ab, ba, bb, aaa, bbb, aba, bba \ldots\}$

7) $r = (a+b)^+$ $L = \{a, b, aa, ab, ba, bb, aa, aaa, bbb, aba, bba \ldots$

8) $(b+\epsilon)(a+\phi)$ $L = \{ba, a\}$

9) $r = (a+b)^2 = (a+b)(a+b)$, $L = \{aa, bb, ab, ba\}$

10) $a^* \cdot a^* = a^*$

11) $r = (ab)^* = \{\epsilon, ab, abab \ldots\}$

12) $\phi + R = R$

13) $\phi R + R\phi = \phi = \phi$

14) $\epsilon \cdot R = R \cdot \epsilon = R$

15) $\varepsilon^* = \varepsilon$

16) $\varepsilon^* = \varepsilon$

17) $\phi^* = \epsilon$

18) $R^* R^* = R^*$

19) $R R^* = R^* R$

20) $\epsilon + R R^* = \epsilon + R^* R = R^*$

21) $(P+Q)^* = (P^* + Q^*)^*$

22) $(PQ)^* P = P(QP)^*$

Q) Design a regular expr which can represent all string start with.

$\Rightarrow$ ~~(ab) a+b~~
~~(ab)*~~
$ab(a+b)^*$

$(a+b)^* ba$ → ending with ba.

Q) starting & ending with a.

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

Q) starting & ending with same symbol

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

Q) start & end with diff^{r} symbol

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