Deterministic Finite Automata (DFA)

The Finite Automata is called "Deterministic Finite Automata", if there is only one path for a Specific input from current State to Next State.

* DFA can be as below.

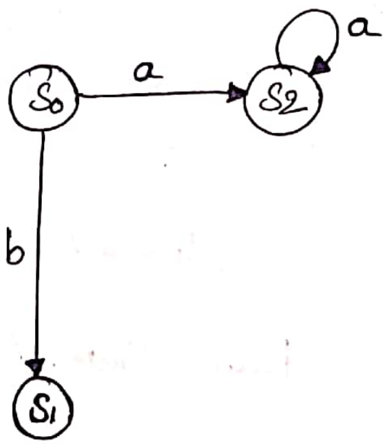


fig: Deterministic Finite Automata (DFA)

* State 'S_0' for input 'a' there is only one paths going to S_2.
* From 'S_0' there is only one path for input 'b' going to 'S_1'

DFA can also be represented by the 5-tuples as described in FSM (Finite State Machine).

DFA also be defined as below

"There is no more than one transition on a particular input symbol."

Scanned by CamScanner

DFA is collection of following things

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

Q = Finite set of states (or) Total no. of states.

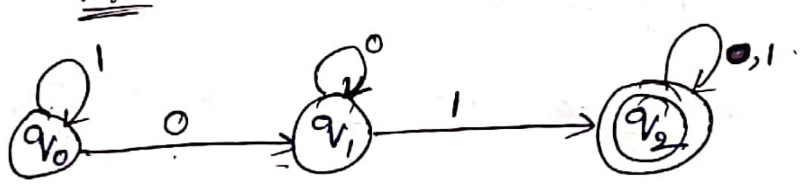
Σ = Finite set of input symbols.

δ = Transition function (or) Mapping Function also denoted by $\delta : Q \times \Sigma \longrightarrow Q$. called as "Next Straight function".

q_0 = Initial State. $(q_0 \in Q)$

F = A set of Final State. $(F \in Q)$.

Example: DFA



$Q = \{q_0, q_1, q_2\}$

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

$\delta : (q_0, 0) \longrightarrow q_1$

$(q_0, 1) \longrightarrow q_0$

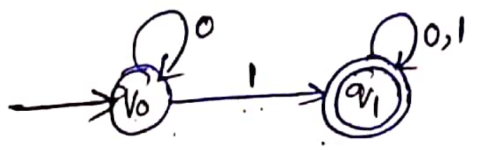
$\delta(q_1, 0) \longrightarrow q_1$

$\delta(q_1, 1) \longrightarrow q_2$; $\delta(q_2, 1) = q_2$

$\therefore F = \{q_2\}$

Scanned by CamScanner

Example ②: NOT DFA Problem – NFA



$Q = \{q_0, q_1\}$

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

We can say, it is not DFA.

$$\delta = \begin{cases} \delta(q_0, 0) = q_0 \\ \delta(q_0, 1) = q_1 \\ \delta(q_1, 0) = q_1 \\ \delta(q_1, 1) = q_\phi \end{cases}$$

Because

$q_0 \xrightarrow{0} q_0$

$q_0 \xrightarrow{1} q_1$

$F = \{q_1\}$.

∴ only one i/p & one o/p statements are forms.

But for q_1

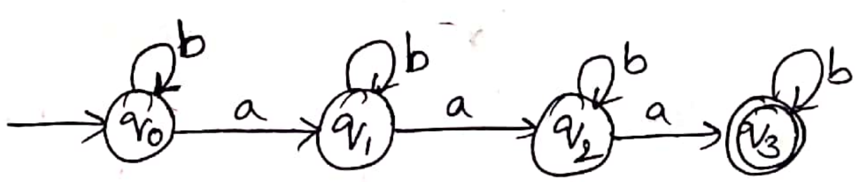
$$\boxed{\begin{array}{l} q_1 \xrightarrow{0} q_1 \\ q_1 \xrightarrow{1} q_1 \end{array}}$$

Here, for one i/p, there is '2'-o/p's are forms.

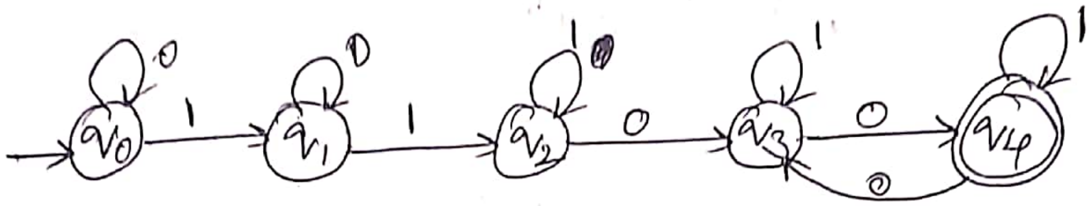
So the above FA Not a "DFA"

Problem

① Design DFA for the string aaa, for $\{a, b\}$



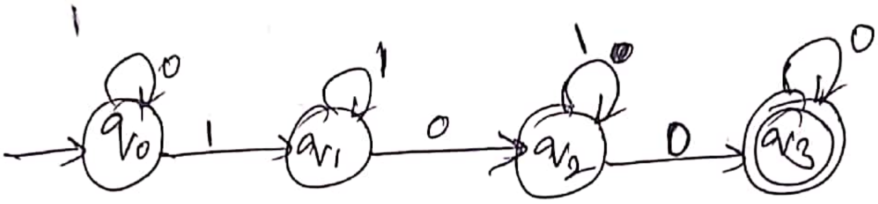
② Design DFA for 1100 as Substring.



11001100

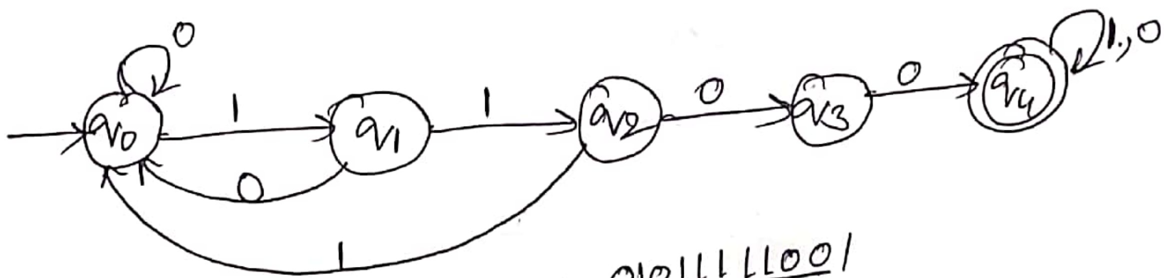
L=011001

(or)



L= 0011001110

(or)



→ 0101111001

→ 0111001

③ Design the DFA for even no. of Zero's & any No. of 1's ?

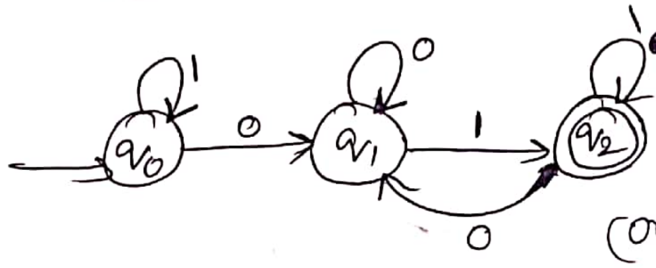
⇒ 1010, 11010, 10110
even No. of 0's
any No. of 1's

(or)

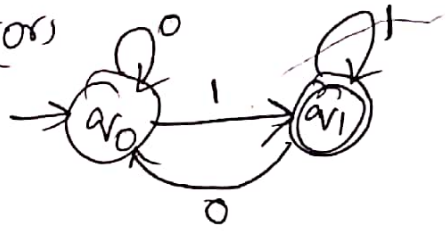
Scanned by CamScanner

4. Construct the DFA that accept the string containing that followed by one?

<u>Soln</u>



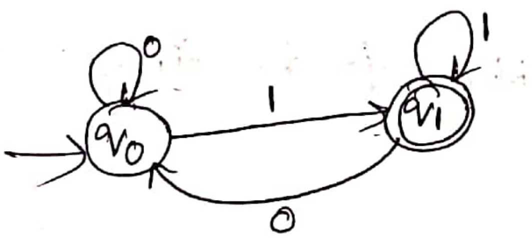
(or)



⇒ 1001

⇒ 110001

5. Design DFA for binary odd Numbers.



⇒ 001 = 1

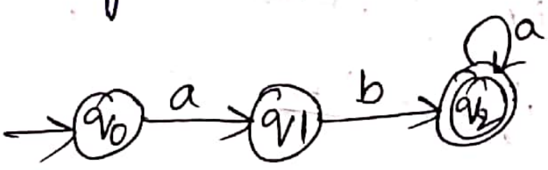
⇒ 011 ⇒ 3

⇒ 101 ⇒ 5

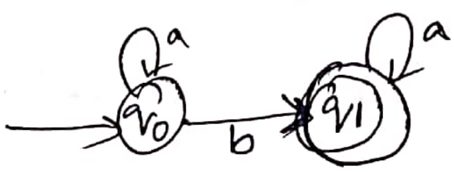
⇒ 111 ⇒ 7

⇒ 1111 = 15

6. Design DFA for the substring "aba"

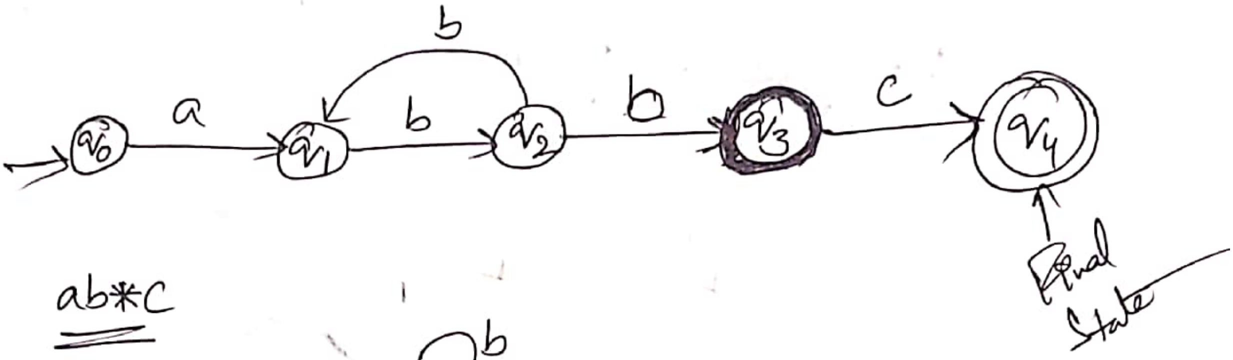


(or)

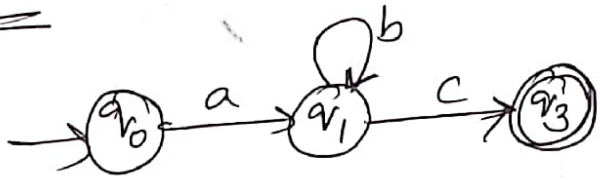


Scanned by CamScanner

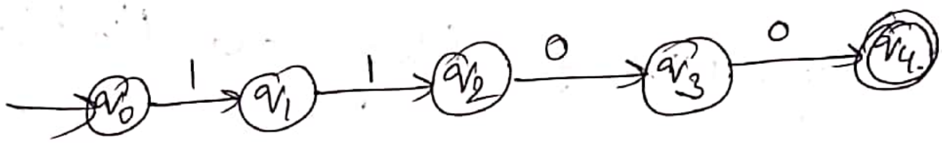
(7) DFA for $a(bb)^*bc$.



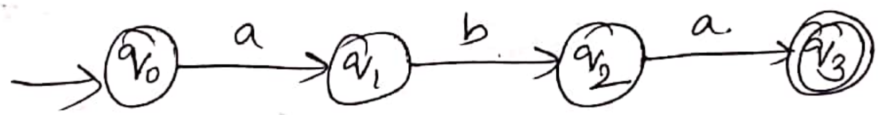
ab^*c



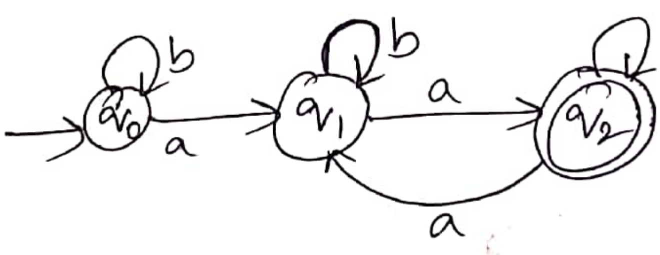
(8) Draw DFA it accepted only 1100?



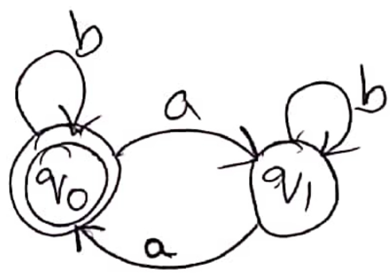
(9) Design the DFA for the string "aba"?

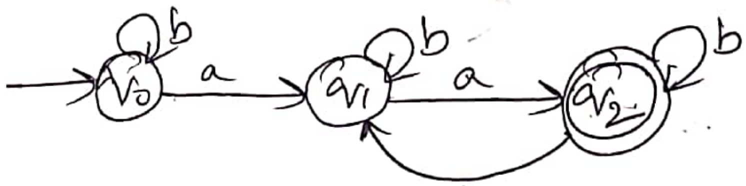


(10) Design DFA for, which accepts even no. of a's over the alphabet "b"



(or)

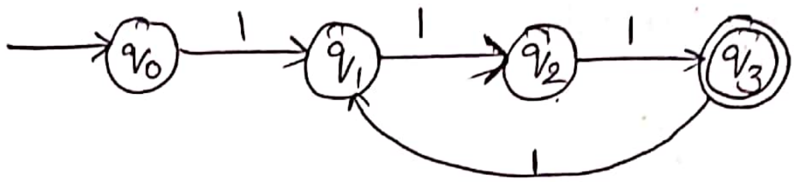




Check: baabbaaaba

Scanned by CamScanner

Design DFA which checks whether the given Unary Number is divisible by '3'



"Unary Number" made up of ones.

No.	Unary No.
3	111
6	111111
9	111111111

} divisible by '3'.

Transition Table :-

Input	1
$\rightarrow q_0$	q_1
q_1	q_2
q_2	q_3
q_3	q_1

we will check now, the 111111 = 6 is divisible by '3' and accepting by the above DFA (or) Not.

Will start with "111111".

The String acceptance Can be Check as below.

Scanned by CamScanner

Start 111111

$\delta(q_0, 111111)$

$\delta(q_1, 11111)$

$\delta(q_2, 1111)$

$\delta(q_3, 111)$

$\delta(q_1, 11)$

$\delta(q_2, 1)$

$\delta(q_3)$. → final State.

We are reaching to Final. So, that the String accepting by DFA

Design DFA which Checks the given binary Number is divisible by three?

Sol: Input Number is "Binary Number".

$\therefore \Sigma = \{0, 1\}$ ⇒ Input Set.

Start State = S:

→ Remainder '0' is by "S_0"

" '1' is by "S_1"

'2' is by "S_2"

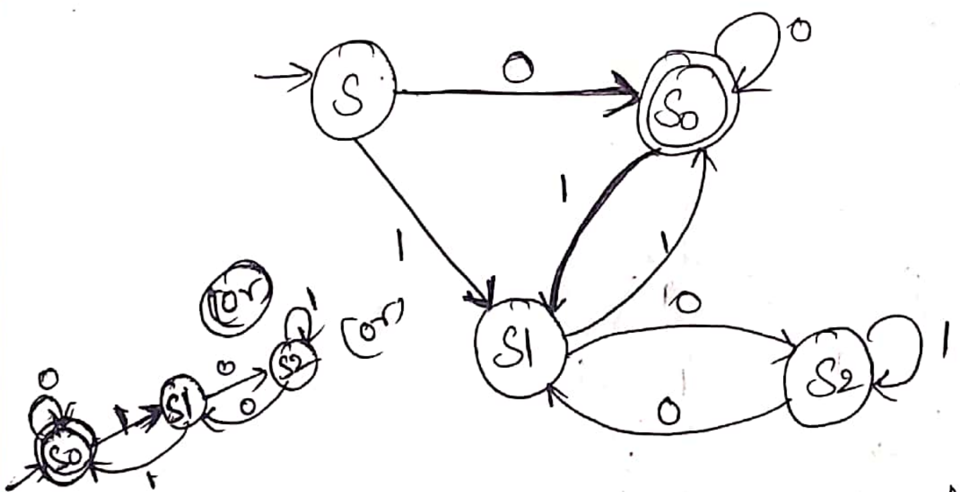
Scanned by CamScanner

Example Strings Consider in binary format like

→ 010 = 2/3 = 2 Remainder.
↑ ↑ ↑
S0 S1 S2

→ 1 0 0 1 ⇒ 9/3 = 0 Remainder
↑ ↑ ↑ ↑
S_1 S_2 S_1 S_0

Let's see the DFA for that



Transition Table

Input	0	1
→S	So	S1
So	So	S1
S1	S2	So
S2	S1	S2

Consider the string, which is divisible by '3' as 10101
21

∴ 10101

δ(S, 10101)

δ(S1, 0101)

δ(S2, 101)

δ(S2, 01)

δ(S1, 1)

δ = S0 ⇒ Final State

Scanned by CamScanner

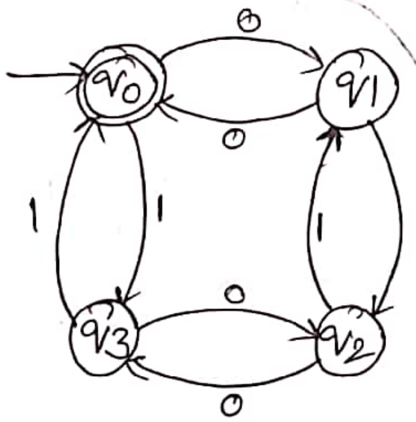
Q: Design DFA which accepts even No. of 0's & even No. of 1's?

Sol. The DFA will Consider '4' Stages for Input '0' & Input '1'

The possible Cases & Impossible Cases are as below.

0's	1's
Even	Odd
odd	even
odd	odd
even	even ✓ possible Case.

Let's try to design the Machine.



$Q = \{q_0, q_1, q_2, q_3\}$

$F = q_0$

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

q_0 : Even-0's ; ~~odd~~ Even-1's

q_1 : odd No. of 0's ; Even No. of 1's

q_2 : odd No. of 0's ; odd No. of 1's

q_3 : even No. of 1's & odd No. of 1's.

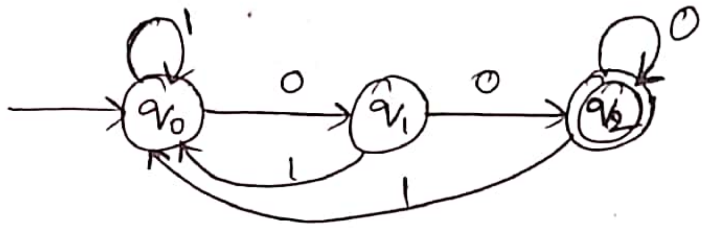
Transition Table

Input	0	1
→ q_0	q_1	q_3
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_2	q_0

Scanned by CamScanner

Q: Design DFA to accept the String that always ends with 00.

Solⁿ



String Acceptance

Ex: 01001100

$\delta(q_0, 01001100)$

$\delta(q_1, 1001100)$

$\delta(q_0, 001100)$

$\delta(q_1, 01100)$

$\delta(q_2, 1100)$

$\delta(q_0, 100)$

$\delta(q_0, 00)$

$\delta(q_1, 0)$

$= q_2 \longrightarrow$ Final State.

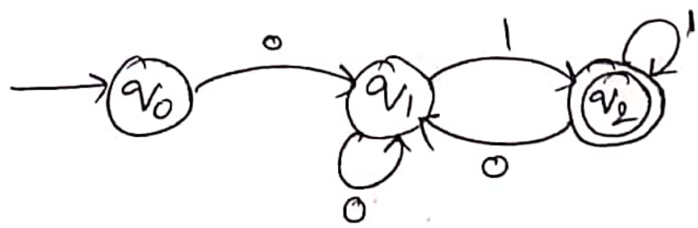
$\because$ 'q_2' is Final State, Hence the i/p accepted.

Transition Table

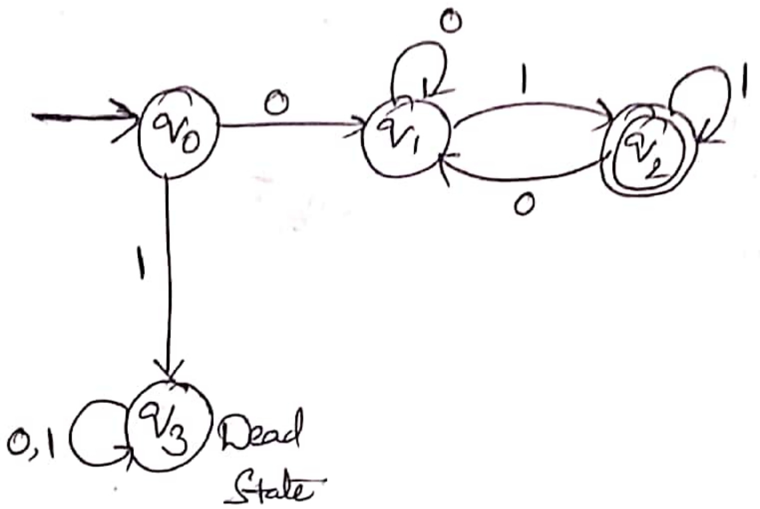
Input	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_0

Q: Construct the transition graph for DFA, which accepts a language over $\Sigma\{0, 1\}$ in which every String Start with '0' and ends with '1'.

Solⁿ



(or)



* If the Input Starts with 1, then it ~~is~~ will be in "q_3". State which is dead State and never lead to final State.

Thus the Machine Strictly handles the Strings Staring with 'o' and ending with '1'

Q: Design DFA to accept L, where L = { String in which 'a' always appears trippled } over the set $\Sigma = \{a, b\}$

Given L = { String in which 'a' always appears trippled} over the set $\Sigma = \{a, b\}$

* Ex: aaab, baaaaa, bbaaab and so on.
* The ~~a~~ 'a' always appears in a clump of '3'.
* The Transition graph (or) DFA will be look like as below.

Scanned by CamScanner

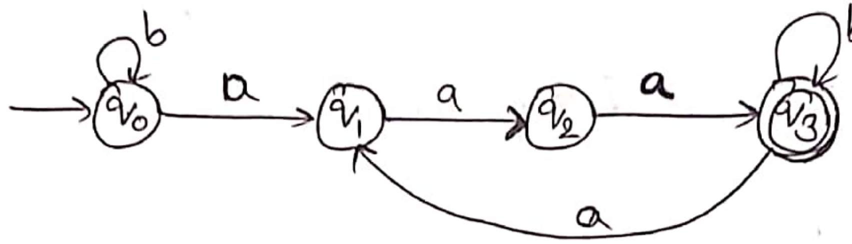


Fig: DFA.

Q: Design DFA to accept 'L', where all the strings in 'L' are such that total no. of a's in them are divisible by '3'

Sol: While testing divisibility by '3', grap the input as remainder 0, 1, 2.

S_0: State of Remainder 0

S_1: State of Remainder 1

S_2: State of Remainder 2

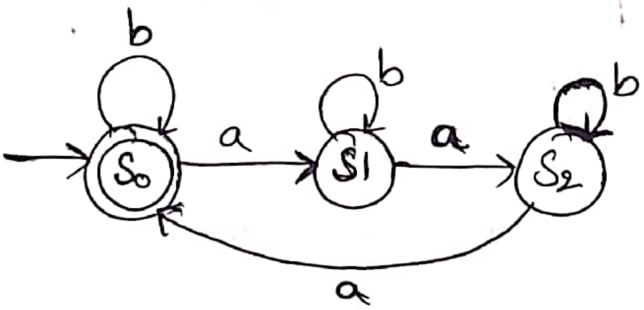


Fig: DFA.

Rem	No. of a's	Count
✓0	Nill	0
1	a	1
2	aa	2
0	aaa	3
1	aaaa	4
2	aaaaa	5
0	aaaaaa	6
⋮	⋮	⋮
0.	aaaaaaaaa	9

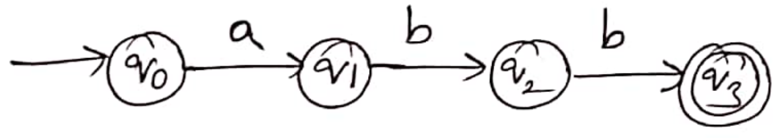
Example strings

1) ababab
2) abbbabba
3) aaaaaa
4) aabbba

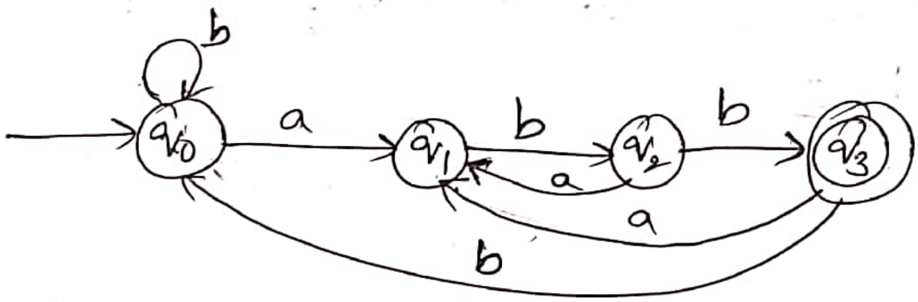
Scanned by CamScanner

Q: Design DFA to accept the strings of a's & b's ending with 'abb' over $\Sigma = \{a, b\}$.

Sol: Step①:



Step②



Check the strings

① babb

② bbbabb

③ bbabbbabb

④ bababb

⑤ abbabb

⑥ abbbabb.

Q: DFA to accept odd & even numbers represented using binary notations.

Sol:

Binary Number that ends with '0' – even Number.

Binary Number that end with '1' – odd Number

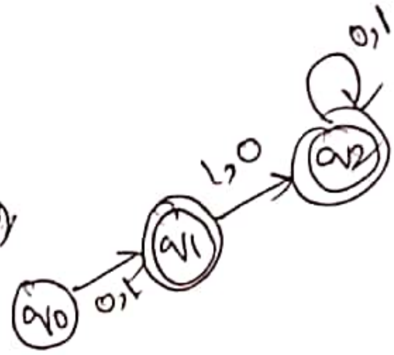
Ex:

E: 00<u>0</u> – 0	E – 10<u>0</u> – 4
001 – 1 – odd	101 – 5 – odd
E: 01<u>0</u> – 2	E – 11<u>0</u> – 6
011 – 3 – odd	111 – 7 – odd

Scanned by CamScanner

DFA :

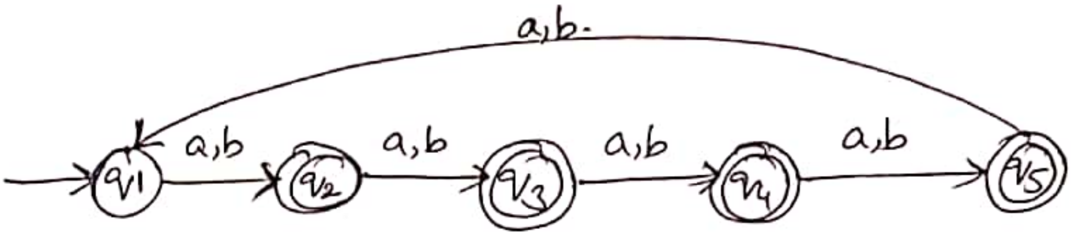
(or)



(or)

Q: Write DFA to accept the language $L = \{L : |w| \bmod 5 \neq 0\}$

Sol: The string which we obtain should not be divisible by '5'. Hence the DFA is —



Transistion Table:

Input	a	b
$\rightarrow q_1$	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_4
q_4	q_5	q_5
q_5	q_1	q_1

String acceptance Checking

a) abbb

$\delta(q_1, abbb)$

$\delta(q_2, bbb)$

$\delta(q_3, bb)$

$\delta(q_4, b)$

$q_5 \rightarrow$ Final State

Scanned by CamScanner

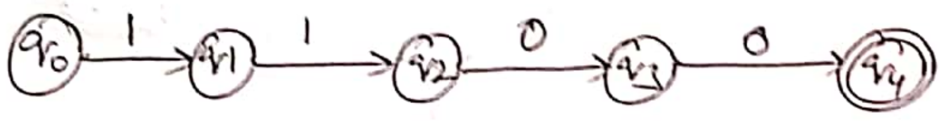
NFA (Non-Deterministic Finite-Automata).

(1)  $\Rightarrow (0+1)^* = (0/1)^*$

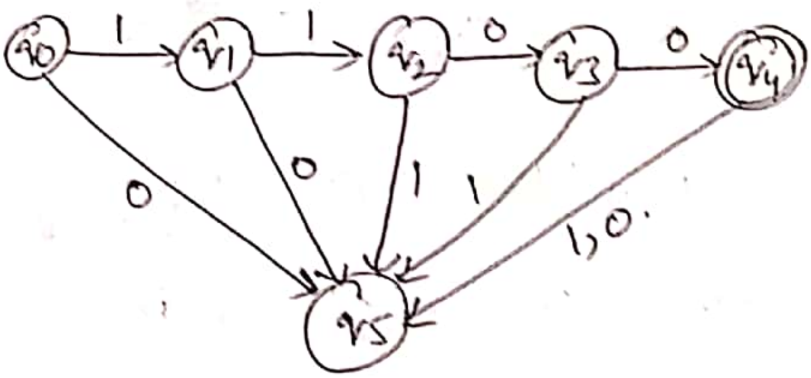
(2) $(0+1)^* 00 (0+1)^*$

(3) Design NFA which accepts only '1100'

NFA:



DFA:

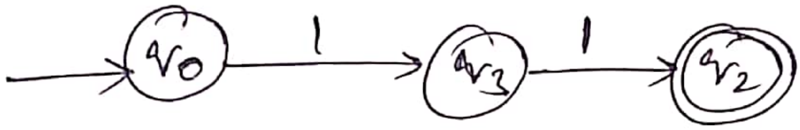


(4) Design NFA that accepts the strings 0's & 1's such that the string will contain 2 consecutive 0's & 1's

Sol: Step 1: Consecutive 0's

Scanned by CamScanner

Step2: <u>Consequtive 1's</u>

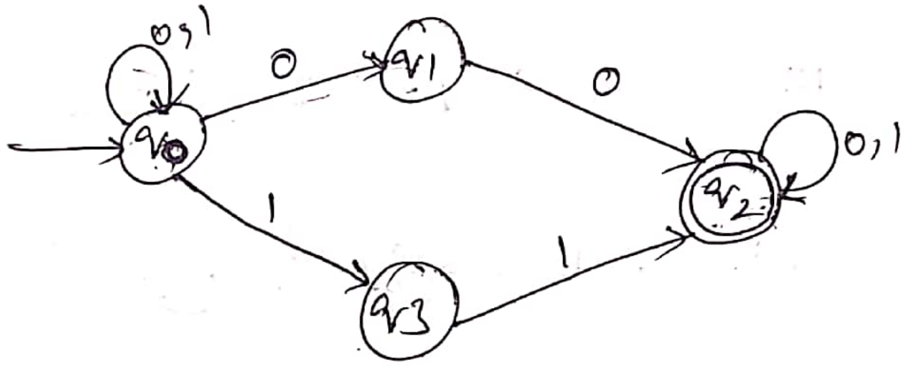


Step③: Two Consequtive 0's or 1's

00

0010

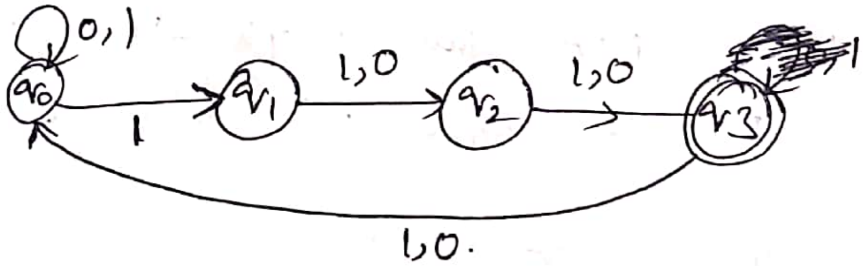
00111
100



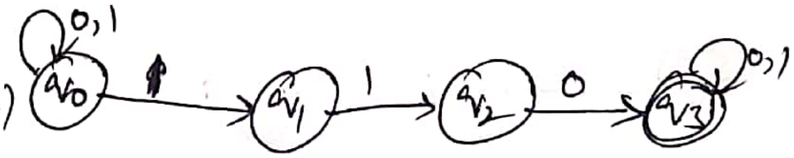
$(0+1)^* (00+11) (0+1)^*$ //.

⑤ Design NFA, that accept set of strings & containing 3rd symbol from right side is one(1)

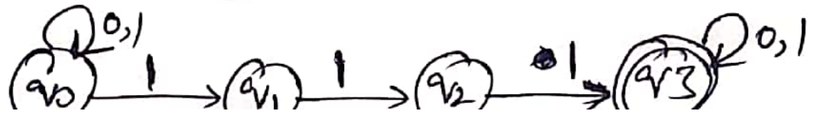
Soln



⑥ Design NFA that accept 1100, 1110 as a substring?

a)


⇒ 1100

b)


⇒ 111.

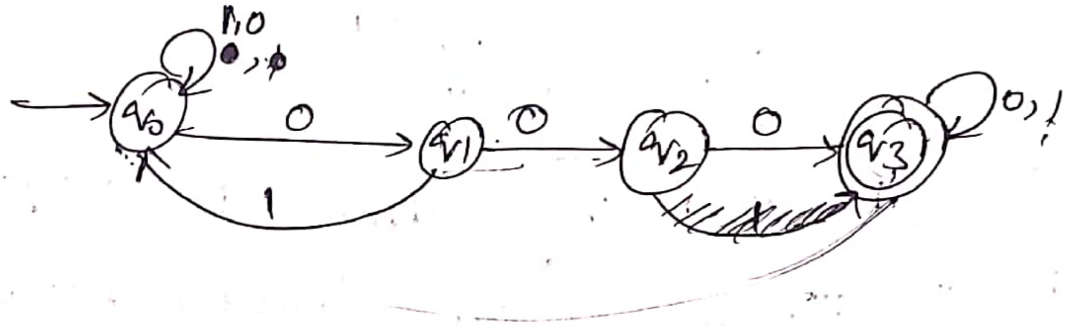
Scanned by CamScanner

(9) Design NFA Consists of the String Contains '3' Consequtive zeros

NFA

DFA:

a) DFA Can be as below.

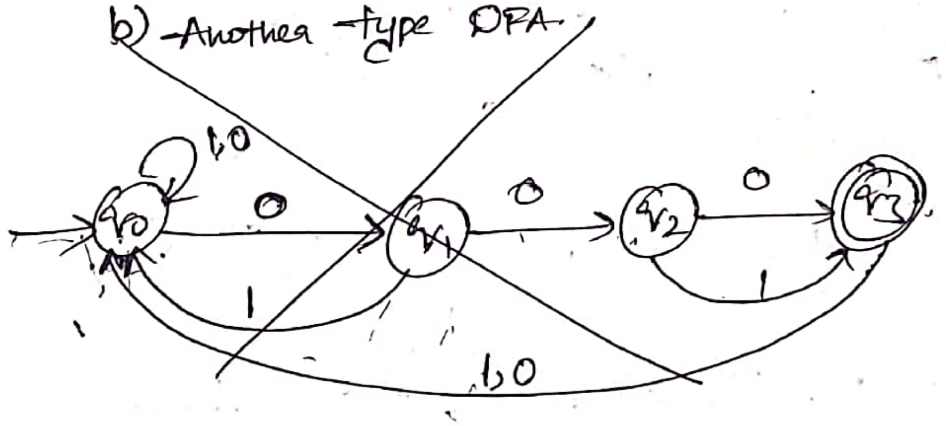


000

(or) 0101000101 0

001

b) Another type DFA

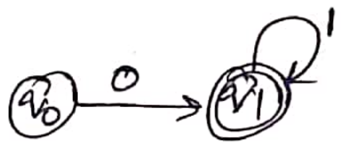


(8) Design NFA For language $L(M) = 01^* / 0^*1$

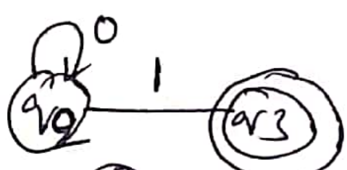
Sol: '/' Indicates '+'

$*$ – Repetitions

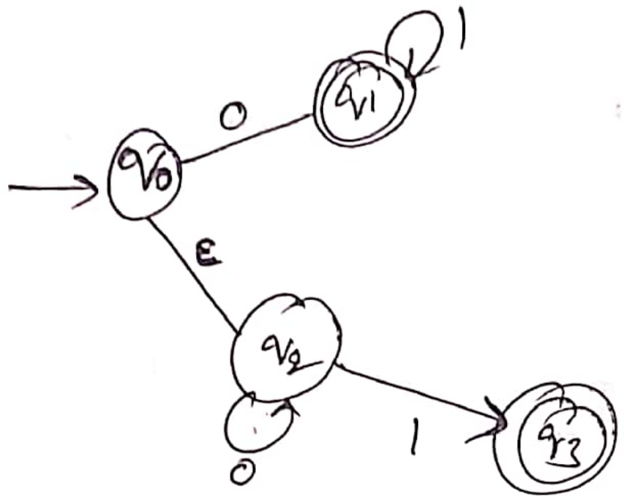
So $\Rightarrow 01^* \Rightarrow$



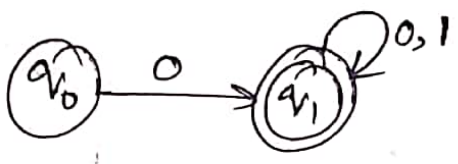
$\Rightarrow 0^*1 \Rightarrow$



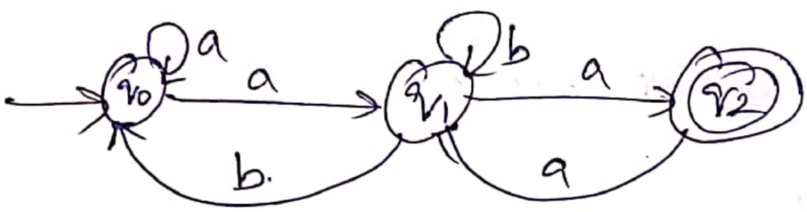
Scanned by CamScanner



$L(M) = 0(1+0)^*$



(9) Findout the String abbaba is accepted or not by the NFA for a given transistion Diagram



Soln: Given abbaba   Method(1)

$\delta(q_0, abbaba) \Rightarrow \delta(\delta(q_0, a), bbaba)$

$\Rightarrow \delta(q_1, bbaba)$

$\Rightarrow \delta(\underline{\delta(q_1, b)}, baba)$

$\Rightarrow \delta(q_1, baba)$

$\Rightarrow \delta(\underline{\delta(q_1, b)}, aba)$

$\delta(q_0, aba) \Rightarrow \delta(\underline{\delta(q_0, a)}, ba)$

$\Rightarrow \delta(q_1, ba) \Rightarrow \delta(\underline{\delta(q_1, b)}, a)$

$\Rightarrow \delta(q_1, a) = q_2, \xi$

$\therefore q_2'$ reached. Its Final State string accepted.

Scanned by CamScanner

Method (2):  $\Rightarrow$ abbaba

$\rightarrow \delta(q_0, \epsilon) = q_0$

$\rightarrow \hat{\delta}(q_0, a) = \delta(\hat{\delta}(q_0, \epsilon), a) \Rightarrow \delta(q_0, a) \Rightarrow \{q_0, q_1\}.$

$\rightarrow \hat{\delta}(q_0, ab) = \delta((q_0, q_1), b)$

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

$= \phi \cup \{q_1, q_0\}.$

$\therefore \boxed{\hat{\delta}(q_0, ab) = \{q_0, q_1\}}$

$\rightarrow \hat{\delta}(q_0, abb) = \hat{\delta}((q_0, q_1), b)$

$\hat{\delta}(q_0, abb) = \{q_0, q_1\}.$

$\rightarrow \hat{\delta}(q_0, abba) = \hat{\delta}((q_0, q_1), a)$

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

$= \{q_0, q_1\} \cup \{q_2\}$

$= \{q_0, q_1, q_2\}$

$\rightarrow \hat{\delta}(q_0, abbab) \Rightarrow \delta(\{q_0, q_1, q_2), b)$

$\Rightarrow \delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b)$

$\Rightarrow \phi \cup \{q_0, q_1\} \cup \phi \Rightarrow \{q_0, q_1\}$

Scanned by CamScanner

$\rightarrow \hat{\delta}(q_0, \underline{abbaba}) = \delta((q_0, q_1), a)$

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

$= \delta\{q_0, q_1\} \cup \{q_2\}$

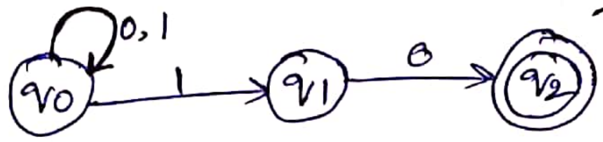
$= \{q_0, q_1, q_2\}$

$\therefore$ Thus the String accepted by NFA.

Scanned by CamScanner

Q(1): The given string '0100' is accepted or not for a given Diagram.

→ NFA.



Soln

$Q = \{q_0, q_1, q_2\}$

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

$\delta : Q \times \Sigma \rightarrow 2^Q$.

$F = q_2$.

Acceptance of string

$\delta(q_0, 0100)$

$= \delta(q_0, 100)$

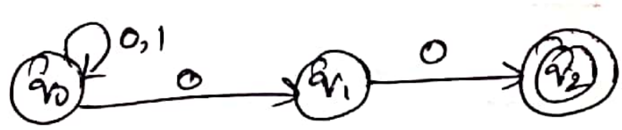
$= \delta(q_1, 00)$

$= \delta(q_2, 0)$.

From q_2 there is no transition with 'zero' again.

∴ So, the '0100' String is not accepting by the given NFA.

If the above NFA is like below.



Acceptance of '0100'

⇒ Path

↓ we are reaching Final State so its accepted.

Scanned by CamScanner

Q(2) Construct NFA for a language

L1 → { Consisting a Substring 0101 }

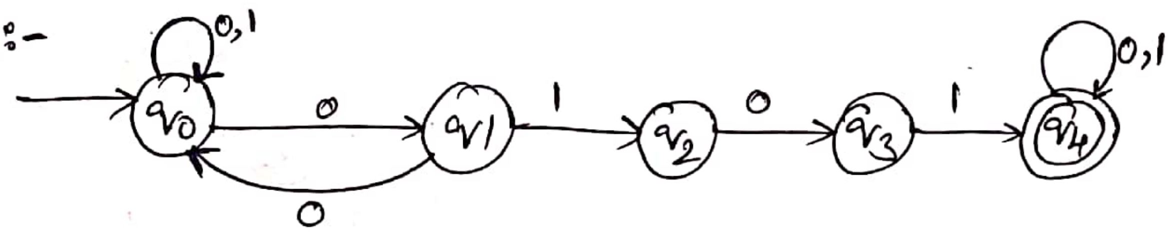
L2 → $\{a^n \cup b^n\}$.

Sol: * Consider 'L1' to design NFA.

* There can be any combination of 0 & 1 in the language but a substring 0101 must be present.

* At last we should get such a substring, that leads to final state or accept state.

L1 :-



String Acceptance:

Ex: 00010101
Lsubstring.

$\delta(q_0, 00010101)$

$\delta(q_0, 0010101)$

$\delta(q_1, 010101)$

$\delta(q_0, 10101)$

$\delta(q_0, 0101)$

$\delta(q_1, 101)$

$\delta(q_2, 01)$

$\delta(q_3, 1) \Rightarrow q_4$ // → Final State

Scanned by CamScanner

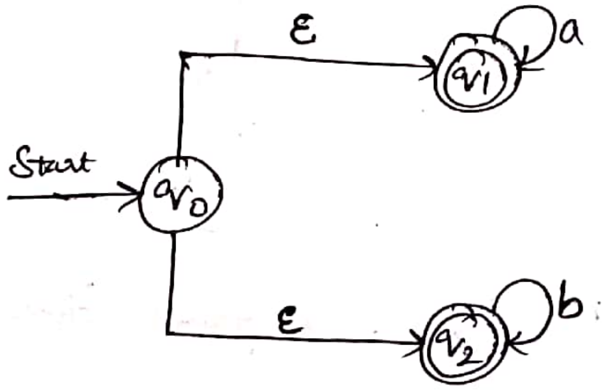
NFA For $L2 = \{a^n \cup b^n\}$.

<u>L2:</u>

The language 'L2' is a language, in which there be any no. of a's (or) any no. of b's.

It accept's $\{a, b, aa, bb, aaa, bbb, \ldots\}$.

Hence the NFA will be as follows.



* NFA Chooses two different state q_1, q_2 for Input ε from q_0 State.
* Here ε (Epsilon) is basically a null move.
* 'ε' move doesn't Carry any symbol from input set 'Σ'
* But a State Change occurs from one State to another.

Q:8 Construct NFA for the language

$$L = \{0101^n \cup 0100 \mid n \geq 0\}$$

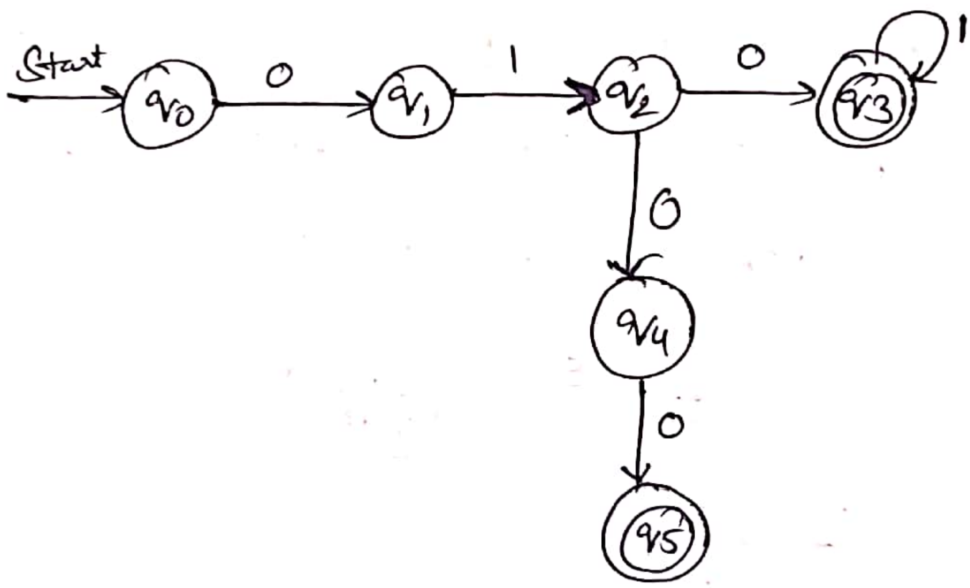
Over. $\Sigma = \{0, 1\}$

<u>Sol:</u> Given NFA For language $L = \{0101^n \cup 0100 \mid n \geq 0\}$ over $\Sigma = \{0, 1\}$

Scanned by CamScanner

The language 'L' First '3' Symbols are Common '010'

we can draw NFA as below.



* The States q_3 & q_5 are final States accepting 0101"

and 0100 respectively.

The NFA Can be denoted by '5'-tuple

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

$M = \left(\underbrace{\{q_0, q_1, q_2, q_3, q_4, q_5\}}_{Q}, \underbrace{\{0,1\}}_{\Sigma}, \underbrace{\delta}_{\delta}, \underbrace{q_0}_{q_0}, \underbrace{\{q_3, q_5\}}_{F} \right).$

Q = No. of ~~Inputs~~ States = States

Σ = No. of Inputs

δ = Transition Function

q_0 = Initial State.

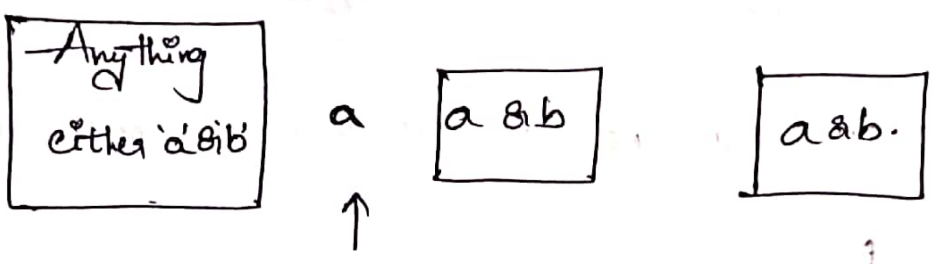
F = q_3, q_5 = Final State.

Scanned by CamScanner

Q.4: Construct a NFA for a language L which accepts all the strings in which the third symbol from right end is always 'a' over $\Sigma = \{a, b\}$.

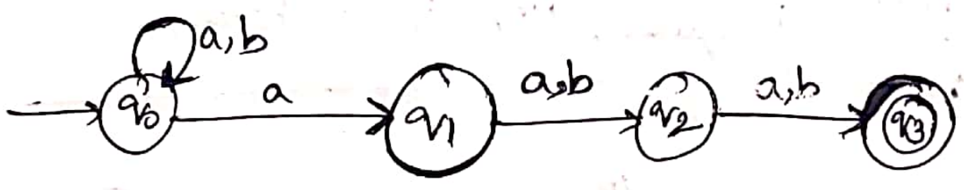
Sol.

The strings in such a language are of the form



Third symbol from right end should always be 'a'

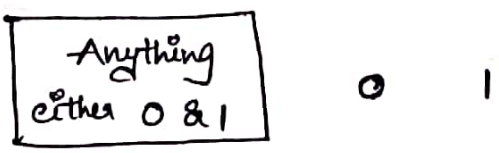
The NFA is:



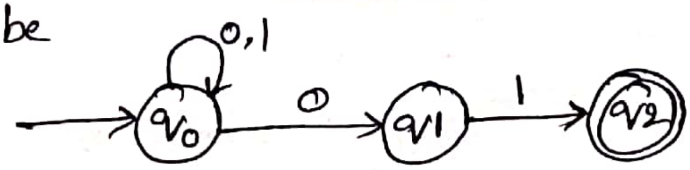
Note: The above figure is NFA, because in state q_0 with input 'a' we can go to either q_0 (or) state q_1

Q.5 Design NFA accepting all strings ending with 01. over $\Sigma = \{0, 1\}$

Sol.



Hence, NFA would be

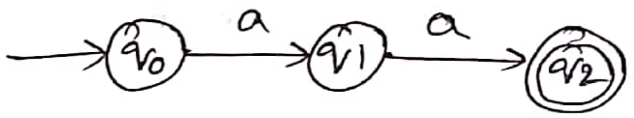


Scanned by CamScanner

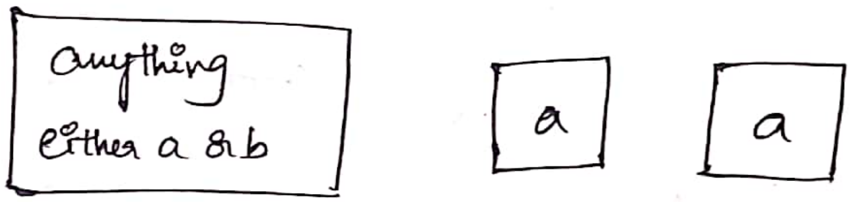
Q:6 Design NFA to accept strings with a's & b's such that the string end with 'aa'

Sol.

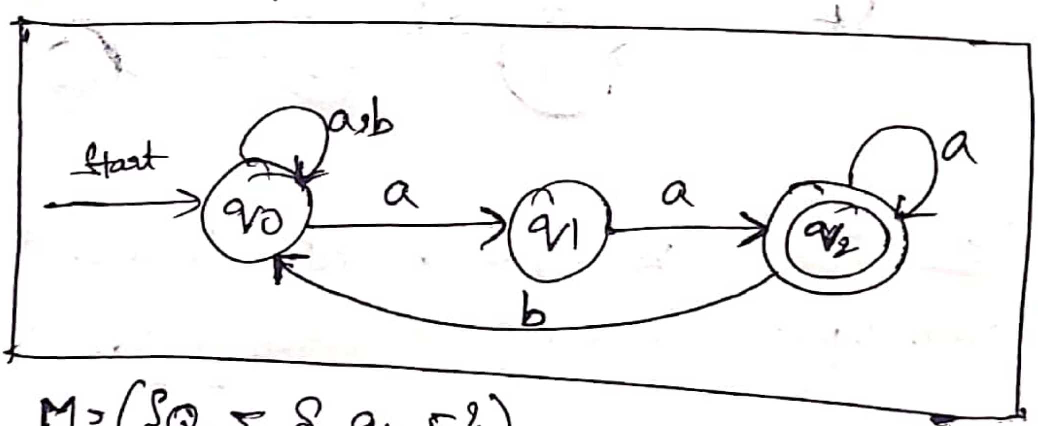
Step ①: Simple FA which accepts a string with aa.



Step ②: There can be a situation as below also.



Step ③: Required NFA is as below.



$M = (\{Q, \Sigma, \delta, q_0, F\})$

$M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

Acceptance:- String – 'aaa'

$\delta(q_0, aaa) \Rightarrow$

$\delta(q_1, aa)$

$\delta(q_2, a) \Rightarrow q_2 \rightarrow$ Final state so The Machine accepted.

Scanned by CamScanner

Acceptance of another String 'ababaabaaa'. :-

$\delta(q_0, ababaabaaa)$

$\delta(q_0, babaabaaa)$

$\delta(q_0, abaabaaa)$

$\delta(q_0, baabaaa)$

$\delta(q_0, aabaaa)$

$\delta(q_1, abaaa)$

$\delta(q_2, baaa)$

$\delta(q_0, aaa)$

$\delta(q_1, aa)$

$\delta(q_2, a) \rightarrow \delta(q_2, \epsilon)$

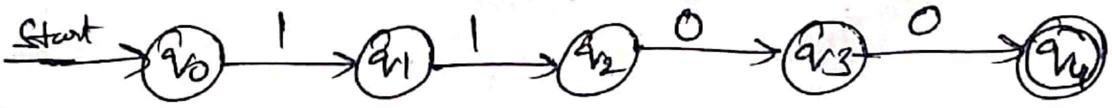
$q_2 \rightarrow$ Final State Reached.

Q:7 Construct a NFA in which double '1' is followed by double '0' over $\Sigma = \{0, 1\}$.

Solution:

Step①: The FA with double '1' is as drawn below.

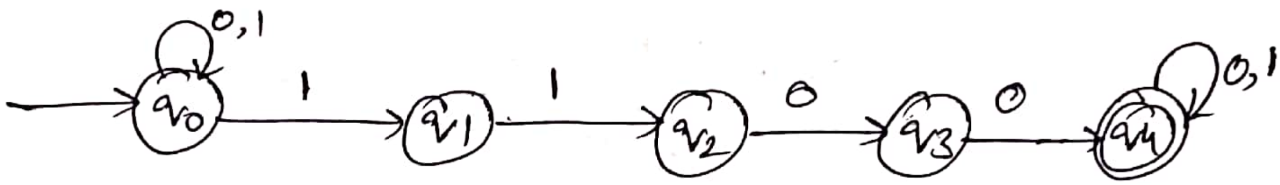
Step②: double '1' Should immediately followed by double '0'



Scanned by CamScanner

Step ③: Now assume there will be a Chance of getting before double '1' any string of 0 & 1. In the same way after double '0' there can be any string of 0 & 1

✱ It means before and after of 1100 there is a chance of the string with 0 & 1. i.e; as below



Transition Table

Input / States	1	0
q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	–
q_2	–	q_3
q_3	–	q_4
q_4	q_4	q_4

String acceptance: "11100"

$\delta(q_0, 11100)$
$\delta(q_0, 1100)$
$\delta(q_0, 100)$
$\delta(q_1, 00)$

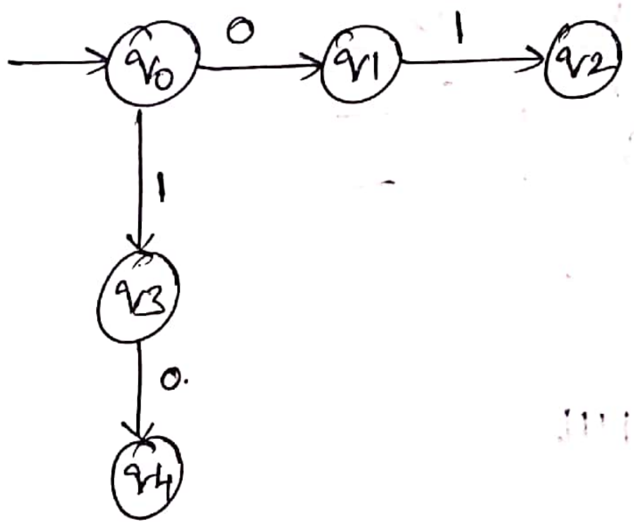
Got Struck! No path after that, so try another path. ⇒

$\delta(q_0, 11100)$
$\delta(q_0, 1100)$
$\delta(q_1, 100)$
$\delta(q_2, 00)$
$\delta(q_3, 0) \Rightarrow \delta(q_4, \varepsilon)$ Final State

Scanned by CamScanner

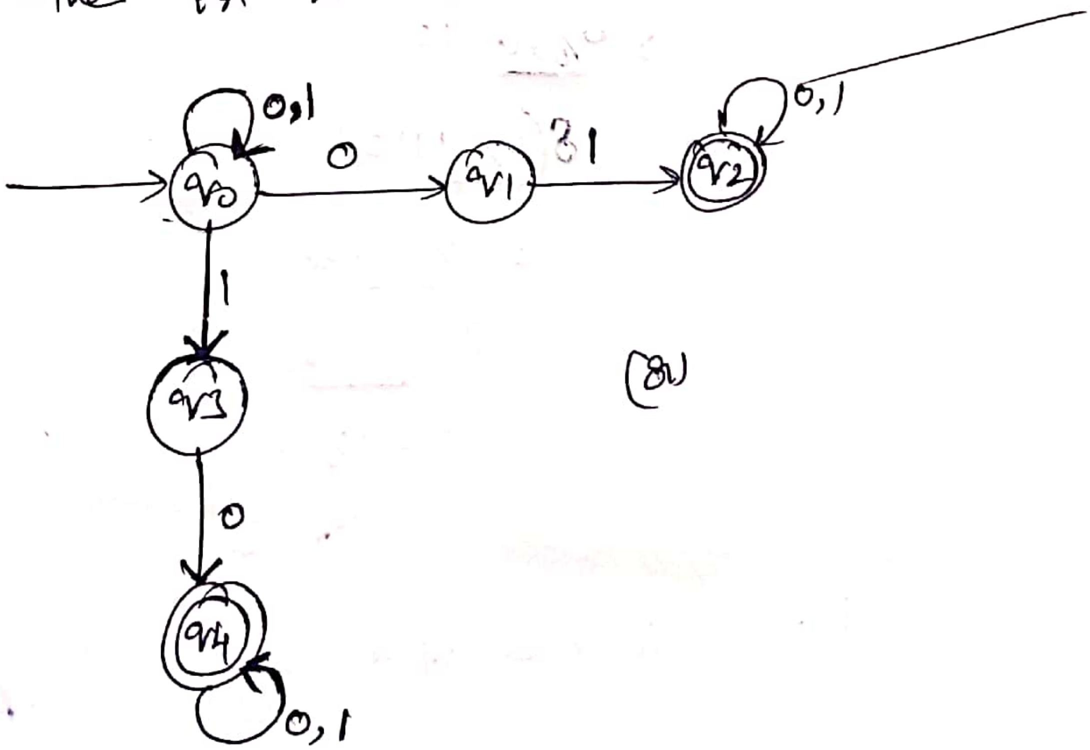
Q.8 Design NFA which accepts the String Containing either '01' or '10' over $\Sigma = \{0, 1\}$

Solution: Step(1): String Containing either '01' or '10'



Step(2): before and after of either '01' or '10' there is a Chance of getting a String with '0' & '1'.

So, the FA will be as below.



Scanned by CamScanner

Transition Table:-

δ

States \ Input	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0, q_1\}$
q_1	—	q_2
q_2	q_2	q_2
q_3	q_4	—
q_4	q_4	q_4

Input Σ

String Acceptance :-

String = 00001110

$\delta(q_0, 00001110)$

$\delta(q_0, 0001110)$

$\delta(q_0, 001110)$

$\delta(q_0, 01110)$

$\delta(q_1, 1110)$

$\delta(q_2, 110)$

$\delta(q_2, 10)$

$\delta(q_2, 0)$

(q_2, ε)

Reached to Final State, So the String accepted.

Scanned by CamScanner

Q:9 Design the NFA transition diagram for the transition table as given below.

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

Sol: Given transistion table and from that we can write as below.

$$M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\})$$

From table we can write

$\delta(q_0, 0) = \{q_0, q_1\}$

$\delta(q_0, 1) = \{q_0, q_2\}$

$\delta(q_1, 0) = \{q_3\}$

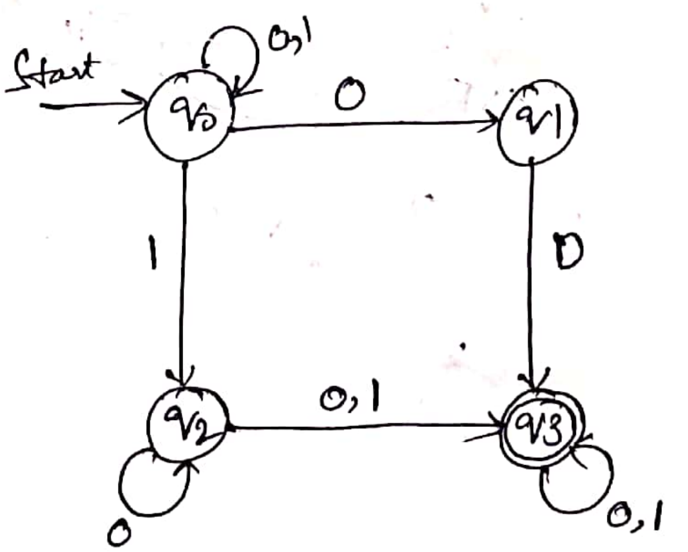
$\delta(q_2, 0) = \{q_2\}, \{q_3\}$

$\delta(q_2, 1) = \{q_3\}$

$\delta(q_3, 0) = \{q_3\}$

$\delta(q_3, 1) = \{q_3\}$.

δ



Scanned by CamScanner

Q:10 Construct a transition diagram for the NDFA

$M = (\{q_1, q_2, q_3\}, \delta, q_1, \{q_3\})$ where δ is given by

$\delta(q_1, 0) = \{q_2, q_3\}$ | $\delta(q_1, 1) = \{q_1\}$

$\delta(q_2, 0) = \{q_1, q_2\}$ | $\delta(q_2, 1) = \phi$

$\delta(q_3, 0) = \{q_2\}$ | $\delta(q_3, 1) = \{q_1, q_2\}$

Sol:

Transition Table:

States \ Input	0	1
q_1	$\{q_2, q_3\}$	$\{q_1\}$
q_2	$\{q_1, q_2\}$	ϕ
q_3	$\{q_2\}$	$\{q_1, q_2\}$

NDFA: