Theory of computation:

TOC: In theoritical Computer Science, the theory of Computation is the branch that deals with whether and how efficiently problems can be solved on a model of Computation, using an algorithm.

This field is divided into three major branches:

1. **Automata Theory**: Deals with definitions & properties of various mathematical models
2. **Computability Theory**: Deals with what can, cannot be computed by the model.
3. **Computational Complexity theory**: Deals with Computable problems based on their hardness.

Automata Theory: Deals with definitions & properties of different types of "Computational models"

Examples of such models:

→ Finite Automata

→ Context-Free Grammars

→ Turing Machines

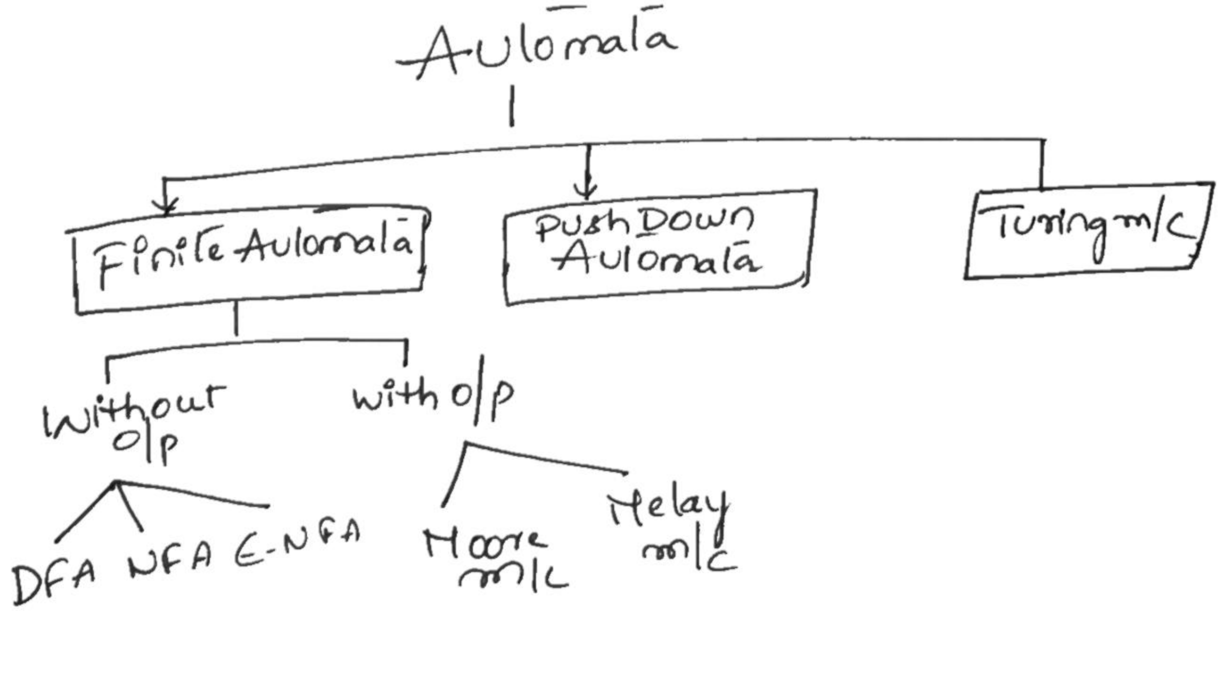
Automata : An automata is a system (abstract machine) which transform information | data by performing some internal functions with different state at different instant of time and produce o|p without participation of human.

$\xrightarrow{i|p}$ Automaton $\longrightarrow$ o|p

→ Automata means self-acting m|c ie; automatic.

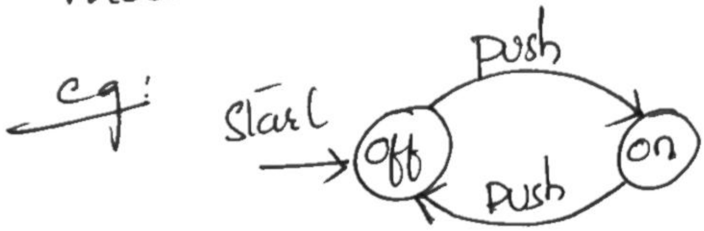
→ In TOC actual m|c is not designed but a mathematical model | abstract m|c is designed.

→ Automata — States (represented by circles)
— Transitions (represented by arrows)



Finite Automata : "An Automaton with finite number of state is called Finite Automata | finite state machine"

eg:



→ FA useful in both H/w & S/w

→ Remembers only finite amount of information

→ Used in S/w for designing & checking the behaviour of digital circuits

→ Used in "Lexical Analyzer" in a Compiler

→ Used in S/w for scanning large bodies of text such as to find occurrences of words, phrases & other patterns.

→ FA being a part of Lexical Analyzer. The Job of automaton to recognize the keyword "then"

Symbol	Meaning
○	State
→○	Start state
◎	End state or Accept state
→	Transition

Central Concepts of Automata Theory:

Symbol: The basic building block to build languages for machine.

→ It is an individual part of input symbols or single character is known as a symbol

eg: digit 0, 1, 2 ... 9

alphabet a, b, c ... z

A, B, C ... Z

spl symbols @, !, $, <, > ...

Alphabet : It is a finite non-empty set of i/p symbols
→ It is represented by sigma Σ
→ eg: $\Sigma = \{0, 1, 2\}$
$\Sigma = \{a, b\}$

String : Finite sequence of symbols from the Alphabet (Σ)
→ It is represented by W
eg: $\Sigma = \{0, 1\}$
W = 011010..
$\Sigma = \{a, b\}$
W = ababb

→ Empty string denoted by epsilon 'ε'
→ In Empty string no character is there & its length is equal to zero.
→ Prefix → of a string is no: of leading symbols of the string
Suffix → of a string is any no: of trailing symbols of the string.

eg: "0011" prefix → 0, 00, 001
suffix → 1, 11, 011

eg: "Mango"
Prefix → man
Suffix → go

Length of the string:- It is equal to the number of Symbols | Characters in the string

→ It is represented by $|w|$ (mod of w)

eg: $w = hello$

$|w| = 5$

$w = \epsilon$ (Empty string)

$|w| = 0$

Operations on String:

(i) Concatenation → $w_1 = ab$ $w_2 = cd$
$w_1 w_2 = abcd$

(ii) ~~Transpose~~ ↔

(iii) palindrome → $w = aba$ palindrome: aba

(iv) Substring → $w = ab$ substring → ϵ, a, b, ab

(v) union → $w_1 = ab$ $w_2 = cd$ $w_1 \cup w_2 = \{ab, cd\}$

(vi) Reverse of a string → $w = ab$ $w^R = ba$

Language: A Language is set of all strings that are choosen from an Alphabet Σ.

→ It is $L \subseteq \Sigma^*$, where L is a Language.

eg:1 $\Sigma = \{a\}$

Language Consist of any no. of a's over Σ

then $L = \{\epsilon, a, aa, aaa \ldots\}$

$L = \{a^n \mid n \geq 0\}$

eg:2 Describe the Language Consist of equal no. of 0's followed by equal no. of 1's over $\Sigma = \{0,1\}$

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

$L = \{\epsilon, 01, 0011, 000111 \ldots\}$

$L = \{0^n 1^n \mid n \geq 0\}$

Language → finite Language ⇒ contain finite no: of strings

Language → Infinite Language ⇒ contain infinite no: of strings

Ways to define a Language:-

(i) Set-Former $\rightarrow$ $\{ w \mid \text{something about } w \}$

eg: $L = \{ a^n \mid n >= 0 \}$

[read as the set of a to the power n such that n is greater than or equal to 0]

(2) Roster | Tabular form $\rightarrow$ $L = \{ \varepsilon, a, aa, aaa \ldots \}$

(3) Regular Expression $\rightarrow$ $L = a^*$

Operations on Languages: $A = \{1,2\}$ $B = \{2,3\}$

① Concatenation of Two Languages

eg: $AB = \{(1,2)(1,3)(2,2)(2,3)\}$

② Union of two Languages

$A \cup B = \{1,2,3\}$

③ Intersection $\rightarrow$ $A \cap B = \{2\}$

④ Difference $\rightarrow$ $A - B = \{1\}$

$B - A = \{3\}$

⑤ Compliment

⑥ Kleene Closure (L^*)

⑦ positive Closure (L^+)

Power of Alphabet :-

$\Sigma \rightarrow$ Alphabet

$\Sigma^k \rightarrow$ power of Alphabet

$k \rightarrow$ set of all strips of length k

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

$\Sigma^0 = \{\varepsilon\}$

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

$\Sigma^2 = \{00,01,10,11\}$

$\Sigma^3 = \Sigma^2\Sigma = \{00,01,10,11\}\{0,1\}$

$= \{000,001,010,001,100,$
$101,110,111\}$

Kleene Closure :- It consists of set of all strings including Length '0'.

→ denoted by L^*, ie; Empty string is also included

eg → $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$

Positive Closure :- It consists of set of all strings except the Empty string

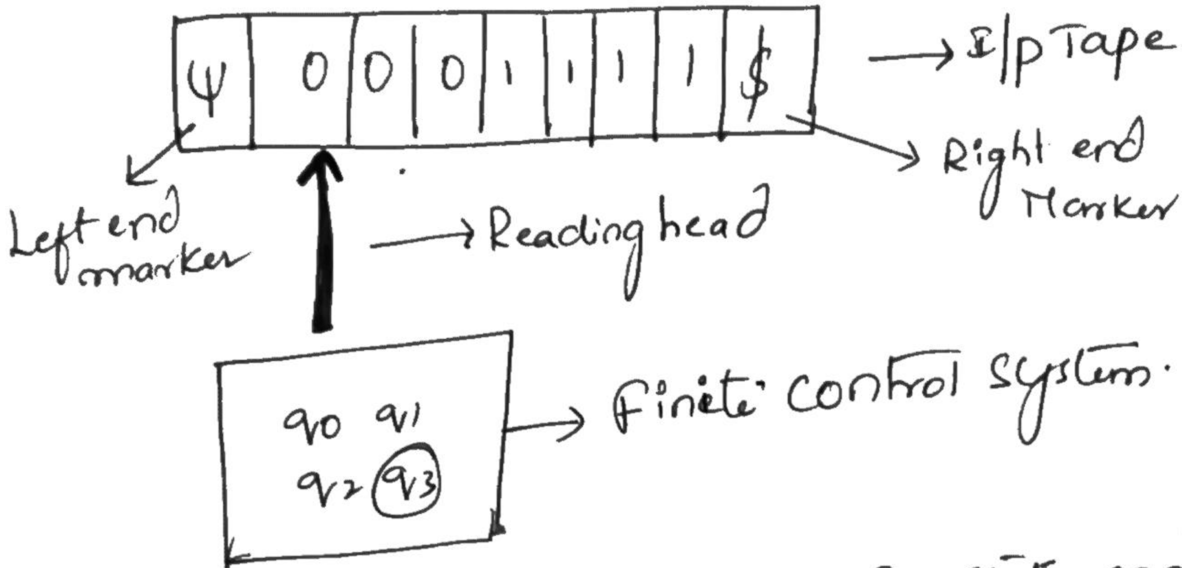
ie; (one or more occurrences).

→ represented by L^+

$L^+ = L^* - \{\epsilon\}$

(or) $L^1 \cup L^2 \cup L^3 \cup \ldots$

Model & Behaviour of FSM or FA:



→ A FA consists of a finite set of states and a set of transitions from one state to another state that occurs on the i/p symbols (chosen from an alphabet 'Σ') & gives o/p

→ The read head is placed at 1st cell immediate right to left end marker and reads i/p symbol one by one until it gets right end marker($) & checks the current state of the FA if it belongs to final state. It represents the given string is valid otherwise not valid.

$$\delta(q_0, w) \in F$$

FA is mainly represented in two way

1. Transition Diagram
2. Transition Table.

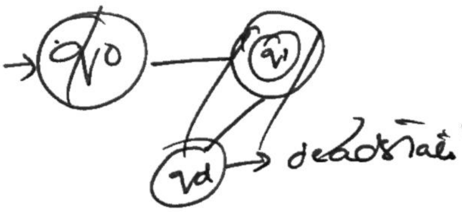
(1) Transition Diagram: In transition diagram set of vertices & edges are present.
The vertices of transition diagram are called "states" & "edges" are called i/p symbols of FA.

There are different types of states in FA

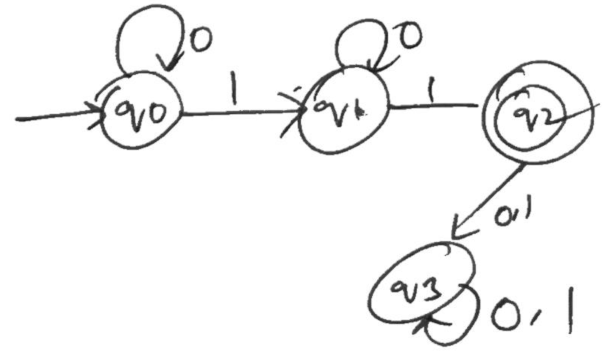
(i) Initial state $\longrightarrow q_0$

(ii) Final state — q_n

(iii) Dead state —



eg



(2) Transition Table:- It consists of rows & columns

rows → no: of states

columns → set of i/p symbols

eg:

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_1	q_2
$* q_2$	q_3	q_3
q_3	q_3	q_3

Formal Definition of FA :-

- The FA is described as five Tuple notation (Quintuple) (5 elements in a tuple)

ie; Quintuple (5 elements in a tuple)

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

Where $Q \rightarrow$ finite & non-empty set of states

$\Sigma \rightarrow$ sequence of symbols from an Alphabet

$\delta \rightarrow$ Transition or Mapping function

$q_0 \rightarrow$ Initial state

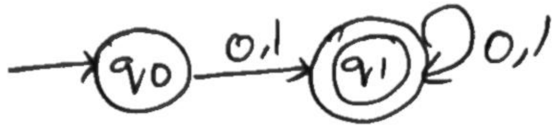
$F \rightarrow$ Set of final state

$$\delta = Q \times \Sigma \rightarrow Q$$

~~$q_0 \in$~~ $q_0 \in Q$

$F \subseteq Q$

→ eg:



$Q = \{q_0, q_1\}$

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

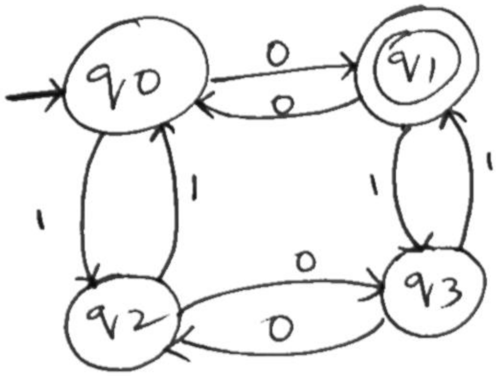
$q_0 \rightarrow$ initial state

$F \rightarrow \{q_1\}$

$\delta : Q \times \Sigma \rightarrow Q$

δ	0	1
$\rightarrow q_0$	q_1	q_1
$* q_1$	q_1	q_1

→ Consider the following transition diagram



Q i) Describe the given transcation diagram using 5 tuple notation

ii) Draw Transition Table

iii) Check whether the following strings are accepted or not

a) 01010110 b) 00110 c) 101011010

Sol^n i) $M = (Q, \Sigma, q_0, \delta, F)$

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

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

ii)

δ	0	1
$\rightarrow q_0$	q_1	q_2
$* q_1$	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

iii) a) $\delta(q_0, 0101\,0110)$
$\delta(q_1, 101011\,0)$
$\delta(q_3, 010110)$
$\delta(q_2, 10110)$
$\delta(q_0, 0110)$
$\delta(q_1, 110)$
$\delta(q_3, 10)$ $\delta(q_1, 0) = q_0$
$q_0 \notin F$ so invalid string

b) 00110
$\delta(q_0, 00110)$
$\delta(q_1, 0110)$
$\delta(q_0, 110)$
$\delta(q_2, 10)$
$\delta(q_0, 0)$
$= q_1 \in F$
Valid string

→ Acceptance of String :- String 'x' is accepted by FA if $\delta(q_0, x) = P \in F$

eg: Find whether 1011 is accepted by following Automata or not.

$$\delta(q_0, 1011) = \delta(q_1, 011)$$
$$= \delta(q_2, 11)$$
$$= \delta(q_2, 1)$$
$$= q_2 \in F$$

$$q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \in F$$

Valid or accepted.

Types of Finite Automata:

(i) Deterministic FA (DFA)

(ii) Non Deterministic FA (NFA)

DFA: A FA is called as a DFA if there is only one path from the current state to next state on every specific i/p symbol

Formal Definition of DFA:-

DFA can be described by using 5 tuple notation

$M \; (Q, \Sigma, q_0, \delta, F)$

where Q finite is non empty set of states

Σ set of i/p symbols

q_0 initial state

M: set of final states

δ transition or mapping function

i.e. $\delta : Q \times \Sigma \rightarrow Q$

where F is a set of final states

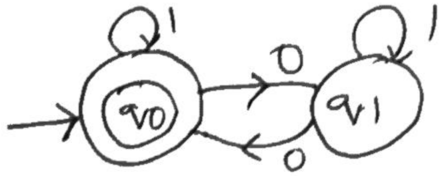
$F \subseteq Q$

Numericals on DFA:

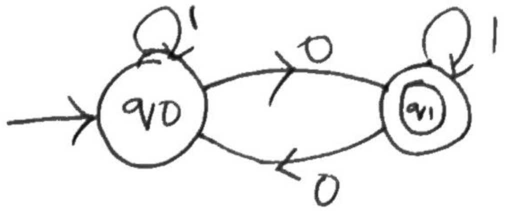
① Construct a DFA for even no: of zeros

$L = \{\epsilon, 1, 00, 001, 00001, 000\}$

When 'ε' is there in language then the start state will be the final state



② odd no: of zeros



③ DFA for Accept the string '01'

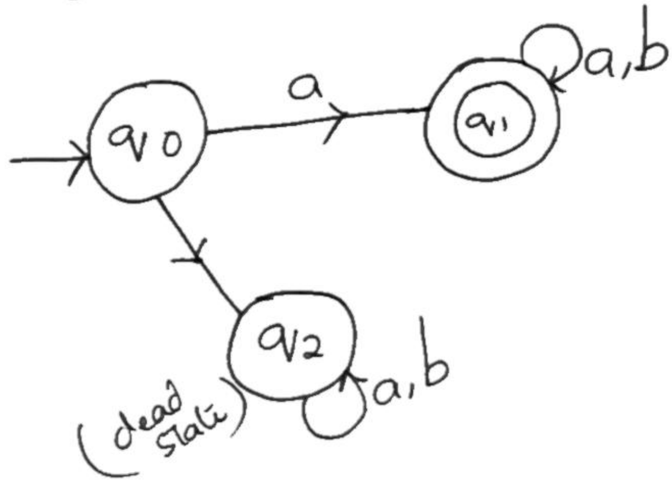
$L = \{01, 010, 001, 0010, 0101 \ldots\}$
101,

Stārt With: (Hint: Expect a dead state)

(4) Construct DFA which accept set of all strings over {a,b}

Which stārt with i) 'a'

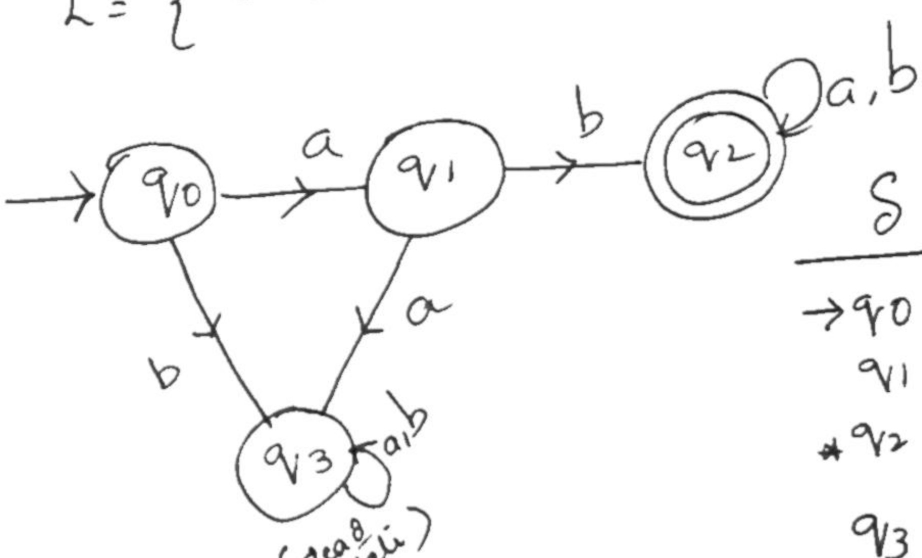
L = { a, ab, aab, aba, aa ... }



δ	a	b
→q0	q1	q2
*q1	q1	q1
q2	q2	q2

ii) Stārt with 'ab'

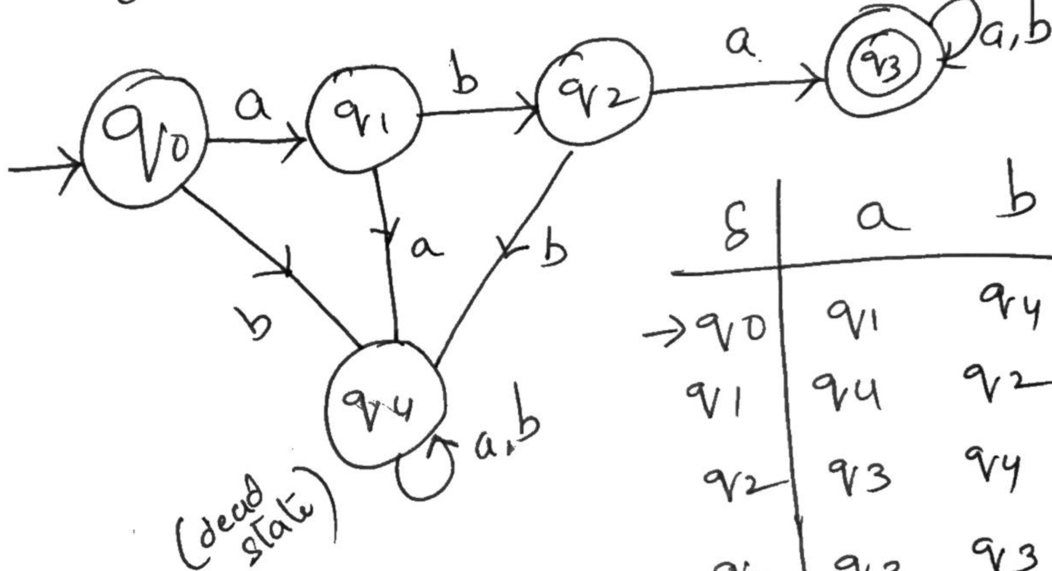
L = { ab, aba, abb, abab, abaa ... }



δ	a	b
→q0	q1	q3
q1	q3	q3
*q2	q2	q2
q3	q3	q3

iii) Start with 'aba'

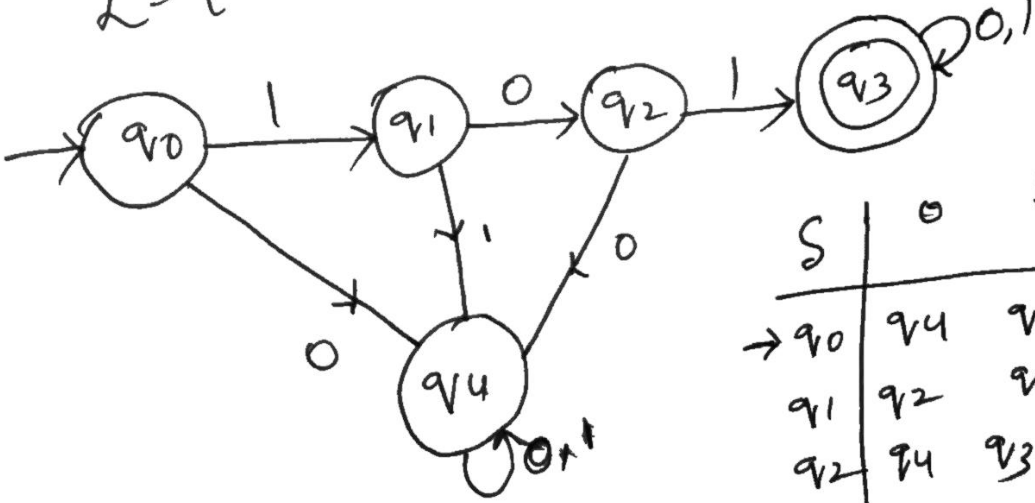
L = {aba, abaa, abab, abaab . . . }



δ	a	b
→q0	q1	q4
q1	q4	q2
q2	q3	q4
*q3	q3	q3
q4	q4	q4

iv) Construct DFA which accepts set of binary strings begins with 101 Σ = {0,1}

L = {101, 1010, 1011, 10110, 101101 . . . }



δ	0	1
→q0	q4	q1
q1	q2	q4
q2	q4	q3
*q3	q3	q3
q4	q4	q4

Substring:

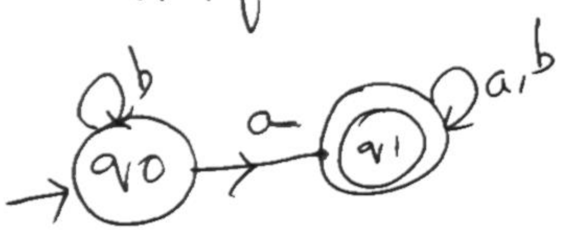
5) Construct DFA which accept set of all strings over $\{a, b\}$ where each string contains

i) a

$L = \{a, aa, ab, ba, bab, aba, aab \dots\}$

min length = 1

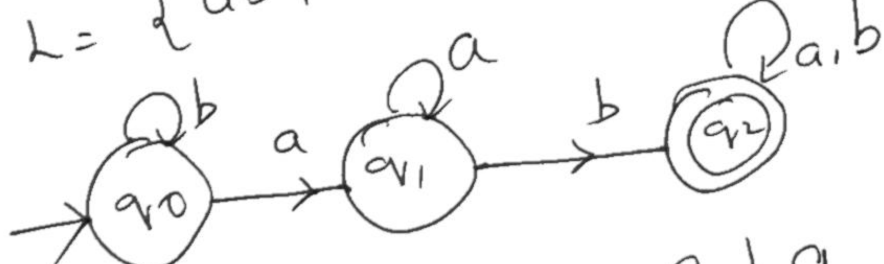
no. of states expected = 2



δ	a	b
q_0	q_1	q_0
q_1	q_1	q_1

ii) 'ab' as substring

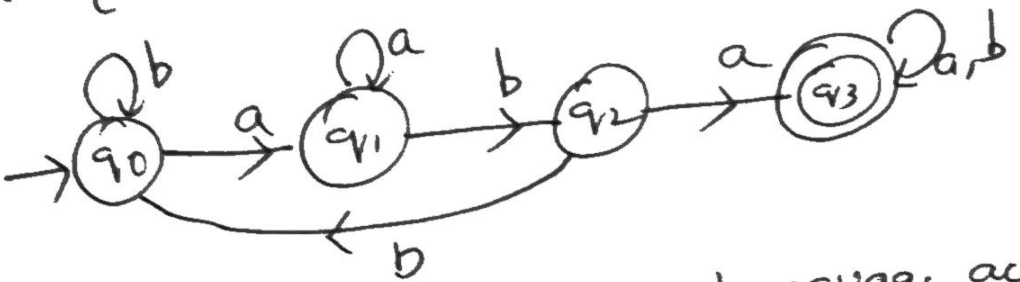
$L = \{ab, aab, aba, bab, baab \dots\}$



δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$* q_2$	q_2	q_2

iii) Contains 'aba'

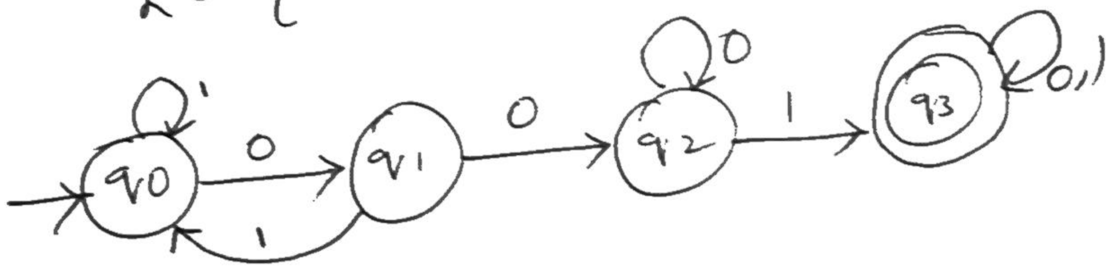
$L = \{aba, aaba, ababa, baba, babaa \dots\}$



iv) Demonstrate DFA for Language accepting strings which contains '001' as substring.

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

$L = \{001, 1001, 0001, 00010 \dots\}$

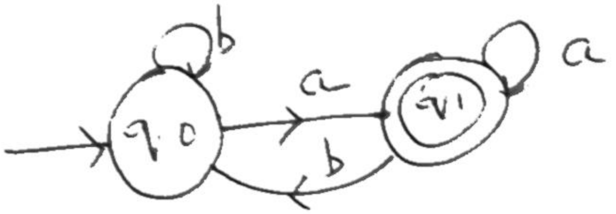


- Ends with:

(5) Construct DFA which accepts set of all strings are $\{a, b\}$ where each string ends with

i) 'a'

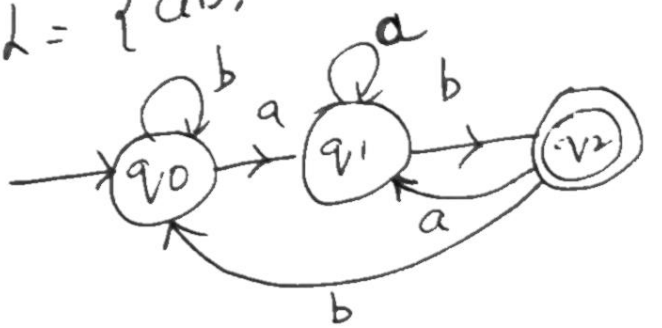
$L = \{a, aa, aba, bba \ldots\}$



δ	a	b
$\rightarrow q_0$	q_1	q_0
$*q_1$	q_1	q_0

ii) 'ab' end with

$L = \{ab, aab, aaab, bab \ldots\}$

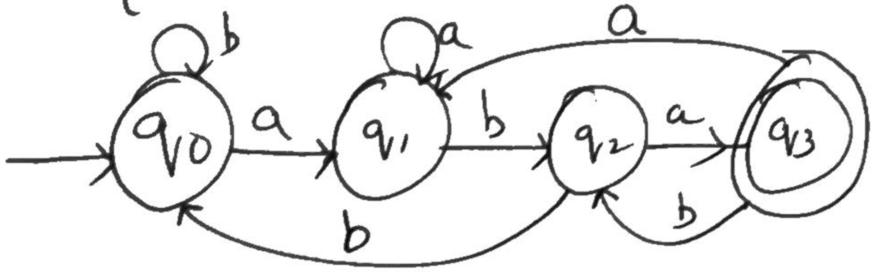


δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_1	q_0

iii) eoe

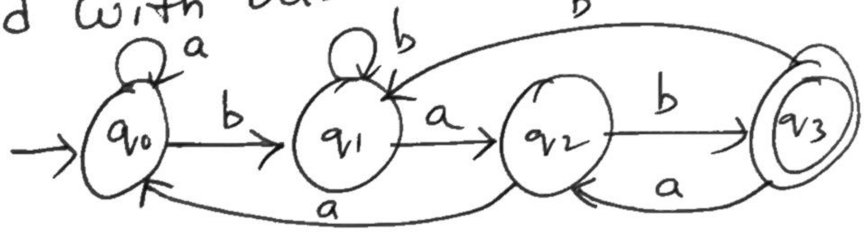
iii) end with aba

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



δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
$*q_3$	q_1	q_2

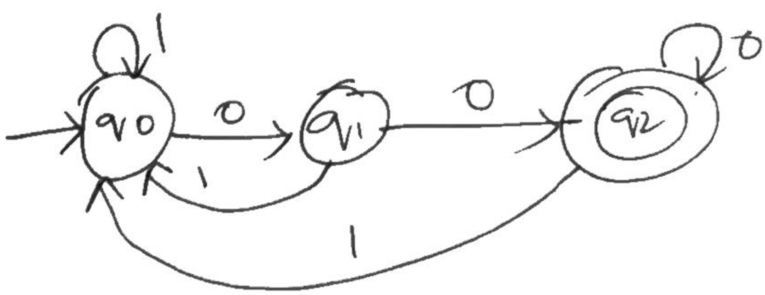
iv) end with bab



δ	a	b
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
$*q_3$	q_2	q_1

v) Construct DFA to accept the strings ends with 00 only over $\Sigma = \{0,1\}$

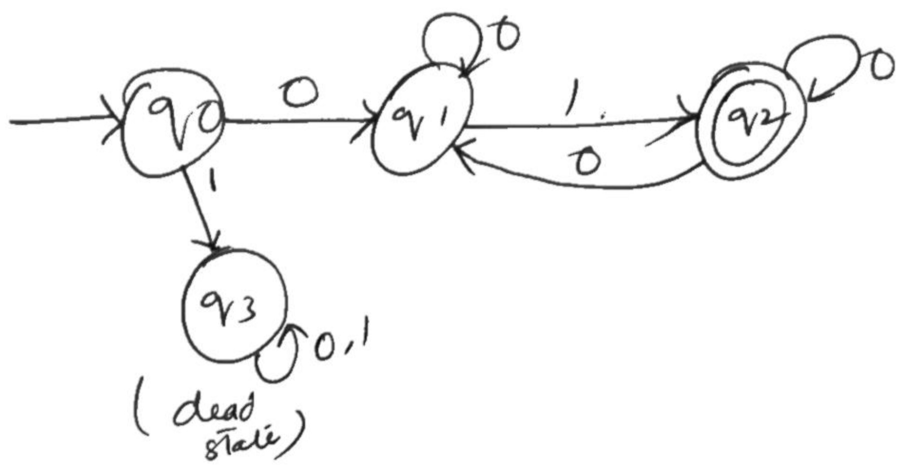
$L = \{00, 000, 100, 0100, 1000 \ldots\}$



δ	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
$*q_2$	q_2	q_0

⑦ Construct DFA that start with '0' end with '1' over $\Sigma = \{0,1\}$

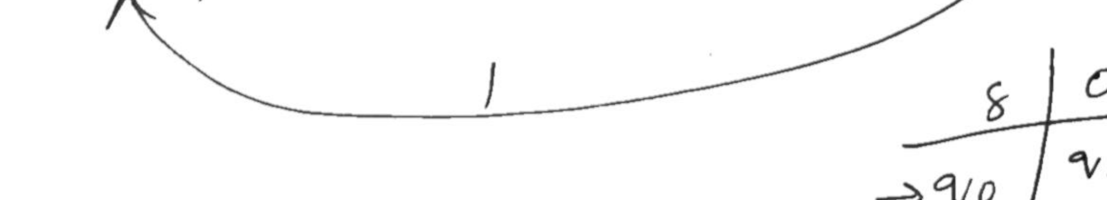
$L = \{0011, \underline{01}, 001, 0101, 0011 \ldots\}$



δ	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_1	q_2
$* q_2$	q_1	q_1
q_3	q_3	q_3

⑧ Construct DFA which accepts strings ending with 0101

$L = \{0101, 10101, 00101, 100101, 01010 \ldots\}$



δ	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_1
q_3	q_1	q_4
$* q_4$	q_3	q_0

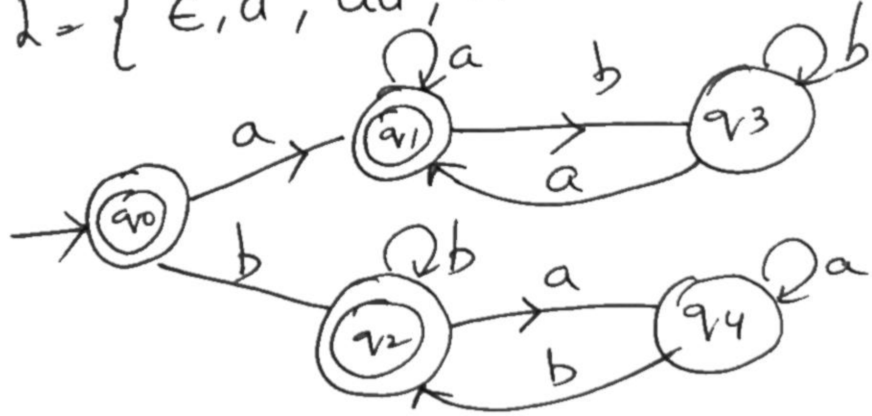
(9) Construct a DFA for the following Lang's

i) $L = \{w \mid w$ starts & ends with same symbol, $w \in (a,b)^*\}$

ii) $L = \{w \mid w$ starts & ends with different symbol, $w \in (a,b)^*\}$

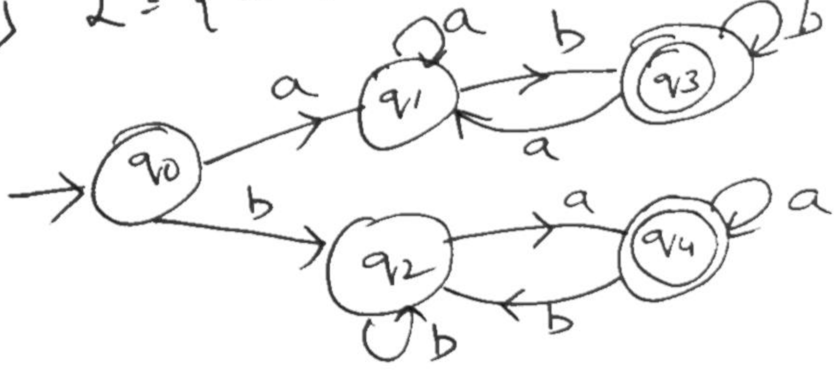
soln

i) $L = \{\epsilon, a, aa, b, aba, aaba, bb \ldots\}$



δ	a	b
$\rightarrow$ (q_0)	q_1	q_2
(q_1)	q_1	q_3
(q_2)	q_4	q_2
q_3	q_1	q_3
q_4	q_4	q_2

soln ii) $L = \{ab, aab, abb \ldots\}$



δ	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_1	q_3
(q_3)	q_1	q_3
q_2	q_4	q_2
(q_4)	q_4	q_2

(10) Construct a DFA which accepts the Strings on $\Sigma = \{0, 1\}$

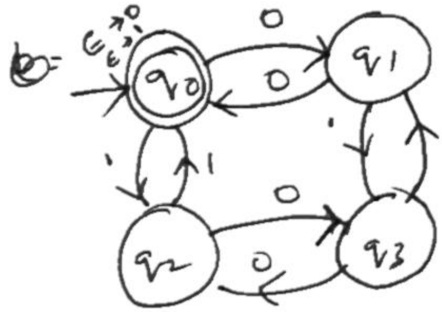
i) Even no: of 0's & even no: of 1's

ii) Even no: of 0's & odd no: of 1's

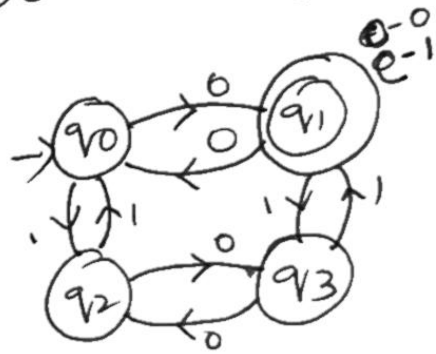
iii) Odd no: of 0's & even no: of 1's

iv) Odd no: of 0's & odd no: of 1's

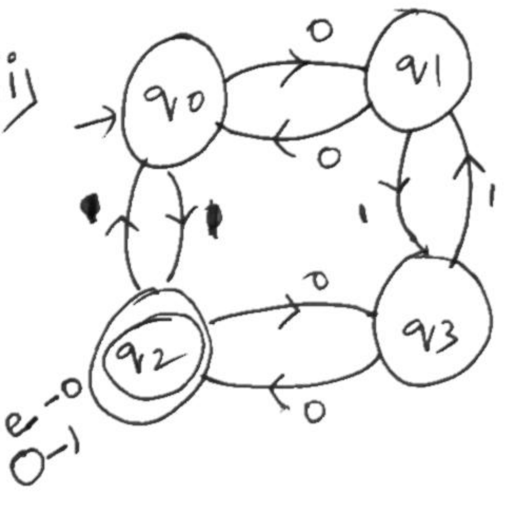
i)



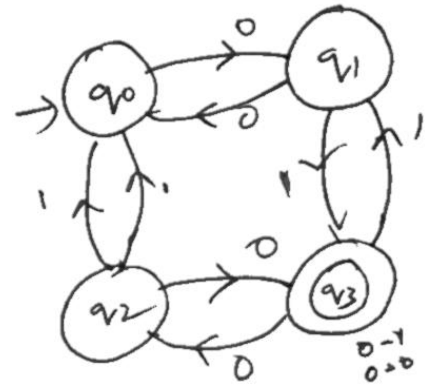
ii)



iii)



iv)



(11) Construct a DFA to accept the strings $\Sigma = \{a, b\}$ having

i) Exactly one 'a'

ii) atleast one 'a'

iii) Exactly 3 a's

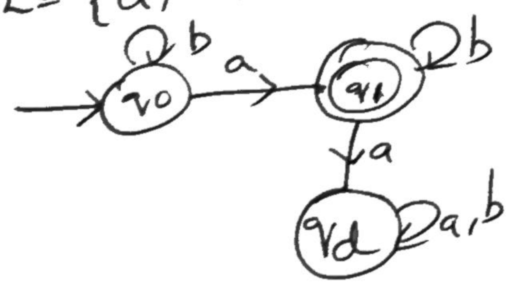
iv) atleast 3 a's

v) almost 3 a's

vi) almost 1 a

i) Exactly one 'a'

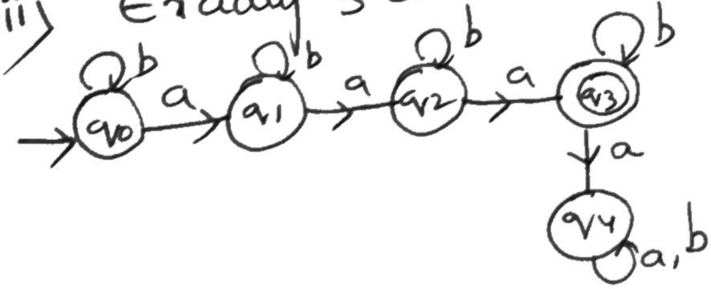
$L = \{a, ab, abb, bab \ldots\}$



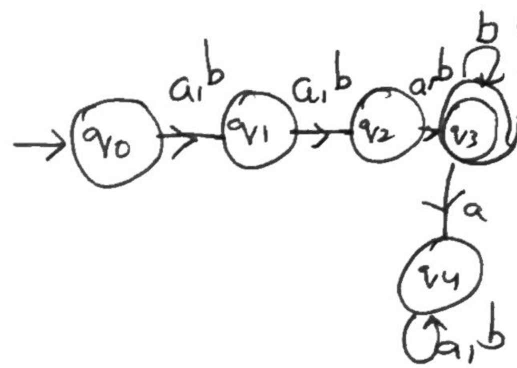
ii) atleast one 'a'

$L = \{a, ab, aba, aab \ldots\}$

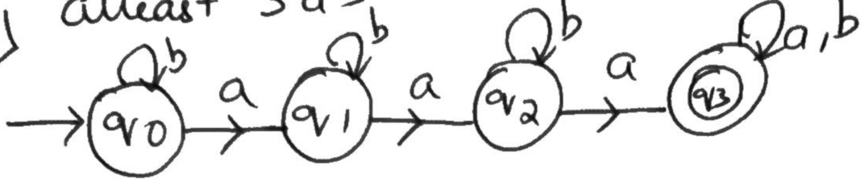
iii) Exactly 3 a's



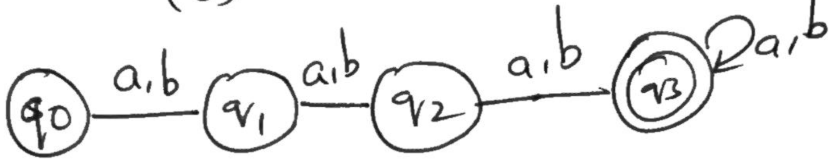
(or)



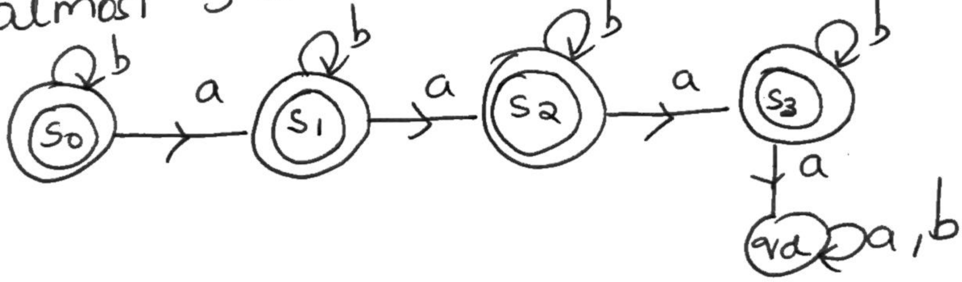
iv) atleast 3 a's



(or)

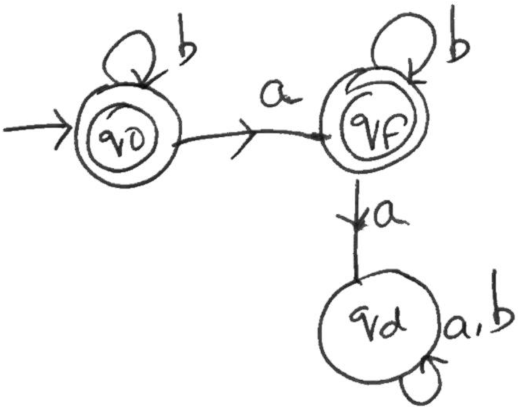


v) almost 3 a's



vi) almost 1 a

$L = \{\epsilon, bbb, a, ab \ldots\}$



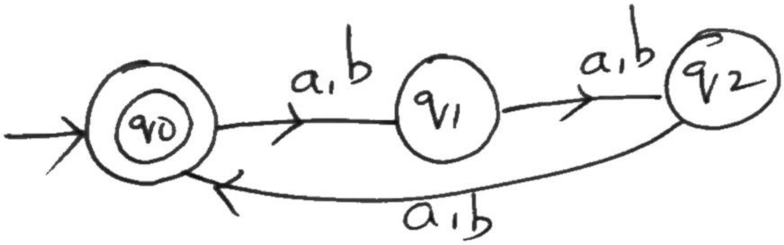
(12) DFA for

i) $\{w \mid |w| \bmod 3 = 0\}$ for $(a,b)^*$

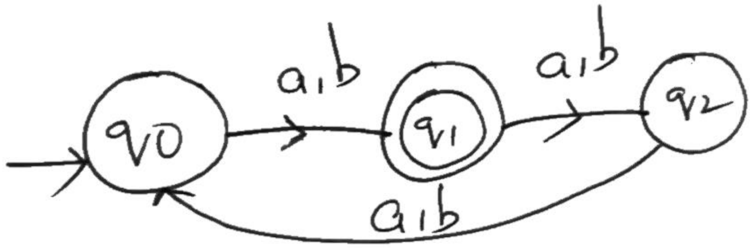
ii) $\{w \mid |w| \bmod 3 = 1\}$ for $(a,b)^*$

iii) $\{w \mid |w| \bmod 5 \neq 0\}$ for $(a,b)^*$

i)

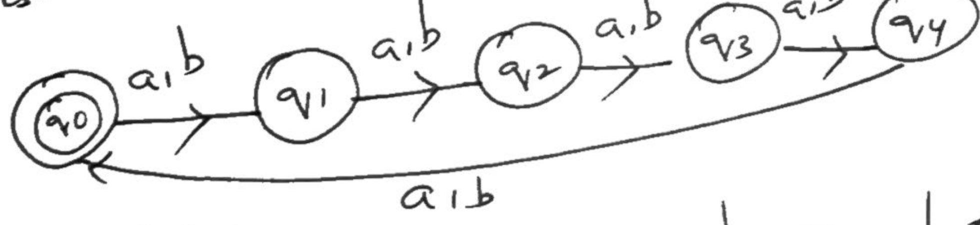


ii)



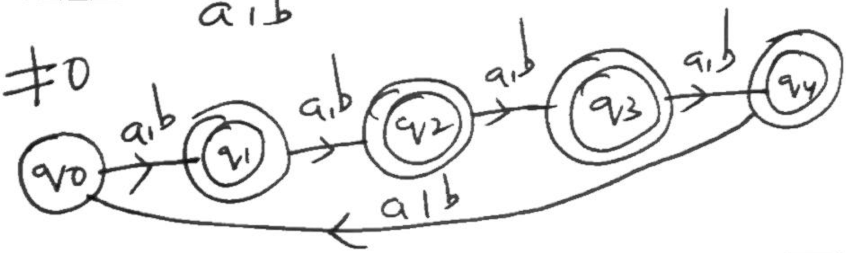
iii) $\{w \mid |w| \bmod 5 \neq 0\}$

first do $|w| \bmod 5 = 0$



Now reverse for $\neq 0$

~~states~~ accepting to nonaccepting



13) DFA for

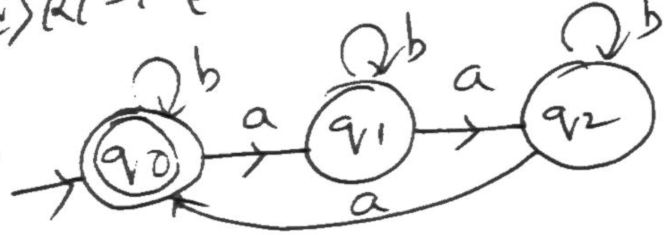
i) $L = \{ w \mid n_a(w) \bmod 3 = 0, \ w \in (a,b)^* \}$

ii) $L = \{ w \mid n_a(w) \bmod 3 = 1, \ w \in (a,b)^* \}$

iii) $L = \{ w \mid n_a(w) \bmod 3 = 2, \ w \in (a,b)^* \}$

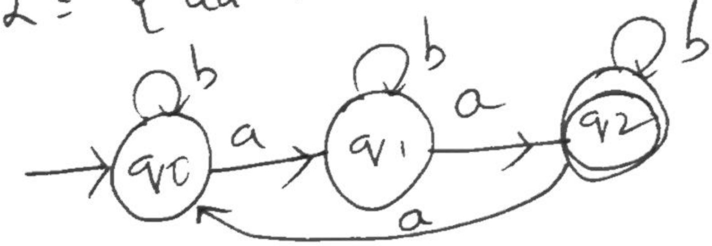
~~iv) L = { w | n~~

i) $L = \{aaabbb, abbaa, bbaaa \ldots\}$



ii) $L = \{abb, aabbbaa, aaaab \ldots\}$

iii) $L = \{aab, aba, abbabaaa \ldots\}$



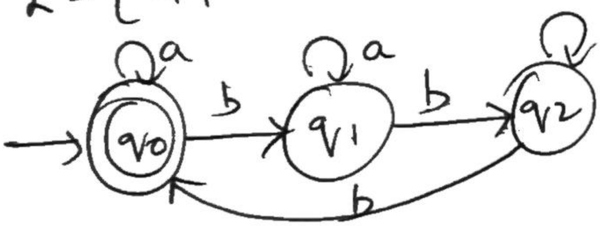
(14) DFA for

i) $L = \{ w \mid n_b(w) \bmod 3 = 0,\ w \in (a,b)^* \}$

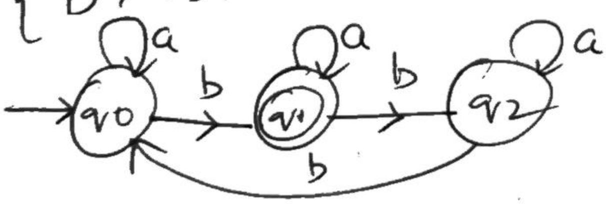
ii) $L = \{ w \mid n_b(w) \bmod 3 = 1,\ w \in (a,b)^* \}$

iii) $L = \{ w \mid n_b(w) \bmod 3 = 2,\ w \in (a,b)^* \}$

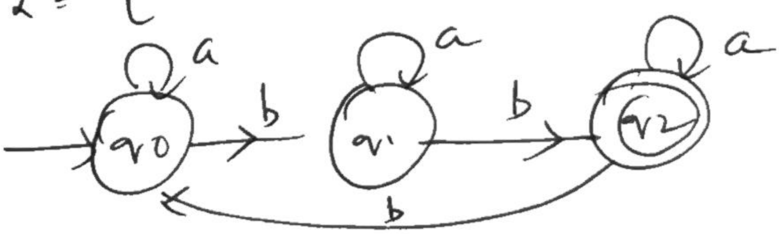
i) $L = \{ a, bbb, abbb, aabbb \ldots \}$



ii) $L = \{ b, ab, ba, abbbb, bbaabb \ldots \}$



iii) $L = \{ bb, abb, bba, abababbab \ldots \}$



(15) Construct DFA for decimal no. divisible by

i) 2

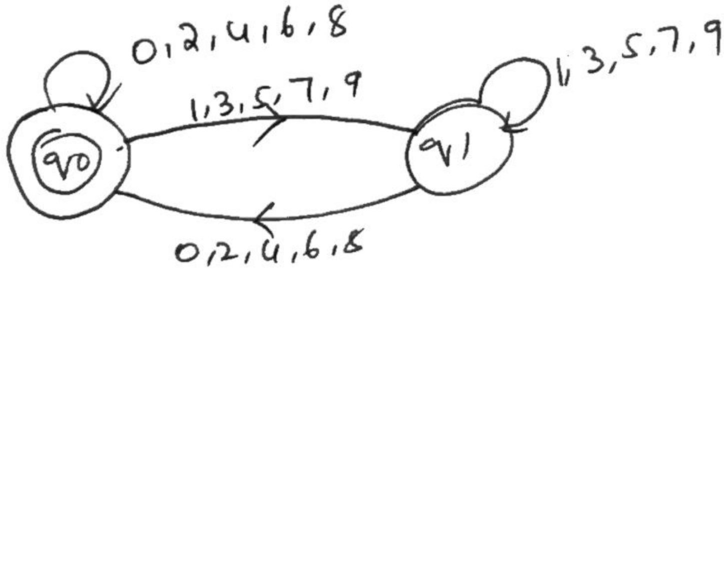
ii) 3

iii) 5

i) $q_0 \rightarrow n \% 2 = 0$

$q_1 \rightarrow n \% 2 = 1$

	{0,2,4,6,8}	{1,3,5,7,9}
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_1

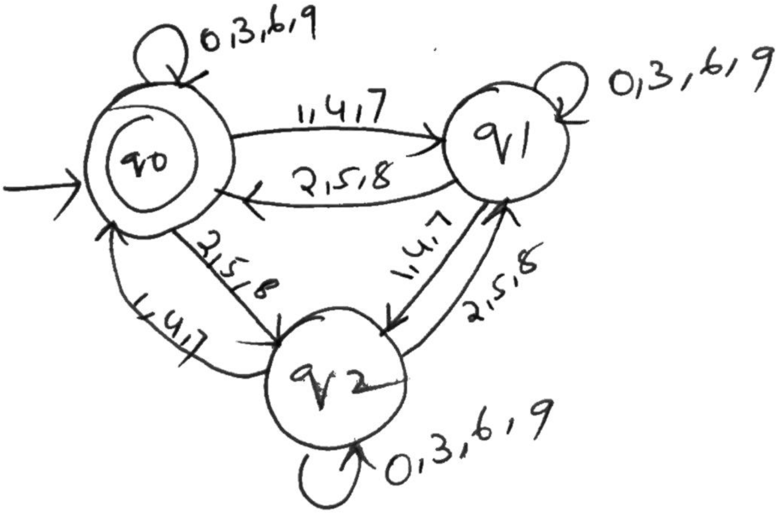


ii) $q_0 \rightarrow n \bmod 3 = 0$

$q_1 \rightarrow n \bmod 3 = 1$

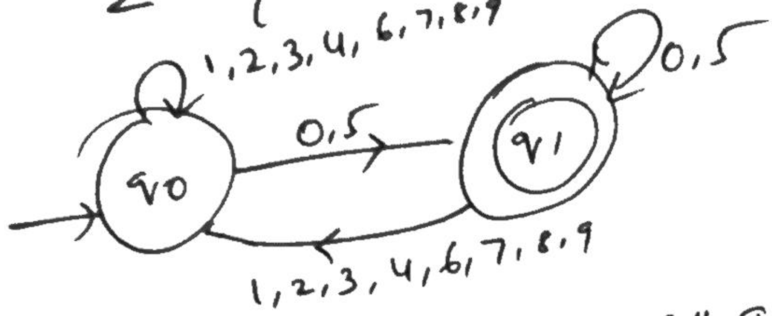
$q_2 \rightarrow n \bmod 3 = 2$

	{0,3,6,9}	{1,4,7}	{2,5,8}
$\rightarrow q_0$	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1



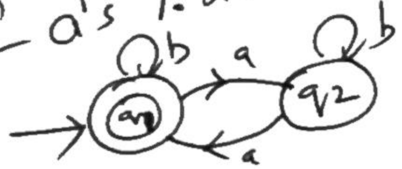
iii) decimal no: divisible by 5

$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$



(16) Describe a DFA that will accept those $p = \{a, b\}$ where no: of a's is divisible by 2 & no: of b's divisible by 3. Sketch Transition table.

sol a's % 2 = 0



DFA 1

b's % 3 = 0

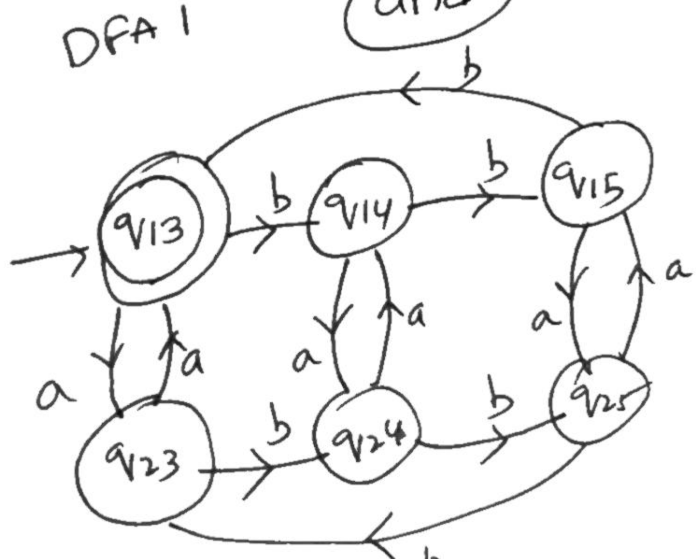
DFA 2

and

Cartesian product

$q_1 \times q_5 = q_{15}$

$q_1 \times q_4 = q_{14}$



Transition table

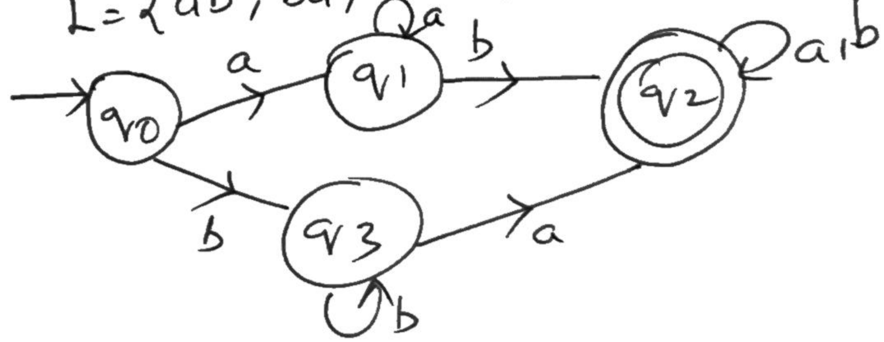
δ	a	b
→(q13)	q23	q14
q14	q24	q15
q15	q25	q13
q23	q13	q24
q24	q14	q25
q25	q15	q23

17) Describe DFA, the Language recognized by the automaton being

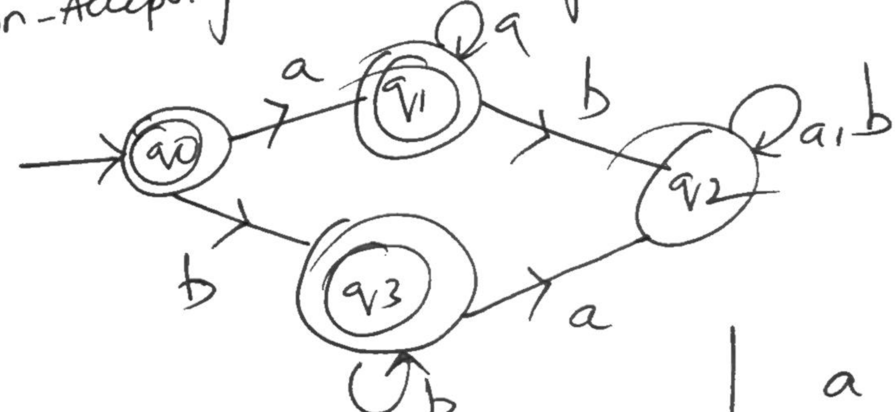
$L = \{ w \mid w$ contains neither the substring ab nor ba$\}$

sol firstly write DFA for contains ab or ba as substring

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



NOW reverse DFA i.e, Accepting to non-Accepting & non-Accepting to Accepting.



	a	b
$\rightarrow * q_0$	q_1	q_3
$* q_1$	q_1	q_2
q_2	q_2	q_2
$* q_3$	q_2	q_3

Non-Deterministic Finite Automata (NFA)

A finite Automata is called NFA if there exists one or more transitions from a state on the same i/p symbol (Choices of moves)

→ It is not compulsary that all states in NFA have to consume all the i/p symbols in 'Σ'

→ Formal Definition of NFA :-

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

$Q \rightarrow$ finite set of states

$\Sigma \rightarrow$ finite set of symbols called the alphabet

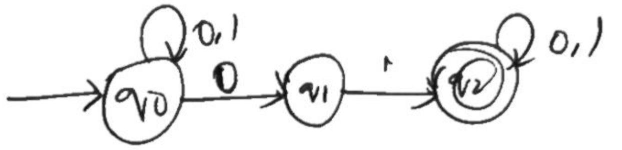
$\delta \rightarrow$ Transition fn where $\delta : Q \times \Sigma \rightarrow 2^Q$

$q_0 \rightarrow$ initial state from where any i/p is ($q_0 \in Q$)

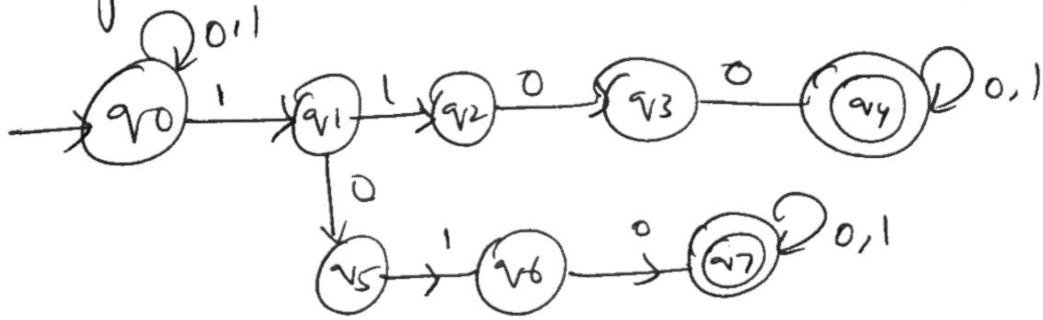
$F \rightarrow$ set of final state Q ($F \subseteq Q$)

'δ' is a Cartesian product of states & symbols giving a state in power set of Q.

eg: let w = x01y where x & y are seq of 0's & 1's



eg: construct an NFA which accept set of strings contains 1100 or 1010 as substring



i/p 01100

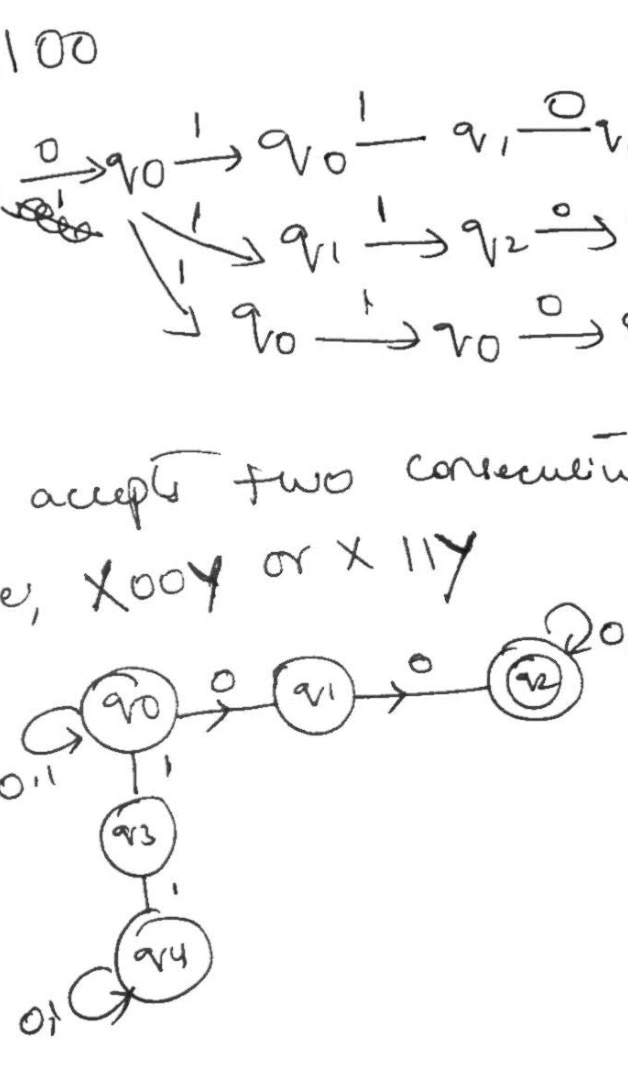
$q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_5 \xrightarrow{0} \times$

$\quad \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_4 \xrightarrow{0} q_4$

$\quad \xrightarrow{1} q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \longrightarrow q_0 \times$

eg:- String accepts two consecutive 0's or 1's

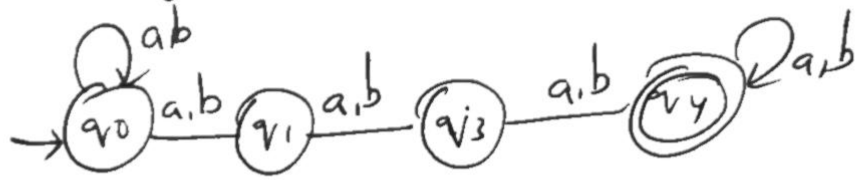
ie, x00y or x11y



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

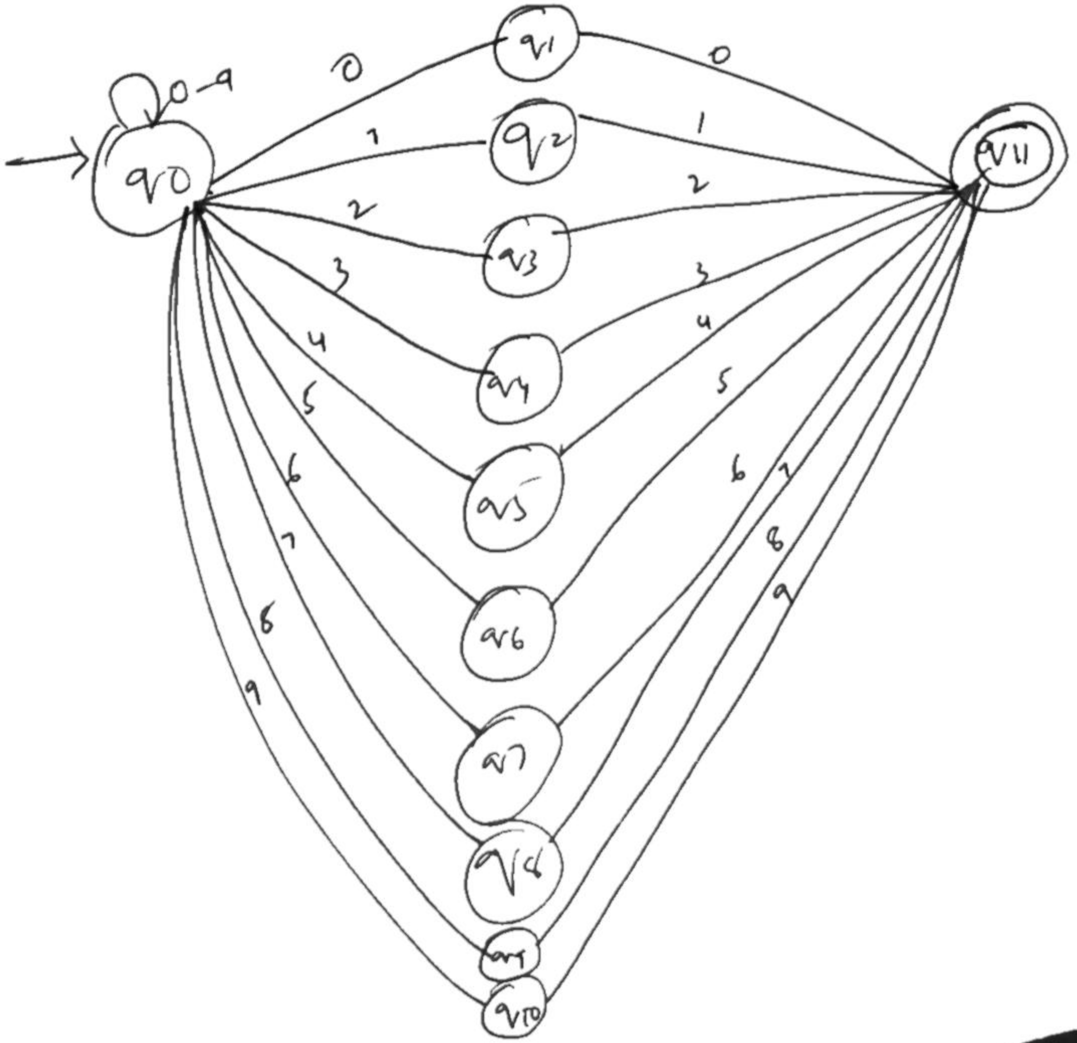
→ NFA for strings in which the 4th symbol from rightend is always 'a' over $\Sigma = \{a, b\}$

$L = \{abbb, aaaa, abab, abaabbb \ldots\}$



→ NFA for $\Sigma \{0 \ldots 9\}$ such that final digit has appeared before.

$L = \{\underline{0}1\underline{0},\ 1\underline{2}13\underline{2},\ 1\underline{2}34\underline{3}\ \}$



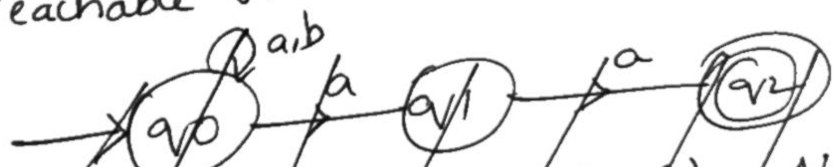
Conversion of NFA to DFA :-

→ A Language that Can be accepted by a DFA Can be accepted by an NFA. Every DFA is an NFA

→ Conversion of NFA to DFA is called Subset Construction or powerset Construction.

→ Each state in the DFA is associated with a set of states in the NFA

→ The start state in the DFA corresponds to the start state of the NFA plus all states reachable via ε-transitions.



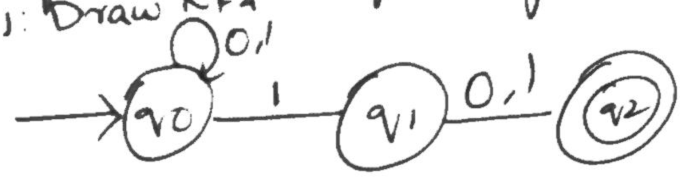
~~let M = (Q, ε, q0, δ, F) → NFA~~

~~M' = (Q'~~

Conversion of NFA to DFA

NFA of all binary strings which 2nd Last bit is '1'

Sol Step 1: Draw NFA



Step 2: first write Transition Table for NFA.

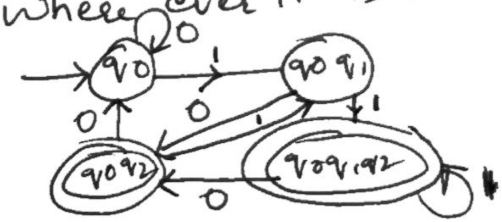
δ	0	1
$\rightarrow q_0$	q_0	$\{q_0 q_1\}$
q_1	q_2	q_2
q_2	—	—

Step 3: Now write Transition Table for DFA

$\rightarrow$ write first row as it is then Expand all states

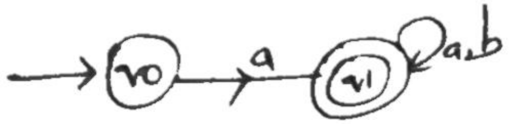
δ	0	1
$\rightarrow q_0$	q_0	$\{q_0 q_1\}$
$q_0 q_1$	$\{q_0 q_2\}$	$\{q_0 q_1 q_2\}$
$q_0 q_2$	q_0	$\{q_0 q_1\}$
$q_0 q_1 q_2$	$\{q_0 q_2\}$	$\{q_0 q_1 q_2\}$

$\rightarrow$ $\because$ q_2 is final state in NFA make q_2's final where ever it is.



DFA

→ L = {Starts with 'a'} $\Sigma = \{a, b\}$

NFA 

NFA

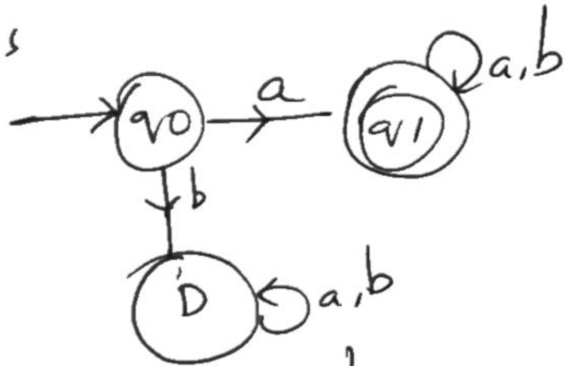
δ	a	b
→q0	q1	—
q1*	q1	q1

→ DFA

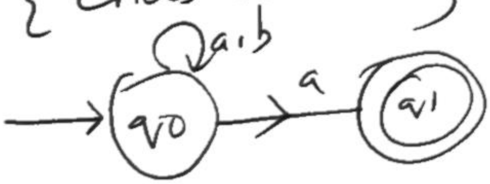
δ	a	b
→q0	q1	D
q1*	q1	q1
D	D	D

(∵ No Transition it has to redirect to dead state)

∴ DFA is



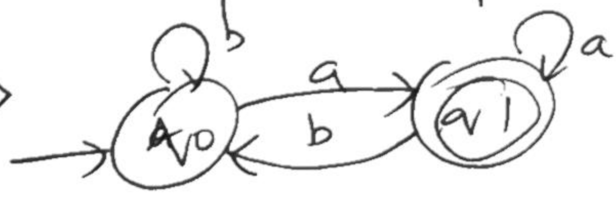
→ L = {ends with 'a'}



δ	a	b
q0	q0q1	q0
q1	—	—

→ DFA

δ	a	b
→q0	q0q1	q0
* q0q1	q0q1	q0

⇒ 

→ Convert NFA to DFA for which NFA with second symbol from LHS is 'a'

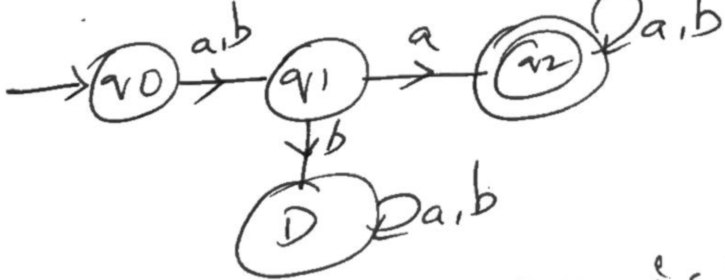
NFA

δ	a	b
→ q_0	q_1	q_1
q_1	q_2	—
* q_2	q_2	q_2

⇒

DFA

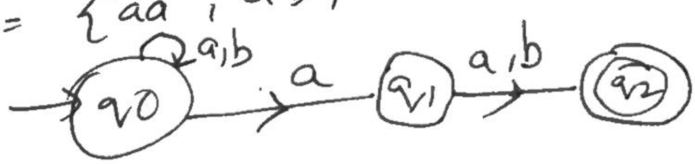
δ'	a	b
→ q_0	q_1	q_1
q_1	q_2	D
q_2	q_2	q_2
D	D	D



→ second symbol from RHS is 'a'

$L = \{aa, ab, aaa, baa, bab \ldots\}$

NFA

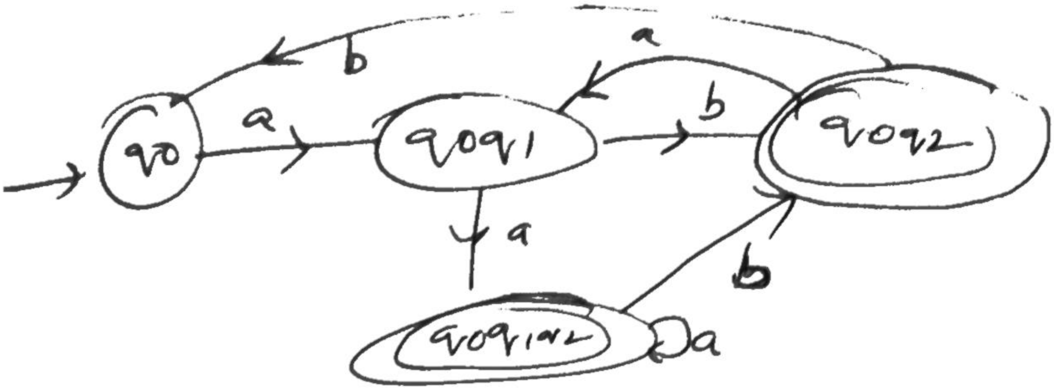


NFA

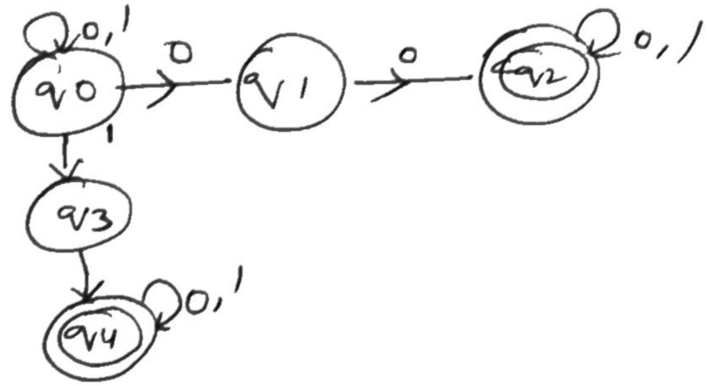
δ	a	b
→ q_0	$\{q_0 q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$
* q_2	—	—

⇒

DFA δ	a	b
→ q_0	$[q_0 q_1]$	$[q_0]$
$[q_0 q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
$[q_0 q_1 q_2]$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
$[q_0 q_2]$	$[q_0 q_1]$	$[q_0]$



→ Given NFA is



NFA

δ	0	1
→ q_0	$\{q_0q_1\}$	$\{q_0q_3\}$
q_1	$\{q_2\}$	–
q_2	$\{q_2\}$	$\{q_2\}$
q_3	–	q_4
* q_4	$\{q_1\}$	$\{q_1\}$

DFA

δ'	0	1
→ q_0	$[q_0q_1]$	$[q_3q_0]$
$[q_0q_1]$	$[q_0q_1q_2]$	$[q_0q_3]$
$[q_3q_0]$	$[q_0q_1]$	$[q_0q_2q_4]$
* $[q_0q_1q_2]$	$[q_0q_1q_2]$	$[q_0q_2q_3]$
* $[q_0q_3q_4]$	$[q_0q_1q_4]$	$[q_0q_3q_4]$
* $[q_0q_2q_3]$	$[q_0q_1q_2]$	$[q_0q_2q_3q_4]$
* $[q_0q_1q_4]$	$[q_0q_1q_2q_4]$	$[q_0q_3q_4]$
* $[q_0q_2q_3q_4]$	$[q_0q_1q_2q_4]$	$[q_0q_2q_3q_4]$
* $[q_0q_1q_2q_4]$	$[q_0q_1q_2q_4]$	$[q_0q_2q_3q_4]$