MODULE 1 [INTRODUCTION]

Why to study Automata Theory:-

-> Automata Theory is a study of abstract computing device.

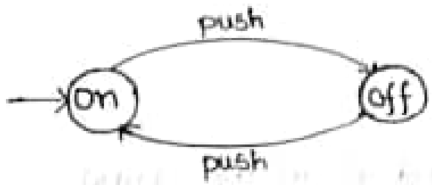
-> Finite Automata are useful model of many important kinds of hardware and software.

-> It is used in

1. Software for designing and checking the behaviour of digital circuits.
2. The LEXICAL ANALYZER of a typical compiler. The compiler component that breaks the input text into logical units such as identifiers, keywords and so on.
3. Software for scanning large bodies of text such as collection of web pages to find the occurance of words, phrases and other patterns.

Examples:-

On/off switch



then

Structural representation:-

-> There are two important notations. Those are

1. Grammar:-

* Grammars are useful models when designing software that process data with recursive structure.

Example:-

$E \rightarrow E + E$

2. Regular Expression:-

* It denotes the structure of data especially in text string.

Example:-

[A - Z][a - z]* [ ][A - Z][A - Z][A - Z]

Gcem CSE

Automata and Complexity:-

-> It is essential for study of limits of computation. There are two important issues.

1. What can computer do?

Solution:- decidability

2. What can computer do efficiently?

Solution:- intraceability

Central concepts of Automata Theory:-

1. Alphabet :-

An alphabet is a finite, non-empty set of symbols and it is represented by 'Σ'.

Example:-

$\Sigma = \{0, 1\}$ → binary alphabet

$\Sigma = \{A, B, \ldots Z\}$ → uppercase alphabet

$\Sigma = \{a, b, \ldots z\}$ → lowercase alphabet

2. Strings:-

String is a finite sequence of symbols that are chosen from some alphabets. It is represented by 'W'.

Example:-

$\Sigma = \{0, 1\}$ → binary alphabet

$W = 0110$

3. Length of string:-

It is the number of positions for the symbols in the string.

Example:-

$W = 101101 \Rightarrow |W| = 4$

4. Empty string:-

It is a string with zero occurances of symbols and it is denoted by 'ϵ' epsilon.

Examples:-

5. Powers of an alphabet:-

If 'Σ' an alphabet set of all strings of a certain length from that alphabet by using an exponential notation. We define Σ^k to be the set of strings of length k each of whose symbols is in Σ.

Example:-

$\Sigma = \{0, 1\}$ (symbols)

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

$\Sigma^1 = \{0, 1\}$ (strings)

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

$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots\ldots$

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots\ldots$

$\Sigma^* \rightarrow$ Kleen closure , $\Sigma^+ \rightarrow$ positive closure

$$\boxed{\Sigma^* = \Sigma^0 \cup \Sigma^+}$$

6. Concatenation of string:-

Let x and y be the strings, then x.y denotes the concatenation of x and y.

Example:-

x = 101
y = 110
x.y = 10110
y.x = 110101
$x.y \neq y.x$

x = 101
$x.\epsilon = 101$
$\epsilon.x = 101$
$x.\epsilon = \epsilon.x = x$
$\therefore \epsilon$ is an identity for concatenation of strings.

x = 101
$x.\emptyset = \emptyset$
$\emptyset.x = \emptyset$
$x.\emptyset = \emptyset.x = \emptyset$

7. Language:-

A set of strings all of which are choosen from some Σ^*

Example:-

1. The language of all strings consisting of n 0's (zeroes) followed by n 1's (ones).

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

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

2. The set of strings of 0's and 1's with an equal number of each

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

$L = \{\epsilon, 01, 10, 0011, 1100, 0101, 1010, 0110, 1001, \ldots\ldots\}$

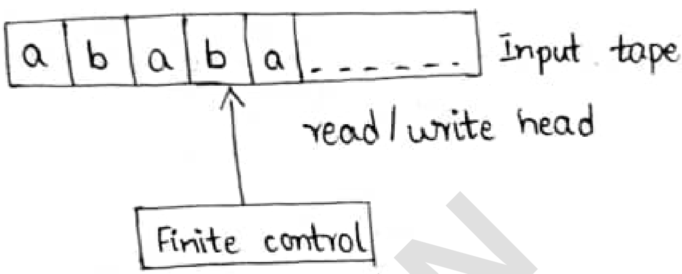
3. The set of binary numbers whose value is prime

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

$L = \{2, 3, 5, 7, 11, 13, \text{----}\}$

$L = \{00010, 11, 101, 111, 1011, \text{--------}\}$

Finite Automata



→ Finite automata can be represented using

1. Input tape :- It is a linear tape having some number of cells each input symbol placed in each cell
2. Finite control :- The finite control decides the next state on receiving the particular input from input tape.
3. Tape reader :- It reads the cells one by one from left to right and at a time only one input symbol is read.

→ There are three types of finite automata

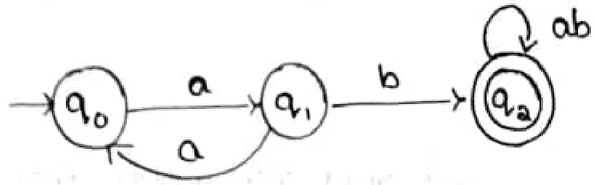
1. Determinstic finite automata (DFA)
2. Non-determinstic finite automata (NFA)
3. Epsilon - NFA (ϵ-NFA)

1. Determinstic finite automata (DFA) :-

The DFA 'A' is a five tuple notation, $A = (Q, \Sigma, \delta, q_0, F)$ where

Q represents finite set of states, Σ represents input alphabet, q_0 is a start state, F represents set of final or accepting state, δ represents it is a transition function that take two arguments such as current state and input symbols and returns a new state.

Consider the DFA



According to DFA definition

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

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

Start state :- q_0

$F = \{q_2\}$

The transition table of the above diagram is

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

$\delta(q_0, a) = q_1$

current state

new state

Simpler notations for DFA:-

There are two simpler notations to represent DFA

i) Transition diagram

ii) Transition table

Extended transition function to strings:-

If δ is our transition function then the extended transition function constructed from δ will be called as $\hat{\delta}$. The extended transition function that takes two arguments

i) current state

ii) a string (W) and returns a new state

$\hat{\delta}(q, W) = p$

$q \rightarrow$ current state

$W \rightarrow$ string

$p \rightarrow$ new state

We define $\hat{\delta}$ by induction on the length of the input string as follows.

<u>Basis</u>:- $\hat{\delta}(q, \epsilon) = q$

If we are in the state q and read no inputs then we are still in state 'q' only.

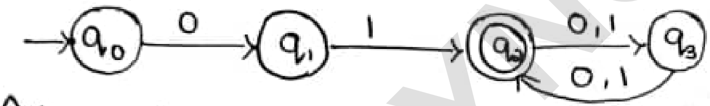
<u>Induction</u>:-

Suppose W is a string of the form xa i.e, a is a last symbol of W, and x is a string consisting of all but not last symbol

$$\hat{\delta}(q, W) = \delta(\hat{\delta}(q, x), a)$$

Consider the following DFA and check whether the DFA is accepted for a string.

1. 011101



$\hat{\delta}(q_0, \epsilon) = q_0$

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

$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_1, 1) = q_2$

$\hat{\delta}(q_0, 011) = \delta(\hat{\delta}(q_0, 01), 1) = \delta(q_2, 1) = q_3$

$\hat{\delta}(q_0, 0111) = \delta(\hat{\delta}(q_0, 011), 1) = \delta(q_3, 1) = q_2$

$\hat{\delta}(q_0, 01110) = \delta(\hat{\delta}(q_0, 0111), 0) = \delta(q_2, 0) = q_3$

$\hat{\delta}(q_0, 011101) = \delta(\hat{\delta}(q_0, 01110), 1), \delta(q_3, 1) = q_2$

The given string is accepted because q_2 is a final state.

2. 01011

$\hat{\delta}(q_0, \epsilon) = q_0$

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

$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_1, 1) = q_2$

$\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0) = \delta(q_2, 0) = q_3$

$\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1) = \delta(q_3, 1) = q_2$

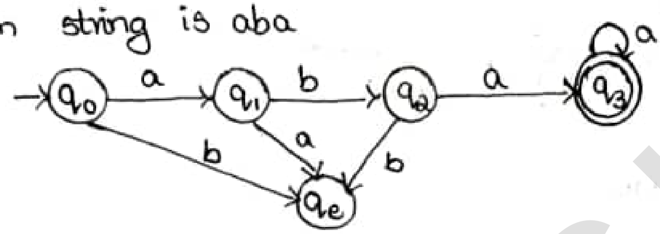
$\hat{\delta}(q_0, 01011) = \delta(\hat{\delta}(q_0, 0101), 1) = \delta(q_2, 1) = q_3$

The given string is not accepted as q_3 is not a final state

Design Problems:-

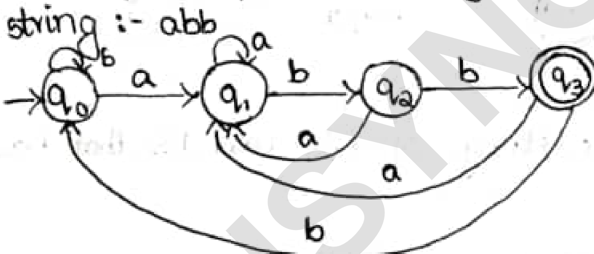
1. Design a DFA to accept the strings of a's and b's beginning with aba over the alphabet $\Sigma = \{a, b\}$

Minimum string is aba



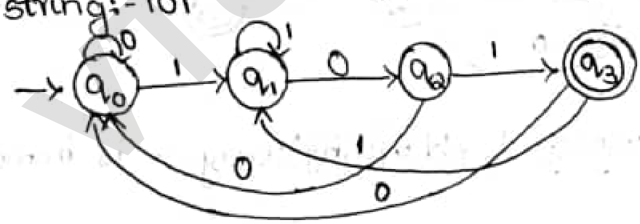
2. Design a DFA to accept the strings of a's and b's ending with abb

Minimum string :- abb



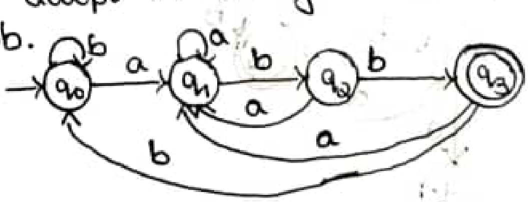
3. Design a DFA to accept the strings of 0's and 1's ending with 101

Minimum string:- 101

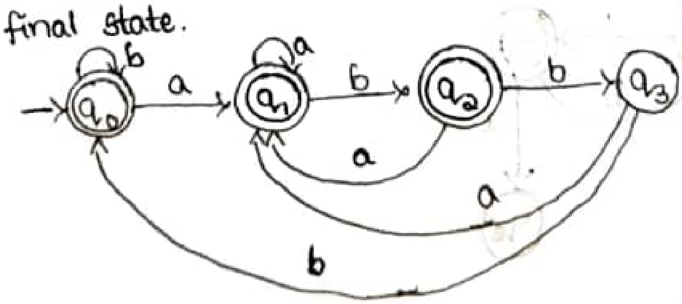


4. Design a DFA to accept the strings of a's and b's which do not end with a string abb.

Ending with abb:-

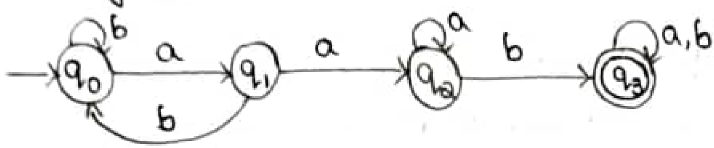


Does not end with abb:- The design is exactly same as the above, but only the final state will become as non-final state and non-final state will become as final state.



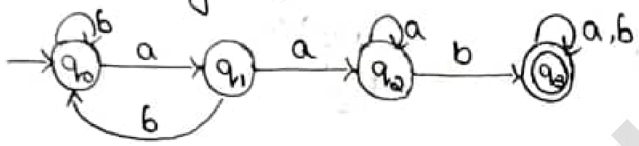
5. Design a DFA to accept the strings of a's and b's having a sub-string aab

Minimum string:- aab

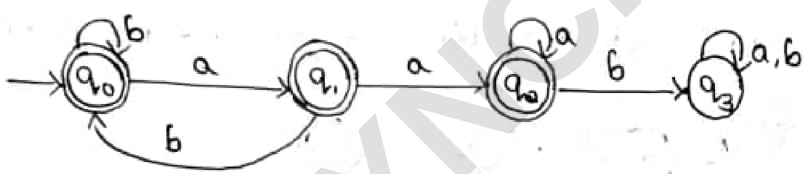


6. Design a DFA to accept the strings of a's and b's that does not have a sub-string aab.

Having a sub-string aab:-

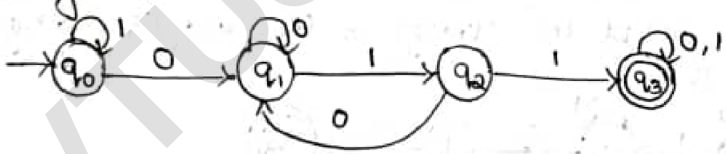


Not having a sub-string aab:-



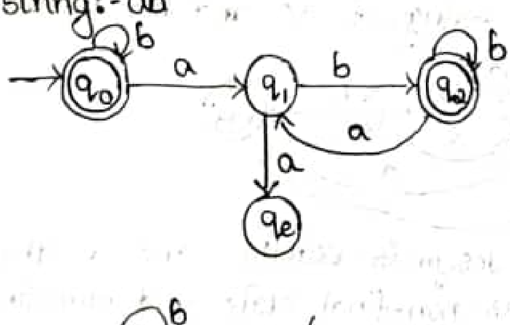
7. Design a DFA to accept the strings of 0's and 1's that has a sub-string 011

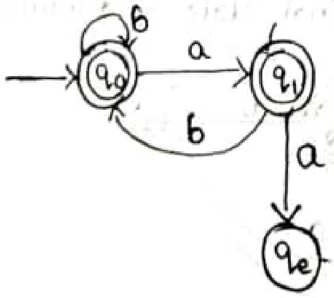
Minimum string:- 011



8. Design a DFA for the language $L=\{W \in \{a,b\}^*$: Every 'a' is immediately followed by a 'b'$\}$.

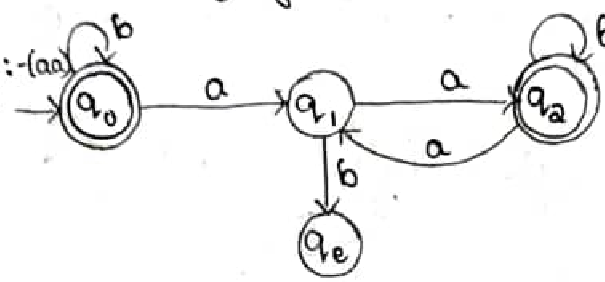
Minimum string:- ab

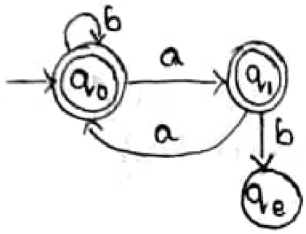




9. Design a DFA for the language $L=\{W \in \{a,b\}^* :$ Every a region in W is of even length$\}$

Minimum string :- (aa)

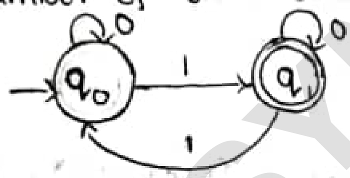




10. Design a DFA for the language $L=\{W \in \{0,1\}^* :$ W has odd parity$\}$

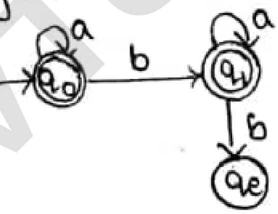
Minimum string :- 1

odd parity :- Number of one's should be odd



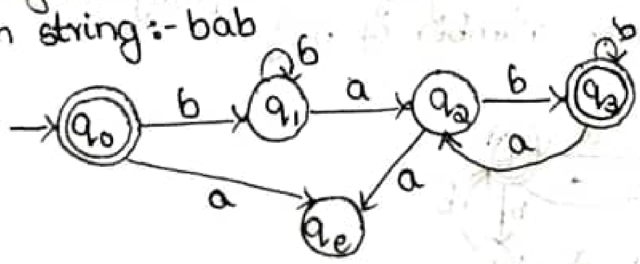
11. Design a DFA for the language $L=\{W \in \{a,b\}^* :$ W contain not more than one b$\}$

Minimum string :- b



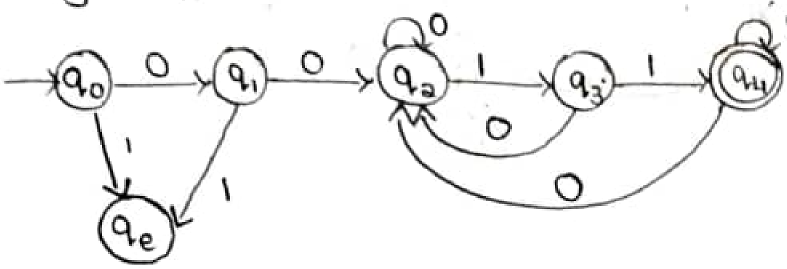
12. Design a DFA to accept the language $L=\{W \in \{a,b\}^* :$ Every a in W is immediately preceeded and followed by b$\}$.

Minimum string :- bab



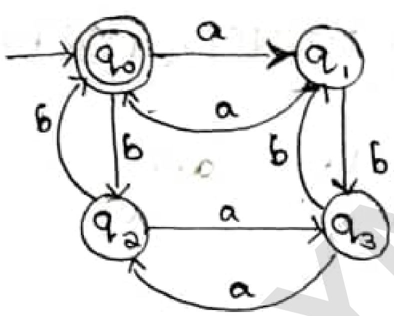
13. Design a DFA to accept strings of 0's and 1's starting with atleast two zeroes and ending with two one's

Minimum string:- 0011



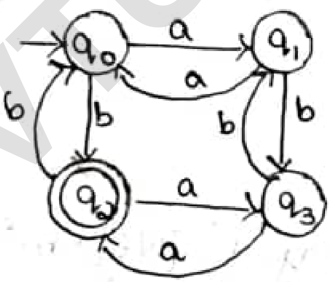
★ 14. Design a DFA for the following language $L=\{W \in \{a,b\}^* : W$ contains an even number of a's and an even number of b's$\}$

Minimum string:- aabb



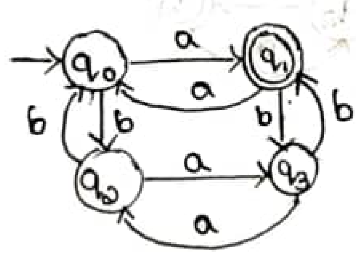
15. Design a DFA for the following language $L=\{W \in \{a,b\}^* : W$ contains an even number of a's and odd number of b's$\}$

Minimum string:- b



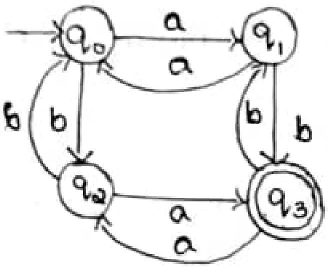
16. Design a DFA for the following language $L=\{W \in \{a,b\}^* : W$ contains an odd number of a's and even number of b's$\}$

Minimum string:- a



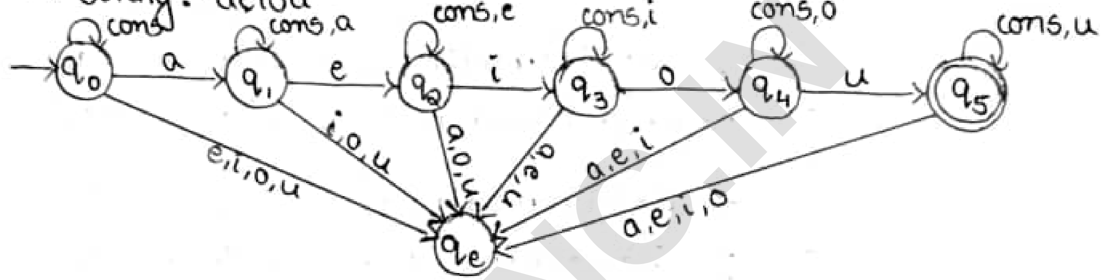
17. Design a DFA for the following language $L = \{W \in \{a,b\}^* : W$ contains an odd number of a's and odd number of b's$\}$.

Minimum string:- ab



18. Design a DFA to accept the following language $L = \{W \in \{a-z\}^* :$ all five vowels (a,e,i,o,u) occurs in W in alphabetical order$\}$.

Minimum string:- aeiou



19. Design a DFA to accept the strings from the set $\{W \in \{a,c,t\}^* : W$ is having a sub-string cat$\}$

Divisible by K- Problems

1. Draw a DFA which checks whether a binary number is divisible by 3.

$\delta(q_i, d) = q_j$

$j = (r * i + d) \bmod K$

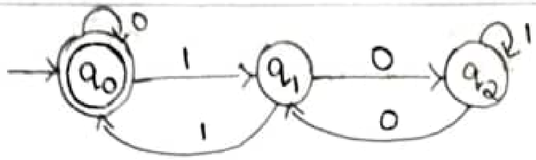
→ divisor = 3

→ digits = 0 and 1

→ remainder = 0, 1, 2

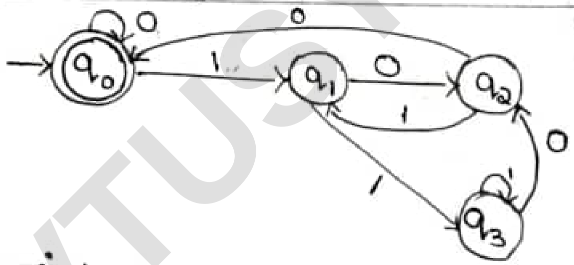
→ radix = 2

Remainder	d	$j = (2 * i + d) \bmod 3$	$\delta(q_i, d) = q_j$
0	0	$j = (2*0+0) \bmod 3 = 0$	$\delta(q_0, 0) = q_0$
	1	$j = (2*0+1) \bmod 3 = 1$	$\delta(q_0, 1) = q_1$
1	0	$j = (2*1+0) \bmod 3 = 2$	$\delta(q_1, 0) = q_2$
	1	$j = (2*1+1) \bmod 3 = 0$	$\delta(q_1, 1) = q_0$
2	0	$j = (2*2+0) \bmod 3 = 1$	$\delta(q_2, 0) = q_1$
	1	$j = (2*2+1) \bmod 3 = 2$	$\delta(q_2, 1) = q_2$

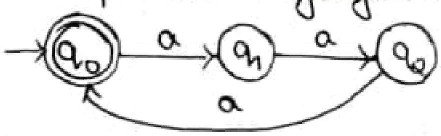


2. Draw a DFA which checks whether a binary number is divisible by 4.

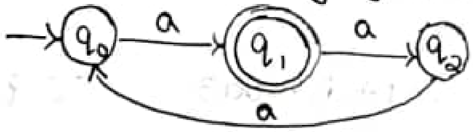
Remainder	d	$j=(2*i+d) \bmod 4$	$\delta(q_i, d)=q_j$
0	0	$j=(2*0+0) \bmod 4=0$	$\delta(q_0, 0)=q_0$
	1	$j=(2*0+1) \bmod 4=1$	$\delta(q_0, 1)=q_1$
1	0	$j=(2*1+0) \bmod 4=2$	$\delta(q_1, 0)=q_2$.
	1	$j=(2*1+1) \bmod 4=3$	$\delta(q_1, 1)=q_3$
2	0	$j=(2*2+0) \bmod 4=0$	$\delta(q_2, 0)=q_0$
	1	$j=(2*2+1) \bmod 4=1$	$\delta(q_2, 1)=q_1$
3	0	$j=(2*3+0) \bmod 4=2$	$\delta(q_3, 0)=q_2$
	1	$j=(2*3+1) \bmod 4=3$	$\delta(q_3, 1)=q_3$



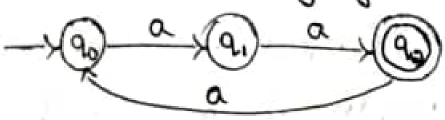
3. Design a DFA to accept the language $L=\{W : |W| \bmod 3=0, \Sigma=\{a\}\}$



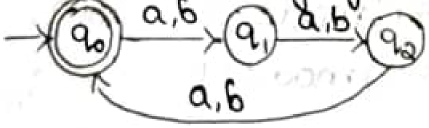
4. Design a DFA to accept the language $L=\{W : |W| \bmod 3=1, \Sigma=\{a\}\}$



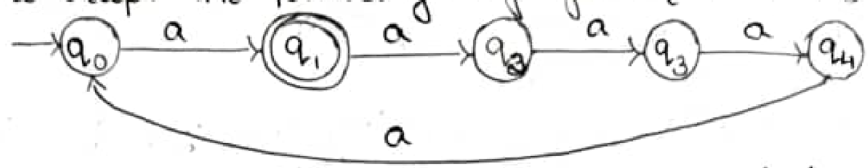
5. Design a DFA to accept the language $L=\{W : |W| \bmod 3=2, \Sigma=\{a\}\}$.



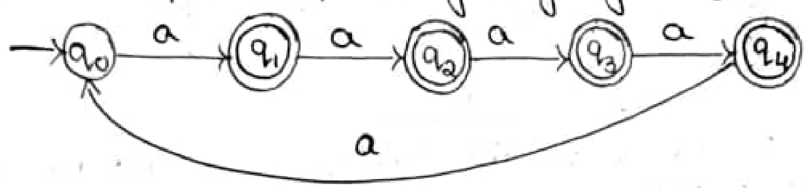
6. Design a DFA to accept the language $L=\{W : |W| \bmod 3=0, \Sigma=\{a,b\}\}$



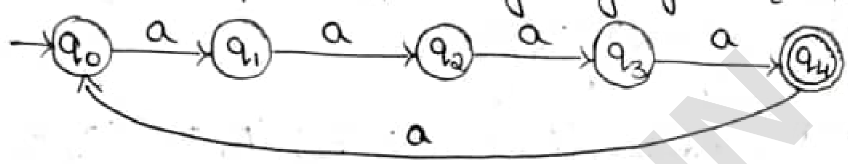
7. Design a DFA to accept the following language $L=\{W : |W| \bmod 5 = 1, \Sigma = \{a\}\}$



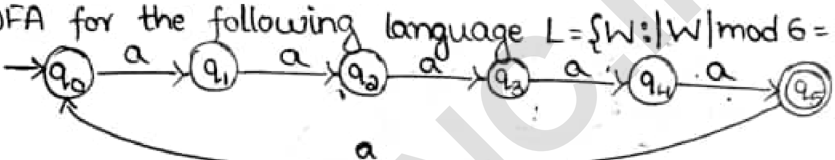
8. Design a DFA to accept the following language $L=\{W : |W| \bmod 5 \neq 0, \Sigma=\{a\}\}$



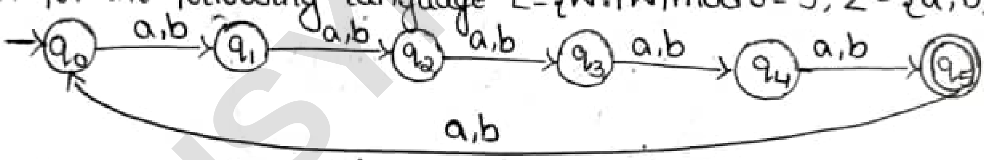
9. Design a DFA to accept the following language $L=\{W : |W| \bmod 5 = 4, \{a\}=\Sigma\}$



10. Design a DFA for the following language $L=\{W : |W| \bmod 6 = 3, \Sigma=\{a\}\}$



11. Design a DFA for the following language $L=\{W : |W| \bmod 6 = 3, \Sigma = \{a,b\}\}$



12. Design a DFA to accept the following language $L = \{W :$

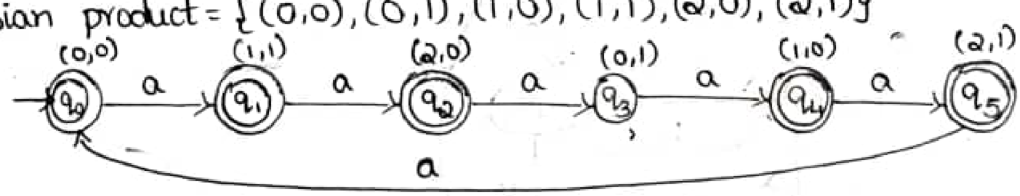
a) $|W| \bmod 3 \geq |W| \bmod 2, \Sigma=\{a\}\}$

b) $|W| \bmod 3 \neq |W| \bmod 2, \Sigma = \{a\}\}$

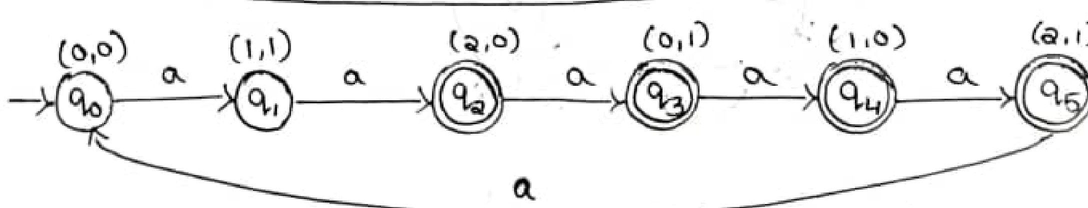
a) Remainder of mod 3 = 0, 1, 2

Remainder of mod 2 = 0, 1

Cartesian product = {(0,0), (0,1), (1,0), (1,1), (2,0), (2,1)}

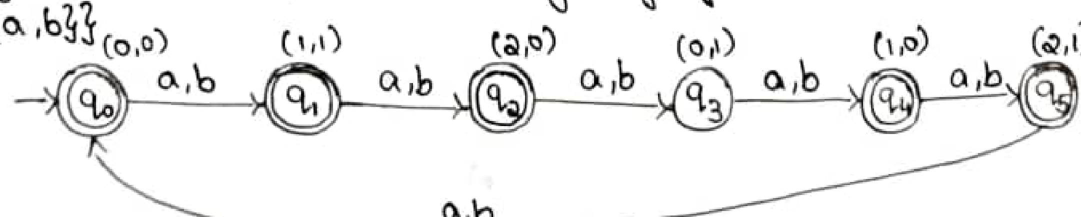


b)



13. Design a DFA to accept the following language $L=\{W : |W| \bmod 3 \geq |W| \bmod 2,$

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

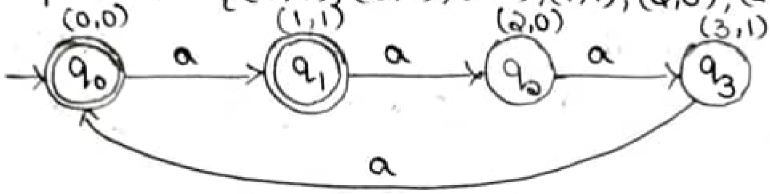


14. Design a DFA to accept the language $L = \{W : |W| \bmod 4 \le |W| \bmod 2, \Sigma = \{a\}\}$

Remainder of mod 4 = 0, 1, 2, 3

Remainder of mod 2 = 0, 1

Cartesian product = {(0,0), (0,1), (1,0), (1,1), (2,0), (2,1), (3,0), (3,1)}



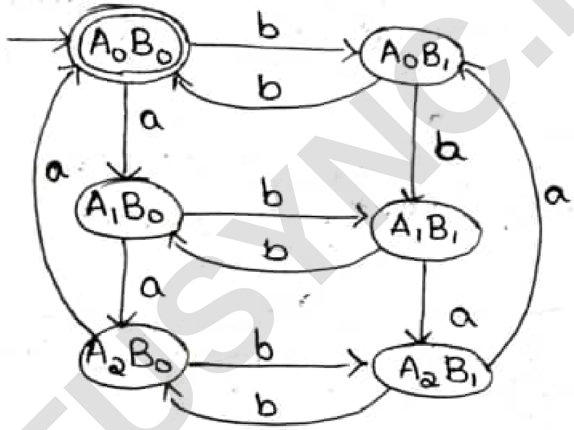
15. Design a DFA to accept the strings of a's and b's such that

i) $L = \{W : W \in \{a,b\}^* \; N_a(W) \bmod 3 = 0 \text{ and } N_b(W) \bmod 2 = 0\}$

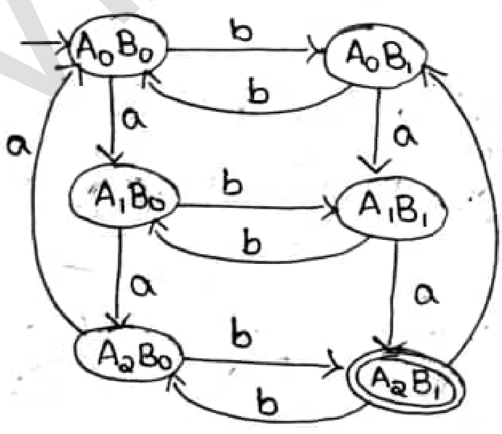
mod 3 remainder = (0, 1, 2) A_0, A_1, A_2

mod 2 remainder = (0, 1) B_0, B_1

Cartesian product = $\{(A_0,B_0), (A_0,B_1), (A_1,B_0), (A_1,B_1), (A_2,B_0), (A_2,B_1)\}$



ii) $L = \{W \in W \in \{a,b\}^* \; N_a(W) \bmod 3 = 2 \text{ and } N_b(W) \bmod 2 = 1\}$

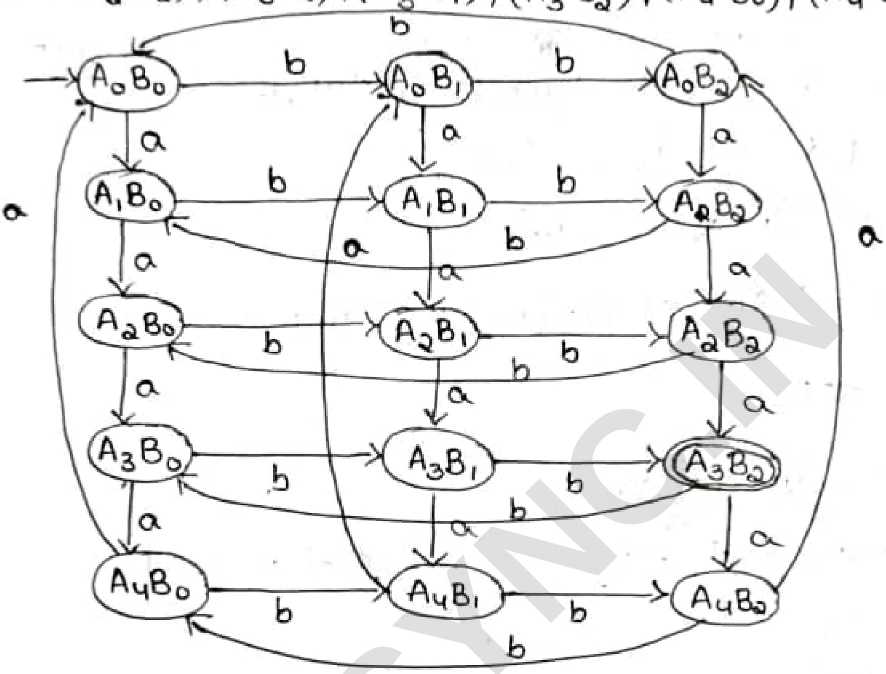


16. Design a DFA to accept a's and b's such that $L = \{W : W \in \{a,b\}^*$ $N_a(W) \bmod 5 = 3$ and $N_b(W) \bmod 3 = 2\}$

mod5 remainder = (0,1,2,3,4) A_0, A_1, A_2, A_3, A_4

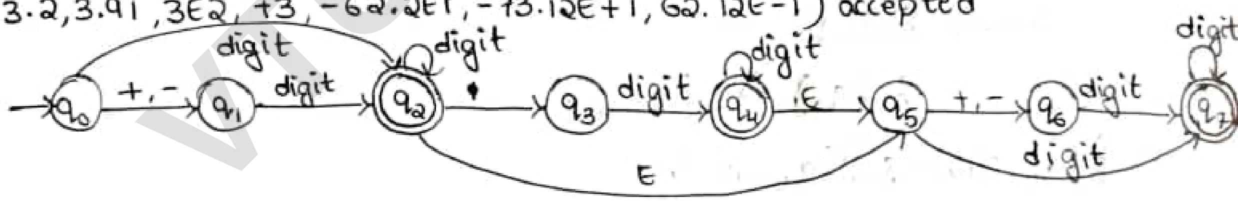
mod3 remainder = (0,1,2) B_0, B_1, B_2

cartesian product = $\{(A_0B_0), (A_0B_1), (A_0B_2), (A_1B_0), (A_1B_1), (A_1B_2), (A_2B_0)$ $(A_2B_1), (A_2B_2), (A_3B_0), (A_3B_1), (A_3B_2), (A_4B_0), (A_4B_1), (A_4B_2)\}$



17. Design a DFA for the following language $L = \{W : W$ is a string representation of floating point numbers$\}$.

(3.2, 3.91, 3E2, +3, -62.2E1, -73.12E+1, 62.12E-1) accepted



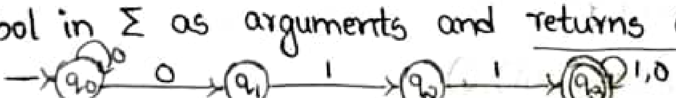
The Language of DFA:-

We can define the language of DFA $A = (Q, \Sigma, \delta, q_0, F)$. The language is denoted by $L(A) = \{W \mid \hat{\delta}(q_0, W)$ is in $F\}$.

NFA (

NFA is a five tuple notation $A = (Q, \Sigma, \delta, q_0, F)$ where Q, Σ, q_0, F are same as DFA and δ is a transition function that takes a state in Q and input symbol in Σ as arguments and returns a subset of Q.

Example:-



δ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\{q_3\}$
$*q_3$	$\{q_3\}$	$\{q_3\}$

Extended transition function in NFA:-

Let $M = \{Q, \Sigma, \delta, q_0, F\}$ be an NFA. The extended transition function '$\hat{\delta}$' is defined as follows

<u>Basis</u>:- $\hat{\delta}(q, \epsilon) = \{q\}$

Induction:-

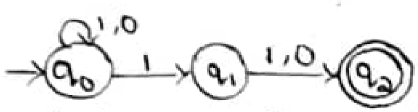
Suppose 'W' is of the form xa where 'a' is a last symbol of W and x is the rest of the symbols in 'W', then

$$\hat{\delta}(q, x) = \{p_1, p_2, \ldots p_k\}$$

Let $\bigcup_{i=1}^{k} \delta(q_i, a) = \{r_1, r_2, \ldots r_m\}$

then, $\hat{\delta}(q, W) = \delta(\hat{\delta}(q, x), a) = \{r_1, r_2, \ldots r_m\}$

1. Consider the NFA



Check whether the given string 01010 is satisfied or not

$\hat{\delta}(q_0, \epsilon) = \{q_0\}$

$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0)$
$= \delta(\{q_0\}, 0) = \{q_0\}$

$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1)$
$= \delta(\{q_0\}, 1) = \{q_0, q_1\}$

$\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0)$
$= \delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \{q_0, q_2\}$

$\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1)$
$= \delta(\{q_0, q_2\}, 1) = \delta(q_0, 1) \cup \delta(q_2, 1)$
$= \{q_0, q_1\}$

$\hat{\delta}(q_0, 01010) = \delta(\hat{\delta}(q_0, 0101), 0)$
$= \delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \{q_0, q_2\}$

$\therefore$ The given string is accepted

ii) 10100

$\hat{\delta}(q_0,\epsilon) = \{q_0\}$

$\hat{\delta}(q_0,1) = \delta(\hat{\delta}(q_0,\epsilon),1)$

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

$\hat{\delta}(q_0,10) = \delta(\hat{\delta}(q_0,1),0)$

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

$= \{q_0,q_2\}$

$\hat{\delta}(q_0,101) = \delta(\hat{\delta}(q_0,10),1)$

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

$= \{q_0,q_1\}$

$\hat{\delta}(q_0,1010) = \delta(\hat{\delta}(q_0,101),0)$

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

$= \{q_0,q_2\}$

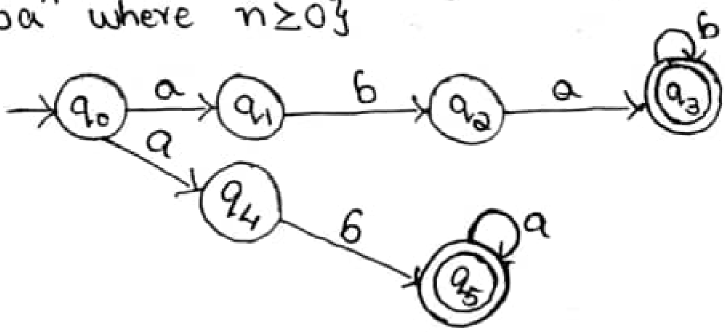
$\hat{\delta}(q_0,10100) = \delta(\hat{\delta}(q_0,1010),0)$

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

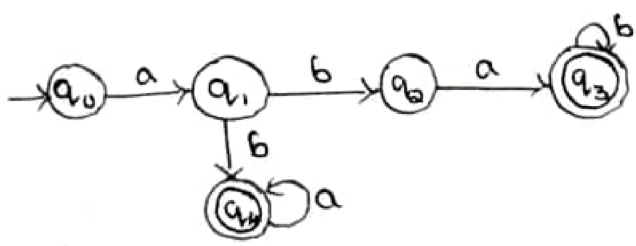
$= \{q_0\}$

$\therefore$ The given string is rejected. because q_0 is not the final state.

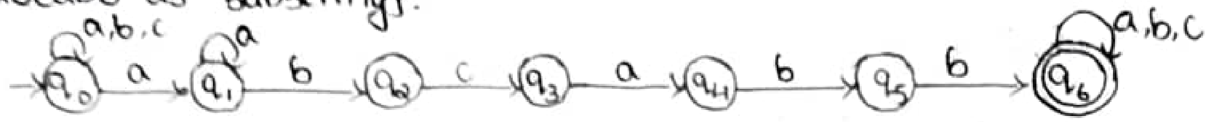
2. Obtain a NFA to accept the following language $L = \{W \in \{a,b\}^* : W$ contains $abab^n$ (or) aba^n where $n \geq 0\}$



(or)



3. Draw a NFA for the following language $L = \{W \in \{a,b,c\}^* : W$ contains abcabb as substring$\}$.



Equivalence of NFA and DFA:-

NFA can be converted into two methods:

i) Subset construction

ii) Lazy method

1. Subset construction:-

Given an NFA $M = (Q_n, \Sigma, \delta_N, q_0, F_N)$ which accepts the language $L(M_n)$. we can find an equivalent DFA $M_D = (Q_D, \Sigma, \delta_D, q_0, F)$ such that $L(M_D) = L(M_N)$

Step-1:- Identify the start state of DFA as q_0

Step-2:- Identify the alphabets. Σ is same for both NFA and DFA.

Step-3:- Identify the Q_D (set of states for DFA). The set of subsets of Q_N will be the states of DFA of Q_D.

For example:- Q_N has n states then Q_D will have 2^n states.
If $n=3$, DFA will have 8 states.

Step-4:- Identify the transitions of DFA i.e, δ_D for each state $\{P_1, P_2 ---- P_k\}$ is Q_D and for each input symbol 'a' in Σ, then the transitions can be obtained as below:

$$\delta_D(\{P_1, P_2, --- P_k\}, a) = \delta_N(P_1, a) \cup \delta_N(P_2, a) \cup ----- \delta_N(P_K, a)$$

Convert the following NFA to DFA using subset construction method and Lazy method.

NFA transition table

δ_N	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
$* q_2$	ϕ	ϕ

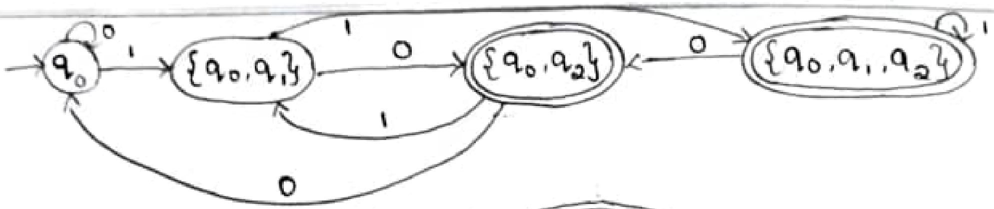
DFA transition table

δ_D	0	1
ϕ	ϕ	ϕ
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
$* q_2$	ϕ	ϕ
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$* \{q_0, q_2\}$	q_0	$\{q_0, q_1\}$
$* \{q_1, q_2\}$	q_2	q_2
$* \{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$

NFA to DFA

Start state:- q_0

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



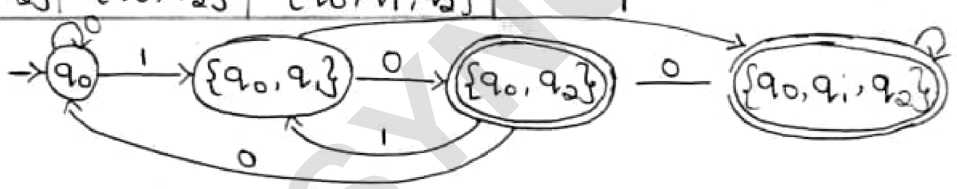
Since $q_1, q_2, \{q_1, q_2\}$ are disconnected because it is not reachable from the start state, so these states can be neglected.

Lazy method:-

Start state:- q_0

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

δ_D	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	$\{q_0, q_1\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$



1. Convert the following NFA to DFA

NFA transition table:-

δ_N	0	1
$\rightarrow p$	$\{p, q\}$	p
q	$\emptyset$	r
$*r$	$\{r, p\}$	q

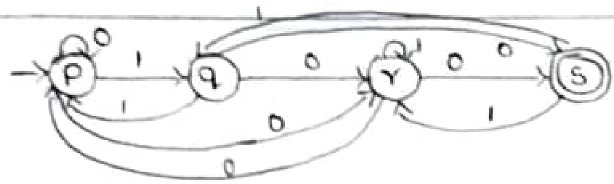
Start state:- p

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

DFA transition table

δ_D	0	1
$\rightarrow p$	$\{p, q\}$	p
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$
$*\{p, r\}$	$\{p, q, r\}$	$\{p, q\}$
$*\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$

2.

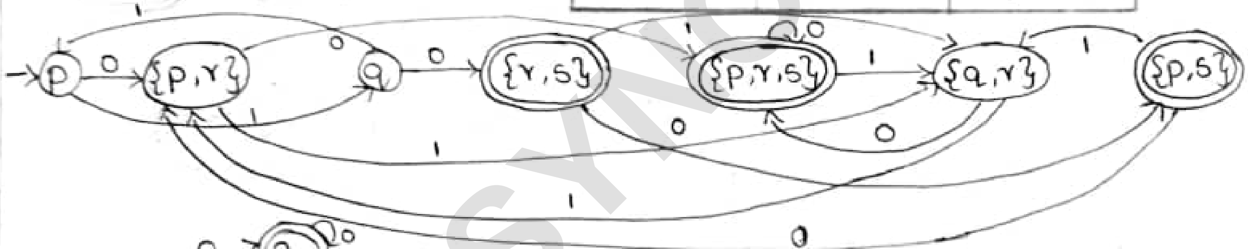


NFA transition table

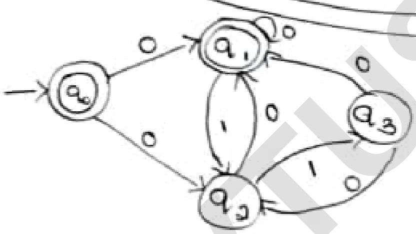
δ_N	0	1
$\rightarrow p$	$\{p, r\}$	q
q	$\{r, s\}$	p
r	$\{p, s\}$	r
$* s$	$\emptyset$	$\{r, q\}$

DFA transition table

δ_D	0	1
$\rightarrow p$	$\{p, r\}$	q
$\{p, r\}$	$\{p, r, s\}$	$\{q, r\}$
q	$\{r, s\}$	p
$*\{p, r, s\}$	$\{p, r, s\}$	$\{q, r\}$
$\{q, r\}$	$\{p, r, s\}$	$\{p, r\}$
$*\{r, s\}$	$\{p, s\}$	$\{r, q\}$
$*\{p, s\}$	$\{p, r\}$	$\{q, r\}$



3.

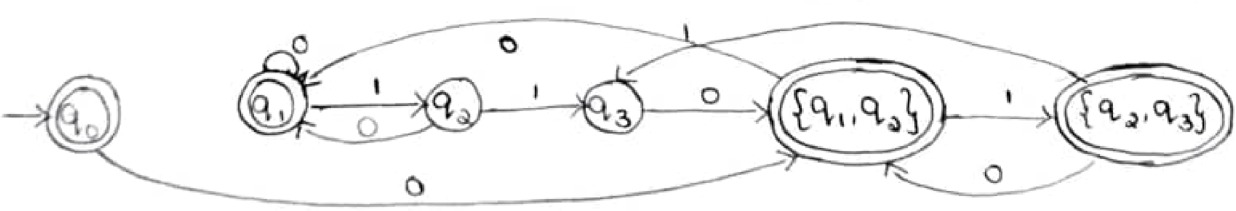


NFA transition table:-

δ_N	0	1
$\rightarrow * q_0$	$\{q_1, q_2\}$	$\emptyset$
q_1	q_1	q_2
q_2	q_1	q_3
q_3	$\{q_1, q_2\}$	$\emptyset$

DFA transition table:-

δ_D	0	1
$\rightarrow * q_0$	$\{q_1, q_2\}$	$\emptyset$
$*\{q_1, q_2\}$	q_1	$\{q_2, q_3\}$
$\{q_2, q_3\}$	$\{q_1, q_2\}$	q_3
$* q_1$	q_1	q_2
q_2	q_1	q_3
q_3	$\{q_1, q_2\}$	$\emptyset$

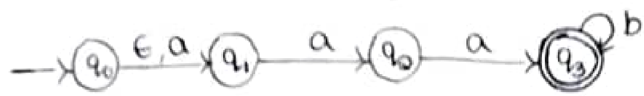


ϵ-NFA:-

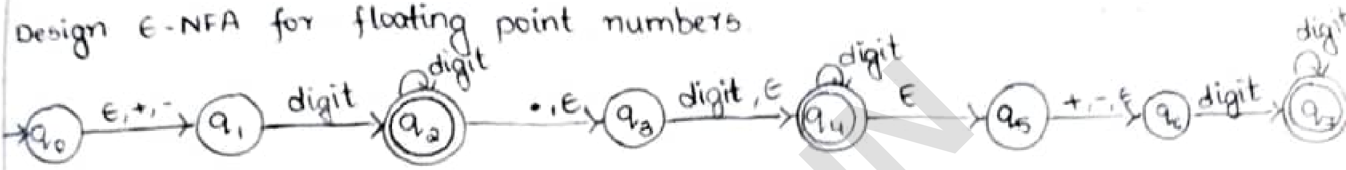
Let $A = (Q, \Sigma, \delta, q_0, F)$, here, q_0, Q, F, Σ are same as NFA. The δ takes two arguments i.e, current state from Q and input symbol i.e, $\Sigma \cup \{\epsilon\}$ and returns sets of states.

Example:-

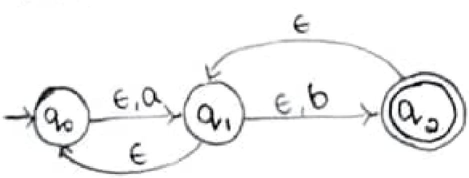
Design an ϵ-NFA for the following language $L = \{W \in \{a,b\}^* : W$ is made up of an optional 'a' followed by 'aa' followed by zero or more number of b's$\}$



Design ϵ-NFA for floating point numbers



ϵ-closures:-

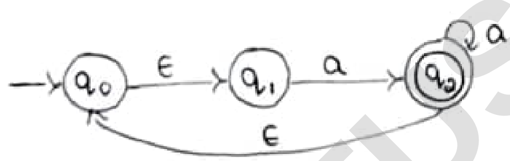


ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

ϵ closure $(q_1) = \{q_1, q_2, q_0\}$

ϵ-closure $(q_2) = \{q_2, q_1, q_0\}$

1)

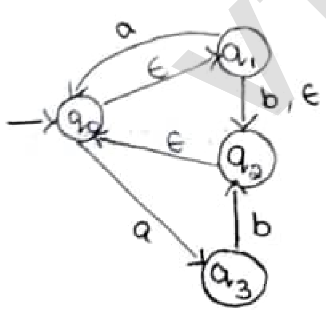


ϵ-closure $(q_0) = \{q_0, q_1\}$

ϵ-closure $(q_1) = \{q_1\}$

ϵ-closure $(q_2) = \{q_2, q_0, q_1\}$

2)



ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

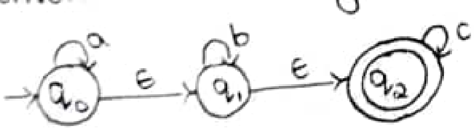
ϵ-closure $(q_1) = \{q_1, q_2, q_0\}$

ϵ-closure $(q_2) = \{q_2, q_0, q_1\}$

ϵ-closure $(q_3) = \{q_3\}$

Conversion from ϵ-NFA to DFA:-

1. Convert the following ϵ-NFA to DFA



1. ϵ-closures:-

ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

ϵ-closure $(q_1) = \{q_1, q_2\}$

ϵ-closure $(q_2) = \{q_2\}$

2. ϵ-NFA transition table:-

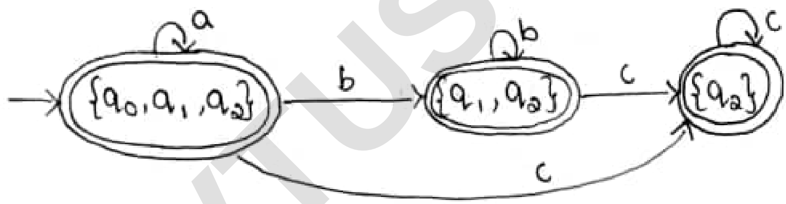
δ	ϵ	a	b	c
$\rightarrow q_0$	$\{q_1\}$	$\{q_0\}$	$\emptyset$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$	$\{q_1\}$	$\emptyset$
$* q_2$	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_2\}$

3. Start state of DFA = ϵ-closure(start state ϵ-NFA)
Start state of DFA = ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

4. DFA transition table

δ_D	a	b	c
$\rightarrow * \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$* \{q_1, q_2\}$	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
$* \{q_2\}$	$\emptyset$	$\emptyset$	$\{q_2\}$

ϵ-closure $(q_0 \cup \emptyset \cup \emptyset) = \epsilon$-closure $(q_0$



Theorem

If $D = (Q_D, \Sigma, \delta_D, q_0, F_D)$ is the DFA constructed from NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N$ by the subset construction then $L(D) = L(N)$

Proof:-

We have to prove that $\hat{\delta}_D\{q_0, w\} = \hat{\delta}_N\{q_0, w\}$

Basis:-

Let $|w| = 0$, i.e, $w = \epsilon$, then $\hat{\delta}_D(\{q_0\}, \epsilon) = \hat{\delta}_N(\{q_0\}, \epsilon) = \{q_0\}$

Induction:-

Let w be of length $n+1$ and assume the statement for length n. Break w as xa where a is the final symbol of w and x is rest of string in w. By the inductive hypothesis $\hat{\delta}_D(\{q_0\}, x) = \hat{\delta}_N(\{q_0\}, x)$
$\{P_1, P_2, \ldots P_K\}$

The inductive part of definition of $\hat{\delta}$ for NFA is

$$\hat{\delta}_N(\{q_0\},w) = \bigcup_{i=1}^{k} \delta_N(p_i,a) \rightarrow \text{equation-1}$$

The subset construction on the other hand will give

$$\delta_D(\{p_1,p_2,\ldots\ldots p_k\},a) = \bigcup_{i=1}^{k} \delta_N(p_i,a) \rightarrow \text{equation-2}$$

Now let us use equation-2

$\hat{\delta}_D(\{q_0\},x) = \{p_1,p_2\ldots\ldots p_k\}$ in the inductive part of definition of $\hat{\delta}$ for DFA

$$\hat{\delta}_D(\{q_0\},w) = \delta_D(\hat{\delta}(q_0,w),a) = \delta_D(\delta\{p_1,p_2\ldots\ldots p_k\})$$

$$\hat{\delta}_D(\{q_0\},w) = \bigcup_{i=1}^{k} \delta_N(p_i,a) \rightarrow \text{equation-3}$$

Compare equation-1 and equation-3, we get

$$\hat{\delta}(\{q_0\},w) = \hat{\delta}_N(\{q_0\},w)$$

Eliminating

Step-1:- If q_0 is the start state of NFA then ϵ-closure(q_0) is the start state of DFA

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

Step-2:- Compute the transitions for DFA.

Let $\{p_1,p_2,\ldots\ldots p_k\}$ is the state in DFA $\delta_D(\{p_1,p_2,\ldots\ldots p_k\},a)$ is computed as follows.

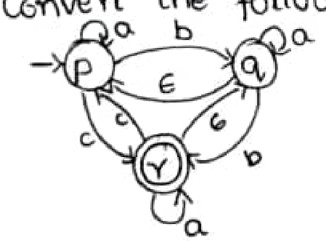
1) Let $\delta_\epsilon(\{p_1,p_2\ldots\ldots p_k\},a) = \{r_1,r_2,r_3\ldots\ldots r_n\}$

2) Then take ϵ-closure$(\{r_1,r_2,\ldots\ldots r_n\})$

Step-3:- If $\{p_1,p_2,\ldots\ldots p_k\}$ is a state in DFA and if the set contains atleast one final state of ϵ-NFA then $\{p_1,p_2,\ldots\ldots p_k\}$ is a final state of DFA

Convert the following ϵ-NFA into DFA

1.



1. ϵ-closure$(p) = \{p\}$

ϵ-closure$(q) = \{q,p\}$

ϵ-closure$(r) = \{r,q,p\}$

2.

δ	ϵ	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
$*\, r$	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

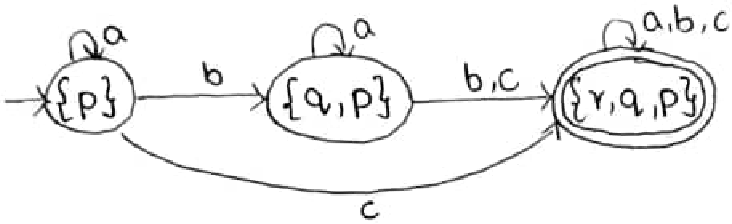
3. Start state of DFA =

ϵ-closure(p)

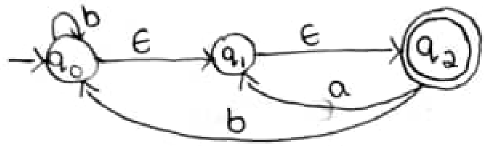
$= \{p\}$

4. DFA transition table

δ_D	a	b	c
$\rightarrow \{p\}$	$\{p\}$	$\{q,p\}$	$\{r,q,p\}$
$\{q,p\}$	$\{q,p\}$	$\{p,q,r\}$	$\{r,q,p\}$
$* \{r,q,p\}$	$\{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$



2.



1. ϵ-closure$(q_0) = \{q_0, q_1, q_2\}$
   ϵ-closure$(q_1) = \{q_1, q_2\}$
   ϵ-closure$(q_2) = \{q_2\}$

2. ϵ-NFA transition table

δ_N	ϵ	a	b
$\rightarrow q_0$	$\{q_1\}$	ϕ	$\{q_0\}$
q_1	$\{q_2\}$	ϕ	ϕ
$* q_2$	ϕ	$\{q_1\}$	$\{q_0\}$

3. Start state of DFA = ϵ-closure(q_0)
$= \{q_0, q_1, q_2\}$

4. DFA transition table

δ_D	a	b
$\rightarrow * \{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$* \{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$