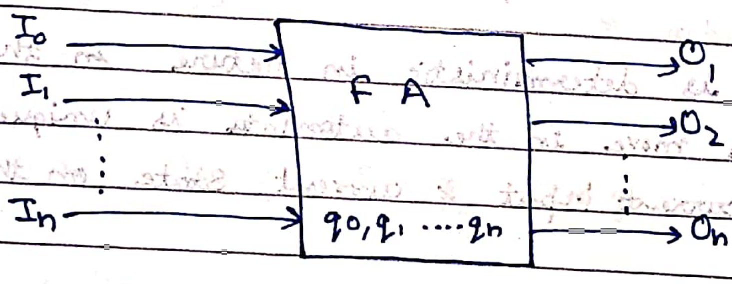
Unit: 1 :

Introduction

Finite Automata: A finite state automata is used to recognize a formal language called regular language. A formal language can be defined as a set of strings accepted by a machine.
A finite state automata consist of finite number of states. Therefore, it is also called a finite state machine. A finite automata is a collection of five tuples.

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

$Q \rightarrow$ finite no. of states
$\Sigma \rightarrow$ No. of input symbols
$\delta \rightarrow$ Transition function or mapping function $\delta(Q, \Sigma) \rightarrow Q$
$q_0 \rightarrow$ Set of starting state
$F \rightarrow$ No. of final state



Automata: is an abstract model of digital computer and Automata has a mechanism to read from input tape. Any language is recognized by some language. Hence, automata are basically language acceptor or language recognizer.

Applications of Automata Theory:

1. Automata theory is a base for the formal language and these formal language are useful in programming languages.

2. Automata theory plays an important role in compiler design.
3. To prove the correctness of the program, we use automata theory.
4. In switching theory and design & analysis of digital circuits, automata theory is applied.
5. Automata theory deals with the design finite state machine.

Finite Automata

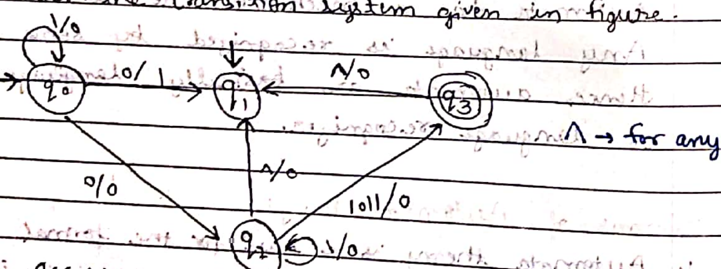
- Deterministic finite Automata (DFA)
- Non-Deterministic finite Automata (NDFA)

~~Consider~~ *

DFA: is deterministic in nature in the sense that each move in the automata is uniquely determined on current input & current state on the other hand.

NDFA: is non-deterministic in nature.

Q. Consider the transition system given in figure.



Λ → for any

accept the 1011011 & 10101 is it accepted by the machine or not?

Sol. 1011011 $\rightarrow q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{1011} q_3^*$

This string is accepted by machine.

10101 $\rightarrow q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} X$.

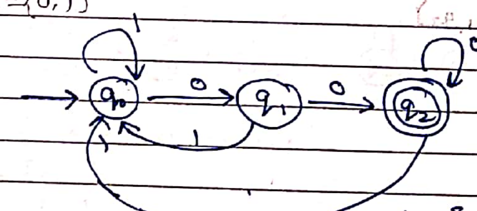
This string is not accepted by machine.

Q. Consider the finite state machine whose transition function δ is given in the table in the form of transition table. Gives the entire sequence of states for the input string 110001.

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

$\rightarrow q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0 \xrightarrow{0} q_2 \xrightarrow{0} q_0 \xrightarrow{0} q_2 \xrightarrow{1} q_3$

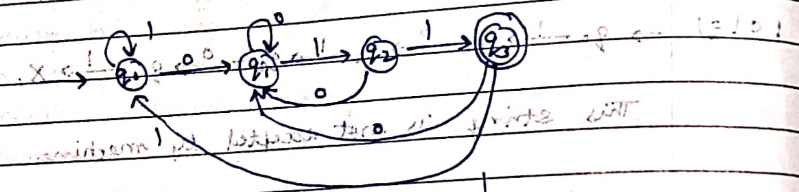
Q. Design a finite automata that accepts set of strings such that every string ends with 00. over alphabet $\Sigma = \{0, 1\}$



$L = \{00, 100, 000, 0000, 0100, \ldots\}$

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
$F = \{q_2\}$

Q. Design a finite automata that accepts set of strings such that every string using alphabet $\Sigma=\{0,1\}$ end with 011.

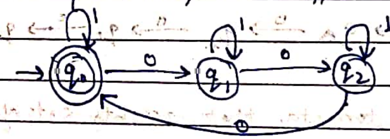


$Q=\{q_0, q_1, q_2, q_3\}$
$\Sigma=\{0,1\}$
$q_0=\{q_0\}$
$F=\{q_3\}$

δ	0	1
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_1	q_0

Q. Construct the finite automata that accepts sets of strings where the number of zeros in every string is multiple of 3. over $\Sigma=\{0,1\}$.

Sol. $L=\{\epsilon, 11111, 1011100011, \ldots$

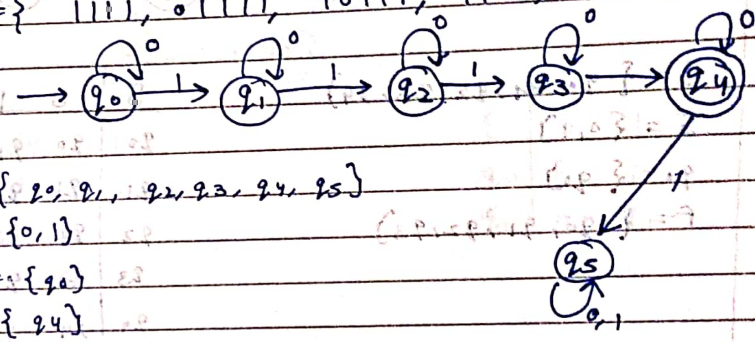


$Q=\{q_0, q_1, q_2\}$
$\Sigma=\{0,1\}$
$q_0=\{q_0\}$
$F=\{q_0\}$

δ	0	1
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

Q. Design a DFA which accepts set of string that containing exactly 1111's in every string over alphabet $\Sigma=\{0,1\}$

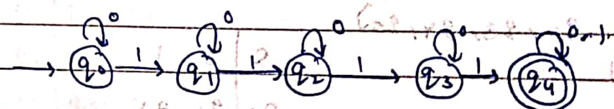
Sol. $L=\{1111, 01111, 10111, 11011, 11110, ---$



$Q=\{q_0, q_1, q_2, q_3, q_4, q_5\}$
$\Sigma=\{0,1\}$
$q_0=\{q_0\}$
$F=\{q_4\}$

δ	0	1
q_0	q_0	q_1
q_1	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_5

Q. Design a DFA which accepts set of string that containining, more than 4 ones in every string over alphabet $\Sigma=\{0,1\}$



$Q=\{q_0, q_1, q_2, q_3, q_4\}$
$\Sigma=\{0,1\}$
$q_0=\{q_0\}$
$F=\{q_4\}$

δ	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_3	q_4
q_4	q_4	q_4

$L = \{ w / n_1(w) < 4 \mid w \in (0,1)^* \}$

Q. No. of ones' less than 4.

$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

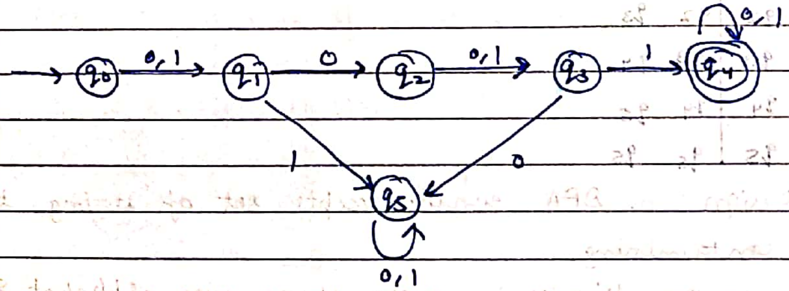
$\Sigma = \{0,1\}$

$q_0 = \{q_0\}$

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

δ	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_3
q_3	q_3	q_4
q_4	q_4	q_4

Q. Design a DFA whose second input is 0 and 4th input is 1. over alphabet $\Sigma = \{0,1\}$



$Q = \{ q_0, q_1, q_2, q_3, q_4, q_5 \}$

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

$q_0 = \{q_0\}$

$F = \{q_4\}$

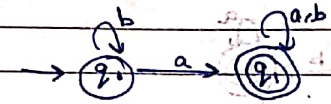
δ	0	1
q_0	q_1	q_1
q_1	q_2	q_5
q_2	q_3	q_3
q_3	q_5	q_4
q_4	q_4	q_4
q_5	q_5	q_5

Q. Design a DFA for the language

$L = \{ w / n_a(w) \geq 1, \ w \in (a,b)^* \}$

$L = \{ w / n_a(w) \leq 3, \ w \in (a,b)^* \}$

Sol. (i)



$Q = \{q_0, q_1\}$

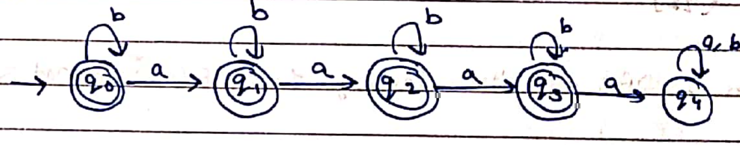
$\Sigma = \{a,b\}$

$q_0 = \{q_0\}$

$F = \{q_1\}$

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_1

(ii)



$Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

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

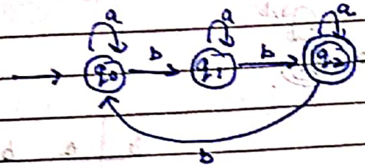
$q_0 = \{q_0\}$

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

δ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_4	q_3
q_4	q_4	q_4

Q. Design DFA for the language $L = \{ w \in (a,b)^* / n_b(w) \bmod 3 \geq 1 \}$

Sol.

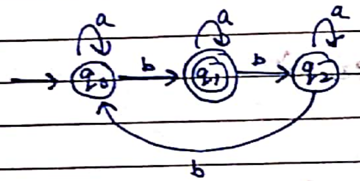


$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$q_0 = \{q_0\}$
$F = \{q_2\}$

δ	a	b
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_0

Q. Design a DFA for the language $L = \{ w \in (a,b)^* / n_b(w) \bmod 3 = 1 \}$.

Sol.

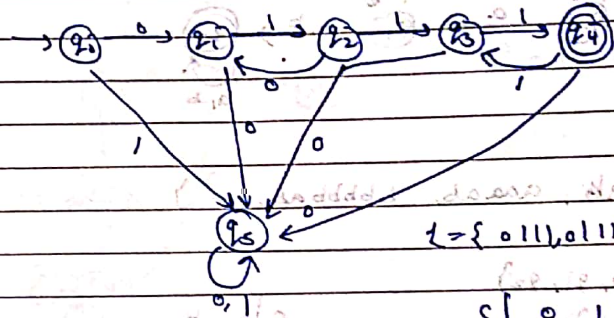


$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$q_0 = \{q_0\}$
$F = \{q_1\}$

δ	a	b
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_0

Q. Design a DFA for the language $L = \{ (01)^i 1^{2j} / i, j \geq 1 \}$

Sol.



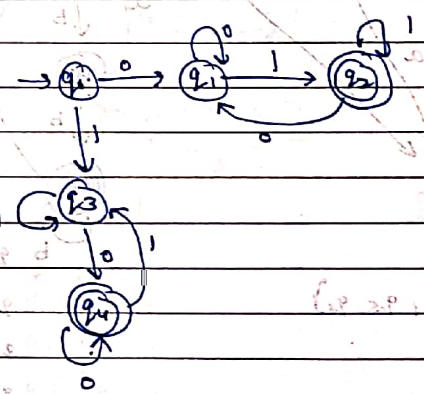
$L = \{ 0111, 011111, 010111, \ldots \}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
$F = \{q_4\}$

δ	0	1
q_0	q_1	q_5
q_1	q_5	q_2
q_2	q_1	q_3
q_3	q_5	q_4
q_4	q_5	q_3
q_5	q_5	q_5

Q. Design a DFA for the language contains string in which left most symbol differ from right most symbol.

$L = \{ 01, 10, 00001, 110000$

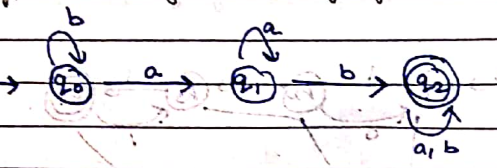


$Q = \{q_0, q_1, q_2, q_3, q_4\}$
$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
$F = \{q_2, q_4\}$

δ	0	1
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_2
q_3	q_4	q_3
q_4	q_4	q_3

Q. Design a DFA for the language $L = \{ w_1 ab w_2 \,/\, w_1, w_2 \in (a,b)^* \}$

Sol.

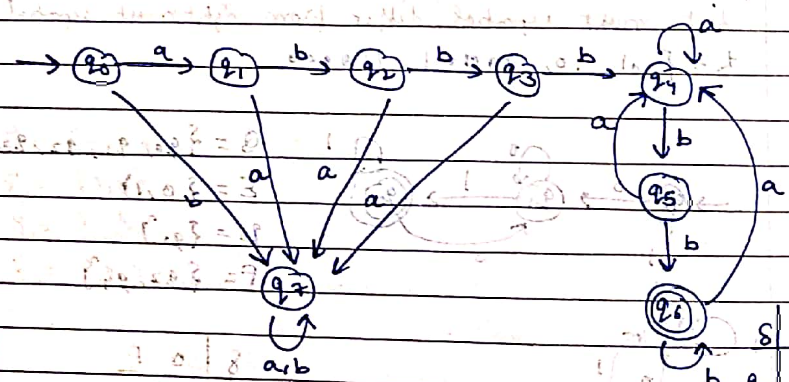


$L = \{ ab, aaaab, bbbbbab, \ldots \}$

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$q_0 = \{q_0\}$
$F = \{q_2\}$

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_2	q_2

Q. Design DFA for the language $L = \{ ab^3 w b^2 \,/\, w \in (a,b)^* \}$

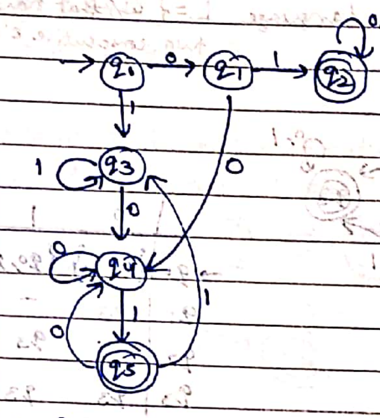


$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$
$\Sigma = \{a, b\}$
$q_0 = \{q_0\}$
$F = \{q_6\}$

δ	a	b
q_0	q_1	q_7
q_1	q_7	q_2
q_2	q_7	q_3
q_3	q_7	q_4
q_4	q_4	q_5
q_5	q_4	q_6
q_6	q_4	q_6

Q. Design a FA over alphabet $\Sigma = \{0, 1\}$ which accepts the set of strings either starts with 01 or end with 01

Sol.



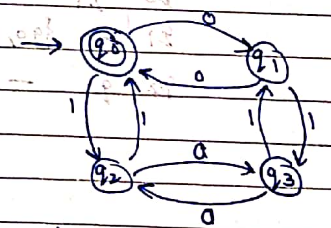
$\Sigma = \{0, 1\}$

δ	0	1
q_0	q_1	q_3
q_1	q_4	q_2
q_2	q_2	q_2
q_3	q_4	q_3
q_4	q_4	q_5
q_5	q_4	q_3

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
$q_0 = \{q_0\}$
$F = \{q_2, q_5\}$

Q. Design DFA which accepts the language $L = \{ w$ / even of 0's and even no. of 1's$\}$ over alphabet on $\Sigma = \{0, 1\}$

Sol: $L = \{ \epsilon, 00, 11, 1001, 0110, 1010, 0101, \ldots \}$



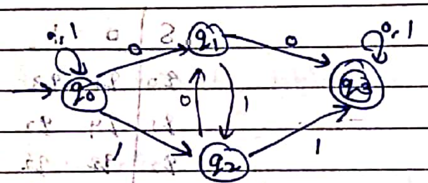
δ	0	1
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
$F = \{q_0\}$

NDFA : $\delta(Q \times \Sigma) = 2^Q$

Q. Design the NFA for the language L = { w / that have atleast two consecutive 0's or two cons. 1's } over $\Sigma = \{0,1\}$

Sol.



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

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

$q_0 = \{q_0\}$

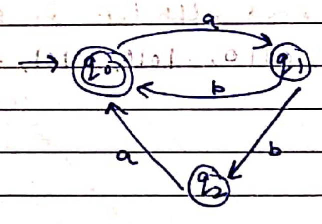
$F = \{q_3\}$

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

Q. Design a NFA for the language $L = \{ab \cup aba\}^*$

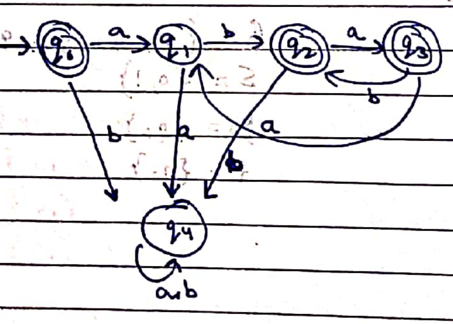
Sol. L = { ε, ab, aba, ababa, abab, abaaba, ... }

NDFA :



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

DFA :



	a	b
q_0	q_1	q_4
q_1	q_4	q_2
q_2	q_3	q_4
q_3	q_1	q_2
q_4	q_4	q_4

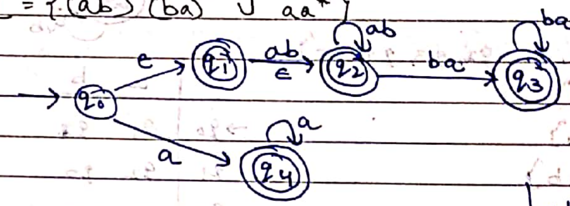
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

$q_0 = \{q_0\}$

$F = \{q_0, q_2, q_3\}$

Q. Design a finite automata which accepting language $L = \{(ab)^* (ba)^* \cup aa^*\}$



	ab	ba	a
$\rightarrow q_0$			

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

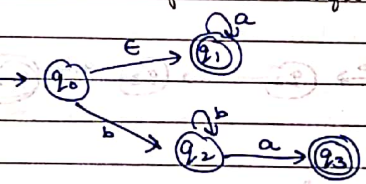
$\Sigma = \{a, b\}$

$q_0 = \{q_0\}$

$F = \{q_1, q_2, q_3, q_4\}$

Q. find the NFA for the language $L = \{(a^n / n \geq 0) \cup (b^n a / n \geq 1)\}$

Sol.



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

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

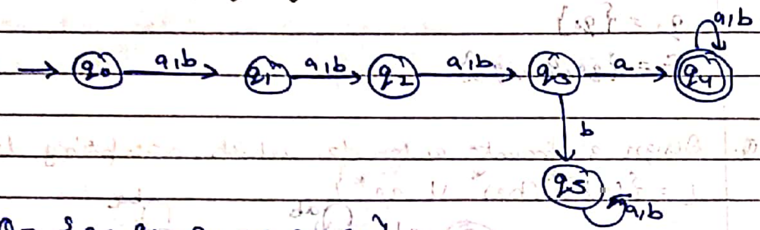
$q_0 = \{q_0\}$

$F = \{q_1, q_3\}$

Scanned by CamScanner

Q. Design a FA in which the 4th symbol from left ~~state~~ end is 'a' always for $\Sigma = \{a, b\}$.

Sol.



DFA:

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

$q_0 = \{q_1\}$

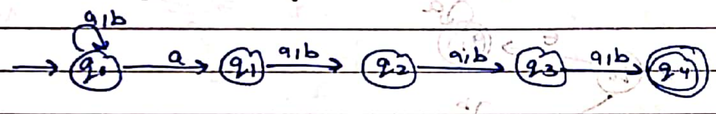
$F = \{q_4\}$

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

	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_4	q_5
q_4	q_4	q_4
q_5	q_5	q_5

Q. Design a FA in which 4th symbol from right end is always 'a' for $\Sigma = \{a, b\}$

Sol.



NDFA:

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

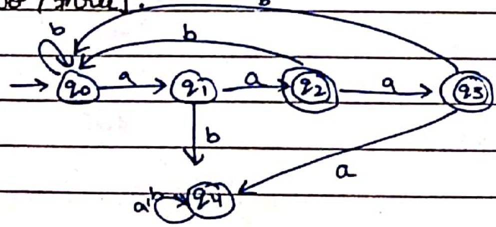
$q_0 = \{q_0\}$

$F = \{q_4\}$

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

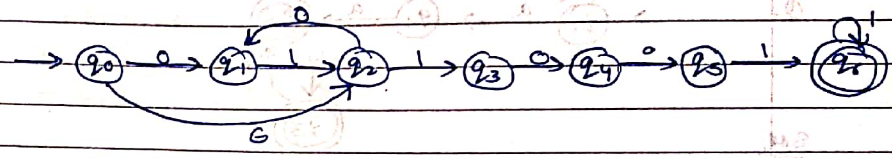
Q. Design a DFA for the language $L = \{w$: every run of a's has either two / three$\}$.

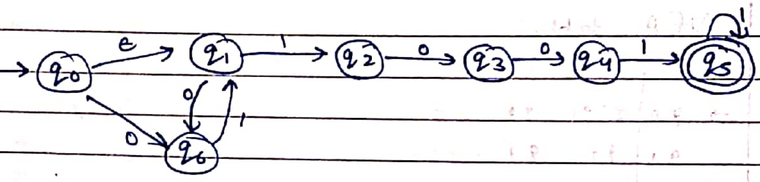
Sol.



Q. Design a FA for the language $L = \{(01)^n 1001^m \mid n \geq 0, m \geq 1\}$

Sol.





Some basic terms:

Kleen Closure: It is also called star closure. This is set of strings of any length including null string. Each string is obtained from input set Σ. The Kleen Star of the empty language ϕ is the empty string ϵ.

Positive closure: It may be defined as $\Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$ that means it consists of all the strings of any length except a null string.

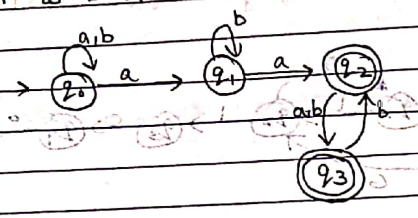
Alphabet: An alphabet is a finite collection of symbols. We can not define symbol formally.

String: String is finite collection of symbols from alphabet.

Language: The language is a collection of appropriate strings.

NFA to DFA conversion

Q.



Sol.

NFA table:

	a	b
→ q_0	q_0, q_1	q_0
q_1	q_2	q_1
* q_2	q_3	q_3
* q_3	–	q_2

DFA table:

	a	b
→ q_0	$\{q_0, q_1\}$	q_0
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$
* $\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
* $\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$
* $\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

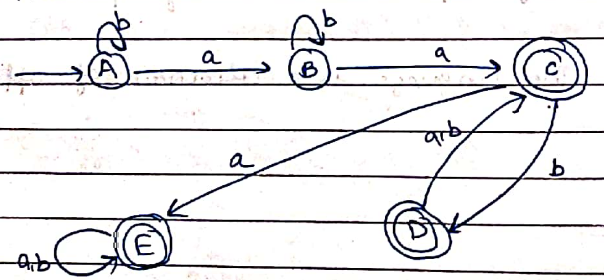
Let $q_0 = A$

$\{q_0, q_1\} = B$

$\{q_0, q_1, q_2\} = C^*$

$\{q_0, q_1, q_3\} = D^*$

$\{q_0, q_1, q_2, q_3\} = E^*$



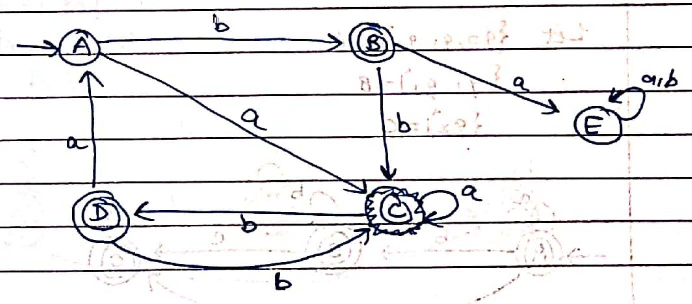
Q. NFA table:

	a	b
→ q_0	q_0, q_1	q_2
q_1	q_0	q_1
* q_2	–	$\{q_0, q_1\}$

DFA table

	a	b
→ q_0	q_0, q_1	q_2
* q_2	–	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_1, q_2\}$
* $\{q_1, q_2\}$	q_0	$\{q_0, q_1\}$

Let $q_0 = A$ $q_2 = B$ $\{q_0, q_1\} = C$ $\{q_1, q_2\} = D$



Concatenation of String:

Two strings are combined together to form a single string.

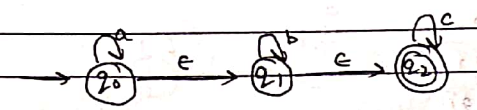
for ex: X = [white] Y = [house]

Then the concatenated string will be

XY = [white house]

NFA with ϵ moves to NFA :

Q.



Convert the following ϵNFA to NFA.

Sol.

ϵ closure of $q_0 = \{q_0, q_1, q_2\}$ *

ϵ closure of $q_1 = \{q_1, q_2\}$ *

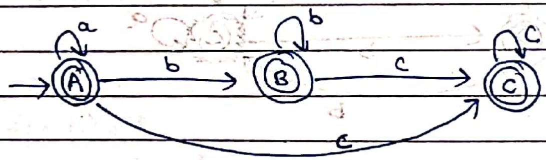
ϵ closure of $q_2 = \{q_2\}$ *

	ϵ closure of a	b	c
*→ $\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$	q_2
* $\{q_1,q_2\}$	–	$\{q_1,q_2\}$	q_2
* q_2	–	–	q_2.

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

$\{q_1,q_2\} = B$

$\{q_2\} = C$



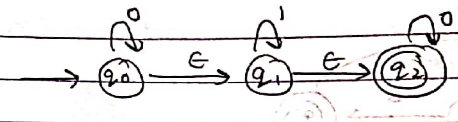
Transpose of string:

The transpose of string is also known as reversal of string.

for example:

if $(a\,b\,a\,b\,b\,b)^T = b\,b\,b\,a\,b\,a$.

Q.



Convert the following ϵ NFA to NFA.

Sol.

ϵ closure of $q_0 = \{q_0, q_1, q_2\}$ * ←

ϵ closure of $q_1 = \{q_1, q_2\}$ *

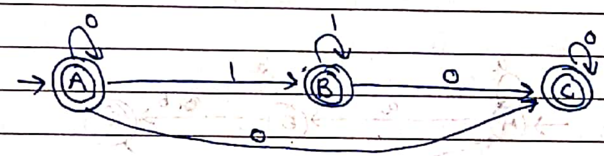
ϵ closure of $q_2 = \{q_2\}$ *

	ϵ closure of 0	1
→ * $\{q_0,q_1,q_2\}$	$\{q_0,q_1,q_2\}\{q_2\}$	$\{q_1,q_2\}$
* $\{q_1,q_2\}$	$\{q_2\}$	$\{q_1,q_2\}$
* $\{q_2\}$	$\{q_2\}$	–

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

$\{q_1,q_2\} = B$

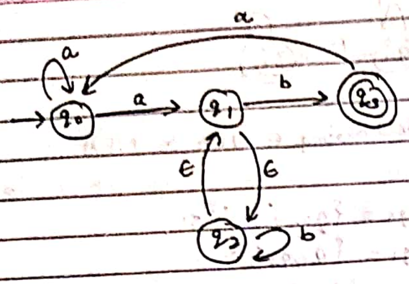
$\{q_2\} = C$



Finite Automata:

Finite automata is a mathematical model always accepts regular languages.

Q.



ϵ closure of $q_0 = \{q_0\}$ $= A$

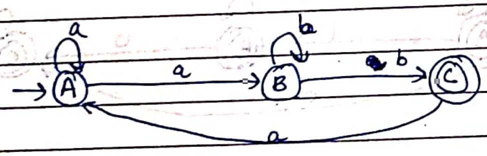
ϵ closure of $q_1 = \{q_1, q_2\}$ $= B$

ϵ closure of $q_2 = \{q_2, q_1\}$

ϵ closure of $q_3 = \{q_3\}$ $= C^*$

ϵ closure of

	a	b
$A = \{q_0\}$	A, B	–
$B = \{q_1, q_2\}$	~~[illegible]~~	B, C
$C = \{q_3\}$	A	–

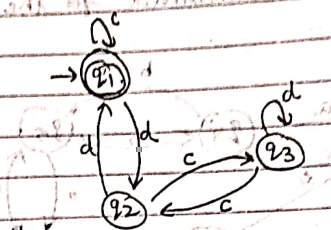


Deterministic finite Automata (DFA):

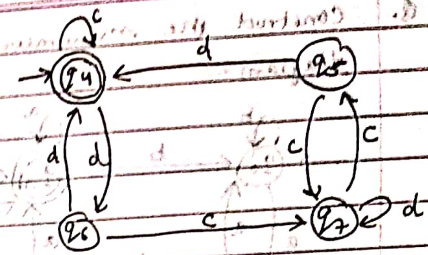
The finite automata is called Deterministic finite Automata if there is only one path for a specific input from current state to next state.

Equivalence of finite Automata

Q.



Machine M_1



Machine M_2

Sol.

	c	d
(q_1, q_4)	(q_1, q_4)	(q_2, q_6)
(q_2, q_6)	(q_3, q_7)	(q_1, q_4)
(q_3, q_7)	(q_2, q_5)	(q_3, q_7)
(q_2, q_5)	(q_3, q_7)	(q_1, q_4)

Both machine equivalent because (NF, F) & (F, NF) are not present.

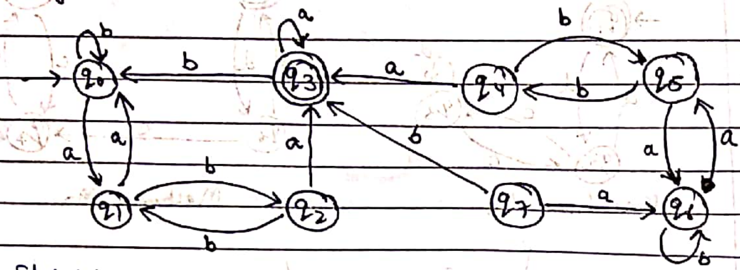
Non-Deterministic finite Automata (NFA):

The concept of Non Deterministic finite Automata is exactly reverse of Deterministic finite Automata. The finite Automata is called NFA when there exists many paths for a specific input from current state to next state.

Thus it is not fixed or determined that with a particular input where to go next. Hence this FA is called non deterministic finite automata.

Minimization of finite Automata

Q. Construct the minimum state automata for the following diagram:



Sol. Step 1:

$\Pi_0 = Q - F$

$\Pi_0 = \{\{q_3\}, \{q_0, q_1, q_2, q_4, q_5, q_6, q_7\}\}$

Step 2:

	a	b		a	b
$\rightarrow q_0$	q_1	q_0	$\rightarrow q_0$	NF	NF
q_1	q_0	q_2	q_1	NF	NF
q_2	q_3	q_1	q_2	F	NF
q_4	q_3	q_5	q_4	F	NF
q_5	q_6	q_4	q_5	NF	NF
q_6	q_5	q_6	q_6	NF	NF
q_7	q_6	q_3	q_7	NF	F
q_3	q_3	q_0			

$\Pi_1 = \{\{q_3\}, \{q_0, q_1, q_5, q_6\} \{q_2, q_4\} \{q_7\}\}$

Step 3:

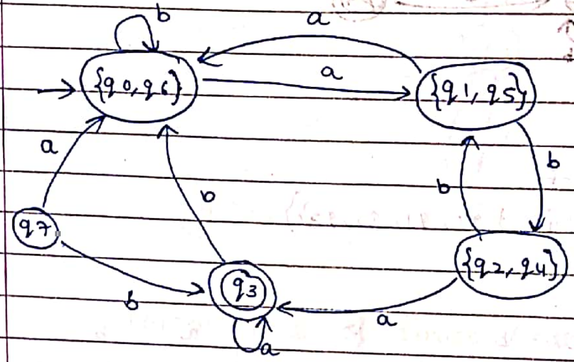
$\Pi_2 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\}, \{q_7\}\}$

Step 4:

$\Pi_3 = \{\{q_3\}, \{q_0, q_6\}, \{q_1, q_5\}, \{q_2, q_4\} \{q_7\}\}$

$\Pi_2 = \Pi_3$

	a	b
$\rightarrow \{q_0, q_6\}$	$\{q_1, q_5\}$	$\{q_0, q_6\}$
$\{q_1, q_5\}$	$\{q_0, q_6\}$	$\{q_2, q_4\}$
$\{q_2, q_4\}$	q_3	$\{q_1, q_5\}$
$* q_3$	q_3	$\{q_0, q_6\}$
q_7	$\{q_0, q_6\}$	q_3

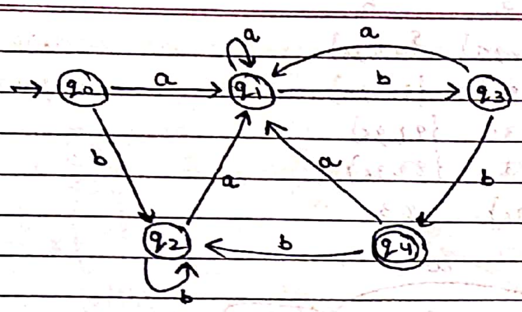


Difference between DFA and NFA

DFA	NFA
1. It stands for deterministic finite automata.	1. It stands for non-deterministic finite automata.
2. Only one transition is possible from one state to another on same input symbol.	2. More than one transition is possible from one state to another on same input symbol.
3. Transition function is: $\delta : Q \times \Sigma \rightarrow Q$	3. Transition function is: $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$
4. ϵ-transition is not possible.	4. ϵ-transition is possible.
5. DFA cannot be converted into NFA.	5. NFA can be converted into DFA.

Page No. 24

Q.



Sol. Step 1:

$\Pi_0 = Q - F$

$= \{\{q_4\}, \{q_0, q_1, q_2, q_3\}\}$

Step 2:

	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	q_4
q_4	q_1	q_2

	a	b
$\rightarrow q_0$	NF	NF
q_1	NF	NF
q_2	NF	NF
q_3	NF	F

$\Pi_1 = \{\{q_4\}, \{q_3\}, \{q_0, q_1, q_2\}\}$

Step 3:

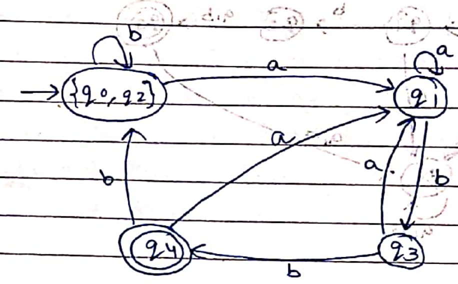
$\Pi_2 = \{\{q_4\}, \{q_3\}, \{q_1\}, \{q_0, q_2\}\}$

$\Pi_3 = \{\{q_4\}, \{q_3\}, \{q_1\}, \{q_0, q_2\}\}$

$\Pi_2 = \Pi_3$

Page No. 25

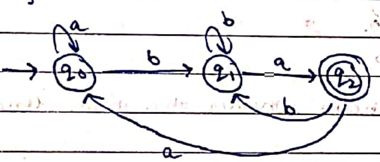
	a	b
$\rightarrow \{q_0, q_2\}$	$\{q_1\}$	$\{q_0, q_2\}$
$\{q_1\}$	$\{q_1\}$	$\{q_3\}$
$\{q_3\}$	$\{q_1\}$	$\{q_4\}$
$* \{q_4\}$	$\{q_1\}$	$\{q_0, q_2\}$



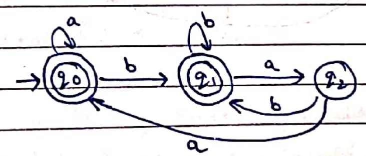
Complement of finite Automata

Q. Draw a finite automata which accepts all the strings not ending with ba. for $\Sigma = \{a, b\}$.

Sol: Ending with ba.

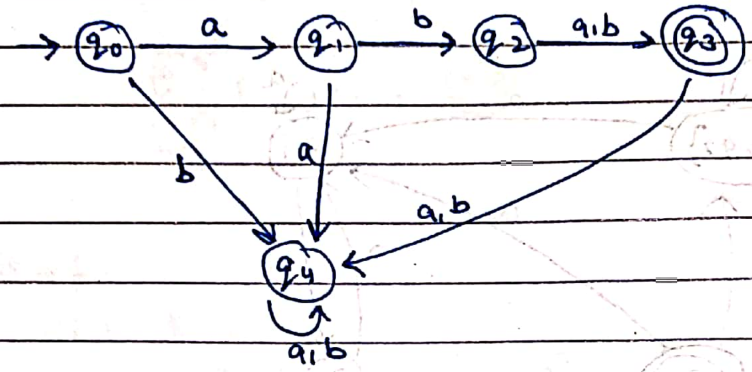


not ending with ba.

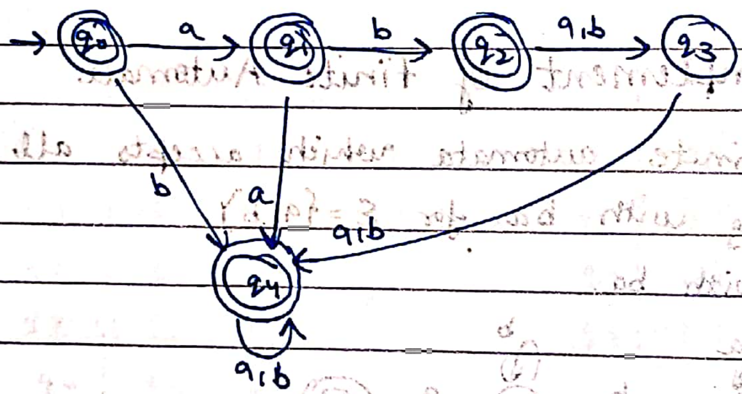


Q. Draw a finite automata which accepts the string aba & abb. Draw the compliment of finite automata which accepts all the strings except aba & abb over $\Sigma = \{a, b\}$

Sol:



Complement:



Q. Design a FA which checks whether the given unary number is divisible by 3.

Sol