Unit-1.

Basic Terms:

- Symbol - abstract entity like a-z, A-Z, 0-9, (, +, - ... (i.e., any character)
- alphabet - represented with Σ (sigma)
  It is a finite non-empty set of symbols.
  Ex: $\Sigma = \{a, b\}$ → binary alphabet.
- string - collection of sequence of symbols over an alphabet Σ.
  Ex: $\Sigma = \{0, 1\}$ 01001 - string
  012001 - not string (in valid)
- Language - set of all the strings over an alphabets

String operations

- Length - represented using ... | |
  $\Sigma = \{0, 1\}$ A = 0101 |A| = 4
- null string - represented with ϵ
- Power of an alphabet:
  Σ is an alphabet, power of alphabet is
  $\Sigma^1, \Sigma^2, \ldots \Sigma^n$

Σ^k - set of all strings whose length is k.

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

$\Sigma^3 = \{000, \ldots, 111\}$

Σ^0 - null string.

Kleen closure: $\Sigma^* = \Sigma^0 \cup \Sigma^1 \ldots \cup \Sigma^n$

Positive closure (over an alphabet Σ):
It is represented with Σ^+

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

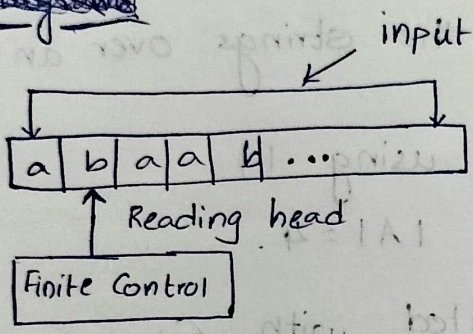
$\Sigma^+ \cdot \Sigma^* - \{\epsilon\}$

$0^*1 \cdot \{01, 001, 0001, \ldots\}$

Finite automata: (mathematical model)

- A finite automata is the first mathematical model of a system with discrete inputs and outputs
- It is useful design tool for finite state system.
- In finite automata, its output depends only on the input.
- Also called finite automata without memory

Finite Automal model 1.1



Representing finite automata
1. Graphical
2. Tablular
3. Mathematical representation

- input tape - stores input strings divided into cells. each cell holds one symbol
- Reading head - reads input string, one symbol at a time.
- finite control - manages functionalities of input tape & Reading head

If finite control reaches the end of input tape - then input string is valid.

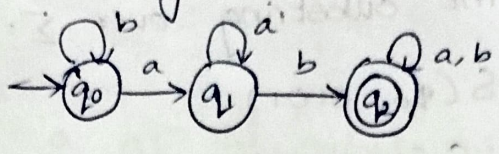
Transition diagram: It is a directed graph associated with finite state system.

- The vertices of the graph corresponds to the states of finite automata
- Edges corresponds to trasition of finite automal

Initial state: A state with incoming arrow. (from nowhere)

Final state : a state represented by double circles.

- A transition diagram consists of number of final states.



1.2. Transition diagram

Transition table : It is the tabular representation of finite ~~ato~~ automata.

- The rows of the table corresponds to states, columns corresponds to inputs.
- The entry for rows corresponds to state q & columns correponds to input a, represented with $\delta(q, a)$

Mathematical model (representation) (of finite automata)

→ formal definition.

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

Q - set of states

Σ - input alphabets

δ - transition functions

q_0 - initial state

F - set of final states

(Initial)	a	b
→ q_0	q_1	q_0
q_1	q_1	q_2
* q_2 (final)	q_2	q_2

1.3. Transition table.

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

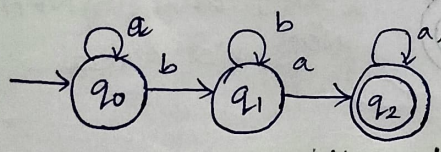
$\Sigma = \{a, b\}$

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

$q_0 = q_0$

$F = \{q_2\}$

Example :



	a	b
→ q_0	q_0	q_1
q_1	q_2	q_1
* q_2	q_2	q_2

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

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

$\delta(q_0, abbab)$

$\vdash \delta(q_0, bbab)$

$\vdash \delta(q_1, bab)$

$\vdash \delta(q_1, ab)$

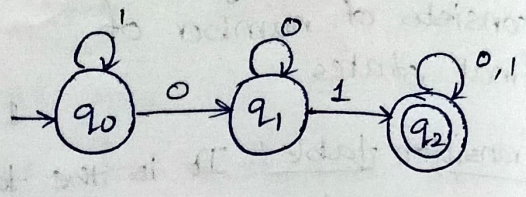
$\vdash \delta(q_2, b)$

$\vdash q_2$ [Accepted]

∴ When reading the input string, if the final state is the result then the string is accepted.

#2 Design finite automata which accepts. 01 as the substring over $\Sigma = \{0, 1\}$

$\delta(q_0, 0101)$
$\vdash \delta(q_1, 101)$
$\vdash \delta(q_2, 01)$
$\vdash \delta(q_2, 1)$
$\vdash q_2$ [accepted]



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

$\delta(q_0, 100) \vdash q_1$
$\vdash \delta(q_0, 00)$ [not accepted]
$\vdash \delta(q_1, 0)$

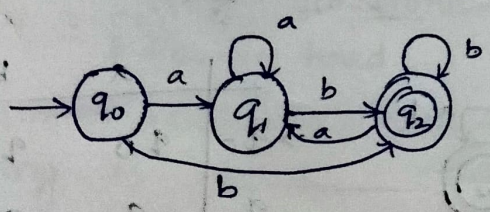
Types of finite automata:

*DFA - Deterministic FA
NFA - Non-deterministic FA.

Finite Automata.
- finite A. without output
  - DFA
  - NFA
  - NFA with ϵ transitions
- FA with output
  - Moore M/C
  - Mealy M/C

Deterministic FA: In DFA, if there is only one path for specific input from the current state to the next state.

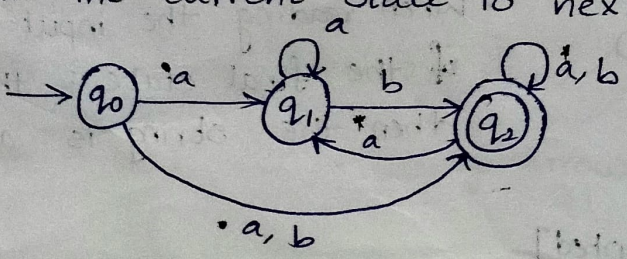
Ex



*Every DFA is NFA but NOT vise versa

DFA does not accept the null transitions. i.e., DFA cannot change the state without any input.

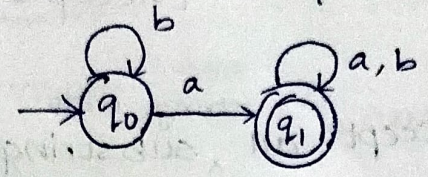
NFA: In NFA, there are multiple paths for a specific input. from the current state to next state



NFA with ϵ transitions: It is a FA with atleast one ϵ transitions.

* Design a DFA to accept 'a' as the substring over an alphabet $\Sigma = \{a, b\}$

transition diagram



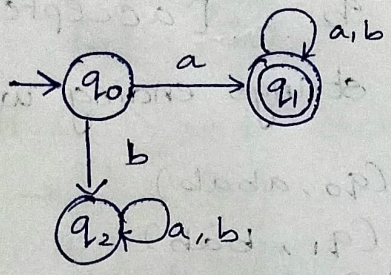
	a	b
$\rightarrow q_0$	q_1	q_0
$*q_1$	q_1	q_1

$\delta(q_0, abb)$ $\vdash \delta(q_1, bb)$

$\vdash \delta(q_1, b)$

$\vdash q_1$ [accepted]

* Design a DFA to accept the strings starting with a over $\Sigma = \{a, b\}$

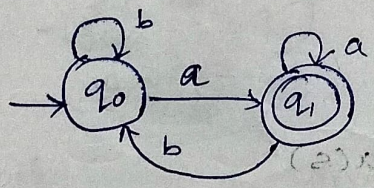


$\delta(q_0, abb)$ $\vdash \delta(q_1, bb)$ $\vdash \delta(q_1, b)$ $\vdash q_1$ [accepted]

$\delta(q_0, ba)$ $\vdash \delta(q_2, a)$ $\vdash q_2$ [rejected]

* Design DFA to accept all the strings that ends with a over $\Sigma = \{a, b\}$

[Write dig, stuple rel., acceptance & rejectance in exam]



$\delta(q_0, aba)$

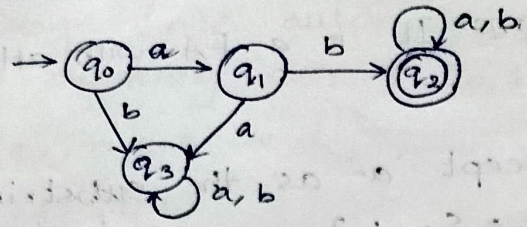
$\vdash \delta(q_1, ba)$

$\vdash \delta(q_0, a)$

$\vdash q_1$ [accepted]

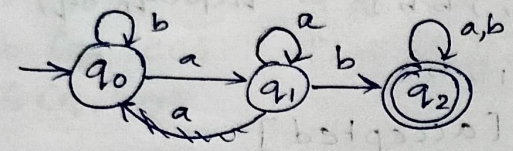
* Design DFA to accept all strings starting with ab over $\Sigma = \{a, b\}$

P.T.O



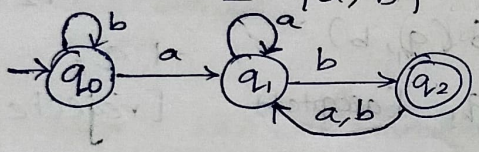
$\delta(q_0, abaab)$
$\vdash \delta(q_1, ba)$
$\vdash \delta(q_2, a)$
$\vdash q_2$
[accepted]

* Design a DFA to accept all strings with substrings ab over $\Sigma = \{a, b\}$



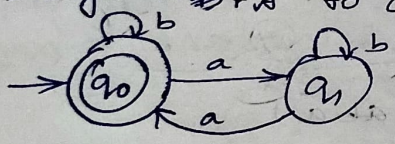
$\delta(q_0, baaba)$
$\vdash \delta(q_0, aaba)$
$\vdash \delta(q_1, aba)$
$\vdash \delta(q_0, ba)$
$\vdash \delta(q_1, a)$
$\vdash q_2$ [accepted]

* Design a DFA to accept all strings ending with ab over $\Sigma = \{a, b\}$



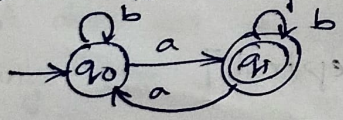
$\delta(q_0, abab)$
$\vdash \delta(q_1, bab)$
$\vdash \delta(q_2, ab)$
$\vdash \delta(q_1, b)$
$\vdash q_2$ [accepted]

• Design a DFA to accept even number a(s) $\Sigma = \{a, b\}$

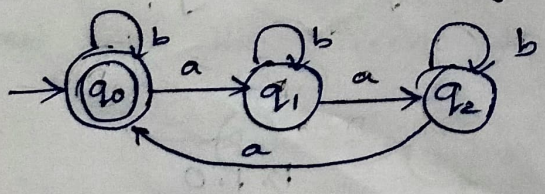


aa
aaaab
aaab

• DFA to accept odd no. of a(s)

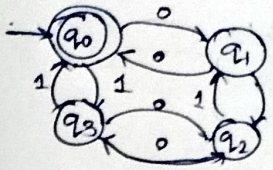


• No. of a(s) divisible by 3 $\Sigma = \{a, b\}$

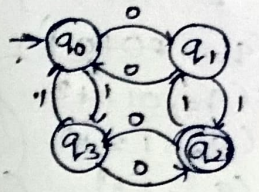


aaa
aaa
aaab

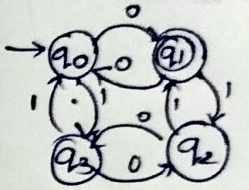
• Design FA which acc
0(s) over $\Sigma = \{0, 1$



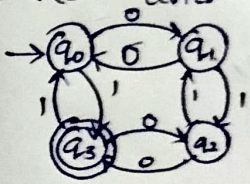
• FA which accept



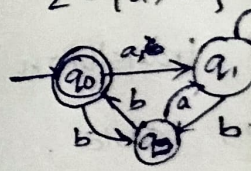
• even 1(s) and



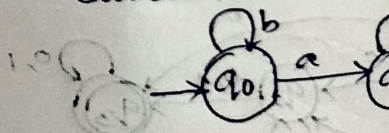
• odd 1(s) and



• Design a DFA to
over $\Sigma = \{a, b\}$



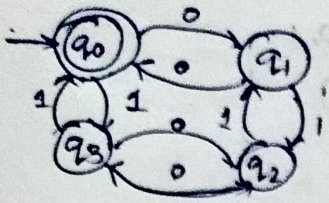
• starts with a



NFA:

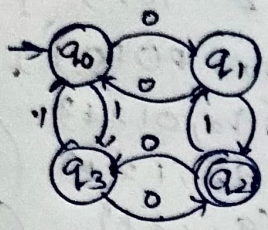
If there exist
specific input

• Design FA which accept all the strings with even 0(s) over $\Sigma = \{0, 1\}$ and even 1(s).

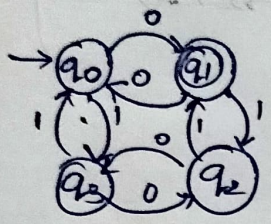


odd 1 odd 0
odd 1 even 0
even 1 even 0
q_1 - even 1 odd 0
Imp

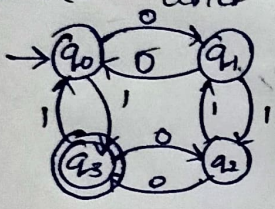
• FA which accept odd ~~even~~ 0's and odd 1(s) over $\Sigma = \{0, 1\}$



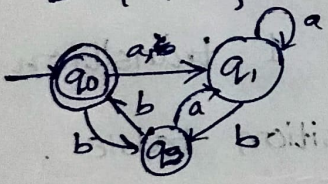
• even 1(s) and odd 0(s) over $\Sigma = \{0, 1\}$

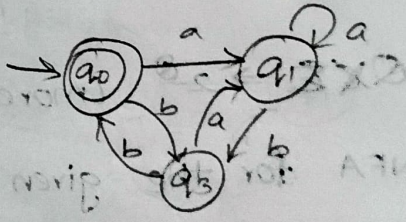


• odd 1(s) and even 0(s) over $\Sigma = \{0, 1\}$

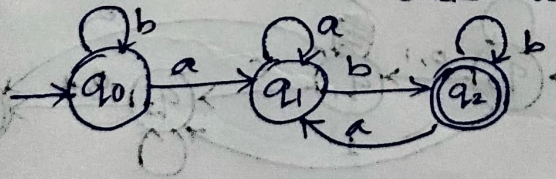


• Design a DFA to accept strings ending with abb over $\Sigma = \{a, b\}$





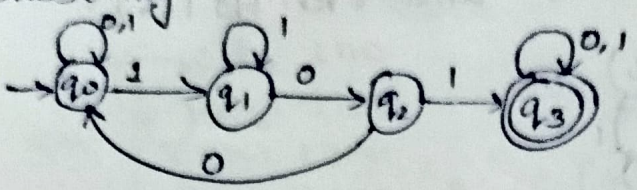
• starts with a and ends with b



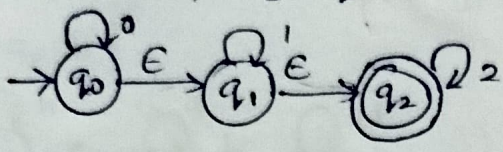
NFA:

If there exist more than one transition for a specific input is NFA

• Design an NFA which accepts all strings with 101 as substring.



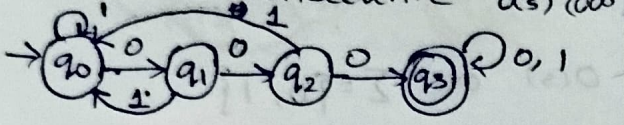
• Design NFA that accepts all the strings consists of any no. of 1(s), 0(s) & 2(s) over $\Sigma = \{0, 1, 2\}$



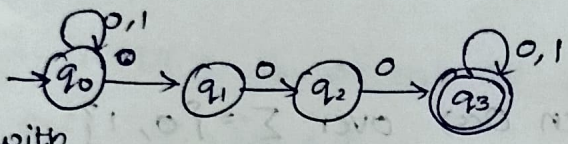
$\delta(q_0, 0012\$)$
$\vdash \delta(q_0, 012\$)$
$\vdash \delta(q_0, 12\$)$
$\vdash \delta(q_1, 2\$)$
~~$\vdash \delta(q_2, \$)$~~
$\vdash q_2$

cross check.

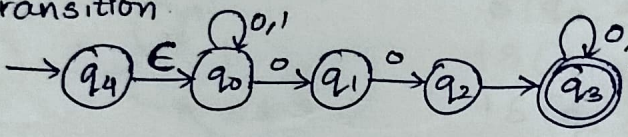
• DFA for 3 consecutive 0(s) (000)



• NFA for 3 consecutive 0(s) (000)



NFA with ε transition



• In DFA, the transition function

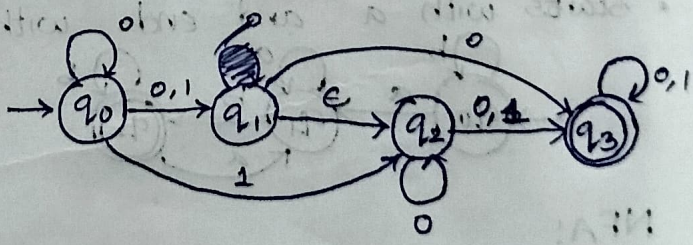
$Q \times \Sigma \to Q$

• In NFA,

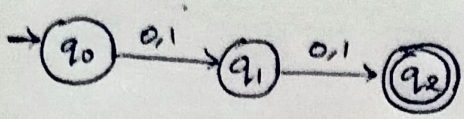
$Q \times \Sigma \to 2^Q$ (more than 1 transition)

• Design NFA for the given transition table.

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

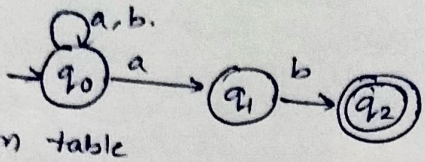


• Construct NFA for all strings of length 2 over $\Sigma = \{0, 1\}$



*** Conversion of NFA to DFA!

Consider the NFA.



Initial state of DFA = initial state of NFA.

Transition table

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

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

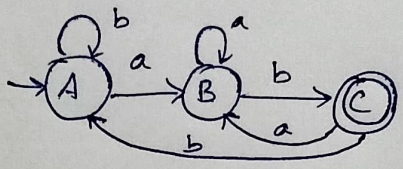
$\delta([q_0, q_1], a) \rightarrow [q_0, q_1]$

$\delta([q_0, q_1], b) \rightarrow [q_0, q_2]$

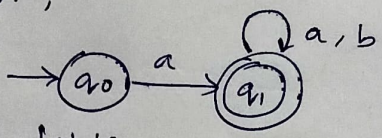
$\delta([q_0, q_2], a) \rightarrow [q_0, q_1]$

$\delta([q_0, q_2], b) \rightarrow [q_0]$

δ	a	b
→ A	B	A
B	B	C
* C	B	A

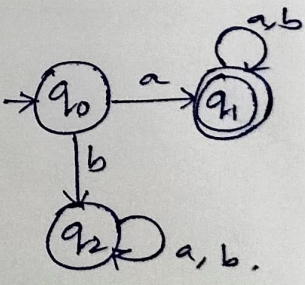


Consider the NFA,



The transition table.

δ	a,	b
→ q_0	q_1	ϕ
* q_1	q_1	q_1

⇒

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