S R M UNIVERSITY

THEORY OF COMPUTATION

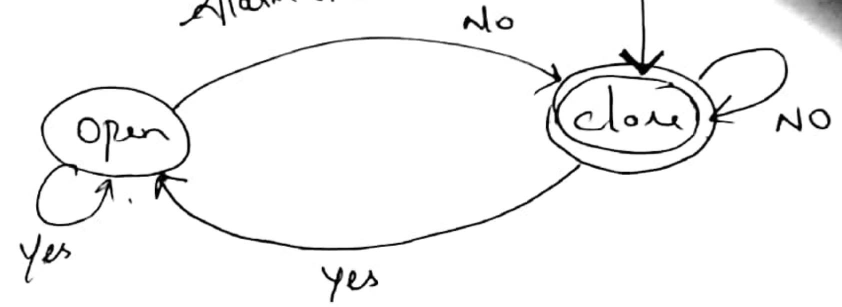
Introduction

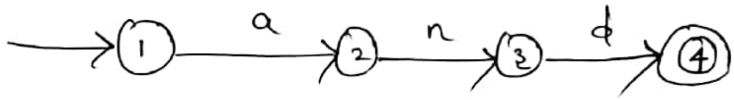
* any task performed by computer / calculator / machine
* Mathematically model a computer or any machine in general & then learn abt the machine
* computational Capabilities of machines, Problem solved by machines, Limitations of M/c

TOC

- Automata Theory (mathematical model)
  - (less memory) * Finite Automata
  - Stack * push down automata
  - Finite tape * Linear bound automata
  - * Turing machine
    - Infinite tape
- Computability Theory (computational Limitation)
  - * Decidable language
  - * undecidable "
- Complexity Theory
  - time
  - Space

Introduction to finite Automata (without using memory)





$\Sigma = \{a, \ldots z\}$

Formal definition [DFA] Deterministic Finite Automata

5 tuples

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

Q - Finite non empty set of states

Σ - Finite non empty set of inputs / input alphabet

δ - Transition function which maps $Q \times \Sigma \rightarrow Q$

$q_0 \in Q$ is initial state

$F \subseteq Q$ set of final states

Symbol – a, b, 0, 1,

$\Downarrow$

"Σ" Alphabet – $\{a, b\}$, $\{a, b, c\}$ $\{0, 1, \dots 9\}$

$\Downarrow$ Some collection of symbol.

String – Sequence of symbol

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

$$\{\underbrace{a, b}_{\text{length } 1}, \underbrace{aa, bb, ab, ba \dots}_{\text{length } 2}\}$$

No of sigma is represent in $|\Sigma|$

How many string of length n is possible over alphabet can be represent by $|\Sigma|^n$

Let length be 2 $\Sigma = \{a, b\}$

a a
a b
b a
b b

Let length be n $\Sigma = \{a, b\}$

$\overline{\{a,b\}}\ \overline{\{a,b\}}\ \overline{\{a,b\}}\ \overline{\quad} \dots n$

$2 \times 2 \times 2 \times \dots n$

2^n

Language ⇒ Collection of strings

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

L_1 = Set of all strings of length 2

$= \{aa, ab, ba, bb\}$

L_2 = set of all string of length 3

$= \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

L_3 = Set of all string where each string start with a.

$= \{a, aa, ab, aaa, aab, aba, abb, \ldots\}$

Power of Σ $\quad \Sigma = \{a, b\}$ $\quad |\epsilon| = 0$

$\Sigma^0 = \{\epsilon\}$ → nullstring Epsillon (special symbol)

$\Sigma^1 = \{a, b\}$

Set of all strings over Σ of length '1'

$\Sigma^2 = \Sigma\Sigma =$ " " length '2'

$\{aa, ab, bb, ba\}$

$\Sigma^3 = \Sigma\Sigma\Sigma =$ " " length 3

$|\Sigma^3| = 8$

Σ^n = n length string

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots$ $\quad \Sigma = \{a, b\}$

infinite → $= \{\epsilon\} \cup \{a, b\} \cup \{aa, ab, bb, ba\} \cup \ldots$

Set of all possible set of all length possible over $\{a, b\}$

Can be called as mother / universal set of language

$L_1 \subseteq \Sigma^*$

$L_2 \subseteq \Sigma^*$

$L_3 \subseteq \Sigma^*$

$\boxed{L_1 \quad L_2 \quad L_3}\ \Sigma^*$

C Prgm $\Sigma = \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1, \ldots, 9, +, *, \ldots\}$

Alphabet is finite

```
void main ()
{ int a, b;
  =
}
```

C-program in toc its string

C-Programming language = Set of all valid programs

$= \{P_1, P_2, P_3 \ldots\}$

P_n present in valid prgm

'L' 'S'

whether string is present in Language

finite then yes

infinite then No

L is finite

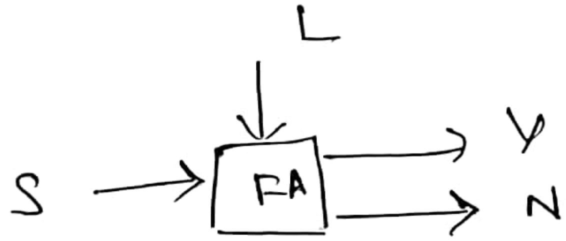
$\Sigma = \{a, b\}$

$L_1 = \{aa, ab, ba, bb\}$

$S = aaa$

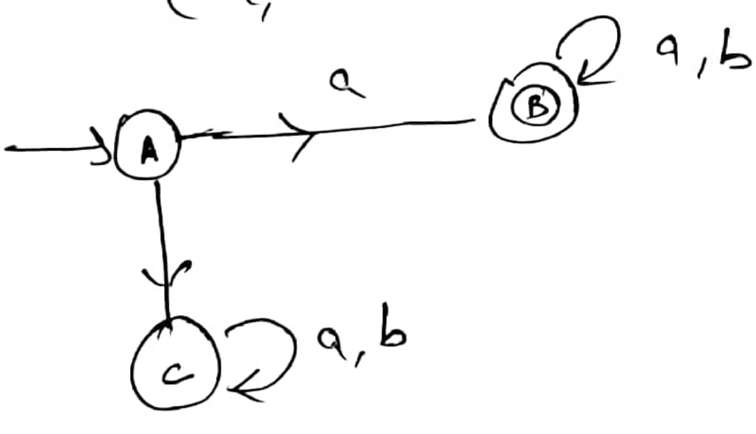
$L_2 = \{a, aa, aaa, ab, \ldots\}$ start with a

$S =$ baba



$L_1 =$ Set of string starting with a

$\{a, aa, ab, aaa, \ldots\}$



Circles are states
double circle are Final state
circle with arrow in right of it initial state

$S = aab$ whether the string present in L_1.

$$A \xrightarrow{a} B \xrightarrow{a} B \xrightarrow{b} B.$$

b b a

if the string said to be accepted it start with initial state and end with final state

$$A \xrightarrow{b} C \xrightarrow{b} C \xrightarrow{a} C$$

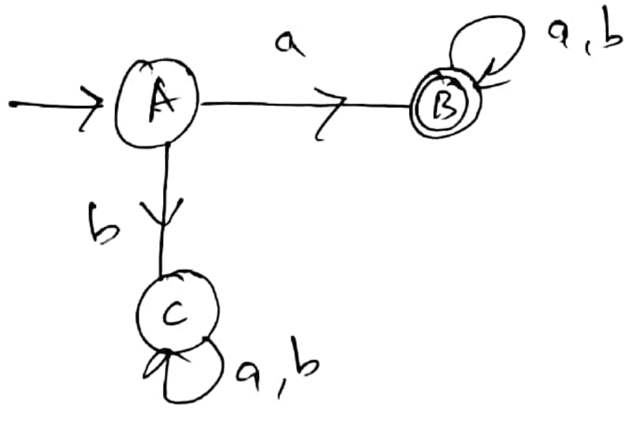
15/july

Finite Automata

- FA with o/p
  - Moore machine
  - Mealy Machine
- without o/p
  - DFA
  - NFA
  - ϵ-NFA

DFA - Deterministic finite Automate

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



$Q = \{A, B, C\}$ → Finite set of state

Set of all states

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

→ input alphabet
Input

$q_0 = A$
↳ start state

$F = \{B\}$
↳ set of final states

$Q \supseteq F$

Q super set of F

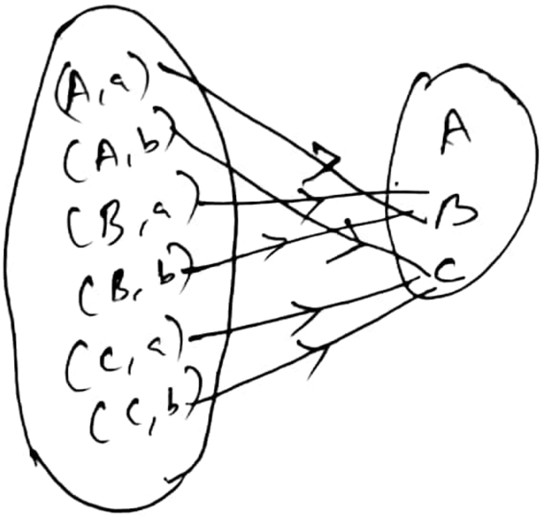
$\delta = Q \times \Sigma \rightarrow Q$

↳ transition fuction

$\{A, B, C\} \times \{a, b\}$

$\{(A, a)\ (A, b)\ (B, a)\ (B, b)\ (C, a)\ (C, b)\}$

Scanned by CamScanner



In DFA for every state for every input there will be exactly one input

transition can be atleast (or) atmost one

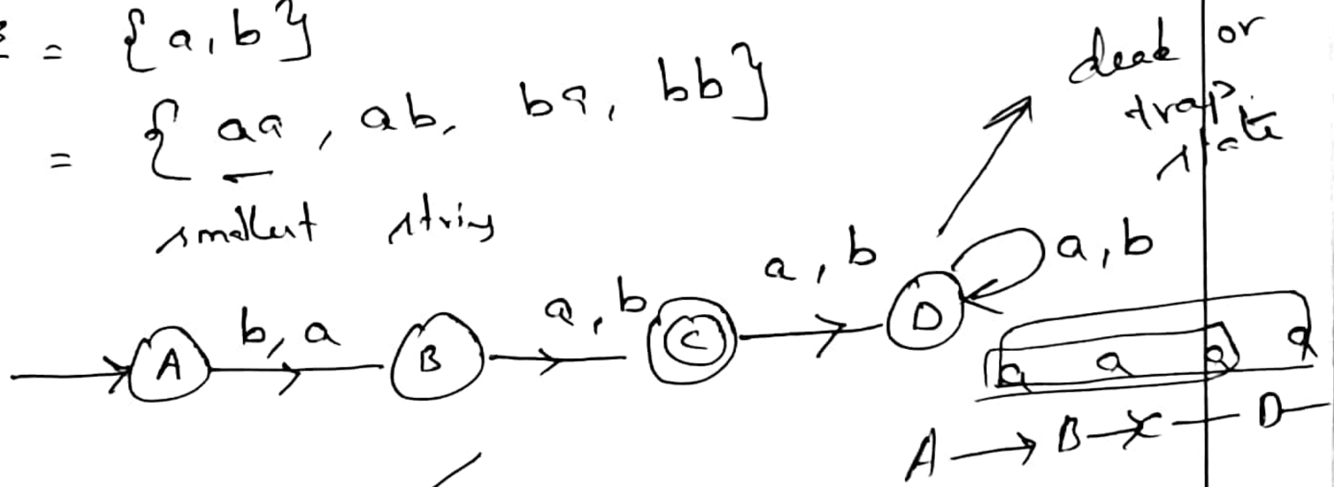
Examples

Q1) Construct a dfa that accepts set of all strings over {a, b} of length 2.

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

$L = \{aa, ab, ba, bb\}$

smallest string



$A \xrightarrow{a} B \xrightarrow{a} C$ ✓

$A \xrightarrow{a} B \xrightarrow{b} C \xrightarrow{b} D$ ✗.

Rejected / non accepting states :-

Dead or trap state is said to be Rejected / non Accepting state.

Acceptance of the string

String is said to be accepted by FA if we are able to reach a final state starting from initial state upon scanning the entire i/p string.

Acceptance of Language

A FA is said to accept a language if all the strings in the language are accepted and all the strings not in the language are rejected

$A \xrightarrow{a} A$ ✓

aa

$A \xrightarrow{a} A \xrightarrow{a} A$ ✓

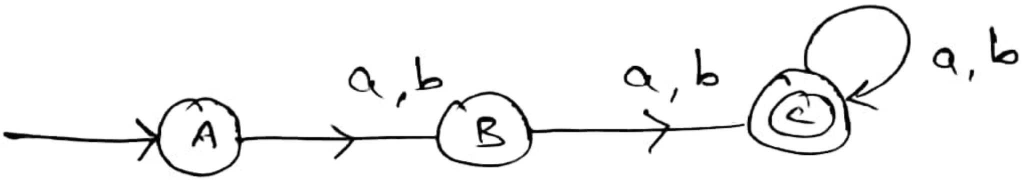
✓ $A \xrightarrow{a} A \xrightarrow{a} A \xrightarrow{b} A$

Q2) Construct the DFA which accepts set of string over {a, b} such than length of string is atleast 2.

$w \in \{a, b\} \quad |w| \geq 2$

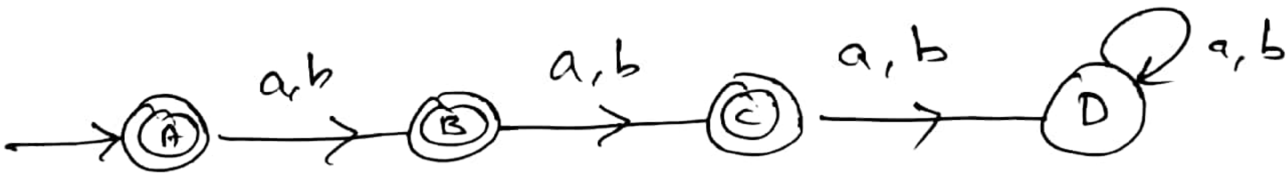
$\Sigma = \{a, b\}$

$L = \{ \underline{aa}, bb, ab, ba, aaa, \ldots, bbb, \ldots \}$



Q3) atmost 2.

$w \in \{a, b\} \quad |w| \leq 2$. → set of strings of length 0, 1, 2



0   1   2   3 or more

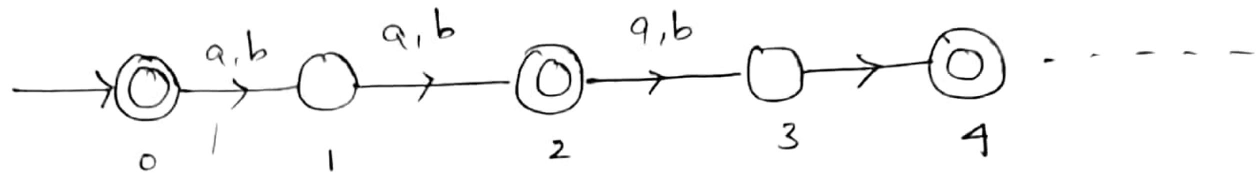
In general $|w| = 2$ $|w| \geq n$ $|w| \leq n$

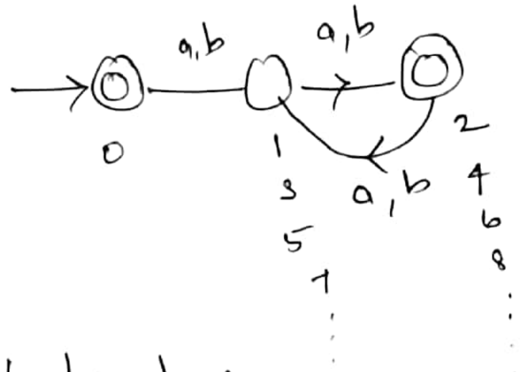
$n+2$ $n+1$ $n+2$

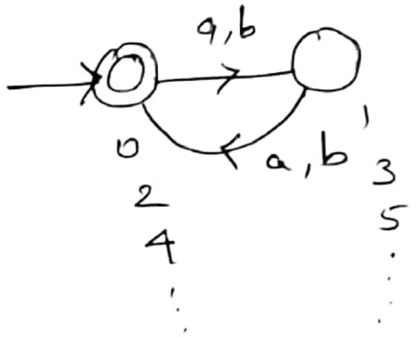
Q) Construct DFA $w \in \{a, b\}$

such that $|w| \bmod 2 = 0$ (set of all even strings)

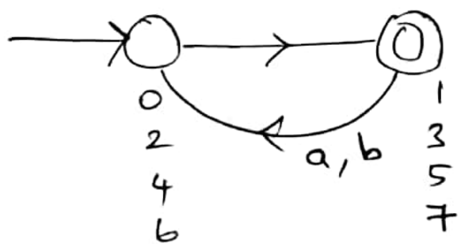
$L = \{ \epsilon, aa, ab, ba, bb, aaaa, bbbb \ldots \}$







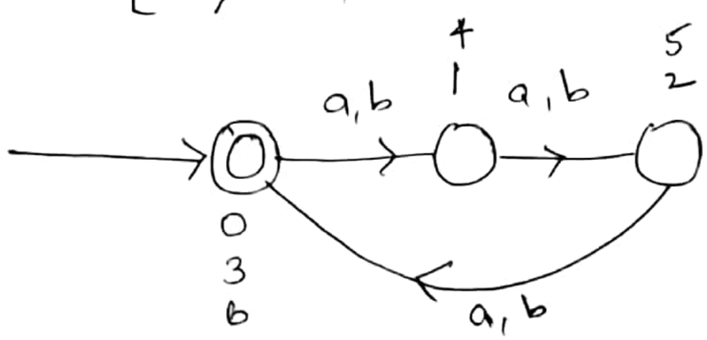
Q) $|w| \bmod 2 = 1$ (set of all odd strings)



$|w| \cong 1 \bmod 2$.

Q) $w \in \{a, b\}^*$ $|w| \bmod 3 = 0$

$L = \{ \epsilon, aaa, aab, bbb, aaaaaa \ldots \}$



In general

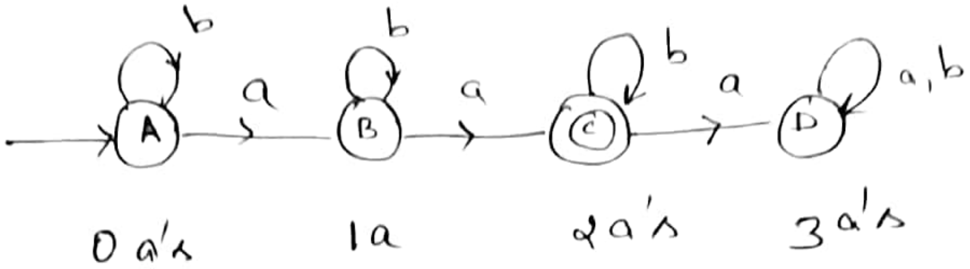
$|w| \bmod n = 0$

'n' of states

Q) Construct DFA $w \in \{a, b\}^*$

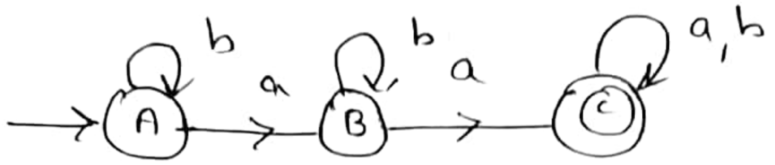
$n_a(w) = 2$

$L = \{ aa, baa, aab, aba, bbaa, \ldots \}$



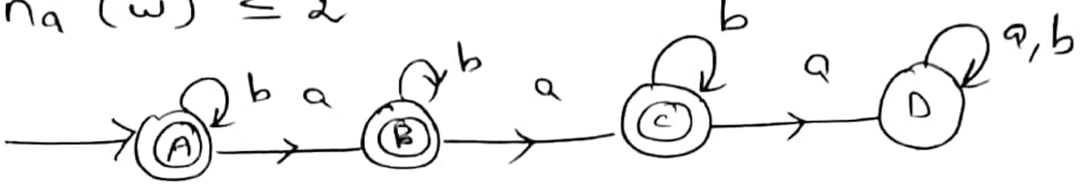
$n_a(w) \geq 2$

Q)



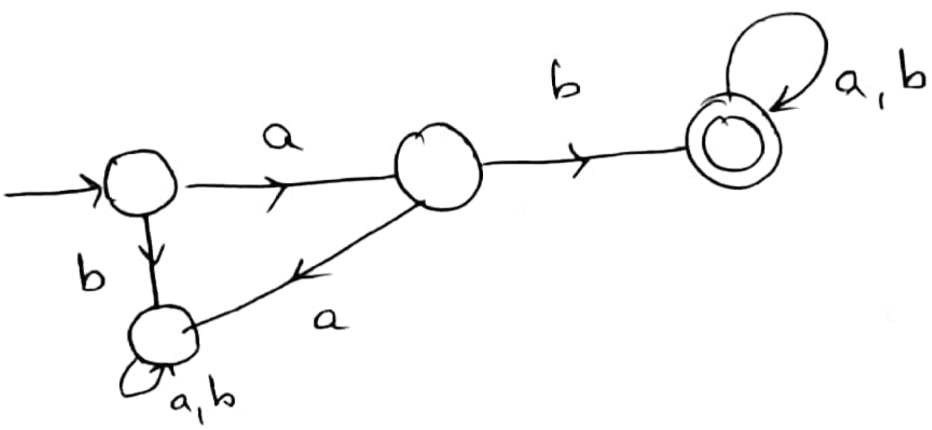
$n_a(w) \leq 2$

Q)



Q) Construct ~~D~~FA, for the language over the alphabet $\Sigma = \{a, b\}$ that contain all string that starts with ab.

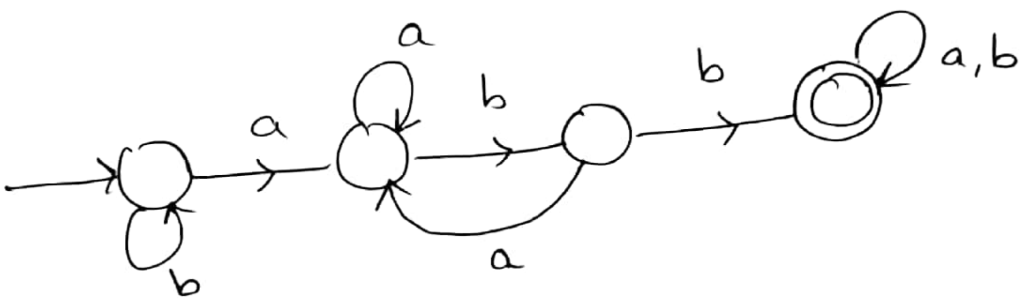
$L = \{ ab, aba, abb, abaa, abab, \ldots \}$



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

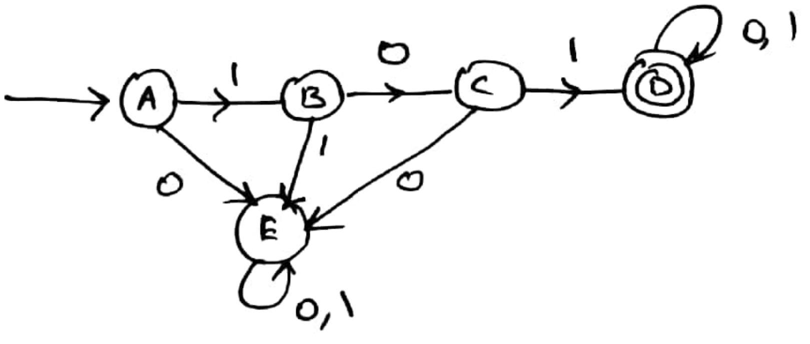
Language contains all strings that have abb as substring.

$L = \{ abb, aabb, abba, aabba, \ldots \}$



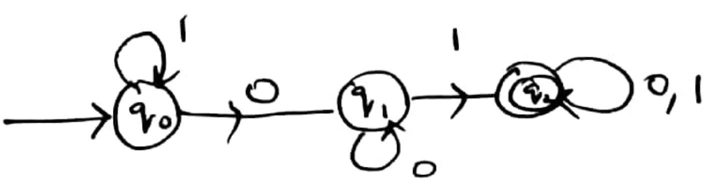
Q) $\Sigma = \{0, 1\}$ begins with 101

$L = \{ 101, 1010, 1011, 10100, \ldots \}$



Q) $\Sigma = \{0, 1\}$ Sequence 01 somewhere in the string.

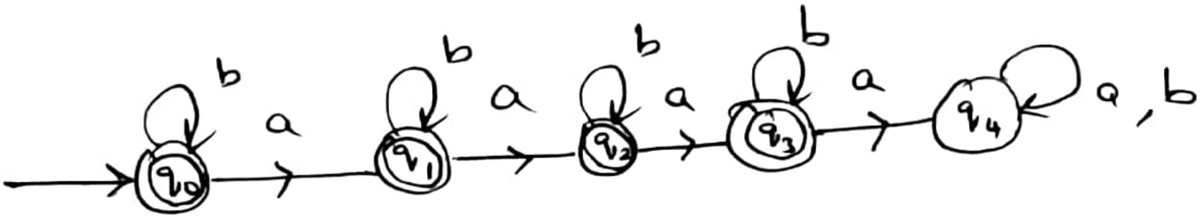
$L = \{ 01, 001, 010, 0101, \ldots \}$



Q) $\Sigma = \{a, b\}$ set of strings contains not more than 3 a's

ϵ, b, a, aa, ab, ba, bb, ab, ...

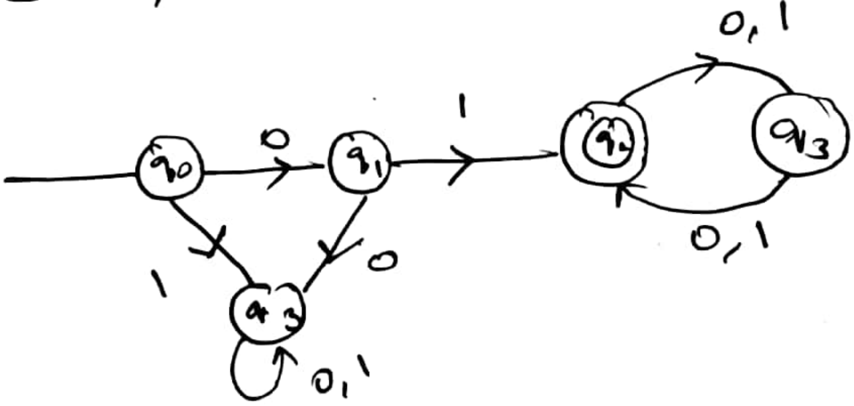
L = { aaa, baaa, aaabb, bbaaa, ababa, ...}



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

L = {w | w is of even length & begin with 01}

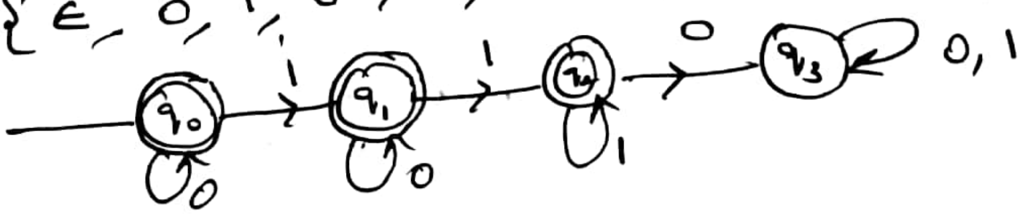
L = { 01, 0100, 0110, 0111, ... }



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

L = {w | w not containing 110}

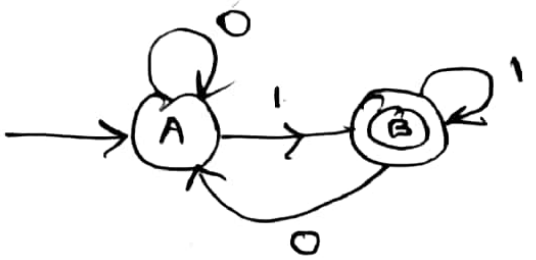
L = {ϵ, 0, 1, 00, 11, 10, 01, ...}



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

$L = \{ w \mid w \text{ ends in a } 1 \}$

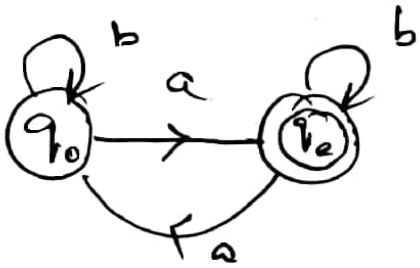
$L = \{ 1, 01, 11, 111, 001, 011, \dots \}$



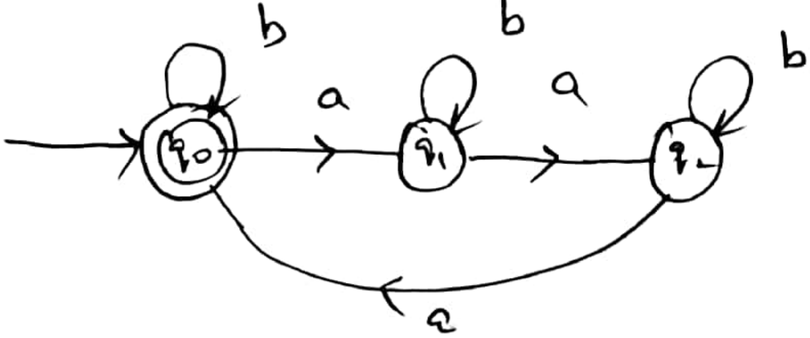
Q) $\Sigma = \{a, b\}$

$n_a(w) \bmod 2 = 0$

$n_a(w) \cong 0 \bmod 2$



$n_a(w) \bmod 3 = 0$



In general $n_a(w) \cong K \bmod N$

n number of states

Finite State System :- [state machine]

→ It is a mathematical model of computation used to design both computer programs & sequential circuits

Finite state Automata [FSA]

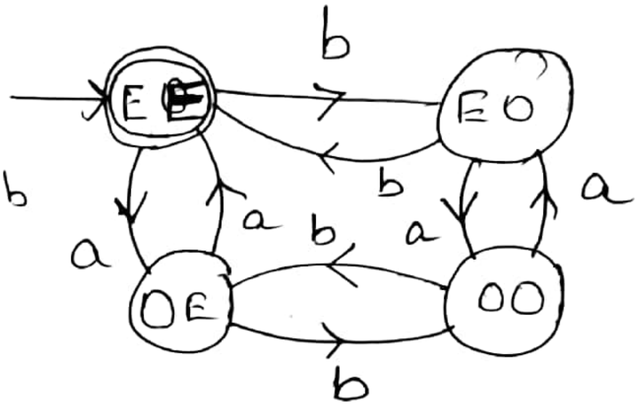
→ It is used to recognize patterns within input taken from some character sets

Q) $n_a(w) \cong 0 \bmod 2$ &&

$n_b(w) \cong 0 \bmod 2$

Method 1 $L = \{ \epsilon, aa, bb, aabb, abab, baba, bbbb, \ldots \}$

$n_a(w)$	$n_b(w)$	
e	o	– aab
o	e	– abb
e	e	ϵ, aa, bb
o	o	– ab



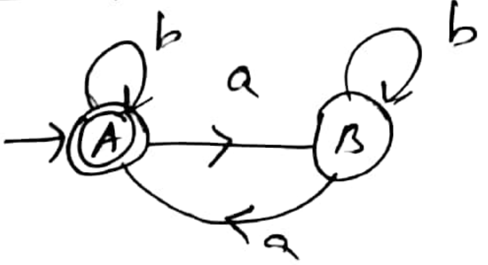
Method 2

Crossproduct Method $(n \times m)$

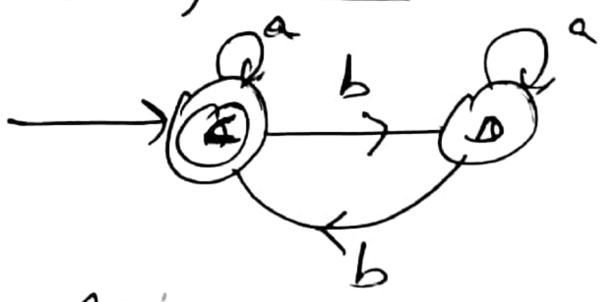
57

Reference No → CT20182465117

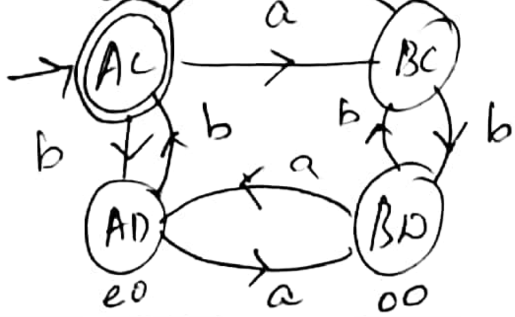
Counting a's



Counting b's



$\{A, B\} \times \{C, D\} = AC, AD, BC, BD$



$\begin{matrix} A \\ C \end{matrix} \xrightarrow{a} \begin{matrix} B \\ C \end{matrix}$

$\begin{matrix} A \\ C \end{matrix} \xrightarrow{b} \begin{matrix} A \\ D \end{matrix}$

$\begin{matrix} B \\ C \end{matrix} \xrightarrow{a} \begin{matrix} A \\ C \end{matrix}$

a'1 b'1

e xx = {ee}

o or e = {ee, eo, oe}

o

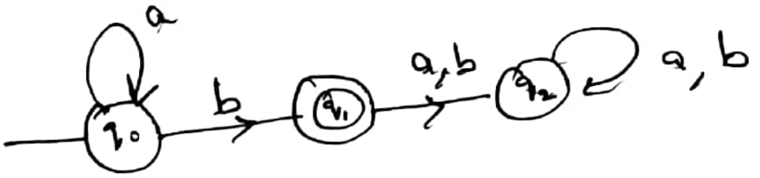
Q) $w \in \{a,b\}^*$ $n_a(w) \equiv 0 \bmod 3$ xx

$n_b(w) \equiv 0 \bmod 2$

Assignment

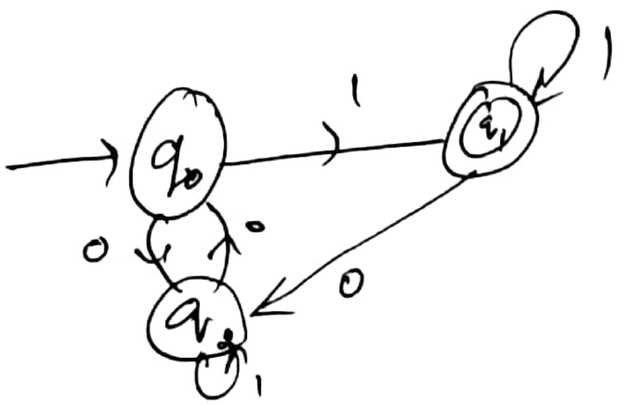
Q) $L = \{a^n b : n \geq 0\}$

L = {b, ab, aab, aaab ...}



Q) L = {w | w contains atleast one 1 and an even no of 0's followed by the last 1}

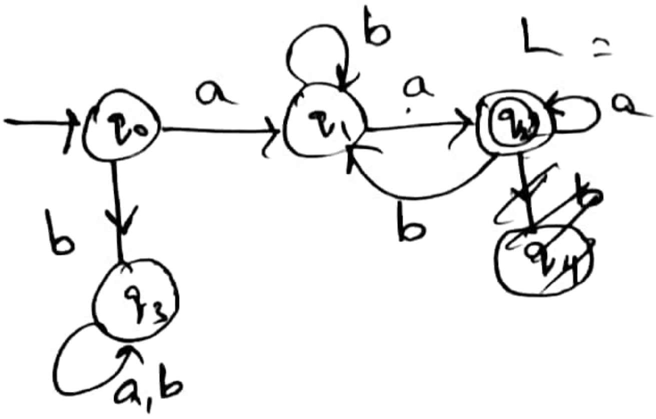
L = {1, 001, 00001, ... 11001}



Scanned by CamScanner

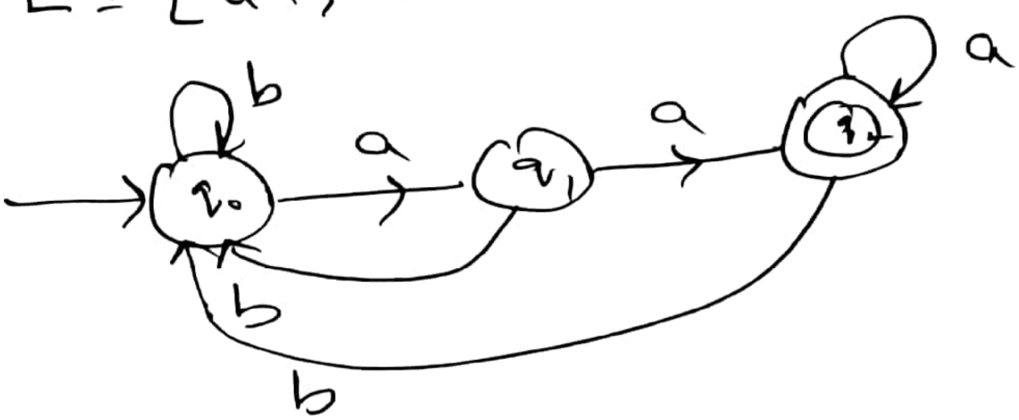
Q) $L = \{awa : w \in \{a, b\}^*\}$

$L = \{aa, aba, abba, abbba \ldots\}$



Q) $L = \{w \mid w \in \{a, b\}^*$ ends with aa$\}$

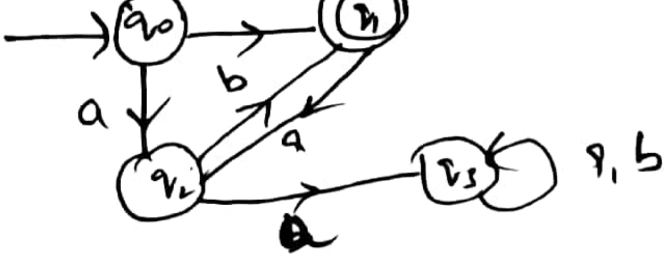
$L = \{aa, baa, bbaa, bbbaaa\}$



Assignment

Q) Ends with b and doesn't contain aa.

$L = \{b, ab, abb, abbab, ababb \ldots\}$



Extended Transition Function of DFA / Properties of Transition Function (δ)

Basic

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

The state of the system can be changed only by an input symbol else remains in original state.

Induction

w is the string of the form $w = xa$ where a is the last symbol of w and x is a string consisting of all but not the last symbol.

(eg) $w = 1101$ $\quad x = 110$ $\quad a = 1$

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

Q) Describe the Language by DFA

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

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

where δ is given by

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

give the transition diagram and entire sequence for i/p string 1101011 using induction method.

$$
\begin{aligned}
\delta(q_0, 1101011) &= \delta(\delta(q_0, 1), 101011) \\
&= \delta(q_1, 101011) \\
&= \delta(\delta(q_1, 1), 01011) \\
&= \delta(q_2, 01011) \\
&= \delta(\delta(q_2, 0), 1011) \\
&= \delta(q_2, 1011) \\
&= \delta(\delta(q_2, 1), 011) \\
&= \delta(q_1, 011) \\
&= \delta(\delta(q_1, 0), 11) \\
&= \delta(q_0, 11) \\
&= \delta((q_0, 1), 1) = \delta(q_1, 1) \\
&= q_2
\end{aligned}
$$

which is not a final state