Theory of compution:

mealy mechine: and moore mechine:

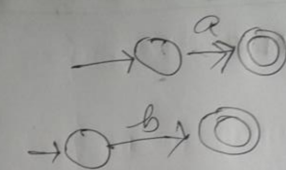
Moore

Regular Expression: (RE)

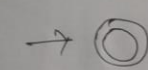
Regular expresion: — ① 6 case } Finite NAutomata

i) $R = a$; where $a \in \Sigma$

$R = b$



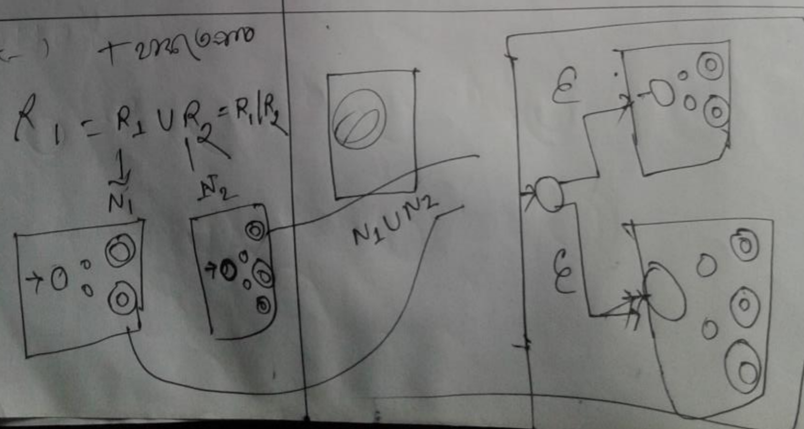
ii) $R = \varepsilon$ empty



iii) $R = \varphi$

(NFA)

iv) $R_1 = R_1 \cup R_2 = R_1 | R_2$

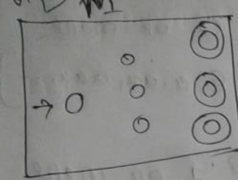


Tutorial-70

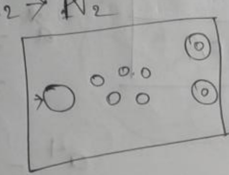
concatanation

(5) $R = R_1 R_2 = R_1 \circ R_2$

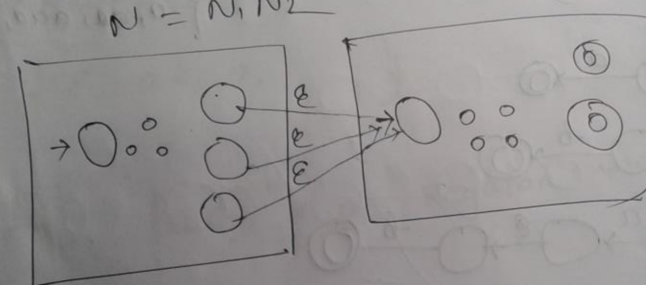
$R_1 \rightarrow N_1$



$R_2 \rightarrow N_2$

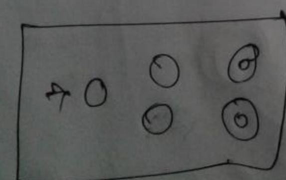


$N = N_1 N_2$

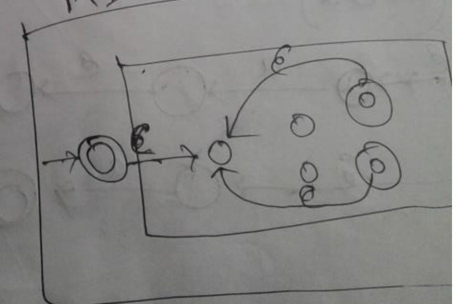


(6) $R = R_1^*$

$R_1 \rightarrow N_1$



R_1^*



Tutorial-71 ⟵ 74

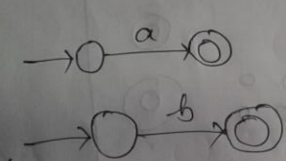
RE To FA
RE to NFA

$a^* = 0$ or more
$= \{\epsilon, a, aa, aaa, \ldots\}$

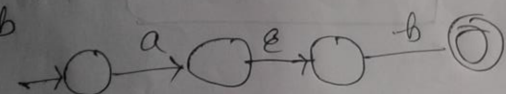
$a^+ = 1$ or more
$= \{a, aa, aaa, \ldots\}$

Example-1

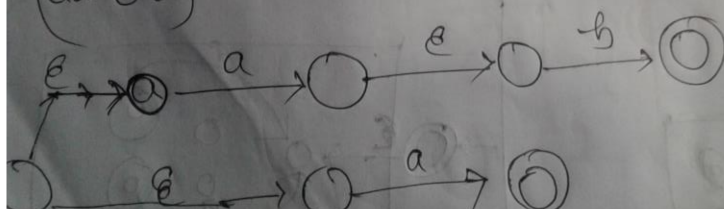
$(ab \cup a)^*$



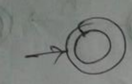
ab

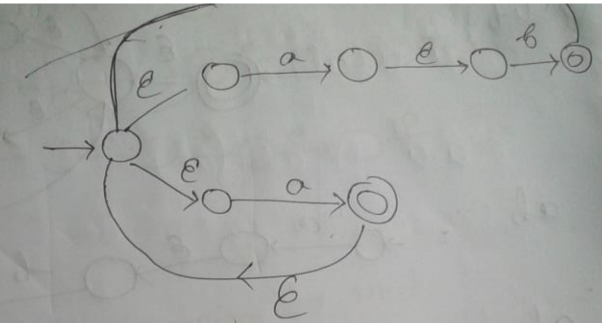


$(ab \cup a)$



$(ab \cup a)^*$

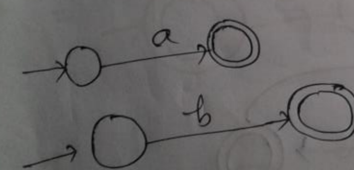




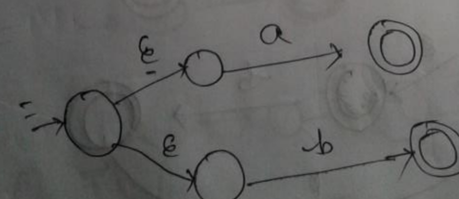
$= (ab \cup a)^*$

$= \{\varepsilon, ab, a, \ldots\}$

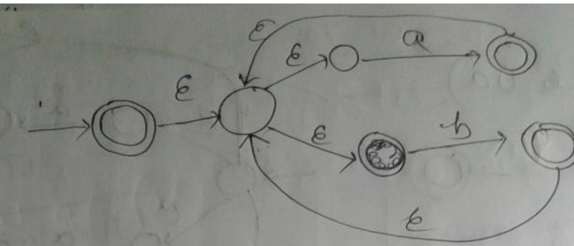
Example: $(a \cup b)^* aba$ Regular express to NFA



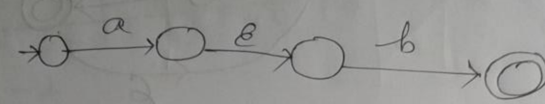
$a \cup b$ =



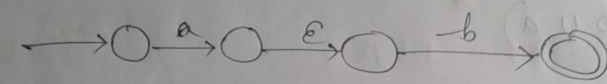
$(a \cup b)^* =$



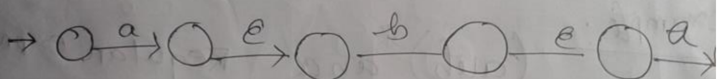
$ab =$



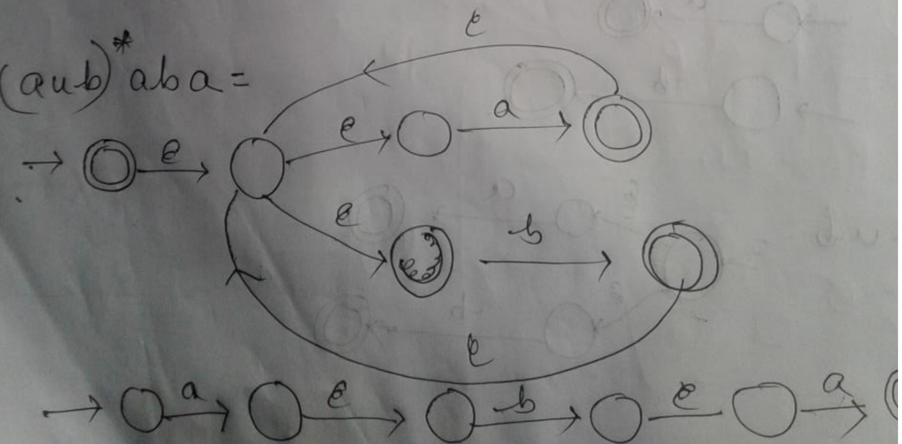
$aba =$

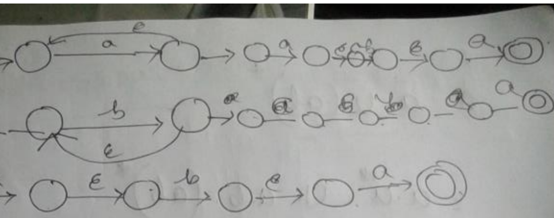


$aba =$



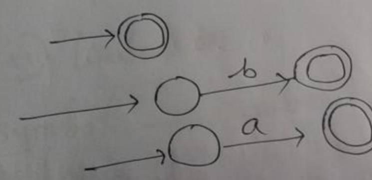
$(a \cup b)^* aba =$



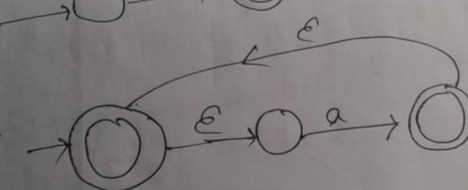


1.

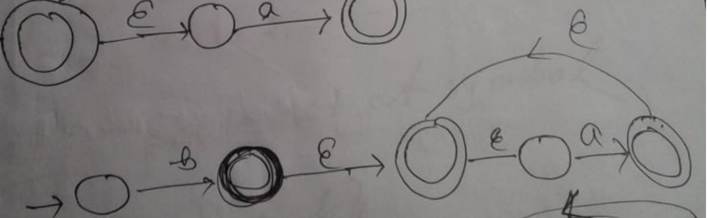
$ba^* + \varepsilon$ Regulen expnesion to NFA



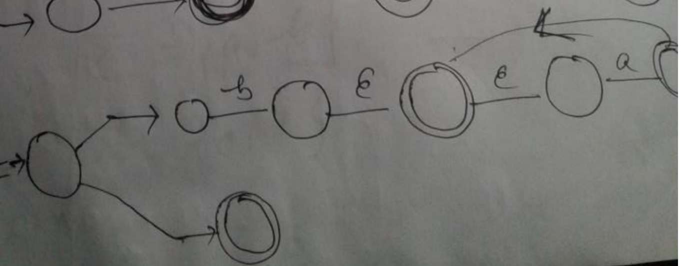
$a^* =$



$b\,a^* =$



$b\,a^* + \varepsilon =$



Theory of computation: Tutorial - 2

TOC

Theory of computation
machine

(i) Complexity
(ii) Capability
(iii) Limitation

symbol → symbol is the basic building block of toc.

Example → a, b, c, - - - -
0, 1, 2, - - -

Alphabet: Finite set of symbol

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

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

String → finite ~~set~~ squence of symbol.

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

$w = 0110$

write the string over the Σ of len = 2

an = {01, 10, 11, 00}

Tutorial: 2

Languge: collection of string. finite/infinite

Example

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

~~Lt~~ Language L_1 = set of all string over Σ of length 2

= {aa, ab, ba, bb} Finite languge.

Languge L_2 = set of all string over Σ of stan with a

$= \{a, abb, aaa, \ldots\}$ infinite

<u>power of Σ</u> $\Sigma = \{a, b\}$

Σ^1 = set of all string over this Σ of length

$= \{a, b\}$

Σ^2 = set of all string over this Σ of len 2

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

$\Sigma^3 = \{aaa, aab, aba, baa, bba, bbb\}$

$\Sigma^4 =$

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

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cdots$$

Σ^* $\boxed{L_1 \; L_2 \; L_3}$ $L_1 \subseteq \Sigma^*$

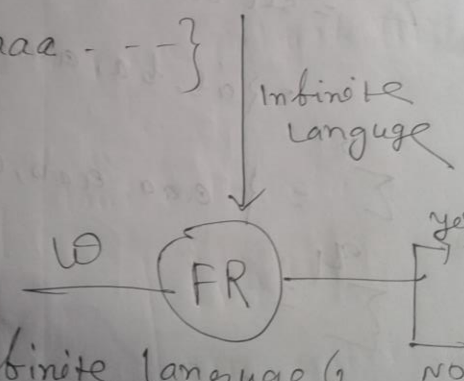
$L_2 \subseteq \Sigma^*$

Tutorial-4

Finite Automata:

$L(m)$ = set of all string start with 'a'

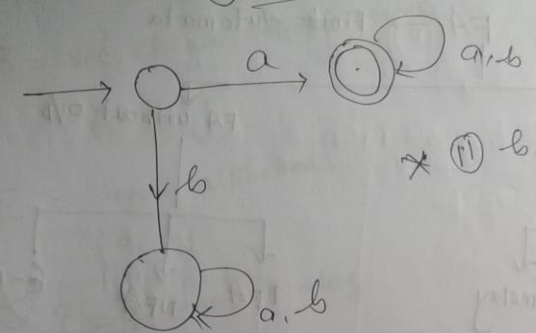
$= \{a, ab, aa, aaa \cdots\}$



infinite languge কে finite languge এ

ইহা নির্ধারণ Finite Automa করে

check (i) a $a \cdot b$ yes



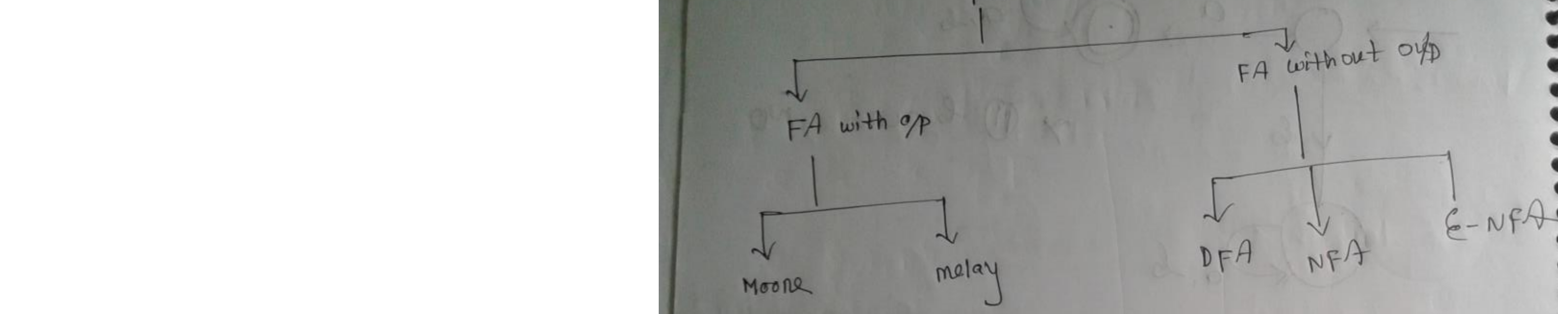
* (ii) b aa NO

In finite Language [illegible] Finite Representation

[illegible] Finite Automa/F. R.

Tutorial: 5

FA → Finite Automata



Determenestic Finite Automata (DFA)

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

L(M) = set of all string start with an 'a';

DFA

DFA

A DFA contains 5 tuples

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

* For each input there will go one state.

Q = set of all ~~string~~ states

$= \{A, B, C\}$

Σ = input alphabets

$\{a, b\}$

q_0 = start states

$= A$

F = set of Final states (অনেক গুলো থাকতে পারে)

$= B$

$\delta \rightarrow$ transition function

$\delta = \underbrace{Q}_{\text{state}} \times \underbrace{\Sigma}_{\text{input}} \rightarrow Q$

states ↓	a	b
→ A	B	C
* B	B	B
C	C	C

একটি input এর অবশ্যই অন্য একটি stats যাওয়া লাগবে।

Tutorial-6

Formal definition of DFA

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

$Q = \{A, B\}$

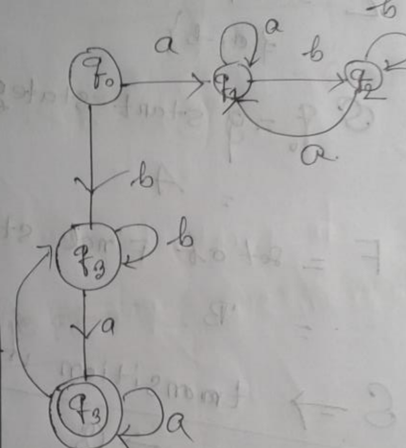
$\Sigma = \{a, b\}$

$F = \{B\}$

$q_0 = A$

$\delta =$

	a	b
A	B	A
B	A	B



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

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

$q_0 = q_0$

F. $F = \{q_2, q_4\}$

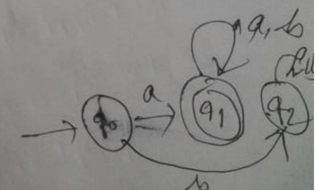
M_3	a	b
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_2
q_3	q_4	q_3
q_4	q_4	q_3

The formal description of a DFA M is

$(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_1\})$ where δ is given by the following table

$Q_0 = \{q_1, q_1, q_2\}$

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

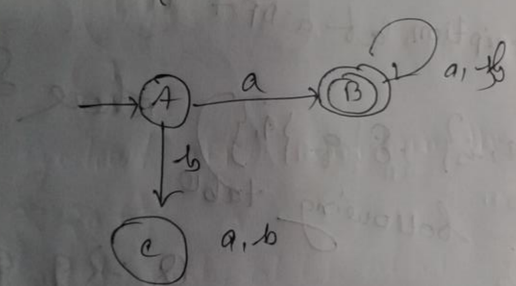


problem:

Construct a DFA which accept set of all string over the $\Sigma = \{a, b\}$ where each string start with 'a'

Sol^n: $\Sigma = \{a, b\}$

$L = \{a, aaa, aaab, abbb, \ldots\}$

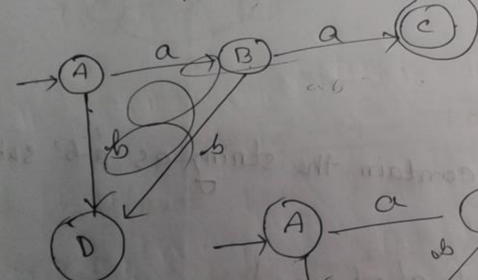


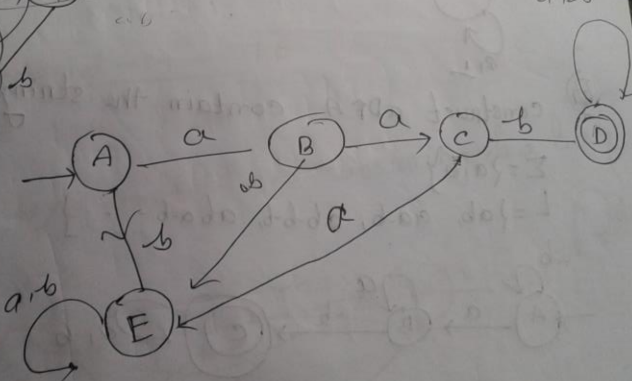
Tutorial —

Q. Construct a DFA machine which set of all string over $\Sigma = \{a, b\}$ where each string start with "aab"

Sol^n: $\Sigma = \{a, b\}$

$L = \{aab, aaba, aabb, \ldots\}$

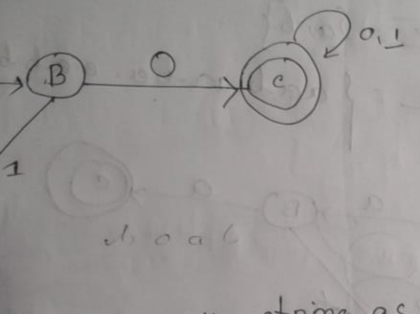




Construct DFA start with '10''

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

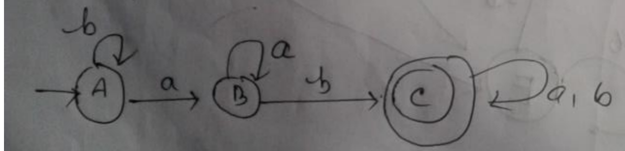
$L = \{10, 1000, 1011, ---\}$



Construct a DFA contain the string as 'ab' substring

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

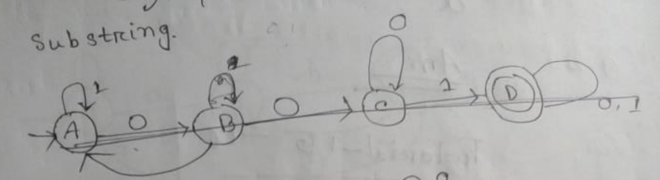
$L = \{ab, aab, abbb, abab ---\}$

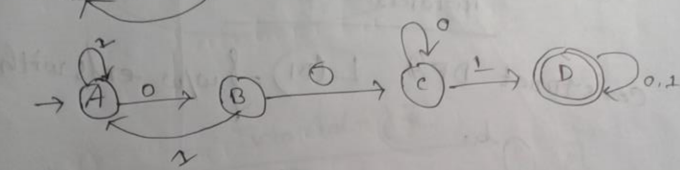


Tutorial — 13

Construct DFA = $\{0, 1\}$

$L(M) = \{ w/w$ contains the string 001 as a substring.

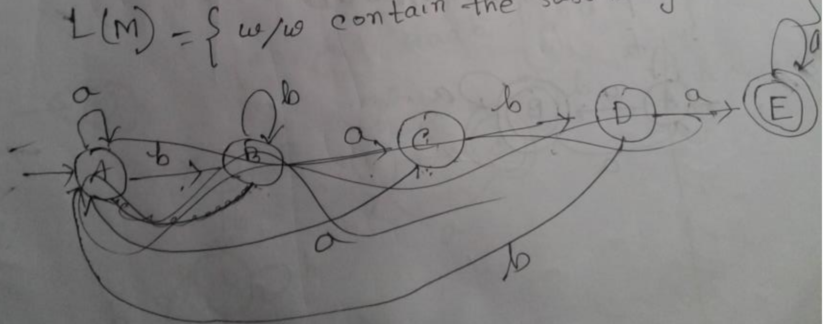


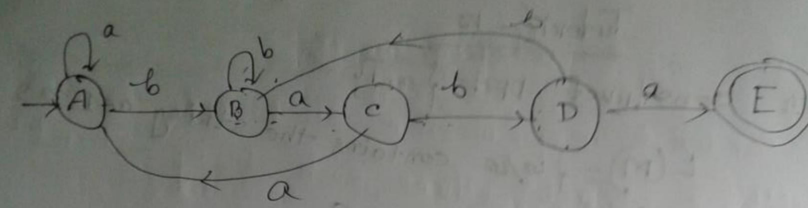


Tutorial — 14 — 16

Construct DFA = $\{a, b\}$

$L(M) = \{ w/w$ contain the substring $baba\}$

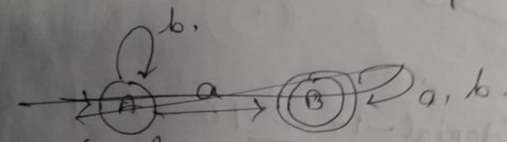




Ans:

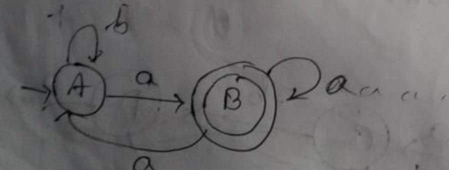
Tutorial-15-16

Q. Construct DFA $L(M) = \{ w / w \text{ ends with 'a'} \}$



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

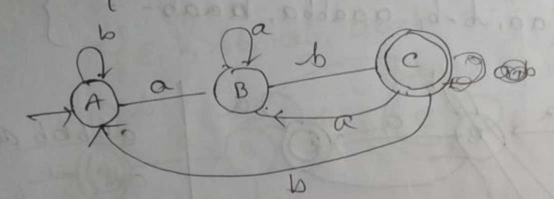
$L = \{a, aa, aba, bba, \ldots\}$



Q. $L = \{ w/w \text{ ends with 'ab'} \}$

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

$L = \{ab, aab, bbab, \ldots\}$

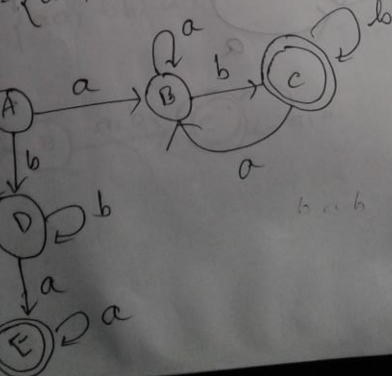


Tutorial - 12

Q. Construct a DFA where $\Sigma = \{a, b\}$

$L(M) = \{ w/w \text{ starts and ends with different symbol} \}$

$L = \{ab, ba, aaab, bbba, \ldots\}$

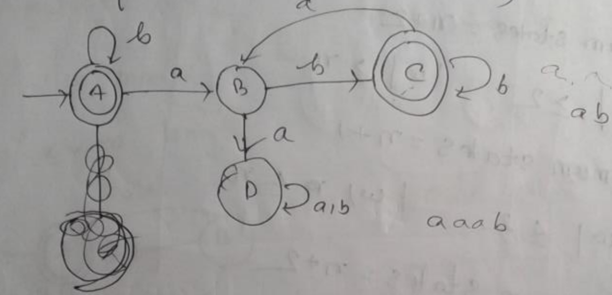


Tutorial – 19

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

$L(M) = \{ w / w$ every a in w is followed by one 'b' $\}$

$= \{ \varepsilon, b, bb, ab, abab \ldots \}$

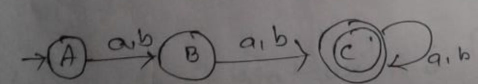


aaab

* Construct a DFA

$L(M) = \{ w / w$ every string which at least length 2 $\}$

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



Tutorial - 27

1\. **Minimal String Length**

$|\omega| = 2$ / $|\omega| = n$

Minimum states $= n+2$

ii) $|\omega| \geq 2$ $\quad |\omega| \geq n$

Minimum states $= n+1$

(iii) $|\omega| \leq 2$ $\quad |\omega| \leq n$

Minimum states $= n+2$

iv) divisible by n

Number of states $= n$

Tutorial → Tube Learning

Moore mechine:

Moore mechine called memory less mechine.

Moore mechine output depand on current stage

There is no relation current input

	Input		output
	0	1	
q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

FOR 0 1 1 1 moore mechine

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

0 0 0 1 0

0 0 0 1 0

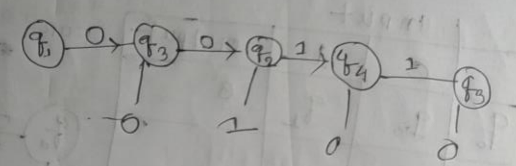
~~Mealy~~ mechine:

mealy mechine:

mealy mechine output depend on current states and current input.

	0	1
q_1	$q_3 \to 0$	$q_2 \to 0$
q_2	$q_4 \to 1$	$q_4 \to 0$
q_3	$q_2 \to 1$	$q_1 \to 1$
q_4	$q_4 \to 1$	$q_3 \to 0$

0 0 1 1



0 1 0 0

For input 0011 respect output mealy mechine is 0 1 0 0

Conversion from mealy machine to moore machine

	0	1
q_1	$q_3 \to 0$	$q_2 \to 0$
q_2	$q_1 \to 1$	$q_4 \to 0$
q_3	$q_2 \to 1$	$q_1 \to 1$
q_4	$q_4 \to 1$	$q_3 \to 0$

q_2 — q_{20}, q_{21}

q_4 — q_{40}, q_{41}

Anothe table

	0	1
q_1	$q_3 \to 0$	$q_{20} \to 0$
q_{20}	$q_1 \to 1$	$q_{40} \to 0$
q_3	$q_{21} \to 1$	$q_1 \to 1$
q_{40}	$q_{41} \to 1$	$q_3 \to 0$
q_{21}	$q_1 \to 1$	$q_{40} \to 0$
q_{41}	$q_{41} \to 1$	$q_3 \to 0$

moore mechine

	0	1	
q_0			
q_1			
q_2			
q_3			
q_4			
q_5			

moore mechine

q_1	0	1	output
q_1	q_3	q_{20}	0
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	1
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

Non-determisistic Finite automata (NFA)

NFA →
- 1. more than one transition
- 2. one "
- 3. zero

$DFA: Q \times \Sigma \rightarrow Q$

$NFA: Q \times \Sigma \rightarrow 2^Q$

Formal defination of NFA

1 $\{Q, \Sigma, \delta, q_0, F)$

Q = (Number of states}

Σ = (" " Input}

q_0 = start state

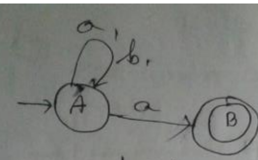
F = set of final states

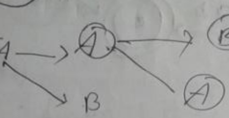
$\delta = Q \times \Sigma = 2^Q$

Example → L = { set of all string which end with 'a'

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

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

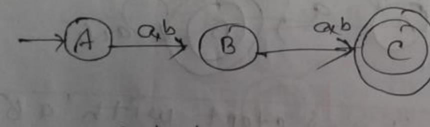




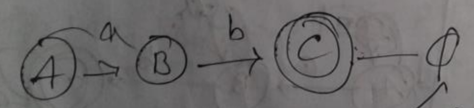
* construct a NFA, $\Sigma = \{a, b\}$

$L = \{ w / w$ the length of string exactly $2 \}$

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

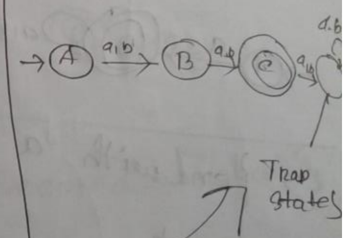


a b b



Death config:

FOR DFA



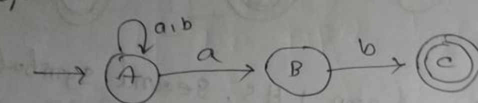
Trap States

NFA to DFA conversion:

DFA - NFA hand ...

L = {w/w ends with ab}

NFA:



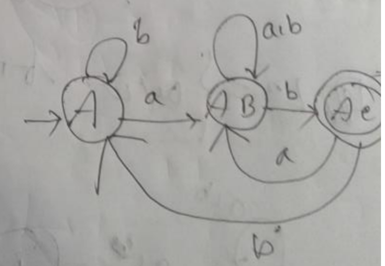
state transition table

	a	b
→ A	{A, B}	{A}
B	{ }	{c}
* c	{ }	{ }

subset construction for DFA.

DFA

	a	b
→ A	[A, B]	[A]
[A, B]	[A, B]	[A, c]
* [A, c]	[A B]	[A]

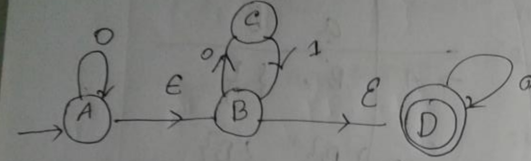


cas-T. -

Epsilon ~~eNFA~~ to NF

topic →

Epsilon NFA to NFA conversi



ε-closer (A) = always close A

ε NFA ⟶ NFA

(Number of states same)

δ (ε closer (A), 0)

	0	1
A	{A, B, C, D}	{D}
B	{D}	{D}
C	{}	{B, D}
D	{D}	{D}

	ϵ^*	0	ϵ^*
A →	A →	A ⟶	A, B, D
	B →	C ⟶	C
	D →	D ⟶	D

	ϵ^*	1	ϵ^*
A →	A ⟶	φ	
	B ⟶	φ	
	D →	D ⟶	D

$\delta(\epsilon\text{ closer}(B), 0)$

$B \xrightarrow{\epsilon^*} B \xrightarrow{0} C \xrightarrow{\epsilon^*} C$

$B \xrightarrow{\epsilon^*} D \xrightarrow{0} D \xrightarrow{\epsilon^*} D$

$B \xrightarrow{\epsilon^*} B \xrightarrow{1} \phi$

$B \xrightarrow{\epsilon^*} D \xrightarrow{1} D \xrightarrow{\epsilon^*} D$

$C \xrightarrow{\epsilon^*} C \xrightarrow{0} \phi$

$C \xrightarrow{\epsilon^*} C \xrightarrow{1} B \xrightarrow{\epsilon^*} D$, B

$D \xrightarrow{\epsilon^*} D \xrightarrow{0} D \xrightarrow{\epsilon^*} D$

$D \xrightarrow{\epsilon^*} D \xrightarrow{1} D \xrightarrow{\epsilon^*} D$

Diagram