(12-09-2023)

Mealy and Moore Machine (Slide)

(Page-3)

Moore ⇒ $\lambda \to Q = \Delta$ | Mealy ⇒ $\lambda \to Q * \Sigma = \Delta$

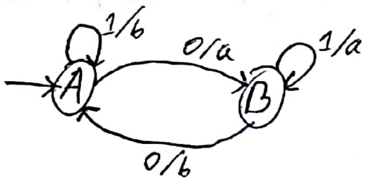
Six Tuple : $(Q, \Sigma, \delta, q_0, \Delta, \lambda)$

Σ → sigma, δ → Delta, Δ → Del, λ → Lemda

Q = Set of state,
Σ = Set of Alphabet, δ = Transition, q_0 = Initial state,
Δ = Set of Output alphabet (Output symbol)
λ = Output Mapping Function

(Page-4)

Mealy Machine

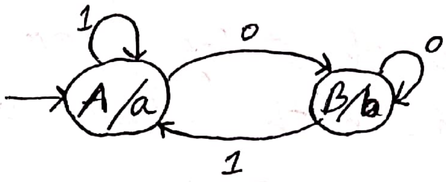


$\Sigma = \{0, 1\}$
$\Delta = \{a, b\}$

Example: 1 0 1 0 → b a a b

$A \xrightarrow[b]{1} A \xrightarrow[a]{0} B \xrightarrow[a]{1} B \xrightarrow[b]{0} A$

Moore Machine



Example: 1010 → a a b a b

$\underset{a}{A} \xrightarrow{1} \underset{a}{A} \xrightarrow{0} \underset{b}{B} \xrightarrow{1} \underset{a}{A} \xrightarrow{0} \underset{b}{B}$

(1 1 0 1 1)

Mealy

$A \xrightarrow[b]{1} A \xrightarrow[b]{1} A \xrightarrow[a]{0} B \xrightarrow[a]{1} B \xrightarrow[a]{1} B$

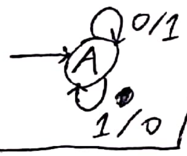
b b a a a

Moore

$\underset{a}{A} \xrightarrow{1} \underset{a}{A} \xrightarrow{1} \underset{a}{A} \xrightarrow{0} \underset{b}{B} \xrightarrow{1} \underset{a}{A} \xrightarrow{1} \underset{a}{A}$

aaabaa

(Page-6) Construction of Mealy Machine by 1's complement



1010 → 0101

$A \xrightarrow{1} \underset{0}{A} \xrightarrow{0} \underset{1}{A} \xrightarrow{1} \underset{0}{A} \xrightarrow{0} \underset{1}{A}$

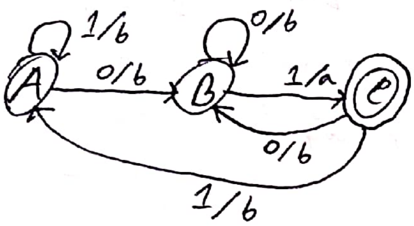
~~(Page-7)~~

(Page-7)

2.) Construct a Mealy Machine that prints 'a' whenever the sequence '01' is encountered in any input binary string.

$01 \rightarrow a$

Ans:



$0001 \rightarrow bbba$

$0010 \rightarrow bbab$

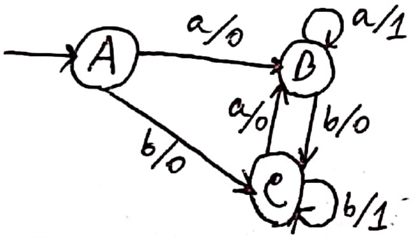
$1000 \rightarrow bbbb$

$\Sigma = \{0, 1\}$, $\Delta = \{a, b\}$

(14-09-2023)

3) Construct a Mealy machine that accepts the language consisting of string over Σ^*, where $\Sigma = \{a, b\}$ and the strings should end with either aa or bb.

⇒



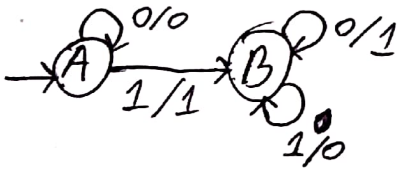
$\Sigma = \{a, b\}$

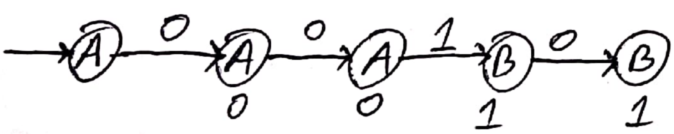
$\Delta = \{0, 1\}$

---

4) Construct a Mealy machine that produces 2's complement of any binary string. (Assume that the last carry bit is ignored)

⇒





2's complement

1001

0110 → 1's comp

\+ 1

0111 → 2's comp

(18-09-2023)

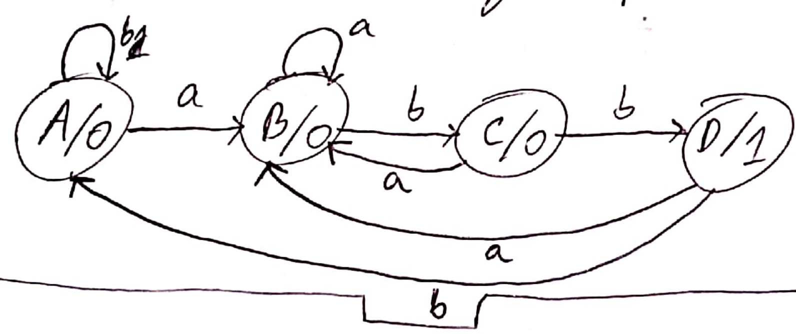
* **Moore machine:** A ~~m~~ Moore machine is a type of finite-state machine where outputs are determined solely by the current state and each state is associated with a fixed output symbol or value.

1) Construction of a Moore machine that prints 'a' whenever the sequence '01' is encountered in any input binary string.

⇒

2) Construction of a Moore machine that accepts the sequence 'abb' in any input strings over {a, b}.

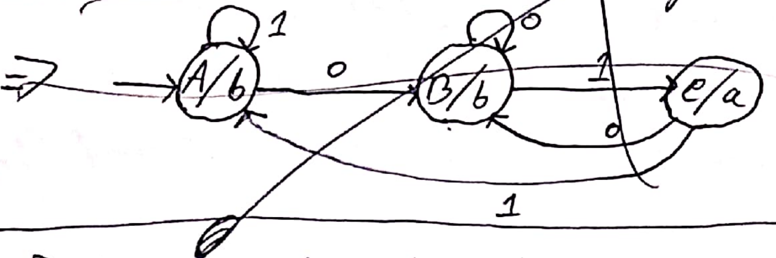
⇒



(18-09-2023)

Construction of Moore Machine

1) Construct a Mealy machine that produces the 1's complement of any binary string.



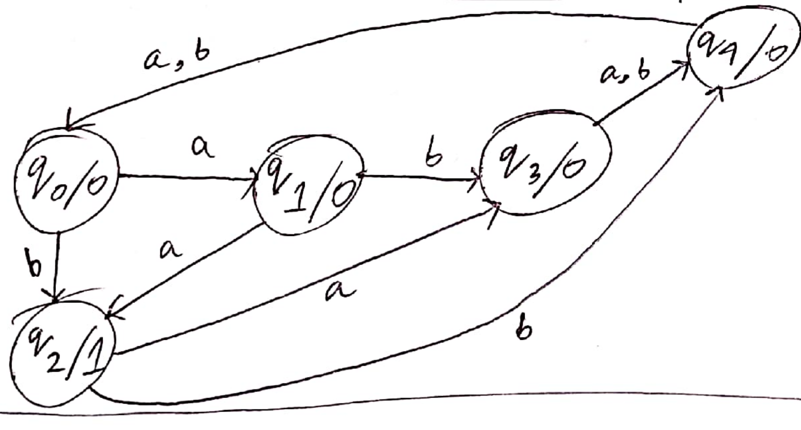
~~2~~ 3) Construction of a Moore machine for the following and run the input sequences.

i) aabab
ii) abbb
iii) ababb

States	a	b	Outputs
q_0	q_1	q_2	0
q_1	q_2	q_3	0
q_2	q_3	q_4	1
q_3	q_4	q_4	0
q_4	q_0	q_0	0

~~i) aabab~~

Machine:



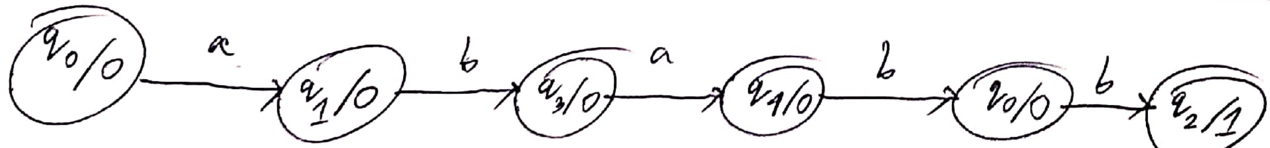
i)

aabab → 001001

ii)

abbb → 00000

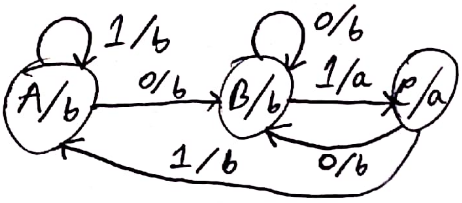
iii)



ababb → 000001

(19-09-2023)

① Moore Machine to Mealy machine.

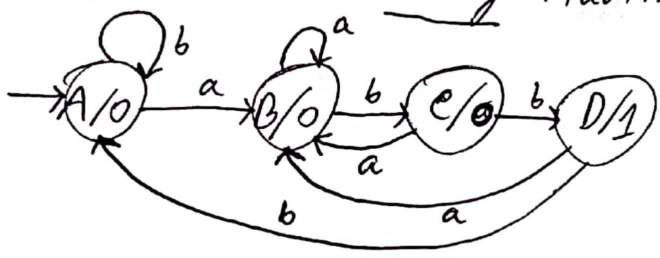


state	0	1	O/P
A	B	A	b
B	B	C	b
C	B	A	a

State	0	1
A	B, b	A, b
B	B, b	C, a
C	B, b	A, b

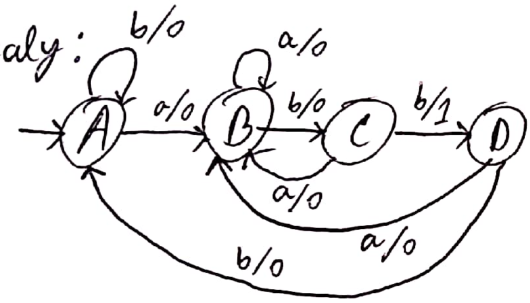
② Moore Machine to Mealy Machine.

Moore:



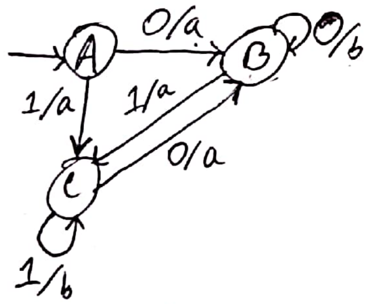
state	a	b	O/P
A	B	A	0
B	B	C	0
C	B	D	0
D	B	A	1

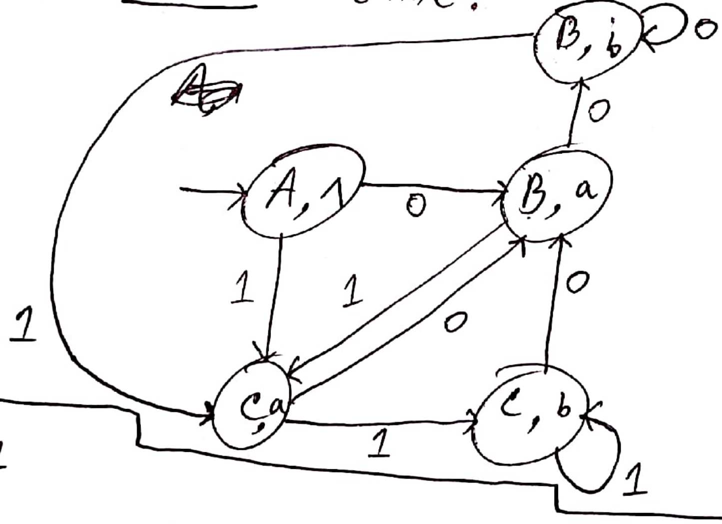
Mealy:



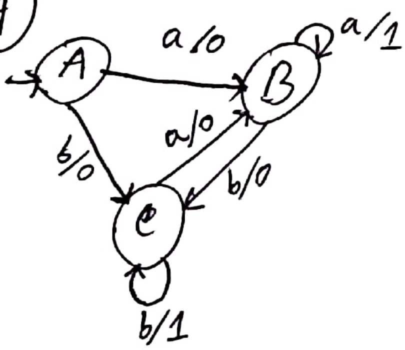
State	a	b
A	B, 0	A, 0
B	B, 0	C, 0
C	B, 0	D, 1
D	B, 0	A, 0

③ Mealy Machine to Moore Machine.



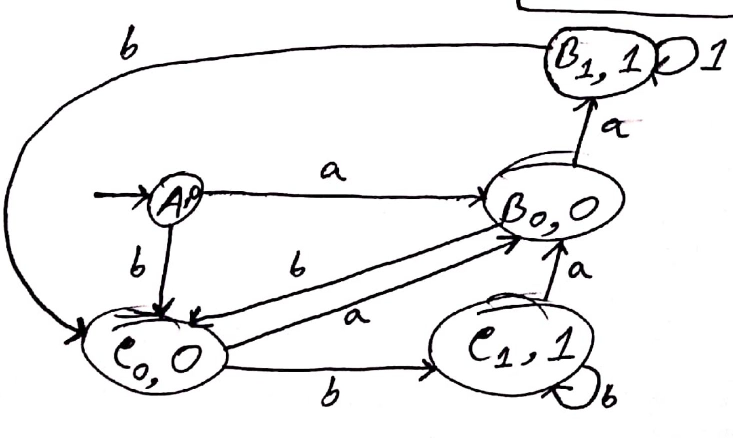


④



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

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



print 1 when 'aa' or 'bb'

Σ^*, $\Sigma = \{a, b\}$

(Mealy to Moore)

(Neso Academy YT)

(5) (Moore machine to Mealy Machine)

Ans:

Moore

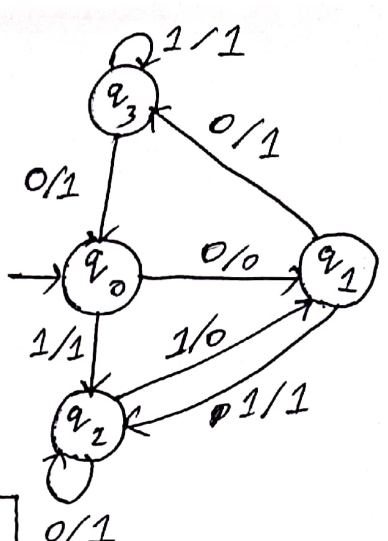
State	0	1	Output
q_0	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_1	1
q_3	q_0	q_3	1

$\Rightarrow$

Mealy

State	0	1
q_0	$q_1, 0$	$q_2, 1$
q_1	$q_3, 1$	$q_2, 1$
q_2	$q_2, 1$	$q_1, 0$
q_3	$q_0, 1$	$q_3, 1$

$\Rightarrow$



(6) (Mealy machine to Moore machine)

Convert the given Mealy Machine that give the 2's complement of any binary input to its equivalent MOORE MACHINE.

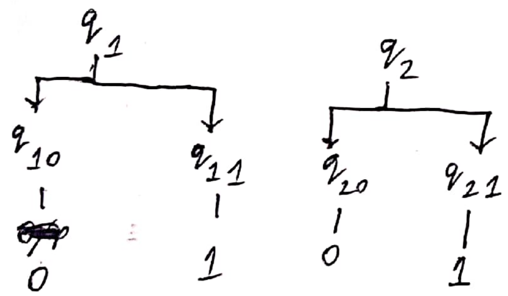
Ans:

(7) Convert the given Mealy Machine to it's equivalent Moore Machine.

State	a	b
q_0	$q_3, 0$	$q_1, 1$
q_1	$q_0, 1$	$q_3, 0$
q_2	$q_2, 1$	$q_2, 0$
q_3	$q_1, 0$	$q_0, 1$

$\Rightarrow$

State	a	b	Output
q_0	$q_3, 0$	$q_{11}, 1$	1
q_{10}	$q_0, 1$	$q_3, 0$	0
q_{11}	$q_0, 1$	$q_3, 0$	1
q_{20}	$q_{21}, 1$	$q_{20}, 0$	0
q_{21}	$q_{21}, 1$	$q_{20}, 0$	1
q_3	$q_{10}, 0$	$q_0, 1$	0



(TOC)

(25-09-2023) Id: 06922200 5101009

Slide: Regular expressions.

(Page-8) Designing Regular Expression

Design RE for the following languages over $\{a, b\}$

i) Language accepting strings of length exactly 2.

$\Rightarrow$

$R = aa+ab+ba+bb = a(a+b)+b(a+b) = (a+b)(a+b)$

ii) Language accepting strings of length at least 2.

$\Rightarrow$ $R = (a+b)(a+b)(a+b)^* = (a\backslash b)(a\backslash b)(a|b)^*$

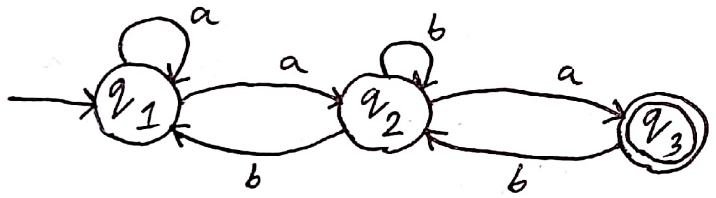
iii) Language accepting strings of length at most 2.

$\Rightarrow L = \epsilon, a, b, aa, ab, ba+bb$

$R = \epsilon+a+b+aa+ab+ba+bb = (\epsilon+a+b)(\epsilon+a+b)$

(Page-9) Find the RE for the following NFA.

$\Rightarrow$



$q_3 = q_2 \cdot a$ — (I)

$q_2 = q_1 \cdot a + q_2 \cdot b + q_3 \cdot b$ — (II)

$q_1 = \epsilon + q_1 \cdot a + q_2 \cdot b$ — (III)

(I) $\Rightarrow q_3 = q_2 a = (q_1 \cdot a + q_2 \cdot b + q_3 \cdot b)a = q_1 aa + q_2 ba + q_3 ba$ — (IV)

(II) $\Rightarrow q_2 = q_1 a + q_2 b + q_3 b = q_1 a + q_2 b + q_2 ab$

$= \underbrace{q_1 a}_{Q} + \underbrace{q_2}_{R} \underbrace{(b+ab)}_{P} = q_1 a (b+ab)^*$ — (V)

(III) $\Rightarrow q_1 = \epsilon + q_1 \cdot a + (q_1 a (b+ab)^*) b$

$= \epsilon + q_1 (a + a(b+ab)^*) b$

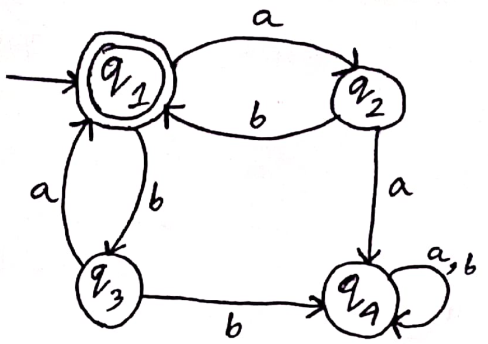
$= \epsilon ((a + a(b+ab)^*) b)^* = (a + a(b+ab)^* b)^*$ — (VI)

So, $q_3 = q_2 \cdot a = q_1 \cdot a (b+ab)^* \cdot a$

$= (a + a(b+ab)^* b)^* \cdot a (b+ab)^* \cdot a \longrightarrow RE$

(05-10-2023)

(Page-11) Find the RE for the following DFA.



$q_1 = \epsilon + q_2 b + q_3 a$ —— (I)

$q_2 = q_1 \cdot a$ —— (II)

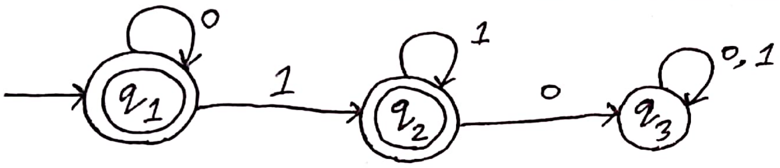
$q_3 = q_1 \cdot b$ —— (III)

$q_4 = q_2 \cdot a + q_3 b + q_4 \cdot a + q_4 \cdot b$ —— (IV)

(I) $\Rightarrow \underbrace{q_1}_{R} = \epsilon + q_2 b + q_3 a = \epsilon + q_1 ab + q_1 ba = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \underbrace{(ab+ba)}_{P}$

$= \epsilon(ab+ba)^* = (ab+ba)^* \rightarrow RE$

(Page-13) Find the RE for the following FA.



$q_1 = \epsilon + q_1 \cdot 0$ —— (I)

$q_2 = q_1 \cdot 1 + q_2 \cdot 1$ —— (II)

$q_3 = q_2 \cdot 0 + q_3 \cdot 0 + q_3 \cdot 1$ —— (III)

(I) $\Rightarrow \underbrace{q_1}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_1}_{R} \cdot \underbrace{0}_{P} = \epsilon \cdot 0^* = 0^*$ $[\epsilon \cdot R = R]$

(II) $\Rightarrow q_2 = 0^* \cdot 1 + q_2 \cdot 1 = 0^* \cdot 1 \cdot (1)^*$

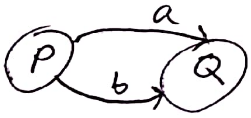
$RE = q_1 \cup q_2 = 0^* + 0^* \cdot 1 \cdot (1)^* = 0^*(\epsilon + 1\ 1^*)$ $[\epsilon + RR^* = R^*]$

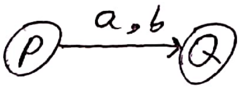
$= 0^*\ 1^*$.

(Page-15) RE to FA

1) $(a+b) \rightarrow$ (A or B)

$\Rightarrow$

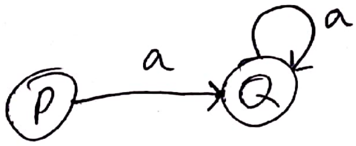




2) $(a.b) \Rightarrow$

3) $a^* \Rightarrow$

4) $a^+ \Rightarrow$



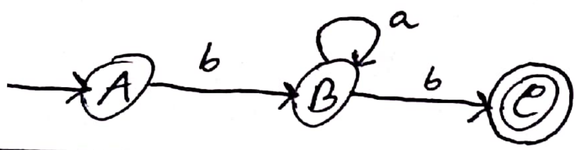
5) $a|b \Rightarrow (a+b)$

(Page-16) RE to FA

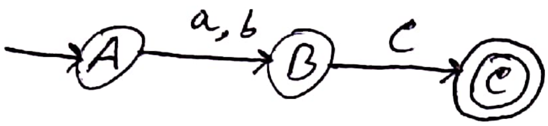
L

1) $ba^*b \rightarrow bb, bab, baab, baa\ldots ab$

$\Rightarrow$



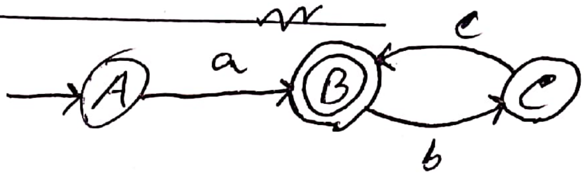
2) $(a+b)c \Rightarrow$



3) $a(bc)^* \rightarrow a, abc, abcbc\ldots\ldots$

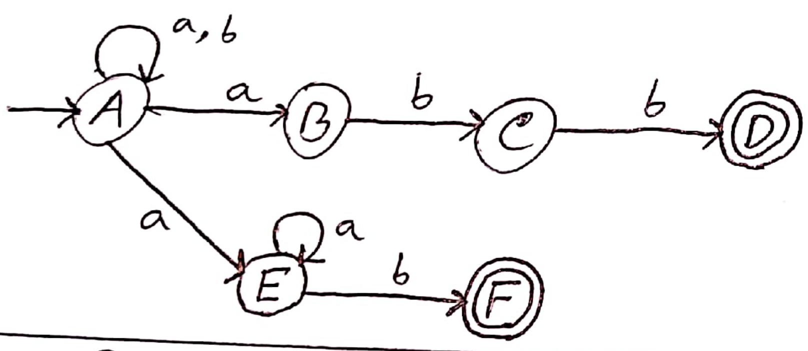
$(a')^*$ $A' = BC$

$\Rightarrow$



4) $(a|b)^*(abb|a^+b)$

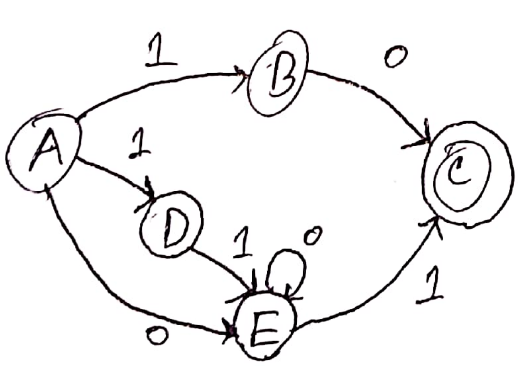
$\Rightarrow$



(Page-18) RE to FA

5) $10 + (0+11)0^*1$

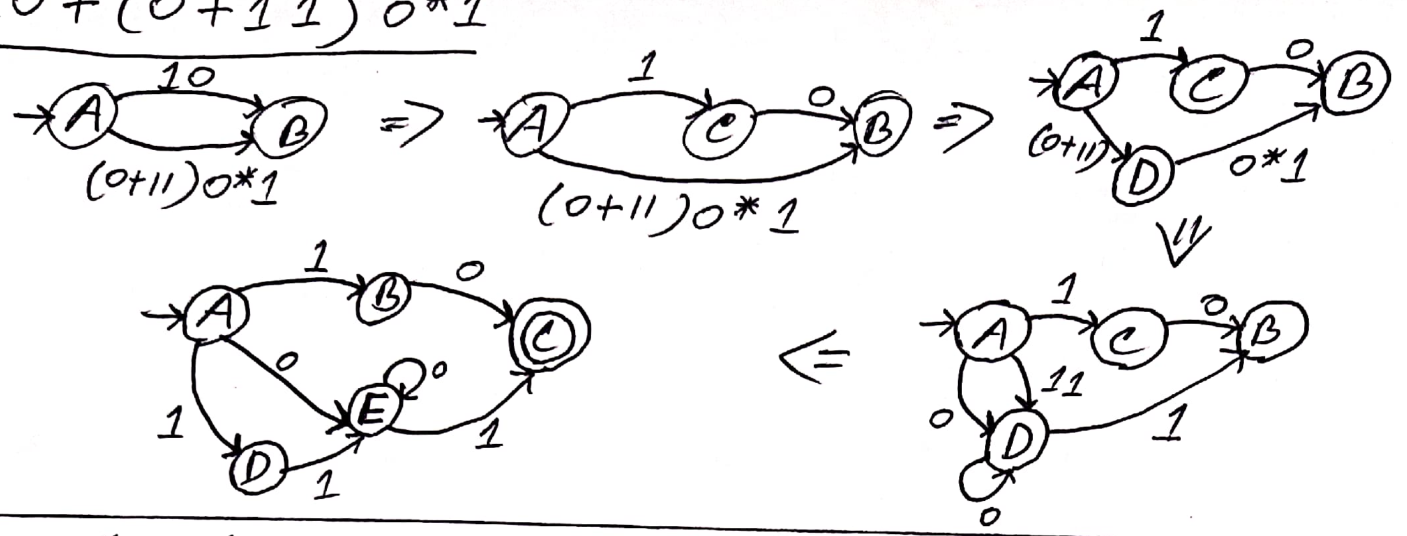
$\Rightarrow$



(10-10-23) (RE to FA)

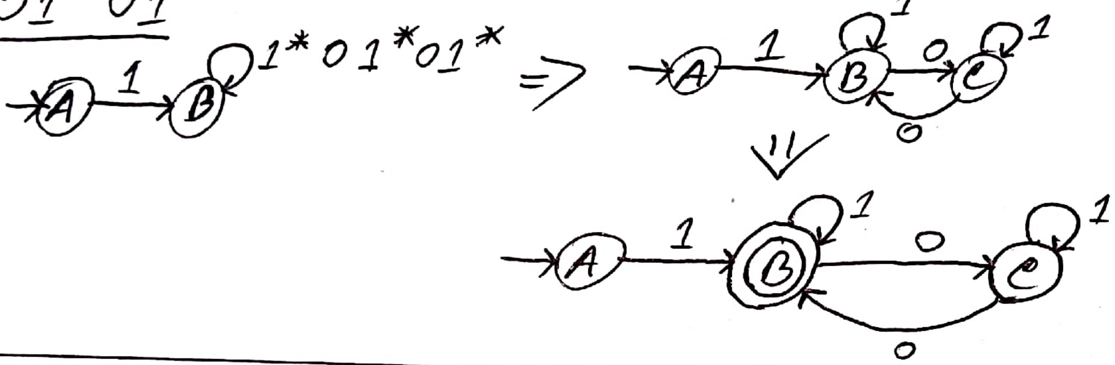
① $10+(0+11)0^*1$

Ans:



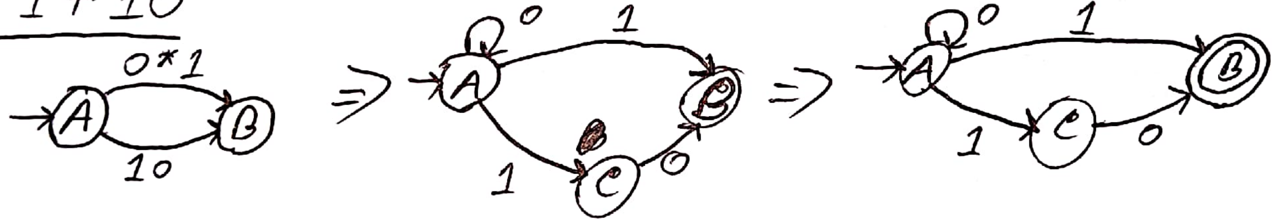
② $1^*01^*01^*$

Ans:



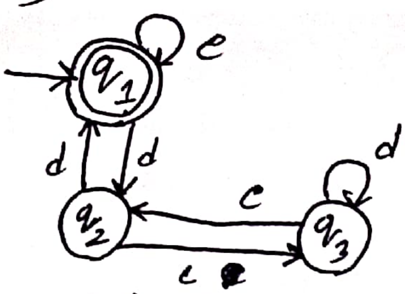
③ 0^*1+10

Ans:

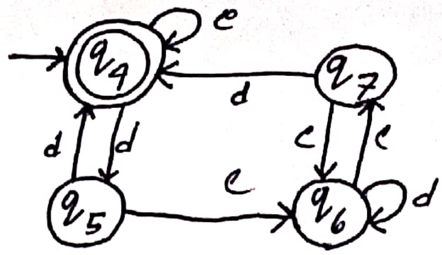


(12-10-2023)

Example-1



Automation-1

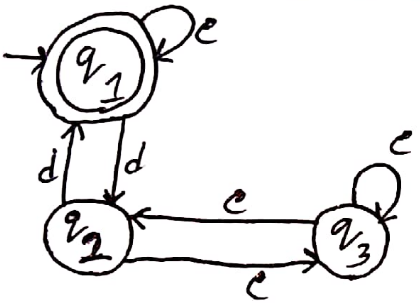


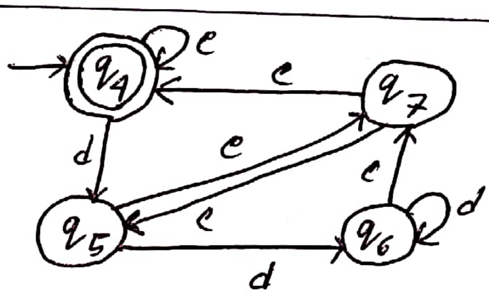
Automation-2

Sates	c	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ F	$\{q_2, q_5\}$ I
$\{q_2, q_5\}$	$\{q_3, q_6\}$ I	$\{q_1, q_4\}$ F
$\{q_2, q_6\}$	$\{q_2, q_7\}$ I	$\{q_3, q_6\}$ I
$\{q_2, q_7\}$	$\{q_3, q_6\}$ I	$\{q_1, q_4\}$ F

$\therefore$ Equivalent

Example-2





Satates	e	d
$\{q_1, q_4\}$	$\{q_1, q_4\}$ F	$\{q_2, q_5\}$ I
$\{q_2, q_5\}$	$\{q_3, q_7\}$ I	$\{q_1, q_6\}$ FS IS

$\therefore$ Not-Equivalent

(Nero Academy) (R.E)

⊛ Prove that $(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$ is equal to $0^*1(0+10^*1)^*$

$\Rightarrow$ L.H.S $=(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$

$=(1+00^*1)[\epsilon+(0+10^*1)^*(0+10^*1)] = (1+00^*1)(0+10^*1)^*$

$=(\epsilon\cdot 1+00^*1)(0+10^*1)^* = (\epsilon+00^*)1(0+10^*1)^*$

$= 0^*1(0+10^*1)^* =$ R.H.S

Lecture - 10 | Context Free Languages

(Example of CFG)

⊛ Consider the grammer $G = (N, T, P, S)$ where $N = \{S, A\}$, $T = \{a, b, c\}$, production rules in P are:

(1) S [?] aSc (2) S [?] aAc (3) A [?] b

A typical derivation in the grammer is:

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaaAccc \Rightarrow aaabccc$

The language generated is, $L(G) = \{a^n b c^n \mid n \geq 1\}$

(Example (2) of CFG)

⊛ Consider a grammer ~~g~~ to generate arithmetic expressions consisting of number and operator symbols $+, -, *, /, \uparrow$. The rules of the grammer can be written as:

(1) E [?] EAE (2) E [?] (E) (3) E [?] $-E$

(4) E [?] $0|1|\ldots|9$ (5) A [?] $+|-|*|/|\uparrow$

Give the derivation of the mathematical expression, $2*(3+5*9)$

(16 - 10 - 2023)

CFL 1* Verify whether the Grammer, $S \to 0B/1A$;

Ans: $S \to 0B$
$\to 00BB [B \to 0BB]$
$\to 001B [B \to 1]$
$\to 0011S [B \to 1S]$
$\to 00110B [S \to 0B]$
$\to 001101S [B \to 1S]$
$\to 0011010B [S \to 0B]$
$\to 00110101 [B \to 1]$

$*A \to 0 | 0S | 1AA | A$; $B \to 1/1S/0BB$ generates the string 00110101

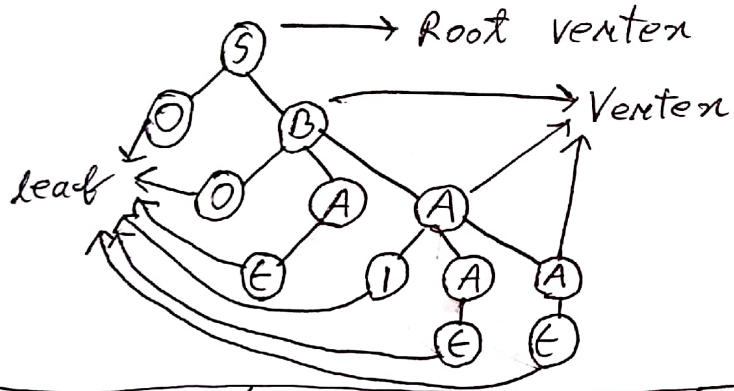
CFL 2* Verify whether the Grammer $S \to aAb$, $A \to aAb/\Lambda$ generates the string $\underbrace{aa}_{a^m}\underbrace{bbb}_{b^n}$
Null

Ans: $S \to aAb$
$\to aaAbb [A \to aAb]$
$\to aaAbbb [A \to aAb]$
$\to aaa\Lambda bbb [A \to \Lambda]$
$\to aaabbb$

×

∴ This grammer will never generate this string.

Derivation Tree / Parse Tree $S \to 0B$, $A \to 1AA/\epsilon$, $B \to 0AA$

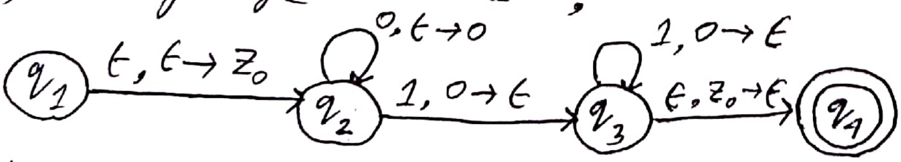


2 type of Derivation Tree by priority:
1) Left D.T
2) Right D.T

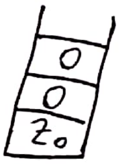
Lecture-12 Pushdown Automata (Components of PDA)

(Graphical notation of PDA)

(page-8) Language : $0^n 1^n$, $>= 0$

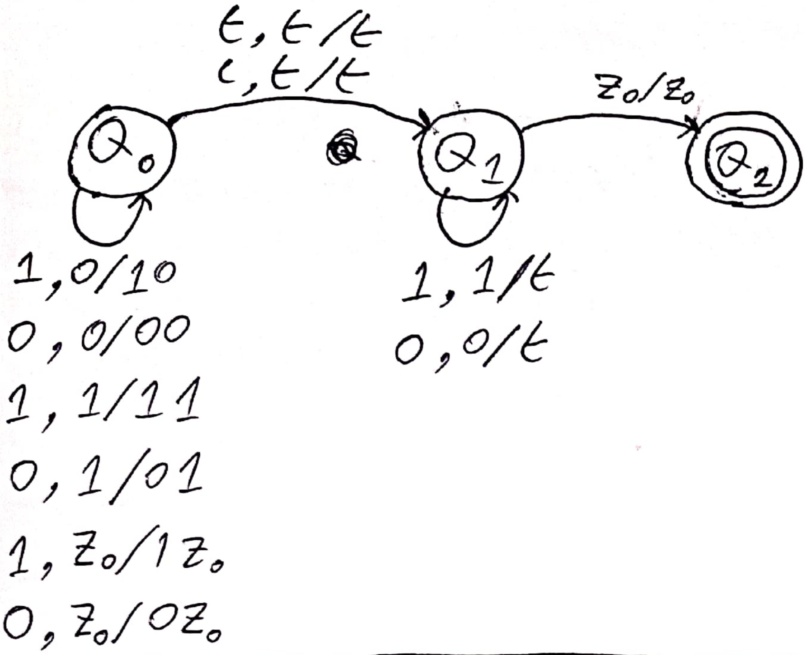


. 0011
. 011
. 001



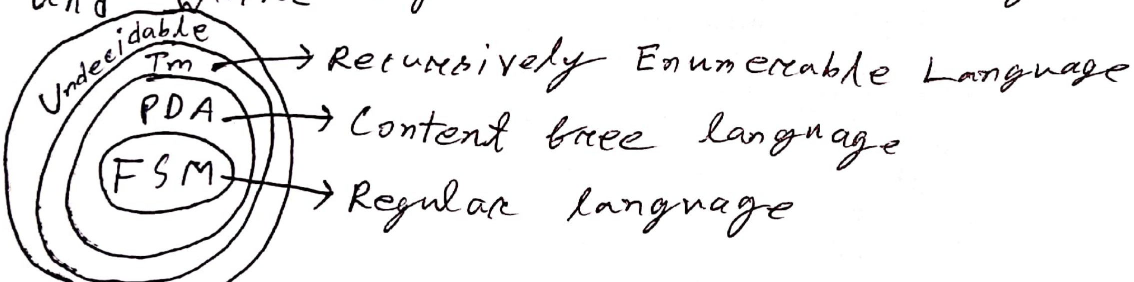
00111 → Reject

Design a palindrome that accepts a palindrome word.



Lecture 13 | Turing Machines

<u>Turing Machine</u>: The TM model uses an infinite tape as its memory. It has a tape head that can read and write symbols and move along the tape.



A T.M machine has <u>7-tuple</u> $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

1) Q is a finite set of state.
2) Σ is the input (finite) alphabet not containing the blank symbol $\sqcup$.
3) Γ is the tape (finite) alphabet where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$.
4) $\delta : Q \times \Gamma \square Q \times \Gamma \times \{L, R\}$ is the transition function.
5) $q_0 \in Q$ is the start state.
6) $q_{accept} \in Q$ is the accept state and
7) $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$

Configuration of T.M

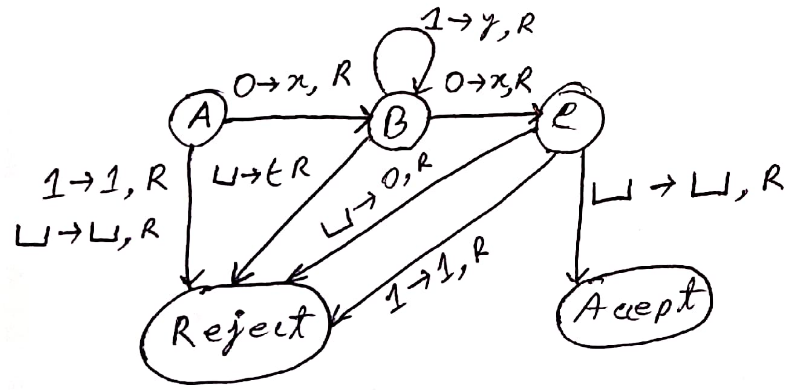
The current state, tape content and head location, a setting of these three items during the computation of a TM at any step is called configuration of that T.M.

- configuration are represented as triples $u\,q\,v$ where $q \in Q$ is the state, $u, v \in \Gamma^*$ form the tape content u, v and the head location is the first symbol of v. Everything beyond the last symbol of v is blank. In particular,
- the start configuration of a TM on input w is $q_0 w$ where q_0 is the start state.
- an accepting configuration is $u\, q_{accept}\, v$ for any $u, v \in \Gamma^*$
- a rejecting configuration is $u\, q_{reject}\, v$ for any $u, v \in \Gamma^*$
- a halting configuration is either accepting or rejecting configuration.

$L = 01^*0$

Read: R, Write: W, Direction: D

$0 \to x, R$



0110
x y y x