Lecture - 79

Final Topic

Context Free Grammar

Use for describing language.

⇒ A CFG is a set of rules / Production used to generate Pattern of strings.

CFG ⇒ 4 tuples $(V/N, T, P/R, S)$

V → Variables / non terminals set

T → Terminals [set of]

P/R → Production / Rules

S → Start Variables

Example: CFG for $\{0^n 1^n \mid n \geq 1\}$

Soln: $0^1 1^1 = \{01\}$

$0^2 1^2 = \{0011\}$

CFG $S \Rightarrow 01 \mid 0S1$ OR,

$S \Rightarrow 01$

$S \Rightarrow 0S1$ } Production

here,

$V = S$

$T = \{0, 1\}$

$S = S$

* **Example:** Identify the terminals, non terminals, start variable from the following grammar —

① $E \Rightarrow E+T \mid T$

$T \Rightarrow T*F \mid F$

$F \Rightarrow (E) \mid id$

② $S \Rightarrow (L) \mid a$

$L \Rightarrow L,S \mid S$

Soln:

① $V = \{E, T, F\}$

$T = \{+, *, (,), id\}$

$S = \{E\}$

② $V = \{S, L\}$

$T = \{'('; 'a', ','\}$

$S = \{S\}$

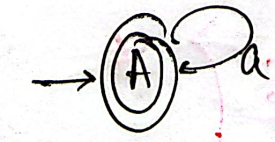
Lecture - 15

Example:

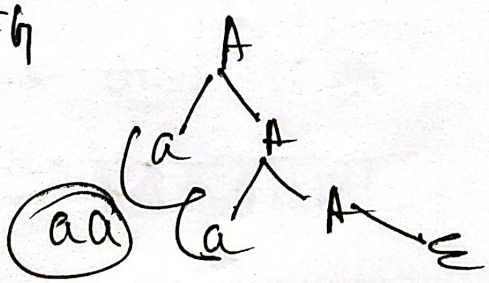
$L = \{a^n \mid n \geq 0\} \mid a^*$

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

$A \Rightarrow aA \mid \varepsilon$



CFG



$S \to 01 \mid 0S1$

0011

$\to 0S1$

$= 00\underline{11}$

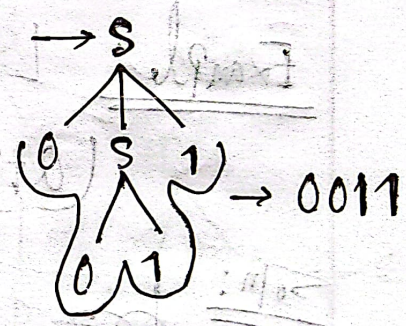
000111

$= 0\underline{S}1$

$= 00\underline{S}11$

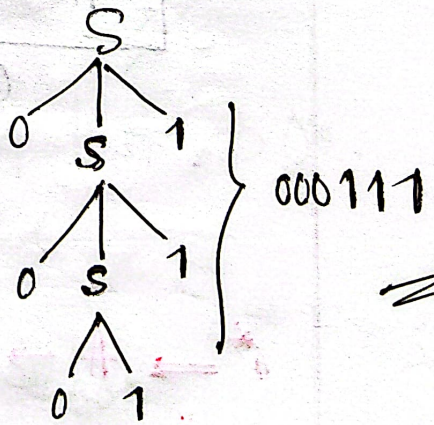
$= 000\underline{111}$

0011



$\to 0011$

000111



000111

Example: $L = \{ a^n \mid a \geq 1 \}$

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

$$\boxed{A = aA \mid a}$$

CFG

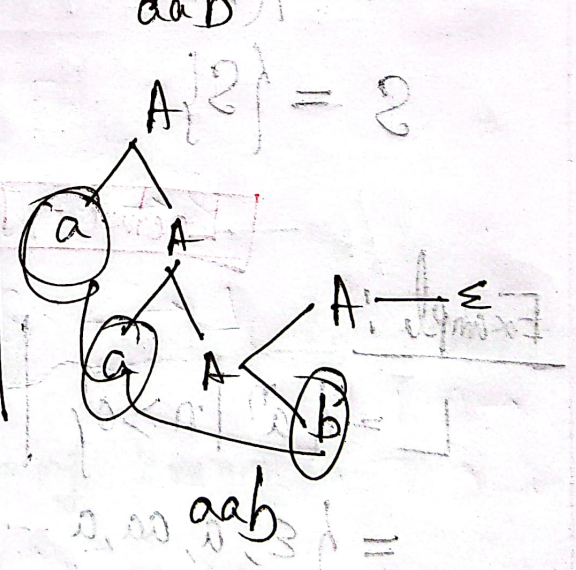
Example: $L = \{$ set of all string over $a, b\}$

$(a+b)^* / (a|b)^*$

Soln:

$$\boxed{A \to aA \mid bA \mid \varepsilon}$$

CFG



$a^* \Rightarrow A \to aA \mid \varepsilon$

$(A+B)^* \Rightarrow A \to aA \mid bA \mid \varepsilon$ *

Example: CFG for {set of all str which length atleast 2}

Soln: $L = \{aa, bb, aab \text{ ---} \}$

$$RE = \underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}\underbrace{(a+b)^*}_{B}$$

$B \Rightarrow aB \mid bB \mid \varepsilon$

$A \Rightarrow a \mid b$

$S \Rightarrow AAB$

abb

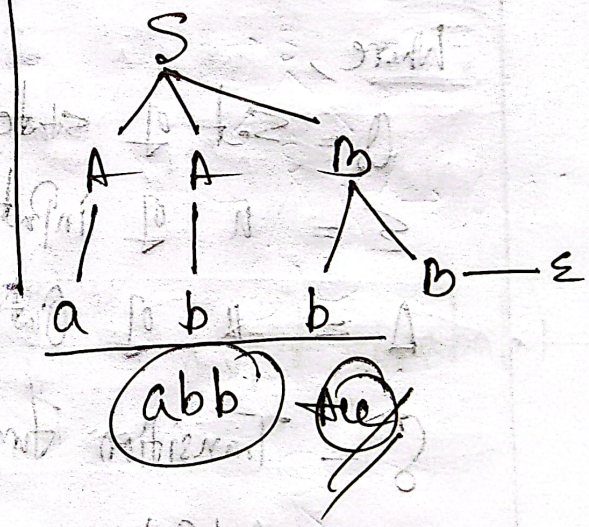


abb ~~ε~~

Example: CFG for set of all str of at least 3 0's.

$$RE = \underbrace{\Sigma 0 \Sigma 0 \Sigma 0 \Sigma}_{\downarrow}$$

$(0+1)^*$

$E \rightarrow 0E \mid 1E \mid \varepsilon$

$S \rightarrow E0E0E0E$

1000

$S = E0E0E0E$

$= 1E0E0E0E$

$= 10E0E0E$

$= 100E0E$

$= 1000E = 1000$

⇒ Finite Automata with Outputs (2)

- Mealy
- Moore

Mealy Machine

$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

Where,

Q = Set of states

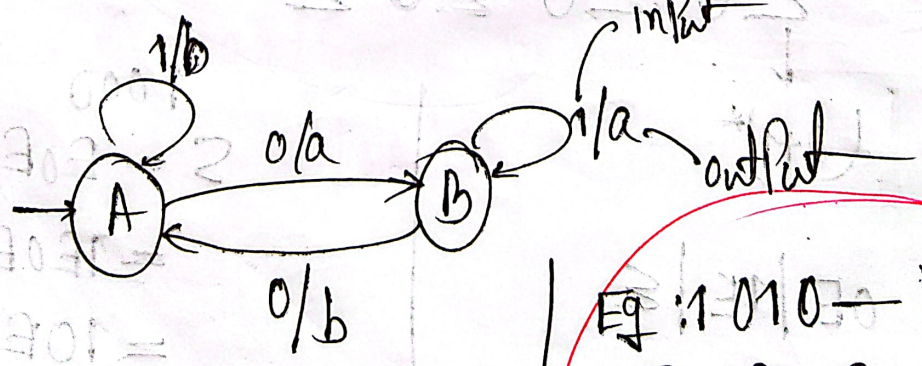
Σ = u of input alphabet

Δ = u of Output u

δ = Transition function = $Q \times \Sigma \to Q$

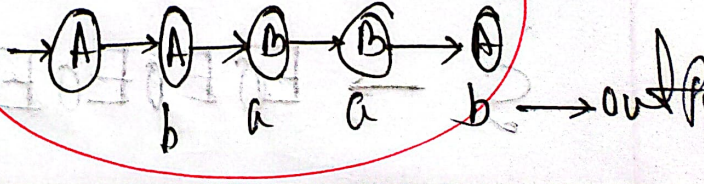
λ = Output u = $Q \times \Sigma \to \Delta$

q_0 = Initial state



input number = Output number

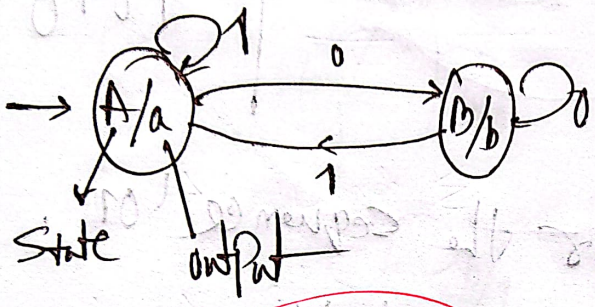
Eg: 1 0 1 0 — input



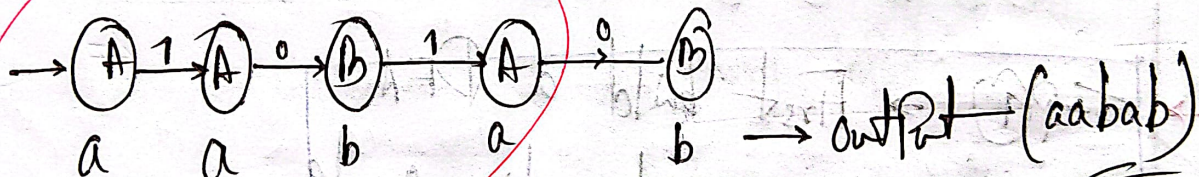
Moore Machine: $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

λ = output function = $Q \rightarrow \Delta$

→ All other same as mealy.

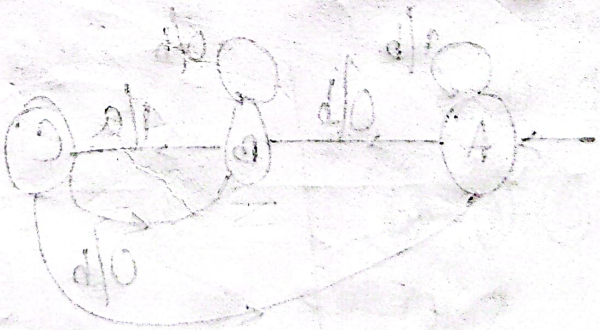


Eg: 1 0 1 0 → input



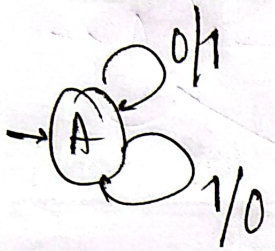
→ output (aabab)

Output = input number + 1



L-28 **Construction of mealy machine**

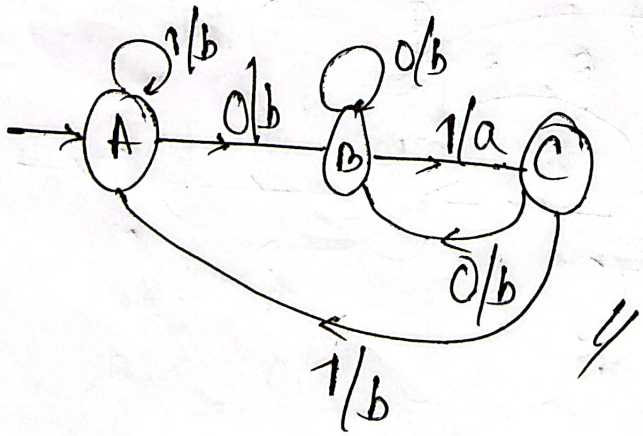
Ex: Construct a mealy machine,
Produce 1's Complement of any binary input.



0 1 0 1
↓ ↓ ↓ ↓
1 0 1 0

Ex: Print 'a' whenever the sequence '01' input, $\Sigma = \{0,1\}$ and Output, $\Delta = \{a,b\}$

Step (1) → First build a DFA
then DFA → Mealy machine Create



0 1 1 0 → in
b a b b → out

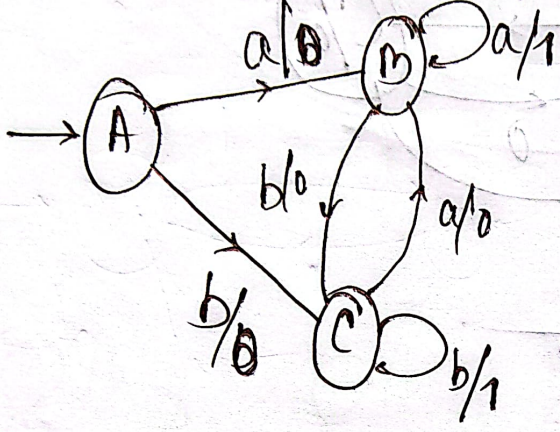
1 0 0 0 0 → in
b b b b b → out

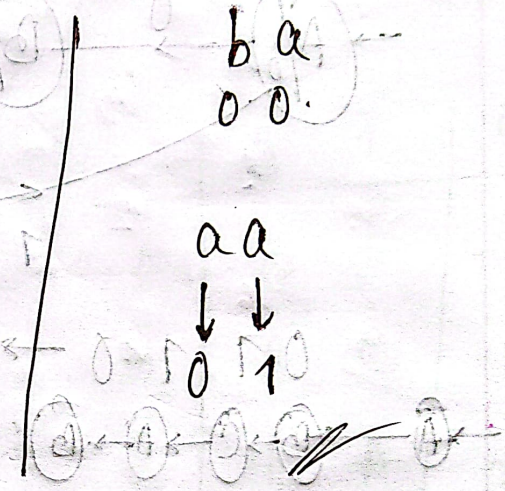
L-29

Ex: $\Sigma = \{a, b\}$ the str sould end with either 'aa' or 'bb'.

aa → 1

bb → 1





L-30

Ex: gives 2's Complement of any binary input.

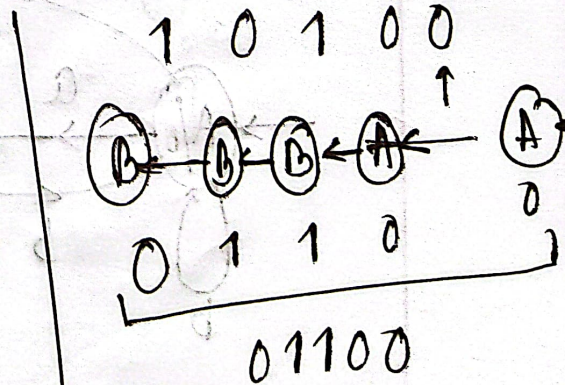
[2's Compliment = 1's Compliment + 1

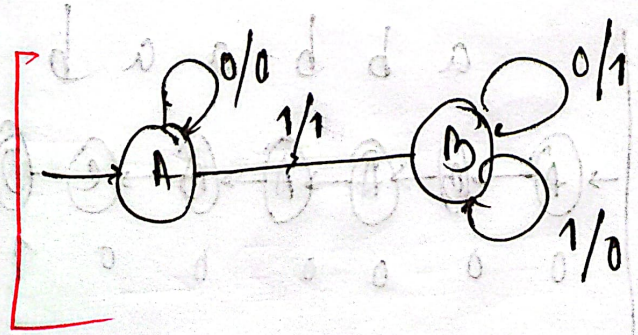
eg. 10100

1's 01011

+1

2's → 01100]





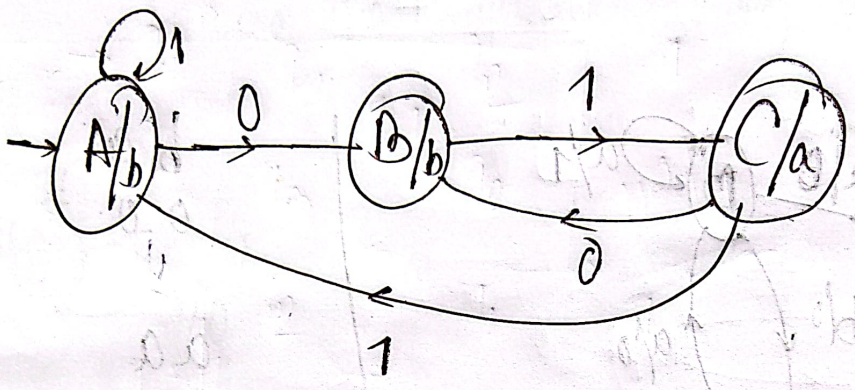
Construction of Moore Machine

L-31

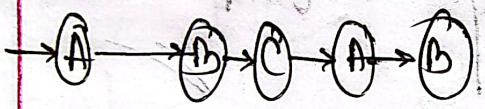
Ex: Print 'a' whenever the seq '01'.

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

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



0 1 1 0 → Input

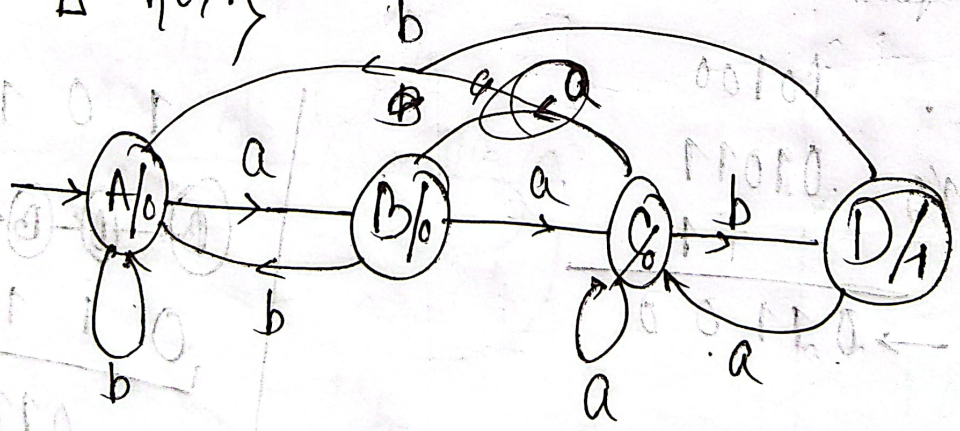


b b a b b → output

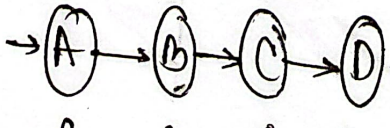
L-32

Ex: $\{a,b\}$ Count the occurance of the sequence 'aab'.

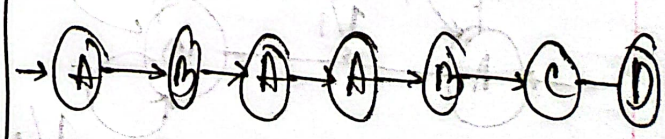
$\Delta = \{0,1\}$



a a b



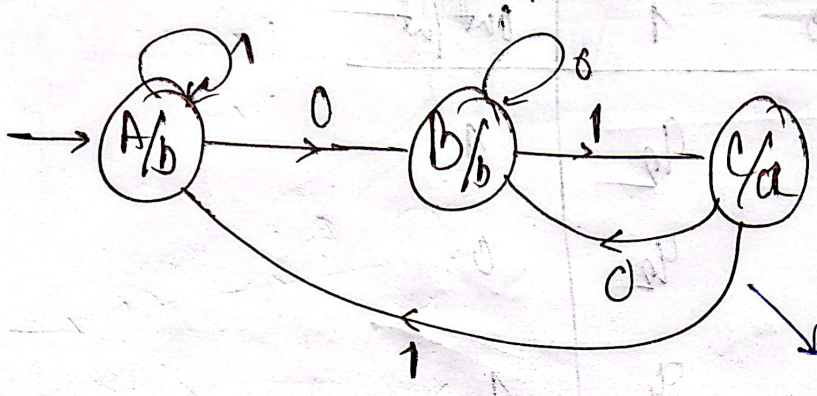
a b b a a b



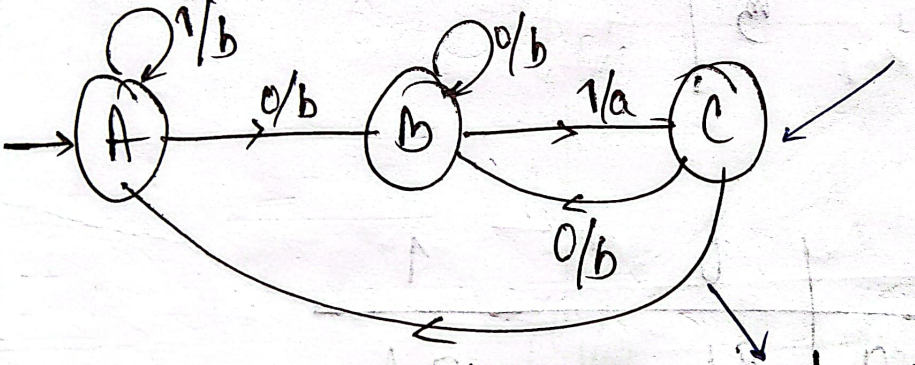
Moore → Mealy

L-39

Ex: Print 'a' whenever the sequence '01'.



	0	1	Out
→ A	B	A	b
B	B	C	b
C	B	A	a



	0	1
→ A	B/b	A/b
B	B/b	C/a
C	B/b	A/b

L-36

Moore :

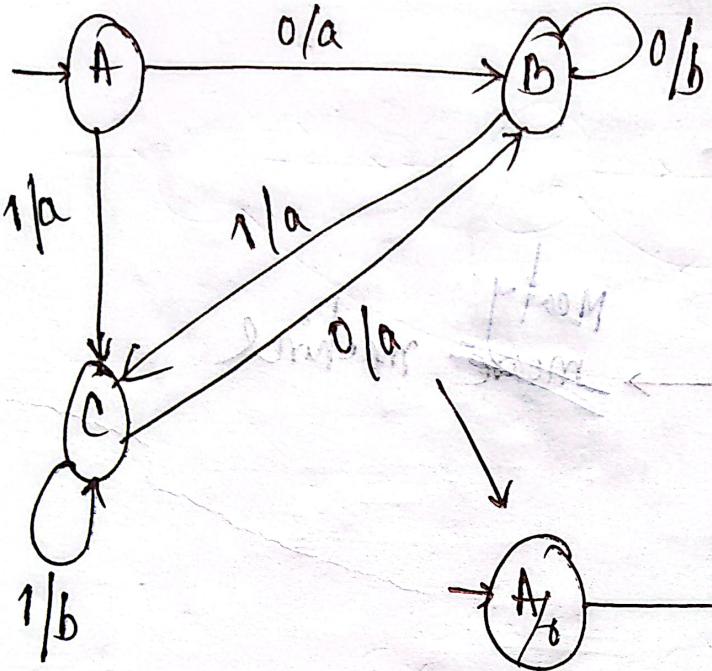
	0	1	output
→ q_0	q_1	q_2	1
q_1	q_3	q_2	0
q_2	q_2	q_4	1
q_3	q_0	q_3	1

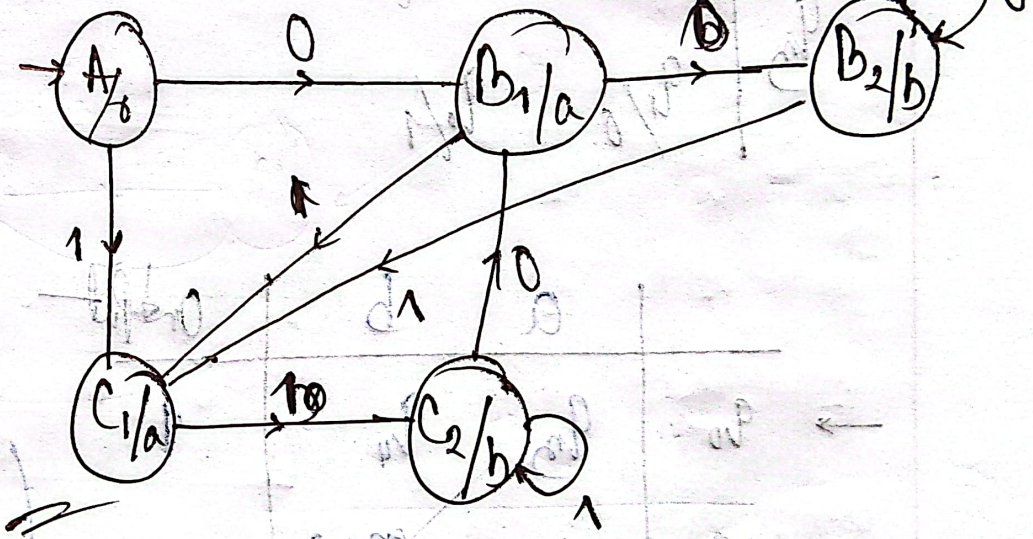
Mealy :

	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	$q_0/0$	$q_3/1$

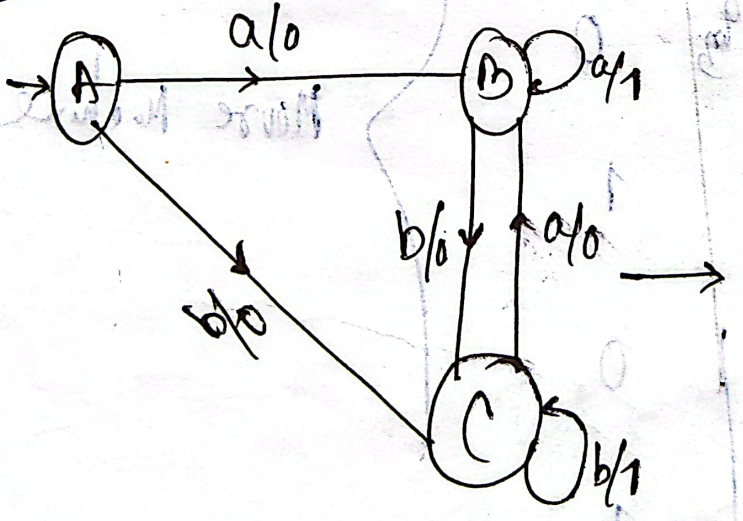
L-37

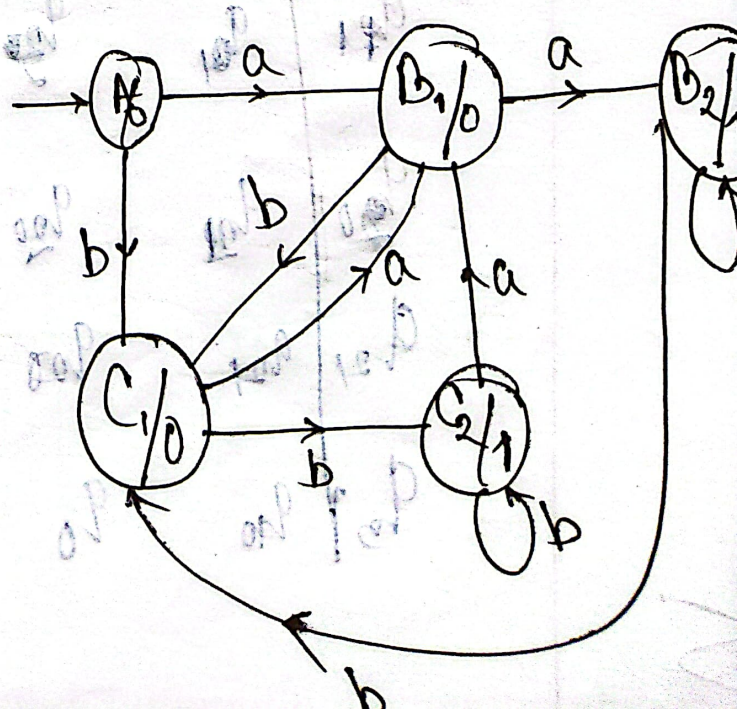
Mealy To Moore





L-38





L-40

Using Transition Table:

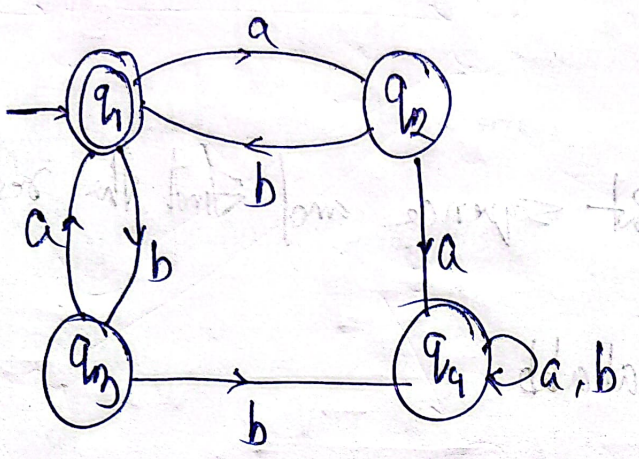
	a	b
→ q_0	$q_3/0$	$q_1/1$
q_1	$q_0/1$	$q_3/0$
q_2	$q_2/1$	$q_2/0$
q_3	$q_1/0$	$q_0/1$

→ Mealy machine

	a	b	output
→ q_0	q_3	q_4	1
~~q_{10}~~	~~q_{01}~~	~~q_3~~	0
q_{21}	q_4	q_{31}	1
q_{20}	q_{11}	q_{20}	0
q_{21}	q_{21}	q_{20}	1
q_3	q_{10}	q_0	0

→ Moore Machine

DFA To Regular Exp



$$q_1 = \varepsilon + q_2 b + q_3 a \quad \text{—— (i)}$$

$$q_2 = q_4 a \quad \text{—— (ii)}$$

$$q_3 = q_1 b \quad \text{—— (iii)}$$

$$q_4 = q_2 a + q_3 b + q_4 a + q_4 b \quad \text{—— (iv)}$$

eq^n (i) $\Rightarrow q_1 = \varepsilon + q_2 b + q_3 a$

$= \varepsilon + q_1 ab + q_1 ba$

$\Rightarrow \underset{R}{q_1} = \underset{Q}{\varepsilon} + \underset{R}{q_1} \underset{P}{(ab+ba)}$

$\therefore q_1 = (ab+ba)^*$

$\longrightarrow R = Q + RP$

$\Rightarrow R = QP^*$

$\because R = R$

$\Rightarrow q_4 = (ab+ba)^*$ (Ans)

L-33

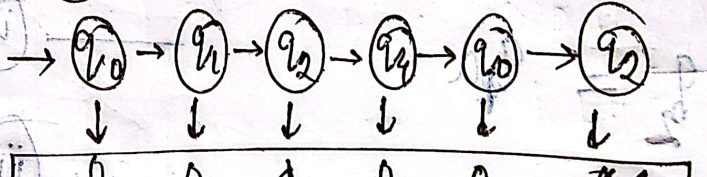
Eg: Run the following input sequence and find the respective output,

(i) aabab (ii) abbb (iii) ababb

	a	b	output
→ 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

Answer:

(i) a a b a b



output

Final Topic

16/10/22

Derivation Tree / Parse Tree [Left / Right]

Regular language

Context Free Grammer

Ambiguous Grammer

L-78 Chomsky Normal Form

In CNF, we have restriction in the length of RHS, which is elements in RHS should either be two Variable or a Terminal. A CFG is in CNF if the Productions are in the following form:

A → a → Terminal | Single

A → BC → 2 Variable

A, B, C → non terminal

L-79

Steps: (CFG To CNF)

1. If S appears in RHS, Add new state S' and $S' \rightarrow S$ is added to the Production.
2. Remove all Null Production [ϵ]
3. Remove all Unit u [$A \rightarrow B$] | Variable → Variable
4. Remove more than Two Variable in RHS [$A \rightarrow ASA$]
5. Remove Together terminal and Variable symbol, [$A \rightarrow aB$] in RHS

Eg: Convert the following CFG to CNF:

P: $S \rightarrow ASA|aB, A \rightarrow B|S, B \rightarrow b|\epsilon$

Step (1):

P: $S' \rightarrow S, S \rightarrow ASA|aB, A \rightarrow B|S, B \rightarrow b|\epsilon$

step ②: $B \to \epsilon$ and $A \to \epsilon$

$B \to \epsilon$ P: $S' \to S$, $S \to AsA|aB|a$, $A \to B|s|\epsilon$, $B \to b$

$A \to \epsilon$ P: $S' \to S$, $S \to AsA|aB|a|As|sA|s$, $A \to B|s$, $B \to b$

step ③: $S \to S$, $S' \to S$, $A \to B$ and $A \to S$

$S \to S$ P: $S' \to S$, $S \to AsA|aB|a|As|sA$, $A \to B|s$, $B \to b$

$S' \to S$ P: $S' \to AsA|aB|a|As|sA$, ~~$A \to B|s$, $B \to b$~~

$S \to AsA|aB|a|As|sA$

$A \to B|s$, $B \to b$

$A \to B$ P: $S' \to AsA|aB|a|As|sA$,

$S \to AsA|aB|a|As|sA$,

$A \to b|s$, $B \to b$

$A \to S$ P: $S' \to AsA|aB|a|As|sA$,

$S \to AsA|aB|a|As|sA$

$A \to b|AsA|aB|a|As|sA$,

$B \to b$

Step ④: $S' \to ASA$, $S \to ASA$ and $A \to ASA$

After removing all these,

P: $S' \to AX \mid aB \mid a \mid AS \mid SA$ | [All [SA] replace with 'X']

$S \to AX \mid aB \mid a \mid AS \mid SA$

$A \to b \mid AX \mid aB \mid a \mid AS \mid SA$

$B \to b$, $X \to SA$

Step ⑤: $S' \to aB$, $S \to aB$ and $A \to aB$

We get,

P: $S' \to AX \mid YB \mid a \mid AS \mid SA$

$S \to AX \mid YB \mid a \mid AS \mid SA$

$A \to b \mid AX \mid YB \mid a \mid AS \mid SA$

$B \to b$, $X \to SA$

$a \to Y$ [All 'a' replaced with Y]

Which is the required CNF for the given CFG.

CFG Lecture - 76

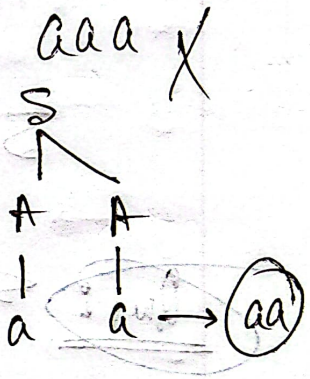
Question: Length at most 2?

Answer: $L = \{\epsilon, a, b, ab, aa, bb, ba\}$

$$RE = \underbrace{(a+b+\epsilon)}_{A}\ \underbrace{(a+b+\epsilon)}_{A}$$

$A \rightarrow a \mid b \mid \epsilon$

$S \rightarrow AA$



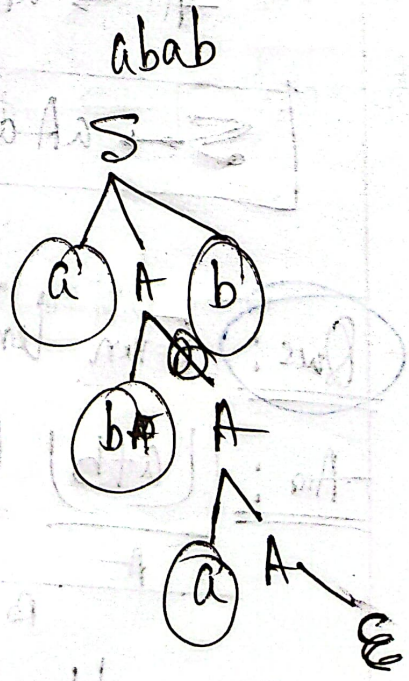
Question: Start with a and end with b.

Answer: $L = \{ab, aaab, abbb \dots\}$

$$RE = a\underbrace{(a+b)^*}_{A}b$$

$A = aA \mid bB \mid \epsilon$

$S = aAb$



Ques: Start and End with different symbol.

Ans: $a(a+b)^*b \mid b(a+b)^*a$

$RE = \underbrace{a(a+b)^*}_{A}b + b\underbrace{(a+b)^*}_{A}a$

$A \rightarrow aA \mid bA \mid \varepsilon$

$S \rightarrow aAb \mid bAa$

Ques: Start and End with same symbol.

Ans: $a\underbrace{(a+b)^*}_{A}a \mid b\underbrace{(a+b)^*}_{A}b \mid a \mid b \mid \varepsilon$

$A \rightarrow aA \mid bA \mid \varepsilon$

$\boxed{S \rightarrow aAa \mid bAb \mid a \mid b \mid \varepsilon}$ CFG

Ques: Even length string.

Ans: $\underbrace{\underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}}_{B}^*$

$A \rightarrow a \mid b$ $S \rightarrow AA$ $\boxed{B \rightarrow BS \mid \varepsilon}$

Push Down Automata (PDA)

A Push down Automata is a way to implement a CFG in a similar way we design DFA for a RG.

→ A DFA can remember a limited amount of information but

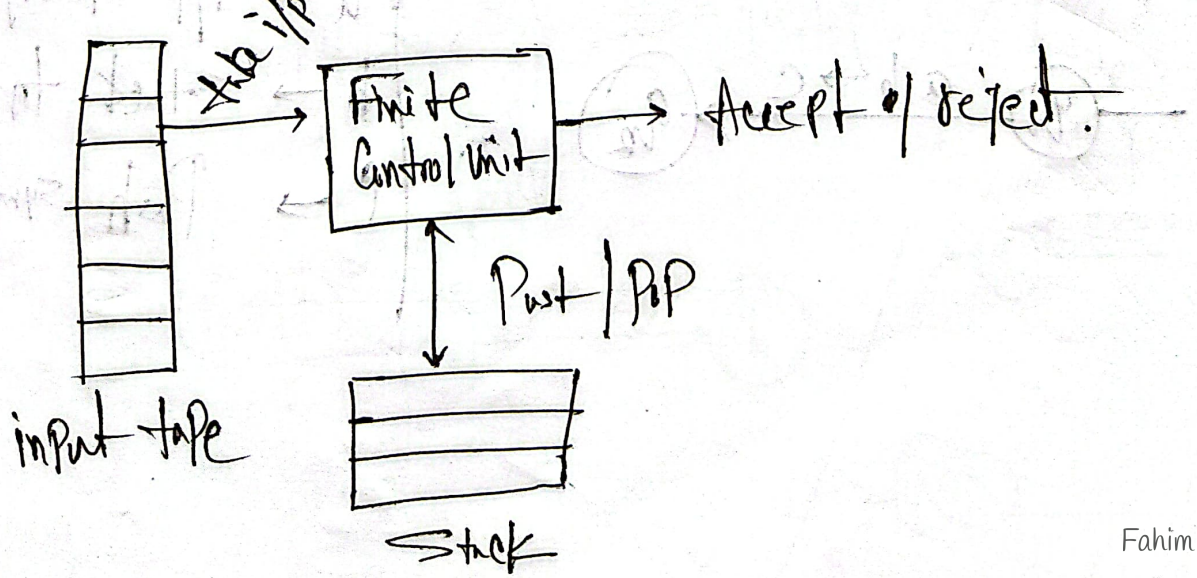
→ A PDA can remember an infinite amount of information.

Basically a PDA is — "Finite state machine + a stack."

→ PDA has three components —

(i) a input tape
(ii) a Control Unit
(iii) a stack with infinite size

⇒ A PDA may or may not read i/p symbol, but it has to read the top of the stack in every transaction.



⊛ A PDA Formal Definition: ✱

7 tuples $(Q, \Sigma, S, \delta, q_0, I, F)$

$Q \rightarrow$ finite no. of state

$\Sigma \rightarrow$ input alphabet

$S \rightarrow$ stack symbol

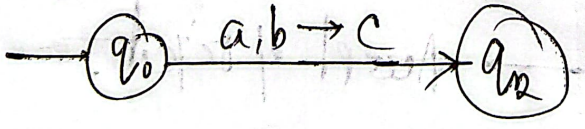
$\delta \rightarrow$ the transaction function $[Q \times (\Sigma \cup \{\in\}) \times S \times Q \times S^*]$

$q_0 \rightarrow$ Initial state $(q_0 \in Q)$

$I \rightarrow$ Initial state top symbol $(I \in S)$

$F \rightarrow$ Set of accepting states $(F \in Q)$

Transaction Diagram:



a → i/p symbol
b → stack top symbol
c → Push symbol

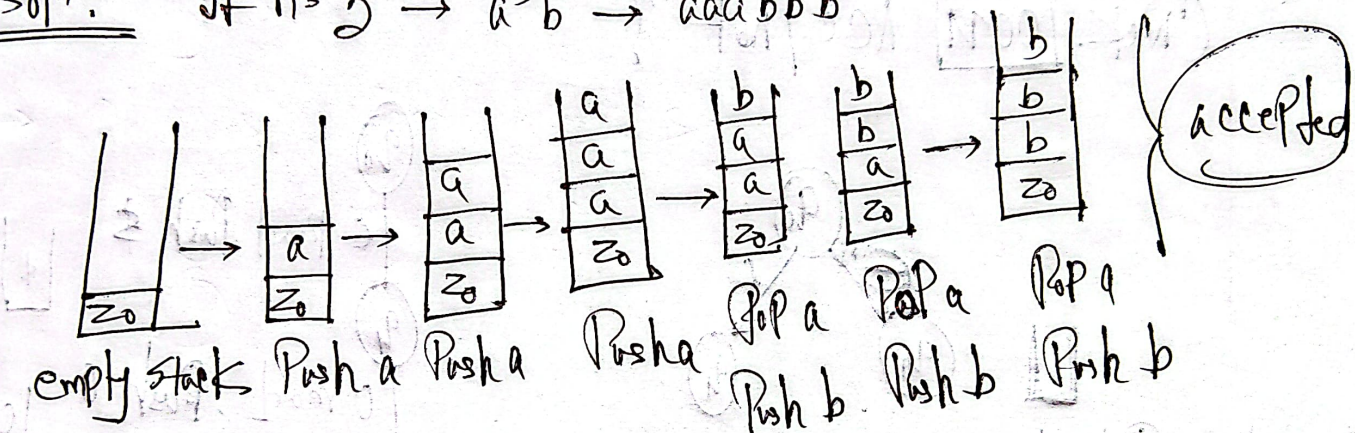
Acceptability : 2 ways

(i) Final state Acceptability

(ii) Empty stack "

Example : Construct PDA that acc $L = \{a^n b^n \mid n \geq 1\}$

Soln : If $n = 3 \rightarrow a^3 b^3 \rightarrow aaabbb$

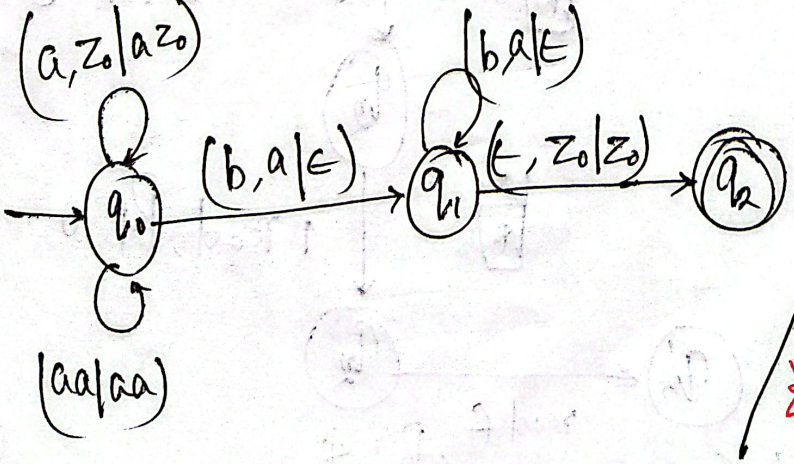


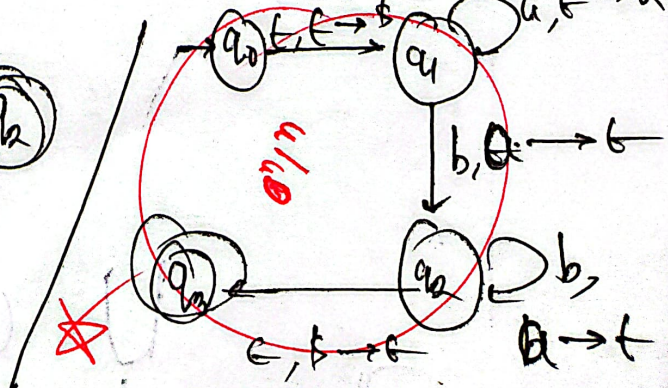
Transition Diagram:

$M = \{Q, \Sigma, S, \delta, q_0, I, F\}$ $\quad (n+\$) = \text{state}$

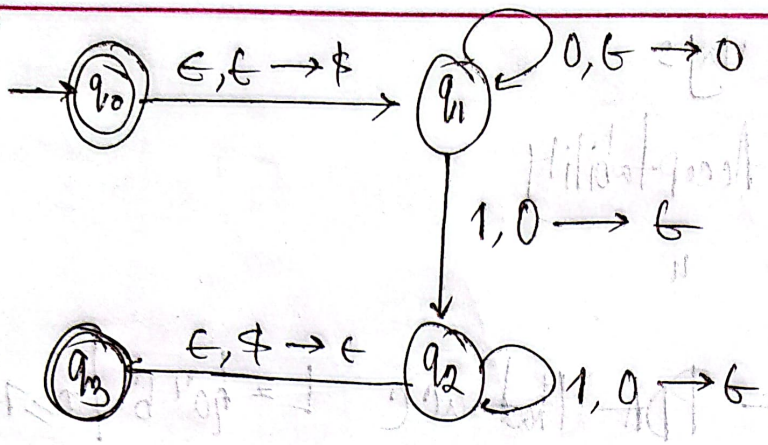
Here, $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$, $S = \{a, Z_0\}$

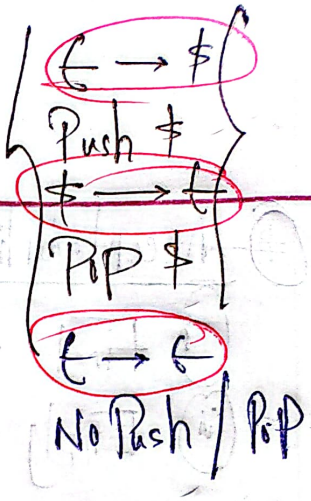
$I = \{q_0\}$



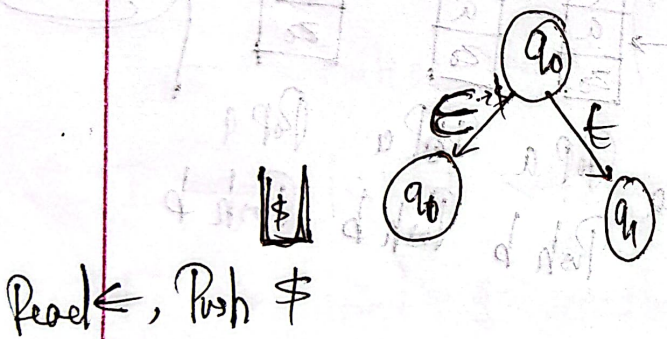


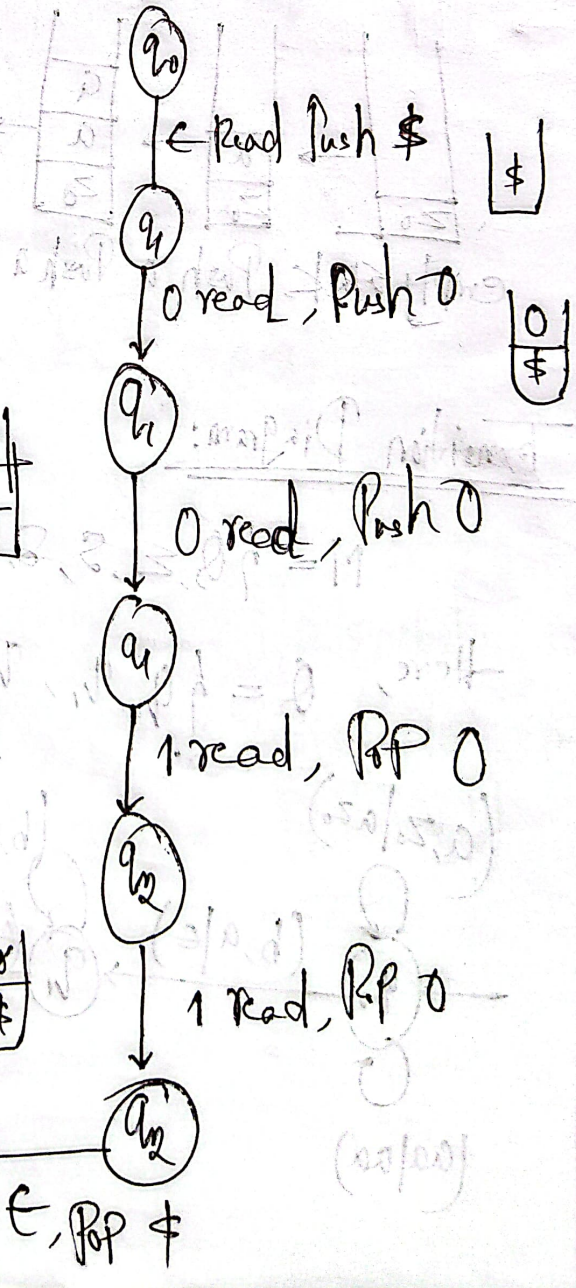
Example:





Check 0011 Acc/not ?





Acc = empty stack