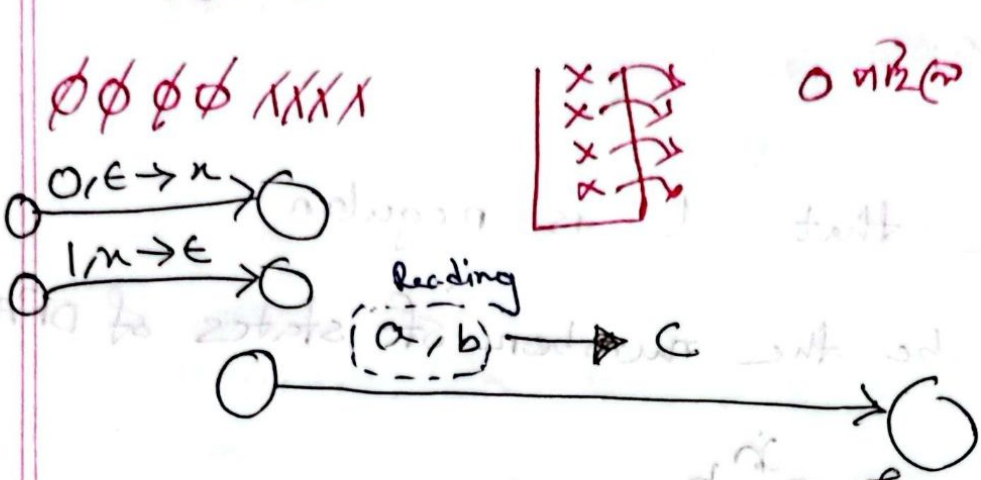
PDA Push Down Automata

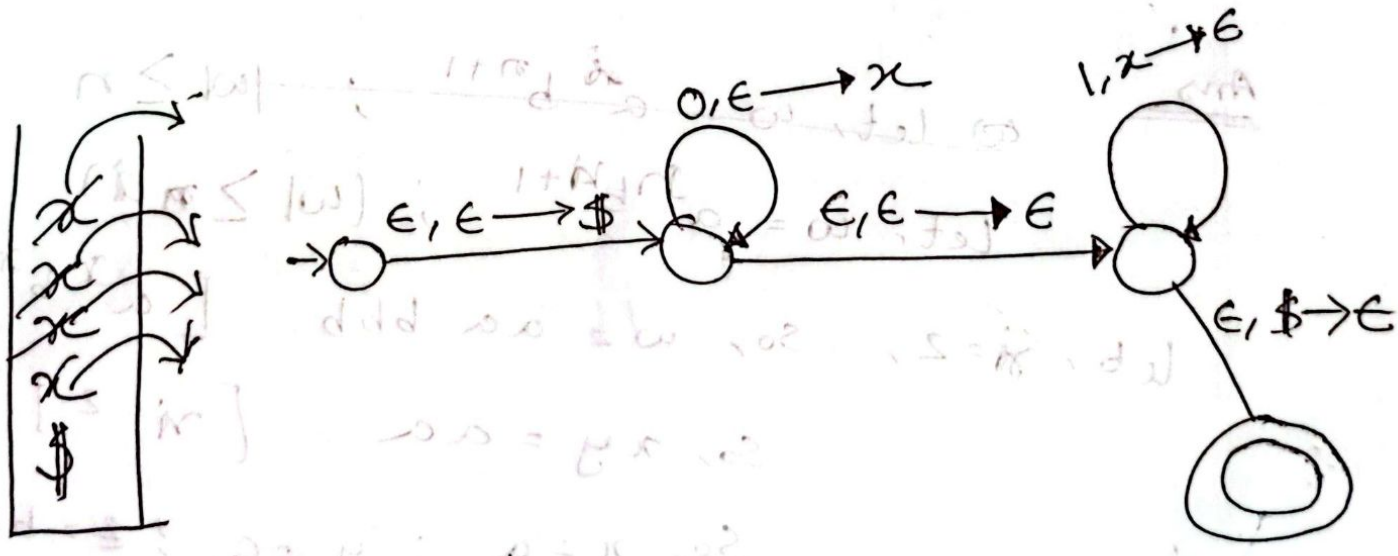


a = input symbol b = pop from stack

c = push into stack

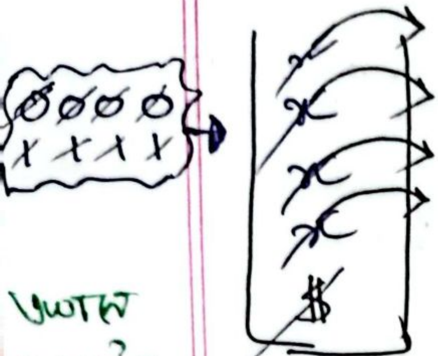
Ex For inputting 0,

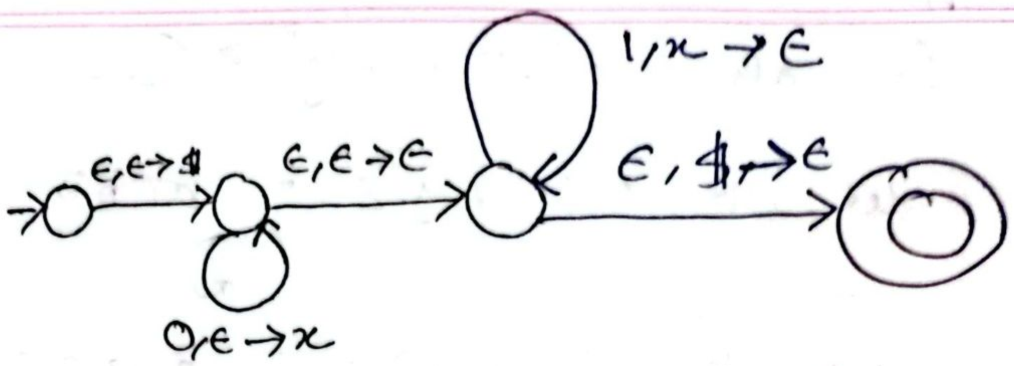
For removing 1,



PDA

$0^n 1^n$





প্রথমে আমাদের

1 নম্বর (i) 1st keeping a $ in stack →

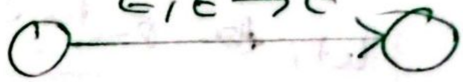
input = ε, pop = ε, push = $

Acceptance state এ যেতে পারবো, cause, remove করার জন্য x আছে.

(ii) 0 পাইলে একটা করে x রাখব stack এ.

0, ε → x ; 0-input, ε-কিছু pop করছি না, x- pus

(iii) Non-deterministically guess.



it means যে 0 পাওয়া শেষ

(iv) 1 পাইলে x pop করব.

input, pop → push

1, x → ε ; 1-input, x pop করছি, ε - push করব মানে কিছুনা

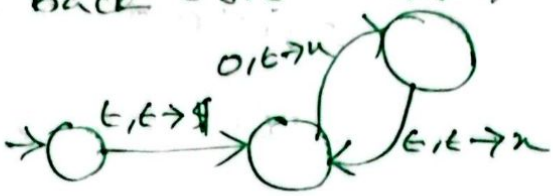
(v) Accepting state

; input nothing, popping $, getting empty stack

011 , 00 1111, 000 111111

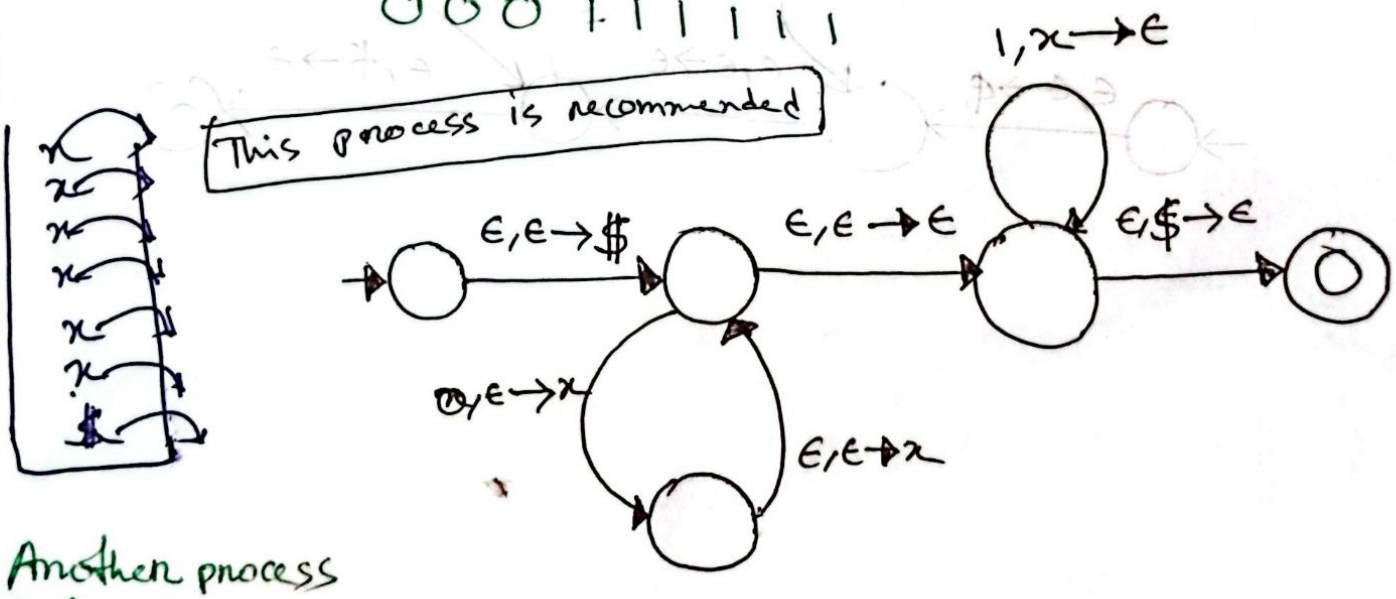
$\#0^n1^{2n}$ এখানে stack এ আগে 0 পাইলে 2টা x রাখতে হবে, and 1 পাইলে 1টা করে x কাটতে হবে.

তাহলে stack এ 2টা করে x রাখা লাগতে পারে, এটার জন্য final state বানাব, যাতে 0 থেকে x রাখতে এবং back করার সময় কিছু না দেখেই x রাখতে.

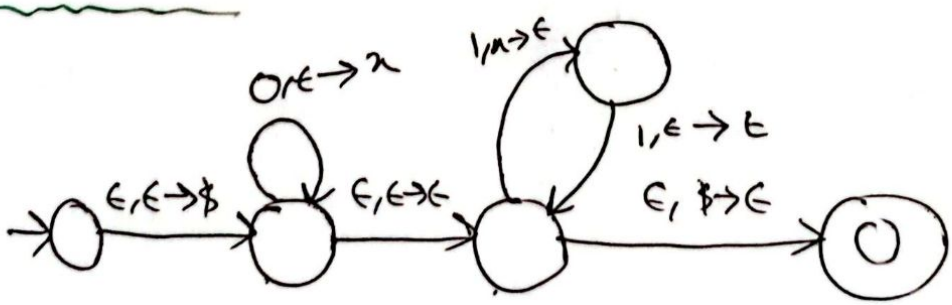


And 1 পাইলে আগে যতটা করে x ফেরত করব.

This process is recommended



Another process



$0^{3n}1^{2n}$

00011

000000 1111

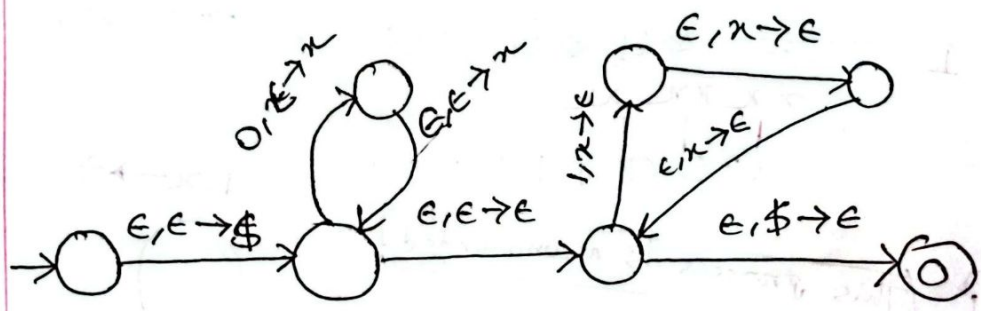
3n and 2n এর LCm = 6

So, 3 টা 0 এর জন্য 6 টা করে x থাকবে

per 1. 3 টা x আছে

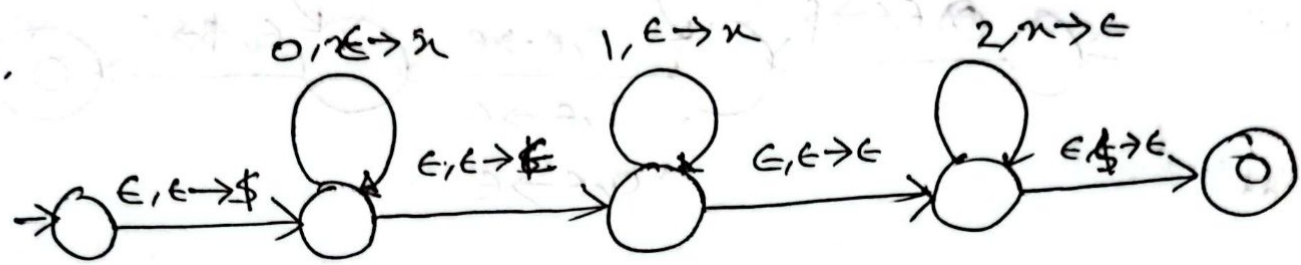
per 0 এর জন্য 2 টা করে x থাকবে

per 1 ——— 3 টা করে x আছে



$\Sigma = \{0,1,2\}$ $0^i 1^j 2^k$; $i+j=k$

এই ভাষায়, আমরা 0, 1 এর জন্য x push and 2 এর জন্য pop.



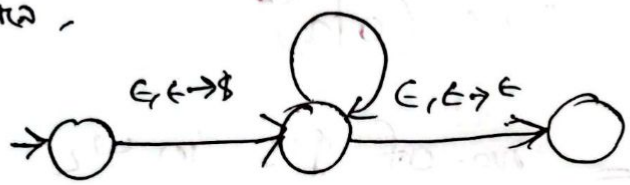
$0^i 1^j 2^k$; $j = k+i$

$j = k+i$

$\therefore k = j-i$

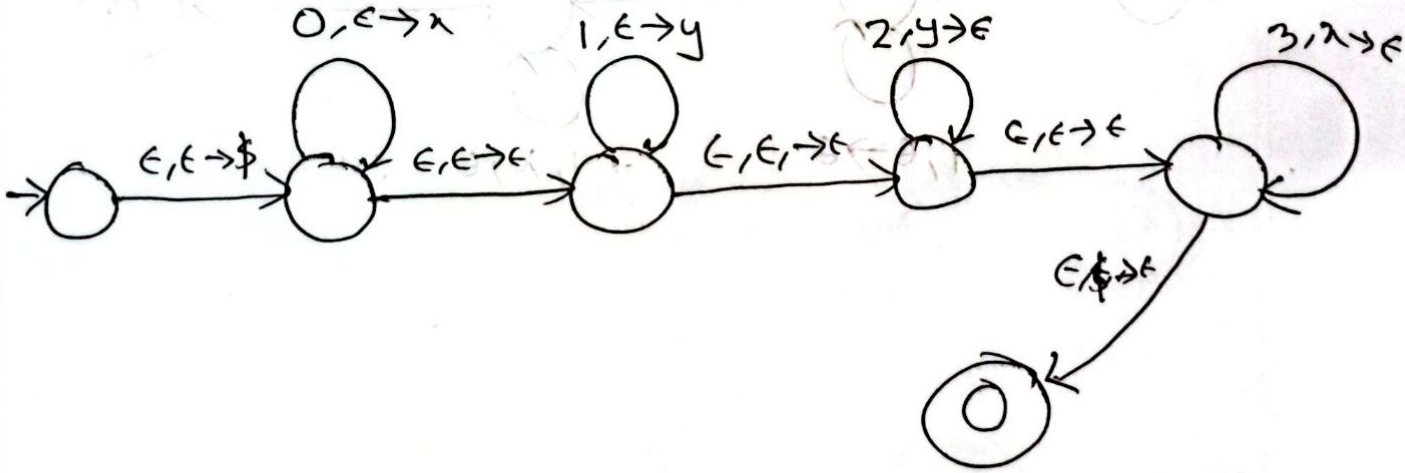
0, ϵ → x

আমরা,



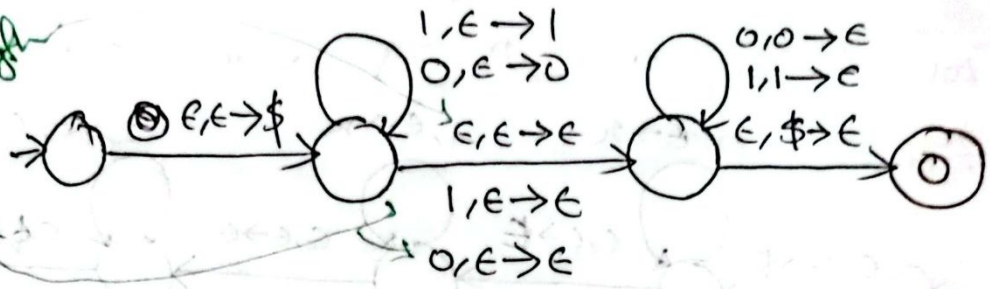
$0^i 1^j 2^j 3^i$

$\Gamma = \{\$, x, y\}$



Palindrome $\Gamma = \{\$, 0, 1\}$ 1110 0 0111

Even length

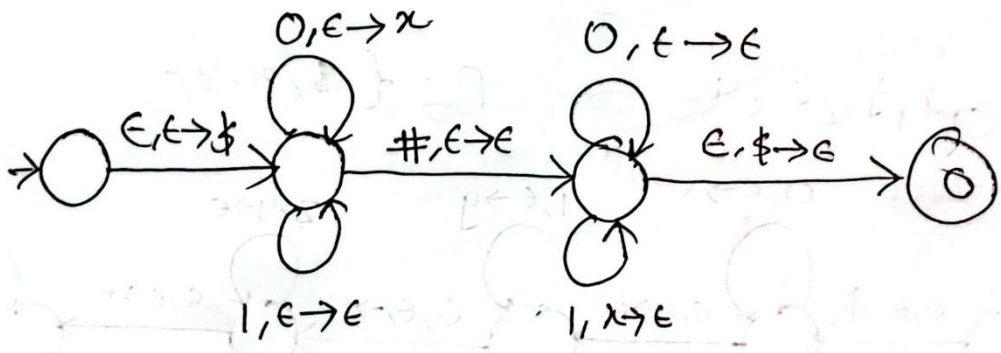


Odd length

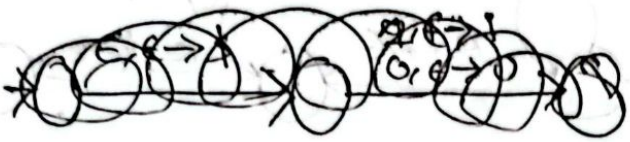
$\omega_1 \# \omega_2$; $\omega_1 \in (0|1)^*$ $\Sigma = \{0, 1, \#\}$

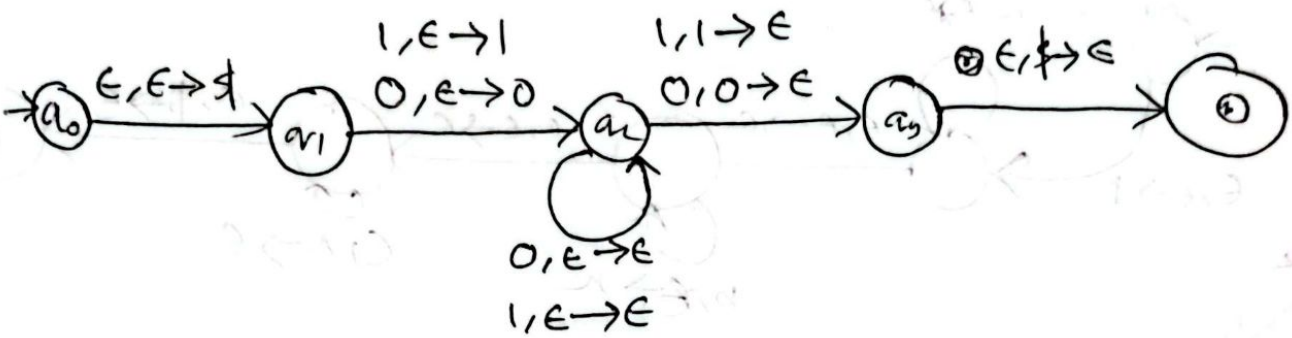
$\omega_2 \in (0|1)^*$

no. of 0's in ω_1 = no. of 1's in ω_2

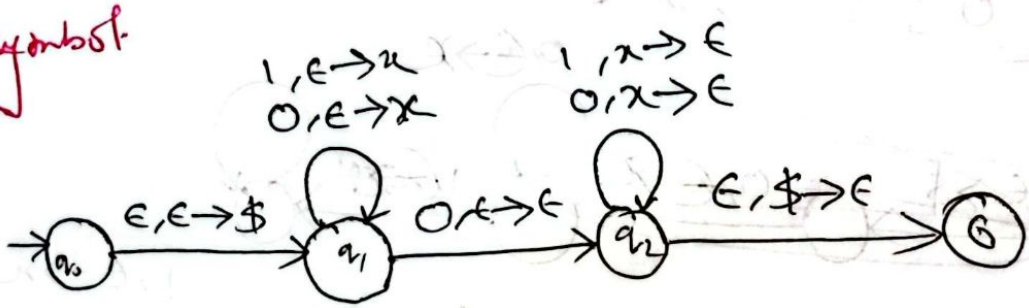


$\{ w \in \{0|1\}^* \mid w$ starts and ends with the same symbol$\}$

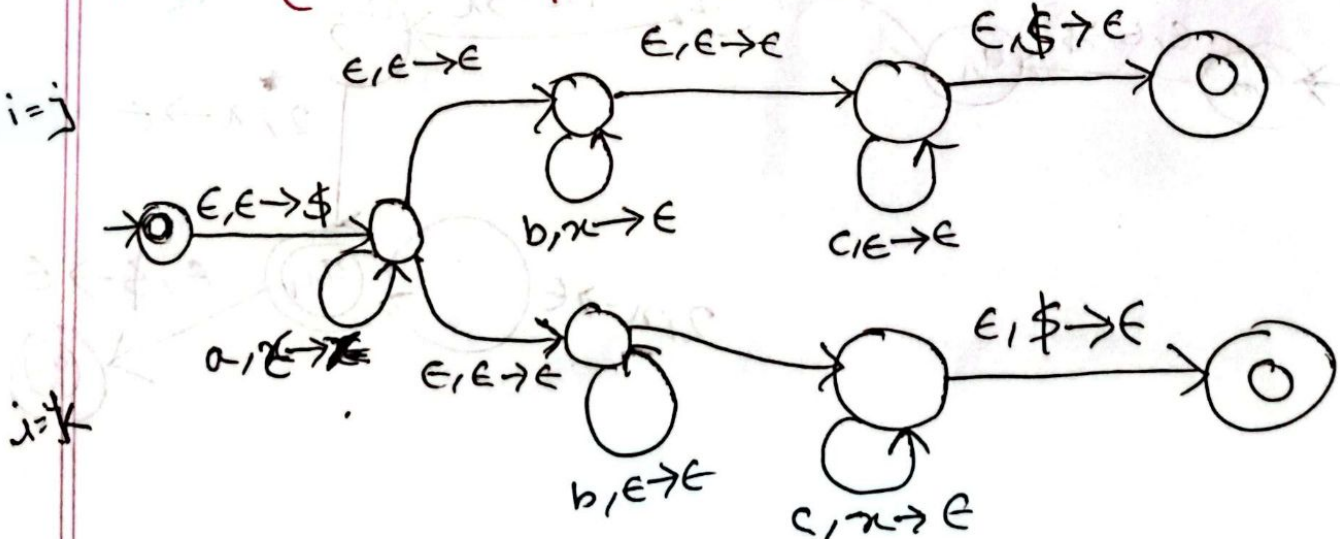




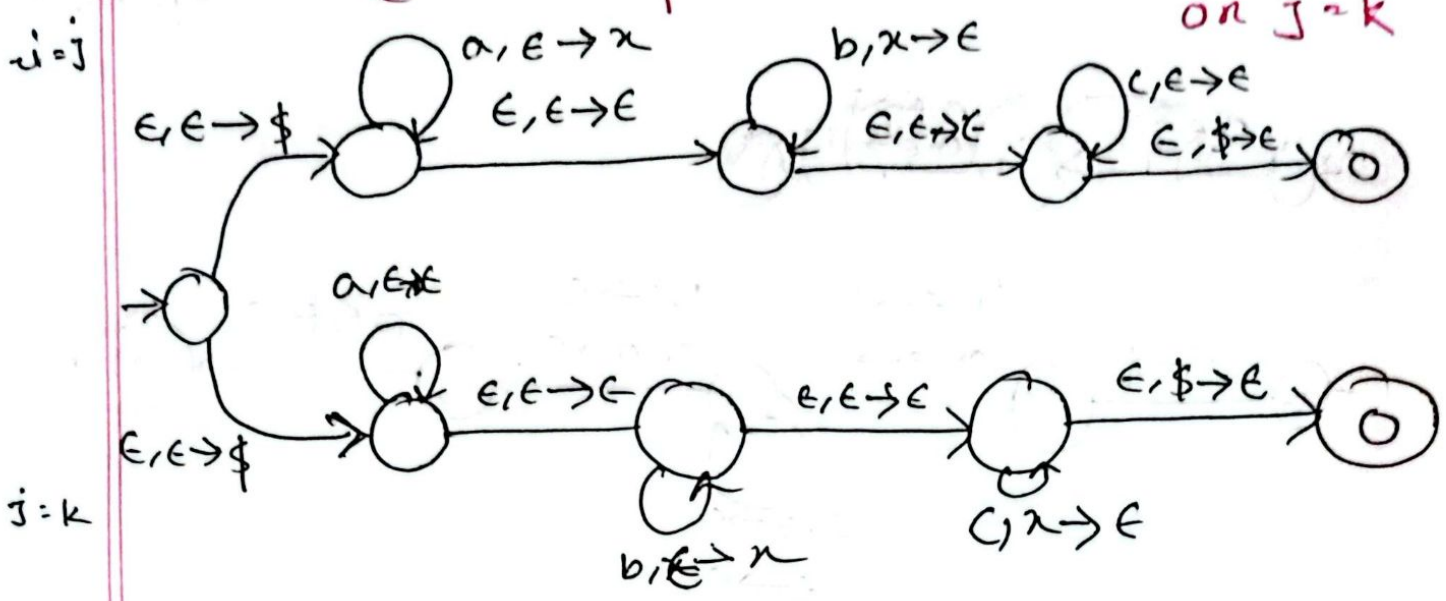
w is of odd length and 0 as its middle symbol



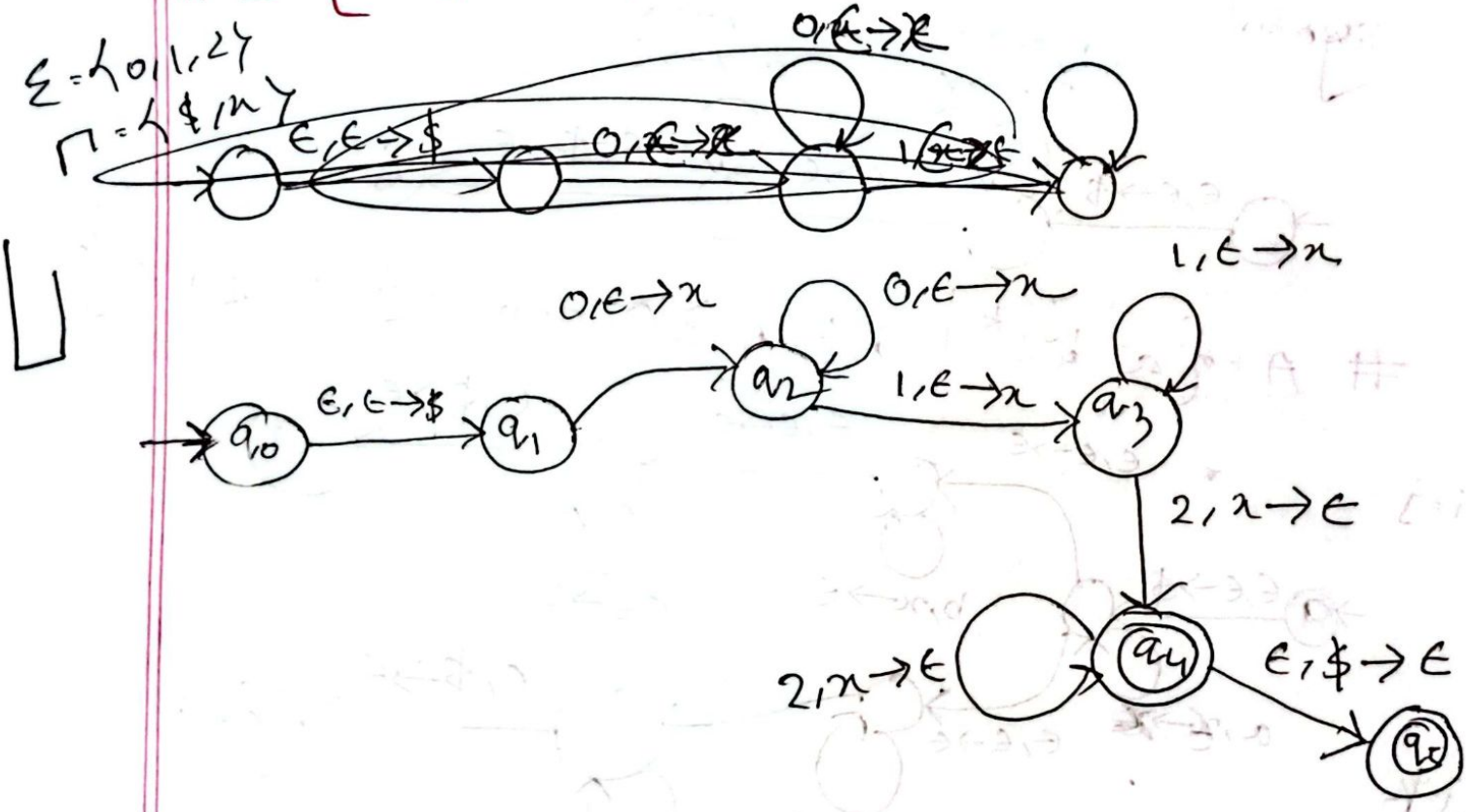
$A = \{ a^i b^j c^k \mid i, j, k \geq 0$ and $i = j$ or $i = k \}$



A = $\{a^i b^j c^k \mid i,j,k \geq 0$ and either $i=j$ or $j=k\}$



L = $\{0^i 1^j 2^k \; ; i+j >= k$ and $i,j,k \geq 0\}$

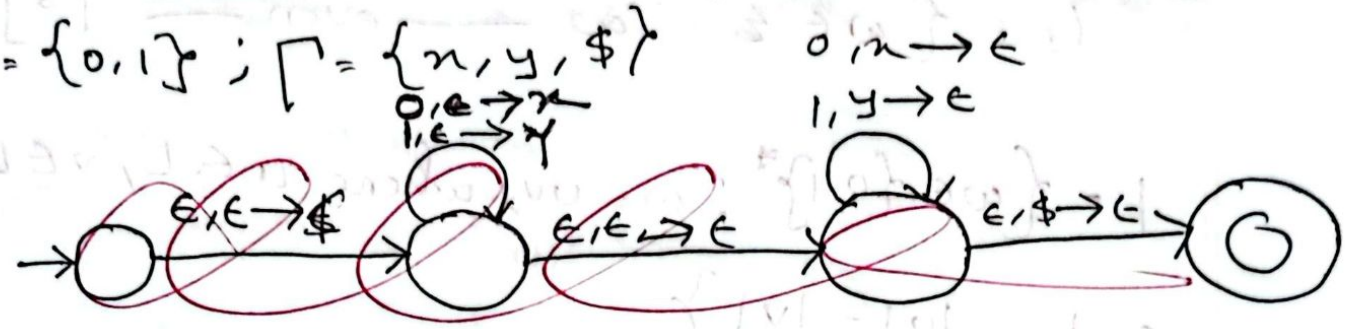


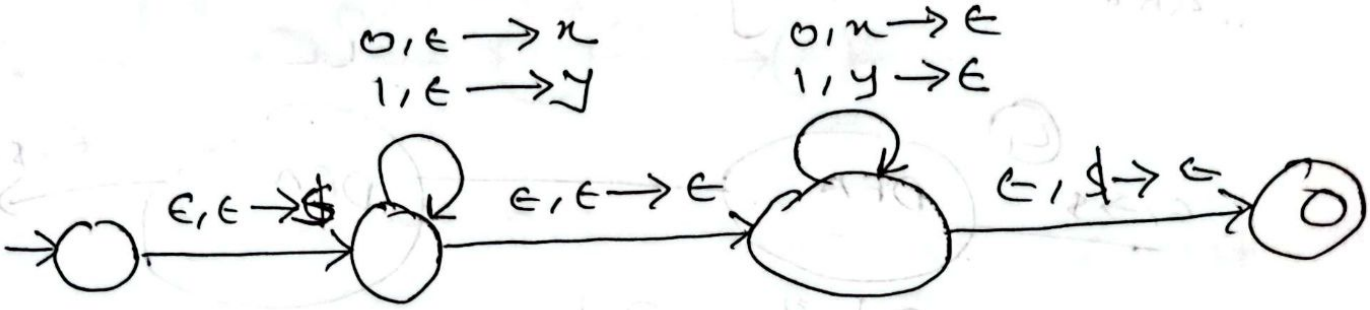
$\underbrace{0011}_{w}\underbrace{1100}_{w^R}$

$L = \{ww^R : w \in \{0,1\}^*\}$

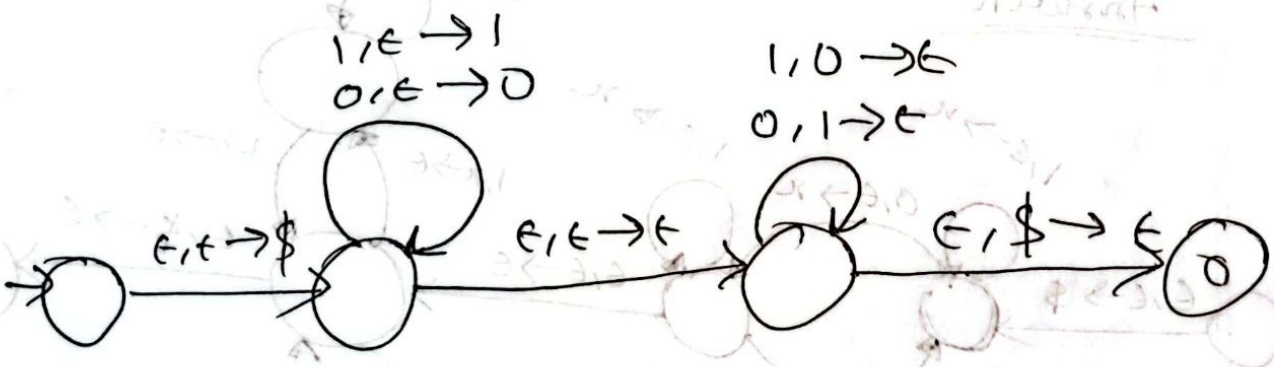
w^R means the reverse of the string.

$\Sigma = \{0,1\}$; $\Gamma = \{x, y, \$\}$





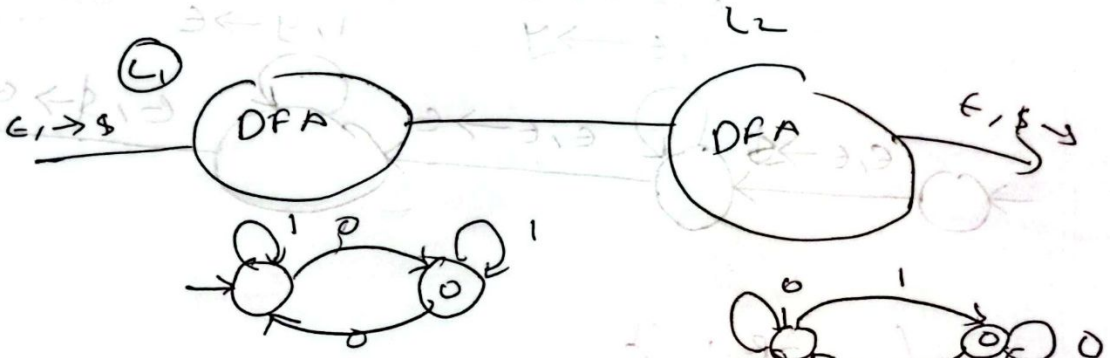
$L = \{w w^R :\}$



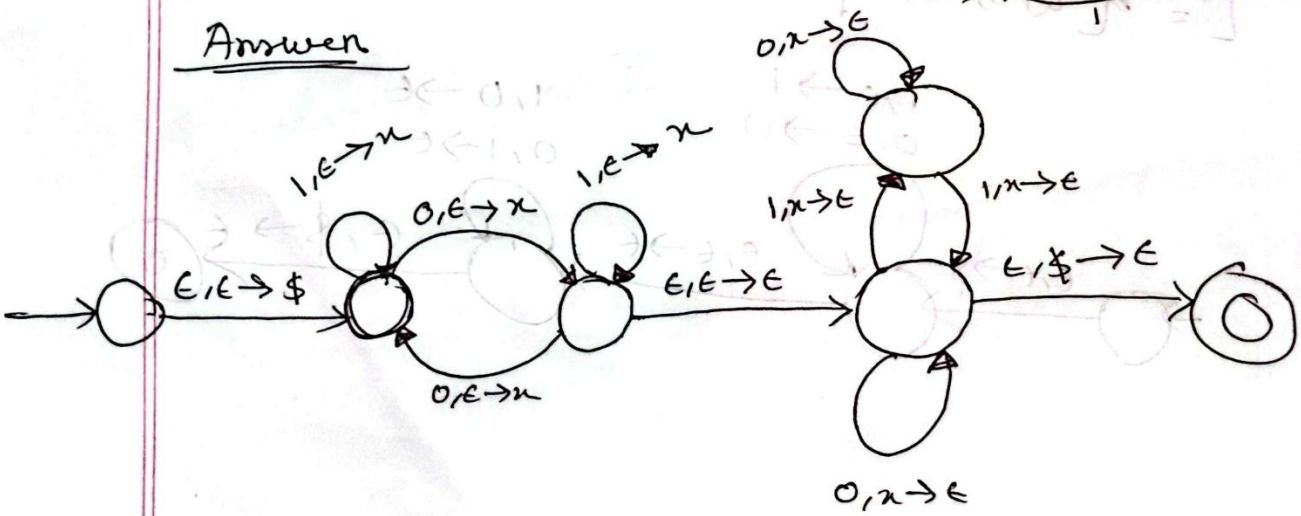
$L_1 = \{ w \in \Sigma^* : w$ contains odd num. of 0's $\}$

$L_2 = \{ w \in \Sigma^* : w$ —— even —— 1's $\}$

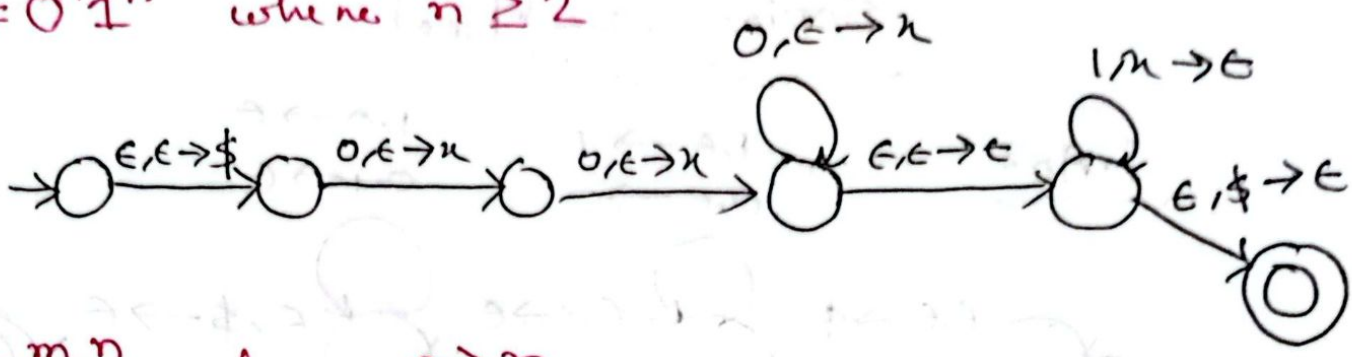
$L = \{ w \in \{0,1\}^* : w = uv$; where $u \in L_1, v \in L_2$ and $|u| = |v| \}$



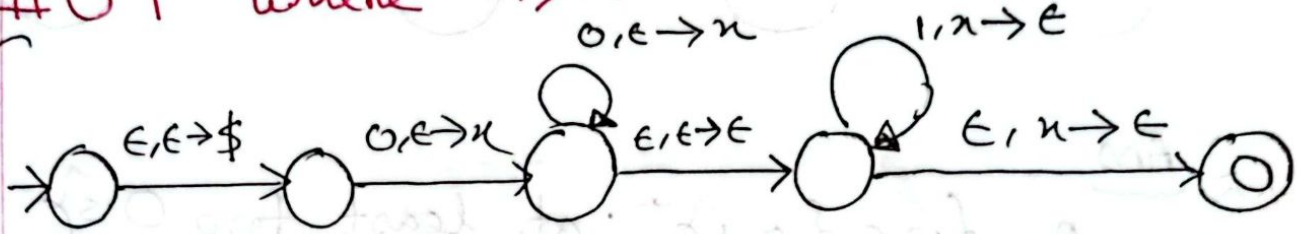
Answer



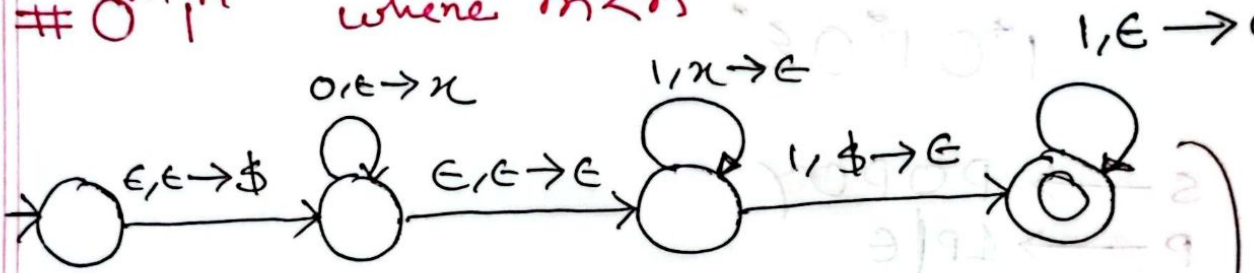
$0^n 1^n$ where $n \geq 2$



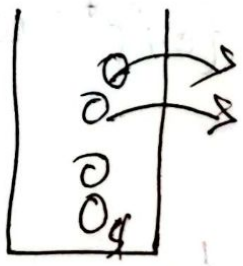
$0^m 1^n$ where $m > n$



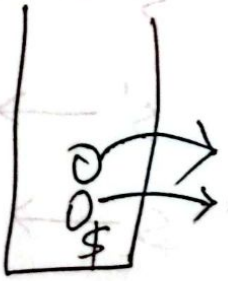
$0^m 1^n$ where $m < n$



00011



00111



$L_2 = \{ \omega_1 \# \omega_2 \}$ Num. of "00" substrings in ω_1 is equal to the Num. of "11" substring in ω_2.

Ex-1 → 00101010 0# 01001000 11