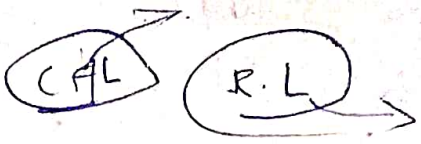
Module-IV

Turing machine



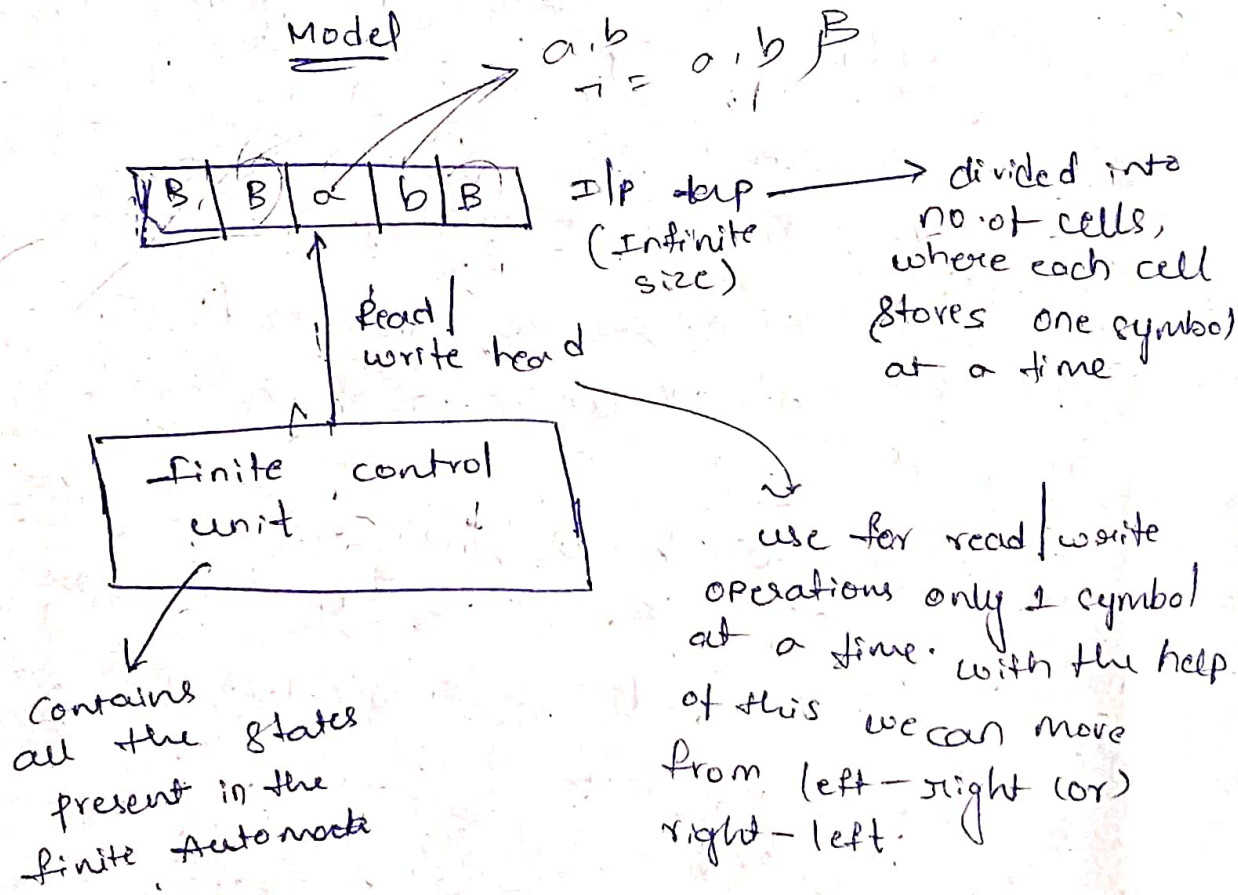
Turing machine accepts or recognizes ~~recursily~~ Recursively Enumerable languages.

1. T.M → R.E.L

F.A → R.L
P.D.A → C.F.L

W.K.T, FA only accepts limited amount of languages, hence we moved to P.D.A to accept infinite languages, but P.D.A only accepts simple languages so, now we moved to T.M to accept complex languages.

Model



Formal Definition:-

7-tuples

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Σ : I/p tape alphabet
- B : Blank symbol (Part of I/p tape)

Ex: from above model

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

$\Gamma = \{a, b, B\}$

$\Sigma \subseteq \Gamma$

δ is transition fu?

$$Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$$

R → right

L → moving in left-direction.

Ex-

$\delta(q_0, a) \longrightarrow (q_2, X, R)$ — modify a to X, read/right move to right

$\delta(q_0, b) \longrightarrow (q_1, Y, L)$ — b to Y, left

Design Turing machine for $L = \{a^n b^n \mid n \geq 1\}$

$\Sigma = \{ab, aabb - - -\}$

logic:- a a b b

- If we need 'a' → change a to X
- Now move head towards right until we find 'b'. If we need 'b' replace with Y.

$\not{b}$ a $\not{b}$ b

X Y

- Now if we need 'a', move left & replace a with X

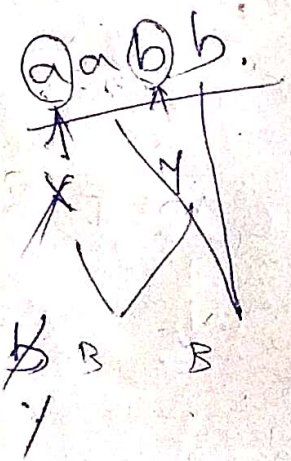
$\not{b}$ $\not{b}$ $\not{b}$ b

X X ↑ Y

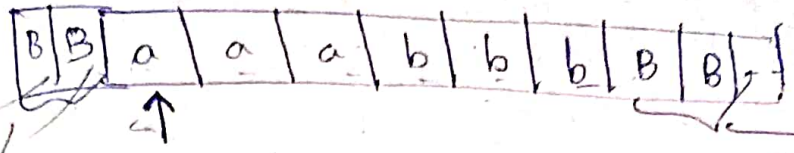
- Move right - -

$\not{b}$ $\not{b}$ $\not{b}$ $\not{b}$

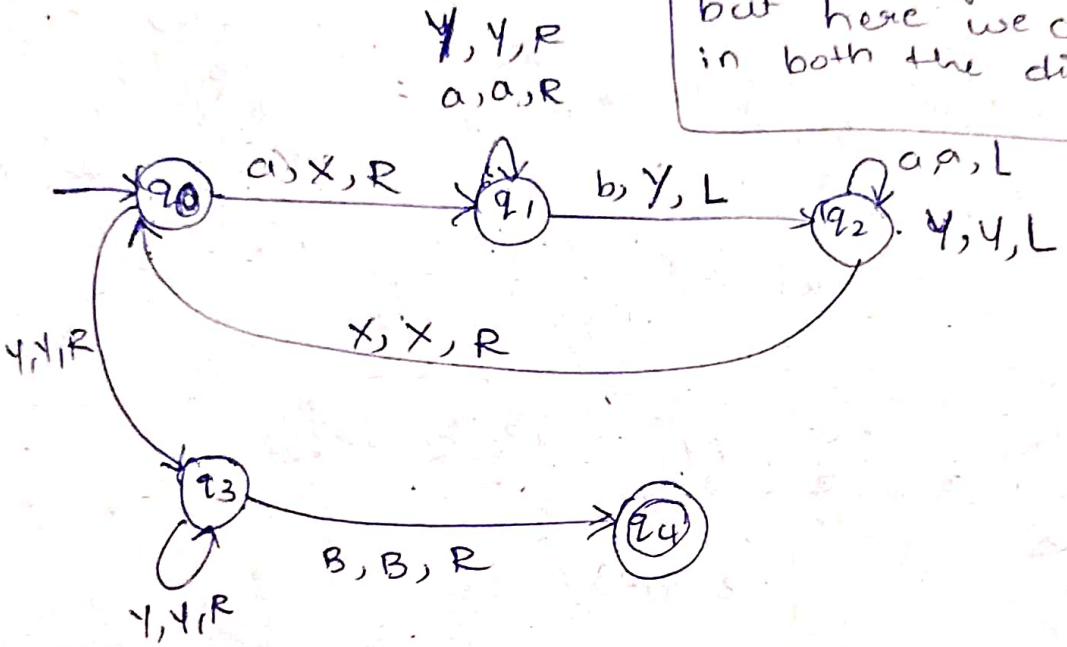
X X Y Y



let string:



In FA, we can move from only left to right, but here we can move in both the directions.



$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, Y, L)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, X) = (q_0, X, R)$

$\delta(q_0, Y) = (q_1, Y, R)$

$\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_0, Y) = (q_3, Y, R)$

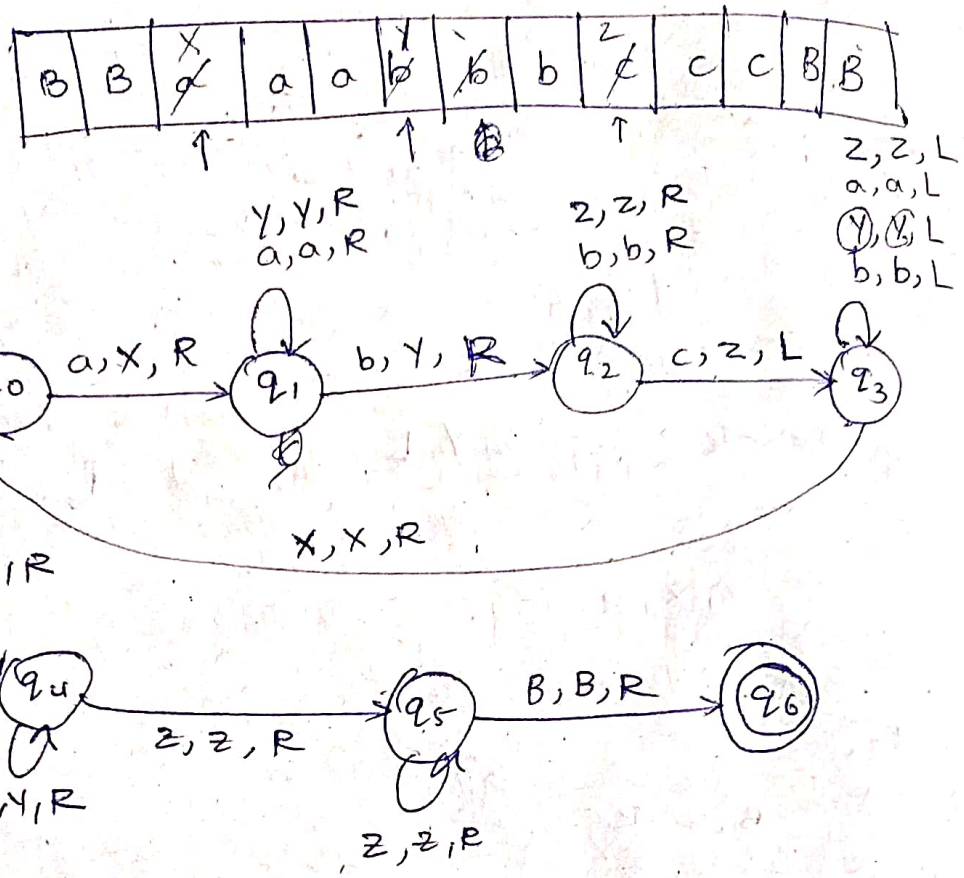
$\delta(q_3, Y) = (q_3, Y, R)$

$\delta(q_3, B) = (q_4, B, R)$

a

a, z0 | a z0

* $L = \{a^n b^n c^n \mid n \geq 1\}$

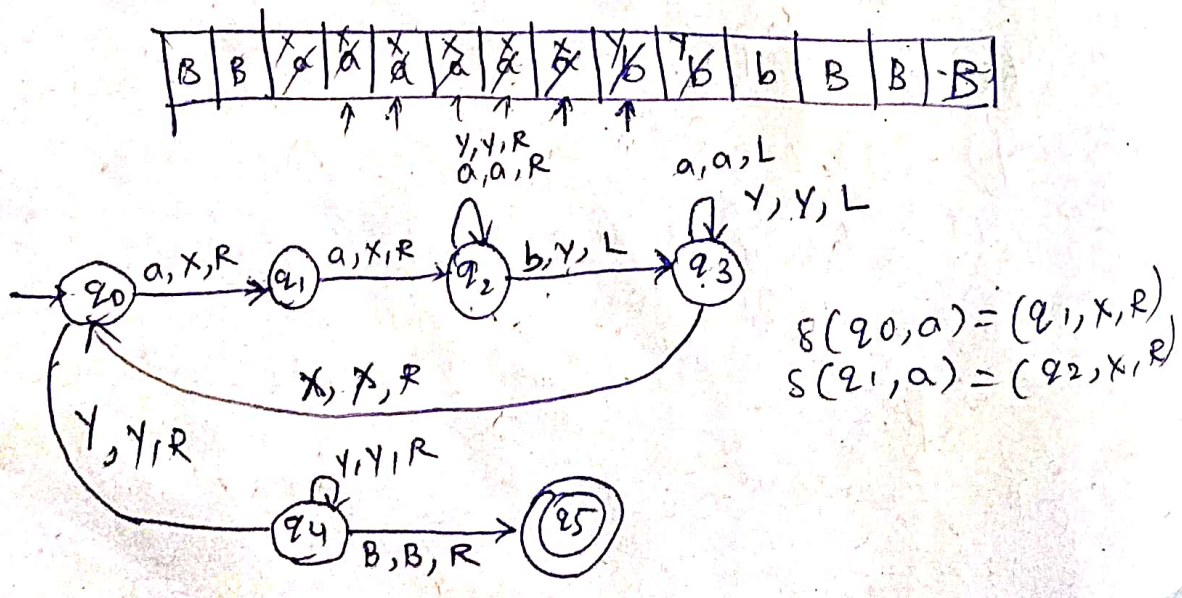


* $L = \{a^{2n} b^n \mid n \geq 1\}$

$L = \{aab, aaaabb, aaaaaabbb - - -\}$

Logic:- (i) Make 2 a's into 2 X's & move right until we get b.

(ii) Make 1 b as Y & move left - -

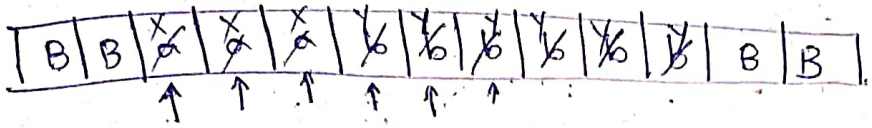


$\delta(q0, a) = (q1, X, R)$

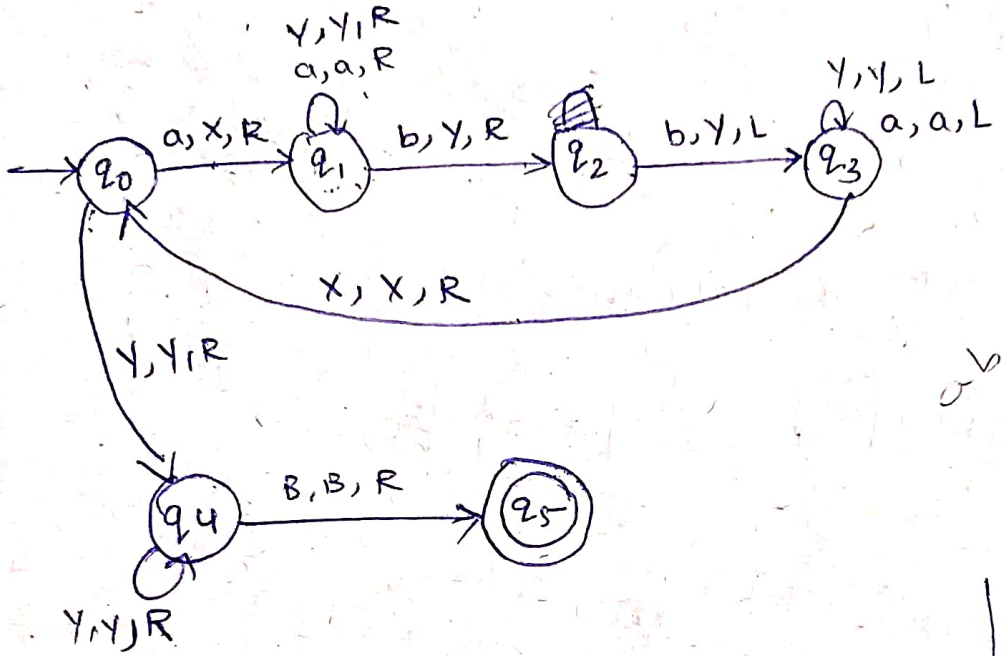
$\delta(q1, a) = (q2, X, R)$

* $L = \{a^n b^{2n} \mid n \geq 1\}$

$L = \{abb, aabbbb, aaabbbbbb - - - - - \}$

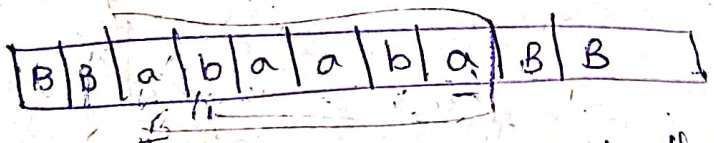


* make 1 a as X
* Make 2 b's as 2 Y's.

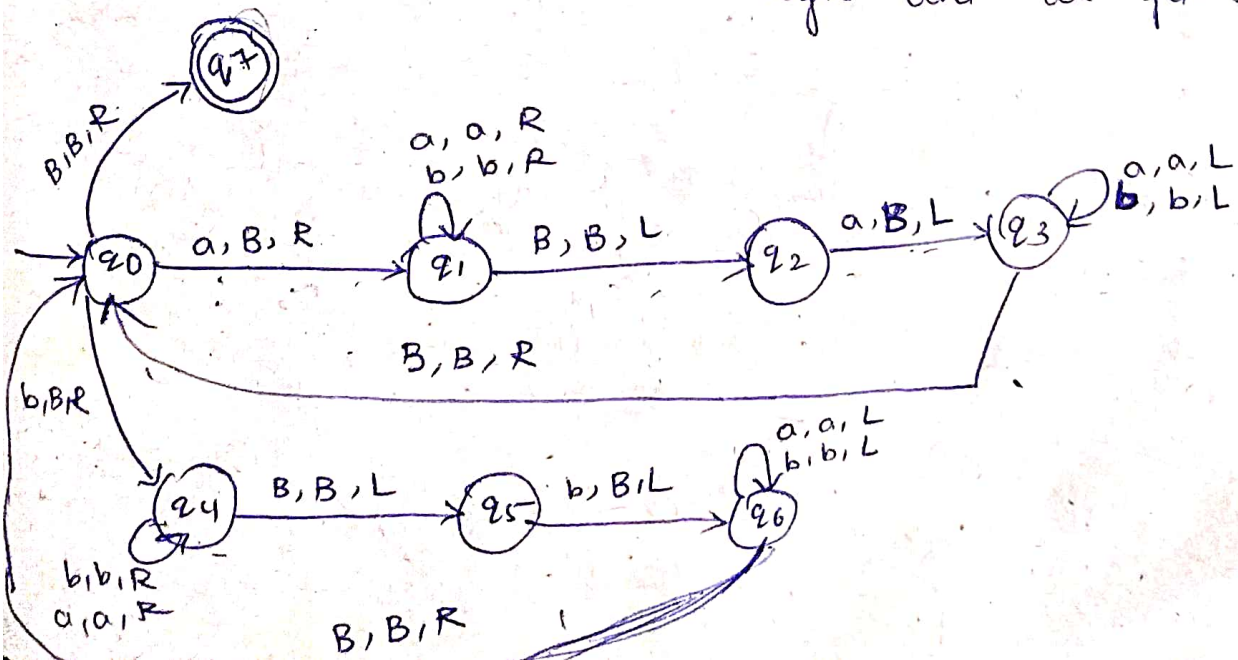


* Even Palindrome

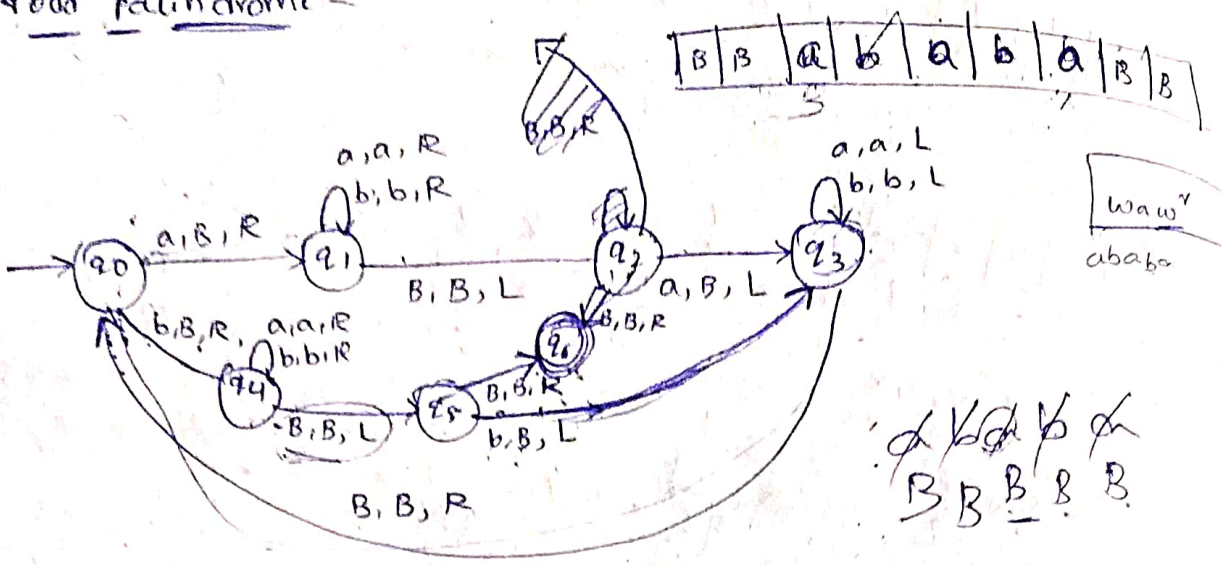
Ex:- aba aba → len = 6



* if there is 'a' make it as 'B' and move towards right until we get 'B'.



* odd palindrome:-



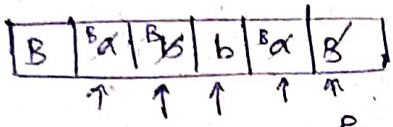
waw^r

ababa

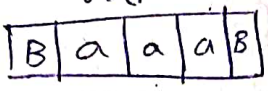
B B B B B

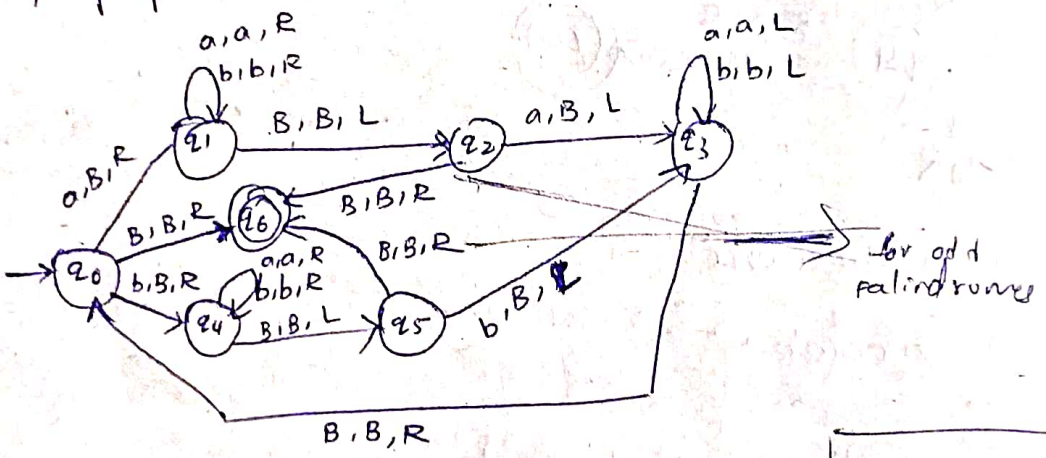
* Both even and odd palindrome

even



odd



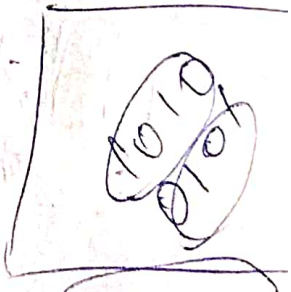


* 1's complement of 2's complement

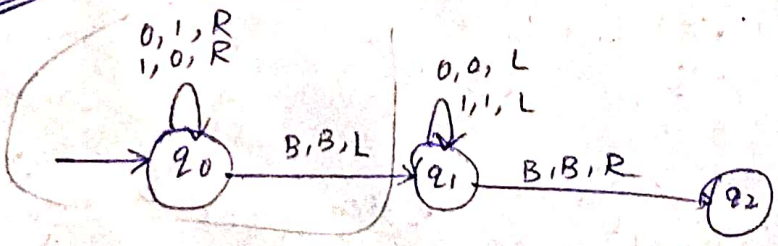
1's:- 1 0 1 0

0 1 0 1

MSB LSB



1's:-



1 0 1 0

0 1 0 1

1

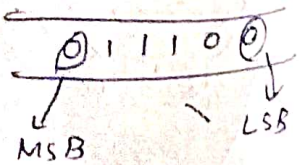
0 1 1 0

1 0 0 1 0 0

0 1 1 0 1 1

1

2's:- To get better result, traverse from LSB to MSB



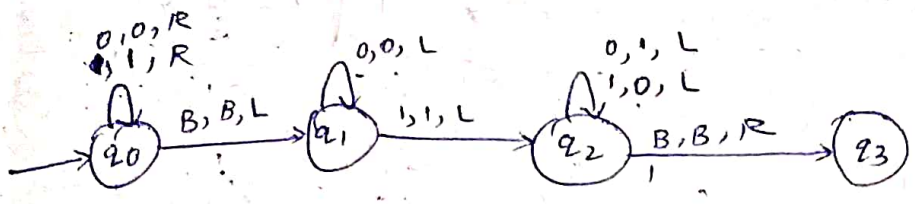
If we want we can stop here

Logic:- from LSB no need to change upto '1'; after '1' change all 0's to 1's & vice versa.

B Y Ø 1 O B

0 1 ⇓

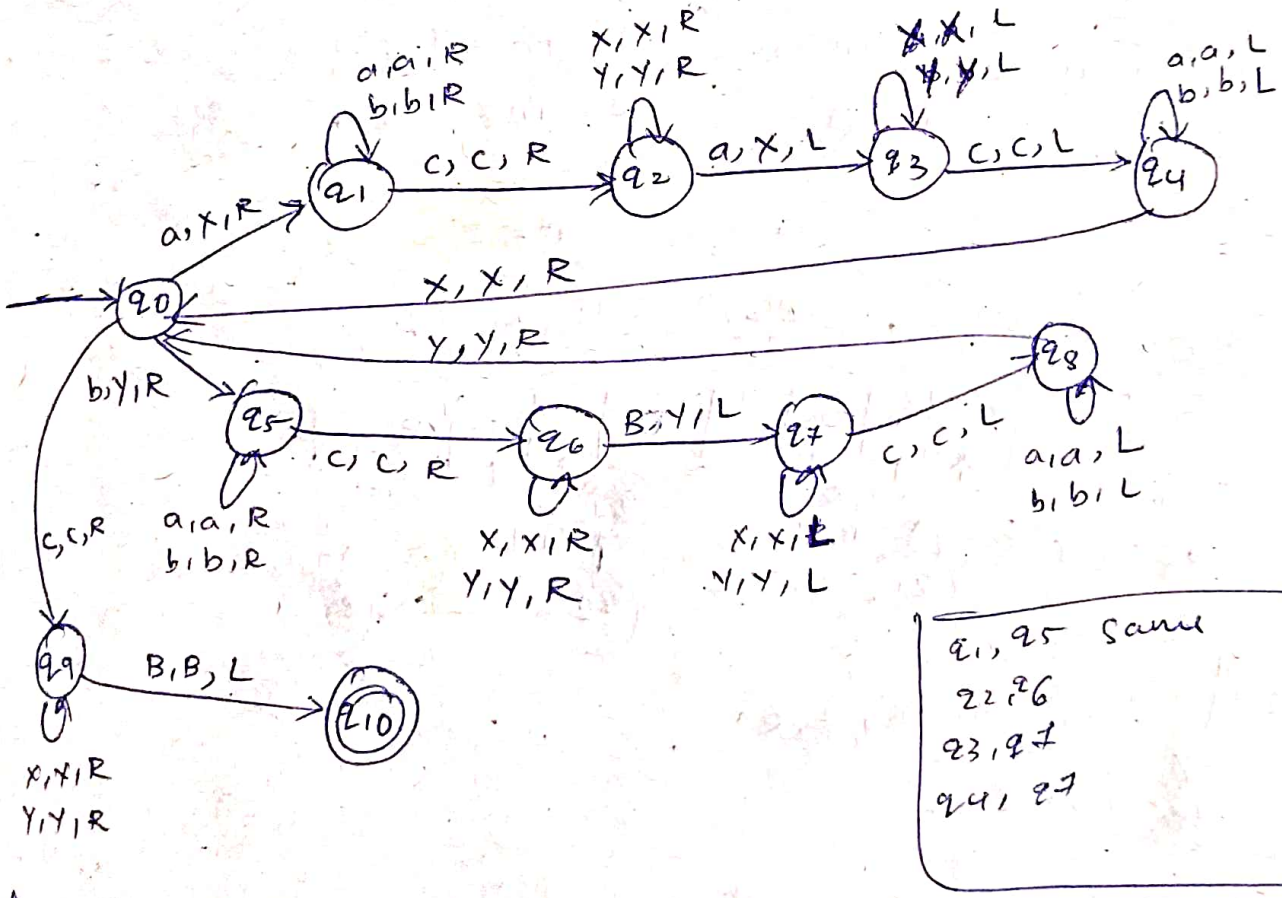
0 1 1 0 ✓



* $L = \{ wcw \mid w \in (a+b)^* \}$

$L = \{ abcab \ldots \}$

Logic:- Replace a with X move towards right till we get 'c', after c if we have a replace with X & move towards left unit we get X, then move right -- Ily for b also, replace b with Y.

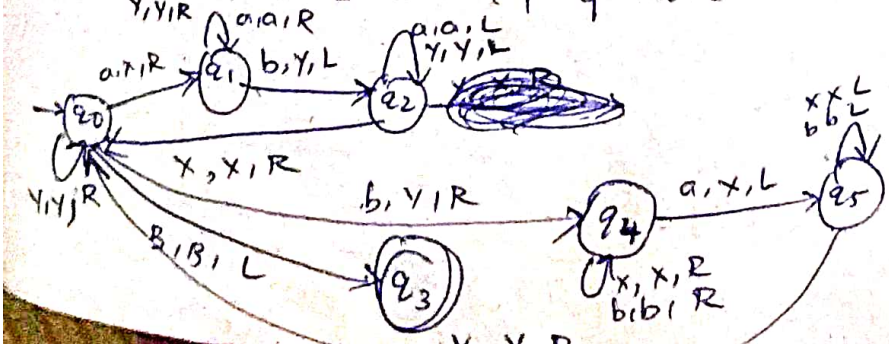


q1, q5 same
q2, q6
q3, q7
q4, q8

* equal no. of a's & equal no. of b's

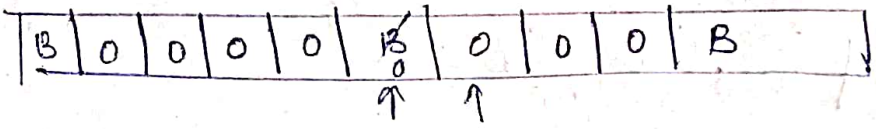
→ Replace a with X and move towards right till we get b, now replace b with Y & move towards left till X. ---

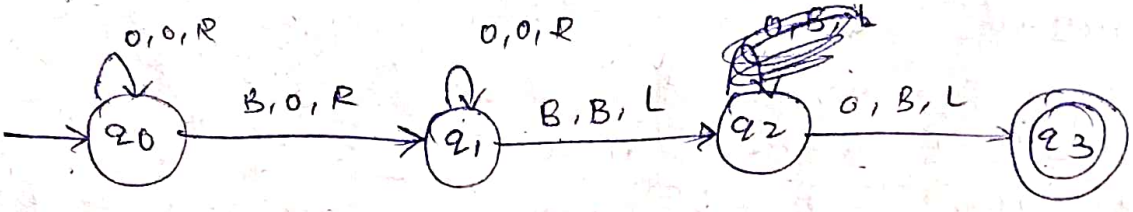
B a a b b B



* Addition of 2 numbers (unary notation) either 0's (or) 1's

let a = 4 b = 3



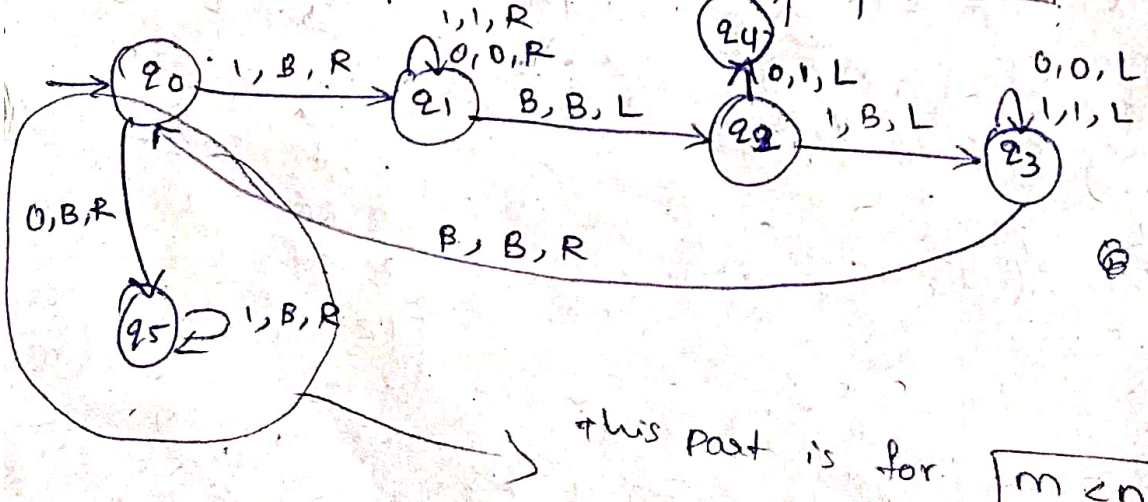


* Subtraction (unary notation)

$$f(m,n) = \begin{cases} m-n & ; \text{if } m \geq n \\ 0 & ; \text{if } m \leq n \end{cases}$$

m = 4 , n = 2

11 11 − 11 = 11



this part is for m < n