Programming Techniques of Turing Machines

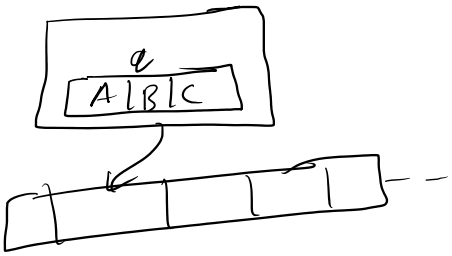
25 November 2020 12:02 PM

1. Storage in state
2. Multiple tracks
3. subroutines ⟶ $m * n \Rightarrow 0^m 1 0^n 1 0^{mn} \Rightarrow 0^3 1 0^2 1 0^6$

Note:- Here none of these techniques are extending the functionality of TM

1) Storage in state :-

finite control not only to represent a position in the program of TM, but also to hold a finite amount of data



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

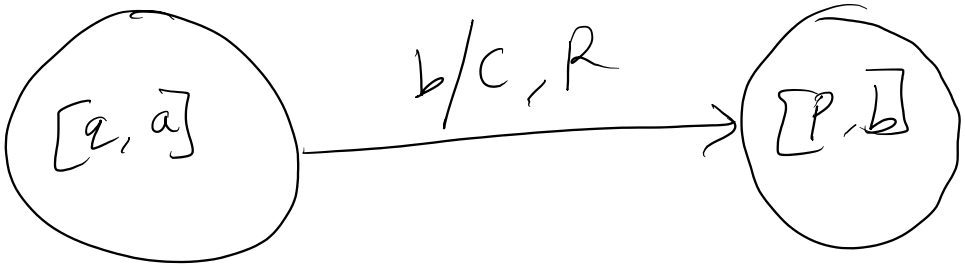
Q is of the form $[q, a]$ $\quad q \in Q$, $a \in \Upsilon$

δ is defined as $\Rightarrow (Q \times \Upsilon) \times \Upsilon \longrightarrow [Q \times \Upsilon] \times T \times [L, R]$

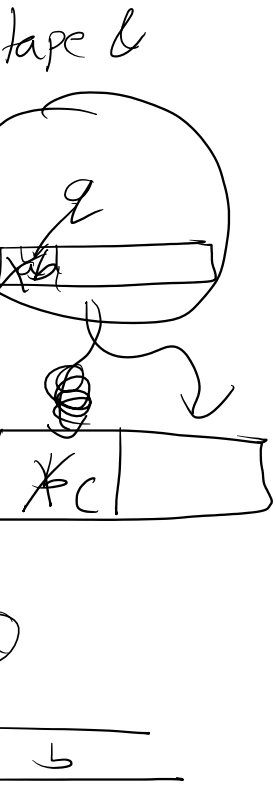
$$\delta([q, a], b) = ([p, b], C, R)$$

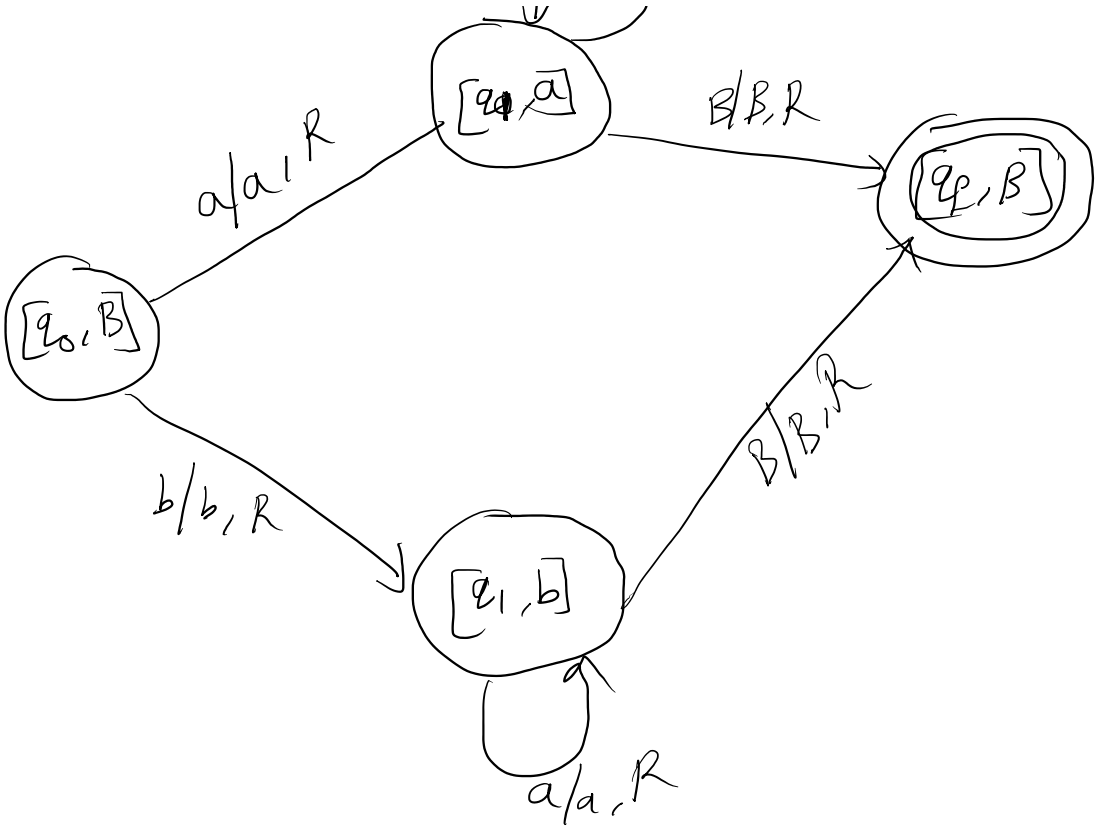
q with a as symbol replaces 'a' by 'b' on seeing b on the

changes state from q to p



Construct TM that accepts the language $\{a b^* \cup b a^*\}$

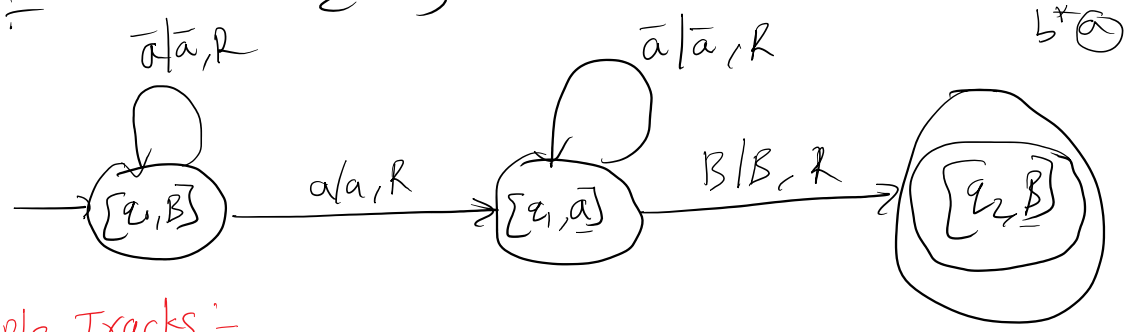

tape b
q
Xb
Xc



Ex: 2 ; Design TM & check that 'a' should not be repeated for strings of a's & b's

$\Sigma = \{a, b\}$ $ab^* \Rightarrow a\bar{a}$

Sol :-



Multiple Tracks :-

In this a single tape is assumed to be divided into tracks.

3) Sub routines :-

function → called → function

} return

a+L

}
return
}

Ex:- Multiplication ⇒ Language ⇒ $0^m 1 0^n 1 0^{mn}$ (input: I_1, I_2)

Informal :-

B 0 0 1 0 0 0 1 B B B

→ X 0 0 1 0

← X X 0 1 0 0

→ X X 0 1 0 0 0 0 0

← copy

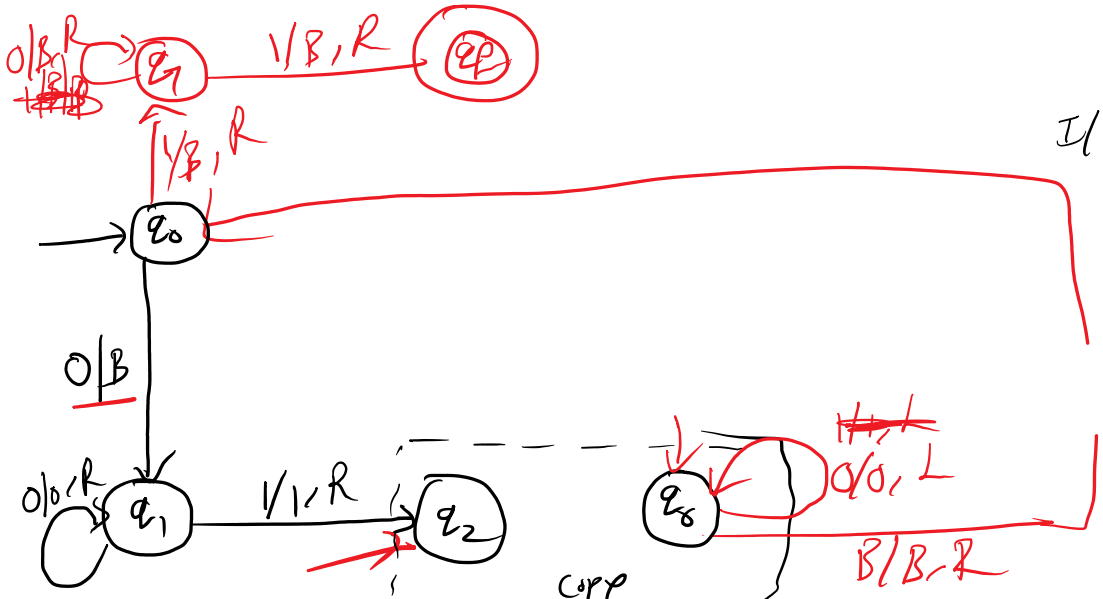
→ subtracting

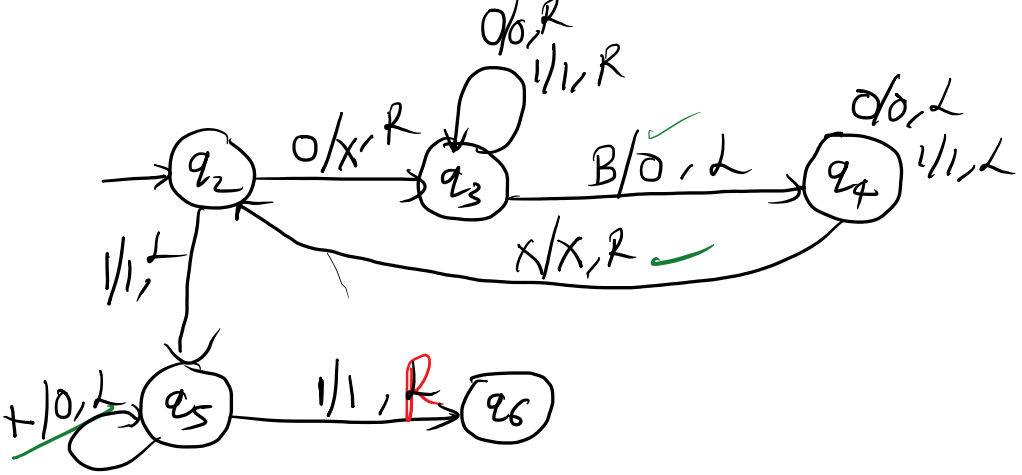
00 X 000 = 000000

Method 1 :- 00 00 00

Method 2 :- 000 000

C





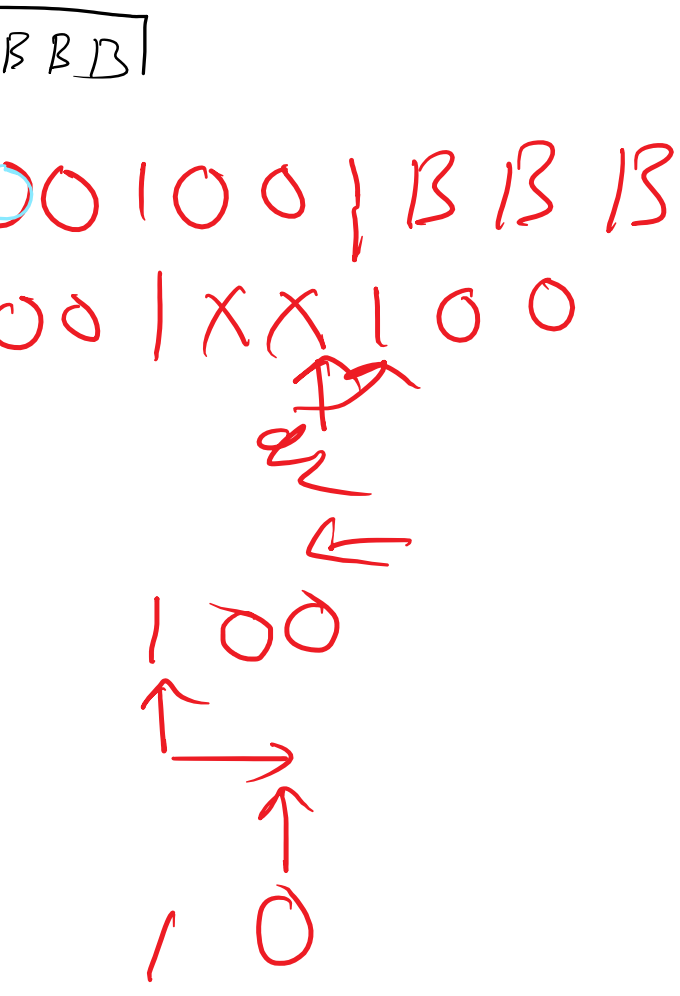
I/P | B 0 3 1 0 2 1

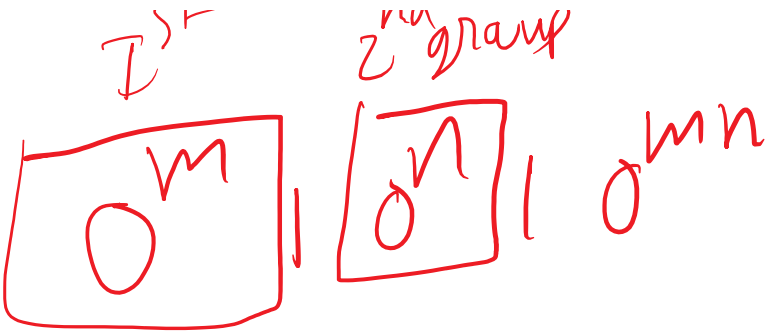
B 0 0

B B 0

3

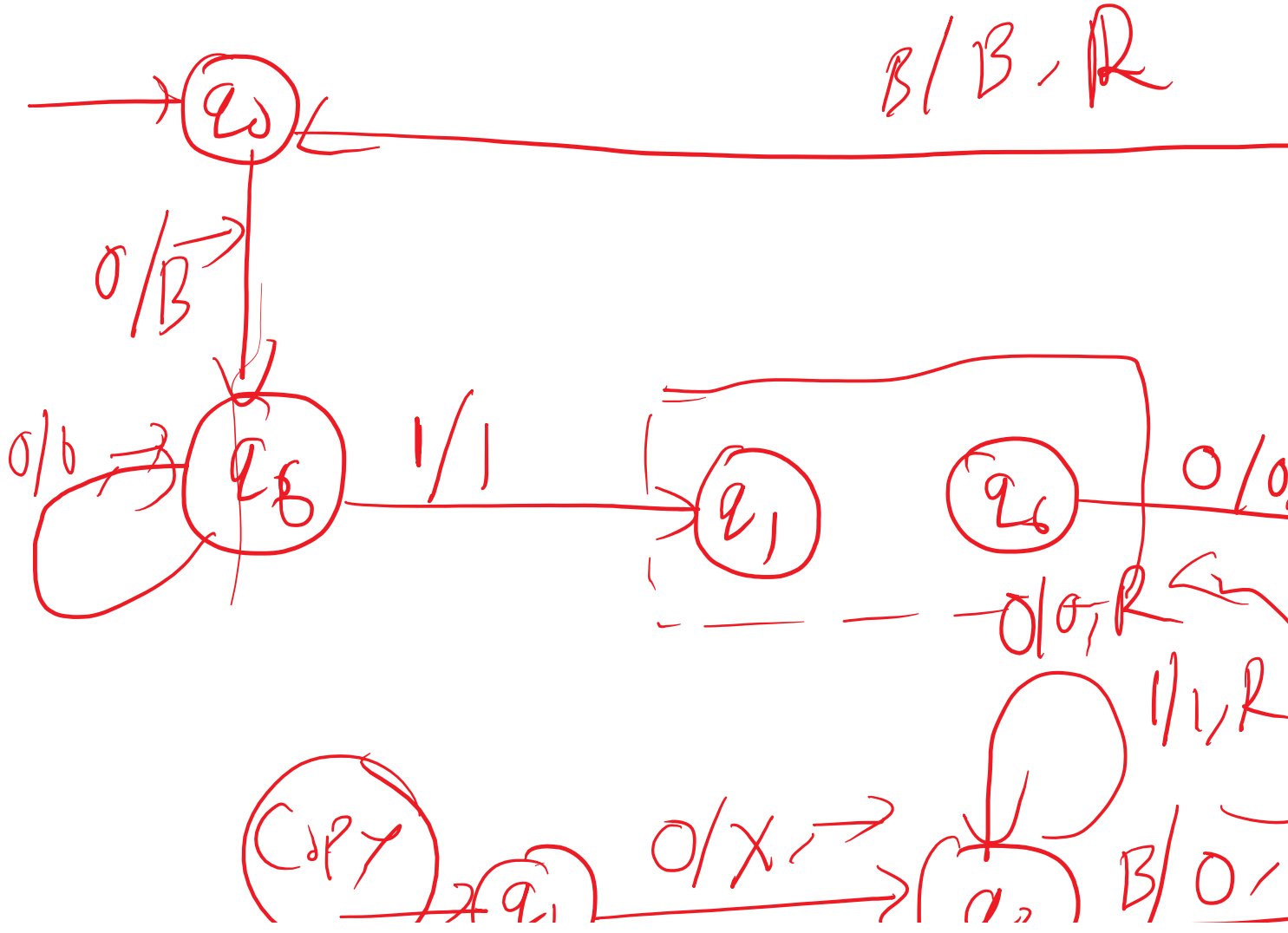
1st 2nd group

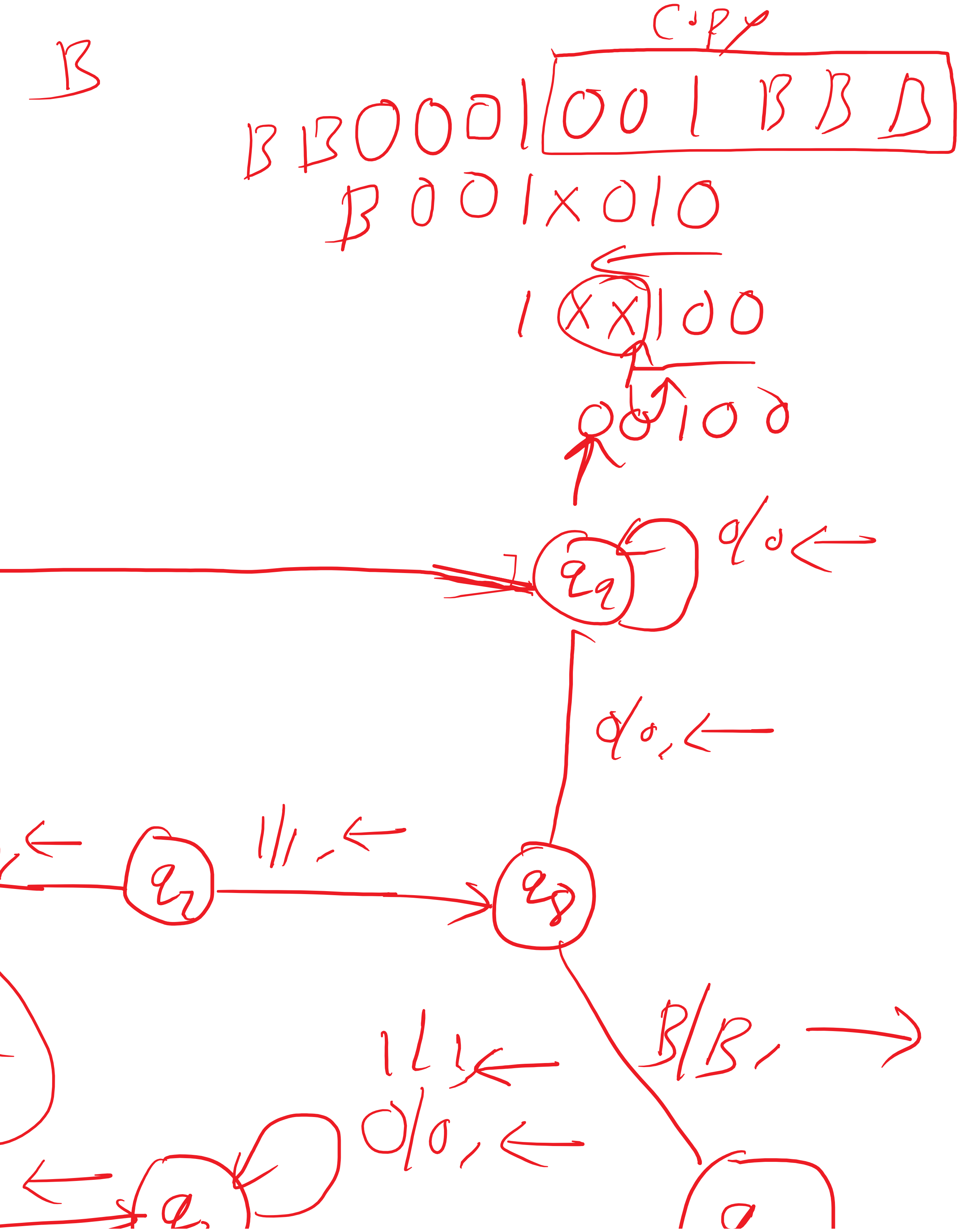


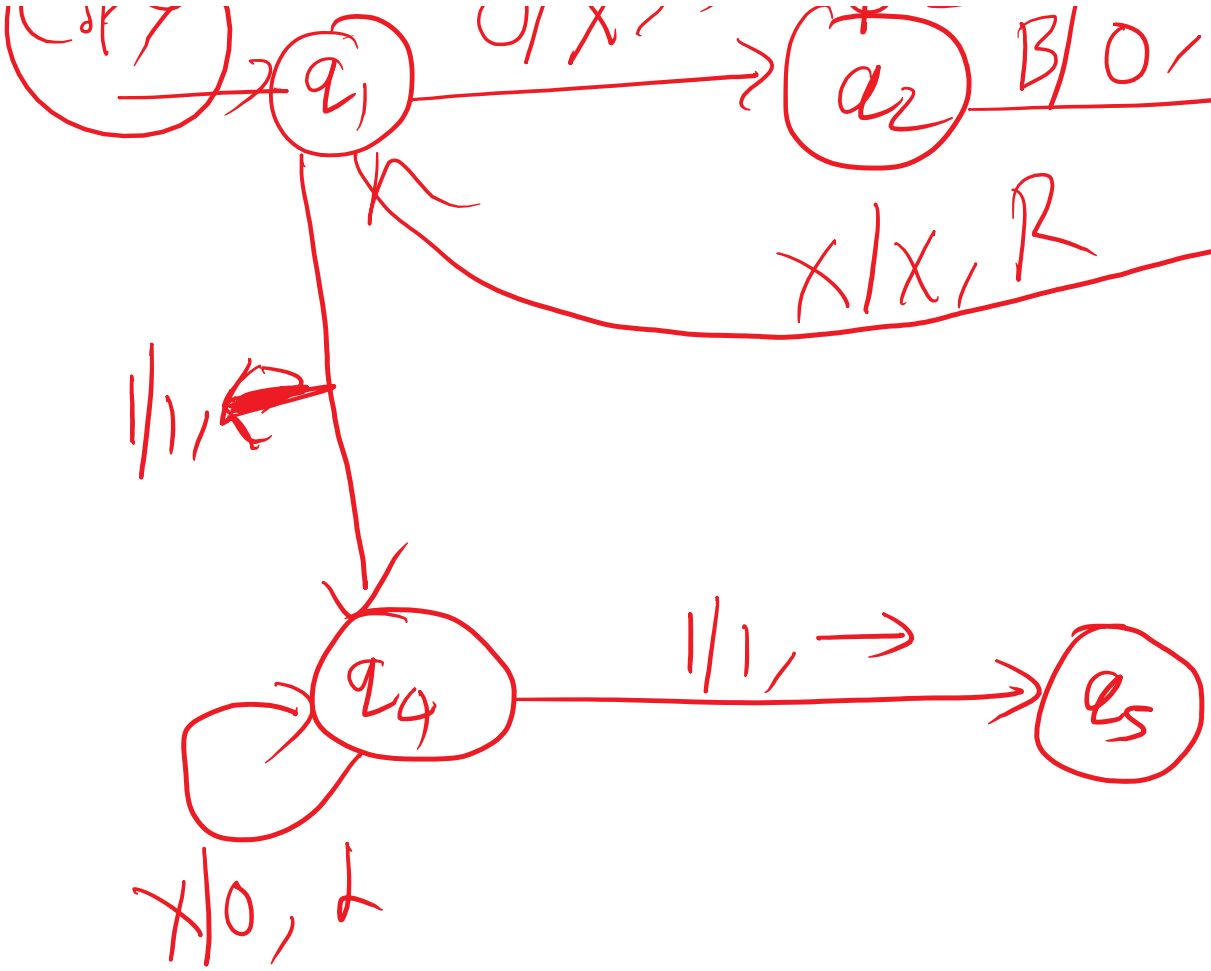


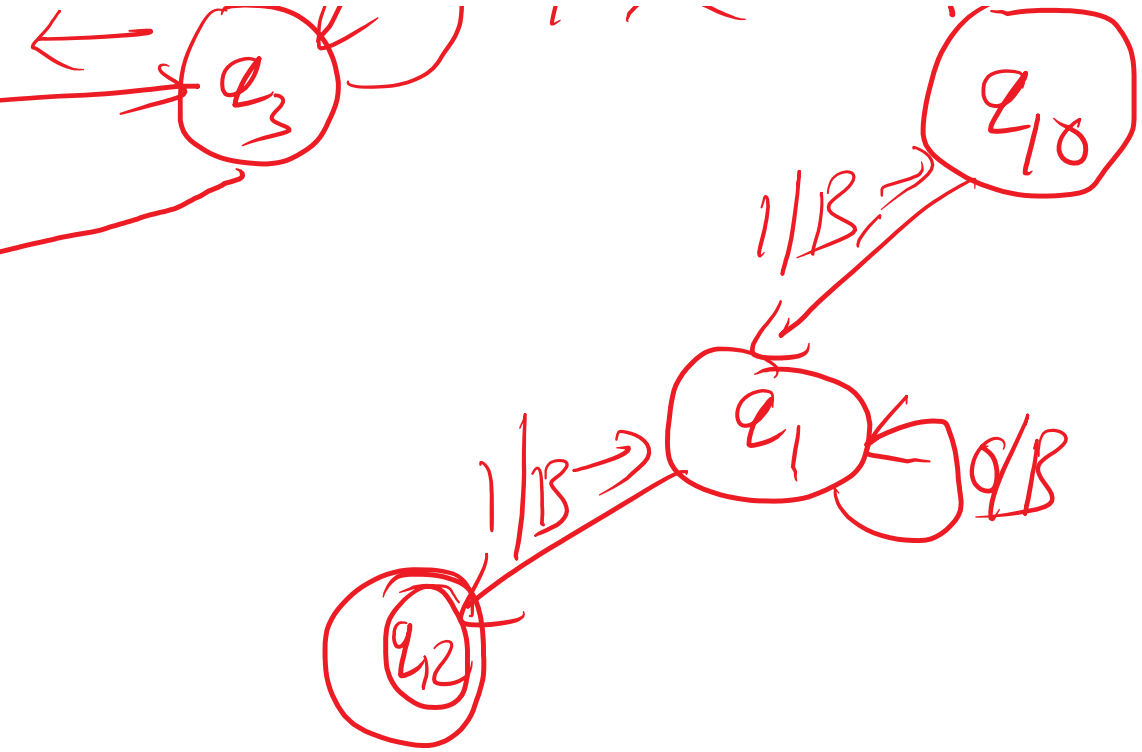
1) Reading m & n

2) Marking first group first zero to




Copy
BB0001001BBB
B001X010
1XX100
00100
q9
0/0,←
1/1,←
q7
q8
B/B,→


q_1
q_2
q_4
q_5
X|X, R
1|1, L
1|1, →
X|0, L


q3
q10
1/B
q1
1/B
q/B
q12