(13) Design a TM to make a copy of string over $\{0,1\}$

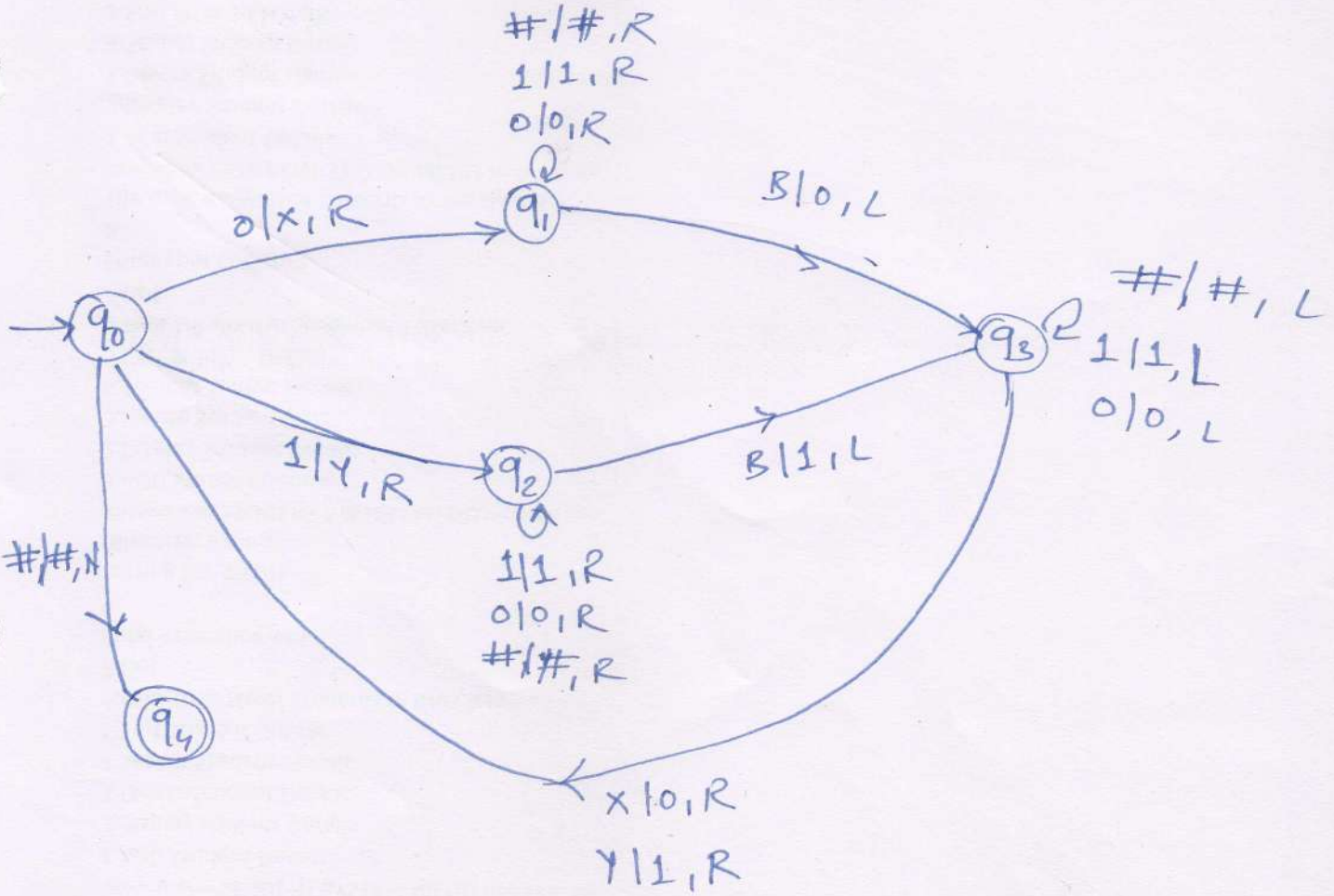
⇒ I/P:

B	1	1	0	0	#	B	...

O/P:

B	1	1	0	0	#	1	1	0	0	B	...

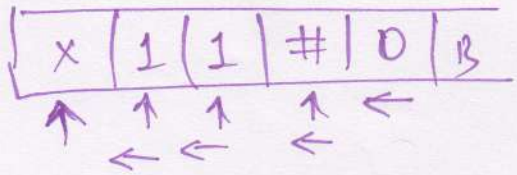
⇒ Two copies are seperated by #

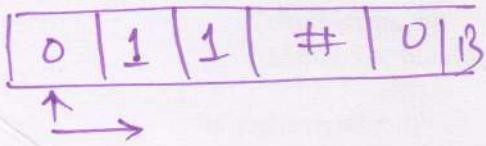


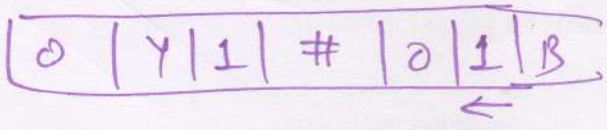
⇒ First input & Blank 'B' should be updated then again update the X & Y inputs.

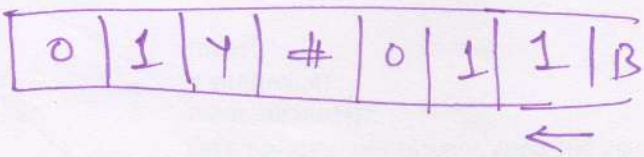
B	0	1	1	#	B

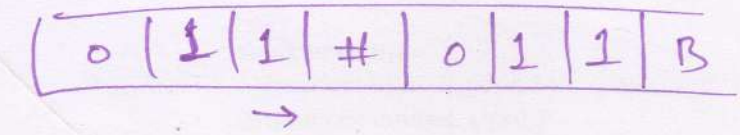
B	X	1	1	#	0


X 1 1 # 0 B


0 1 1 # 0 B


0 Y 1 # 0 1 B


0 1 Y # 0 1 1 B


0 1 1 # 0 1 1 B

(14) Construct a TM for checking well formness parenthesis.

⇒ To solve this, we need to match every occurence of "(" for every occurence of ")". At the end if any parenthesis is unmatched then the given string is declared not balanced.

① First search for the occurence of ")", for this process, in the initial state q_0 ignore all "(" until ")" is seen.

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

② On the occurence / finding ")" replace it by 'X' change to new state & travel left for the first occurence of "(". It is used to find "(" for ")" while travelling back it can see 'X'.

$$\delta(q_0,)) = (q_1, X, L)$$

$$\delta(q_1, X) = (q_1, X, L)$$

③ If "(" is found, replace it by 'X', If X is not found, enter into rejecting state. In this for ex. q_4 acts as both initial state & return state

$$\delta(q_4, (\,) = (q_0, X, R)$$
$$\delta(q_4, B) = (q_3, B, R)$$

④ Repeat step 1 & 2 until a B is encountered.

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

⑤ If B is encountered enter into new state & check if is no ")" unbalanced.

Y Y R
X X R
R R R

Y, Y, L
X X L

Y Y, L
X X, L

① (B
q_0 q_2
q_3 ←

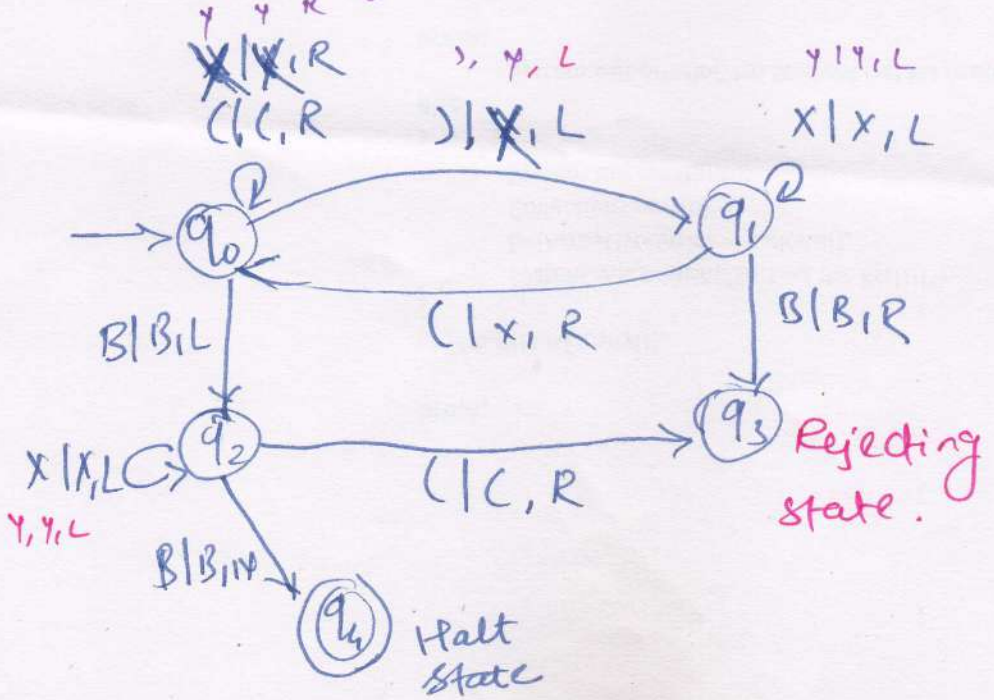
③ If "(" is found, replace it by 'X', If X is not found, enter into rejecting state. In this for ex. q_4 acts as both initial state & return state

$$\delta(q_4, (\,) = (q_0, X, R)$$
$$\delta(q_4, B) = (q_3, B, R)$$

④ Repeat step 1 & 2 until a B is encountered.

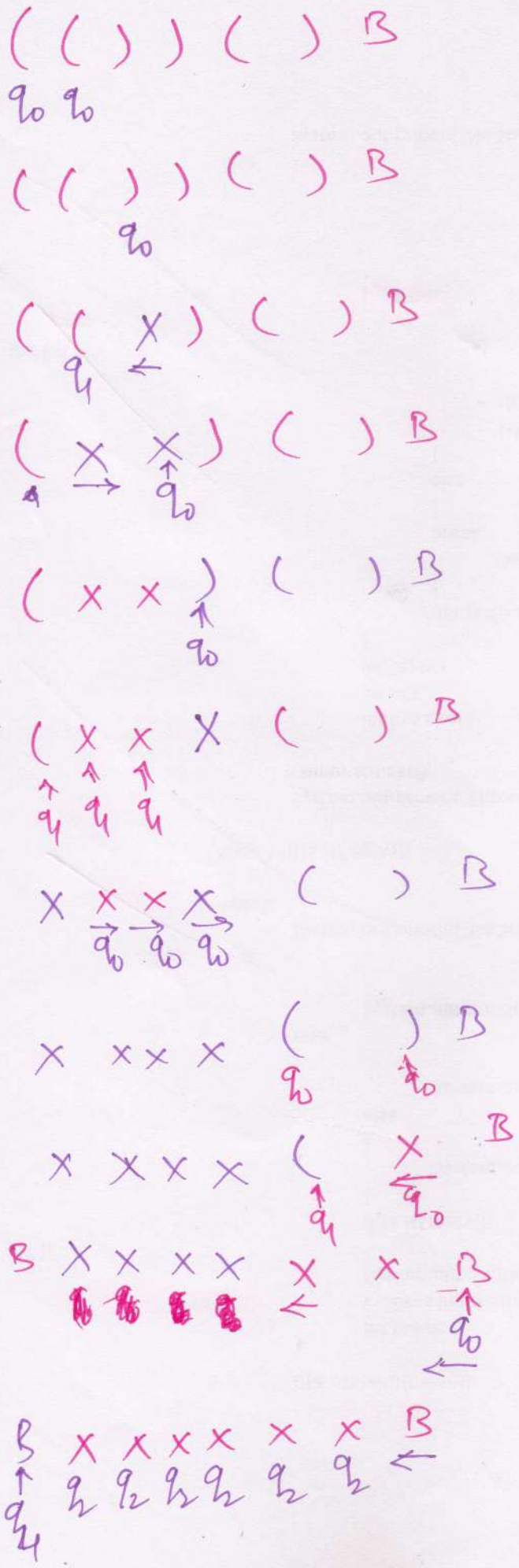
$$\delta(q_0, X) = (q_0, X, R)$$
$$\delta(q_0, B) = (q_2, B, L)$$

⑤ If B is encountered enter into a new state & check if is no ")" unbalanced.



① (B
q_0 q_2
q_3

② B) B
q_0 q_4
B X
q_3



(15) Design a TM which recognizes palindrome over alphabet {a, b}

⇒

① | a | b | b | a | B | . . .

B (q0) b a B

B b B B

B B B B

B B B B ⇒ Accept.

② | b | a | a | b | B

B a a b B

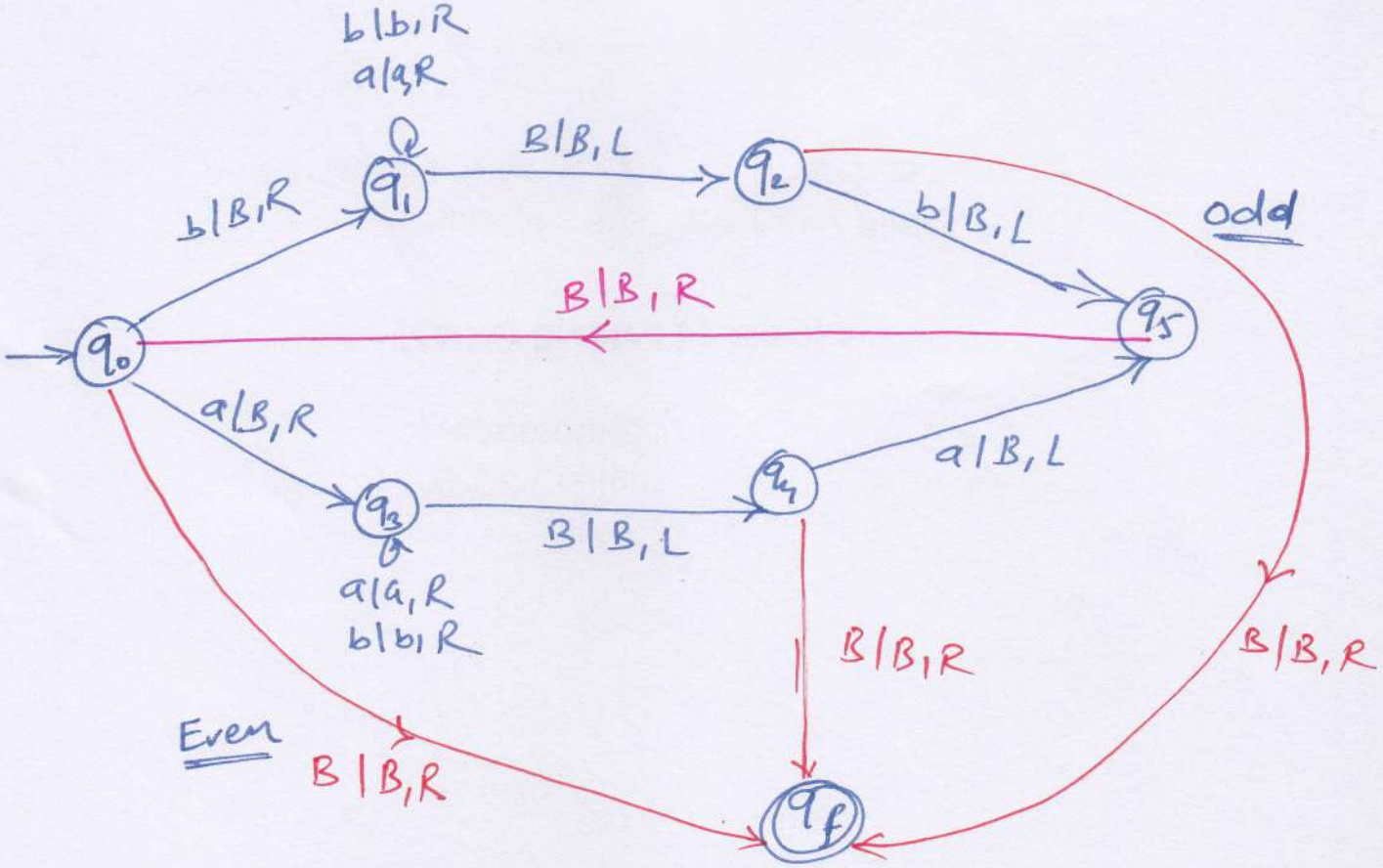
B a a B B

B B a B B

B B B B B (q3)

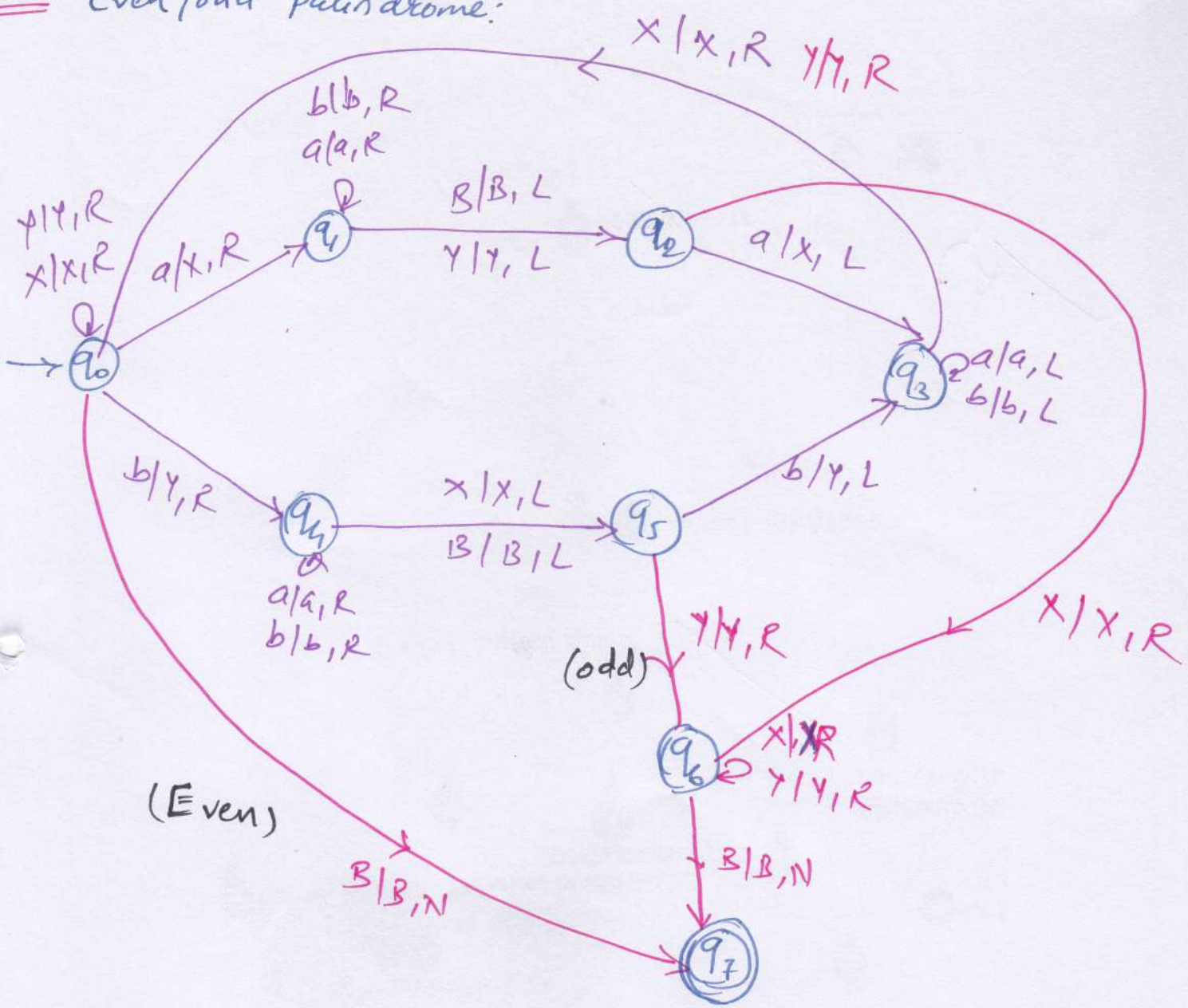
B B B B B (q4) Accept

③ w = abbaB



DR

OR Even/odd palindrome:



W = | a | b | a | B |

X → b a B ←

X b X ← B

X → Y → X ← B

X X X → B Accept

W = | b | a | b | B |

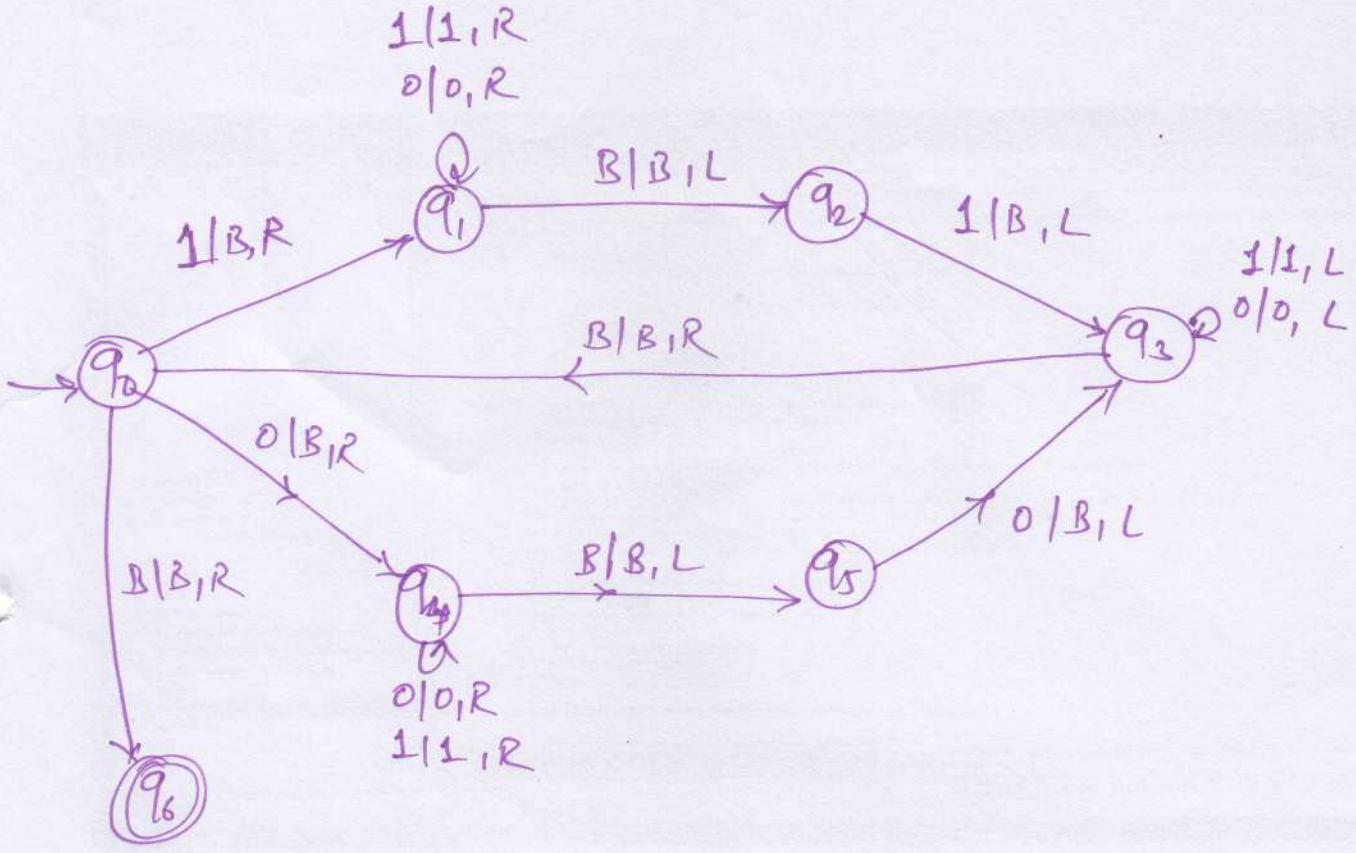
Y → a b B

Y a Y ← B

Y → X → Y B

Y X → Y → B

16) Design a TM for accepting strings of the language defined as $L = \{ ww^R \mid w \in (0+1)^* \}$



W = X 0 0 1 B

B 0 0 1 B ←

B 0 0 B B ↔

B B 0 B B → → → ←

Accept B B B B B → ←

W = 0 0 0 0 B

B 0 0 0 B → ←

B 0 0 B B → ←

B B 0 B B → ←

B B B B B ← Accept

W = 1 0 0 1 B

W = 0 1 0 B

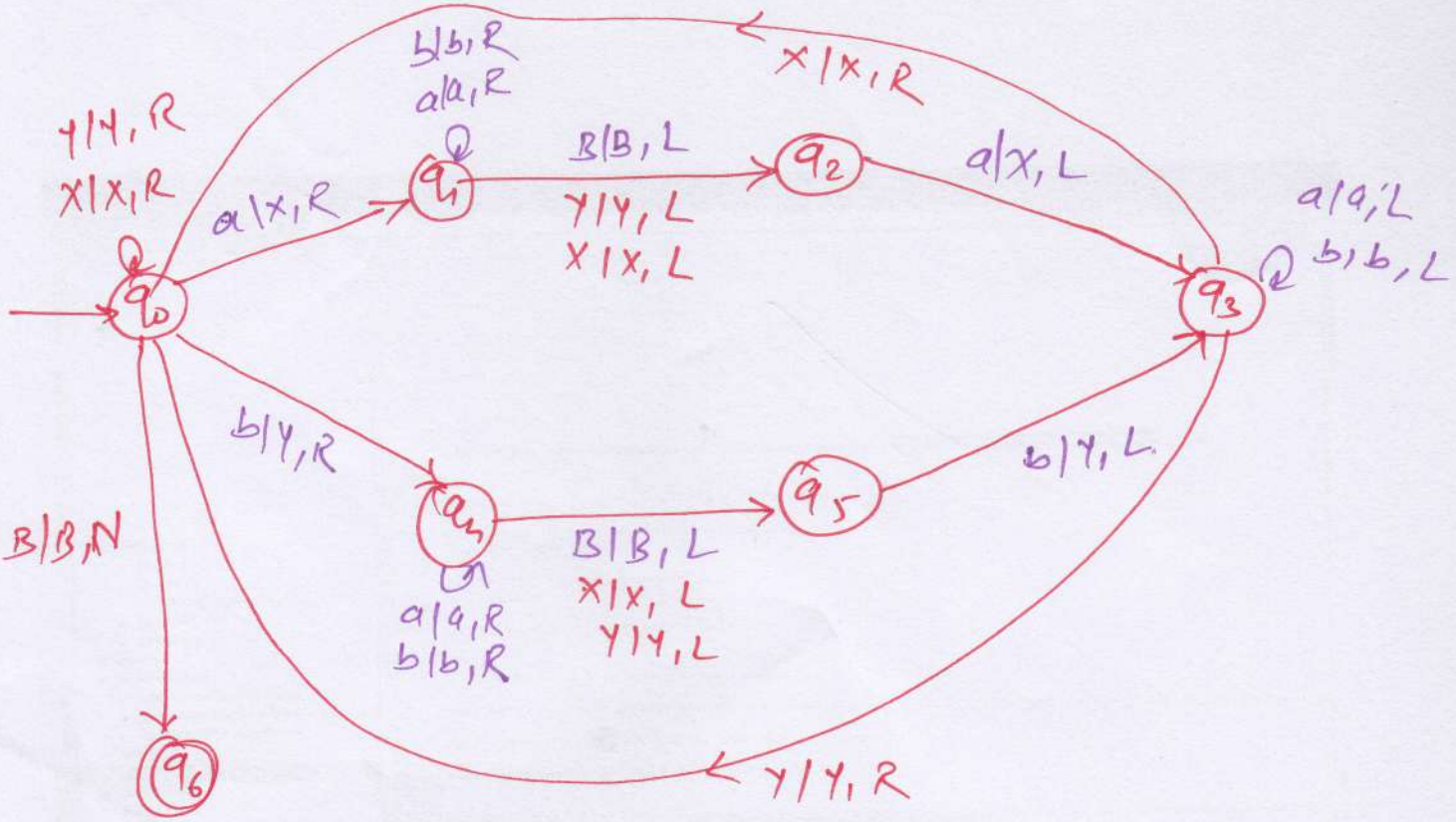
B 1 0 B

B 1 B B

B B B B Rejected

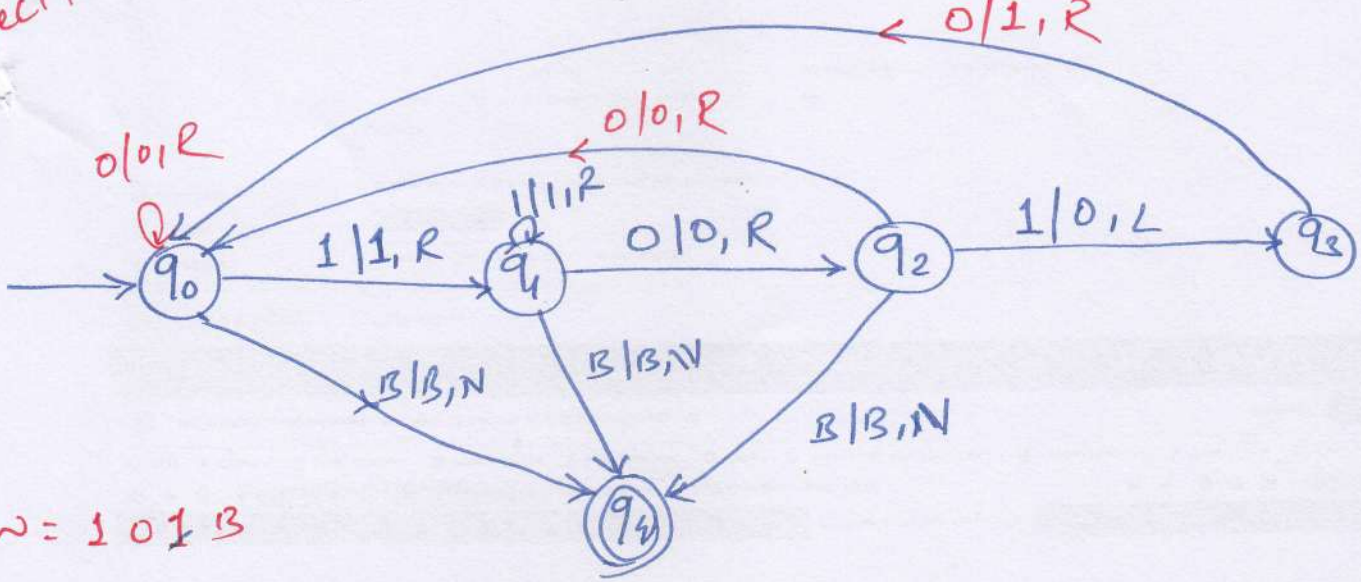
q_0 q_2

(16) Even palindrome for $\Sigma = \{a, b\}$



w = abbaB , w = babaB , w = aaaa.

(17) Design a TM that recognizes occurence of substring 101 & replaces it with 110.



(K) w = 1 0 1 B

1 0 0 B

1 1 0 B

① from q_0 to final state

11 10 1 00 B

1 1 1 0 0 0 0 B

1 1 1 1 0 0 0 B

② from q_1 to final state

0 0 10 1 1 B

0 0 1 0 0 1 1 B

0 0 1 1 0 1 1 B

③ From q_2 to final state

1 1 1 0 1 1 1 0 B

1 1 1 0 0 1 1 0 B

1 1 1 1 0 1 1 0 B

④ $w = 11101010B$

---

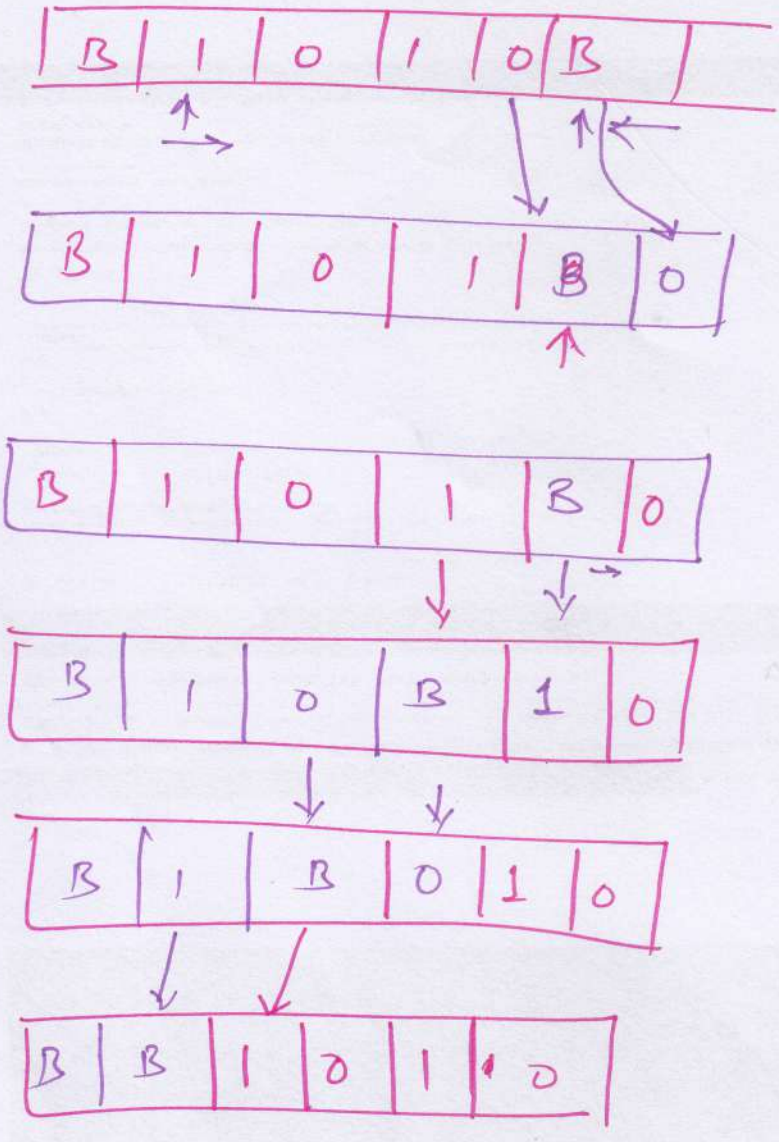
⑱ Design a TM to perform right shift operation on a binary no.

→ A right shifting TM will shift the i/p string right by 1 place.

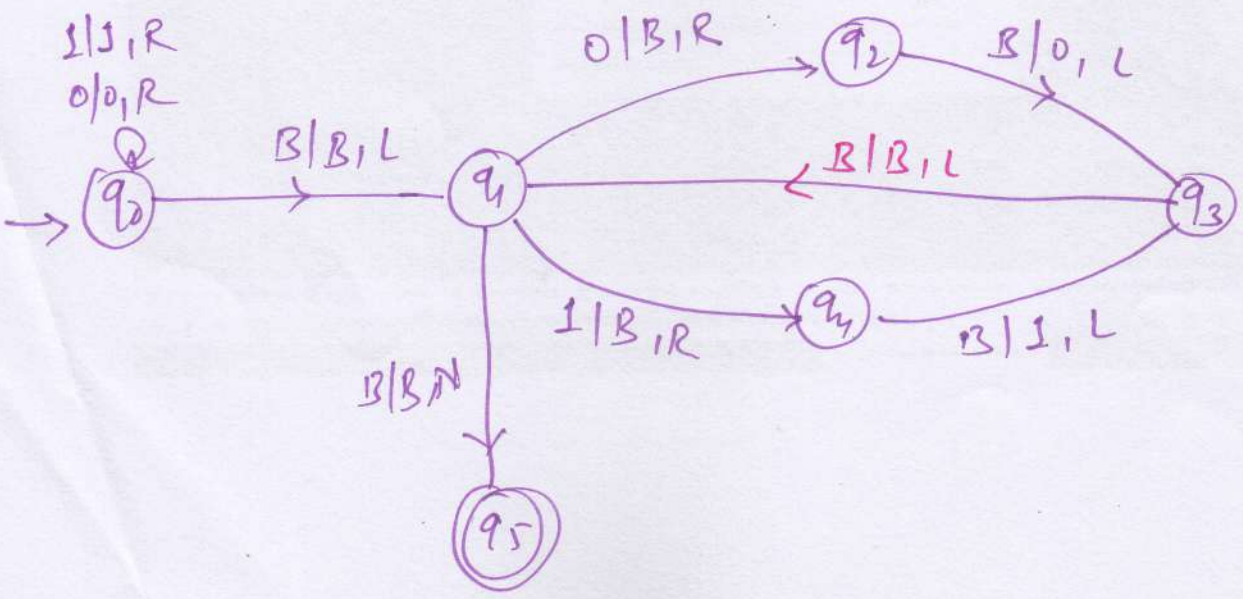
→ A right shifting should start from the rightmost character.

→ Each character is shifted to right starting from right end & working towards left end.

Ex:- 1 0 1 0

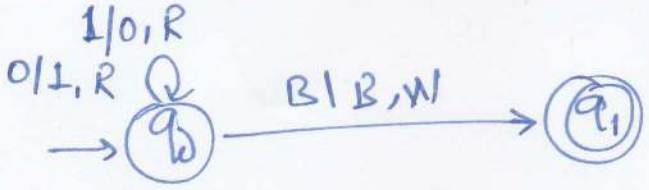


Starting from Right most



(19) Design a TM to accept strings $L = (0+1)^*$

(20) TM for finding 1's complement of given no.



---

* Turing Machine as Computational Machine :-

→ To use TM as a computational Machine, it is required to place nos. as 0^m.

⇒ Suppose it is required to add two nos. i.e. $f(m+n) = m+n$; then the nos. m & n are to be placed on the tape as $0^m 1 0^n$; where 1 is a seperator for the nos. m & n once Processing is completed & TM halts, the tape would have the contents as $0^{(m+n)}$ which is required result of Computation.

Decimal No.	Unary no.
0	B
1	0
2	00
3	000
⋮	
n	$0 \ldots . 0^n$

(21) Design a TM to add two nos. a & b

→ let the nos. 2 & 3, The add^n of two nos. = 5

So, the nos. are placed as $B\,0^2\,1\,0^3\,B$.

After processing the type content would be

$B\,0^5\,B$

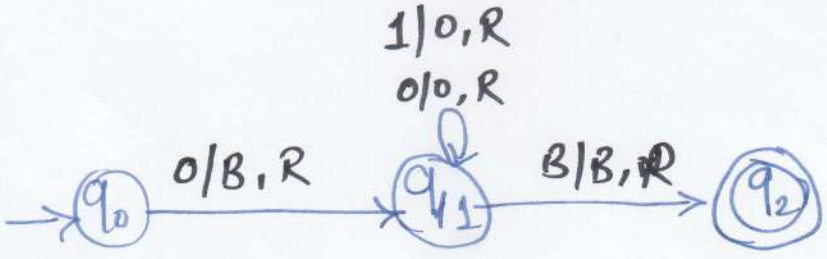
⇒ Replace the occurence of 0 by B & move to right & replace 1 to 0 so that it is in required form as $B\,0^5\,B$.

Sequence of Steps is given—

① $\delta(q_0, 0) \Rightarrow (q_1, B, R)$

② In q_1 state move to right on '0' & when I/p is 1 replace it by '0'

$$\delta(q_1, 0) = (q_1, 0, R)$$
$$\delta(q_1, 1) = (q_1, 0, R)$$
$$\delta(q_1, B) = (q_2, B, R)$$



$W = 00\,1000$

B 0 0 1 0 0 0 B

$q_0 \Rightarrow$ B 0 1 0 0 0 B
(q_1)

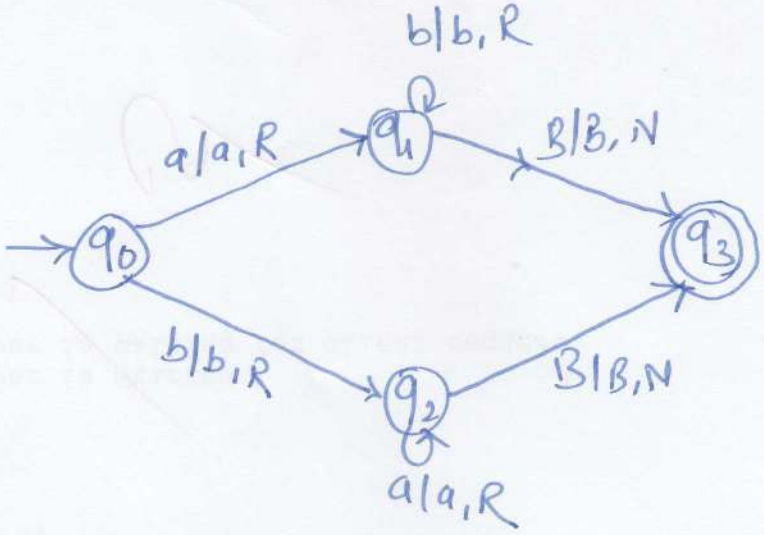
B 0 0 0 0 0 B
(q_1)

B 0 0 0 0 0 B
(q_1)

$\Rightarrow W = 000100$ $\Rightarrow W = 00100$

(22) Design a TM that accepts strings of ab^* or ba^*

$\Rightarrow$ $w = (ab^* + ba^*)$



---

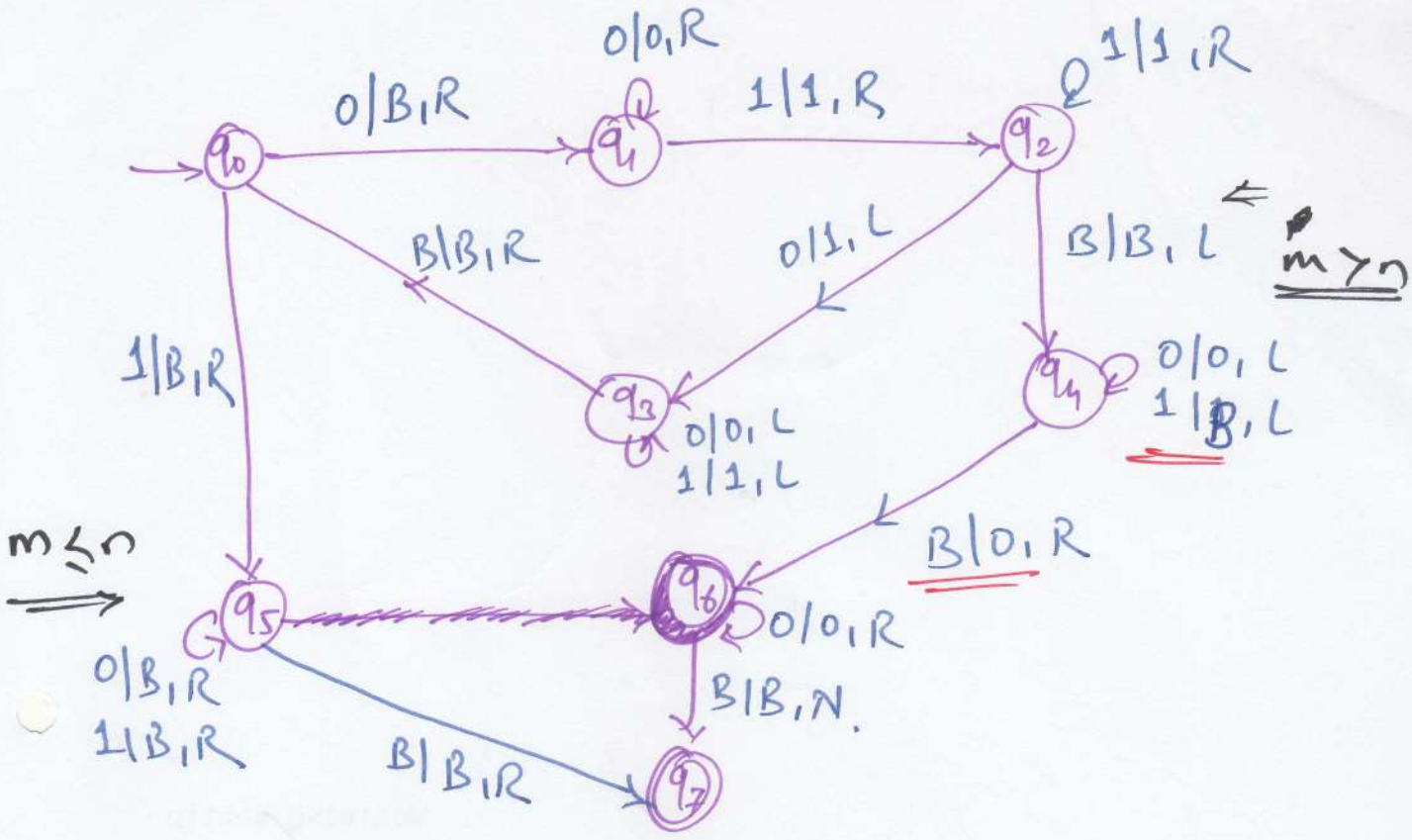
(23) Design a TM to compute proper subtraction of two unary nos. The function f is:-

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

$\Rightarrow$ TM starts with tape content as $0^m 1 0^n$

After processing TM halts with 0^{m-n} on its tape.

first no. (m) > second no. (n)



① $3-2=1=0$

0 0 0 1 0 0 B
B 0 0 1 0 0 B
B 0 0 1 1 0 B
B B 0 1 1 1 B
B B B 1 1 1 B
B B 0 B B B B ✓

② $4-2=2$

0 0 0 0 1 0 0 B
B 0 0 0 1 1 0 B
B B 0 0 1 1 1 B
B B B 0 1 1 1 B
B B 0 0 B B B B

③ $2-3=-1$

0 0 1 0 0 0 B
B 0 1 1 0 0 B
B B 1 1 1 0 B
B B B B B B B