Example 1: Design TM to check whether a string over {a, b} contains equal number of a's and b's. [Dec-2008, Dec-2009, May-2010, May-2012, May-2014]

Sol^n :⇒

Step 1: Logic:

1. Locate first 'a' or first 'b'
2. If it is 'a' then locate 'b' rewrite them as x.
3. If it is 'b' then locate 'a' rewrite them as x
4. Repeat steps from 1 to 3 till every a or b is rewritten as x.

Step 2: Transition diagram

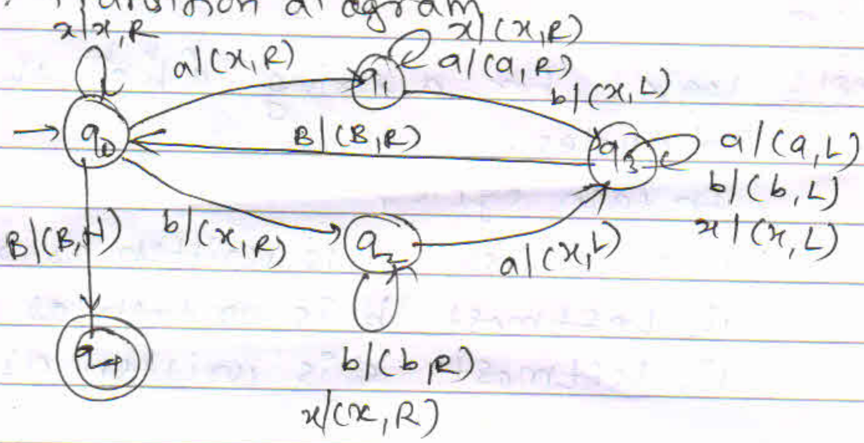


Fig 1: Equal number of a's & b's

Example 2: Construct TM for checking well formedness of parenthesis. [Dec-2007, May-2008, May-2009, Dec-2014]

Sol^n :⇒

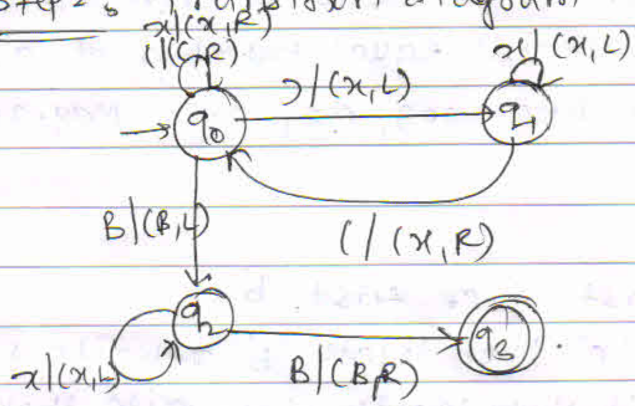
Step 1: Logic:

In each cycle, the left most ')' is written as X then the head moves left to locate the nearer '(' and it is changed to X.

The cycle of computation are shown below:

Input string is assumed to be B(()())()B

Step 2: Transition diagram



Example 3: Design TM which recognizes words of the form $a^n b^n c^n \mid n \geq 1$ [Dec-2009, Dec-2011, Dec-2013]

Sol^n: ⇒

Step 1: Logic: For a string $a^n b^n c^n$ the TM will need 'n' cycles.

In each cycle,

i. Leftmost 'a' is written as x

ii. Leftmost 'b' is written as y

iii. Leftmost 'c' is written as z.

Step 2: Construct transition diagram

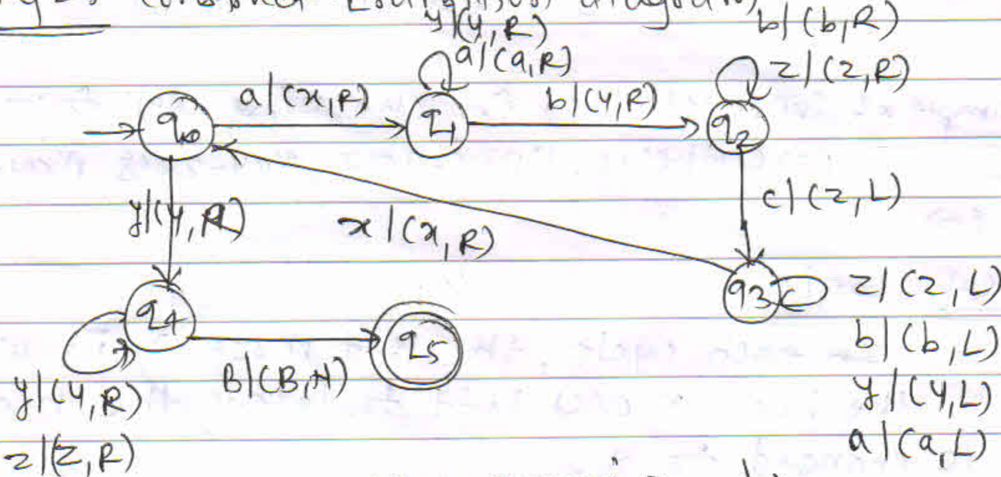


fig1: Transition diagram for $a^n b^n c^n$ where $n \geq 1$.

Example 4: Design TM which recognizes palindromes over alphabet {a,b} [May-2008, Dec-2009, May-2010, May-2011, Dec-2013, May-2014]

Soln:⇒

step1: A palindrome can have one of the following forms:

i. ww^R

ii. waw^R

iii. wbw^R

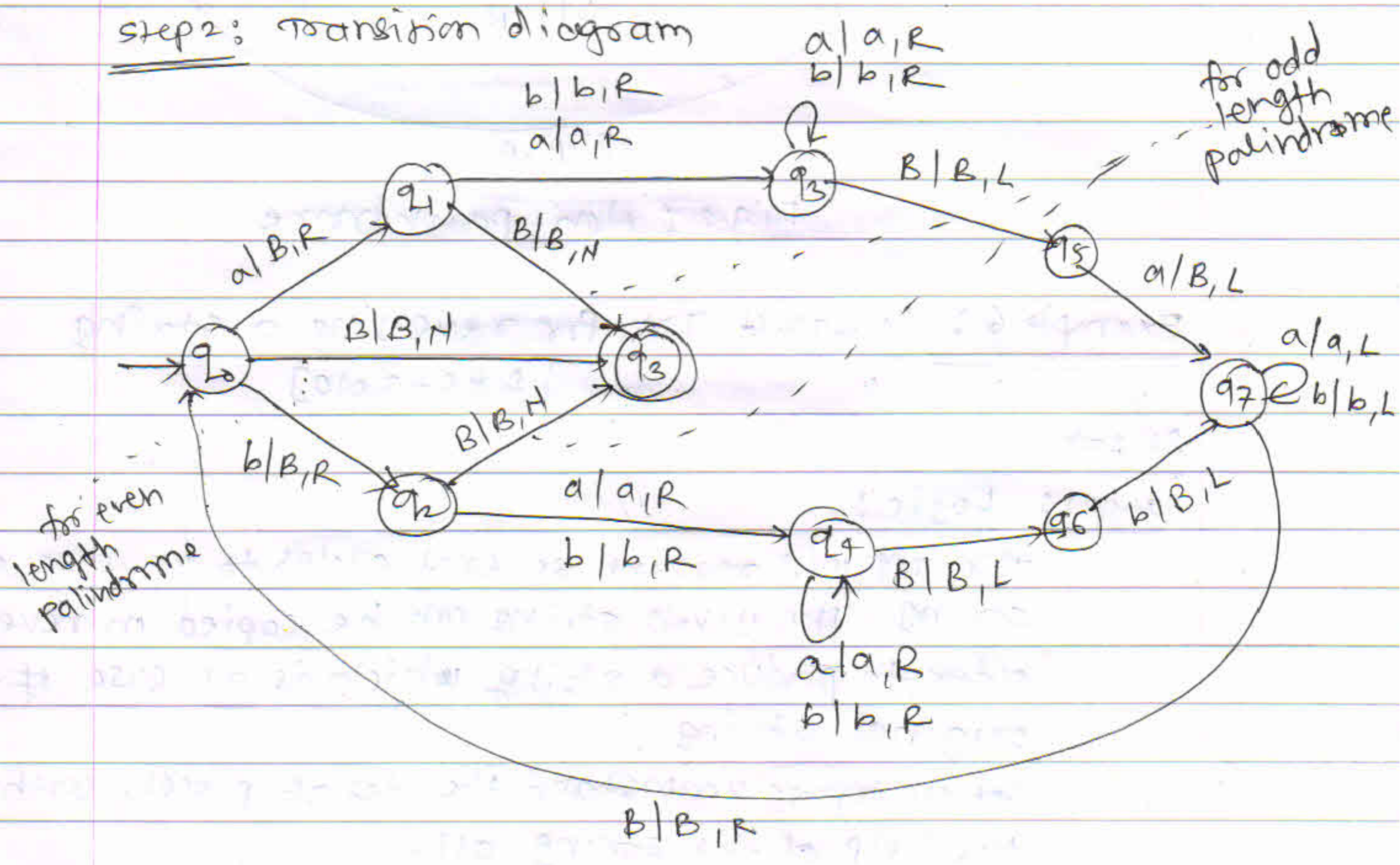
where, w is a string over {a,b} with $|w| \geq 0$

Logic:

i. Algorithm requires 'n' cycles where $|w| = n$

ii. In each cycle, first character is matched with the last character and both are erased.

step2: Transition diagram



Example 5: Draw a transition diagram for a TM accepting the following language
L = { The language of all non-palindromes over {a, b} } [Dec-2009]

Soln :⇒

Step 1: If the tape does not contain 'a' in the state q_5 or the tape does not contain 'b' in state q_6 then it is a non-palindrome.

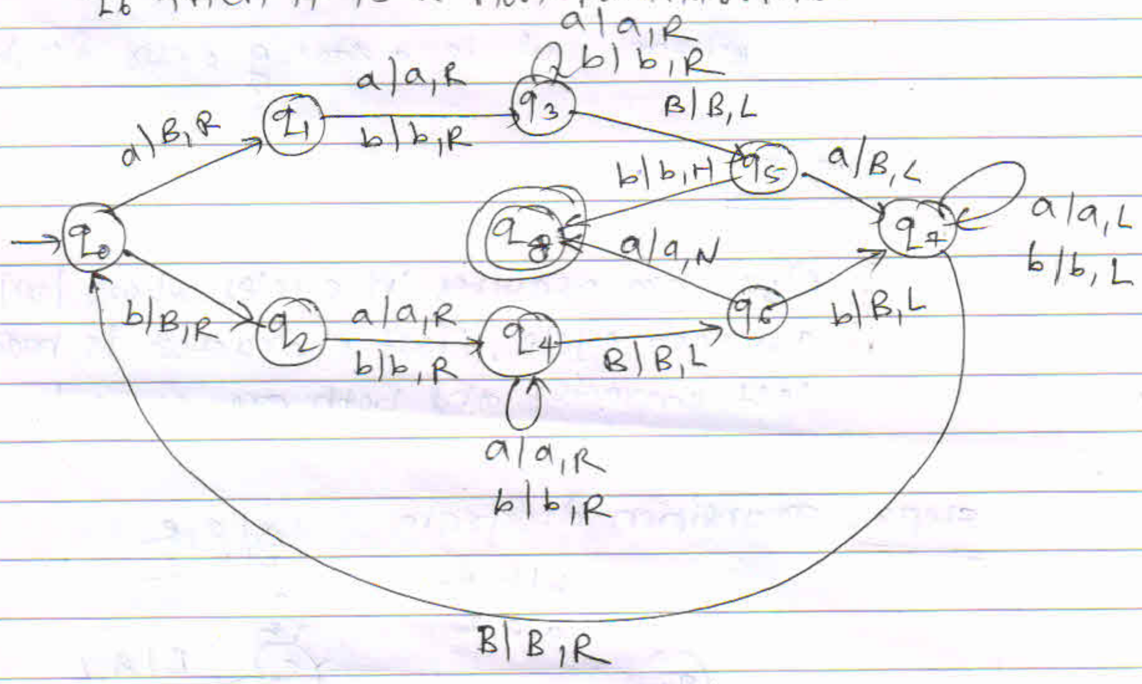


Fig 2: Non-palindrome

Example 6: Construct TM for reversing a string [Dec-2010]

Soln :⇒

Step 1: Logic:

The TM will require several cycles to reverse a string. The given string can be copied in reverse order to produce a string which is reverse of the original string.

Let us try to understand the design process with the help of the string 011.

Original string : B011B

After 1^{st} cycle : B01X1B

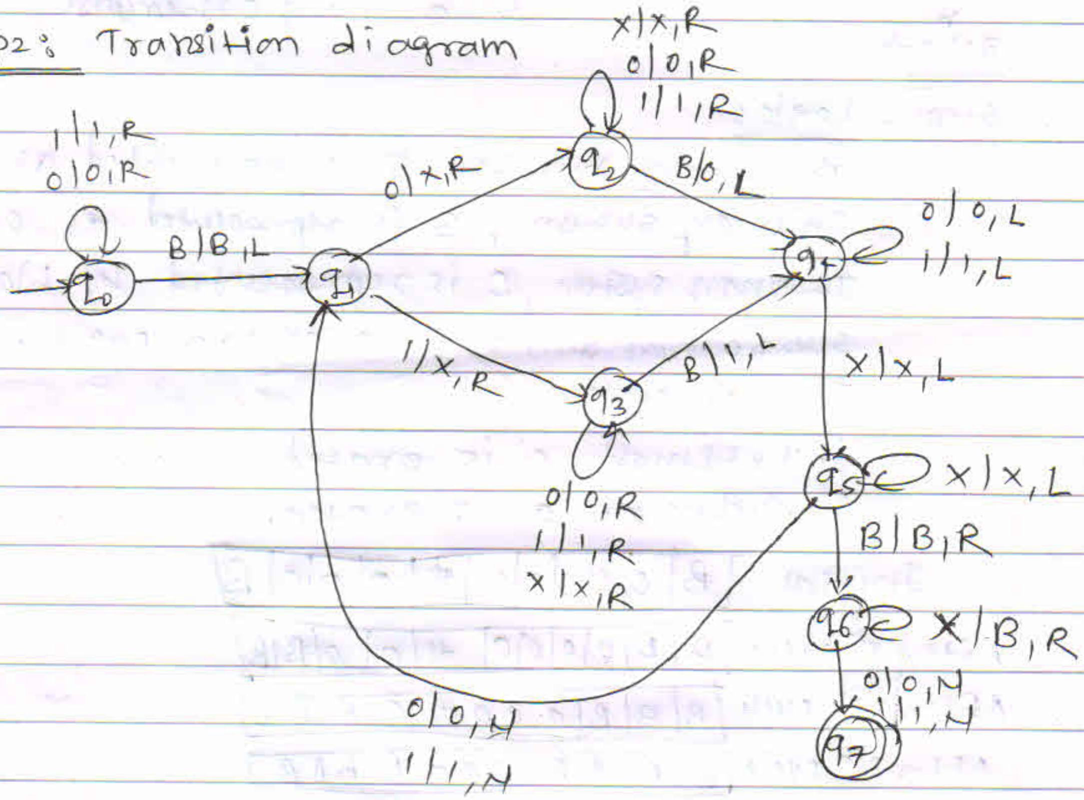
After 2^{nd} cycle : B0XX11B

After 3^{rd} Cycle : BXXX110B

sebsequently, x's can be erased.

Initially, the head is positioned on the left most symbol. The head is advanced to the right most symbol in the state q_1.

Step2: Transition diagram



Example 7: Design TM to increment a binary number by 1.

Sol^n :⇒

Step1: Logic:

Binary Number : B 100111 B

$$\begin{array}{r} B\ 100111\ B \\ +1 \\ \hline B\ 101000\ B \end{array}$$

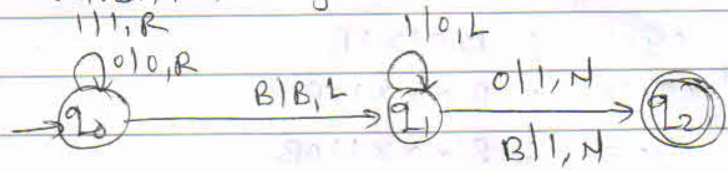
Read number from Left to right ~~upto~~ blank comes. Then

i. if LSB = 0 ~~add~~ put 1 at LSB place and stop at final state

ii. ~~if~~ until '1' comes from LSB side put 0 at that place and if blank occurs put 1 at that place stop at final state

iii. while '1' comes from LSB side put 0 if '0' occurs put '1' and stop process at final state.

step2: Transition diagram



Example 8: Design TM to compute proper subtraction of two unary numbers. The proper subtraction function f is defined as follows:

$$f(m,n) = \begin{cases} m-n & ; \text{if } m>n \\ 0 & ; \text{otherwise} \end{cases}$$ [Dec-2008, May-2013]

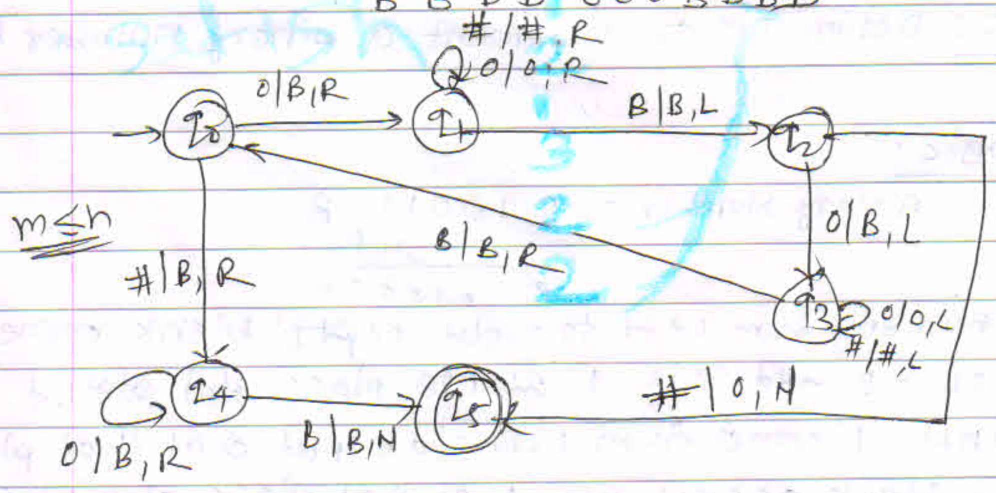
Sol^n:→

Step1: Logic:

The unary number 5 is represented as 00000
In unary system, 3 is represented as 000
In unary system, 0 is represented by blank tape.
Subtraction will require several cycle.
In each cycle:
i. Leftmost '0' is erased.
ii. Rightmost '0' is erased.

Initial | B | 0 | 0 | 0 | 0 | 0 | # | 0 | 0 | 0 | B |
After 1st cycle | B | B | 0 | 0 | 0 | 0 | # | 0 | 0 | B | B |
After 2nd cycle | B | B | B | 0 0 0 # 0 B B | B |
After 3rd cycle | B B B B 0 0 # B B B B |

B B B B 0 0 0 B B B B



Date:
Page:

Example 9: Design TM to find the value of $\log_2(n)$ where 'n' is any binary number.

Sol^n :→

Step 1: Logic:

$\log_2 n$ of any number 'n' lying bet^n 2^n and 2^{n+1} is given by n.

i.e. if $2^n \leq n < 2^{n+1}$ then $\log_2^n = n$

Let us consider the case of a number,

$n = 36$

$2^5 \leq 36 < 2^6$

$\therefore \log_2(36) = 5$

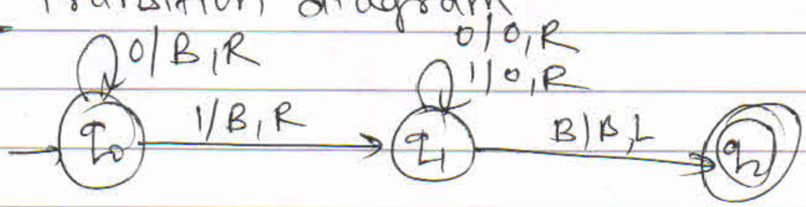
36 can be written as 100100 in binary

- Any number 'n' satisfying the condition $2^5 \leq n < 2^6$ can be written as 1XXXXX where X stands for either 1 or 0.

- $\log_2(1XXXXX)$ can be calculated by erasing the most significant bit '1' and remaining other bits as '0'

Unary representation of 5 is 00000

Step 2: Transition diagram



Example 1: Design TM to check whether a string over {a, b} contains equal number of a's and b's. [DEC-2008, DEC-2009, May-2010, May-2012, May-2014]

Sol :⇒

Step 1 :

Logic :

B baabbabaabaB

b|(x,R)

B xabbabaabaB

a|(x,L)

B xxbbabaabaB

x|(x,R)

B xxbbabaabaB

x|(x,R)

B xxbbabaabaB

b|(x,R)

B xxxbabaabaB

b|(b,R)

B xxxbabaabaB

a|(x,L)

B xxxbxbaabaB

b|(b,L)

B xxxbxbaabaB

x|(x,R)

B xxxbxbaabaB

b|(x,R)

B xxxxxbaabaB

x|(x,R)

B xxxxxbaabaB

b|(b,R)

B xxxxxbaabaB

a|(x,L)

B xxxxxbxabaB

B xxxxxxxabaB

B xxxxxxxxbaB

B xxxxxxxxxaB

steps:

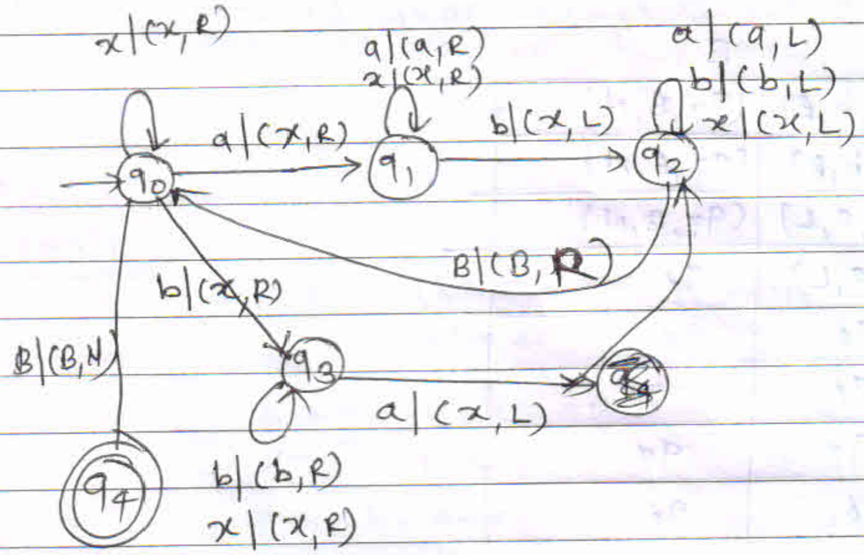
1. Identify first 'a' or first 'b'.

2. If it is 'a' then replace it with x and move to the right in input and identify 'b' and replace 'b' with x and move to left until leftside blank i.e. B comes

3. If it is 'b' then replace it with x and move to the right in input and identify 'a' once it is identified replace it with x and move to left until blank to the left side comes.

4. Repeat step 1 to 3 until ~~right side blank comes.~~ every a & b gets replaced with x.

Step2: construct transition diagram



Step3: Transition Table

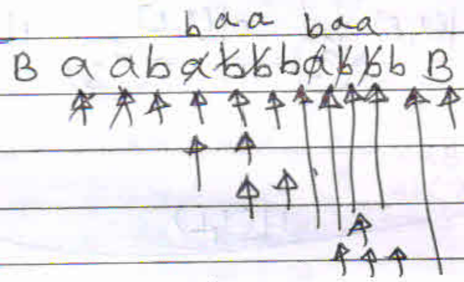
Q \ Σ	a	b	x	B
$\to q_0$	(q_1, x, R)	(q_3, x, R)	(q_0, x, R)	(q_4, B, N)
q_1	(q_1, a, R)	(q_2, x, L)	(q_1, x, R)	$q\phi$
q_2	(q_2, a, L)	(q_2, b, L)	(q_2, x, L)	(q_0, B, R)
q_3	(q_2, x, L)	(q_3, b, R)	(q_3, x, R)	$q\phi$
q_4^*	$q\phi$	$q\phi$	$q\phi$	$q\phi$

Example 2: Design TM to replace every occurrence of abb by baa

Sol:⇒

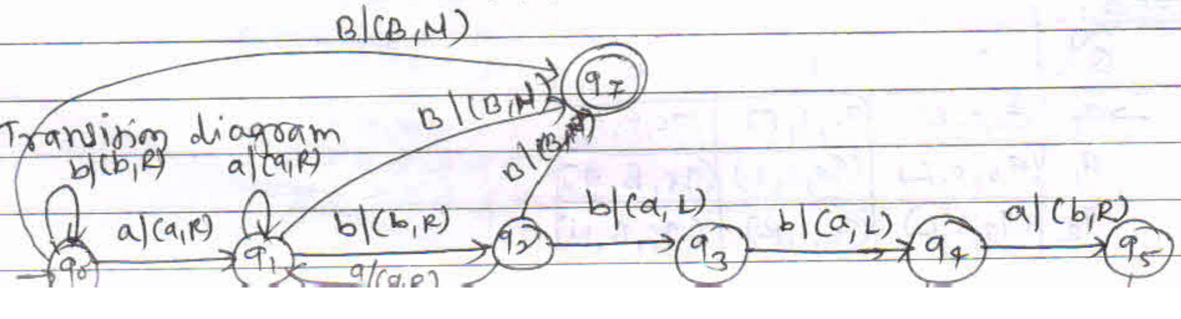
step1:

Logic:



output: B a a b b a a b b a a b b

step2: Transition diagram



step3: Transition Table

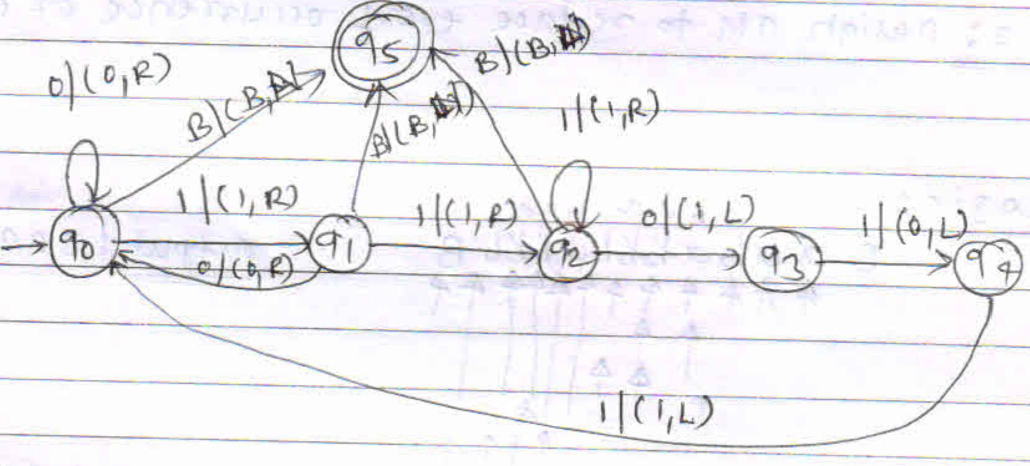
Q \	a	b	B
q_0	(q_1, a, R)	(q_0, b, R)	(q_7, B, H)
q_1	(q_1, a, R)	(q_2, b, R)	(q_7, B, H)
q_2	(q_7, a, R)	(q_3, a, L)	(q_7, B, H)
q_3	q_ϕ	(q_4, a, L)	q_ϕ
q_4	(q_5, b, R)	q_ϕ	q_ϕ
q_5	(q_6, a, R)	q_ϕ	q_ϕ
q_6	(q_0, a, R)	q_ϕ	q_ϕ
q_7	q_ϕ	q_ϕ	q_ϕ

Example 3: Design TM to replace string 110 by 101 in binary input string. [Dec-2006]

n
Sol: →

step1: logic :

step2: construct transition diagram



step3:

Q \	0	1	B
$\rightarrow q_0$	$(q_0, 0, R)$	$(q_1, 1, R)$	(q_5, B, N)
q_1	$(q_0, 0, R)$	$(q_2, 1, R)$	(q_5, B, N)
q_2	$(q_3, 1, L)$	$(q_2, 1, R)$	(q_5, B, N)
q_3	~~$(q_4, 0, L)$~~ q_ϕ	$(q_4, 0, L)$	q_ϕ

Example 4: Design TM which recognizes words of the form $a^n b^n c^n \mid n \geq 1$

[Dec-2009, Dec-2011, Dec-2013]

sol: ⇒

Step 1:

Logic:

B aaabbbcccB

1. Identify leftmost 'a' & replace it by 'x'
2. Then, identify leftmost 'b' and replace it by 'y'
3. then, identify leftmost 'c' and replace it by 'z'.
4. continue above process until ~~all~~ tape contains all ~~symbols~~ symbols replace by x, y, z.

Example: w = BaaabbbcccB

BaaabbbcccB
replace with x

BxaabbbcccB
keep 'a' as it is & move to right i.e. a|(a,R)

BxaabbbcccB
a|(a,R)

BxaabbbcccB
First b, b/(y,R)

BxaaybbcccB

BxaaybbcccB

BxaaybbcccB

BxaaybbzccB

BxaaybbzccB

BxxaybbzccB

BxxayybzccB

BxxayybzzcB

BxxayybzzcB

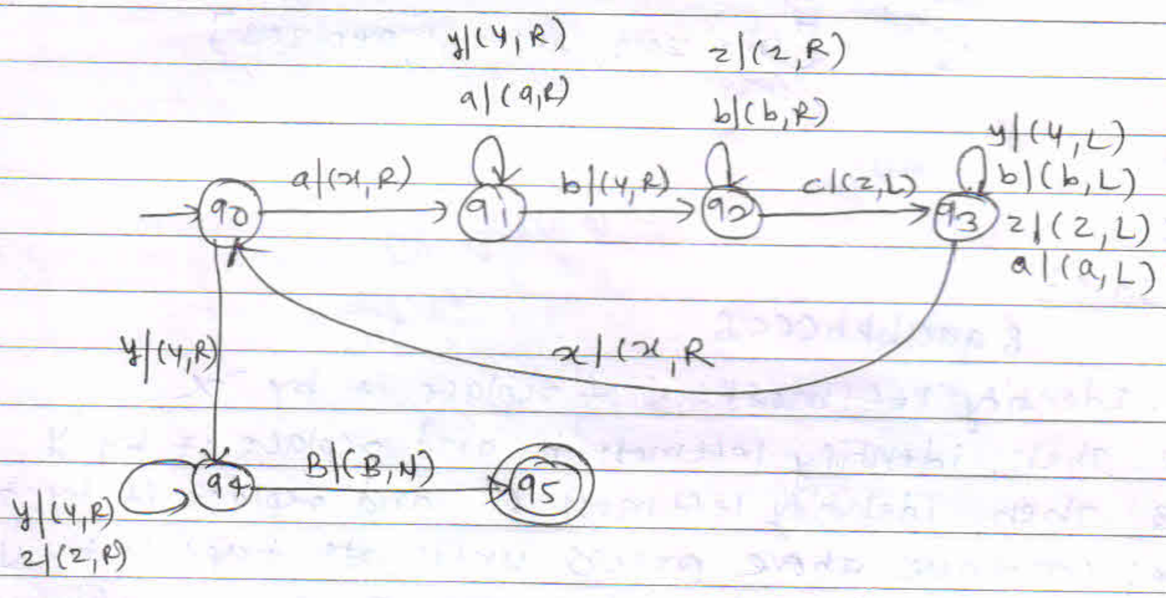
BxxayybzzcB

BxxxyybzzcB

BxxxyyyzzcB

BxxxyyyzzzB

Step2: Transition diagram



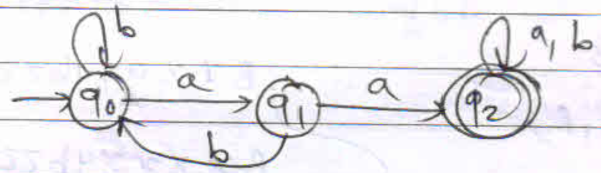
Example 5: Design and write out in full a turning machine that scans to the right until it finds two consecutive a's and then halts over the language of {a,b}.

[May-2009]

Soln:→

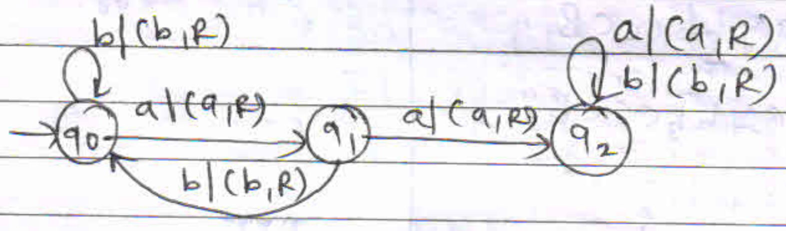
1. Design a DFA to recognize strings with substring aa
2. Design a TM from the DFA.

Step1: DFA



~~Fig 1: DFA for 'aa' as a substring~~

Step2: TM from the DFA



Example 6: Design a TM which recognizes palindromes over $\Sigma \{a, b\}$

[May-2008, Dec-2009, May-2010, May-2011, Dec-2013, May-2014]

Soln :⇒

Step 1:

B B B B B B B

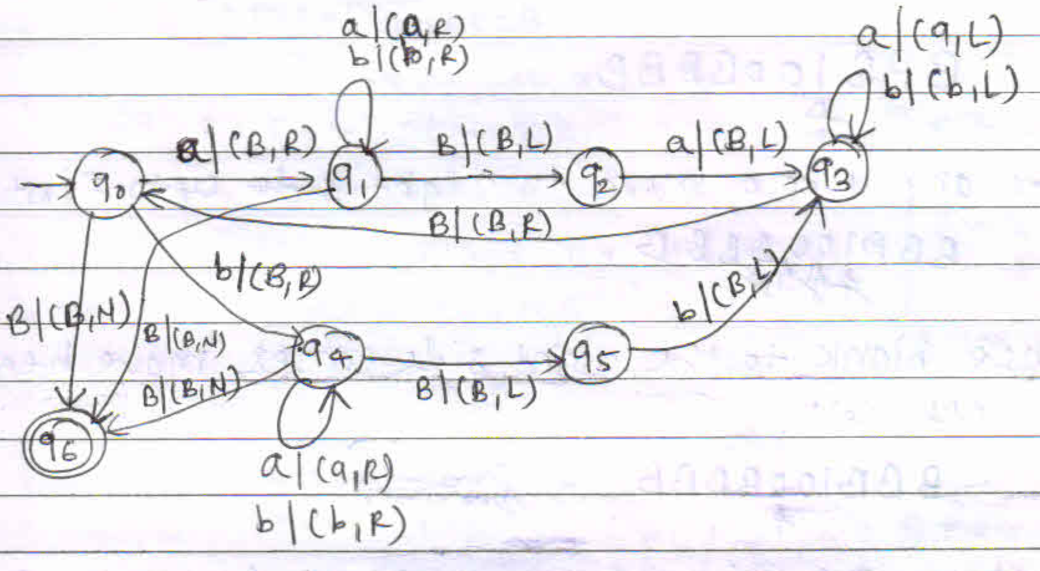
w = B abbabbab B

A plindrome can have one of the following forms:

1. ww^R
2. waw^R
3. wbw^R

where, w is a string over {a, b} with $|w| \geq 0$

Step 2:



Example 7: Draw a transition diagram for a TM accepting the following language.

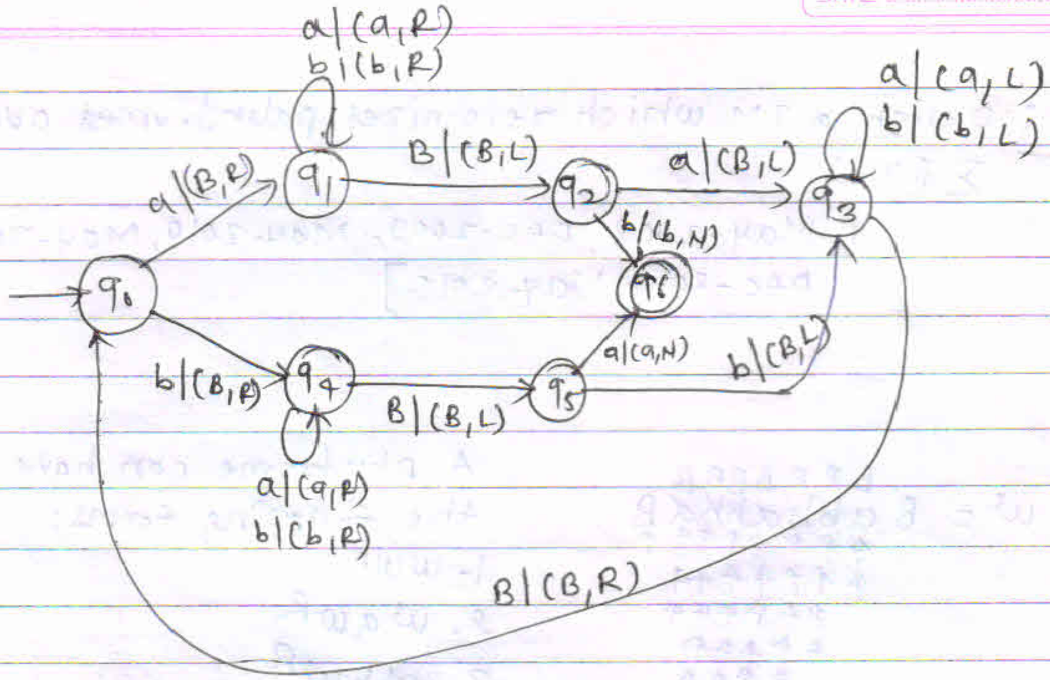
L = {The language of all non palindromes over {a, b}}

[Dec-2009]

Soln :⇒

Step 1: construct TM for palindrome of a string {a, b}

Step 2: Negate the machine by changing final and non final ...



Example 8: construct a TM for reversing a string. [Dec-2010]

Sol:

Step 1:

Logic:

BBB100BBBB

a. for any 1 or 0 move to right ~~side~~ upto first blank

BBB100BBBB

b. once blank to the right side comes, move head to left cell.

BBB100BBBB

c. ~~Replace~~ (i) If '0' replace 0 to X & move to right. BBB10XB

a. If right element is B then replace Blank by 0 and move to left cell.

~~BBB10XB~~

~~BBB~~ BBB10X0BBB

b. If left cell contains X, then keep X as X & move to left continue step © until left side contains B, then keep B|(B,R)

OR

ii. If '1' replace 1 to x & move to right

BBB10xB

a. If right cell is B then replace Blank by 1 and move to left cell

BBB10x1BBB

b. ~~If left cell contains x, then keep x as x & move to left~~

If left cell contains 0, then keep 0/0,L

If left cell contains 1, then keep 1/1,L

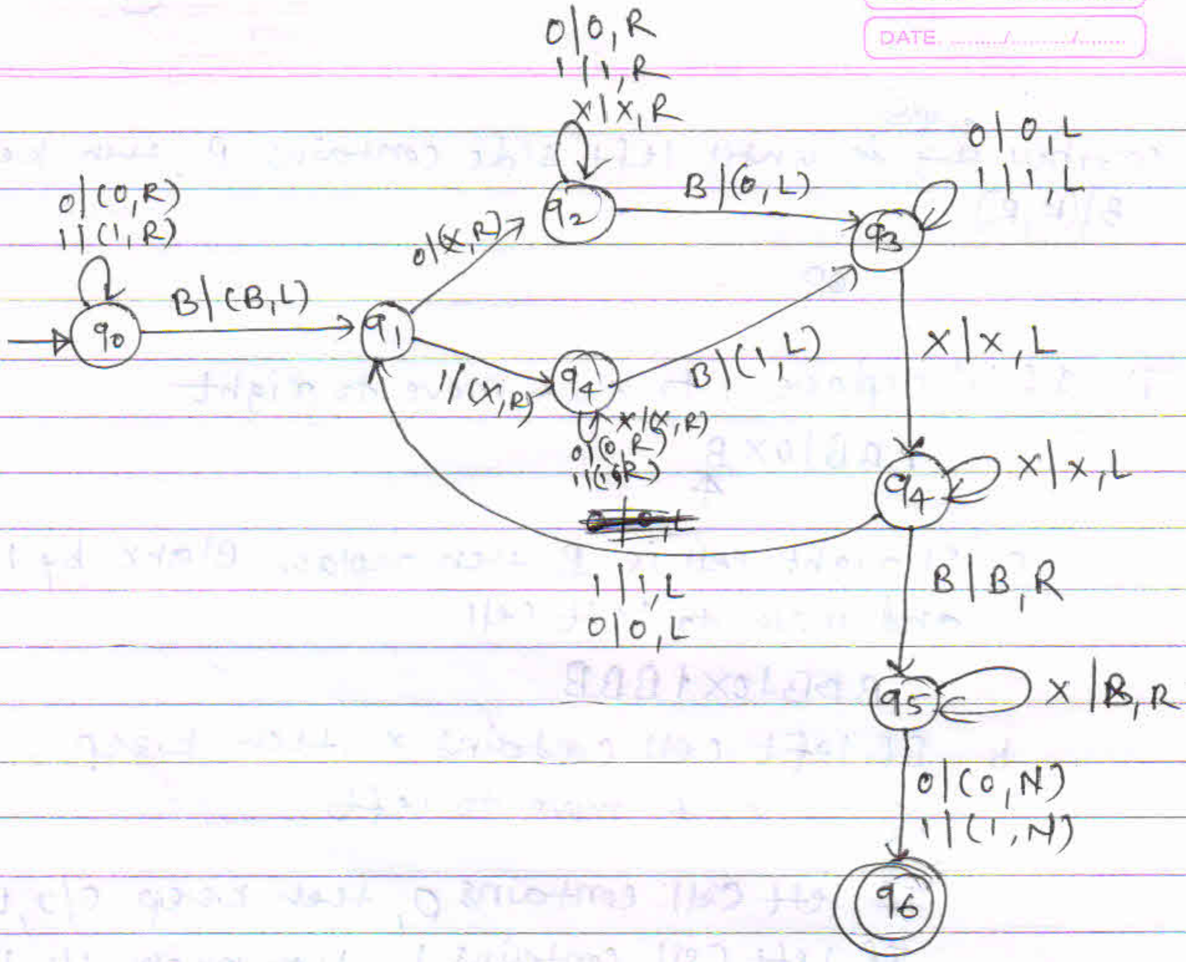
continue step © until left side contains B then keep B/(B,R).

Example 1: BBB100BBBB

2: BBB100BBBB
3: BBB100BBBB
4: BBB100BBBB
5: BBB100BBBB
6: BBB10xBBBB
7: BBB10x0BBB
8: BBB10x0BBB
9: BBB10x0BBB
10: BBB1xx0BBB
11: BBB1xx0BBB
12: BBB1xx0BBB
13: BBB1xx00BB
14: BBB1xx00BB
16: BBB1xx00BB
17: BBB1xx00BB
18: BBB1xx00BB
19: BBBxxx00BB
20: BBBxxx00BB
21: BBBxxx00BB
22: BBBxxx00BB
23: BBBxxx00BB
24: BBBxxx00BB
25: BBBxxx001B
26: BBBxxx001B
27: BBBxxx001B
28: BBBxxx001B
32: BBBBBx001B
33: BBBBBB001B

PAGE NO.

DATE

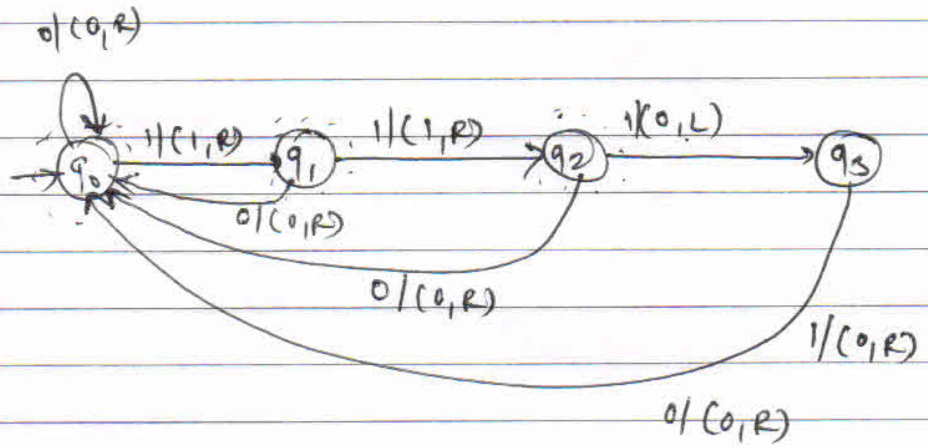

0|0,R
1|1,R
x|x,R
0|(0,R)
1|(1,R)
B|(B,L)
0|(x,R)
B|(0,L)
0|0,L
1|1,L
1|(x,R)
B|(1,L)
x|(x,R)
0|(0,R)
1|(1,R)
x|x,L
x|x,L
1|1,L
0|0,L
B|B,R
x|B,R
0|(0,N)
1|(1,N)
q_0
q_1
q_2
q_3
q_4
q_4
q_5
q_6

Example : Design TM to replace 111 to 100 over $\Sigma = \{0,1\}$

Sol^n :→

Step 1 : Suppose, w = 0010110111011110

Step 2 : Transition diagram



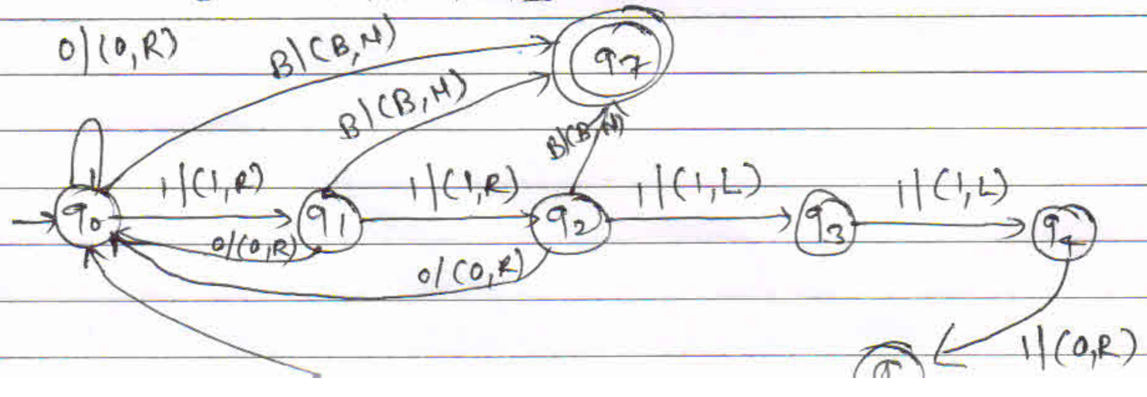
Step 3 : Transition Table

Q\Γ	0	1	B
→ q_0	q_0 0 R	q_1 1 R	Accept
q_1	q_0 0 R	q_2 1 R	Accept
q_2	q_0 0 R	q_3 0 L	REJECT
q_3	q_0 0 R	q_0 0 R	REJECT

Example : Design TM to replace 111 by 011

Sol^n :⇒

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



Example 10 : Construct TM that recognizes the language :

$$L = \{ 0^n 1^m : n, m >= 0 \}$$

[May - 2006]

Soln :→

step1 : Logic :

$$L = \{ \varepsilon, 0, 1, 00, 11, 01, 001, \ldots \}$$

step2 : Construct Transition diagram

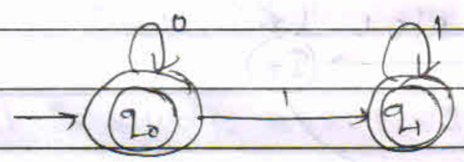


Fig 1 : DFA

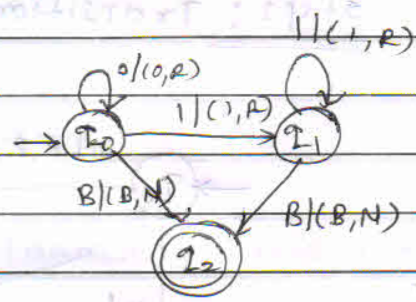


Fig 2 : TM

step3 : Transition ~~Dia~~ Table

Q \ Γ	0	1	B
→q_0	$q_0$0R	$q_1$1R	q_2BN
q_1	-	$q_1$1R	q_2BN
q_2*	-	-	-

Example 11 : Design TM to add two unary numbers.

Soln :→

Step1 :

Logic : Addition of two unary numbers can be performed through append operation.

To add two numbers 5 and 3 .

1. Initial configuration of tape :

$$B\ \underbrace{0\,0\,0\,0\,0}_{\substack{5 \\ \omega_1}}\ \#\ \underbrace{0\,0\,0}_{\substack{3 \\ \omega_2}}\ B$$

2. String w_1 is appended to w_2.

Final Answer: B 00000000 B

while every '0' from w_1 is getting appended to w_2, '0' from w_1 is erased. w_2 contains 8 0's which is sum of 5 and 3.

Step 2: Transition Diagram

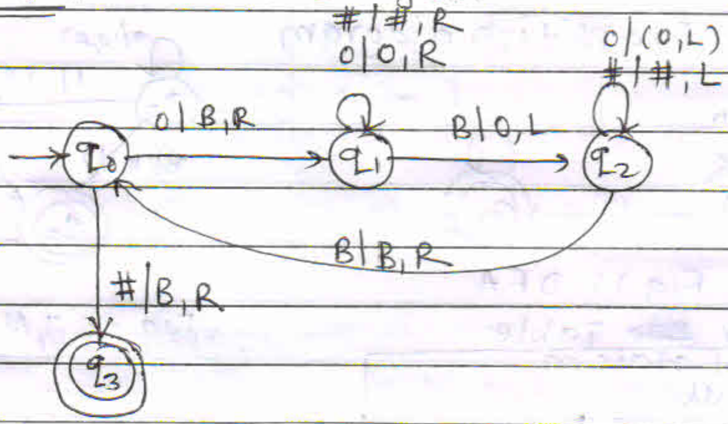


Fig 1: Transition Diagram

Step 3: Transition Table

Q	0	#	B
$\rightarrow q_0$	q_1BR	q_3BR	–
q_1	$q_1 0R$	$q_1\#R$	$q_2 0L$
q_2	$q_2 0L$	$q_2\#L$	q_0BR
q_3^*	–	–	–

← ~~Queue symbols (front symbol in queue)~~