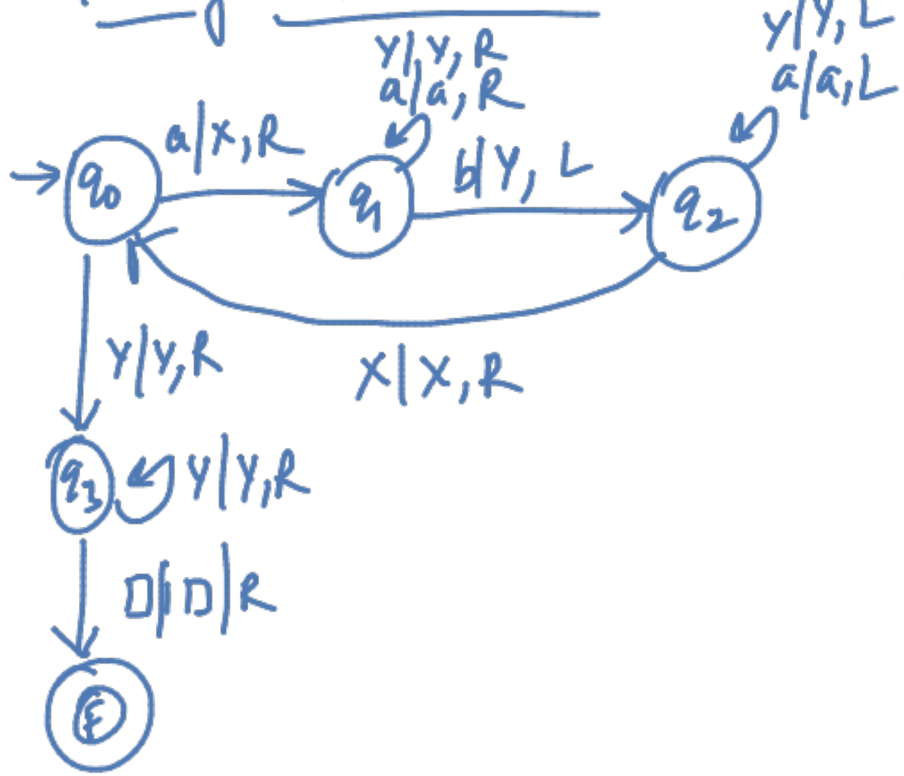
1) $L = \{a^n b^n , n \geq 1\}$

Logic

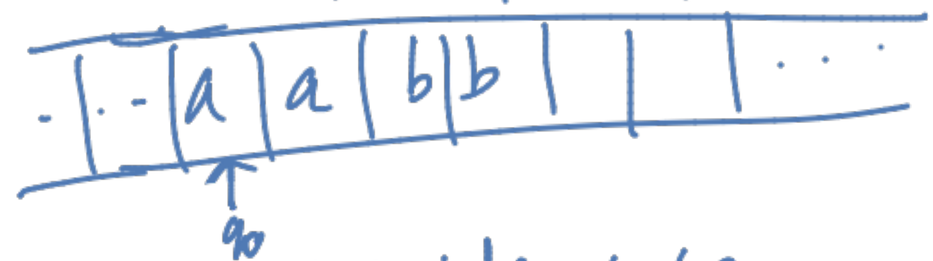
- Match a 'b' for every 'a'
- Mark the matched symbols

Encode
matched 'a' as 'X'
matched 'b' as 'Y'

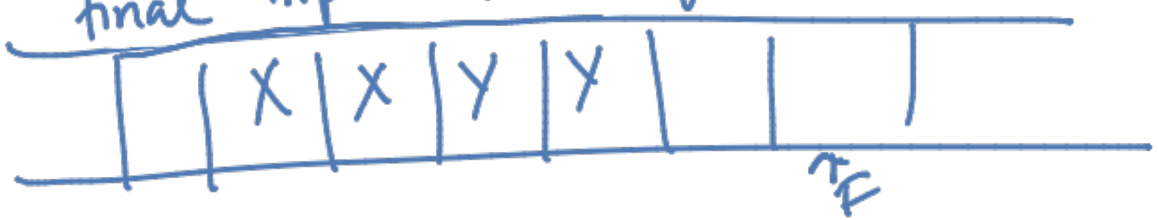
Turing Machine



Initial input tape Conf:-



final input tape Conf:-



Trace aabb

q_0 aabb
$\vdash$ X q_1 abb
$\vdash$ X a q_1 bb
$\vdash$ X q_2 a Y b
$\vdash$ q_2 X a Y b
$\vdash$ X q_0 a Y b
$\vdash$ X X q_1 Y b
$\vdash$ X X Y q_1 b
$\vdash$ X X q_2 Y Y
$\vdash$ X q_2 X Y Y
$\vdash$ X X q_0 Y Y
$\vdash$ X X Y q_3 Y
$\vdash$ X X Y Y q_3 □ □
$\vdash$ X X Y Y □ F □

2) $L = \{a^n b^n c^n , n \geq 1\}$

Logic

Match a 'b' and a 'c' for every a.

Encode all the marked symbols as:-

a - 'X'

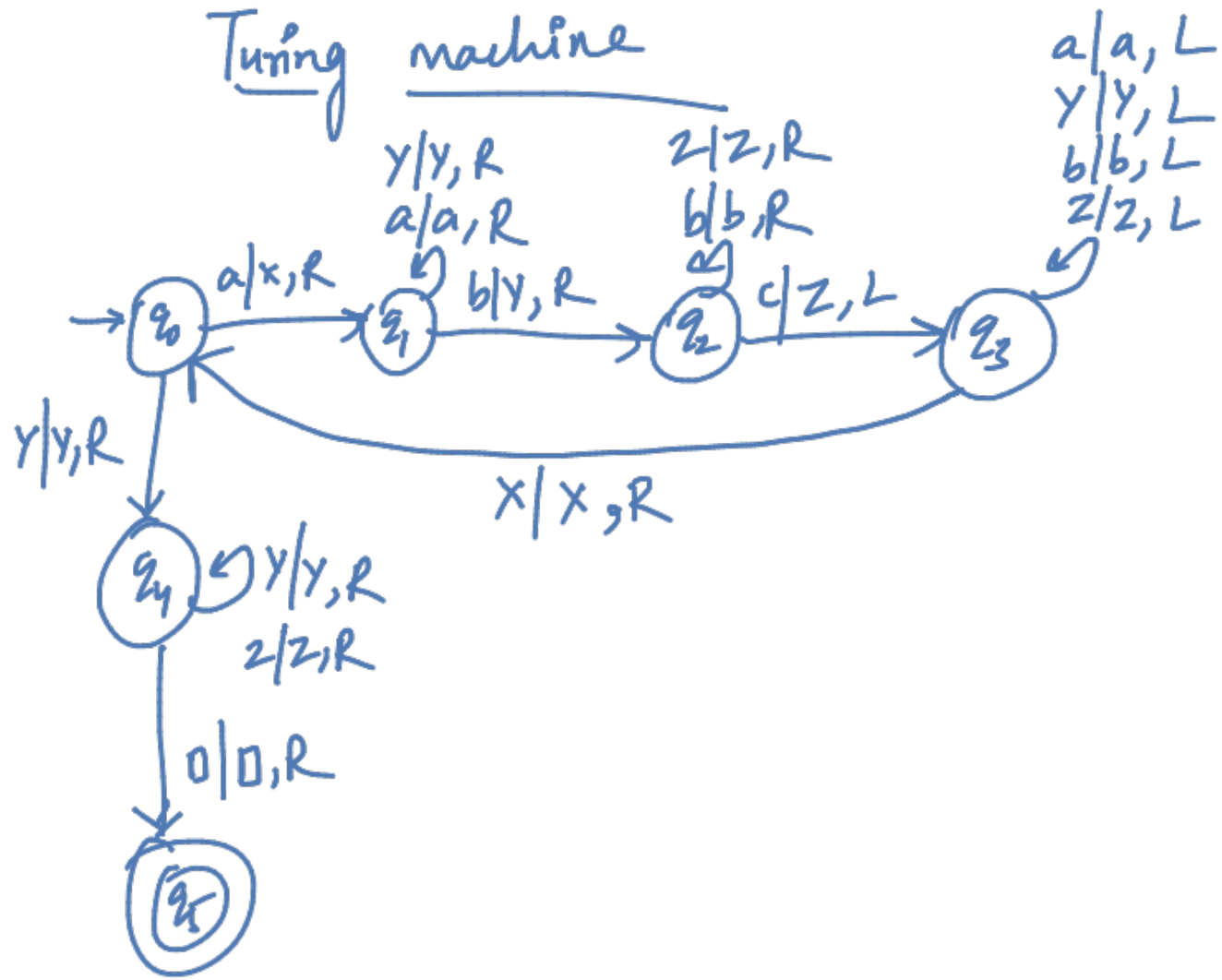
b - 'Y'

c - 'Z'

Input tape

Initial conf: |a|a|b|b|c|c| |

final conf: |X|X|Y|Y|Z|Z| |

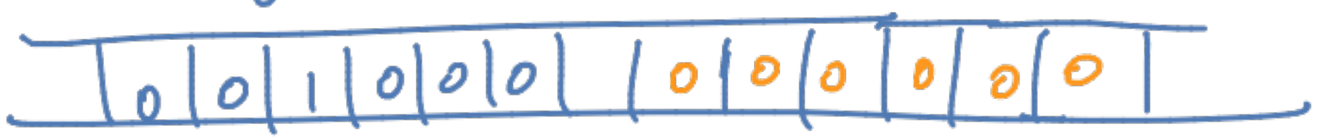


3) Construct a Turing Machine for the language of multiplication

Input tape

Initial conf:

final conf:



Given input:-

$0^m 1 0^n$

produce output:-

0^{m*n}

Logic :-

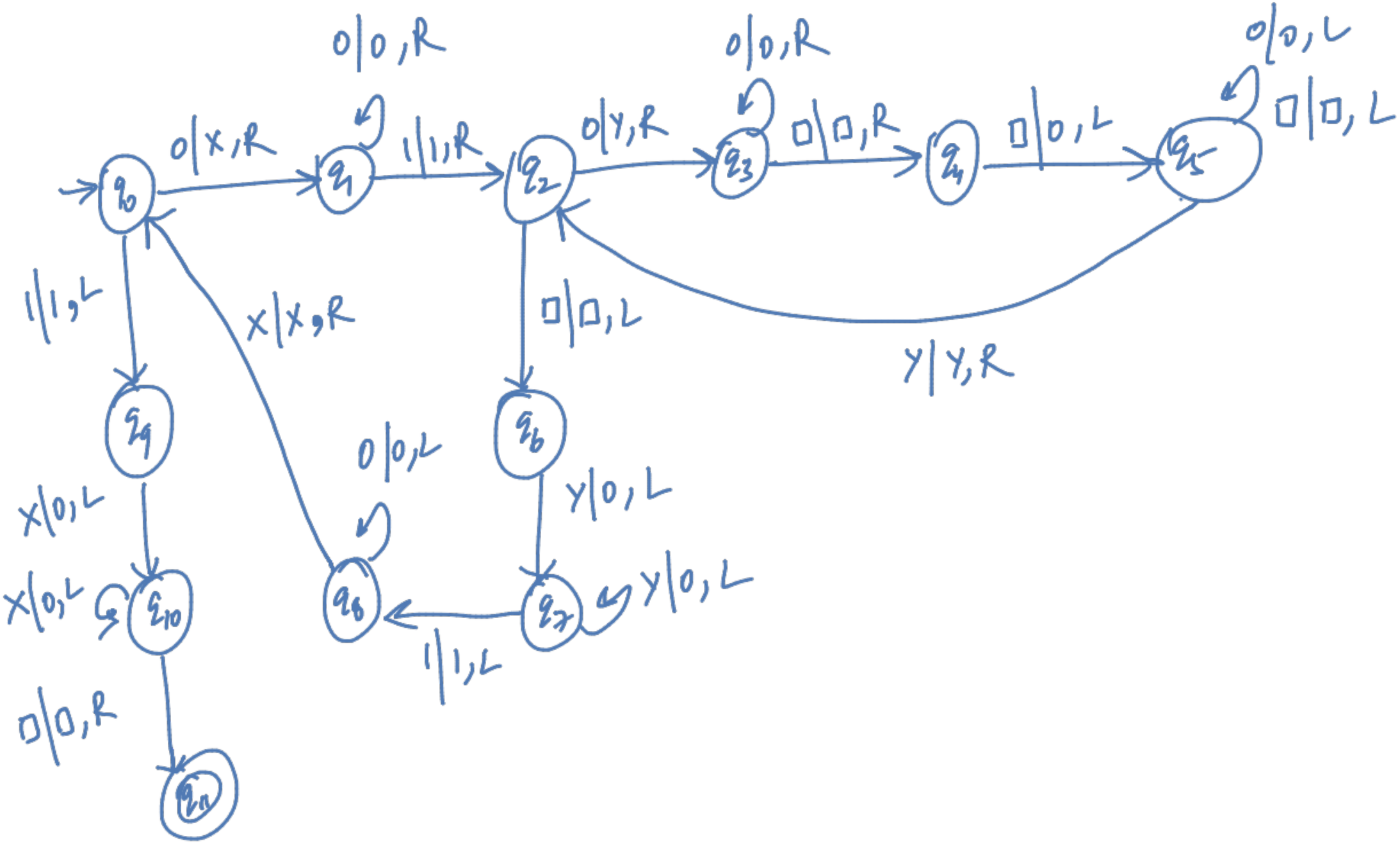
Repeat n (m times)

3) Continued...

Input tape

Time																
0		0	0	1	0	0	0									→ Initial conf.
1		X	0	1	0	0	0									
2		X	0	1	Y	0	0		0							
3		X	0	1	Y	Y	0		0	0						
4		X	0	1	Y	Y	Y		0	0	0					
5		X	0	1	0	0	0		0	0	0					
6		X	X	1	0	0	0		0	0	0					
7		X	X	1	Y	0	0		0	0	0	0				
8		X	X	1	Y	Y	0		0	0	0	0	0			
9		X	X	1	Y	Y	Y		0	0	0	0	0	0		
10		X	X	1	0	0	0		0	0	0	0	0	0		
11		0	0	1	0	0	0		0	0	0	0	0	0		→ final conf.

3) Continued . . .


0|0,R
0|0,R
0|0,L
0|X,R
1|1,R
0|Y,R
□|□,R
□|□,L
□|□,L
1|1,L
X|X,R
□|□,L
Y|Y,R
0|0,L
Y|0,L
X|0,L
X|0,L
Y|0,L
1|1,L
□|0,R

4) $L = \{ww, w \in \{a,b\}^+\}$

ww is an even length string.

Two main steps :-

1) Find the mid point :- Break the even length string into 2 equal parts.

for example :- if the input is aabaab

output should be aab□aab

2) Match w □ w

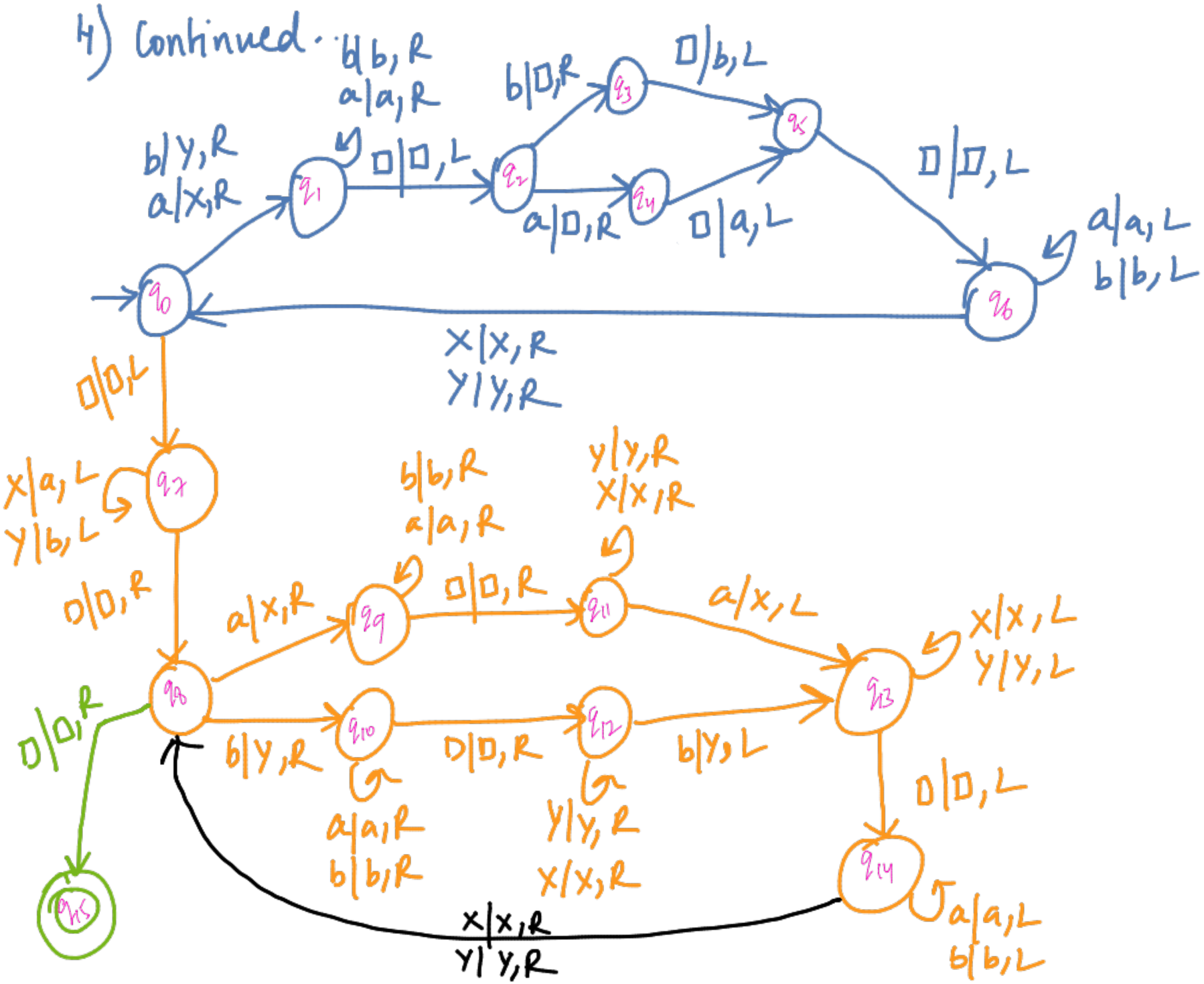
4) Continued....

	a	a	b	a	a	b			
	X	a	b	a	a		b		
	X	X	b	a		a	b		
	X	X	y		a	a	b		
	a	a	b		a	a	b		
	X	a	b		X	a	b		
	X	X	b		X	X	b		
	X	X	y		X	X	y		

Divides even length string into 2 equal halves

matches w□w

4) Continued...


b|b,R
a|a,R
b|□,R
□|b,L
b|y,R
a|x,R
□|□,L
a|□,R
□|a,L
□|□,L
a|a,L
b|b,L
X|X,R
Y|Y,R
□|□,L
x|a,L
y|b,L
□|□,R
a|x,R
□|□,R
a|x,L
x|x,L
y|y,L
□|□,R
b|y,R
□|□,R
b|y,L
□|□,L
a|a,R
b|b,R
y|y,R
x|x,R
a|a,L
b|b,L
x|x,R
y|y,R
q0
q1
q2
q3
q4
q5
q6
q7
q8
q9
q10
q11
q12
q13
q14

5) $L = \{0^n 1^{n^2}, n \geq 1\}$

Logic :-

↳ We mark the first 0 as Z and check for 'n' copies of '1'.

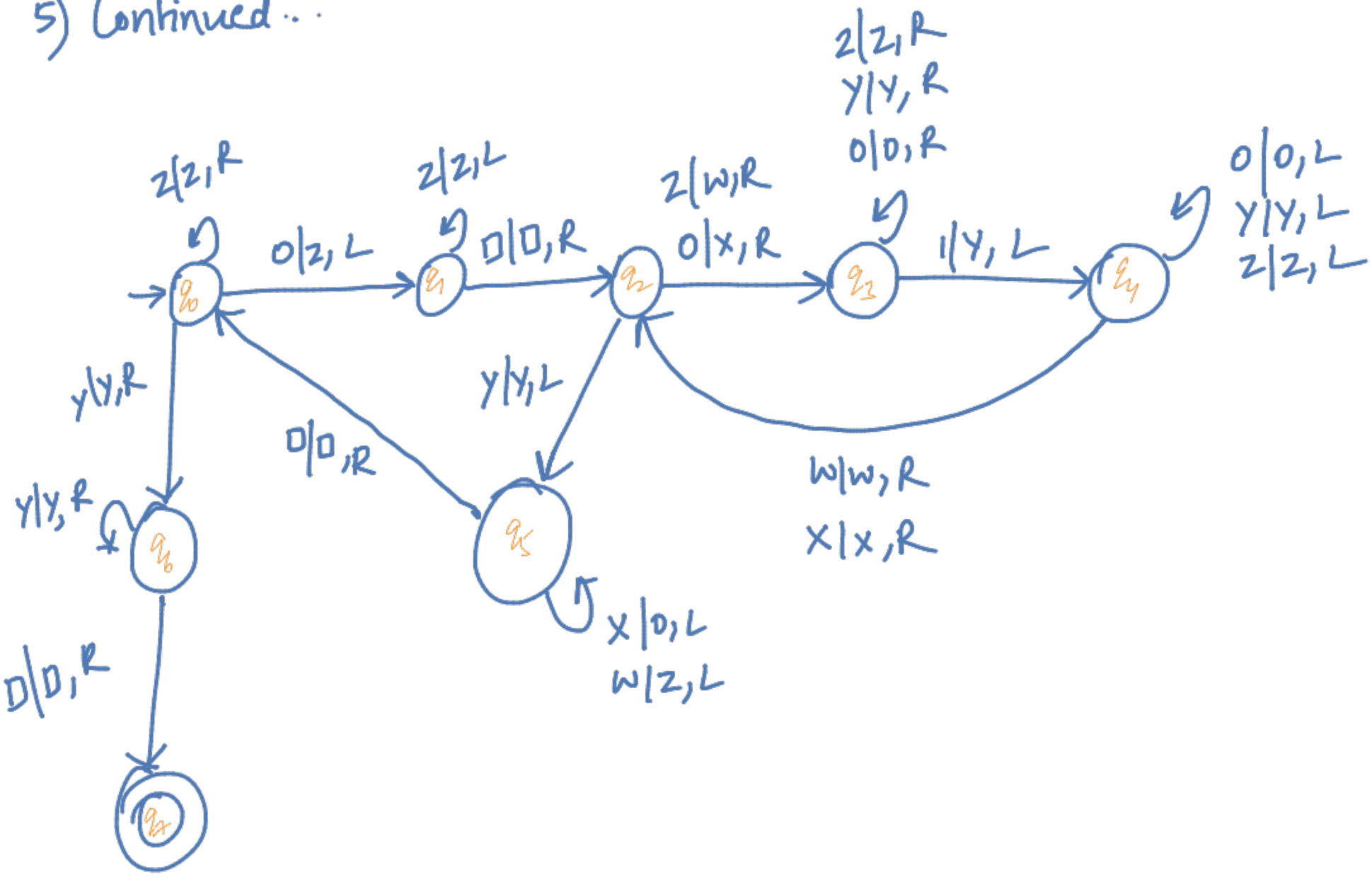
↳ Second 0 is then marked as Z and 'n' copies of '1' are checked.

↳ We repeat the above process until n times n copies of '1' are checked.

5) Continued....

		O	O	O	I	I	I	I	I	I	I	I	I	
		Z	O	O	I	I	I	I	I	I	I	I	I	
		W	O	O	Y	I	I	I	I	I	I	I	I	
		W	X	O	Y	Y	I	I	I	I	I	I	I	
		W	X	X	Y	Y	Y	I	I	I	I	I	I	
		Z	O	O	Y	Y	Y	I	I	I	I	I	I	
		Z	Z	O	Y	Y	Y	I	I	I	I	I	I	
		W	Z	O	Y	Y	Y	Y	I	I	I	I	I	
		W	W	O	Y	Y	Y	Y	Y	I	I	I	I	
		W	W	X	Y	Y	Y	Y	Y	Y	I	I	I	
		Z	Z	O	Y	Y	Y	Y	Y	Y	I	I	I	
		Z	Z	Z	Y	Y	Y	Y	Y	Y	I	I	I	
		W	Z	Z	Y	Y	Y	Y	Y	Y	Y	I	I	
		W	W	Z	Y	Y	Y	Y	Y	Y	Y	Y	I	
		W	W	W	Y	Y	Y	Y	Y	Y	Y	Y	Y	
		Z	Z	Z	Y	Y	Y	Y	Y	Y	Y	Y	Y	

5) Continued...


z|z,R
z|z,L
z|w,R
o|x,R
z|z,R
y|y,R
o|o,R
o|o,L
y|y,L
z|z,L
o|z,L
□|□,R
1|y,L
y|y,R
y|y,L
□|□,R
w|w,R
x|x,R
y|y,R
x|□,L
w|z,L
□|□,R

6) Language of subtraction $= \{a^n b^m c^K,\ K = n-m,\ n, m \geq 1\}$

Input :- $a^n b^m c^K$

output :- $a^n b^m c^K \sqcap c^K$ if K is positive

$a^n b^m c^K \sqcap -c^K$ if K is negative

Logic :-

↳ Match a's & b's

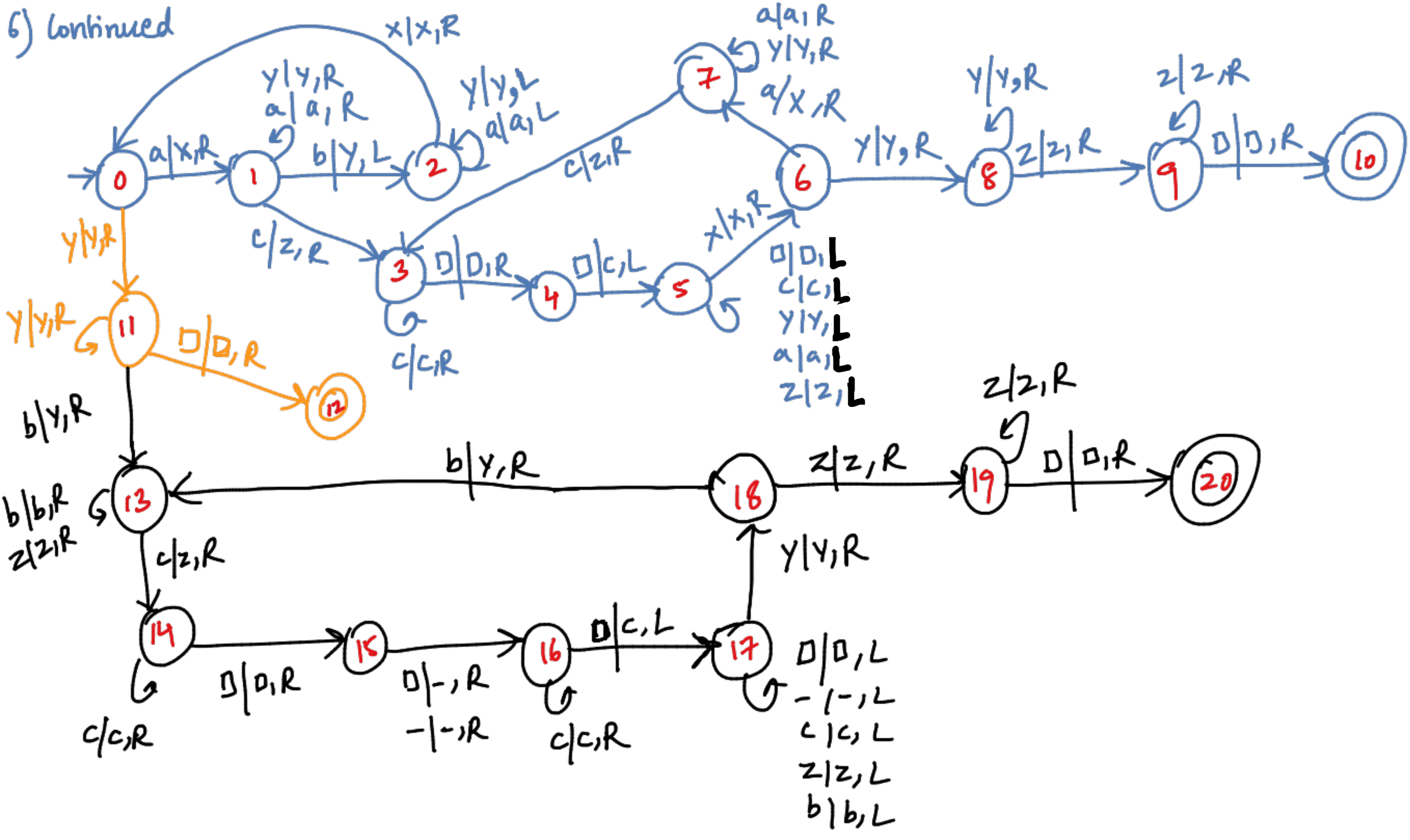
↳ if a's are more it should match with #c's.

• A copy of each c should then be made after the $\sqcap$.

↳ if #b's > #a's, leftover b's must be equal to #c's

• A copy of #c's, preceded with a '-' sign should be made after the blank.

6) Continued


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
a|X,R
b|Y,L
c|Z,R
y|y,R
a|a,R
x|x,R
y|y,L
a|a,L
c|c,R
□|□,R
□|c,L
x|x,R
□|□,L
c|c,L
y|y,L
a|a,L
z|z,L
a|a,R
y|y,R
a|X,R
y|y,R
z|z,R
z|z,R
□|□,R
b|y,R
b|b,R
z|z,R
c|z,R
□|□,R
□|-,R
-|-,R
c|c,R
□|c,L
y|y,R
□|□,L
-|-,L
c|c,L
z|z,L
b|b,L

6) Continued.. $L = \{a^n b^m c^k , k = n-m , n, m \geq 1\}$

Logic

- Match a's & b's. 3 cases are possible :-

  1) #a's > #b's

  2) #a's = #b's

  3) #b's > #a's

- In case ① and ③ we must match leftover a's and b's respectively with the #c's in the string.
- In case #b's > #a's we must write a '_' sign before copying the c's.

7) $L = \{0^{2^n}, n \geq 0\}$

$L = \{0, 00, 0000, 00000000, \ldots\}$

Logic

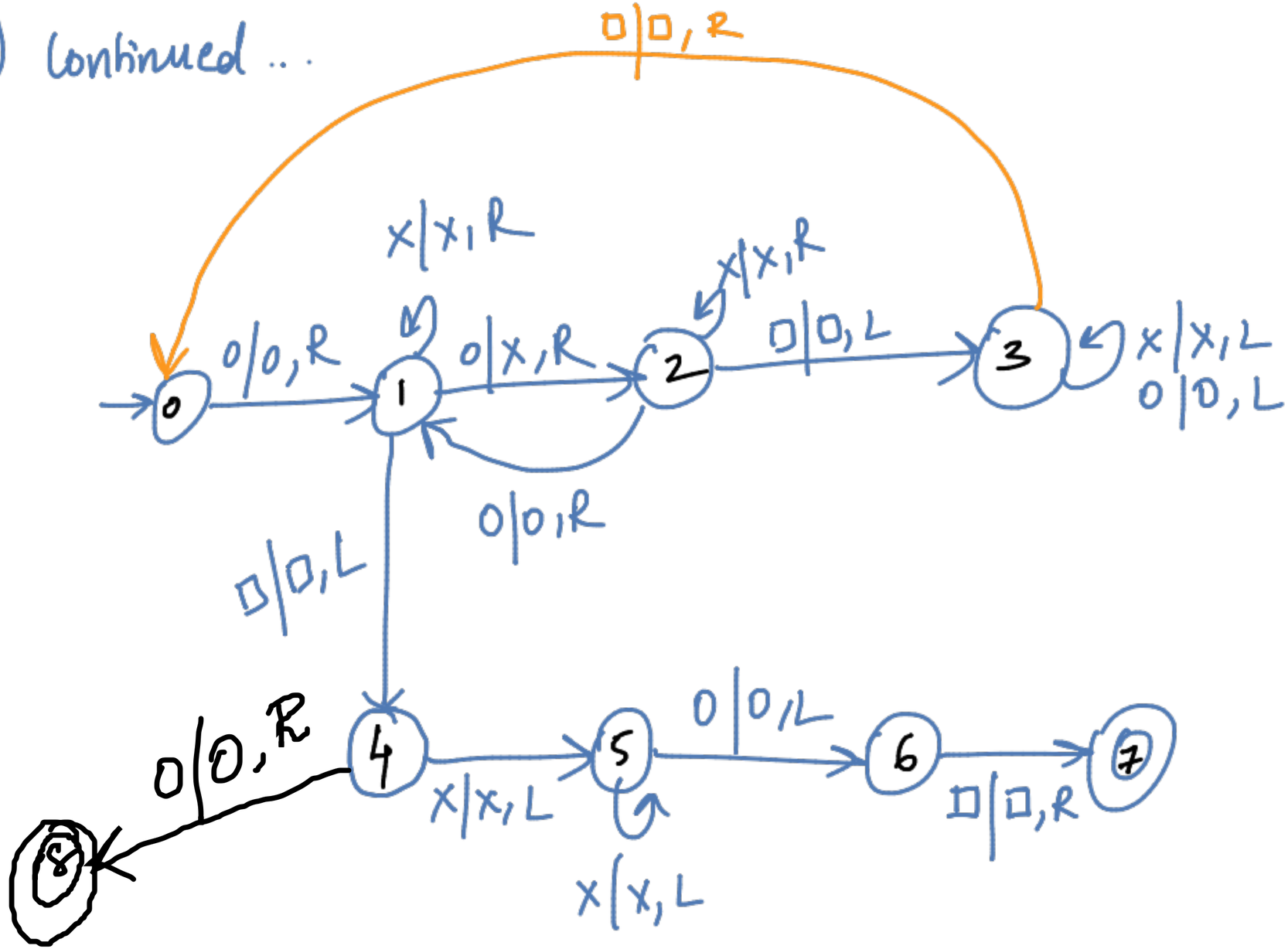
Recursively divide the #0's by 2 until only 1 zero is left..

Example

Mark alternate 0's as X

		#0's
0 0 0 0 0 0 0 0 0		8
0 X 0 X 0 X 0 X		4 (8/2)
0 X X X 0 X X X		2 (4/2)
0 X X X X X X X		1 (2/2) ⇒ accept

7) Continued ...



Acknowledgement - The Notes are prepared by Prof. Preet Kanwal.

8) $L = \{0^{n^2}, n \geq 1\}$ n is a perfect square

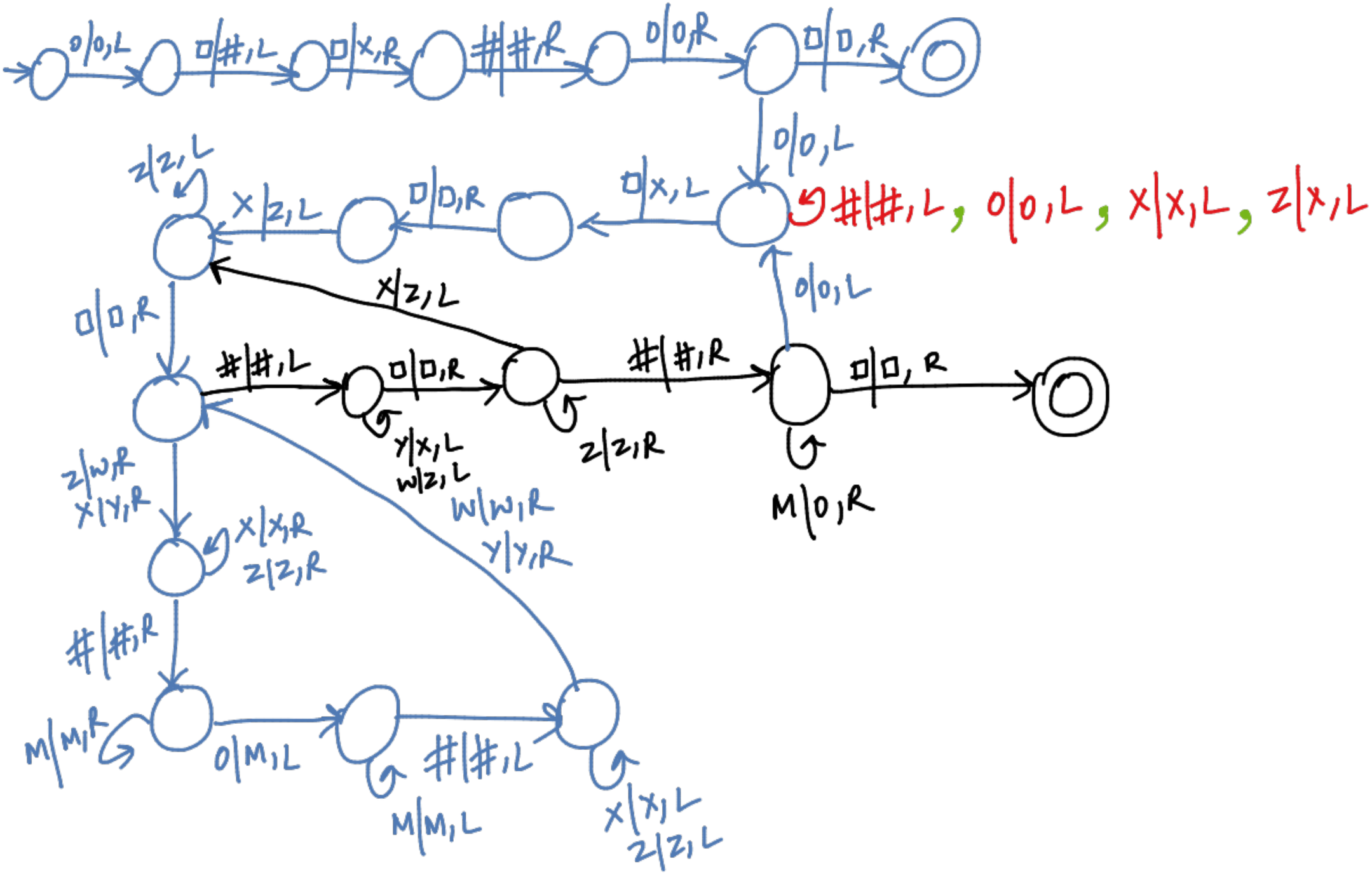
$L = \{0, 0000, 000000000, 0000000000000000, \ldots\}$

$n=1$ $n=2$ $n=3$ $n=4$

Logic :-

- We will have to guess the value of n.
- In order to do that we begin with n=1 by placing an X at the beginning separated by a # from the input.
- In case the value of $n > 1$ we add one more X and make the check.
- Above process is repeated until the machine halts.

8) Continued...


0|0,L
□|#,L
□|X,R
#|#,R
0|0,R
□|□,R
0|0,L
□|X,L
X|Z,L
□|□,R
Z|Z,L
#|#,L, 0|0,L, X|X,L, Z|X,L
X|Z,L
0|0,L
0|□,R
#|#,L
□|□,R
#|#,R
□|□,R
Y|X,L
W|Z,L
Z|Z,R
M|0,R
Z|W,R
X|Y,R
X|X,R
Z|Z,R
W|W,R
Y|Y,R
#|#,R
M|M,R
0|M,L
#|#,L
M|M,L
X|X,L
Z|Z,L

0^4

				0	**0**	**0**	**0**		
			#	**0**	**0**	**0**	**0**		
		X	**#**	**0**	**0**	**0**	**0**		
	X	**X**	**#**	**0**	**0**	**0**	**0**		
	Z	**X**	**#**	**0**	**0**	**0**	**0**		
	W	**X**	**#**	**M**	**0**	**0**	**0**		
	W	**Y**	**#**	**M**	**M**	**0**	**0**		
	Z	**X**	**#**	**M**	**M**	**0**	**0**		
	Z	**Z**	**#**	**M**	**M**	**0**	**0**		
	W	**Z**	**#**	**M**	**M**	**M**	**0**		
	W	**W**	**#**	**M**	**M**	**M**	**M**		
	Z	**Z**	**#**	**0**	**0**	**0**	**0**	**ACCEPT**	

0^6

					0	0	0	0	0	0	
				#	0	0	0	0	0	0	
			X	#	0	0	0	0	0	0	
		X	X	#	0	0	0	0	0	0	
		Z	X	#	0	0	0	0	0	0	
		W	X	#	M	0	0	0	0	0	
		W	Y	#	M	M	0	0	0	0	
		Z	X	#	M	M	0	0	0	0	
		Z	Z	#	M	M	0	0	0	0	
		W	Z	#	M	M	M	0	0	0	
		W	W	#	M	M	M	M	0	0	
		Z	Z	#	0	0	0	0	0	0	
		X	X	#	0	0	0	0	0	0	
	X	X	X	#	0	0	0	0	0	0	
	Z	X	X	#	0	0	0	0	0	0	
	W	X	X	#	M	0	0	0	0	0	
	W	Y	X	#	M	M	0	0	0	0	
	W	Y	Y	#	M	M	M	0	0	0	
	Z	X	X	#	M	M	M	0	0	0	
	Z	Z	X	#	M	M	M	0	0	0	
	W	Z	X	#	M	M	M	M	0	0	
	W	W	X	#	M	M	M	M	M	0	
	W	W	Y	#	M	M	M	M	M	M	
	Z	Z	X	#	M	M	M	M	M	M	
	Z	Z	Z	#	M	M	M	M	M	M	
	W	Z	Z	#	M	M	M	M	M	M	No corresponding 0 to match, hence rejected

0^9

					0	0	0	0	0	0	0	0	0
				#	0	0	0	0	0	0	0	0	0
			X	#	0	0	0	0	0	0	0	0	0
		X	X	#	0	0	0	0	0	0	0	0	0
		Z	X	#	0	0	0	0	0	0	0	0	0
		W	X	#	M	0	0	0	0	0	0	0	0
		W	Y	#	M	M	0	0	0	0	0	0	0
		Z	X	#	M	M	0	0	0	0	0	0	0
		Z	Z	#	M	M	0	0	0	0	0	0	0
		W	Z	#	M	M	M	0	0	0	0	0	0
		W	W	#	M	M	M	M	0	0	0	0	0
		Z	Z	#	0	0	0	0	0	0	0	0	0
		X	X	#	0	0	0	0	0	0	0	0	0
	X	X	X	#	0	0	0	0	0	0	0	0	0
	Z	X	X	#	0	0	0	0	0	0	0	0	0
	W	X	X	#	M	0	0	0	0	0	0	0	0
	W	Y	X	#	M	M	0	0	0	0	0	0	0
	W	Y	Y	#	M	M	M	0	0	0	0	0	0
	Z	X	X	#	M	M	M	0	0	0	0	0	0
	Z	Z	X	#	M	M	M	0	0	0	0	0	0
	W	Z	X	#	M	M	M	M	0	0	0	0	0
	W	W	X	#	M	M	M	M	M	0	0	0	0
	W	W	Y	#	M	M	M	M	M	M	0	0	0
	Z	Z	X	#	M	M	M	M	M	M	0	0	0
	Z	Z	Z	#	M	M	M	M	M	M	0	0	0
	W	Z	Z	#	M	M	M	M	M	M	M	0	0
	W	W	Z	#	M	M	M	M	M	M	M	M	0
	W	W	W	#	M	M	M	M	M	M	M	M	M
	Z	Z	Z	#	M	M	M	M	M	M	M	M	M
	Z	Z	Z	#	0	0	0	0	0	0	0	0	0