# ✳ Theory of Computation (TOC) - Assignment Number - 5

**Name :-** Kaustubh Shrikant Kabra.
**Class :-** Third Year Engineering.
**Div :-** A        **Roll Number :-** 38
**Batch :-** T-2
**Department :-** Computer Department.
**College :-** AISSMS's IOIT.

## ★ Construct Turing Machine Equal no. of a's and equal number of b's.

→ 1] We have to scan the input from left to right.

2] Convert first 'a' and first 'b' in the scanning to 'x', then in the second turn convert second 'a' and second 'b' to 'x', and so on. We have to repeat the process until we convert all a's and b's to 'x'.

3] Character scanned in between 'a' and 'b' will not be changed -
    Let us take a string 'aabb' -
    I) Scan the input from the left.
    II) Our string looks like this -

| B | a | a | b | b | B |
|---|---|---|---|---|---|

4] Now we see that we get out first 'a' at the first position and first b in the third position. We convert these 'a' and 'b' to 'x'.
    Now the character 'a' we get in between 'a' and 'b'. So it will remain the same. When we read our first b we move our pointer to left. The pointer will move to the left until it gets a Blank(B). Now our string look this

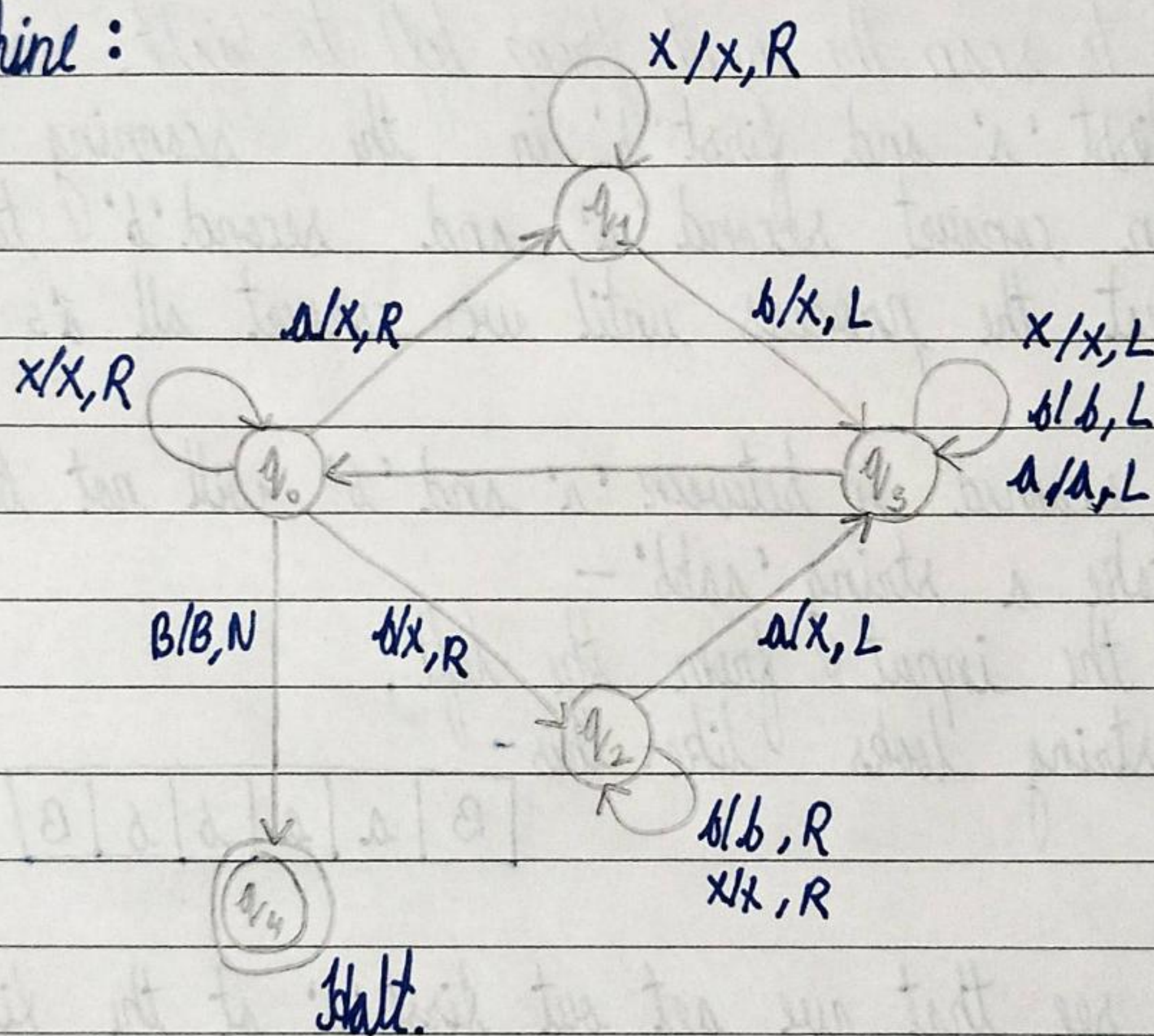| B | x | a | x | b | B |
|---|---|---|---|---|---|

5) Our pointer is at Blank (B). We again scan this input from left to right and convert second 'a' and second 'b' to 'x'. When we read our second b we move our pointer to left. The pointer will move to the left until it gets a Blank (B). Now our string looks like this

| B | X | X | X | X | B |
|---|---|---|---|---|---|

6) We repeat this process untill all a's and b's converted to x.

7) As we see that we convert all a's and b's to 'x', Hence our machine will halt.

Turing Machine :



$x/x, R$

$a/x, R$

$x/x, R$

$b/x, L$

$x/x, L$
$b/b, L$
$a/a, L$

$B/B, N$      $b/x, R$

$a/x, L$

$q_0$  $q_1$  $q_2$  $q_3$  $q_4$

$b/b, R$
$x/x, R$

Halt.

★ Design Turing Machine for 2's complement of binary number.

→ The logic for computing two's complement is, we read the binary string from LSB to MSB. From LSB we keep all the zero's as it is and move left till we do not get 1. After reading first 1 from LSB, we move left and therefore we convert 0 to 1 and 1 to 0 and go moving towards left. The process continues upto leftmost △

For example: The binary number 0110 its two's complement can be computed as:

$$
\begin{array}{r}
0\ 1\ 1\ 0 \\
\hline
1\ 0\ 0\ 1 \\
+\ \ \ \ \ \ 1 \\
\hline
1\ 0\ 1\ 0
\end{array}
$$

1's complement

Adding 1

Two's complement of 0110.

By our idea also we get the same result.

| △ 0 1 1 0 △ | Move to rightmost end upto △ |
| ↑ | |
| △ 0 1 1 0 △ | Move left |
| ↑ | |
| △ 0 1 1 0 △ | It is 0, so keep it is and move left. |
| ↑ | |
| △ 0 1 1 0 △ | First 1 from LSB has encountered so keep it as it is and move left. |
| ↑ | |
| △ 0 1 1 0 △ | complement it to 0 and move left. |
| △ 0 0 1 0 △ | Convert it to 1 and move left |
| △ 1 0 1 0 △ | Since △ is reached, stop 1. |

The TM could be —

Start
$(0, 0, R)$
$(1, 1, R)$

$(0, 0, L)$

$(0, 1, L)$
$(1, 0, L)$

$(\Delta, \Delta, L)$

$(1, 1, L)$

$(\Delta, \Delta, S)$

Halt