|  |  |  |
| --- | --- | --- |
|  | NATIONAL UNIVERSITY  OF COMPUTER & EMERGING SCIENCES  PESHAWAR CAMPUS |  |

**Student Name: Usman Manzoor Roll No: 19P-0068 Sec: 5B**

**Problem Set :** Implement 01 **Semester:** FALL 2021

**Date Set :** Monday December 20, 2021 **Due Date :** Friday Dec.24, 2021

**Course :** CS301 Theory of Automata **Instructor:** Shakir

language {𝑎n𝑏2n𝑐3n | where n belongs to Natural numbers}

Descriptive Definition:

b’s followed by a’s and c’s followed by a’s then b’s are twice times of a’s and number of c’s are thrice times of a’s.

Words:

{abbccc, aabbbbccccccc, aaabbbbbbccccccccc, aaaabbbbbbbbcccccccccccc, aaaaabbbbbbbbbbccccccccccccccc, aaaaaabbbbbbbbbbbbcccccccccccccccccc, aaaaaaabbbbbbbbbbbbbbccccccccccccccccccccc, aaaaaaaabbbbbbbbbbbbbbbbcccccccccccccccccccccccc,...}

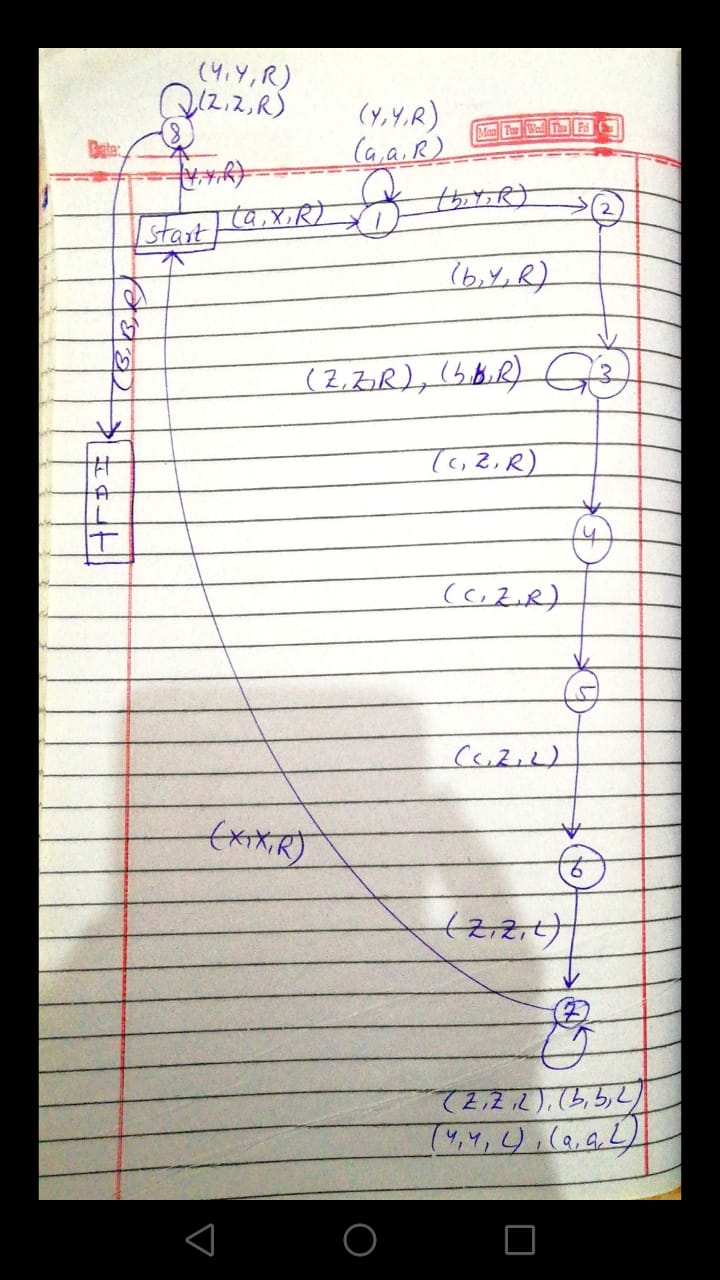
**Report:**

In this Turing Machine I made the Logic that in every go the first starting **a** is replaced with X then two **b’s** replaced with Y and then three starting **c’s** replaced with Z. This happens again and again till all a’s , b’s and c’s will not converted into X , Y and Z.

e.g :

consider the String aabbbbcccccc when it is going to the right it replaces the string in this format XaYYbbZZZccc after replacing third c it moves towards Left and move until it finds X. Now he finds X and again moving towards right the string becomes (B XXYYYYZZZZZZ B) now it is again moving towards left in search of X and if he finds it he checks for **a** if there is no a it means all b’s and c’s are converted into Y and Z Now my tape is moving towards right direction and when It finds B it goes to the **HALT** state.

**DIAGRAM :**



**CODE :**

string = input("Please Enter String: ")

length = len(string) + 2

tape = length \* ['B']

i = 1

tapehead = 1

for s in string: #loop to place string in tape

    tape[i] = s

    i += 1

state = 0

accept = False

def action(input\_char, replace\_with, move, new\_state):

    global tapehead, state

    if tape[tapehead] == input\_char:

        tape[tapehead] = replace\_with

        state = new\_state

        if move == 'L':

            tapehead -= 1

            return True

        elif move == 'R':

            tapehead += 1

            return True

    return False

oldtapehead = -1

while(oldtapehead != tapehead):

    oldtapehead = tapehead

    print(tape , "with tapehead at Node", tapehead, "on state" , state)

    if state == 0:

        if action('a', 'X', 'R', 1) or action('Y', 'Y', 'R', 8):

            pass

    elif state == 1:

        if action('a', 'a', 'R', 1) or action('Y', 'Y', 'R', 1) or action('b','Y','R',2):

            pass

    elif state == 2:

        if action('b', 'Y', 'R', 3):

            pass

    elif state == 3:

        if action('b', 'b', 'R', 3) or action('c', 'Z', 'R', 4) or action('Z', 'Z', 'R', 3):

            pass

    elif state == 4:

        if action('c', 'Z' , 'R', 5):

            pass

    elif state == 5:

        if action('c', 'Z', 'R', 6):

            pass

    elif state == 6:

        if action('c', 'c', 'L', 7):

            pass

    elif state == 7:

        if action('a', 'a' , 'L' , 7 ) or action('Y', 'Y', 'L' ,7) or action('c', 'c' , 'L' , 7) or action('Z', 'Z', 'L' , 7) or action('b', 'b', 'L' , 7) or action('X', 'X', 'R' , 0) :

            pass

    elif state == 8:

        if action('Y', 'Y', 'R', 8) or action('Z', 'Z', 'R', 8) or action('B', 'B', 'R', 9):

            pass

    elif state == 9:

        accept = True

    else:

        accept = True

if accept:

    print("String accepted on state = ", state)

else:

    print("String not accepted on state = ", state)