

Theory of automata

ROLL NO: 19P0012 NAME: AITZAZ TAHIR CH SECTION: BS(CS)-5B Turing machine project



**Language:**

{ww | w ∈ {0,1} }

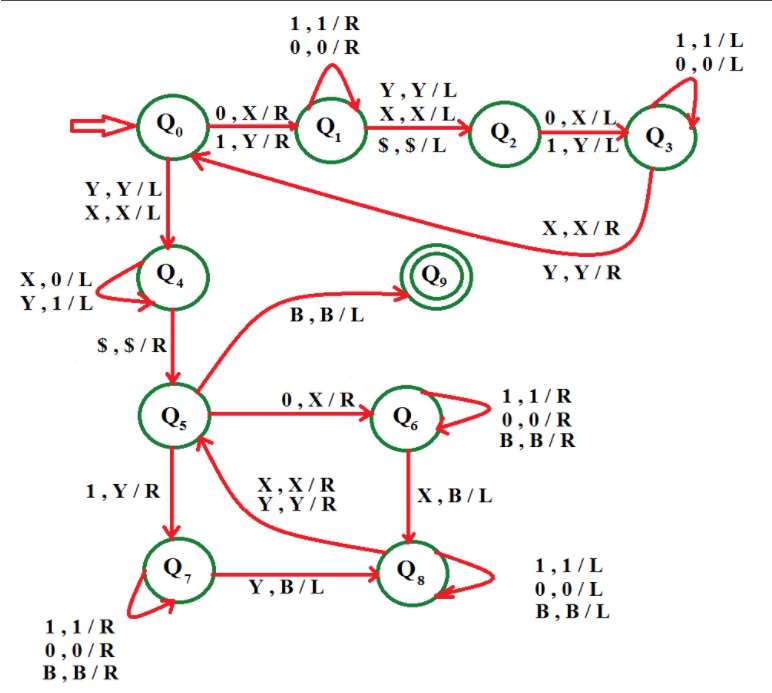
**Descriptive Definition:**

Every string of 0’s and 1’s which is followed by itself falls under this language.

**Words:**

{00, 11, 0000, 0101, 1010, 1111, 000000, 001001, 010010, 011011, 100100, 101101, 110110, 111111….}

**Turing machine:**

****

**Approach Used**

The first thing is to find the midpoint of the string, convert a 0 or 1 from the beginning of the string into X or Y respectively and a corresponding 0 or 1 into X or Y from the end of the string. After continuously doing it a point is reached when all 0’s and 1’s have been converted into X and Y respectively. At this point, you are on the midpoint of the string. So, our first objective is fulfilled.

Now, convert all X’s and Y’s on the left of the midpoint into 0’s and 1’s. At this point the first half the string is in the form of 0 and 1. The second half of the string is in the form of X and Y.

Now, start from the beginning of the string. If you have a 0 then convert it into X and move right till reaching the second half, here if we find X then convert it into a blank(B). Then traverse back till find an X or a Y. We convert the 0 or 1 on the right of it into X or Y respectively and correspondingly, convert its X or Y in the second half of string into a blank(B).

Keep doing this till converted all symbols on the left part of the string into X and Y and all symbols on the right of string into blanks. If any one part is completely converted but still some symbols in the other half are left unchanged then the string will not be accepted. If you did not find an X or Y in the second half for a corresponding 0 or 1 respectively in the first half. Then also string will not be accepted.

**Example** – Let string 1 0 1 1 0 1, so w = 1 0 1 and string is of form (ww).

The first thing that we do is to find the midpoint. For this, we convert 1 in the beginning into Y and move right till the end of the string. Here we convert 1 into y. Now our string would look like Y 0 1 1 0 Y. Now move left till, find a X or Y. When we do so, convert the 0 or 1 right of it to X or Y respectively and then do the same on the right end. Now our string would look like Y X 1 1 X Y. Thereafter, convert these 1’s also and finally it would look like Y X Y Y X Y. At this point, you have achieved the first objective which was to find the midpoint. Now convert all X and Y on the left of midpoint into 0 and 1 respectively, so string becomes 1 0 1 Y X Y. Now, convert the 1 into Y and move right till, find Y in the beginning of the right part of the string and convert this Y into a blank (denoted by B). Now string looks like Y 0 1 B X Y. Similarly, apply this on 0 and x followed by 1 and Y. After this string looks like Y X Y B B B. Now that you have no 0 and 1 and all X and Y on the right part of the string are converted into blanks so our string will be accepted.

**Code:**

string = input("Enter String: ")

length = len(string) + 2

tape = ['B']\*length

i = 1

tapehead = 1

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

    tape[i] = s

    i += 1

state = 0

#assigning characters to variable so that don't have to use characters each time

X, Y, S, B, R, L = 'X', 'Y', 'S', 'B', 'R', 'L'

oldtapehead = -1

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

while(oldtapehead != tapehead): #if tapehead not moving that means terminate Turing machine

    oldtapehead = tapehead

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

    if state == 0:

        if action('0', X, R, 1) or action('1', Y, R, 1) or action(Y, Y, L, 4) or action(X, X, L, 4):

            pass

    elif state == 1:

        if action('1', '1', R, 1) or action('0', '0', R, 1) or action(Y, Y, L, 2) or action(X, X, L, 2) or action(B, B, L, 2):

            pass

    elif state == 2:

        if action('0', X, L, 3) or action('1', Y, L, 3):

            pass

    elif state == 3:

        if action('1', '1', L, 3) or action('0', '0', L, 3) or action(X, X, R, 0) or action(Y, Y, R, 0):

            pass

    elif state == 4:

        if action(X, '0', L, 4) or action(Y, '1', L, 4) or action(B, B, R, 5):

            pass

    elif state == 5:

        if action(B, B, L, 9) or action('0', X, R, 6) or action('1', Y, R, 7):

            pass

    elif state == 6:

        if action('1', '1', R, 6) or action('0', '0', R, 6) or action(B, B, R, 6) or action(X, B, L, 8):

            pass

    elif state == 7:

        if action('1', '1', R, 7) or action('0', '0', R, 7) or action(B, B, R, 7) or action(Y, B, L, 8):

            pass

    elif state == 8:

        if action('1', '1', L, 8) or action('0', '0', L, 8) or action(B, B, L, 8) or action(X, X, R, 5) or action(Y, Y, R, 5):

            pass

    elif state == 9:

        accept = True

    else:

        accept = True

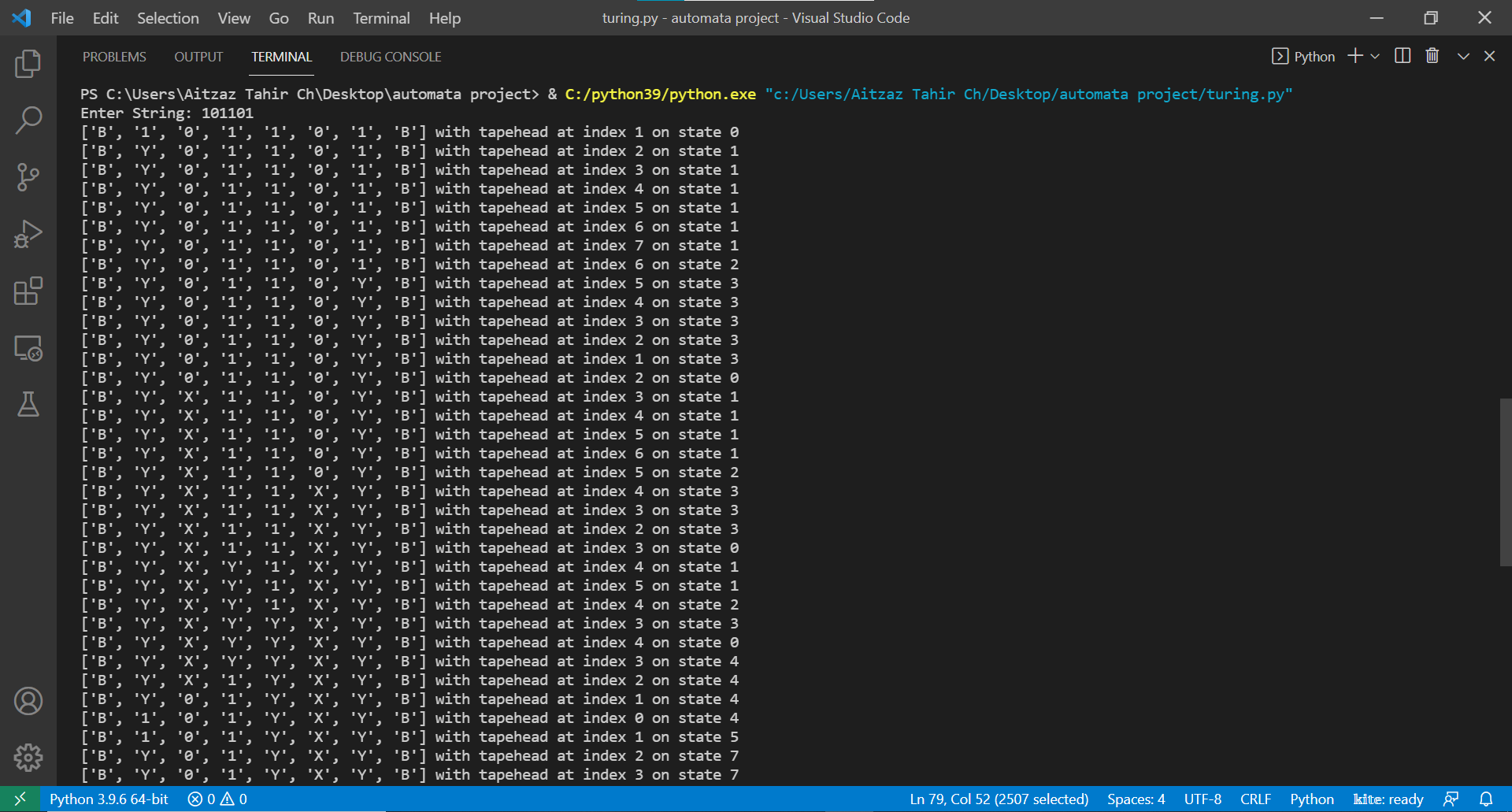
if accept:

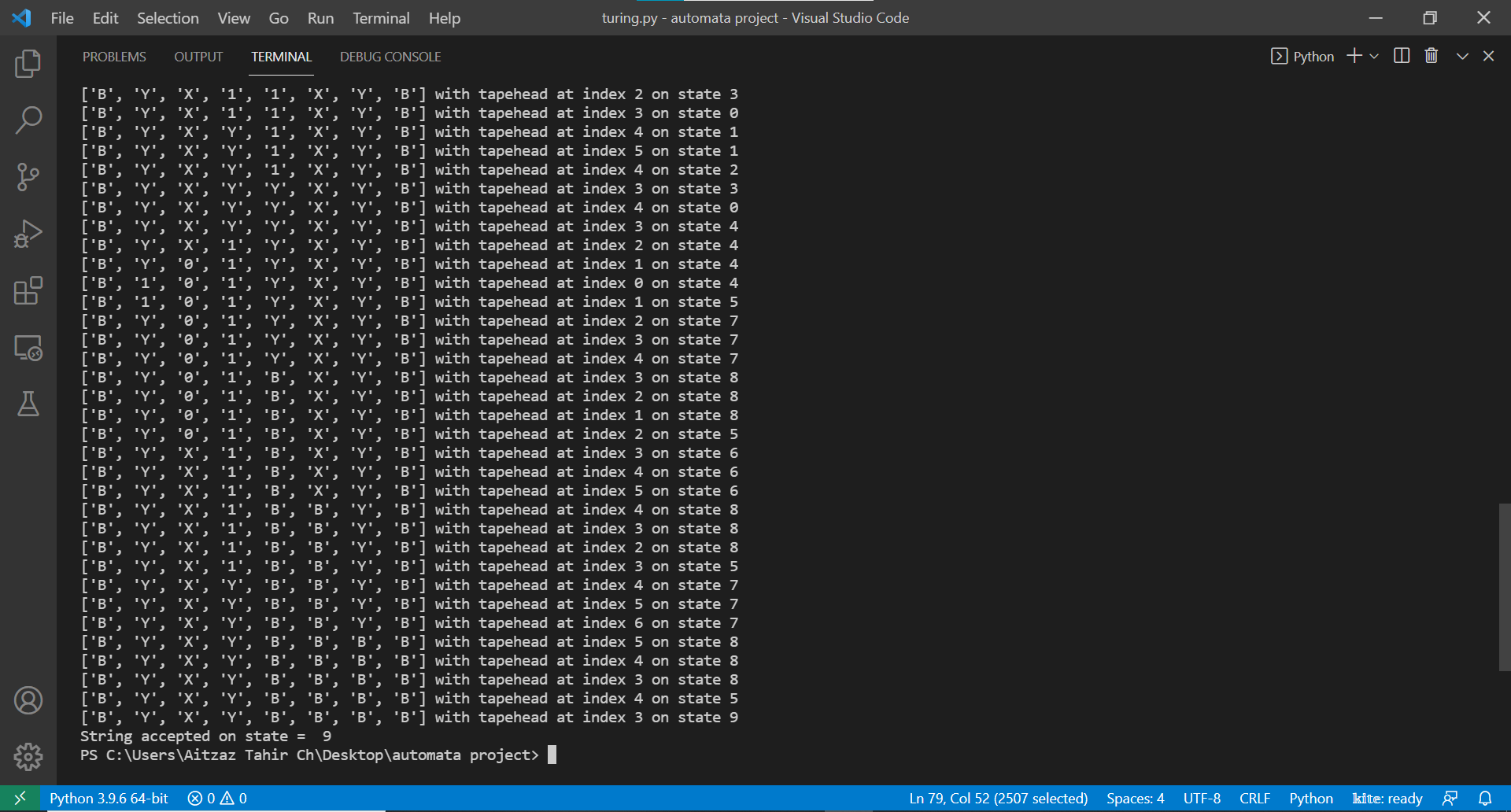
    print("String accepted on state = ", state)

else:

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

**Output:**

****

****

**------------------------------------------------------------------------------------------------------------------------------------------**