Theory of Computation and Compiler design Digital Assessment 2

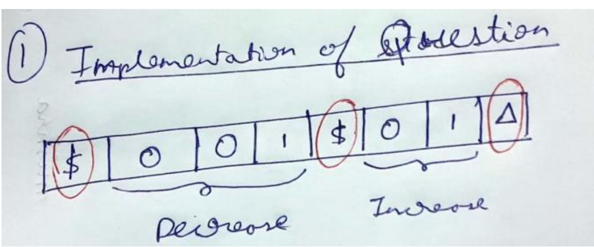
Name: Om Ashish Mishra

Registration Number: 16BCE0789

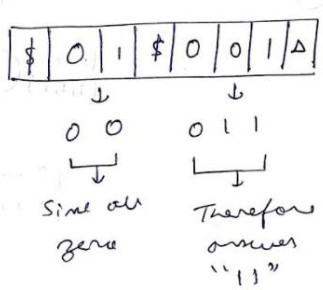
Slot: G1+TG1

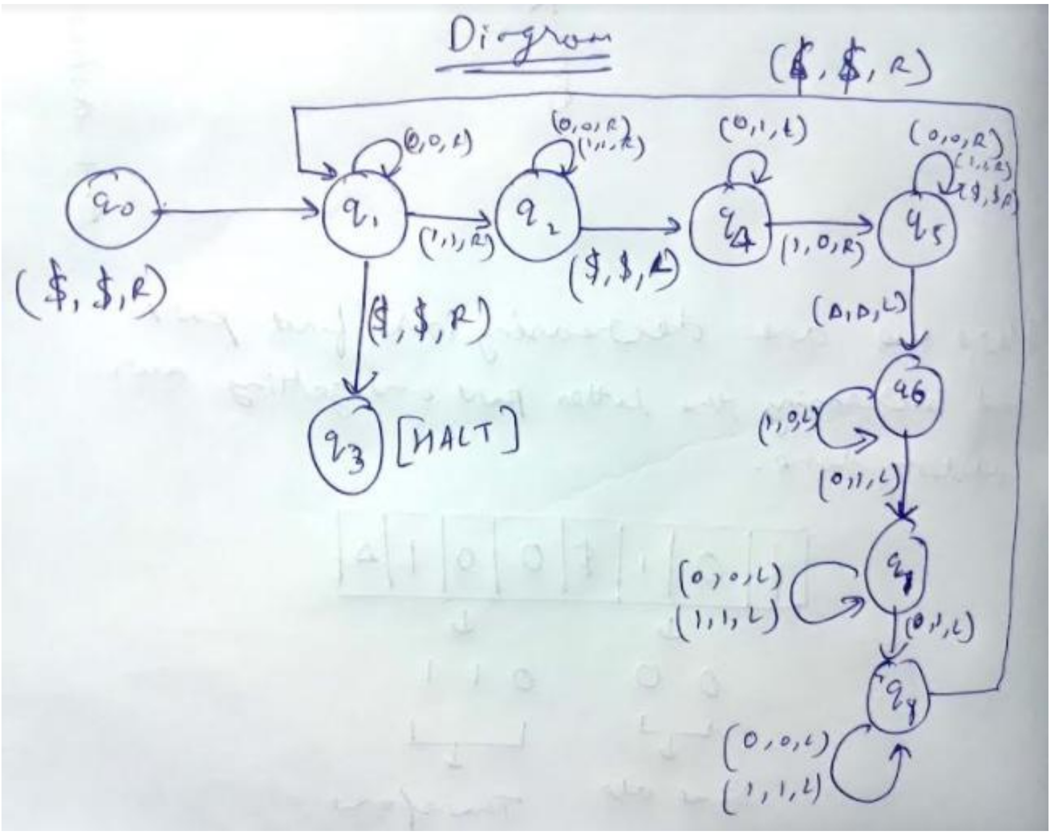
1. Write a classical Turing machine program that computes the addition of non-negative binary numbers. The numbers to be added are given as non-empty binary strings that may have different lengths. Leading zeros are allowed for the input, but not for the output (except for result 0). The two numbers to be added are separated by a blank. For example, input “1#10" yields output “11" and input “01#001" yields output “10”.

The Answer:



Here we are decreasing the first part and increasing the latter part and getting our addition done.





The next step contains the Transition Function:-

Transition State

$(q_0, \$) \rightarrow (q_1, \$, R)$

$(q_1, 0) \rightarrow (q_1, 0, R)$

$(q_1, 1) \rightarrow (q_2, 1, R)$

$(q_2, \$) \rightarrow (q_4, \$, L)$

$(q_4, 1) \rightarrow (q_5, 0, R)$

$(q_5, \$) \rightarrow (q_5, \$, R)$

$(q_5, 0) \rightarrow (q_5, 0, R)$

$(q_5, 1) \rightarrow (q_5, 1, R)$

$(q_5, \Delta) \rightarrow (q_6, \Delta, L)$

$(q_6, 1) \rightarrow (q_6, 0, L)$

$(q_6, 0) \rightarrow (q_7, 1, L)$

$(q_7, 0) \rightarrow (q_7, 0, L)$

$(q_7, 1) \rightarrow (q_7, 1, L)$

$(q_7, 0) \rightarrow (q_8, 1, L)$

$(q_8, \$) \rightarrow (q_1, \$, R)$

$(q_1, \$) \rightarrow (q_3, \$, R)$ [HALT]

$(q_8, 0) \rightarrow (q_8, 0, L)$

$(q_8, 1) \rightarrow (q_8, 1, L)$

$(q_2, 0) \rightarrow (q_2, 0, R)$

$(q_2, 1) \rightarrow (q_2, 1, R)$

$(q_4, 0) \rightarrow (q_4, 1, L)$

The Idea of implementation:

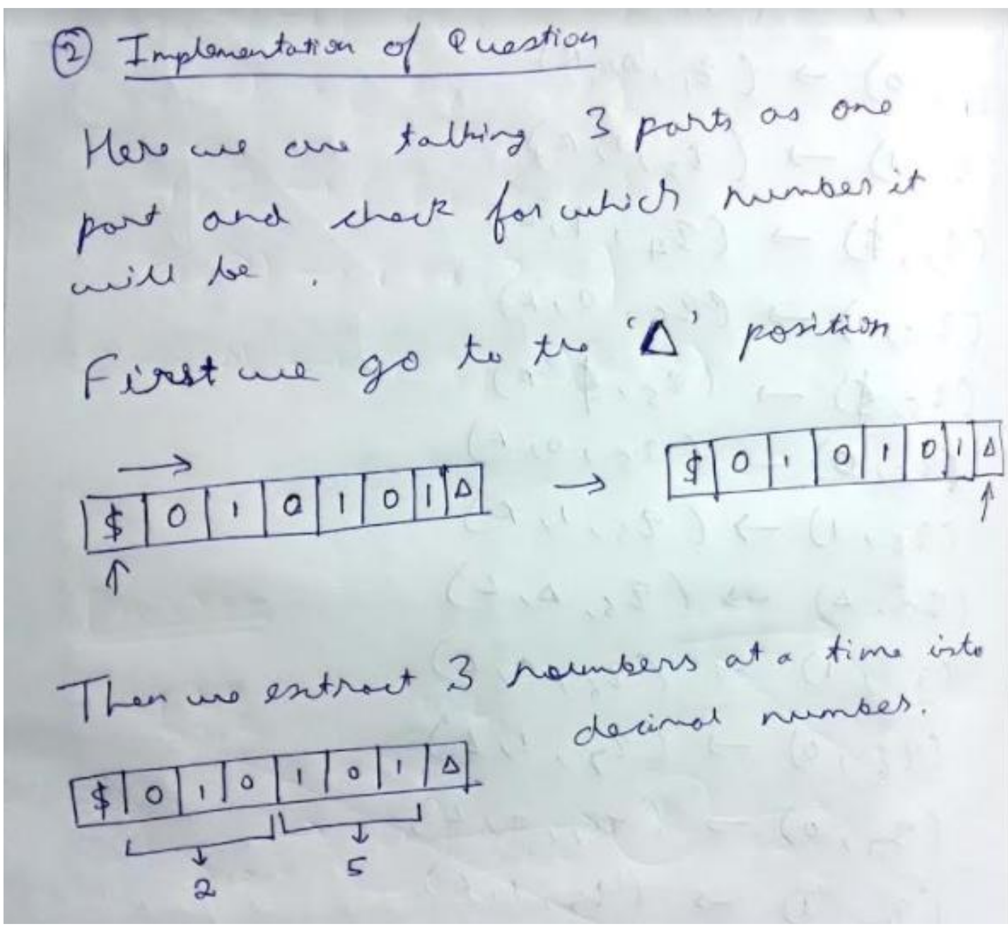
First we have to consider two parts one for decrement of the value and other for increment of the value.

Example: **“01#001"** here the value of “01” will decrease and the value of the “001” will increase. Then we shall remove the extra zeroes and make it possible to get the required answer.

2. Write a classical Turing machine program that computes the conversion of a binary string into an octal string such that the numeric value remains the same. Leading zeros are allowed for the input,but not for the output (except for result 0).

For example, input “1011" yields output “13", since 10112 = 11 = 138, and input “01010001"yields output “10", since 010100012 = 81 = 1218.

The Answer:



Transition State

$(q_0, \$) \rightarrow (q_1, \$, R)$

$(q_1, 0) \rightarrow (q_1, 0, R)$

$(q_1, 1) \rightarrow (q_2, 1, R)$

$(q_2, \$) \rightarrow (q_4, \$, L)$

$(q_4, 1) \rightarrow (q_5, 0, R)$

$(q_5, \$) \rightarrow (q_5, \$, R)$

$(q_5, 0) \rightarrow (q_5, 0, R)$

$(q_5, 1) \rightarrow (q_5, 1, R)$

$(q_5, \Delta) \rightarrow (q_6, \Delta, L)$

$(q_6, 1) \rightarrow (q_6, 0, L)$

$(q_6, 0) \rightarrow (q_7, 1, L)$

$(q_7, 0) \rightarrow (q_7, 0, L)$

$(q_7, 1) \rightarrow (q_7, 1, L)$

$(q_7, 0) \rightarrow (q_8, 1, L)$

$(q_8, \$) \rightarrow (q_1, \$, R)$

$(q_1, \$) \rightarrow (q_3, \$, R)$ [HALT]

$(q_8, 0) \rightarrow (q_8, 0, L)$

$(q_8, 1) \rightarrow (q_8, 1, L)$

$(q_2, 0) \rightarrow (q_2, 0, R)$

$(q_2, 1) \rightarrow (q_2, 1, R)$

$(q_4, 0) \rightarrow (q_4, 1, L)$

$$(q_9, 3) \rightarrow (q_9, 3, R)$$

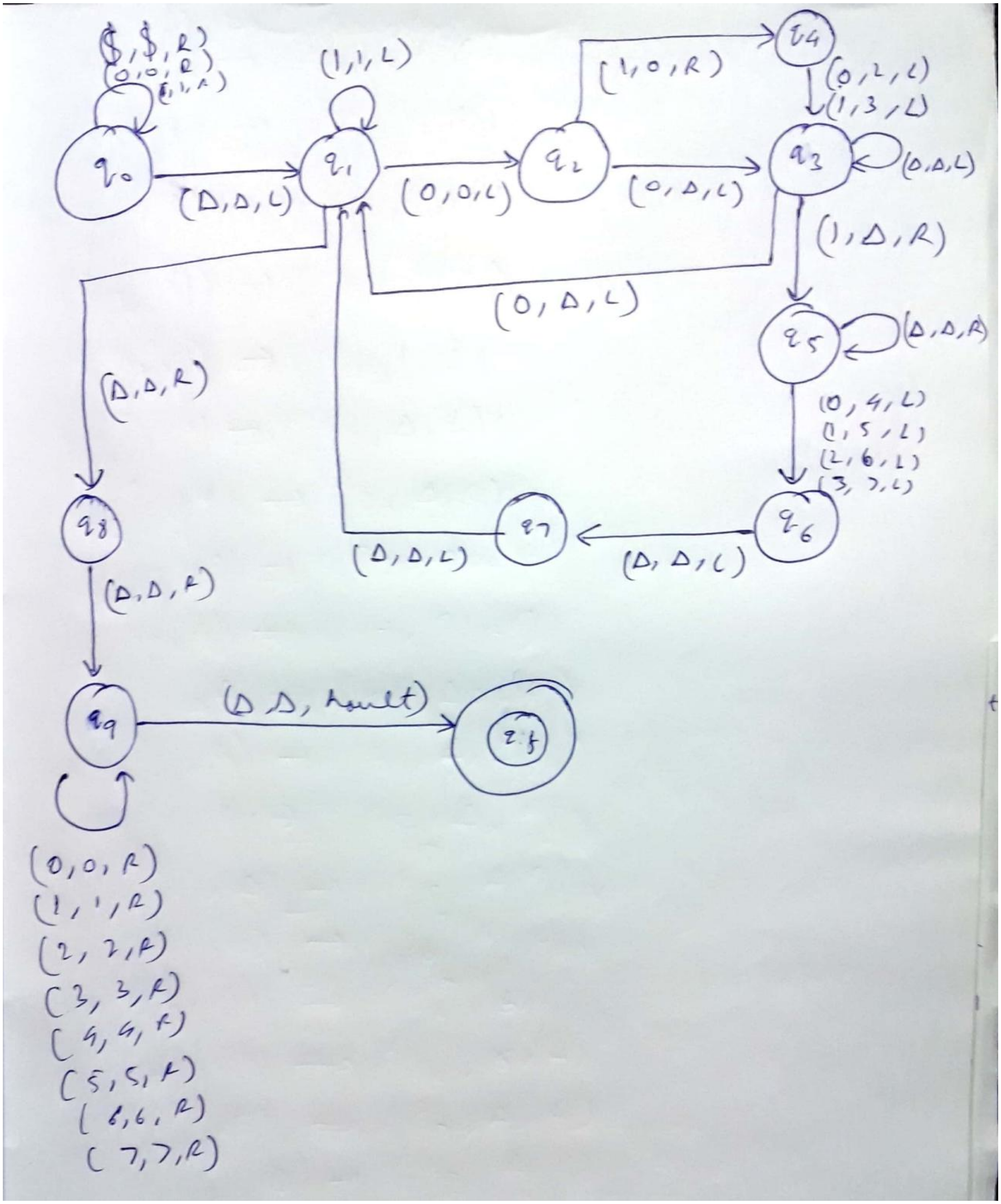
$$(q_9, 4) \rightarrow (q_9, 4, R)$$

$$(q_9, 5) \rightarrow (q_9, 5, R)$$

$$(q_9, 6) \rightarrow (q_9, 6, R)$$

$$(q_9, 7) \rightarrow (q_9, 7, R)$$

$$(q_9, \Delta) \rightarrow (q_f, \Delta)(\text{halt})$$



3. Write a classical Turing machine program that `accepts' the language L = {w1_w2_w2 | w1,w2 belongs {a, b, c}*}: the program yields the result „Y' if the argument string w belongs {a, b, c,*}* is in L, and `N' if it is not.

For example, input “cca*abc*abc" yields output “Y", since abc = abc, thus cca_abc_abc 2 L,and input \abc*abc*cba" yields output \N", since abc 6= cba, hence abc_abc_cba =2 L.

The Answer:

$$(q_0, a) \rightarrow (q_0, a, R)$$

$$(q_0, b) \rightarrow (q_0, b, R)$$

$$(q_0, c) \rightarrow (q_0, c, R)$$

$$(q_0, *) \rightarrow (q_1, *, R)$$

$$(q_1, a) \rightarrow (q_2, x, R)$$

$$(q_2, a) \rightarrow (q_2, a, R)$$

$$(q_2, b) \rightarrow (q_2, b, R)$$

$$(q_2, c) \rightarrow (q_2, c, R)$$

$$(q_2, *) \rightarrow (q_3, *, R)$$

$$(q_3, a) \rightarrow (q_4, x, L)$$

$$(q_4, *) \rightarrow (q_4, *, L)$$

$$(q_4, a) \rightarrow (q_4, a, L)$$

$$(q_4, b) \rightarrow (q_4, b, L)$$

$$(q_4, c) \rightarrow (q_4, c, L)$$

$$(q_4, x) \rightarrow (q_1, x, R)$$

$(q_3, c) \rightarrow (q_6, z, L)$

$(q_6, z) \rightarrow (q_1, z, R)$

$(q_1, *) \rightarrow (q_7, *, R)$

$(q_7, x) \rightarrow (q_7, x, R)$

$(q_7, y) \rightarrow (q_7, y, R)$

$(q_7, z) \rightarrow (q_7, z, R)$

$(q_7, \Delta) \rightarrow (q_f, \Delta, halt)$

$(q_5, x) \rightarrow (q_5, x, L)$

$(q_5, *) \rightarrow (q_5, *, L)$

$(q_5, c) \rightarrow (q_5, c, L)$

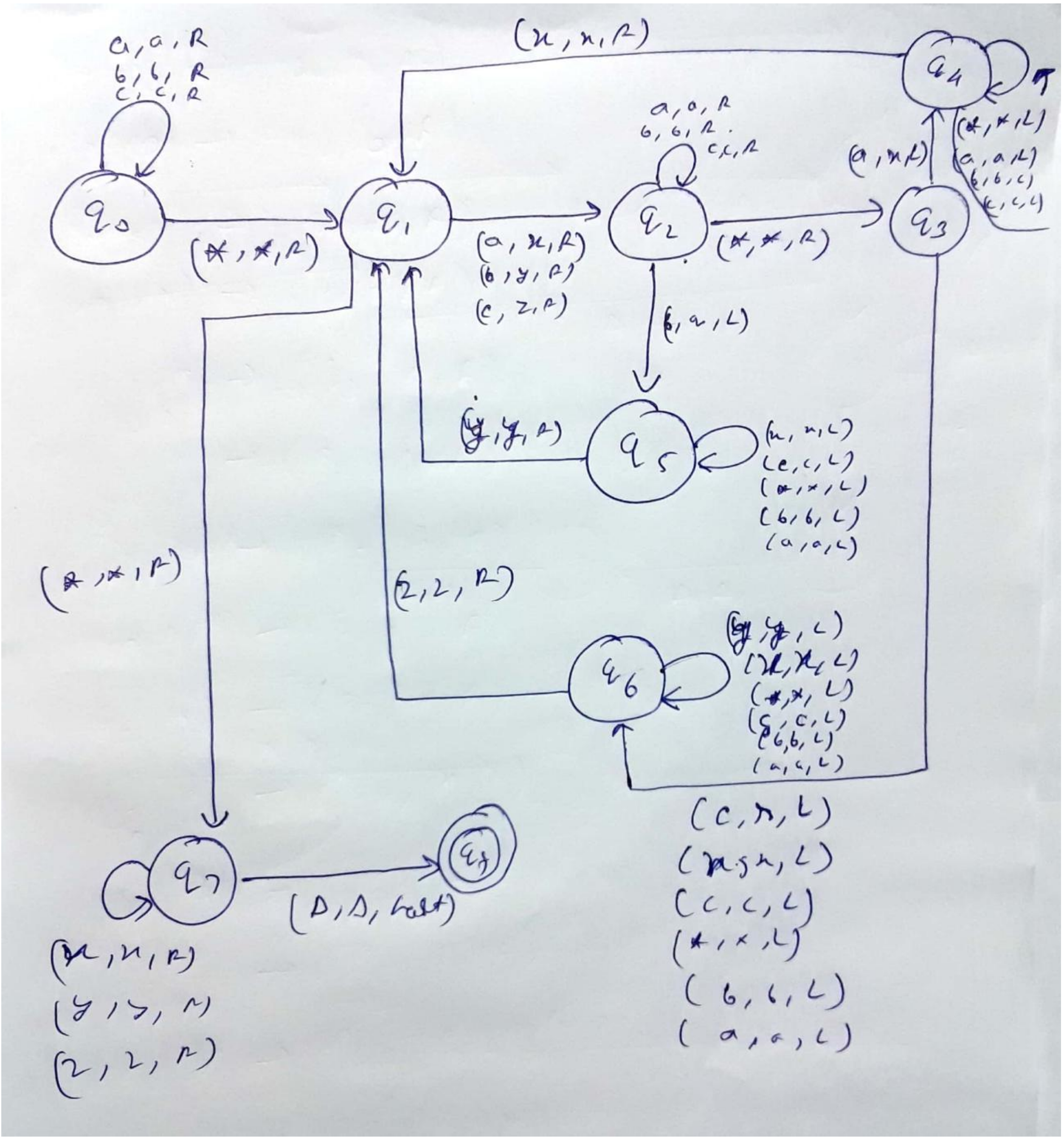
$(q_5, b) \rightarrow (q_5, b, L)$

$(q_5, a) \rightarrow (q_5, a, L)$

$(q_6, y) \rightarrow (q_6, y, L)$

$(q_6, x) \rightarrow (q_6, x, L)$

$(q_6, *) \rightarrow (q_6, *, L)$

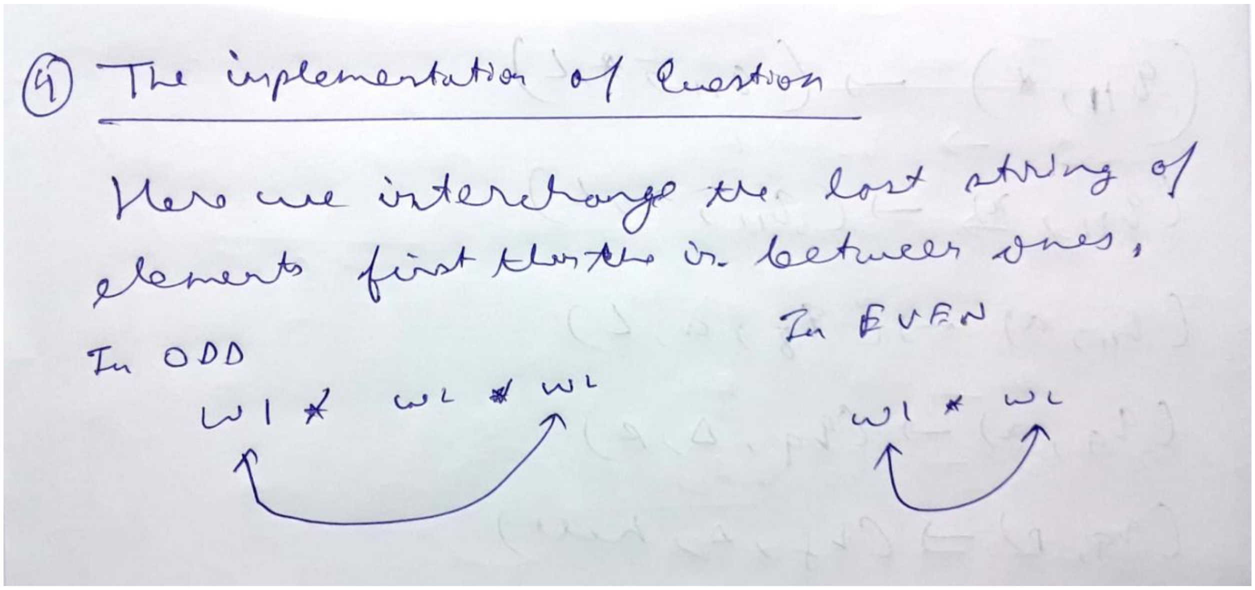


4. Write a classical Turing machine program that maps a string w1.w2. …….wn belongs {a,b,*}*, with n > 0, wi belongs {a,b}* for 1 <= i <= n, to the

string wn. …..w2.w1, reversing the order of the substrings w1 to wn, but not the substrings themselves.

For example, input “aab*bb*aaab" yields output “aaab*bb*aab", and input “abb*ab" yields output “ab*abb"

The Answer:



(4) The implementation of Question

Here we interchange the last string of elements first then the in between ones,

In ODD

w1 * w2 * w2

In EVEN

w1 * w2

The Transition states :-

$(q_0, a) \rightarrow (q_0, a, R)$

$(q_0, b) \rightarrow (q_0, b, R)$

$(q_0, *) \rightarrow (q_1, *, R)$

$(q_1, a) \rightarrow (q_1, a, R)$

$(q_1, b) \rightarrow (q_1, b, R)$

$(q_1, *) \rightarrow (q_2, *, R)$

$(q_2, a) \rightarrow (q_3, x, R)$

$(q_3, b) \rightarrow (q_3, b, R)$

$(q_3, a) \rightarrow (q_3, a, R)$

$(q_3, \Delta) \rightarrow (q_4, P, R)$

$(q_4, a) \rightarrow (q_4, a, L)$

$(q_4, b) \rightarrow (q_4, b, L)$

$(q_4, x) \rightarrow (q_2, x, R)$

$(q_2, b) \rightarrow (q_5, x, R)$

$(q_7, x) \rightarrow (q_7, x, L)$

$(q_8, a) \rightarrow (q_8, a, L)$

$(q_8, b) \rightarrow (q_8, b, L)$

$(q_8, *) \rightarrow (q_2, *, R)$

$(q_3, x) \rightarrow (q_3, y, R)$

$(q_3, P) \rightarrow (q_4, P, R)$

$(q_4, y) \rightarrow (q_4, x, L)$

$(q_5, x) \rightarrow (q_5, y, R)$

$(q_6, y) \rightarrow (q_6, x, L)$

$(q_8, \Delta) \rightarrow (q_9, \Delta, R)$

$(q_4, \Delta) \rightarrow (q_4, \Delta, L)$

$(q_4, P) \rightarrow (q_4, P, L)$

$(q_{11}, *) \rightarrow (q_{11}, *, L)$

$(q_{11}, x) \rightarrow (q_{11}, x, L)$

$(q_{11}, a) \rightarrow (q_8, a, L)$

$(q_9, x) \rightarrow (q_9, \Delta, R)$

$(q_9, P) \rightarrow (q_f, \Delta, halt)$

The Diagram