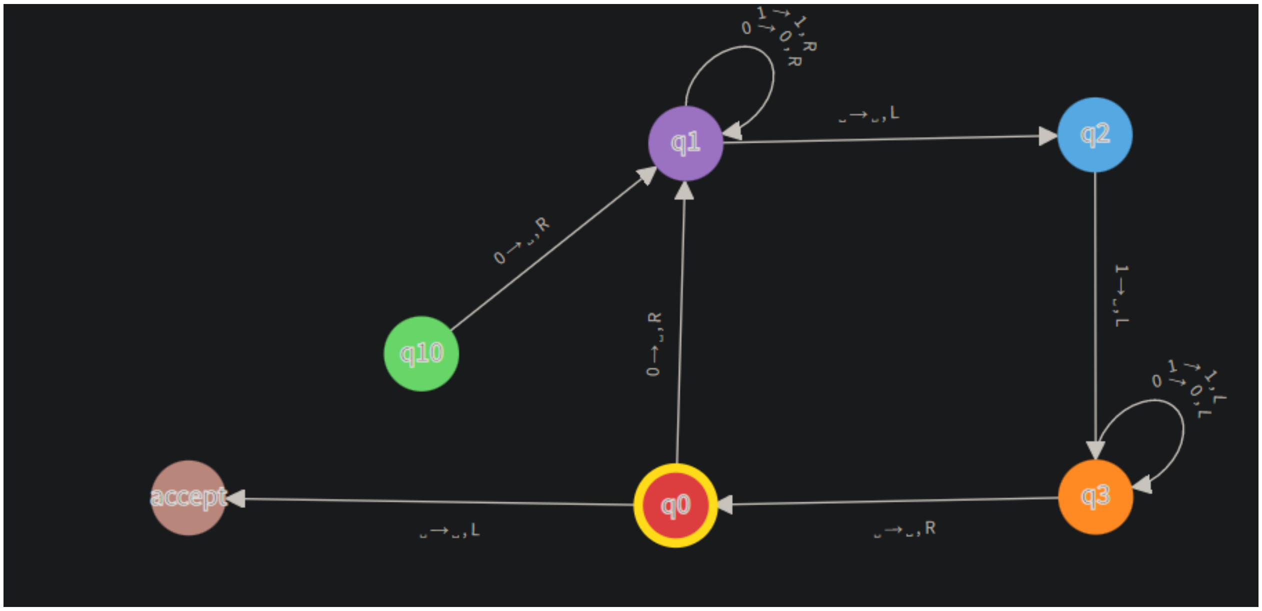
QUESTION 1:



Q10= start state, it checks whether input is empty or not. If it is empty, it cannot go to accept state. If it is not, deletes 0. And if it does not start with 0, it stops.

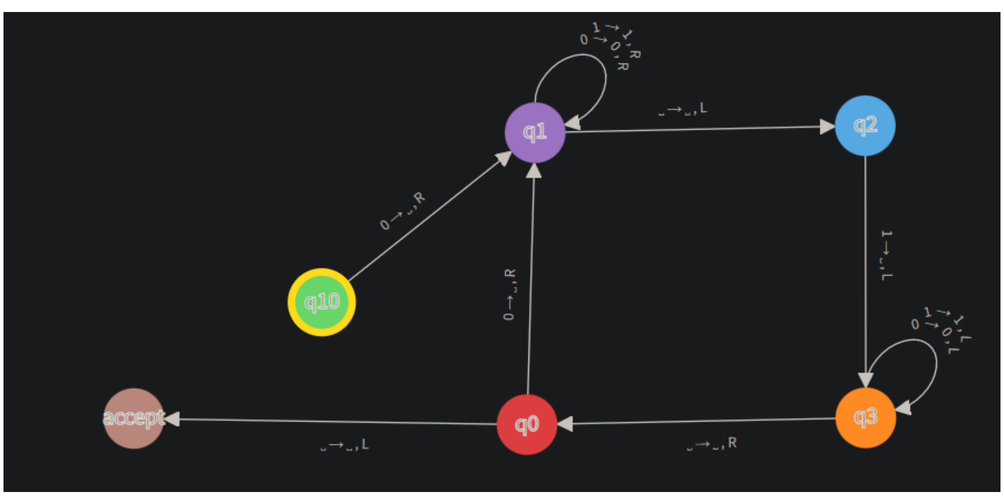
Q1 = goes to the end of the input, finds the blank, then comes back to 1.

Q2 = deletes 1.

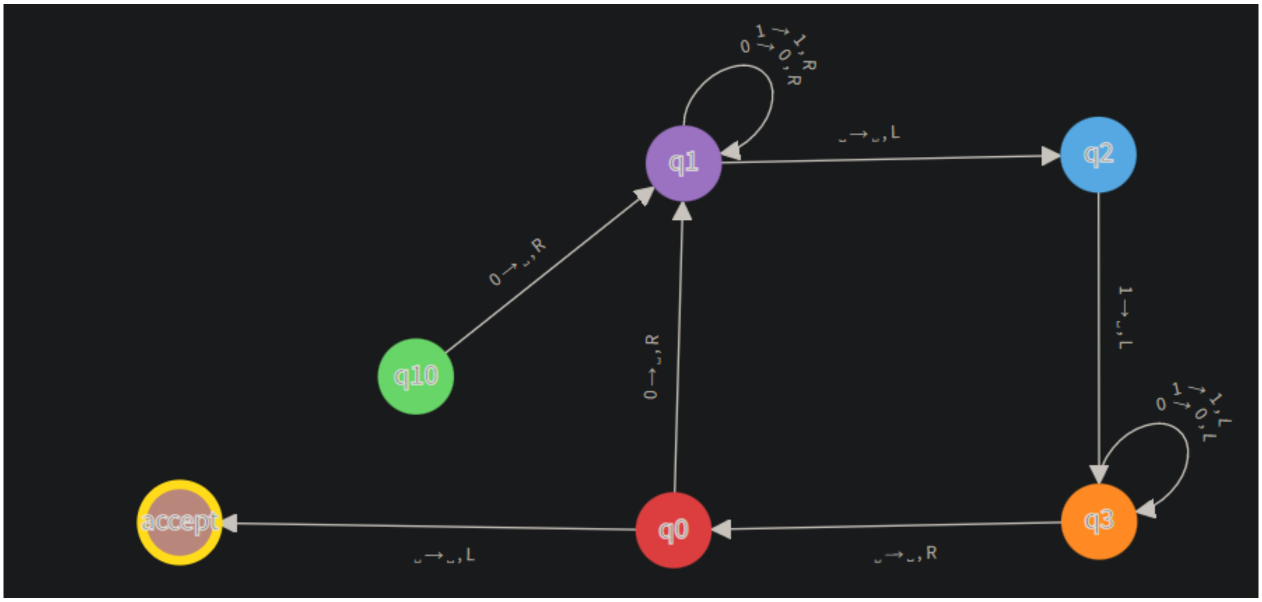
Q3 = goes to the start of the input, finds the blank, then comes back to 0.

This machine counts 1's and 0's by deleting them. If they are in the language L = $0^N 1^N$, machine accepts it.

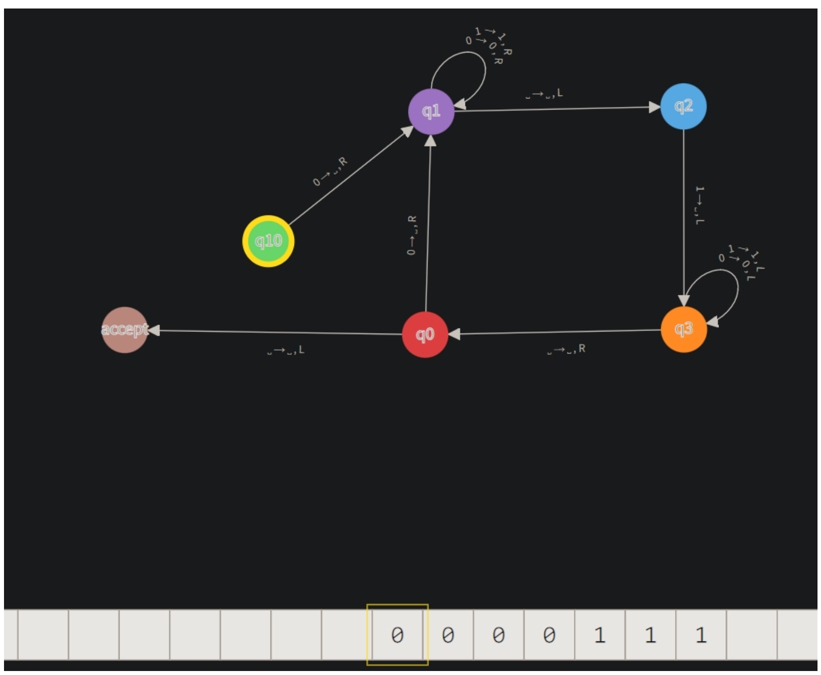
Initial state of 000111:



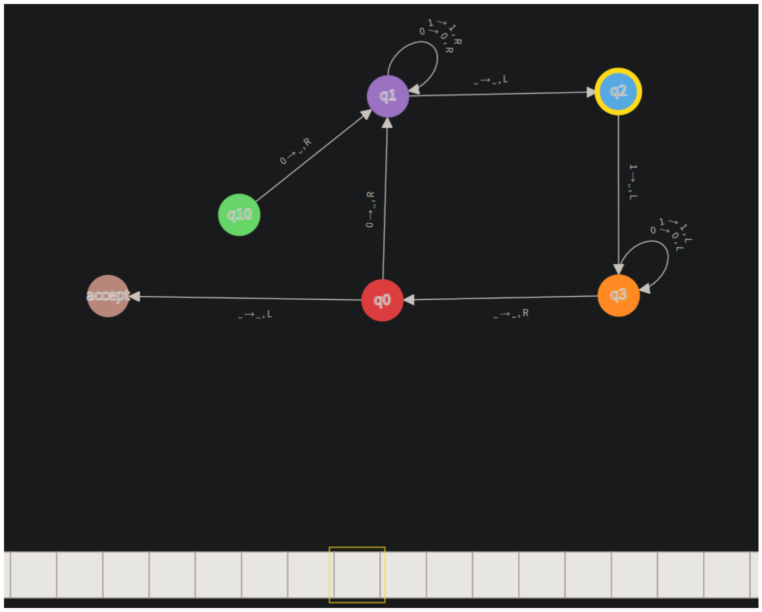
Final state of 000111:



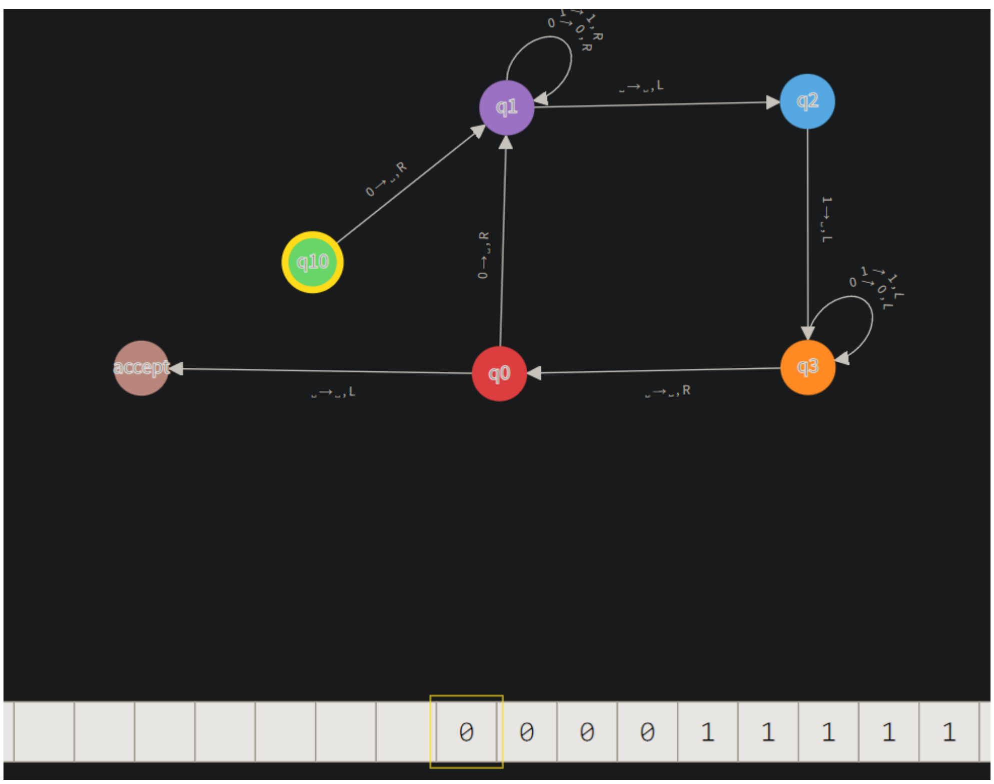
Initial state of 0000111:



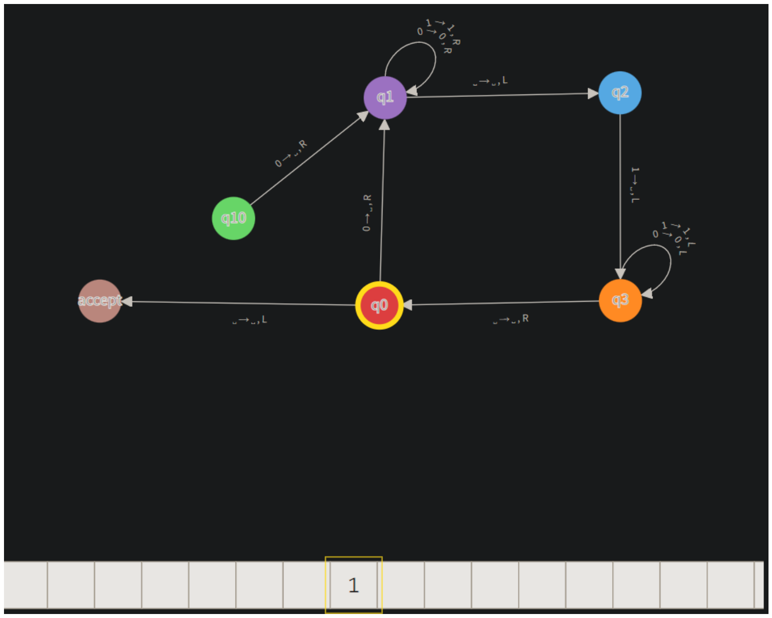
Final state of 0000111:



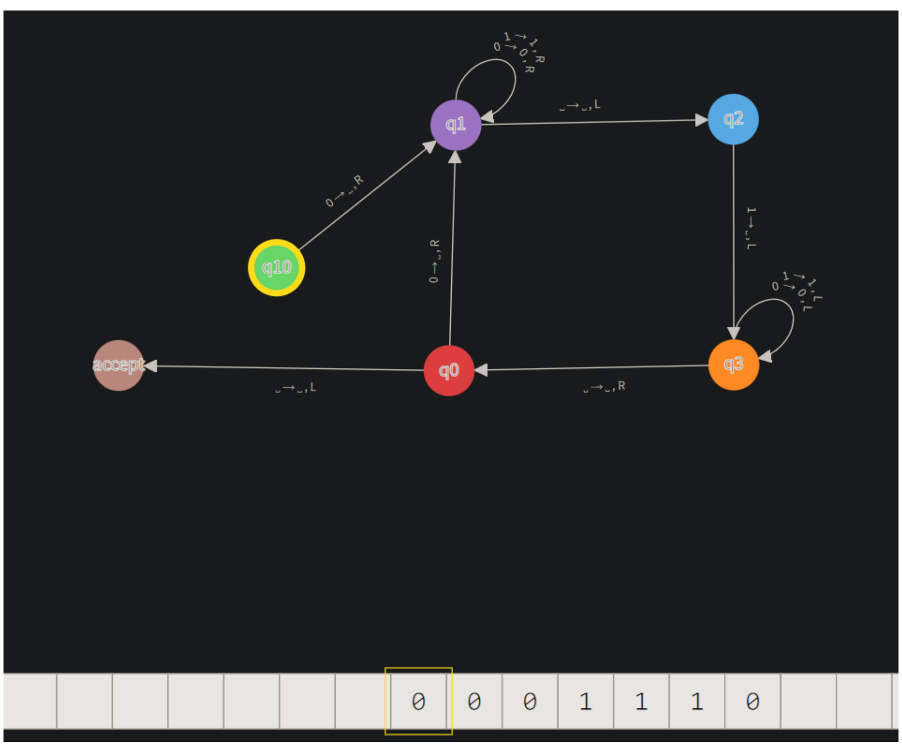
Initial state of 000011111:



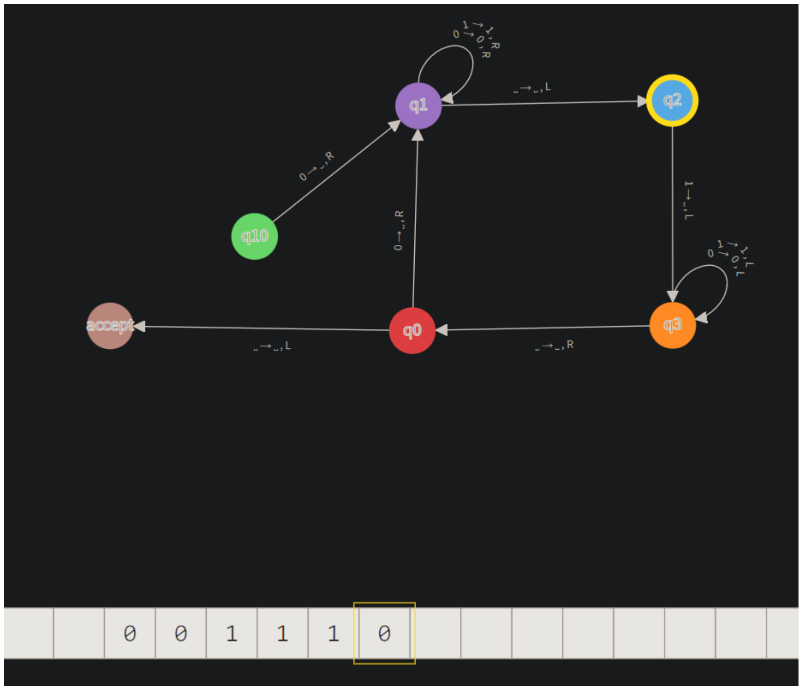
Final state of 000011111:



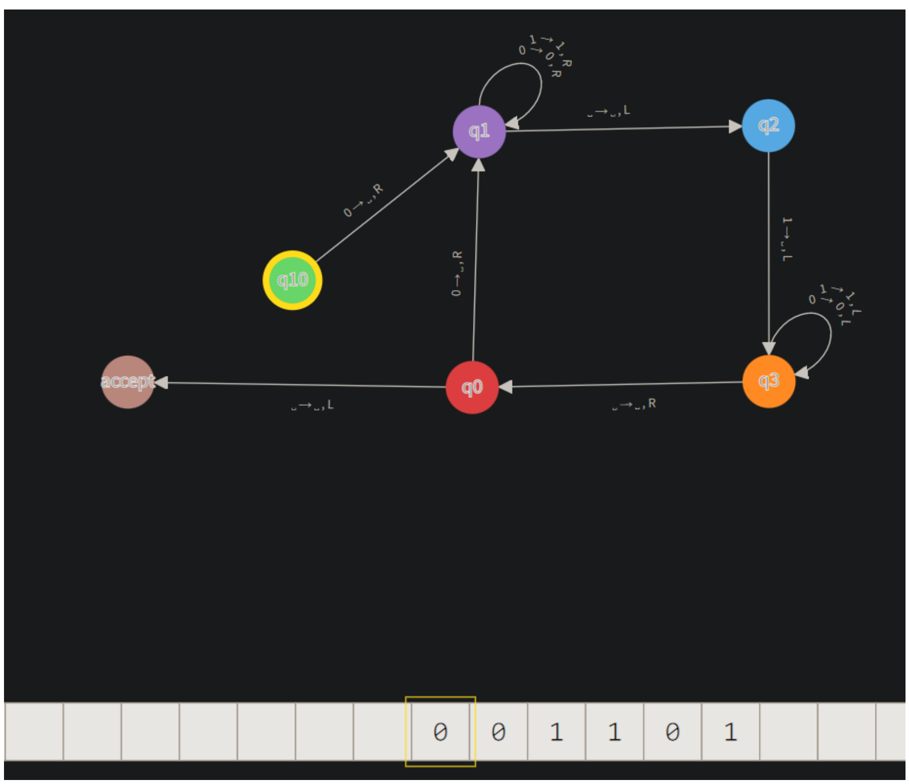
Initial state of 0001110:



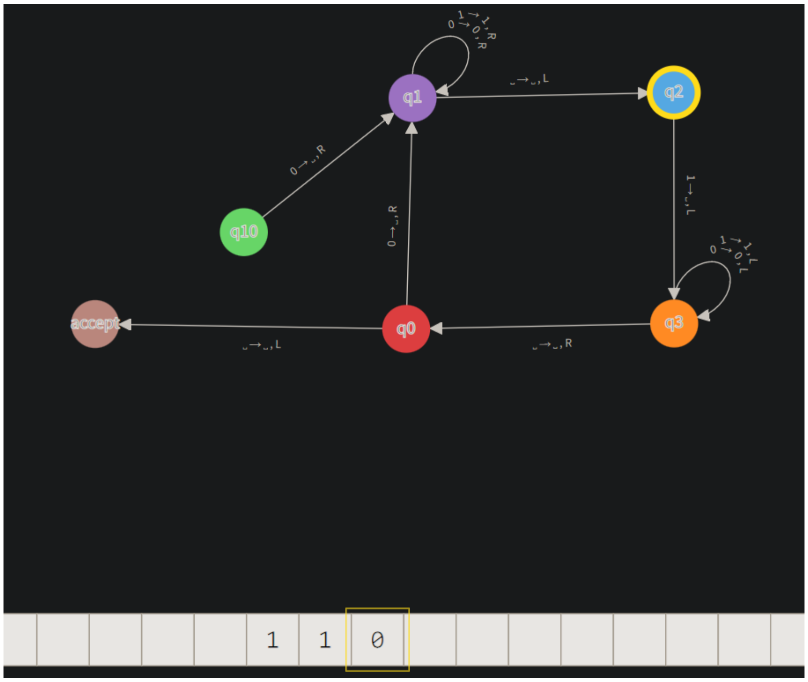
Final state of 0001110:



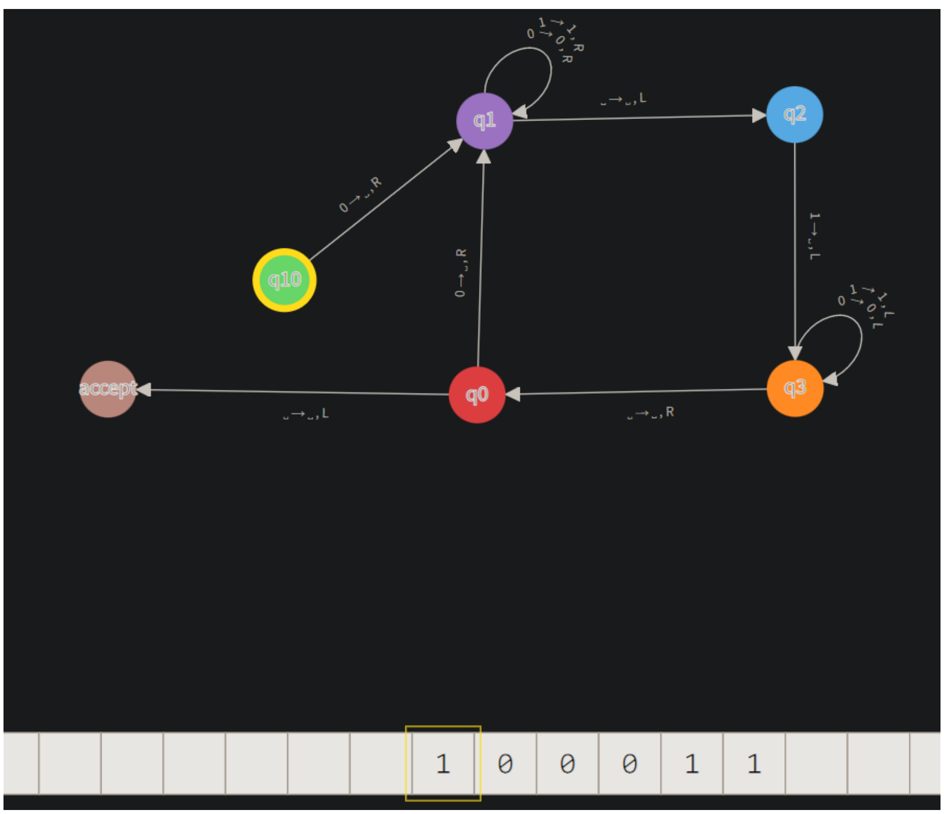
Initial state of 001101:



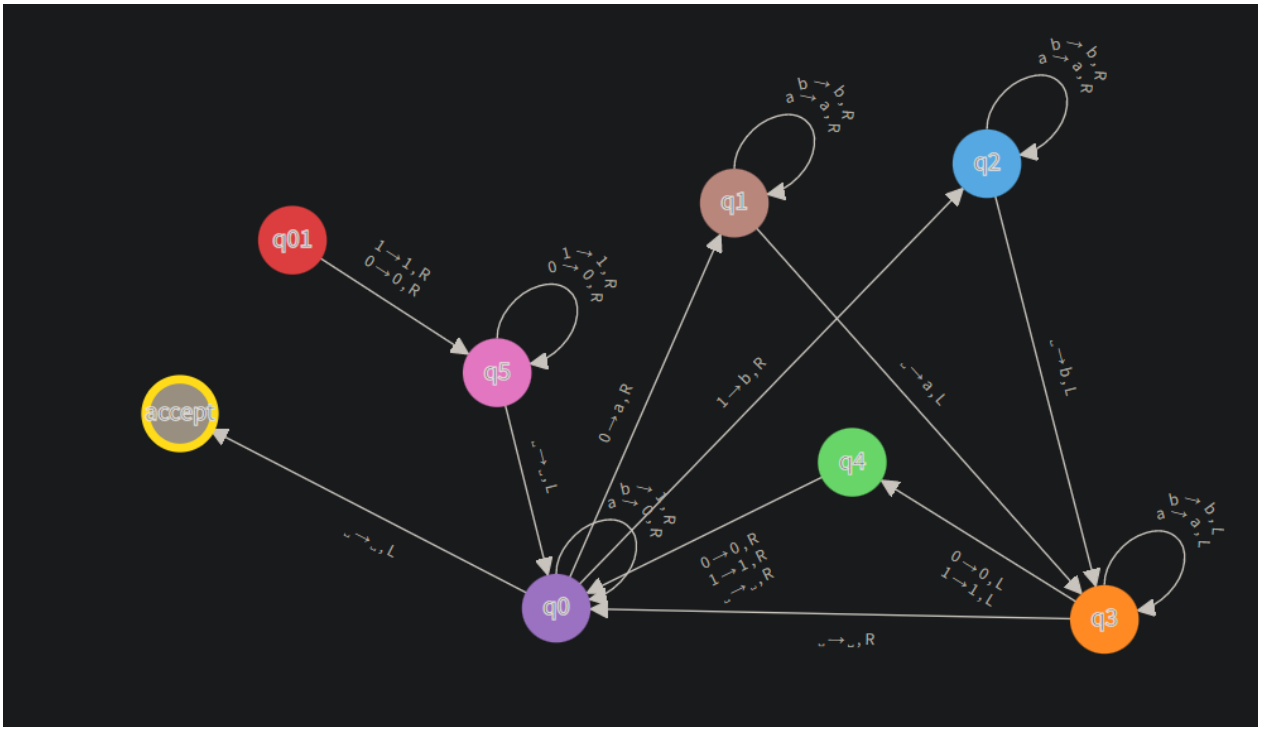
Final state of 001101:



Initial and Final state of 100011:



Question 2:



q01: it is the start state. It checks whether the input is empty or not.

q5: It goes to the end of the input, when it comes to the end, it comes back to the last element of the input.

q0: This state is there to separate 1's and 0's. If the element is 0, it sends it to q1 and makes it a. If the element is 1, it sends it to q2 and makes it b. If it is blank it means it is finally accepted.

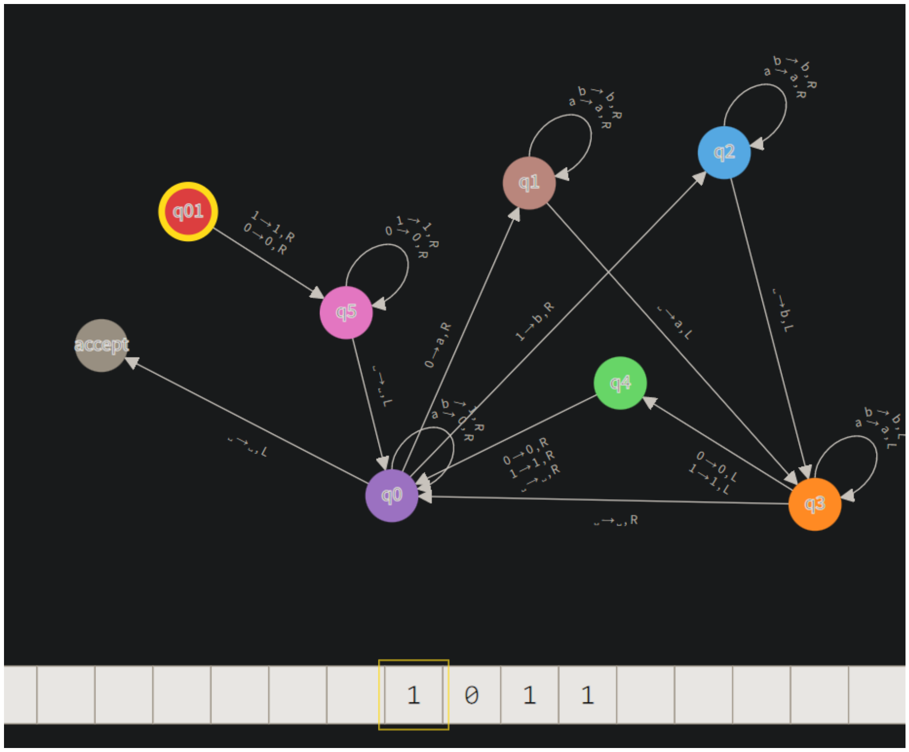
q1: it goes to the end to find the blank. When it finds it, blank becomes a and goes to q3.

q2: similar to q1, it does the same operation, but blank becomes b.

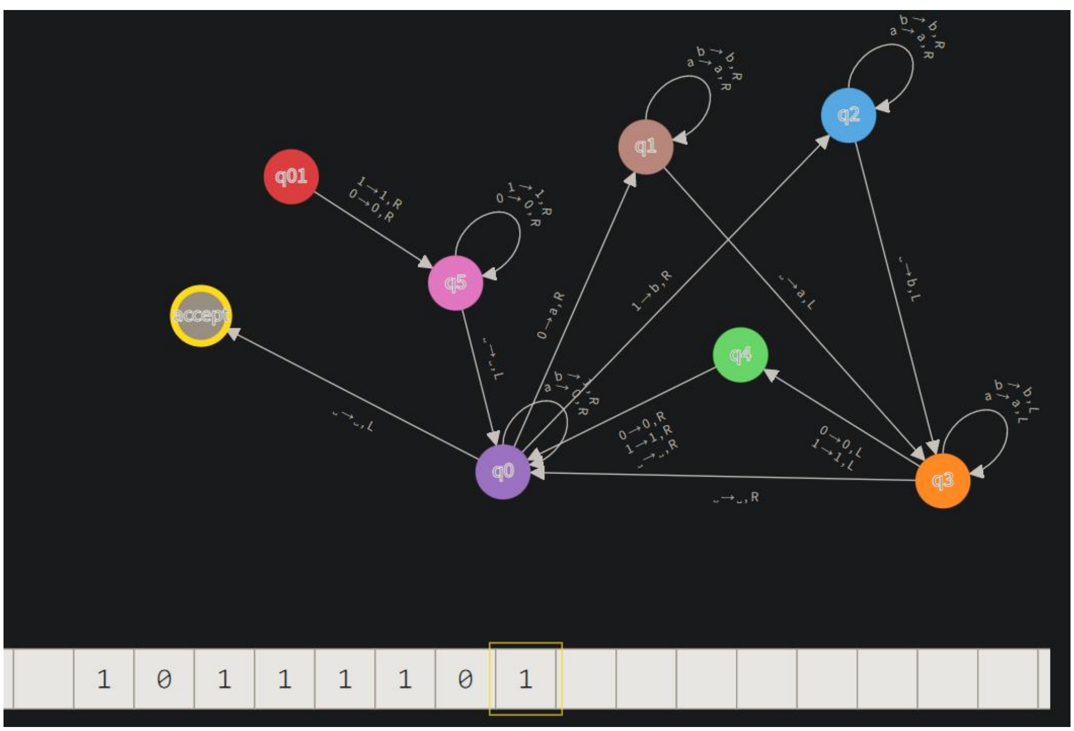
q3: if this state finds 1's and 0's, it goes to q6. If it finds blank, it goes to q0.

q6: this state must find 1's and 0's or it halts. And goes to q0 without changing the input.

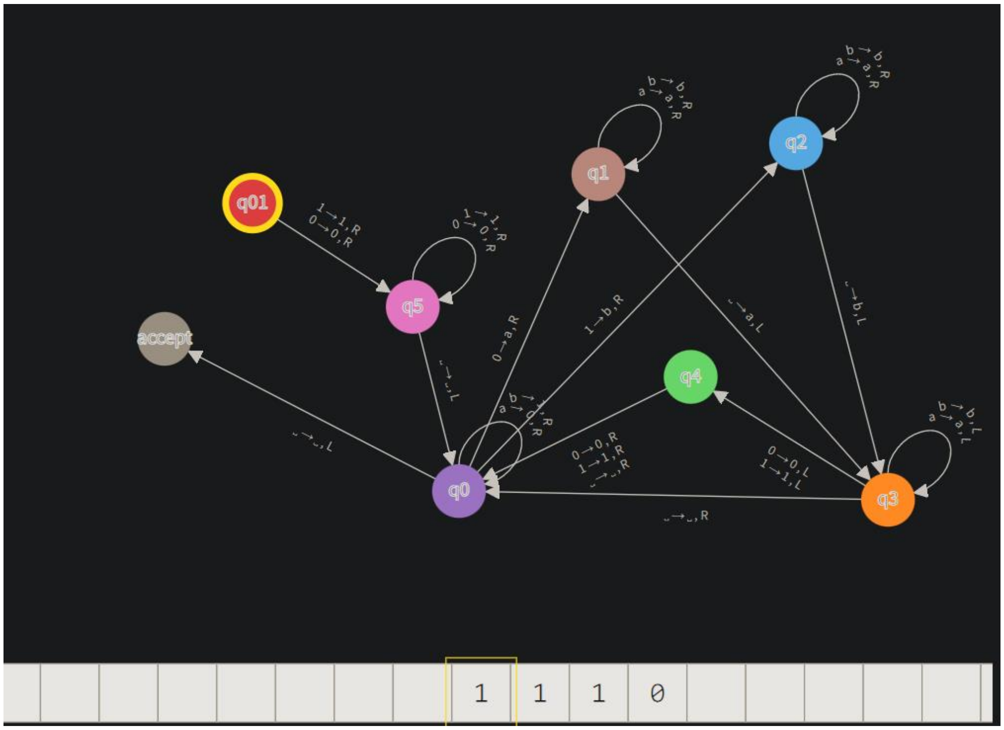
Initial state of 1011:



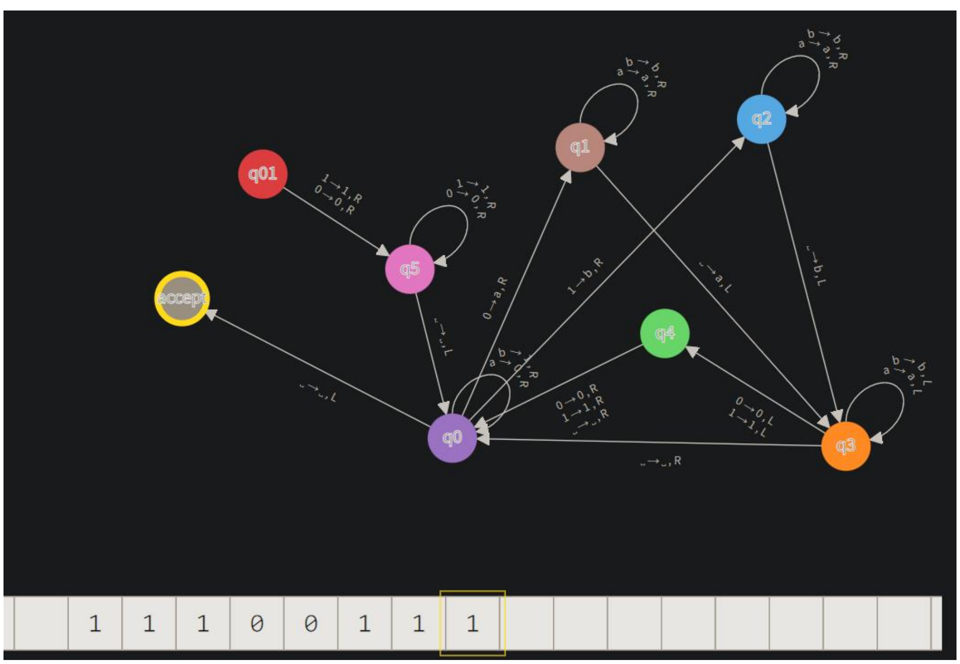
Final state of 1011:



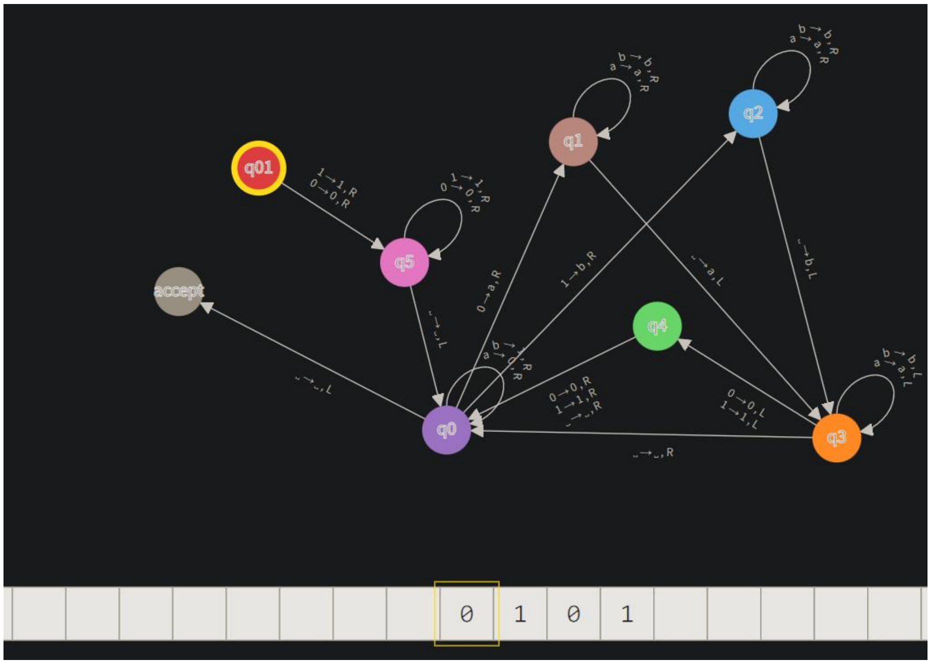
Initial state of 1110:



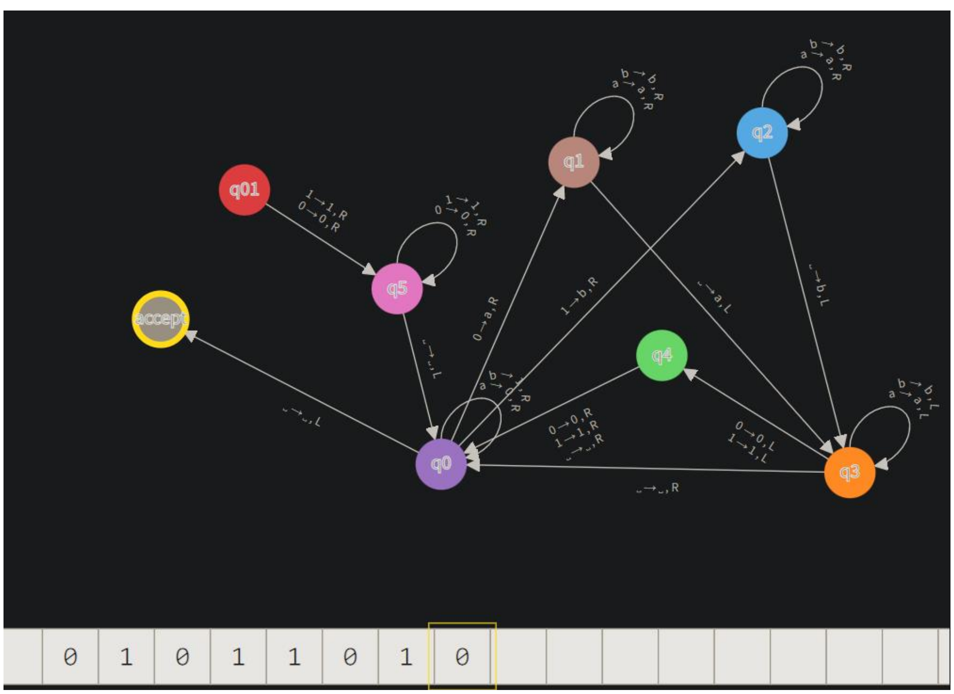
Final state of 1110:



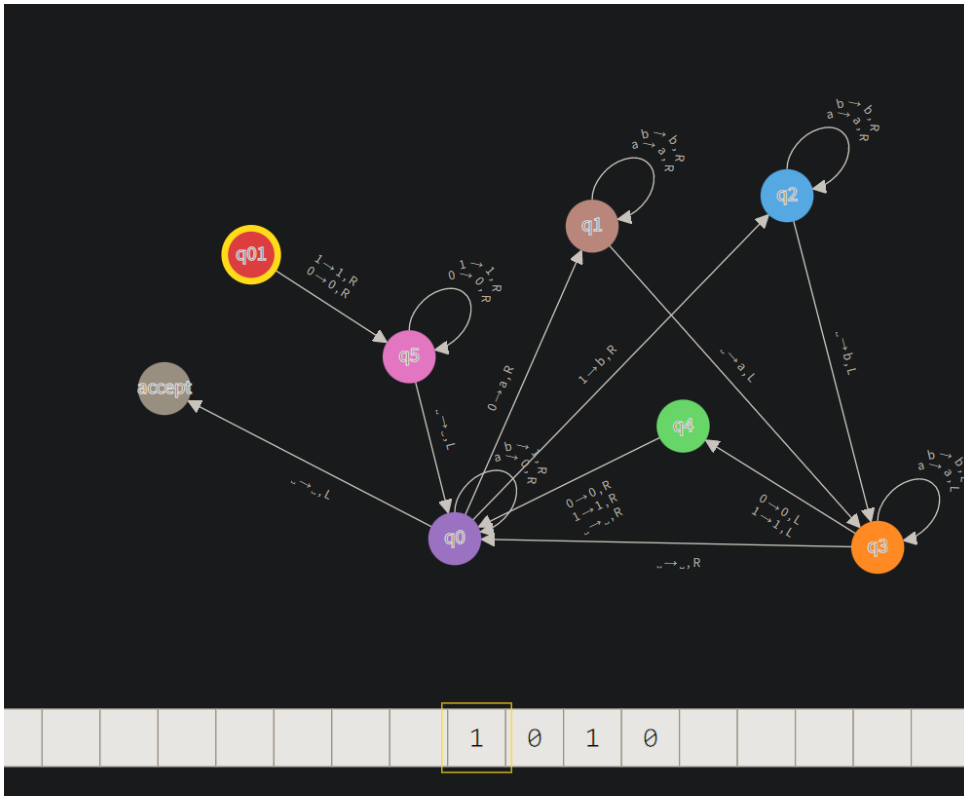
Initial state of 0101:



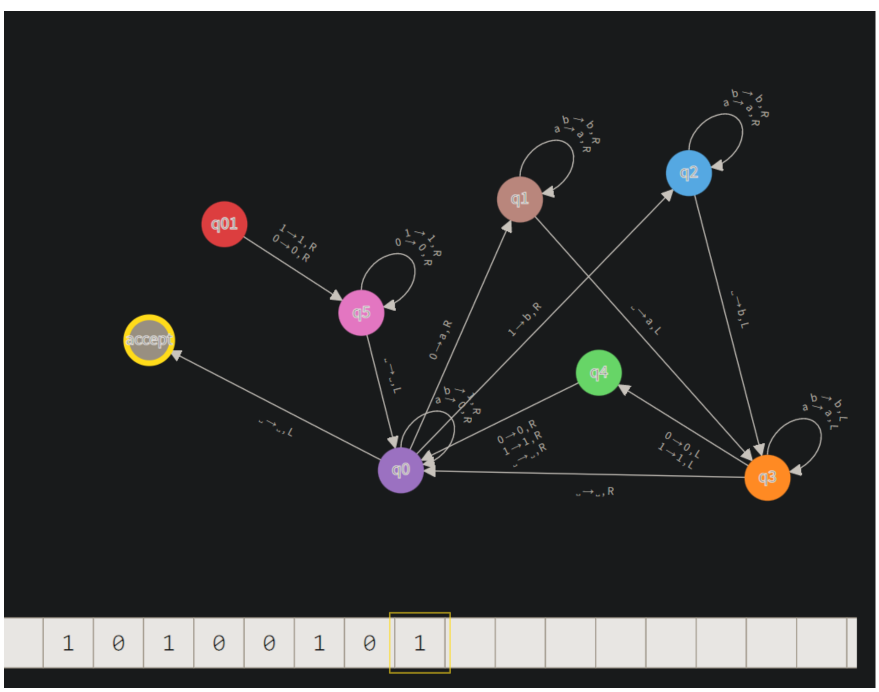
Final state of 0101:



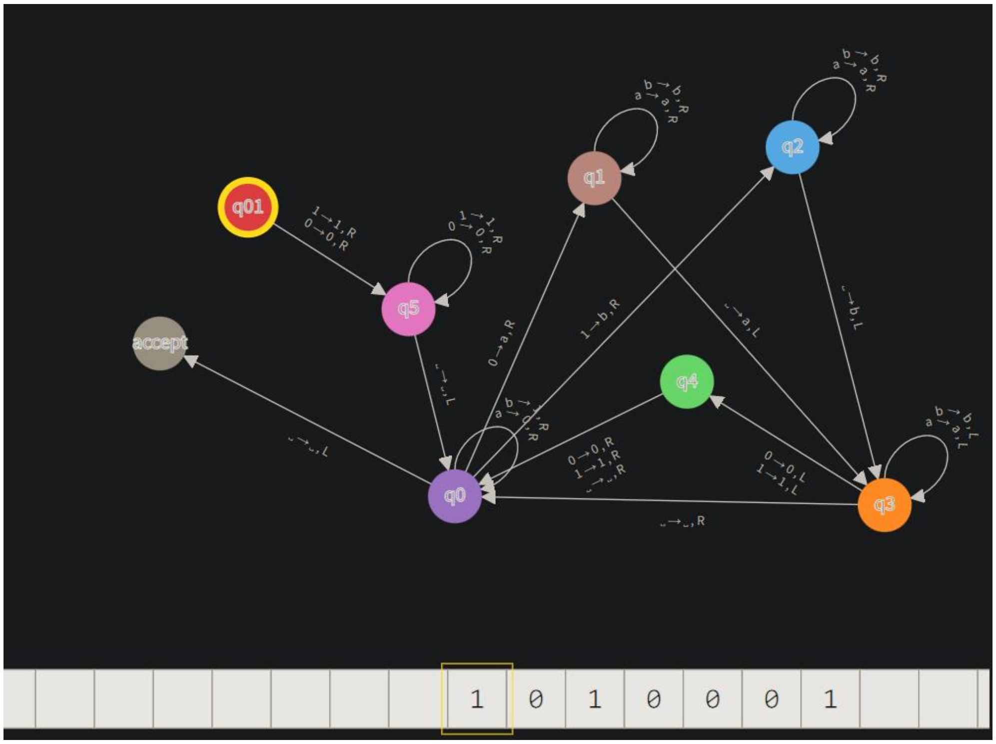
Initial state of 1010:



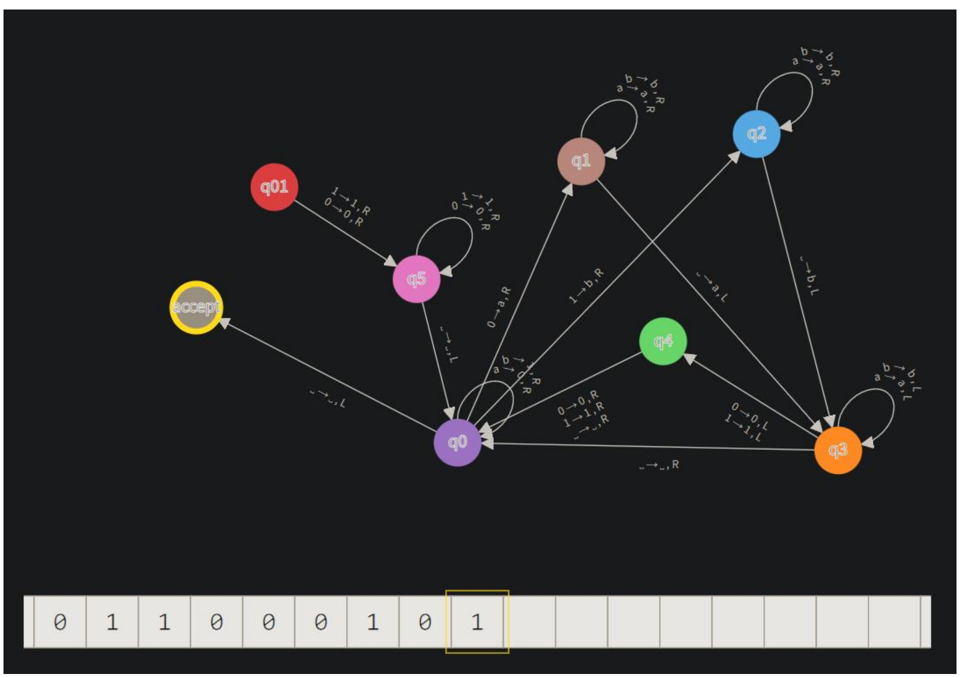
Final state of 1010:



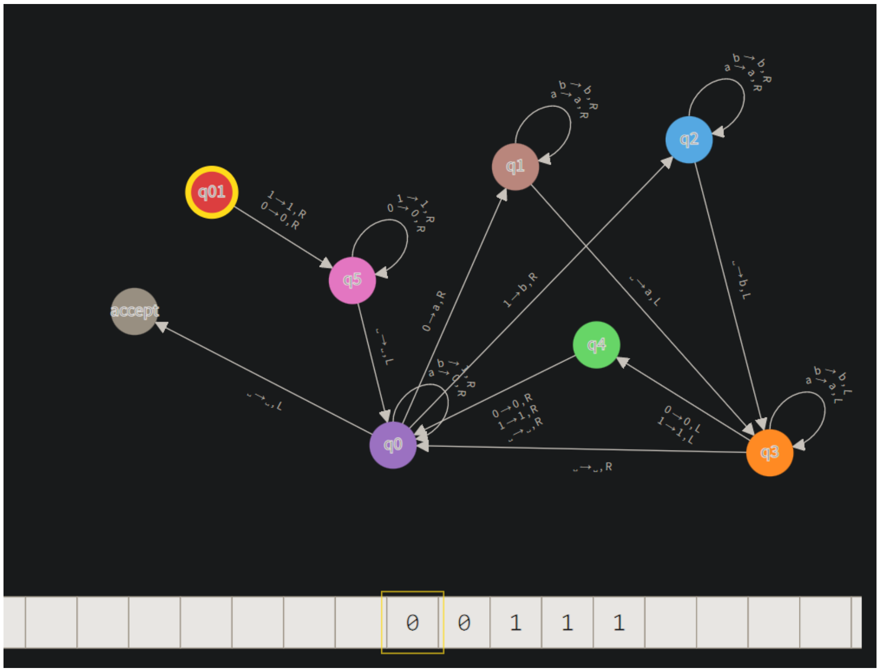
Initial state of 1010001:



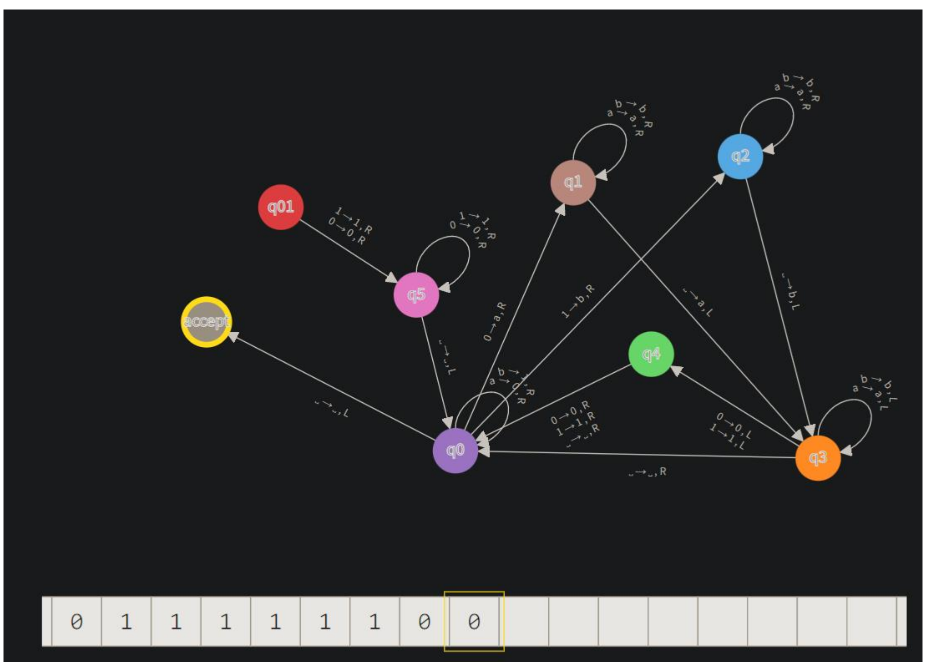
Final state of 1010001:



Initial state of 00111:



Final state of 00111:



Question 3:

We will define a Two-Dimensional (2D) Turing Machine as a TM with a two-dimensional tape. While reading and writing symbols or changing states are the same as the original TM, the 2D machine has capability to move left, right, up or down. So the transition function is now defined as:

$\delta : Q x \Sigma - > Q x \Sigma x \{L, R, U, D\}$

A Turing machine M is polynomial time if ∀ inputs I of length n for sufficiently large n there exist c1, c2 such that M(I) halts in less than $c1 \cdot n^{c2}$ steps.

A language L is in P if there exists a deterministic Turing machine M that decides L in polynomial time.