

# Student Information

Name : Emre Geçit  
ID : 2521581

## Question 1

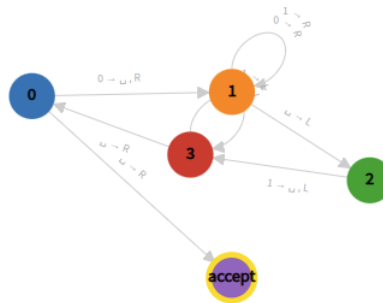
Design a Turing machine which recognizes the language  $L = \{0^N 1^N | N \geq 1\}$ .  
 $\Sigma = \{0, 1, \sqcup\}$ , means that you cannot write any other symbol than these symbols.

### States:

- State 0 (Initial state): If the tape head is on a 0, changes it with a blank space, moves the tape head to right, and passes to the state 1. If it is on a blank space, passes to the acceptance state.
- State 1: This state's task is finding the right end of the string. After finding the rightmost symbol in the string, the machine passes to state 2.
- State 2: If the tape head is on a 1, changes it with a blank space, moves the tape head to left, and passes to the state 3. If it is on a blank space, passes to the acceptance state.
- State 3: This state's task is finding the left end of the string. After finding the leftmost symbol in the string, the machine passes to state 0.

Note that in these descriptions, undefined transistions leads to rejection of the string.

Figure 1: Turing machine which recognizes the language  $L = \{0^N 1^N | N \geq 1\}$



Sample inputs:

Figure 2: Input = 000111

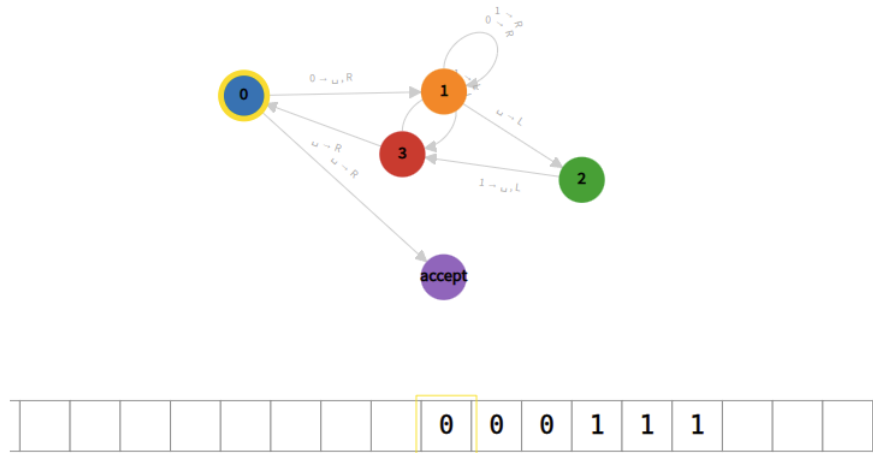


Figure 3: 000111 accepted

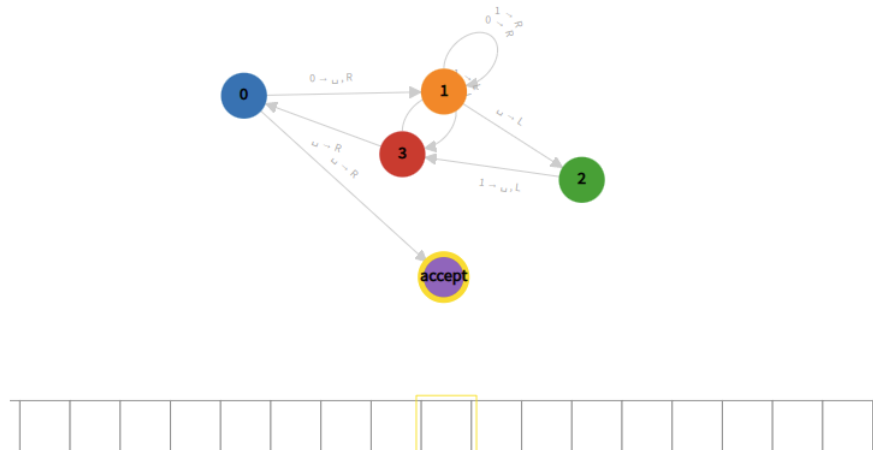


Figure 4: Input = 0000111

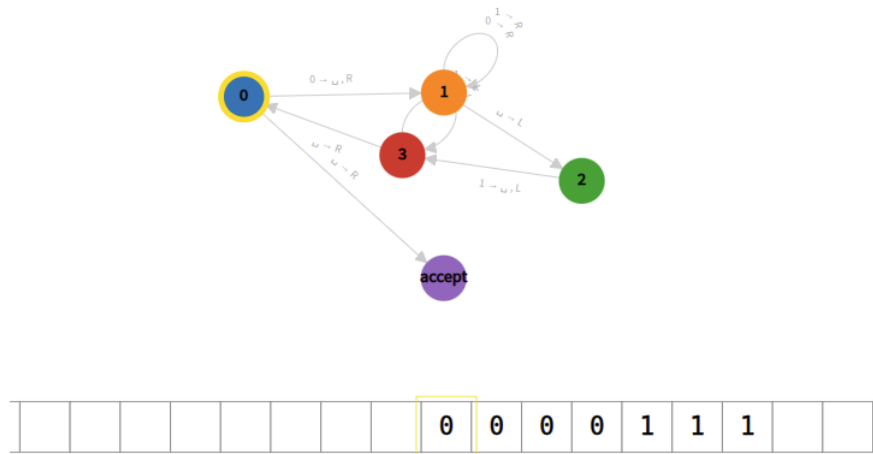


Figure 5: 0000111 rejected

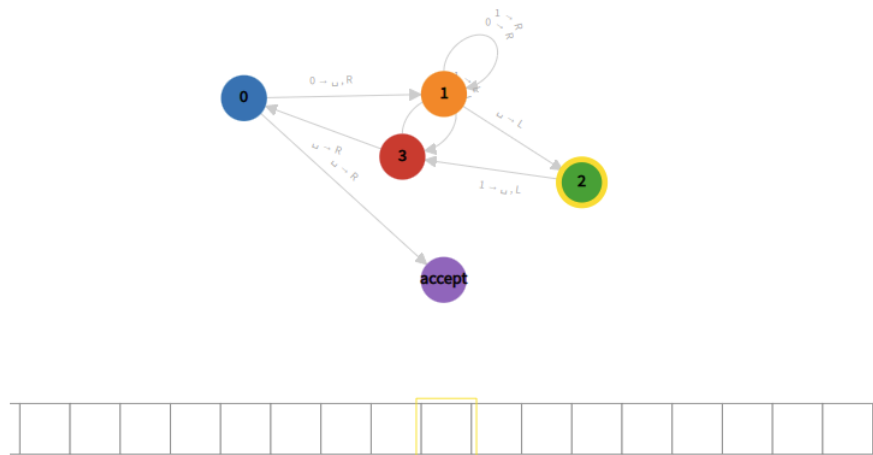


Figure 6: Input = 000011111

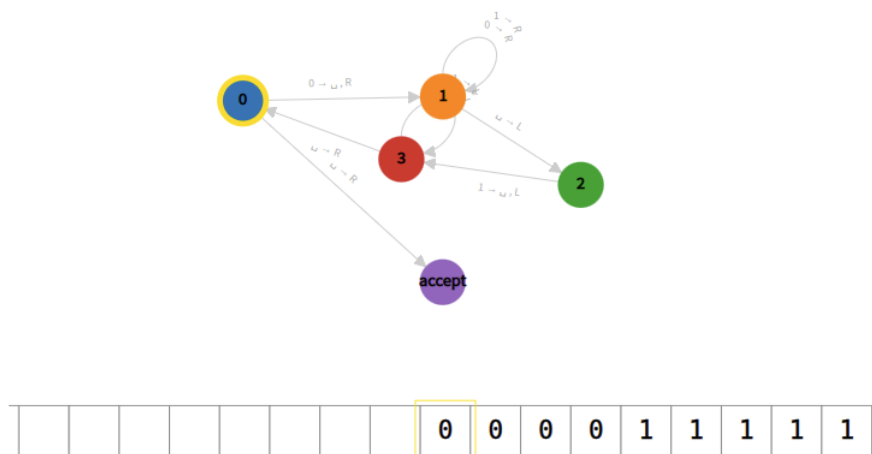


Figure 7: 000011111 rejected

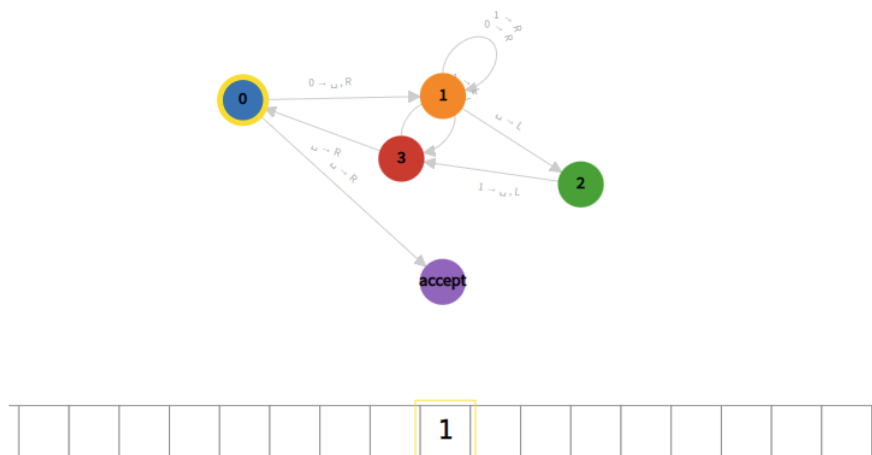


Figure 8: Input = 0001110

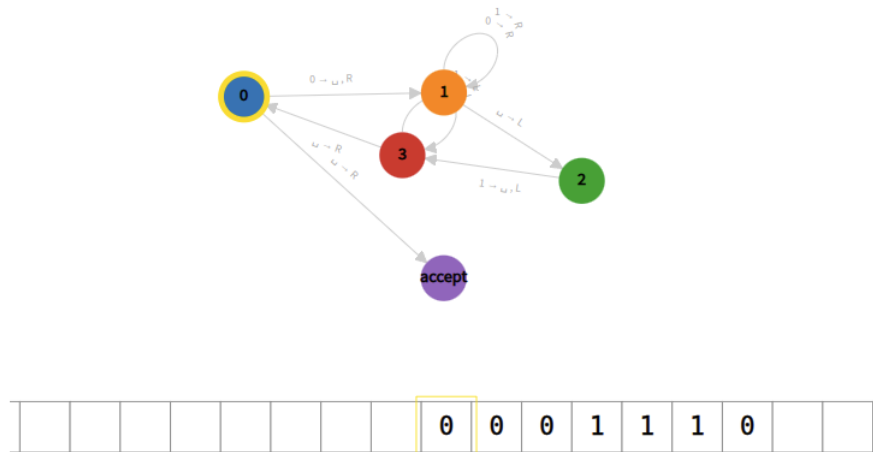


Figure 9: 0001110 rejected

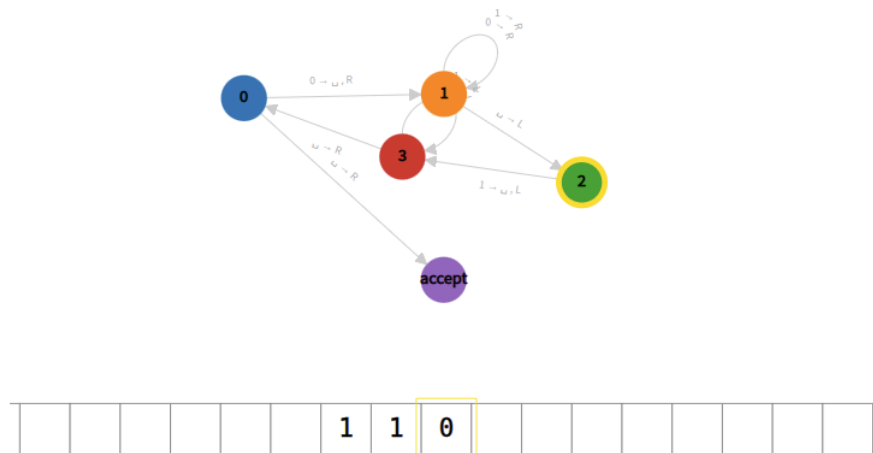


Figure 10: Input = 100011

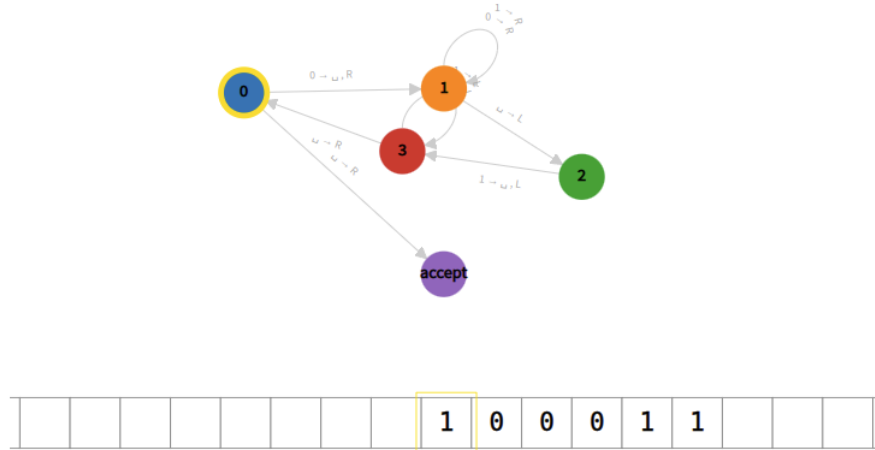
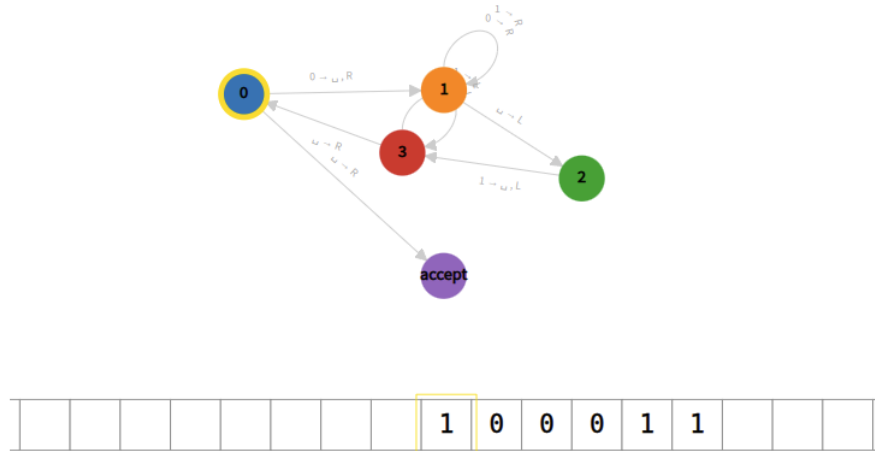


Figure 11: 100011 rejected



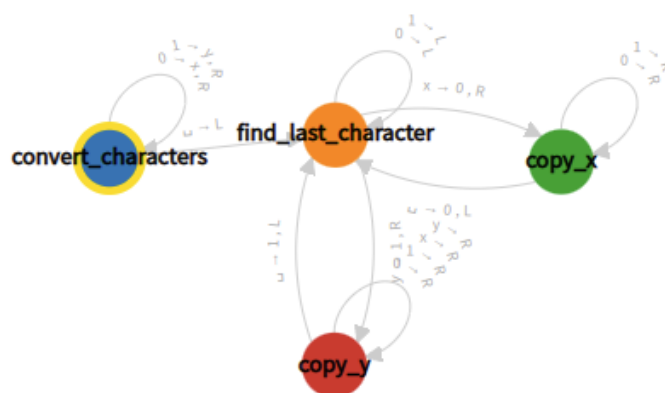
## Question 2

Design a Turing machine that computes the function  $f(w) = ww^R$ ,  $\Sigma(w) = \{0, 1\}$ .

## States:

- `convert_symbols` (Initial state): This state changes all zeros with x's, and all ones with y's. Then, it moves the tape head to the right.
- `find_last_character`: This state finds the last symbol that has not been copied and passes to `copy_y` if the tape head is on a y, or to `copy_x` if the tape head is on a x.
- `copy_y`: This state goes to the end of the tape, and writes a 1.
- `copy_x`: This state goes to the end of the tape, and writes a 0.

Figure 12: Turing machine which computes the function  $f(w) = ww^R$



Sample inputs:

Figure 13: Input = 1011

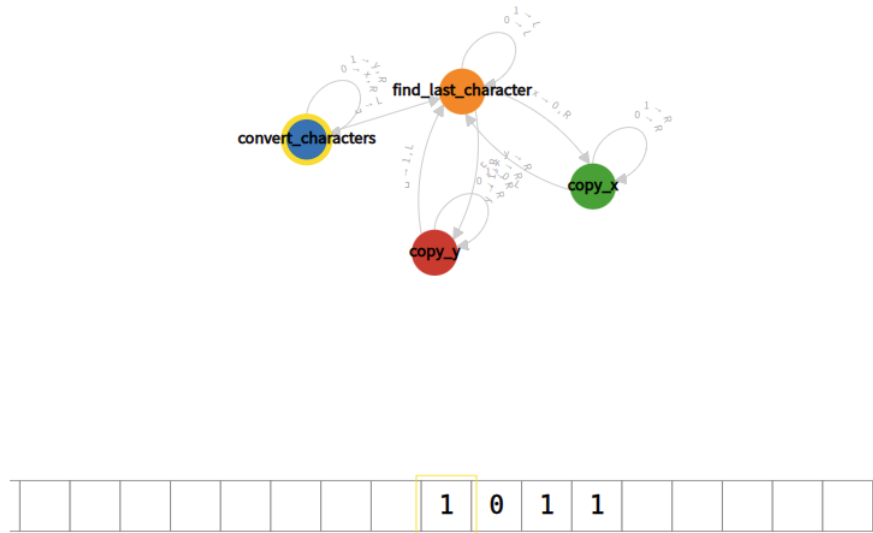


Figure 14: Output = 10111101

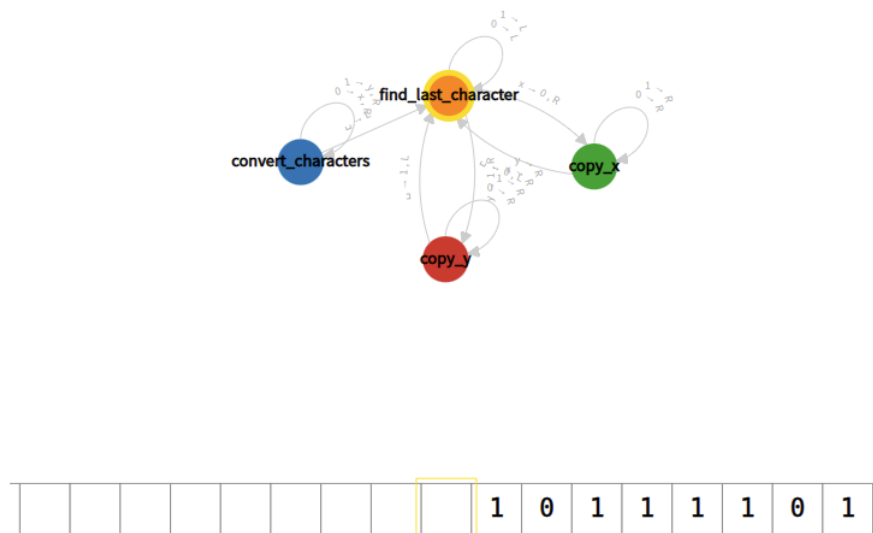




Figure 15: Input = 1110

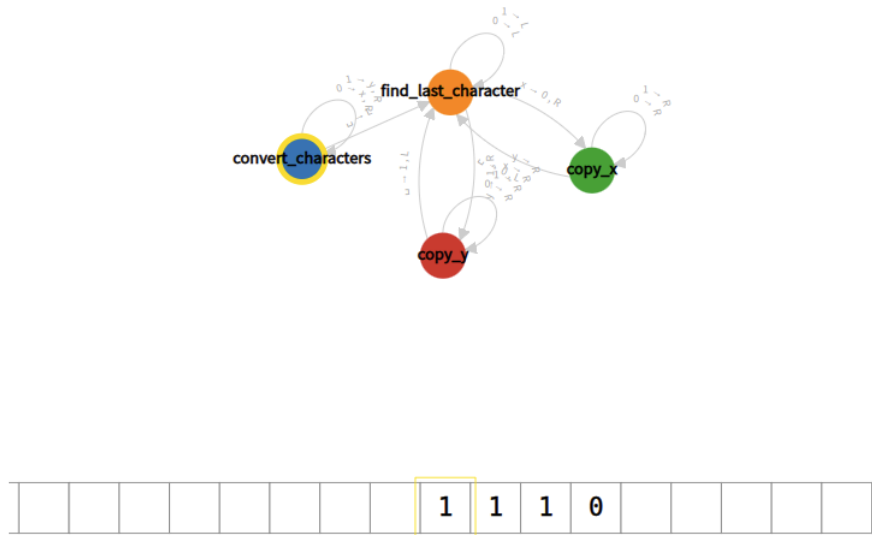


Figure 16: Output = 11100111

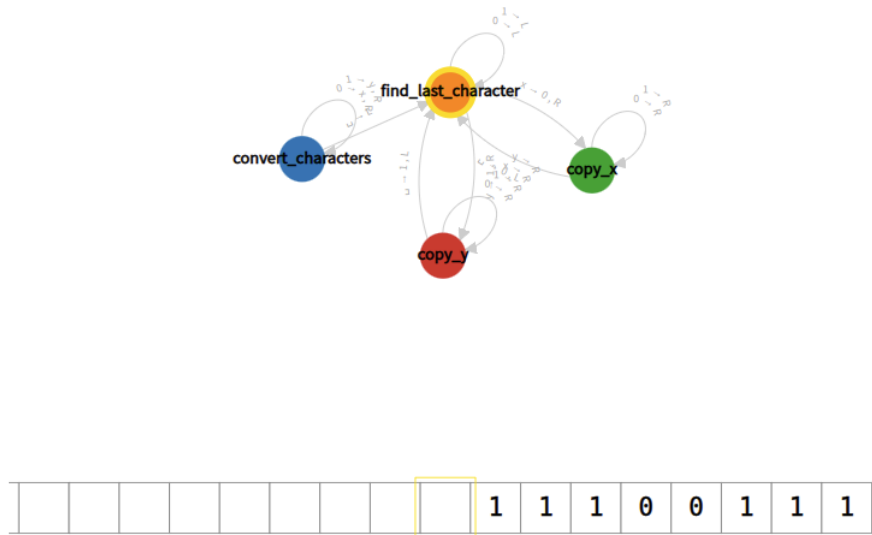


Figure 17: Input = 0101

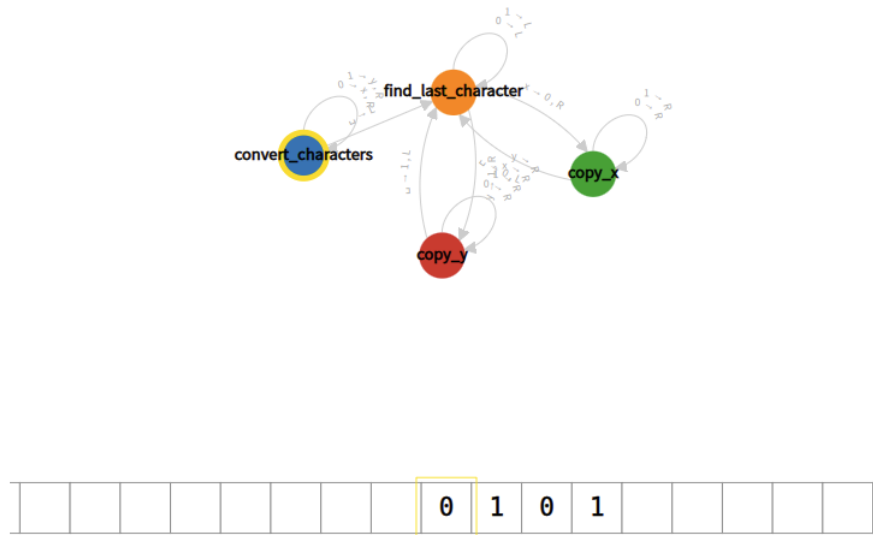


Figure 18: Output = 01011010

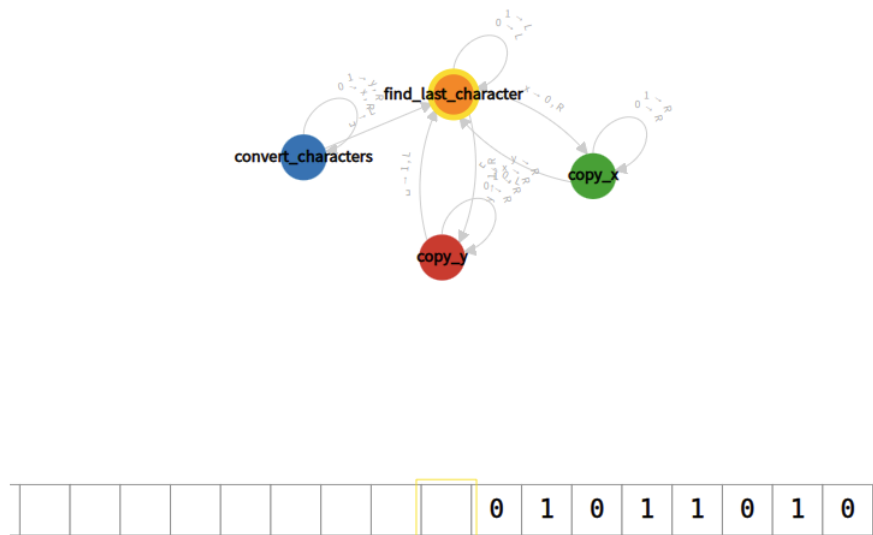


Figure 19: Input = 1010

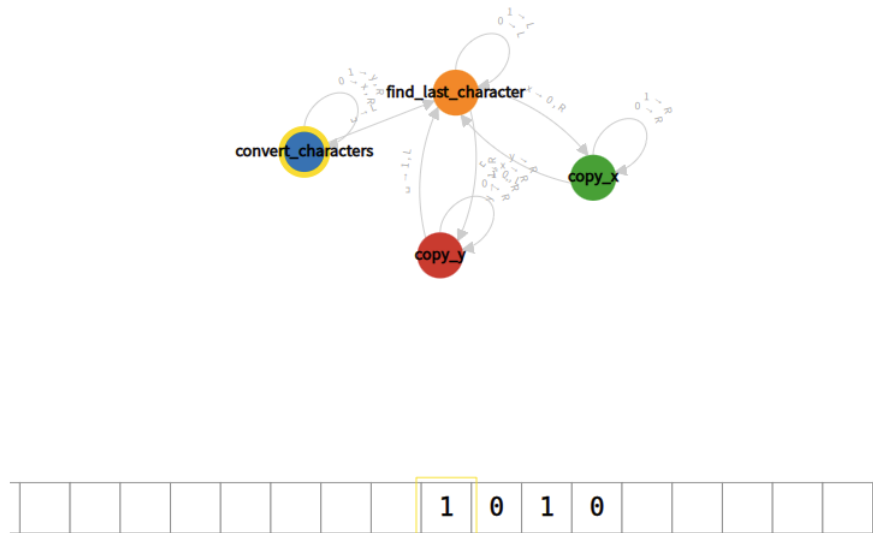


Figure 20: Output = 10100101

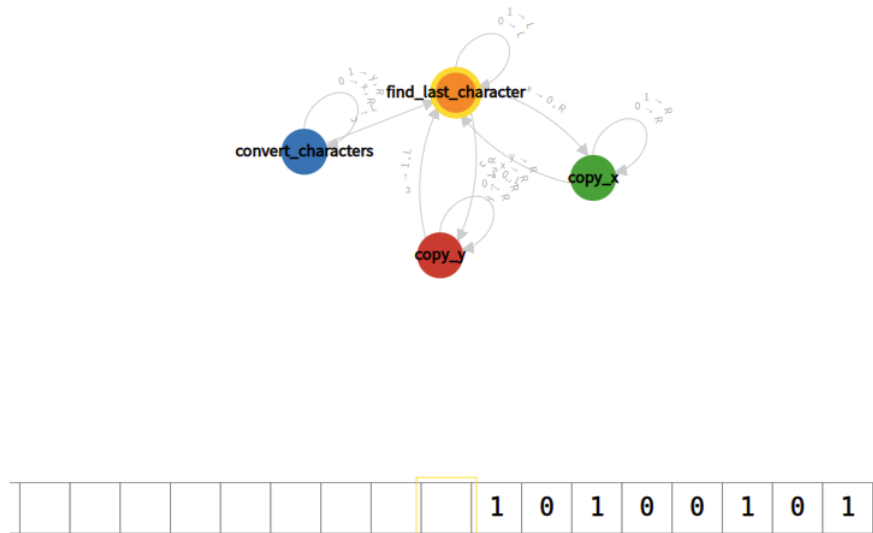


Figure 21: Input = 1010001

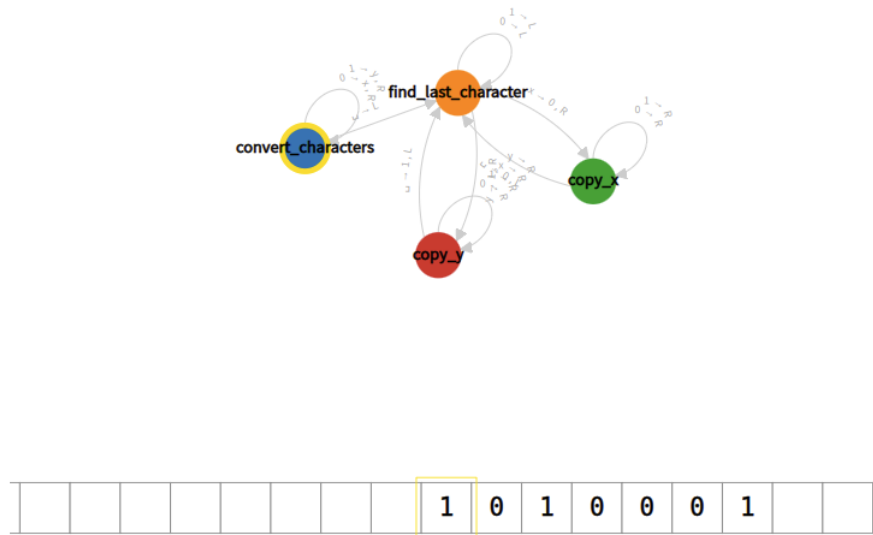


Figure 22: Output = 10100011000101

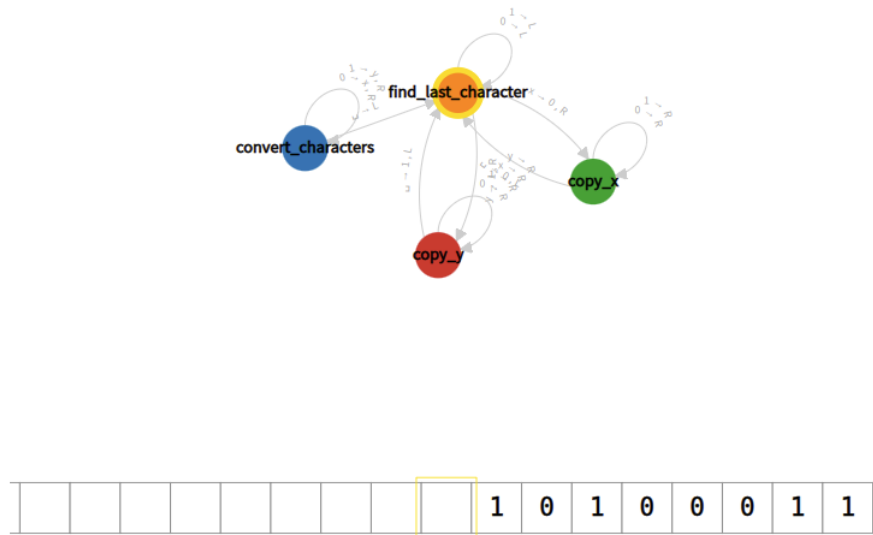


Figure 23: Input = 00111

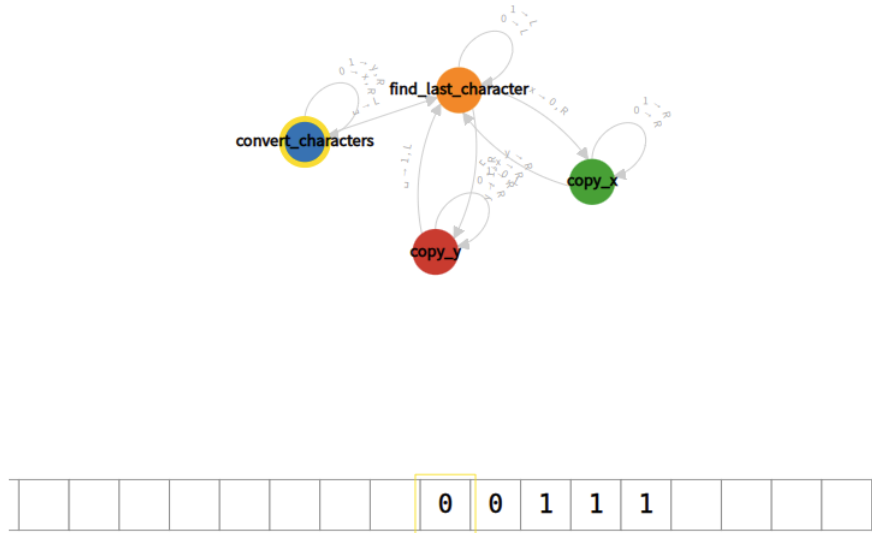
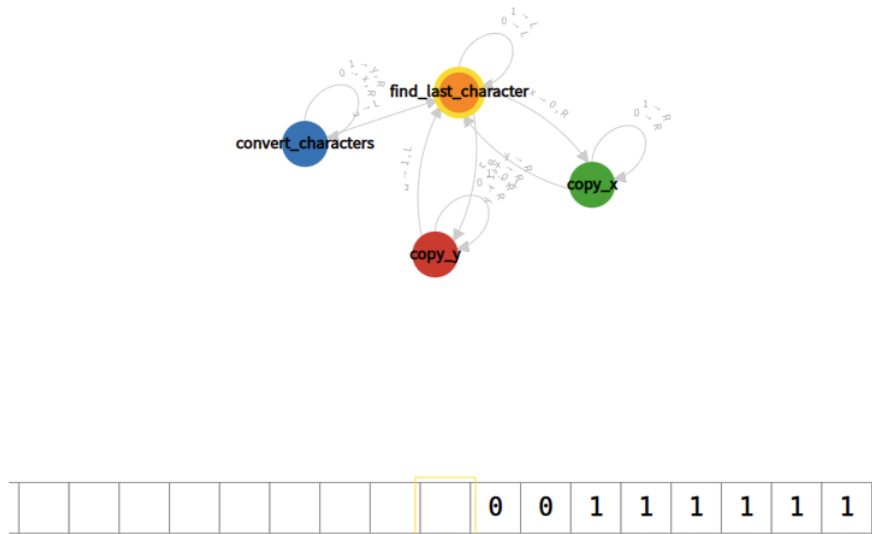


Figure 24: Output = 0011111100



## Question 3

Formally define a Turing machine with a 2-dimensional tape, its configurations, and its computation. Define what it means for such a machine to decide a language  $L$ . Show that  $t$  steps of this machine, starting on an input of length  $n$ , can be simulated by a standard Turing machine in time that is polynomial in  $t$  and  $n$ .

### Solution:

1) A Turing machine with a 2-dimensional tape is a quintuple  $(K, \Sigma, \delta, s, H)$ , where

- $K$  is a finite set of states;
- $\Sigma$  is a finite set of symbols containing the blank symbol  $\sqcup$  but not containing  $\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow$ ;
- $s \in K$  is the initial state;
- $H \subseteq K$  is the set of halting states;
- $\delta$ , the transition function,  $\delta : (K - H) \times \Sigma \rightarrow (K - H) \times (\Sigma \cup \{\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow\})$ ;

2) A Turing machine with a 2-dimensional  $M$  is said to decide a language  $L$  if  $L = L(M)$  and  $M$  halts on every input.???????????

3) First of all, we should define the mechanics of the classical Turing machine  $M'$ , that will simulate our 2-dimensional machine  $M$ .

The tape of  $M'$  must somehow contain all information in all tapes of  $M$ . A simple way of achieving this is by thinking that the  $M'$  is divided into several tracks, with each "track" devoted to the simulation of a row of the 2d-tape.