Student Information

Name: Burak Metehan Tunçel
ID: 2468726

Answer 1

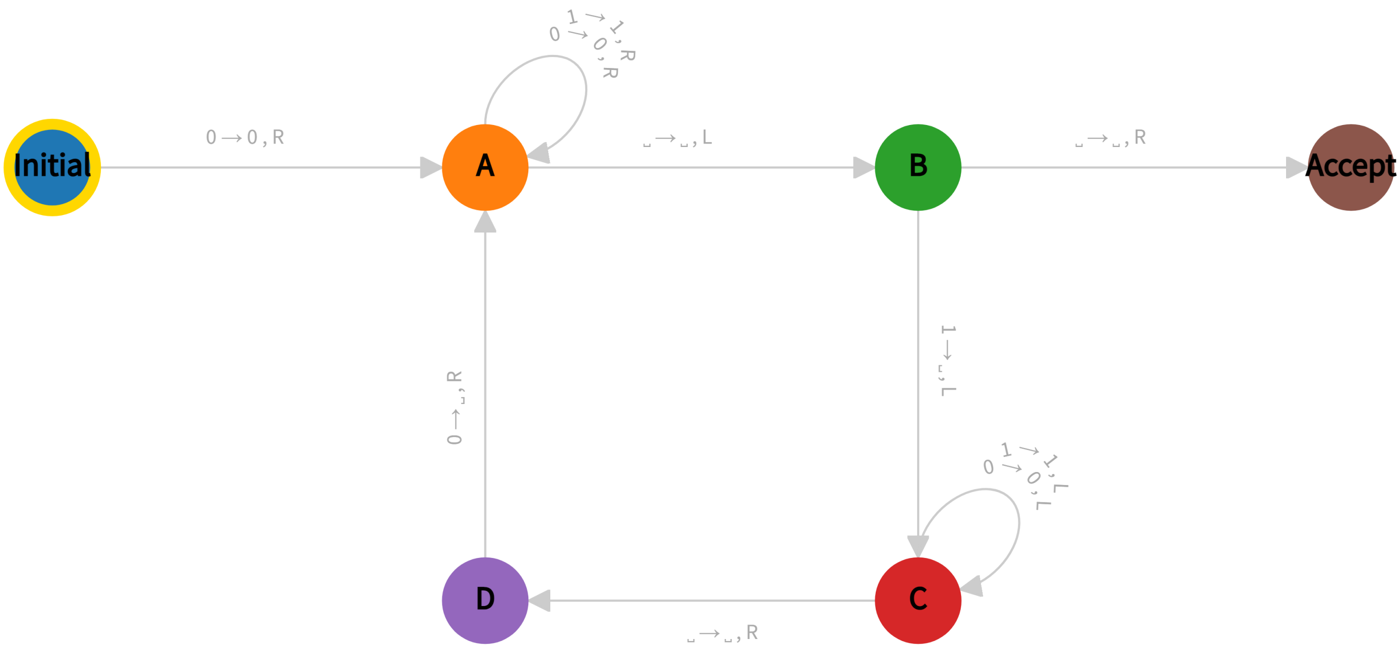

Figure 1: *Screenshot of the Turing Machine designed for Q1*

Descriptions of States

State: Initial

It is the start state. If string starts with **1** or ⊔, machine immediately stops in this state. If string starts with **0**, head is moved to **R** and state of machine goes to state **A**.

State: A

It scans to the right for the first ⊔.

- If ⊔ is found, head is moved one left and state of machine goes to state **D**.
- If ⊔ is not found, head is moved one right and state of machine is still in **A**.

State: B

It is state indicating that the head is in rightmost **1** *or input is correct and it should be accepted.*

- If **1** is read, ⊔ is written and head is moved to left. State of machine goes to state **C**.
- If ⊔ is read, it indicates that the input is correct and should be accepted. Therefore, state of machine goes to state **Accept**.

State: C

It scans to the left for the first ⊔.

- If ⊔ is found, head is moved one right and state of machine goes to state **D**.
- If ⊔ is not found, head is moved one left and state of machine is still in **C**.

State: D

It is state indicating that the head is in leftmost **0**.

- If **0** is read, ⊔ is written and head is moved to right. State of machine goes to state **A**.
- If **0** is not read, it indicates that the input is not correct and shouldn't be accepted. Therefore, state of machine is not changed and crash.

State: Accept

It is state indicating that the input is correct and accepted. Therefore, the machine stops, successfully.

Input Samples

Input: 000111 | Accept

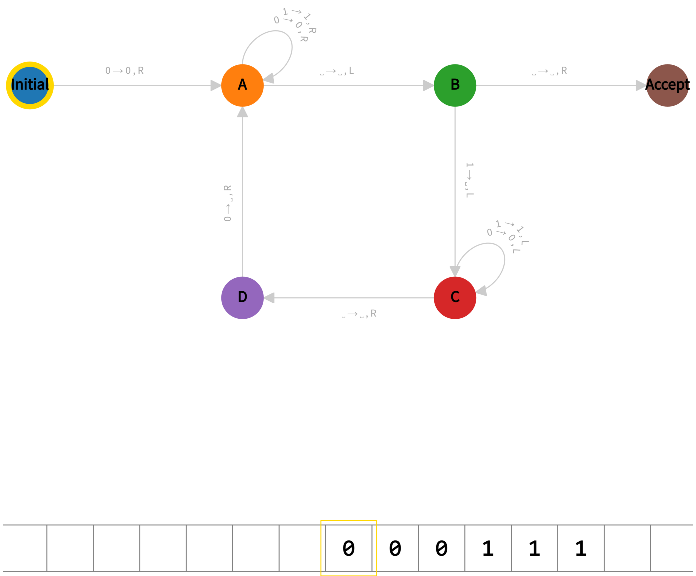

Figure (a): Initial State for **000111**

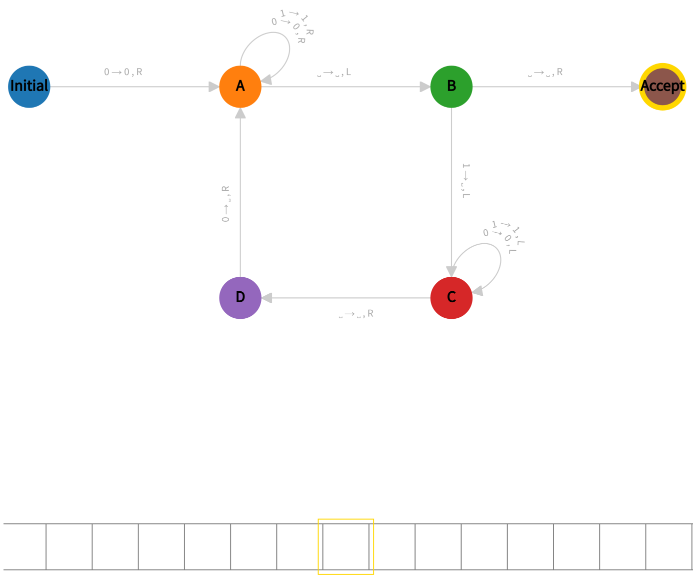

Figure (b): End State for **000111**

Figure 2: States for **000111**

Input: 0000111 | Reject

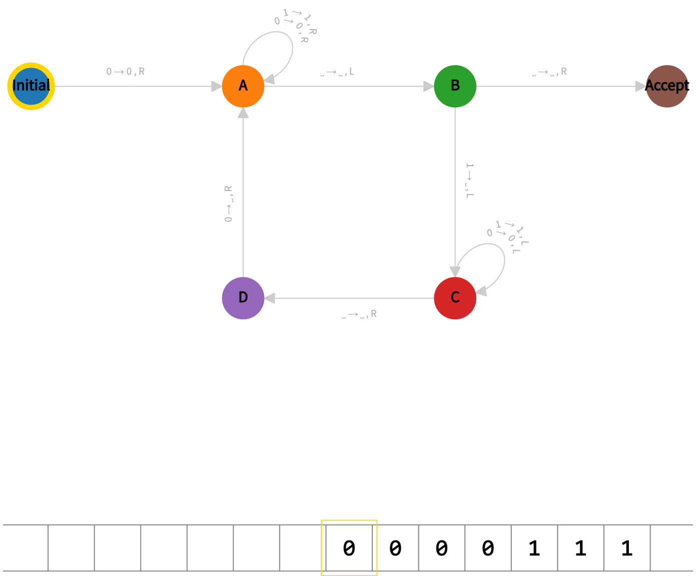

Figure (a): Initial State for **0000111**

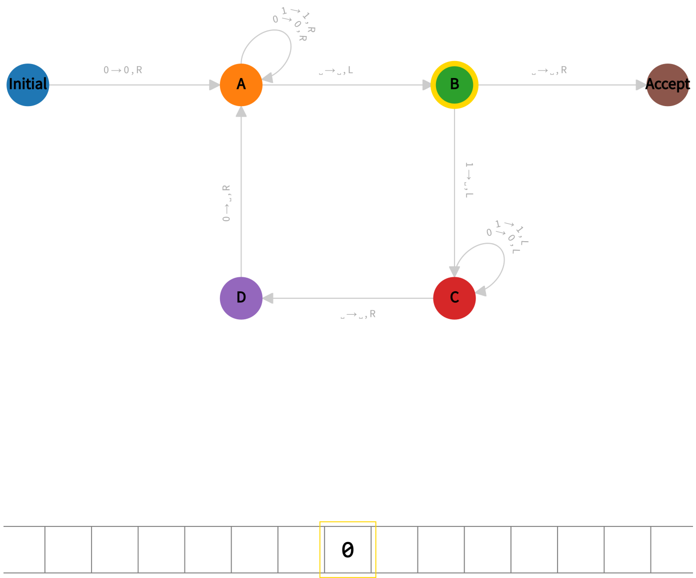

Figure (b): End State for **0000111**

Figure 3: States for **0000111**

Input: 000011111 | Reject

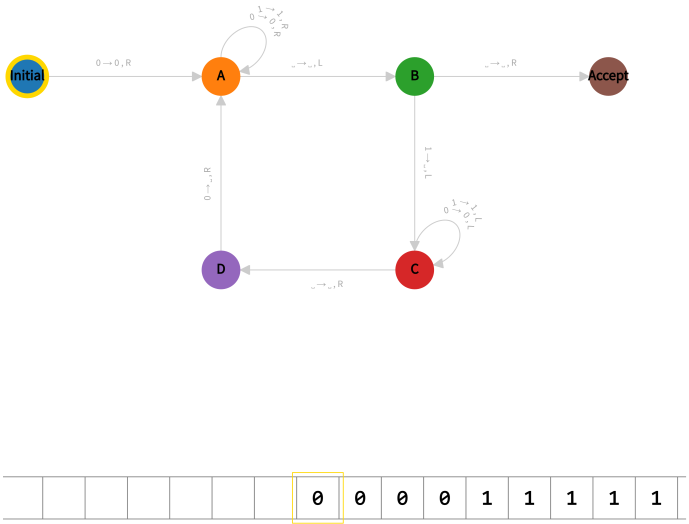

Figure (a): Initial State for **000011111**

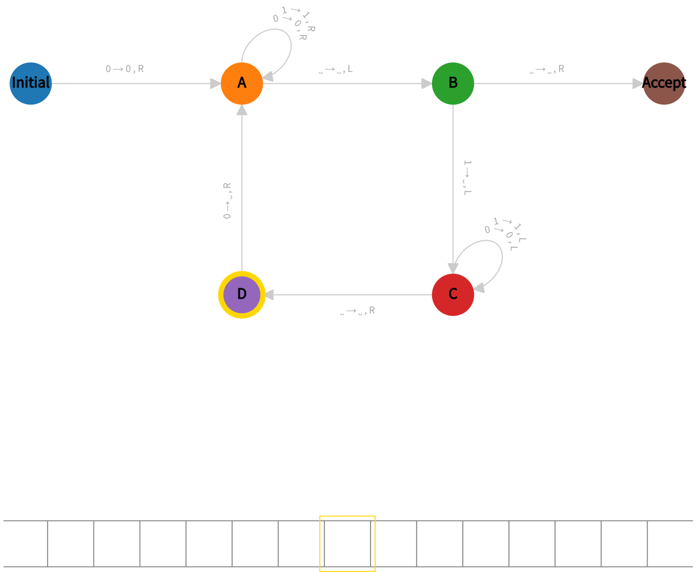

Figure (b): End State for **000011111**

Figure 4: States for **000011111**

Input: 0001110 | Reject

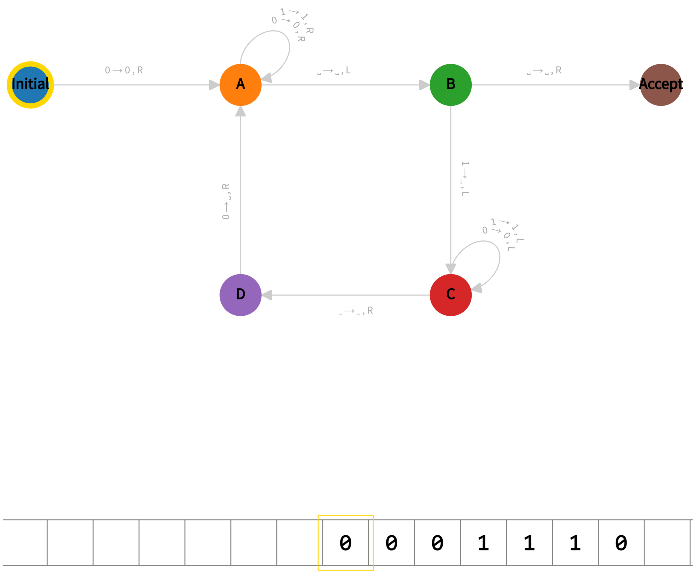

Figure (a): Initial State for **0001110**

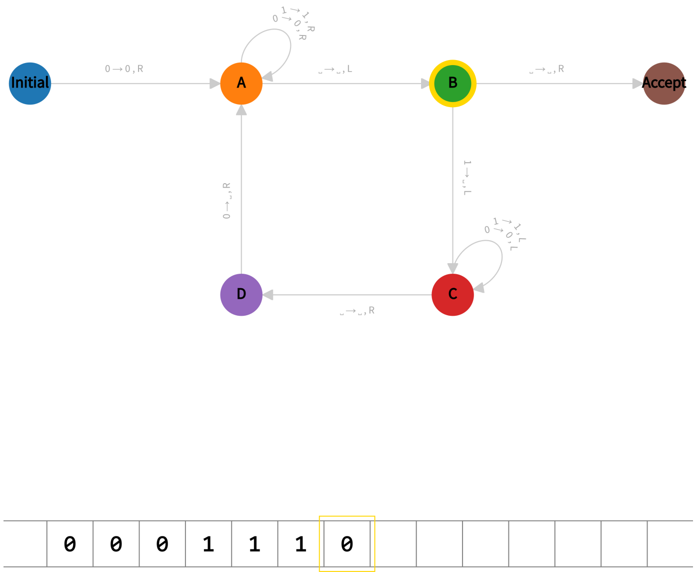

Figure (b): End State for **0001110**

Figure 5: States for **0001110**

Input: 001101 | Reject

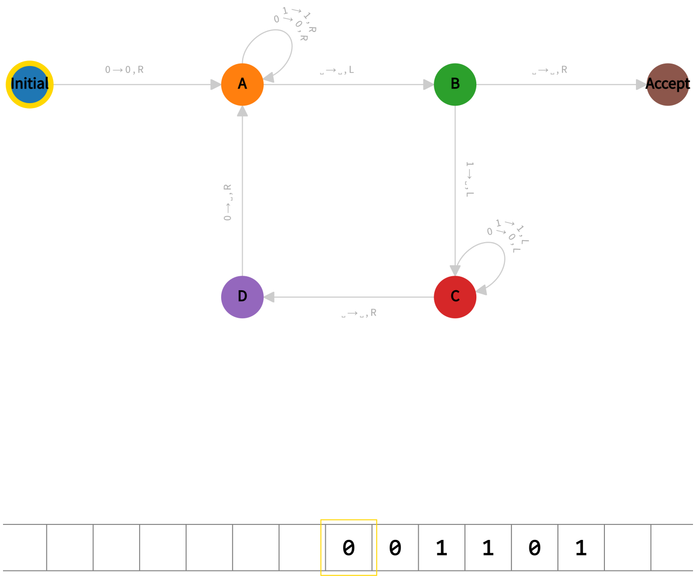

Figure (a): Initial State for **001101**

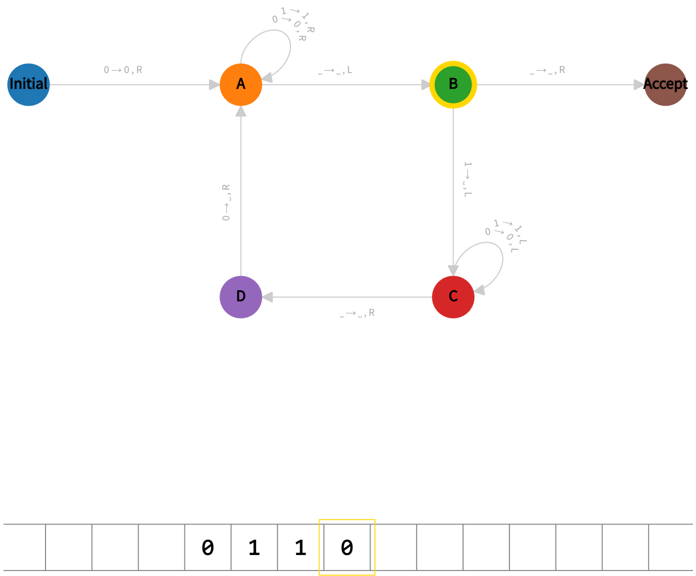

Figure (b): End State for **001101**

Figure 6: States for **001101**

Input: 100011 | Reject

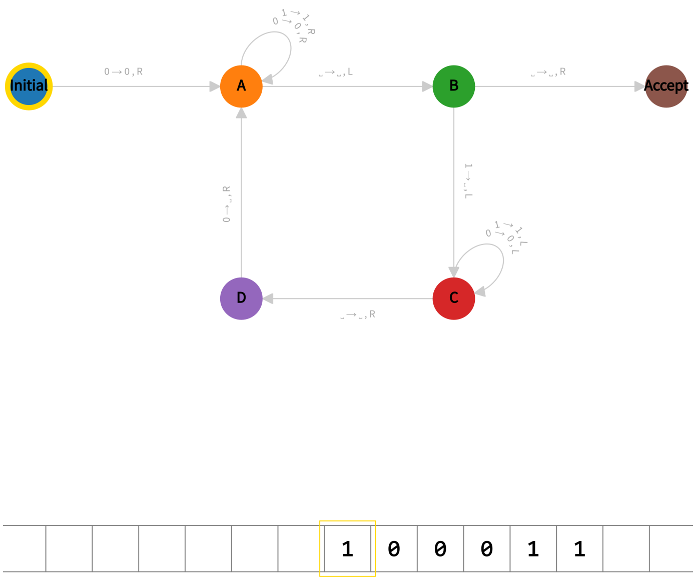

Figure (a): Initial State for **100011**

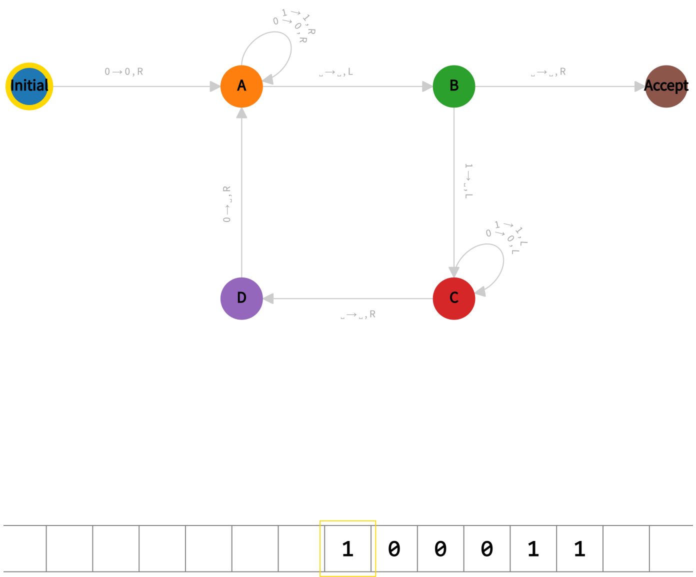

Figure (b): End State for **100011**

Figure 7: States for **100011**

Answer 2

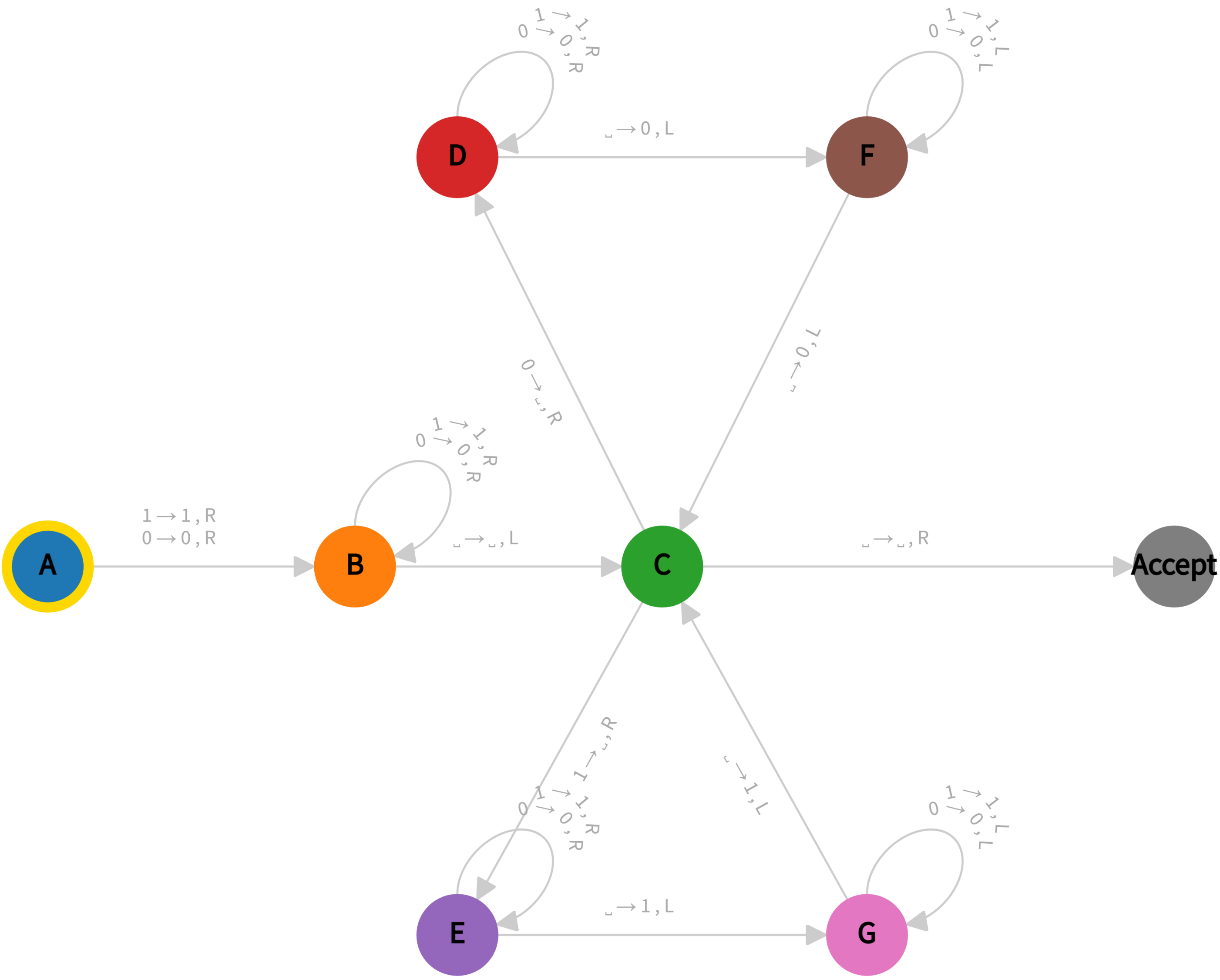

Figure 8: *Screenshot of the Turing Machine designed for Q2*

Descriptions of States

State: A

It is the start state. If string starts with $\sqcup$, machine immediately stops in this state. If string starts with **0** or **0**, head is moved to right and state of machine goes to state **B**.

State: B

It scans to the right for the first $\sqcup$. When $\sqcup$ is found, head is moved to left and state of machine goes to state **C**. It is used only once to find the last symbol in the string.

State: C

It is the state indicating that the symbol below the head needs to be put the end of string; that is, head is on the symbol that will be copied to the end of the string if state of machine is in state **C**. In this state,

- If **0** is read, a $\sqcup$ is temporarily placed and head is moved right and state of machine goes to state **D** (The upper part of states, as shown in Figure 8. This part remembers the symbol was **0**).
- If **1** is read, a $\sqcup$ is temporarily placed and head is moved right and state of machine goes to state **E** (The lower part of states, as shown in Figure 8. This part remembers the symbol was **1**).
- If $\sqcup$ is read, it means that computing is done without any error or crash. State of the machine goes to state **Accept**

State: D

It scans to the right for the first $\sqcup$. The important part is that it knows (symbol that will be copied was **0**) symbol will be written is **0**. When $\sqcup$ is found,

- **0** is placed,
- head is moved left, and
- state of the machine goes to state **F**.

State: F

It scans to the left for the $\sqcup$ that is placed temporarily. The important part is that it knows (copied symbol was **0**) symbol will be written is **0**. When $\sqcup$ is found,

- **0** is reinserted to its old place,
- head is moved left (next is symbol that will be copied or a $\sqcup$), and
- state of the machine goes to state **C**.

State: E

It scans to the right for the first $\sqcup$. The important part is that it knows (symbol that will be copied was **1**) symbol will be written is **1**. When $\sqcup$ is found,

- **1** is placed,
- head is moved left, and
- state of the machine goes to state **G**.

State: G

It scans to the left for the $\sqcup$ that is placed temporarily. The important part is that it knows (copied symbol was **1**) symbol will be written is **1**. When $\sqcup$ is found,

- **1** is reinserted to its old place,
- head is moved left (next is symbol that will be copied or a $\sqcup$), and
- state of the machine goes to state **C**.

State: Accept

It is state indicating that computing is done without any error or crash. Therefore, the machine stops, successfully.

Input Samples

Input: 1011

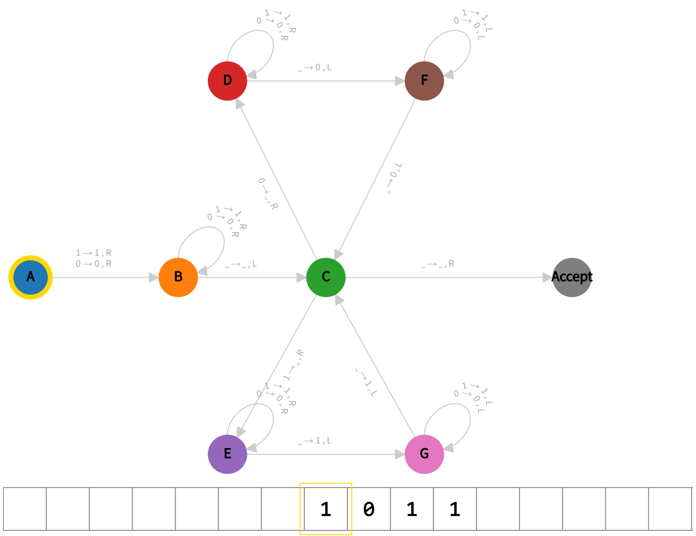

Figure (a): Initial State for **1011**

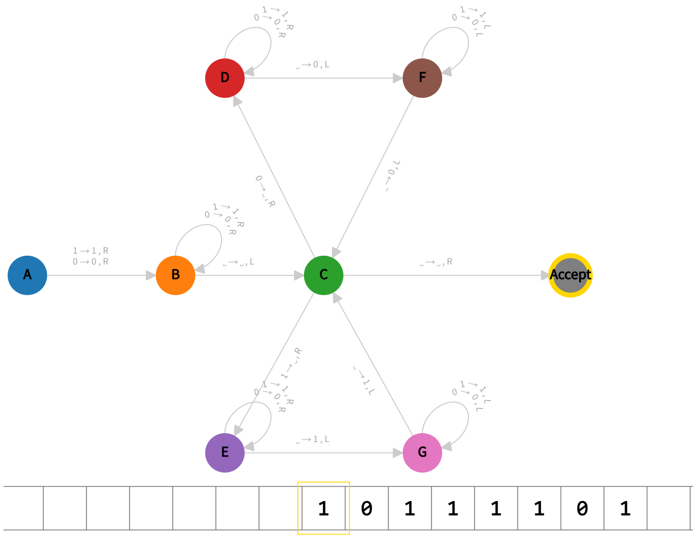

Figure (b): End State for **1011**

Figure 9: States for **1011**

Input: 1110

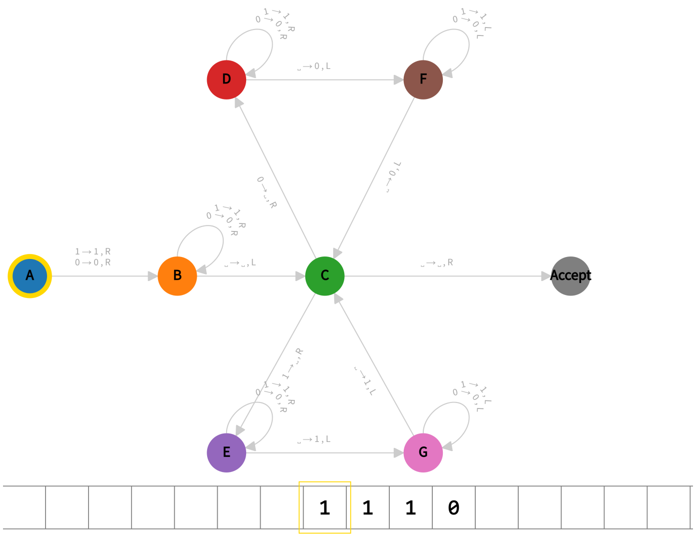

Figure (a): Initial State for **1110**

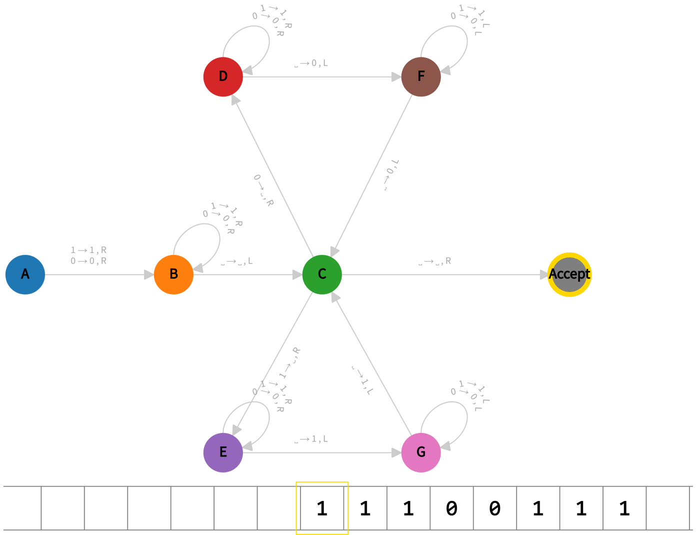

Figure (b): End State for **1110**

Figure 10: States for **1110**

Input: 0101

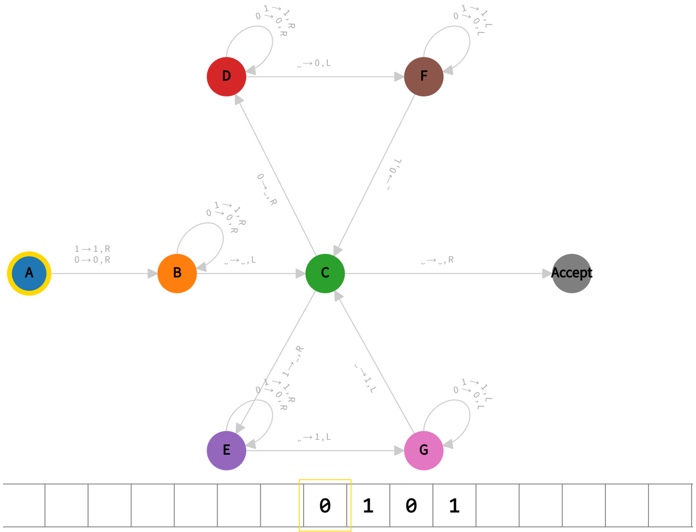

Figure (a): Initial State for **0101**

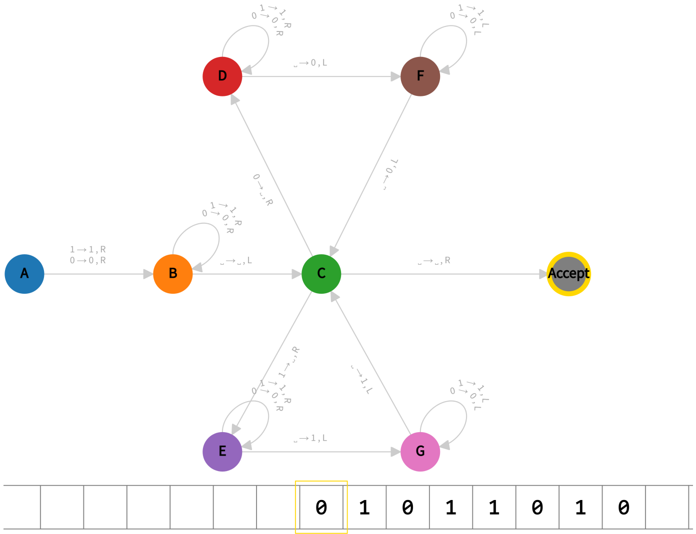

Figure (b): End State for **0101**

Figure 11: States for **0101**

Input: 1010

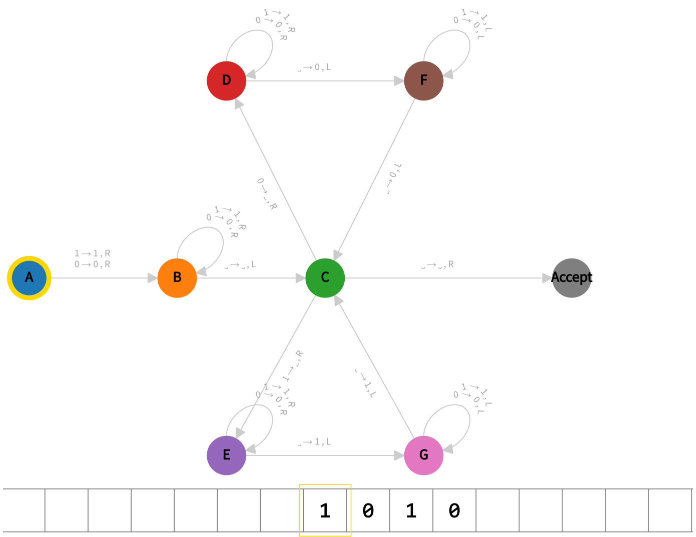

Figure (a): Initial State for **1010**

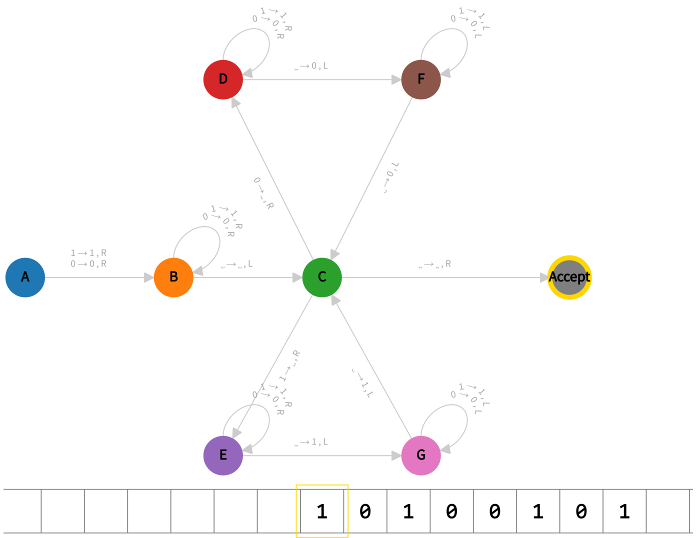

Figure (b): End State for **1010**

Figure 12: States for **1010**

Input: 1010001

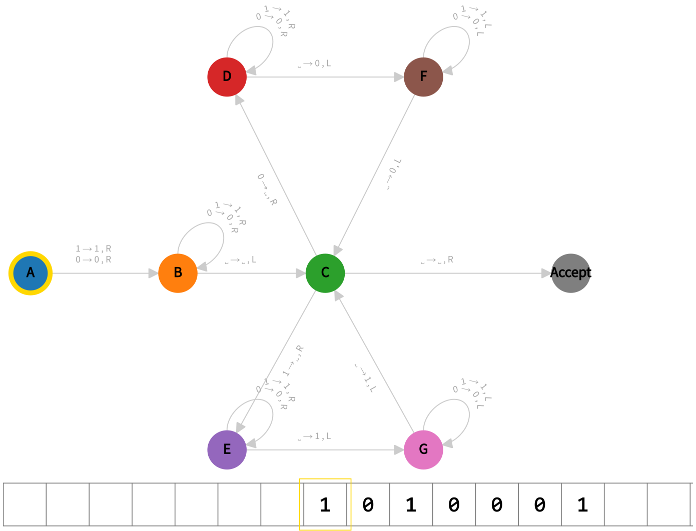

Figure (a): Initial State for **1010001**

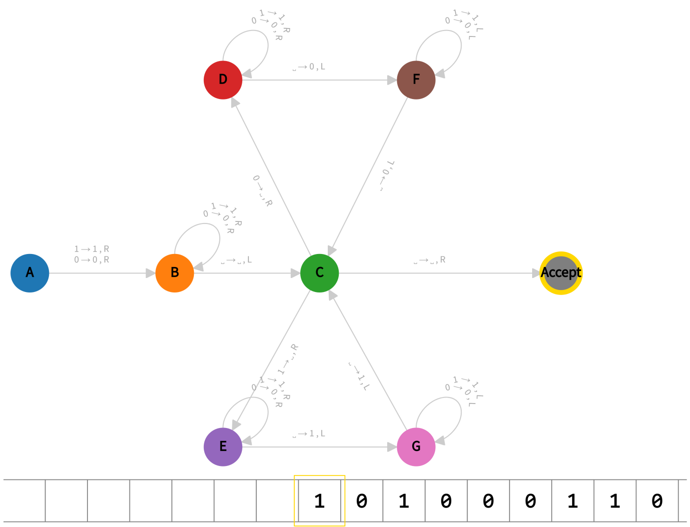

Figure (b): End State for **1010001**

Figure 13: States for **1010001**

Due to the head place, the whole output cannot be seen in Figure 13.b.

Input: 00111

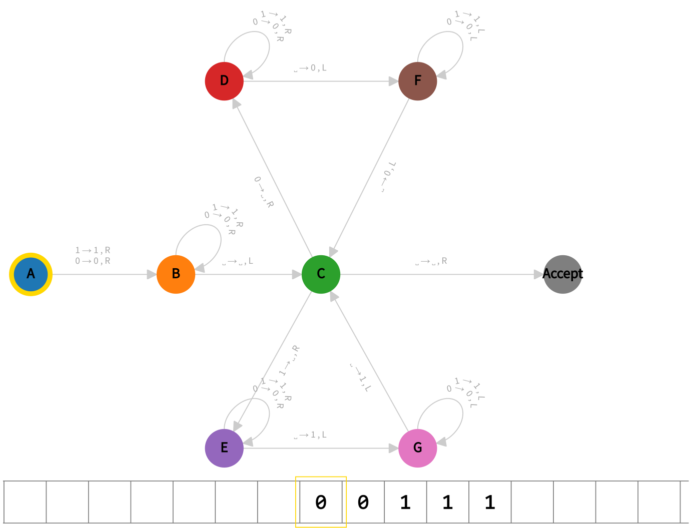

Figure (a): Initial State for **00111**

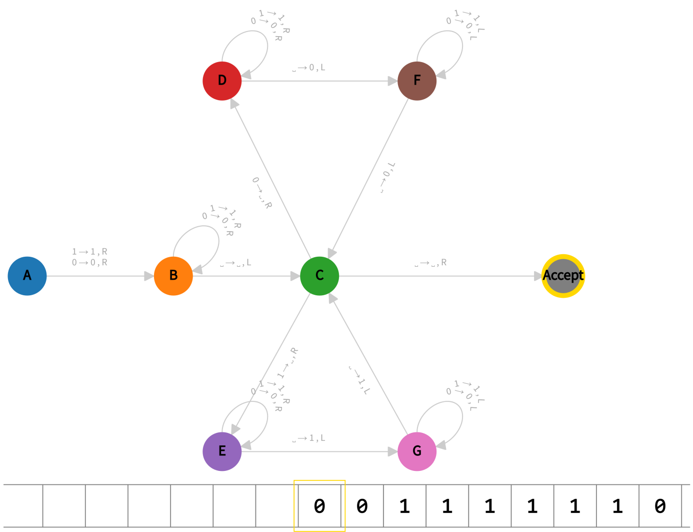

Figure (b): End State for **00111**

Figure 14: States for **00111**

Due to the head place, the whole output cannot be seen in Figure 14.b.

Answer 3

Let $M_{2D} = (K, \Sigma, \delta, s, H)$ be a 2-dimensional tape where

- K is the finite set of states
- Σ is the input alphabet with $\sqcup$
- $s \in K$, initial state
- $H \subseteq K$, halting states
- $\delta \ : \ ((K - H) \times \Sigma) \mapsto K \times (\Sigma \cup \{\leftarrow, \rightarrow, \uparrow, \downarrow\})$ is the transition function.

Configuration

- Let the input string w be of the form:

$$\begin{matrix} & \triangledown & \triangledown & \triangledown & \cdots \\ \triangleright & w_{11} & w_{12} & w_{13} & \cdots \\ \triangleright & w_{21} & w_{22} & w_{23} & \cdots \\ \triangleright & w_{31} & w_{32} & w_{33} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix}$$

- Also, let the i^{th} row is denoted as w_i, j^{th} column is denoted as w_j and let $\underline{w_{ij}}$ show that head of tape is on symbol w_{ij}.

So, configuration can be shown as follows,

$$\left(q, \begin{matrix} & \triangledown & \triangledown & \triangledown & \triangledown & \cdots \\ \triangleright & w_{11} & \cdots & w_{1j} & \cdots & \cdots \\ \triangleright & \vdots & \ddots & \cdots & \cdots & \cdots \\ \triangleright & w_{i1} & \vdots & \underline{w_{ij}} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix} \right) \text{ or } (q, \sigma)$$

where q is the state of the machine, σ is the symbol that is located on w_{ij} and head is on the w_{ij}.
As an example, consider the following:

$$\left(q, \begin{matrix} & \triangledown & \triangledown & \triangledown \\ \triangleright & a & a & b \\ \triangleright & a & \underline{b} & a \\ \triangleright & b & b & b \end{matrix} \right)$$

where q is the state of the machine, the symbol that is located on w_{ij} is b and head is on the this b.

Computation

Computation may be done as follows:

$$\left(q, \begin{matrix} & \triangledown & \triangledown & \triangledown & \triangledown & \cdots \\ \triangleright & w_{11} & \cdots & w_{1j} & \cdots & \cdots \\ \triangleright & \vdots & \ddots & \cdots & \cdots & \cdots \\ \triangleright & w_{i1} & \vdots & \underline{w_{ij}} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix} \right) \vdash_{M_{2D}} \left(p, \begin{matrix} & \triangledown & \triangledown & \triangledown & \triangledown & \cdots \\ \triangleright & w_{11} & \cdots & w_{1j} & \cdots & \cdots \\ \triangleright & \vdots & \ddots & \cdots & \cdots & \cdots \\ \triangleright & w_{i1} & \vdots & \underline{w'_{ij}} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{matrix} \right) \text{ where } \delta(q, w_{ij}) = (p, w'_{ij})$$

As an example, consider the following:

$$\left(q, \begin{array}{cccc} & \triangledown & \triangledown & \triangledown \\ \triangleright & a & a & b \\ \triangleright & a & \underline{b} & a \\ \triangleright & b & b & b \end{array}\right) \vdash_{M_{2D}} \left(q, \begin{array}{cccc} & \triangledown & \triangledown & \triangledown \\ \triangleright & a & a & b \\ \triangleright & a & \underline{a} & a \\ \triangleright & b & b & b \end{array}\right) \text{ where } \delta(q,b) = (p,a)$$

$$\left(q, \begin{array}{cccc} & \triangledown & \triangledown & \triangledown \\ \triangleright & a & a & b \\ \triangleright & a & \underline{b} & a \\ \triangleright & b & b & b \end{array}\right) \vdash_{M_{2D}} \left(q, \begin{array}{cccc} & \triangledown & \triangledown & \triangledown \\ \triangleright & a & a & b \\ \triangleright & a & \underline{\sqcup} & a \\ \triangleright & b & b & b \end{array}\right) \text{ where } \delta(q,b) = (p,\sqcup)$$

Meaning of Deciding a Language L

Deciding a language L for such a machinev M is the following:

If $\Sigma_0 \subseteq \Sigma - \{\sqcup, \triangleright\}$ is an alphabet and L is a language such that $L \subseteq \Sigma_0^*$. It is said that M decides a language L if one of the followings is true for any string $w \in \Sigma_0^*$:

- $w \in L \Rightarrow M$ accepts w.
- $w \notin L \Rightarrow M$ rejects w.

Showing that simulating t steps is polynomial in t and n.

In order to simulate t steps of M_{2D}, first $2D$ input should be written on a $1D$ tape with a special character, let say \$, such that:

$$\begin{array}{ccccc} \multicolumn{5}{c}{\underline{M_{2D}\text{ tape}}} \\ & \triangledown & \triangledown & \triangledown & \triangledown \\ \triangleright & a & b & a & a \\ \triangleright & b & a & & \\ \triangleright & b & b & & \\ \vdots & & & & \end{array} \qquad \begin{array}{c} \underline{M_{1D}\text{ tape}} \\ \\ \triangleright abaa\$ba\$bb\ldots \end{array}$$

Based on this scheme t steps can be simulated in polynomial time such that:

- $\leftarrow$ (**L**eft): Left move in $1D$ is similar to in $2D$ except:

  On $2D$ tape, if head is on left boundary and tries to move left, it will stay put. On $1D$ tape, if \$ is read while moving left it will stay out instead.

- $\rightarrow$ (**R**$ight$): Right move in $1D$ is similar to in $2D$ except:

  On $2D$ tape, if head moves the **R**$ight$ to the end of the input, content of $1D$ tape should be shifted from that point (in order to insert $\sqcup$'s that are not shown.)

- $\uparrow$ (**U**p): Up move can be implemented such that count the distance from the closest \$ on the left, then find the cell having similar distance from its \$.

  For ceiling boundary and tries to move $\uparrow$, it will stay put. In $1D$ tape, if the tape head is on cells before \$ character, no transition need to be defined.

- $\downarrow$ (**D**own): Similar to $\uparrow$.

All actions can be simulated in polynomial time since for a given action, at every step of the simulation either a constant time operation or $O(n^k)$ times for some $k \geq 1$. So, t steps of M_{2D} can be achieved in $t \cdot O(n^k)$ which is polynomial in t and n.