CSE-233 : Section A

Summer 2020

Deterministic Turing Machine

Reference:

i) Book2 Chapter 3.1

ii) Neso Academy

Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

Overview

Consider the following languages-

1. a^*, a^*b^*
2. a^nb^n, ww^R
3. $a^nb^nc^n$, ww

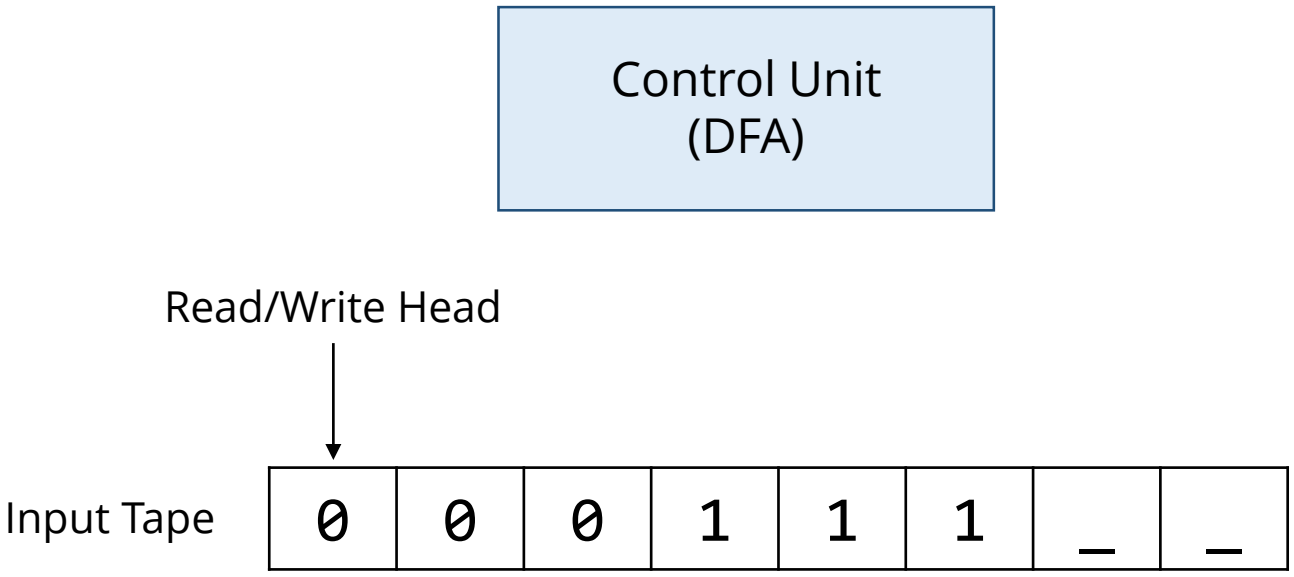
Overview

Consider the following languages-

1. a*, a*b* : can be detected using DFA/NFA
2. $a^n b^n$, ww^R : can be detected using PDA
3. $a^n b^n c^n$, ww : needs something more powerful

Overview

In a Turing machine, we have direct access to the input tape.

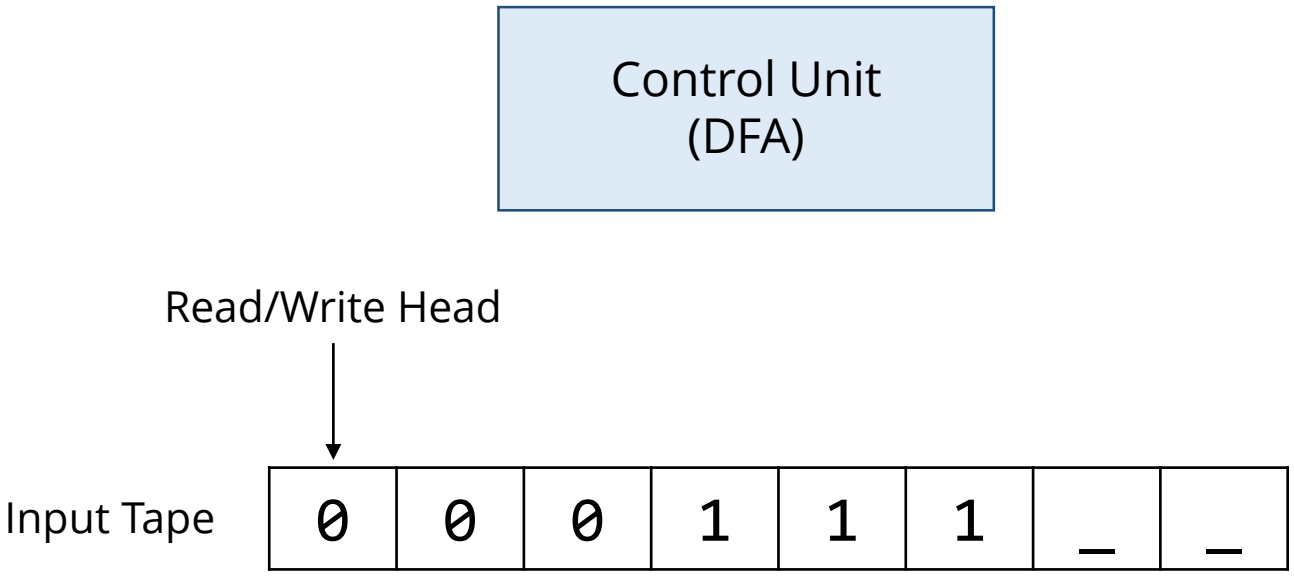


The head at each transition-

1. reads a symbol
2. writes a symbol
3. moves left/right

Overview

In a Turing machine, we have direct access to the input tape.



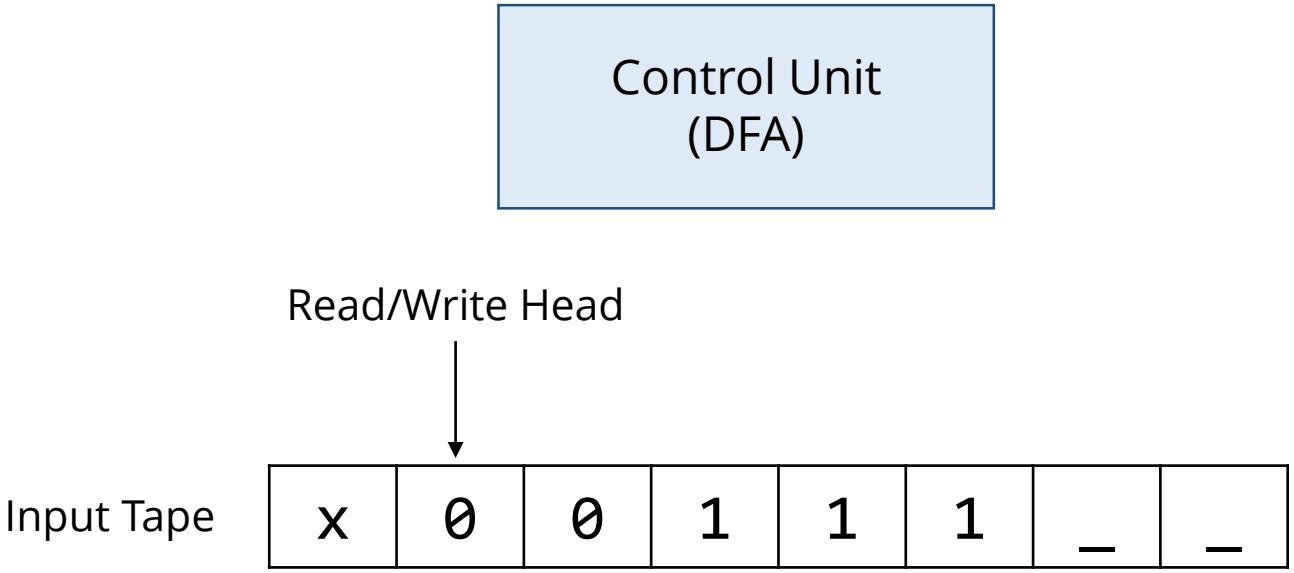
$0 \rightarrow x, R$

This means if the head is pointing at 0, we'll write x in its place,

and move the head to the right

Overview

In a Turing machine, we have direct access to the input tape.



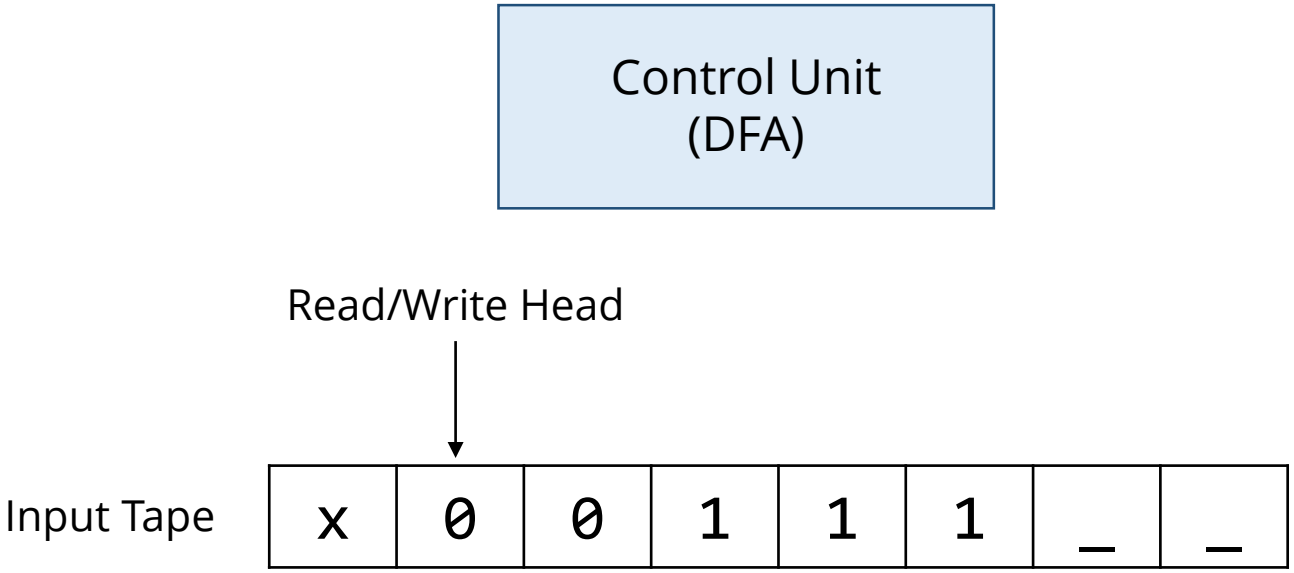
$0 \rightarrow x, R$

This means if the head is pointing at 0, we'll write x in its place,

and move the head to the right

Overview

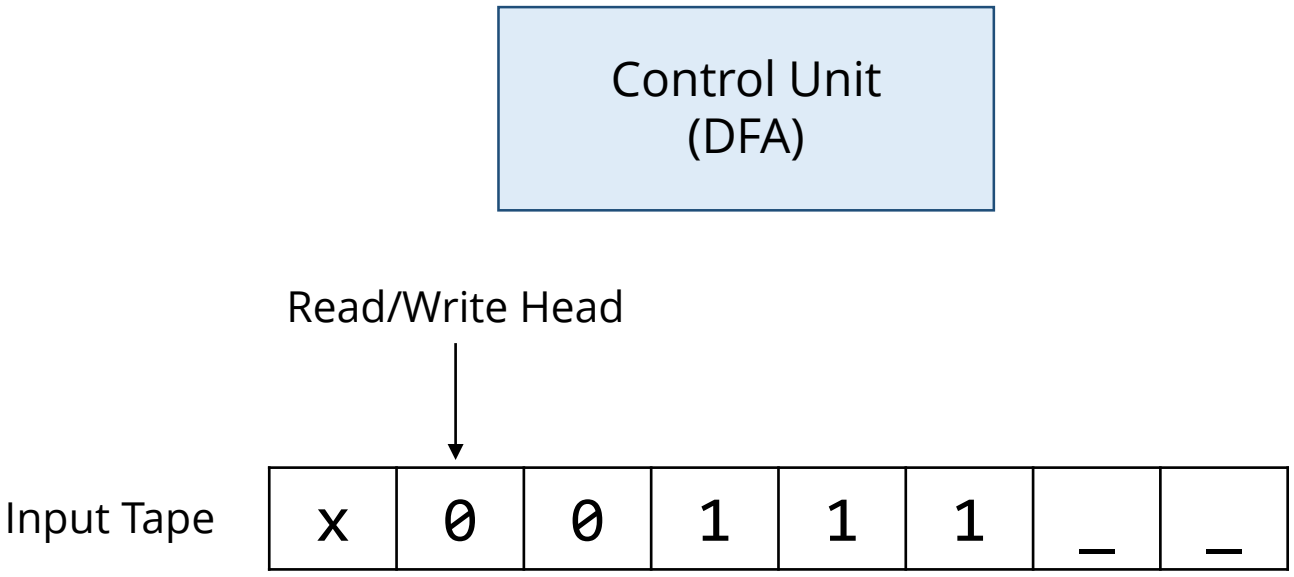
In a Turing machine, we have direct access to the input tape.



What if we write 0 → y, L?

Overview

In a Turing machine, we have direct access to the input tape.

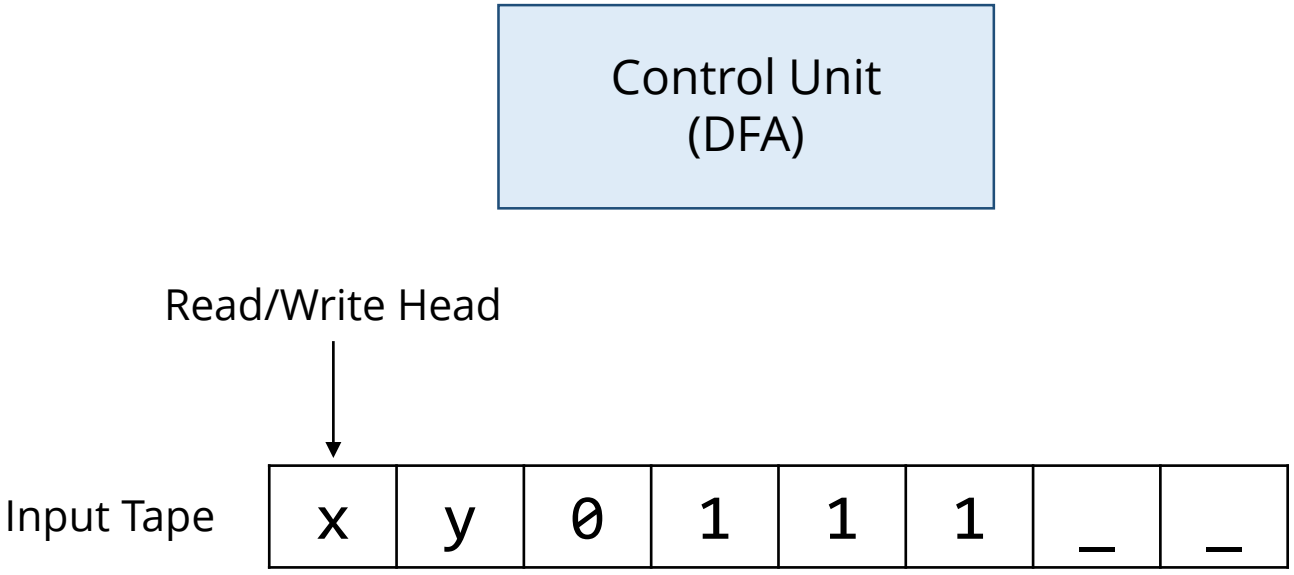


What if we write 0 → y, L?

This means if the head is now pointing at 0, we'll write y in its place, and move the head to the left

Overview

In a Turing machine, we have direct access to the input tape.

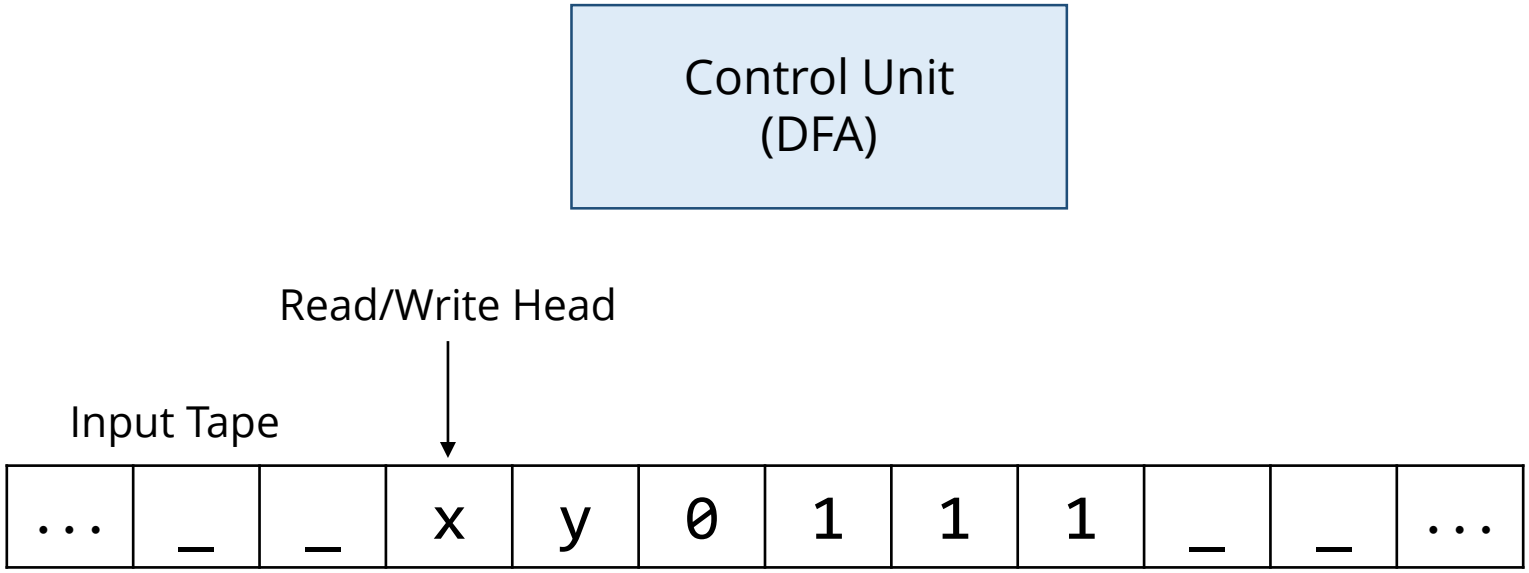


What if we write 0 → y, L?

This means if the head is now pointing at 0, we'll write y in its place, and move the head to the left

Overview

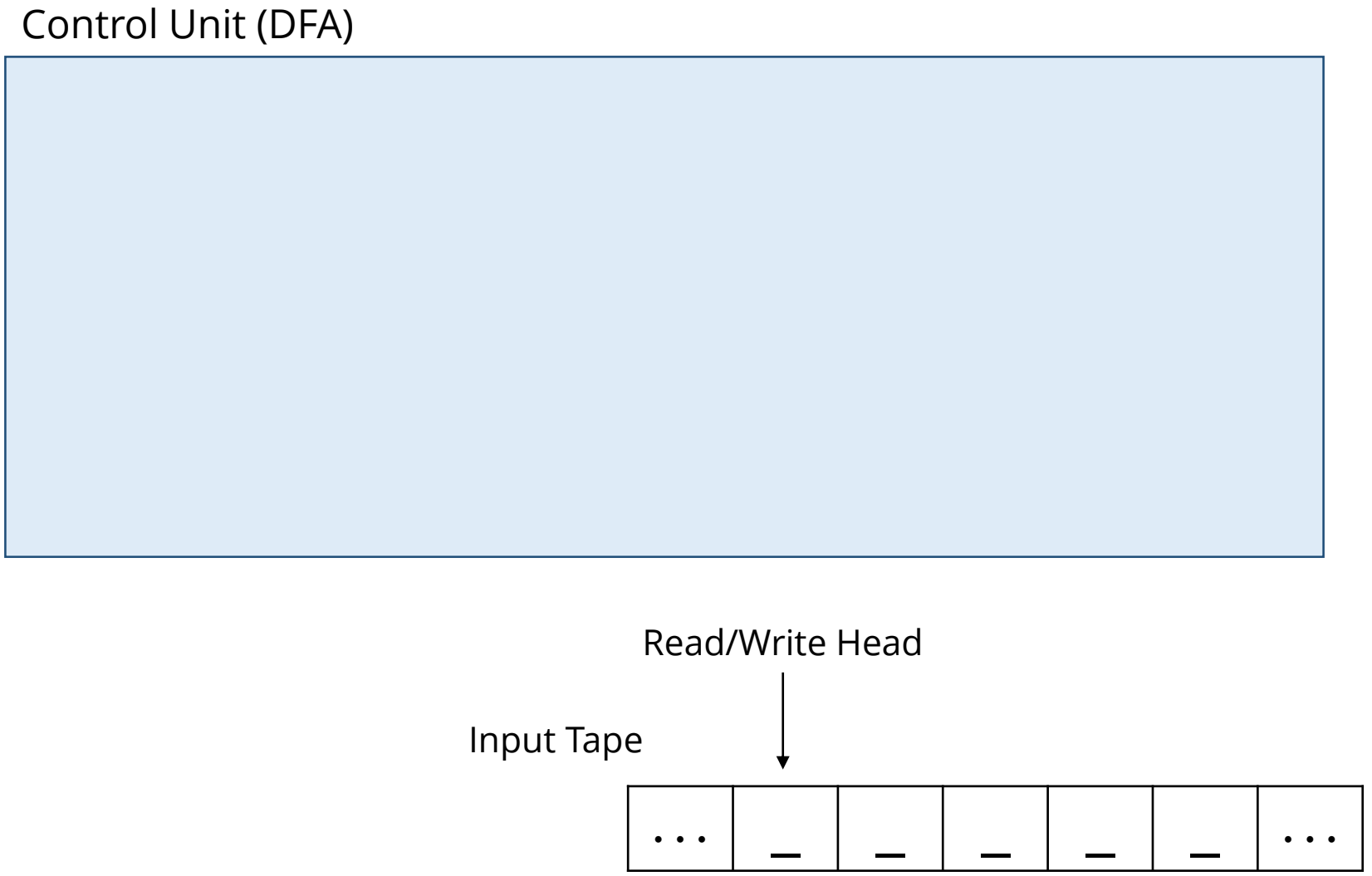
In a Turing machine, we have direct access to the input tape.



The input tape is infinite. The head points at the first symbol of the input. The left and the right of the input string is filled with infinite blank symbols _.

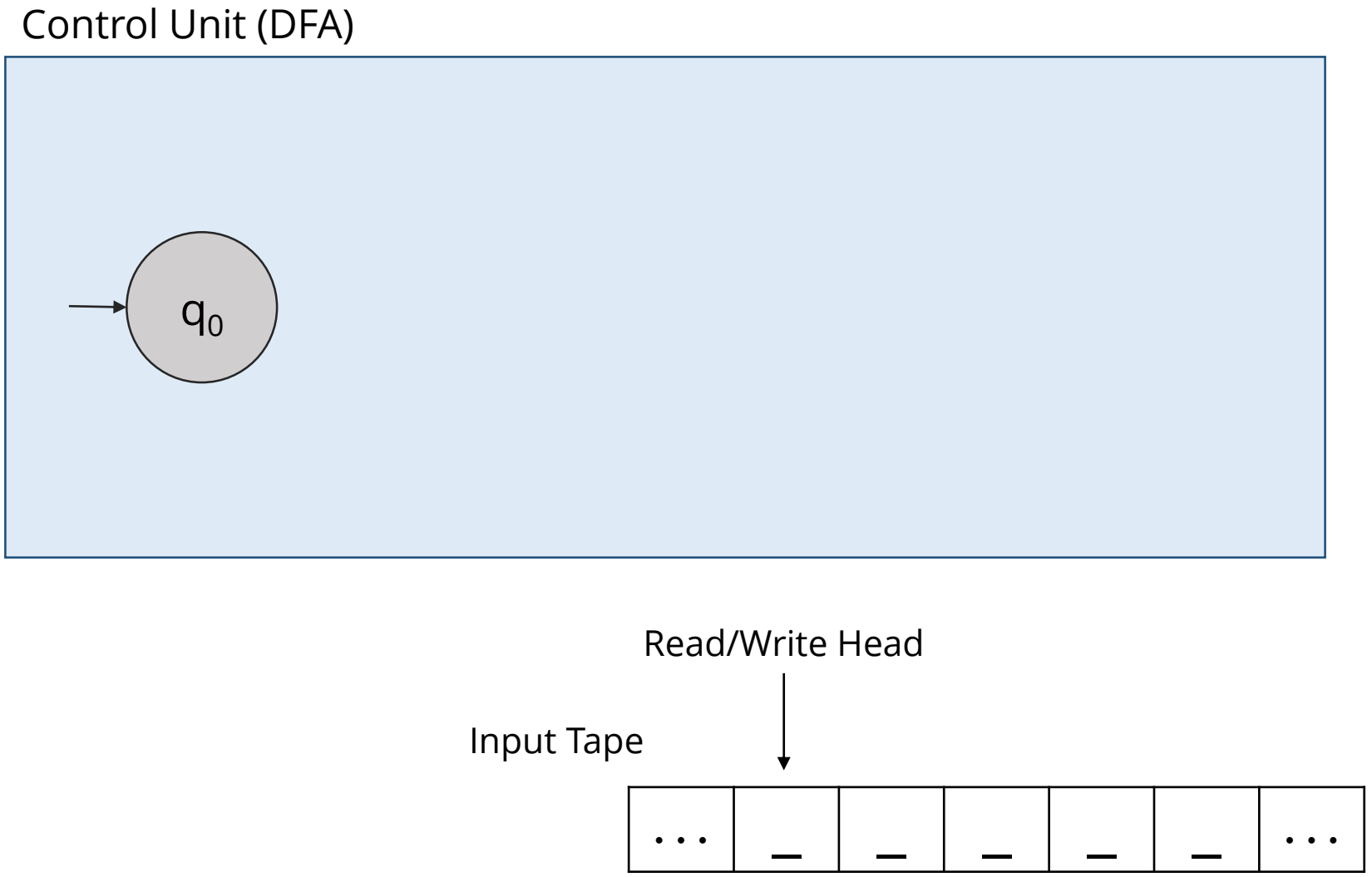
Example 1

A Turing machine that writes 110 in the tape



Example 1

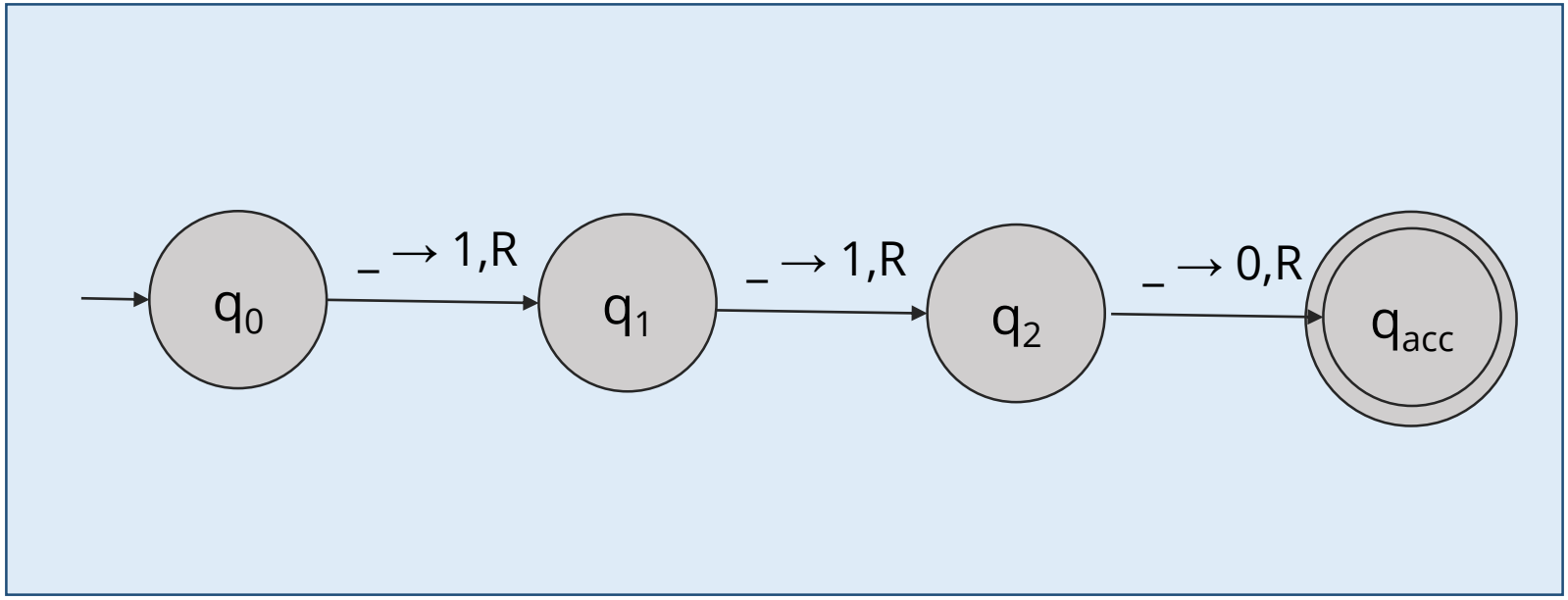
A Turing machine that writes 110 in the tape

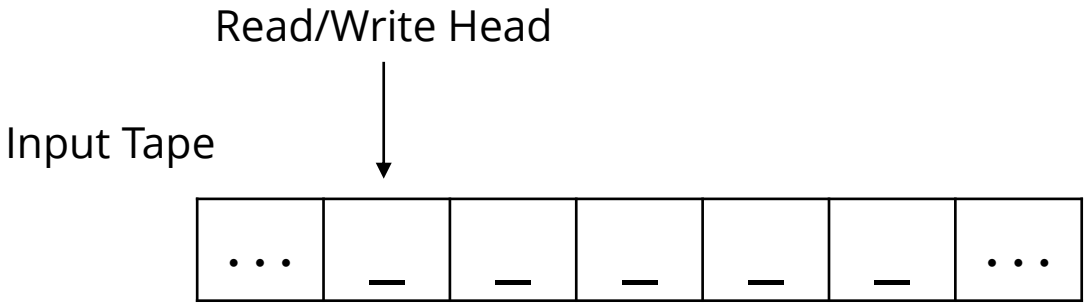


Example 1

A Turing machine that writes 110 in the tape

Control Unit (DFA)

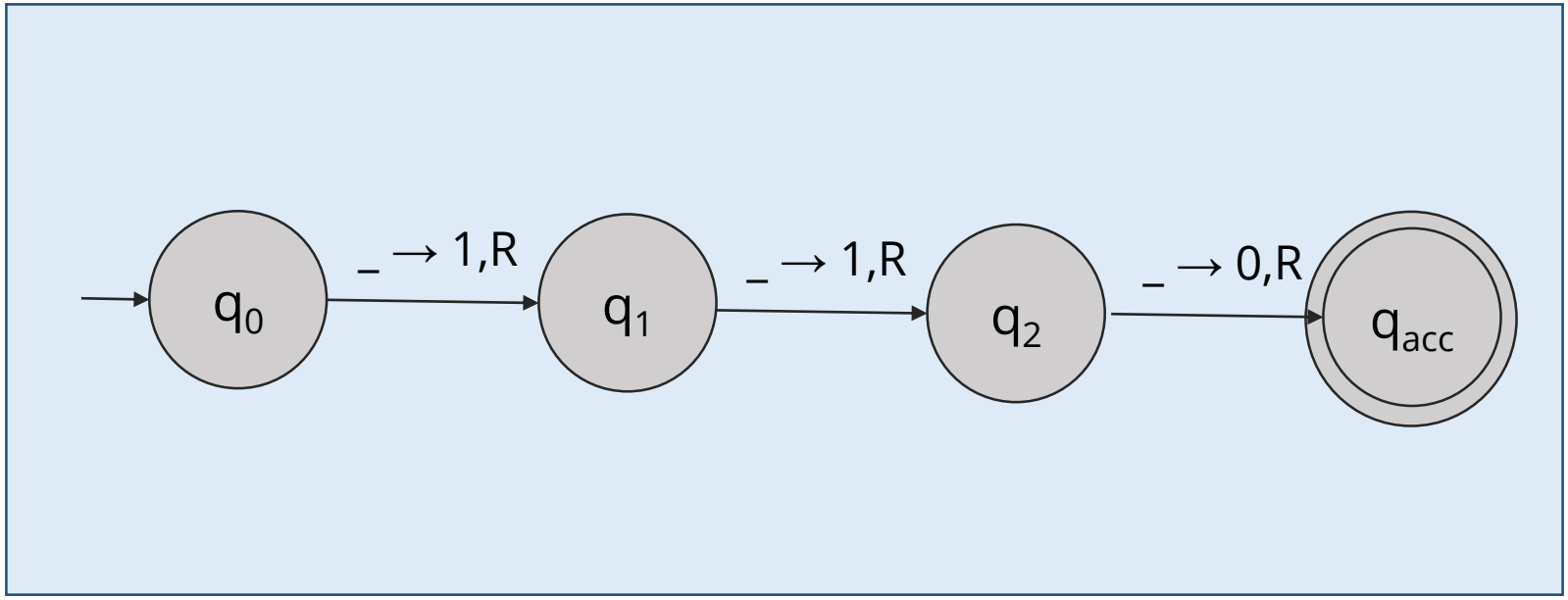


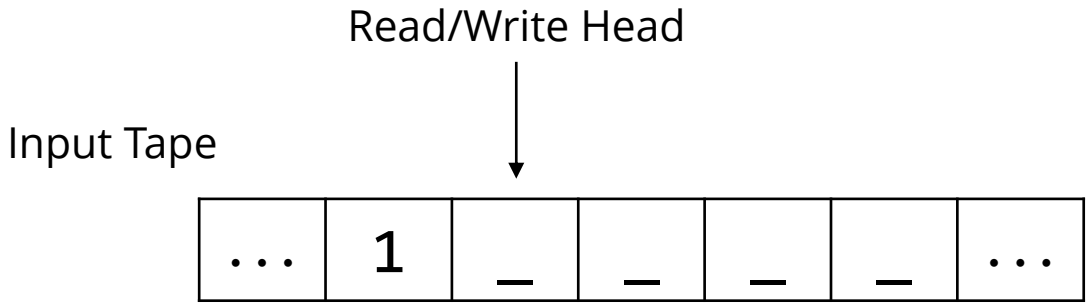


Example 1

A Turing machine that writes 110 in the tape

Control Unit (DFA)

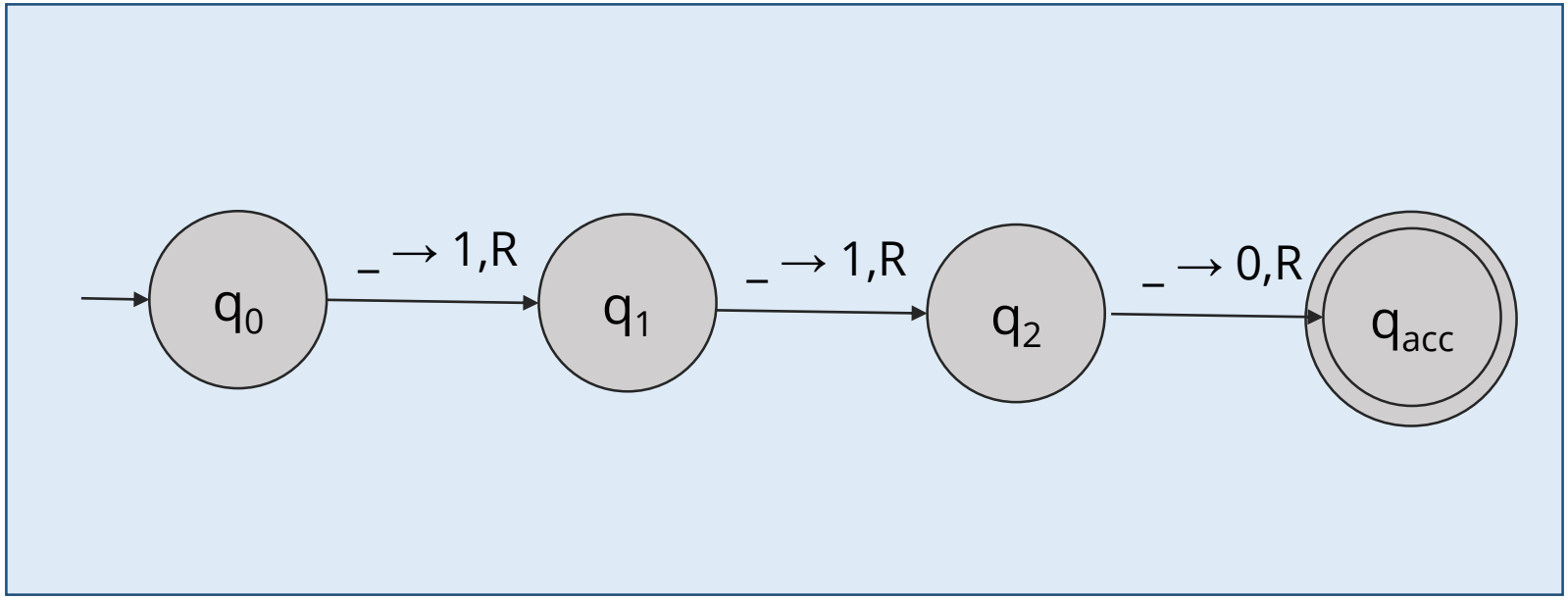


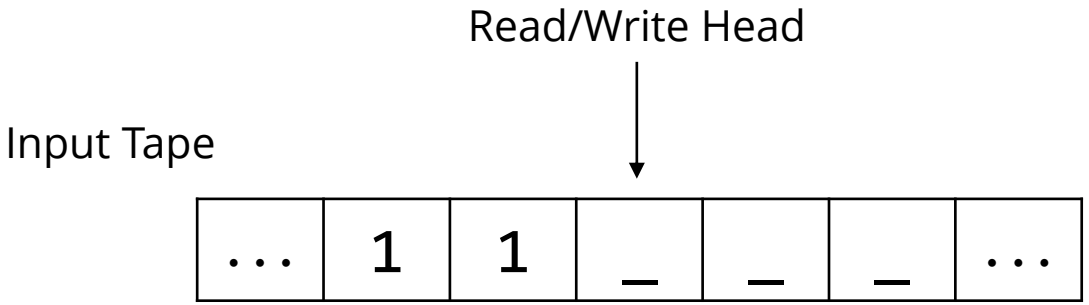


Example 1

A Turing machine that writes 110 in the tape

Control Unit (DFA)

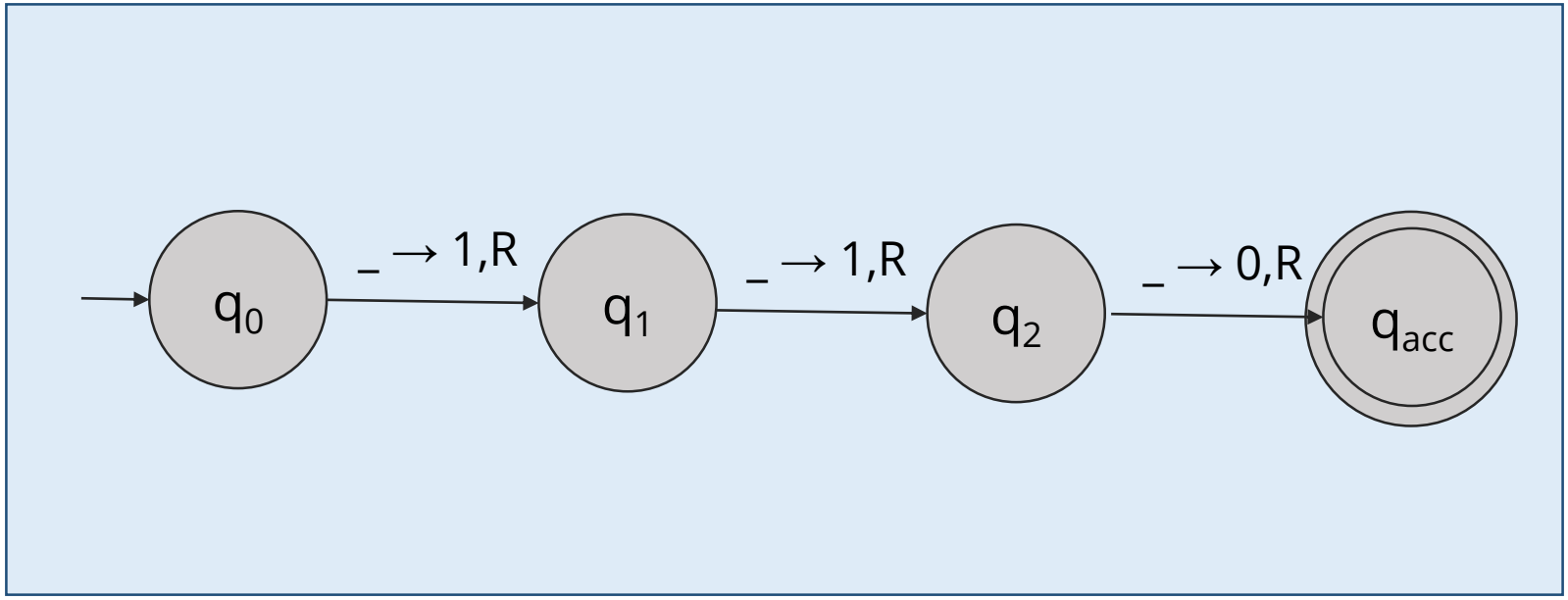


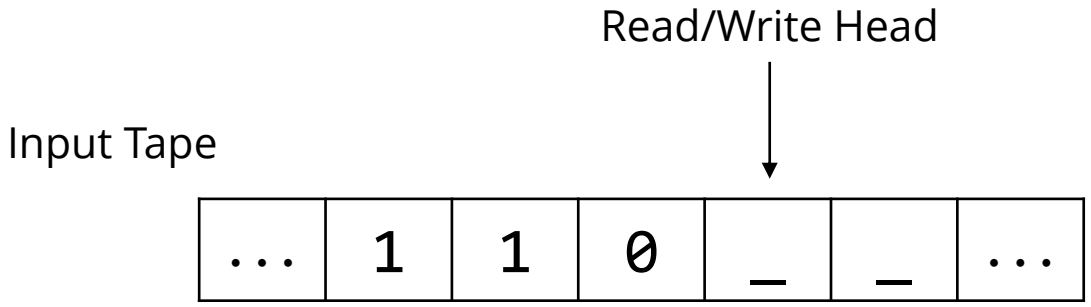


Example 1

A Turing machine that writes 110 in the tape

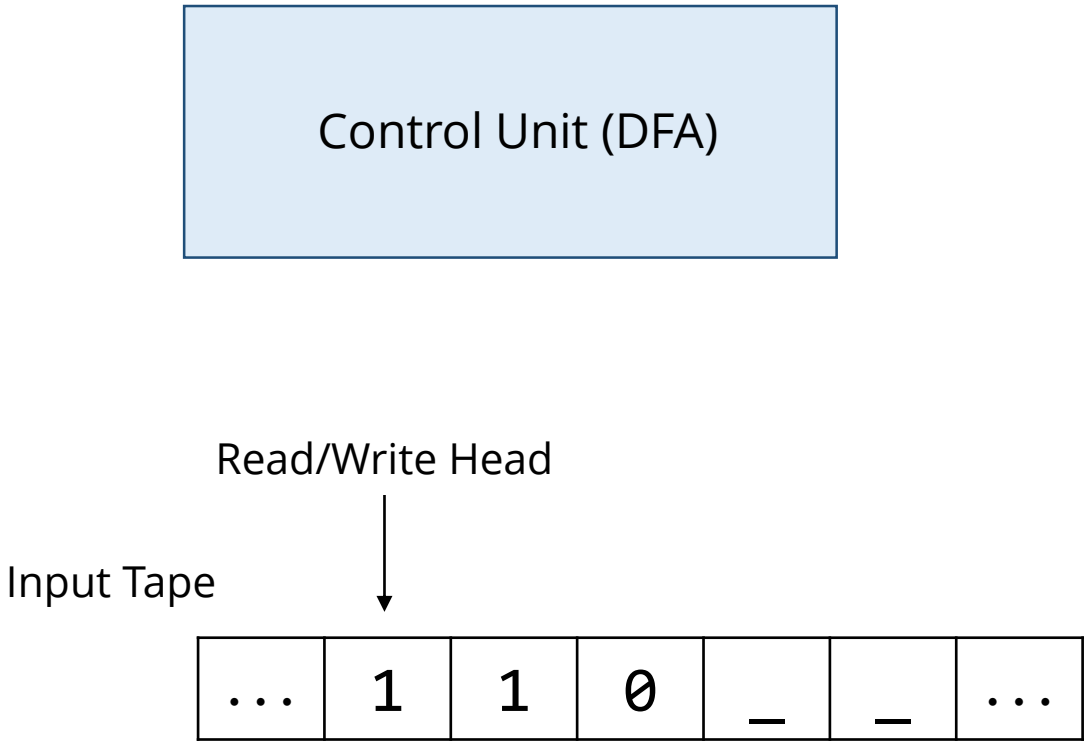
Control Unit (DFA)





Task

A Turing machine that toggles the input in the tape.



Instantaneous Description

1011$q_7$01111 means

- the tape is 101101111,
- the current state is q_7
- the head is currently on the second 0.

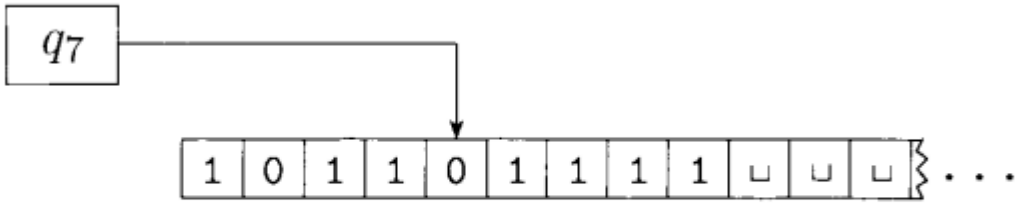


FIGURE 3.4
A Turing machine with configuration 1011$q_7$01111

Simulate Example 1

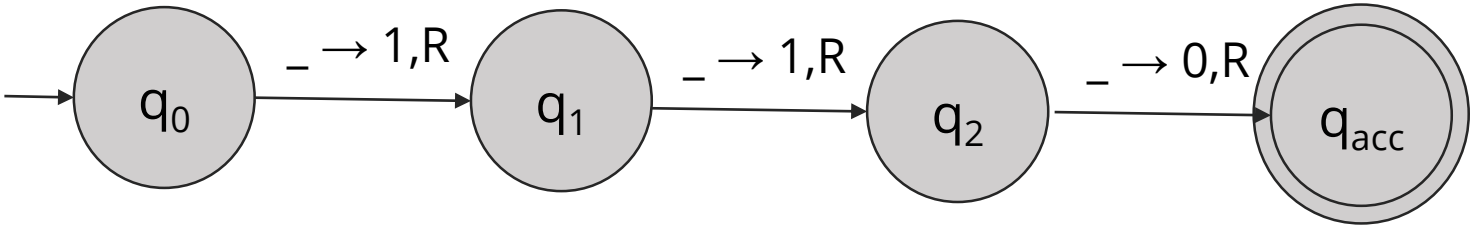
A Turing machine that writes 110.

q_0 _ _ _

1 q_1 _ _

11 q_2 _

110 q_{acc}



Simulate Task Solution on String 1011

A Turing machine that toggles the input in the tape.

$q_0$1011_

Simulate Task Solution on String 1011

A Turing machine that toggles the input in the tape.

$q_0 1011\_$

$0 q_0 011\_$

$01 q_0 11\_$

$010 q_0 1\_$

$0100 q_0\_$

$0100\_ q_{acc}$

Formal Description of a Turing Machine

A ***Turing machine*** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, where Q, Σ, Γ are all finite sets and

1. Q is the set of states,
2. Σ is the input alphabet not containing the ***blank symbol*** $\sqcup$,
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
4. $\delta: Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\text{L}, \text{R}\}$ is the transition function,
5. $q_0 \in Q$ is the start state,
6. $q_{\text{accept}} \in Q$ is the accept state, and
7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{reject}} \neq q_{\text{accept}}$.

What does this transition function mean?

$\delta(q_1, Z) = (q_2, Y, L)$

What does this transition function mean?

If $\delta(q_1, Z) = (q_2, Y, L)$ then,

in state q_1, scanning Z under its tape head, the TM:

1. Changes the state to q_2.
2. Replaces Z by Y on the tape.
3. Moves the head one square in direction Left.

Note: L = move left; R = move right.

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {}

Tape Symbols = {}

States = {}

Transition Function: ?

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = {}

Transition Function: ?

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = $\{q_0, q_{acc}\}$

Transition Function: ?

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = $\{q_0, q_{acc}\}$

Transition Function:

- $\delta(q_0, 0) =$
- $\delta(q_0, 1) =$
- $\delta(q_0, \_) =$

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = $\{q_0, q_{acc}\}$

Transition Function:

- $\delta(q_0, 0) = (q_0, 0, R)$
- $\delta(q_0, 1) =$
- $\delta(q_0, \_) =$

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = $\{q_0, q_{acc}\}$

Transition Function:

- $\delta(q_0, 0) = (q_0, 0, R)$
- $\delta(q_0, 1) = (q_{acc}, 0, R)$
- $\delta(q_0, \_) =$

Example 2

Write the formal definition of a Turing machine, that

- When encounters a 1, changes it to 0 and reaches final state.
- If reaches a blank, changes it to 1 and move left.

Input Symbols = {0, 1}

Tape Symbols = {0, 1, _}

States = $\{q_0, q_{acc}\}$

Transition Function:

- $\delta(q_0, 0) = (q_0, 0, R)$
- $\delta(q_0, 1) = (q_{acc}, 0, R)$
- $\delta(q_0, \_) = (q_0, 1, L)$

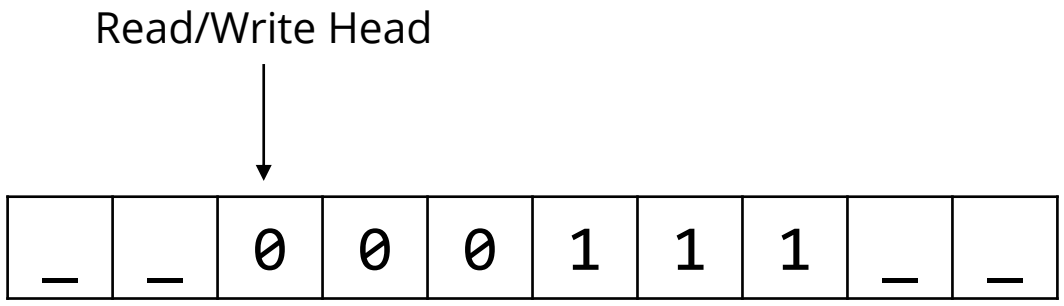
Task

Design a Turing Machine that recognizes the string 01*0

Simulate the string 01110 for the machine

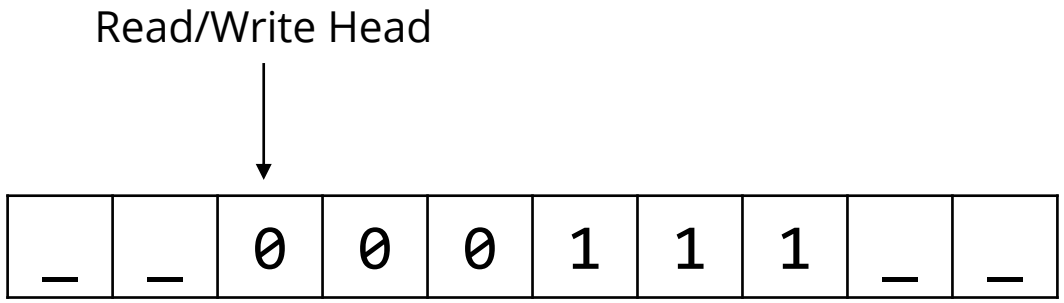
Example 3

Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



Example 3

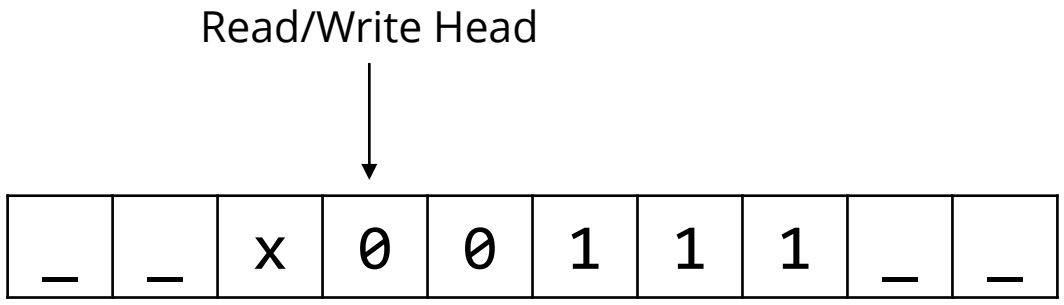
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

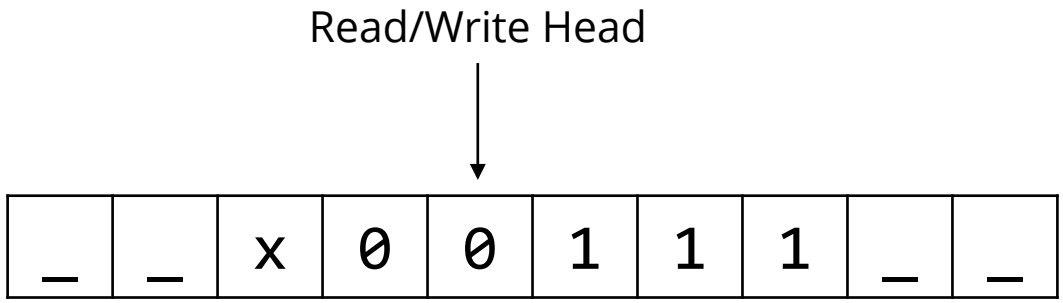
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

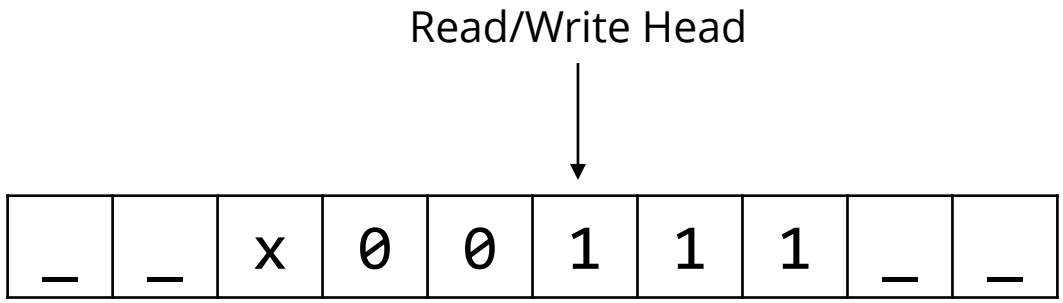
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

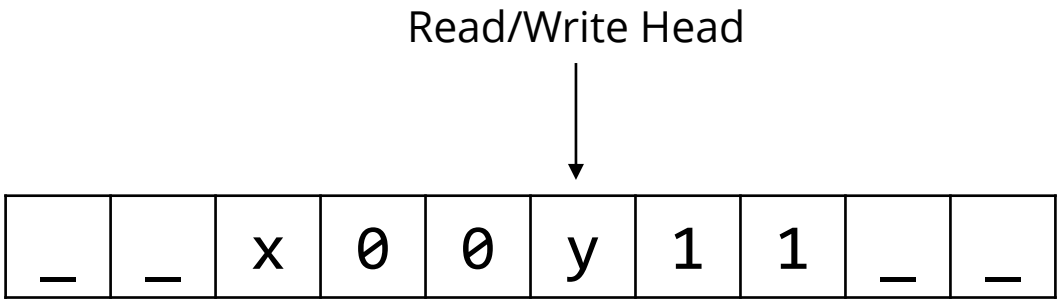
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

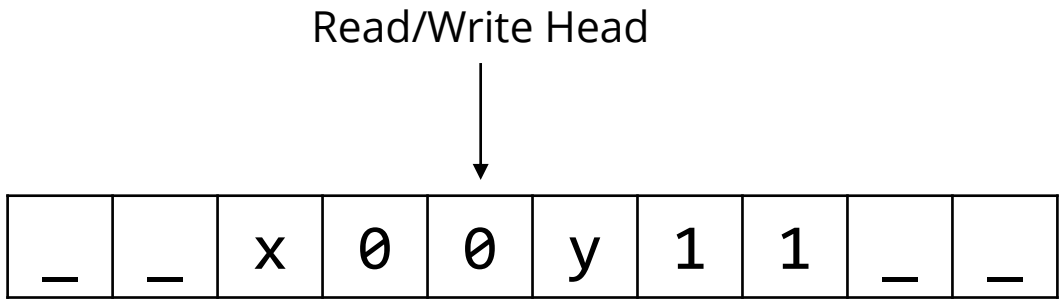
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

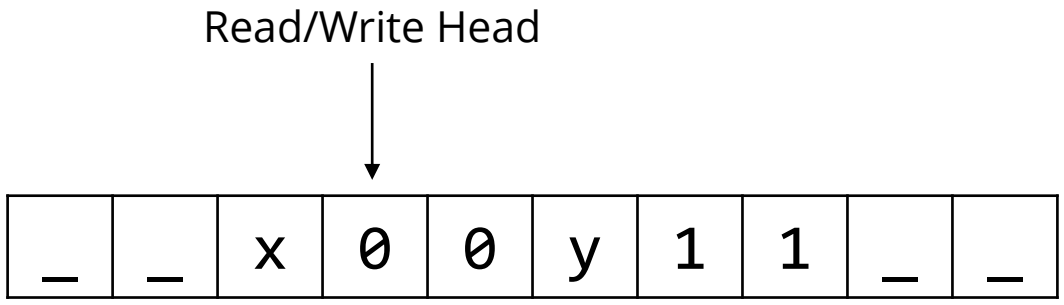
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

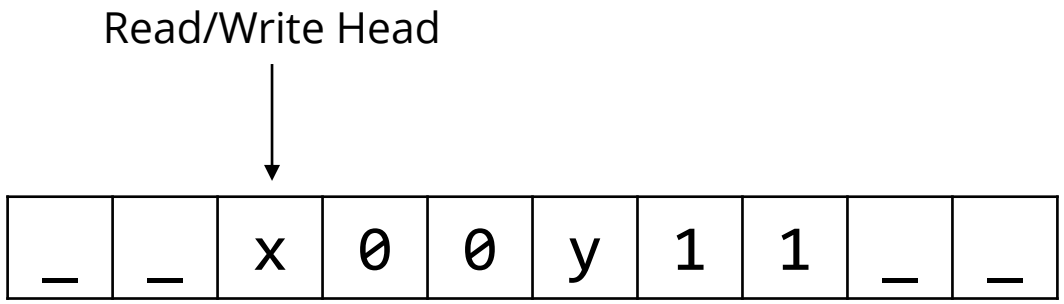
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

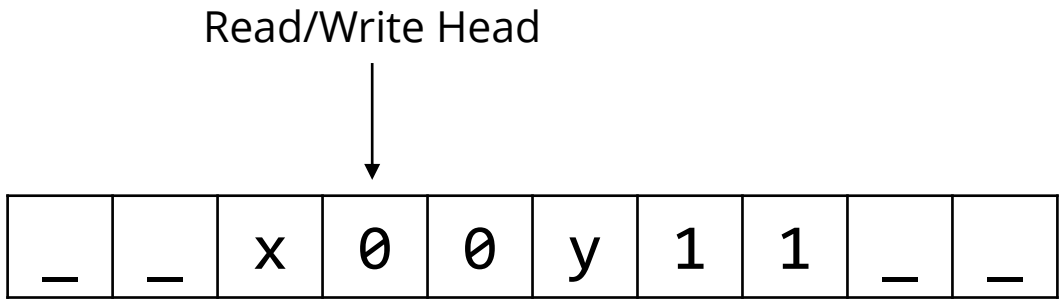
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

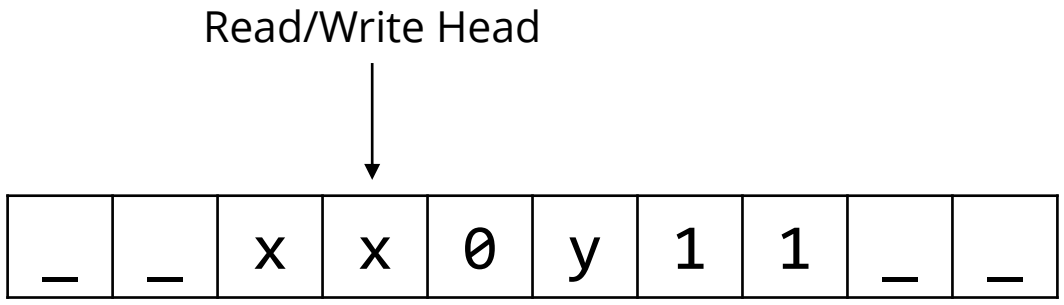
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

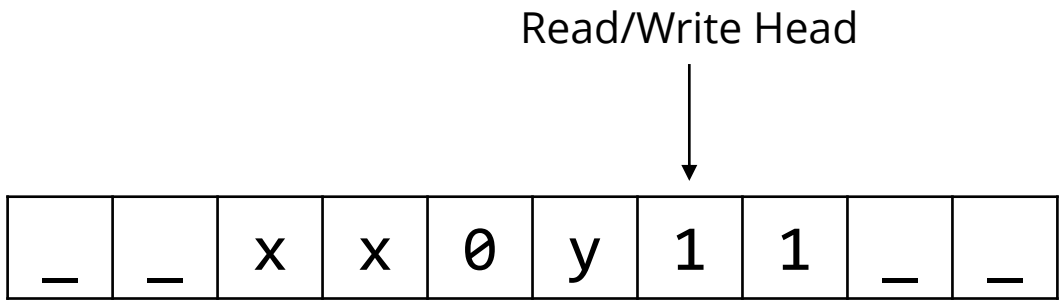
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

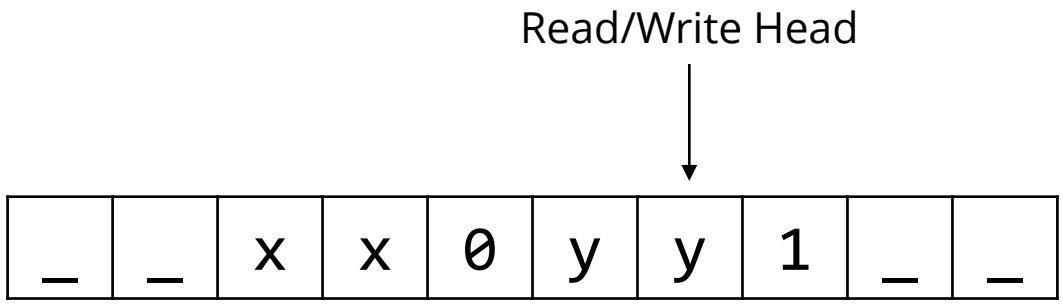
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

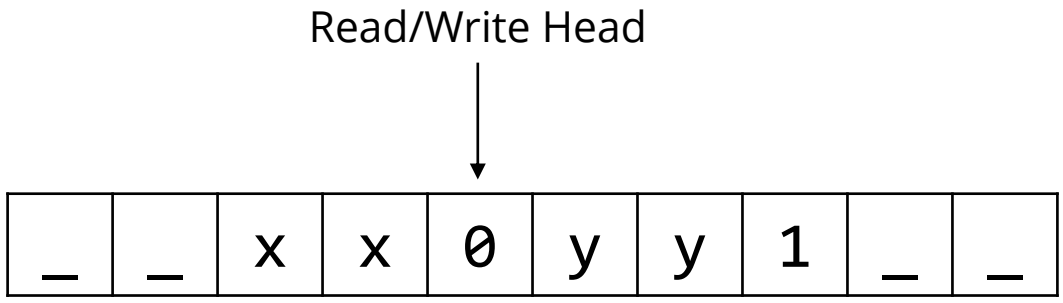
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

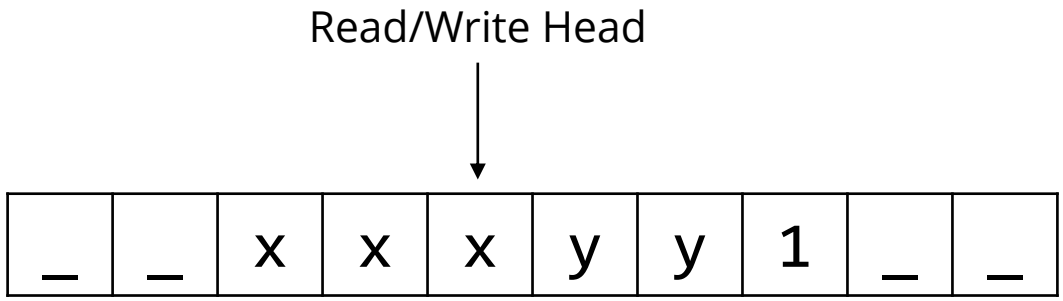
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

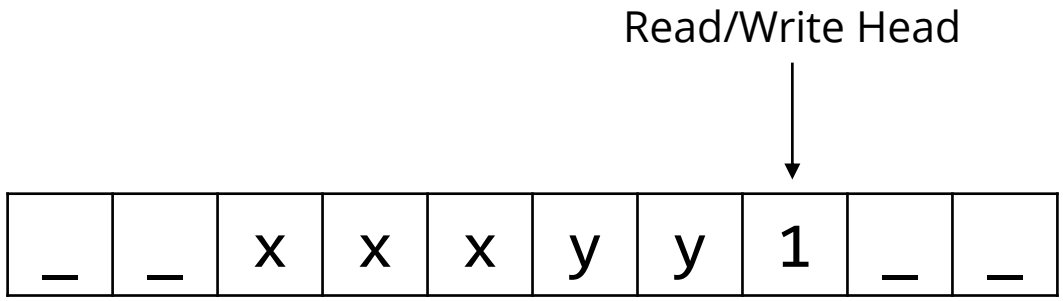
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

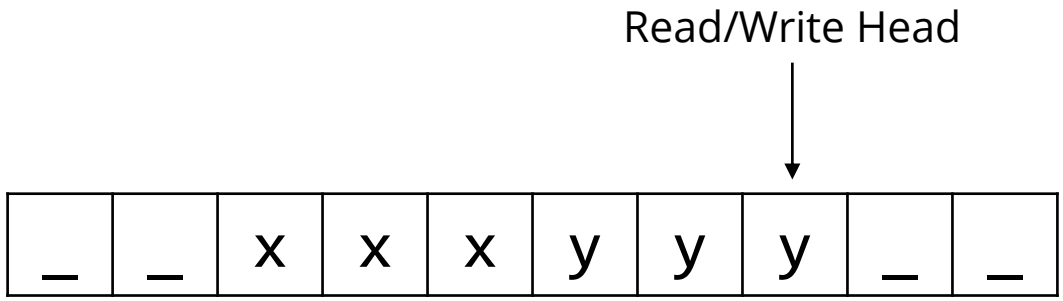
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

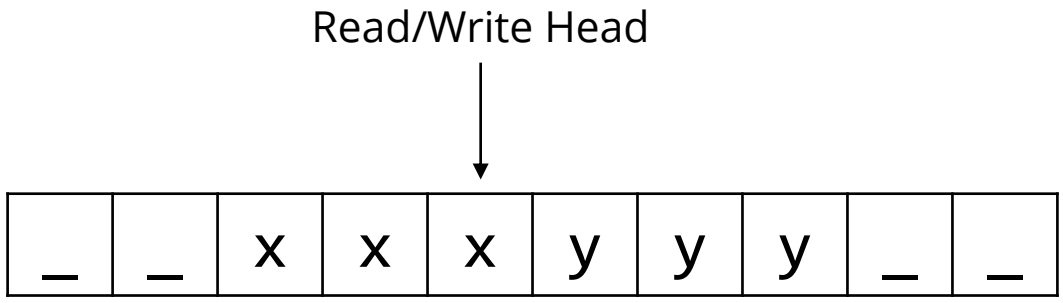
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

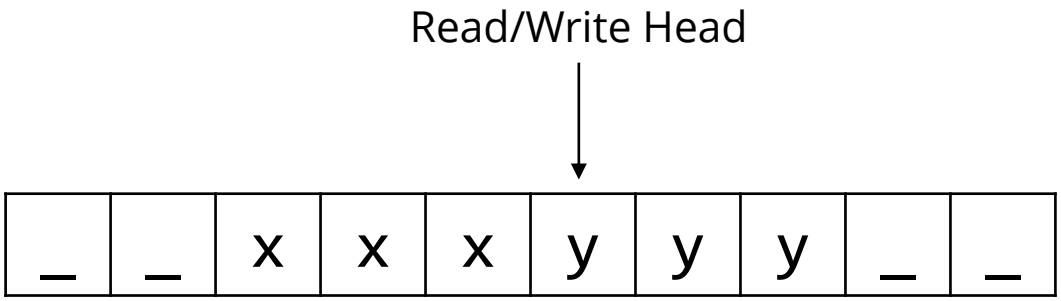
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

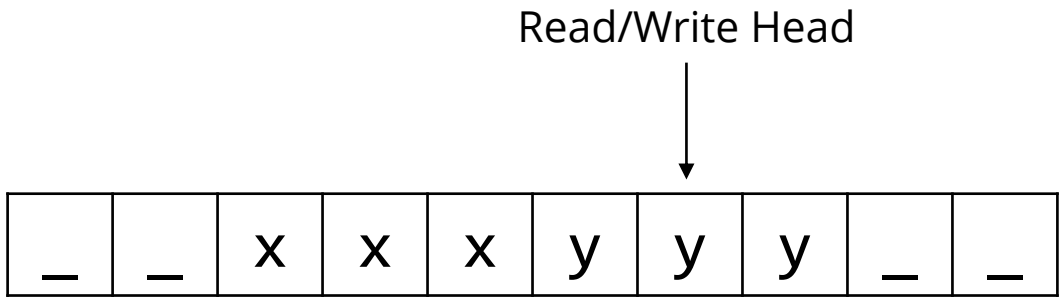
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

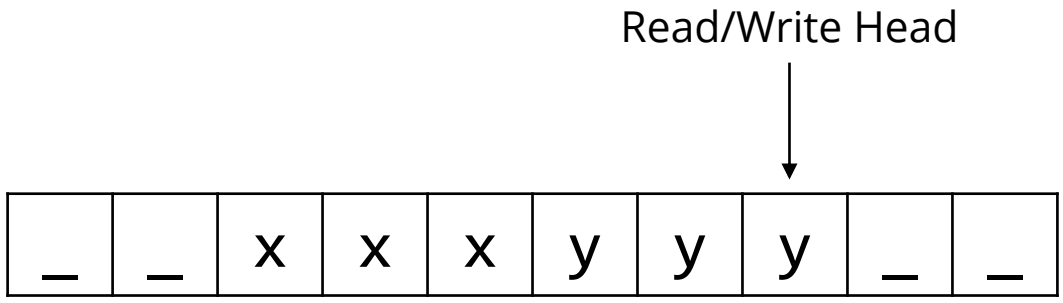
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

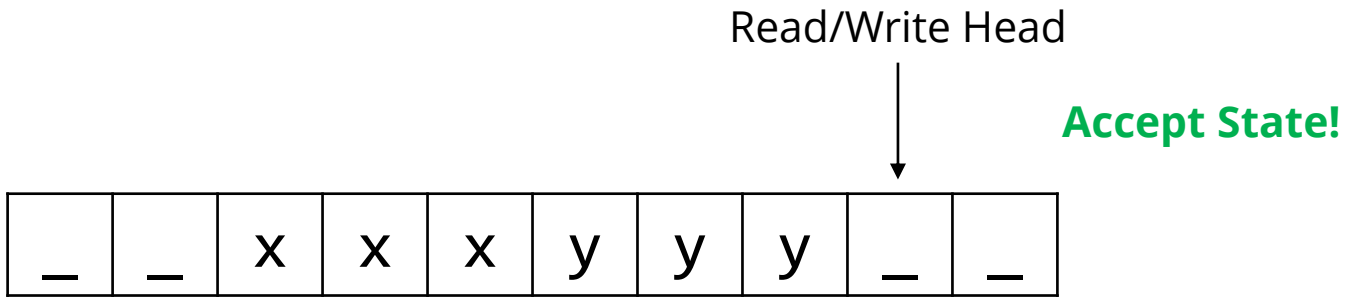
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Example 3

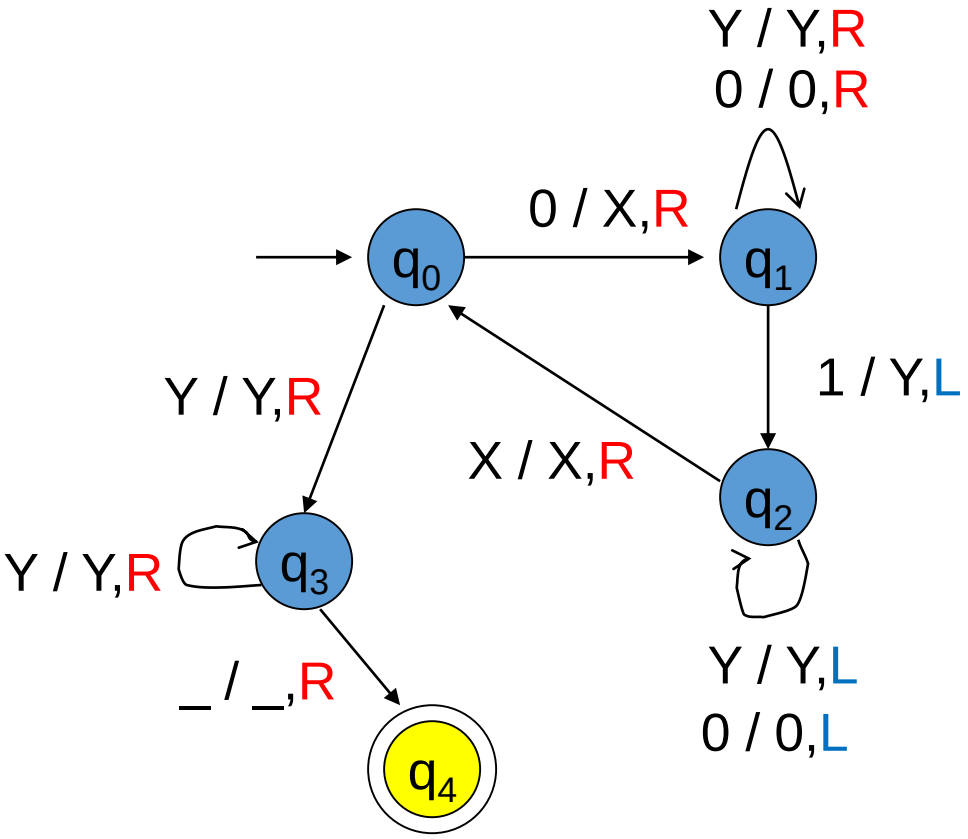
Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

State Diagram of Control

Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$



1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.

Transition Function

Design a Turing Machine that recognizes the Language L = $\{0^n1^n \mid n \geq 1\}$

	Next Tape Symbol				
Curr. State	0	1	X	Y	B
→ q_0	(q_1,X,R)	-	-	(q_3,Y,R)	-
q_1	(q_1,0,R)	(q_2,Y,L)	-	(q_1,Y,R)	-
q_2	(q_2,0,L)	-	(q_0,X,R)	(q_2,Y,L)	-
q_3	-	-	-	(q_3,Y,R)	(q_4,_,R)
*q_4	-	--	-	-	-

1. Mark next unread 0 with X and move right
2. Move to the right all the way to the first unread 1, and mark it with Y
3. Move back (to the left) all the way to the last marked X, and then move one position to the right
4. If the next position is 0, then goto step 1.
   Else move all the way to the right to ensure there are no excess 1s. If not move right to the next blank symbol and stop & accept.