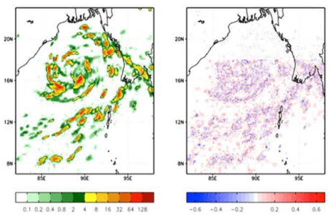
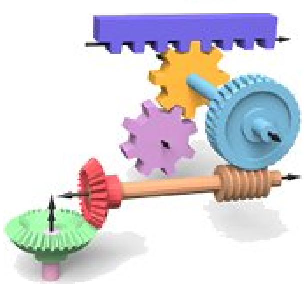
# Finite Automata

CS 2800: Discrete Structures, Spring 2015

Sid Chaudhuri

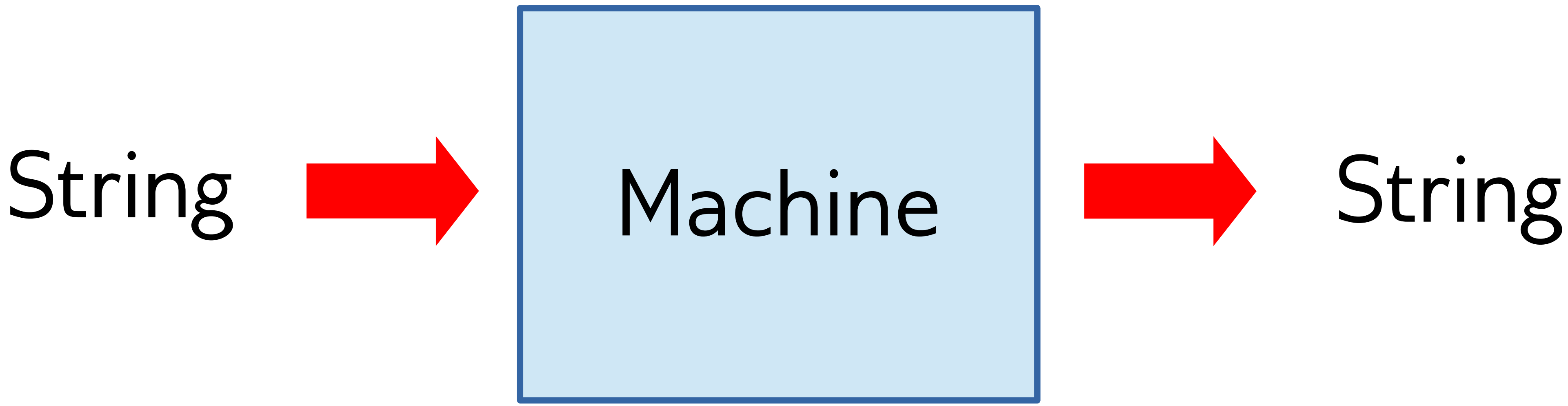

<!DOCTYPE html>
<html id="home-layout">
<head>
<meta http-equiv="content-type" conte
<title>Source Code Pro</title>
<!-- made with <3 and AFDKO -->
<meta name="keywords" content="sans,
monospace, open source, coding, for
<link rel="stylesheet" type="text/css
</head>
<body>
<div id="main">



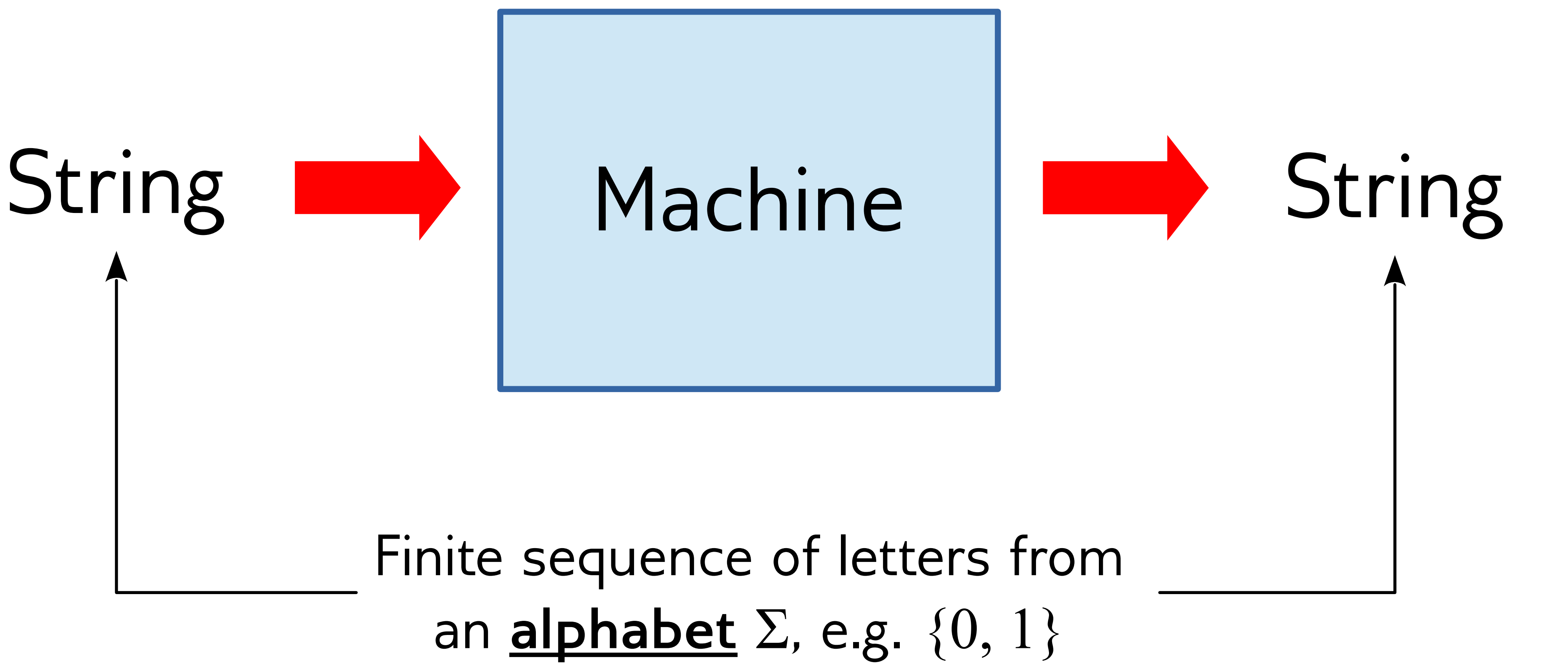



<!DOCTYPE html>
<html id="home-layout">
<head>
<meta http-equiv="content-type" conte
<title>Source Code Pro</title>
<!-- made with <3 and AFDKO -->
<meta name="keywords" content="sans,
monospace, open source, coding, for
<link rel="stylesheet" type="text/css
</head>
<body>
<div id="main">
LIFE CYCLE OF A
FLOWERING PLANT
REPORT
Seed Germination

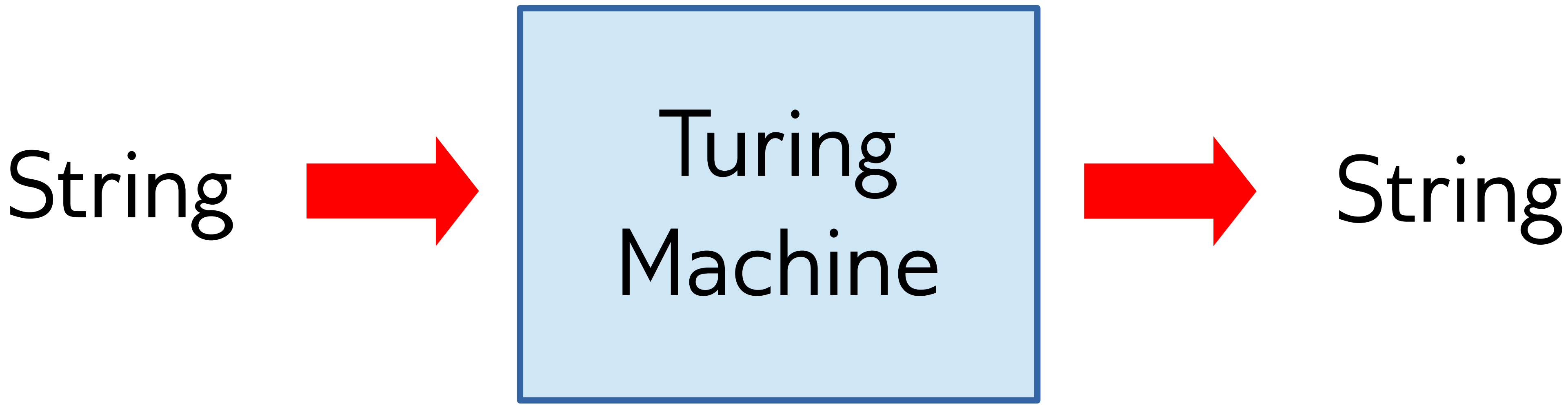
# A simplified model



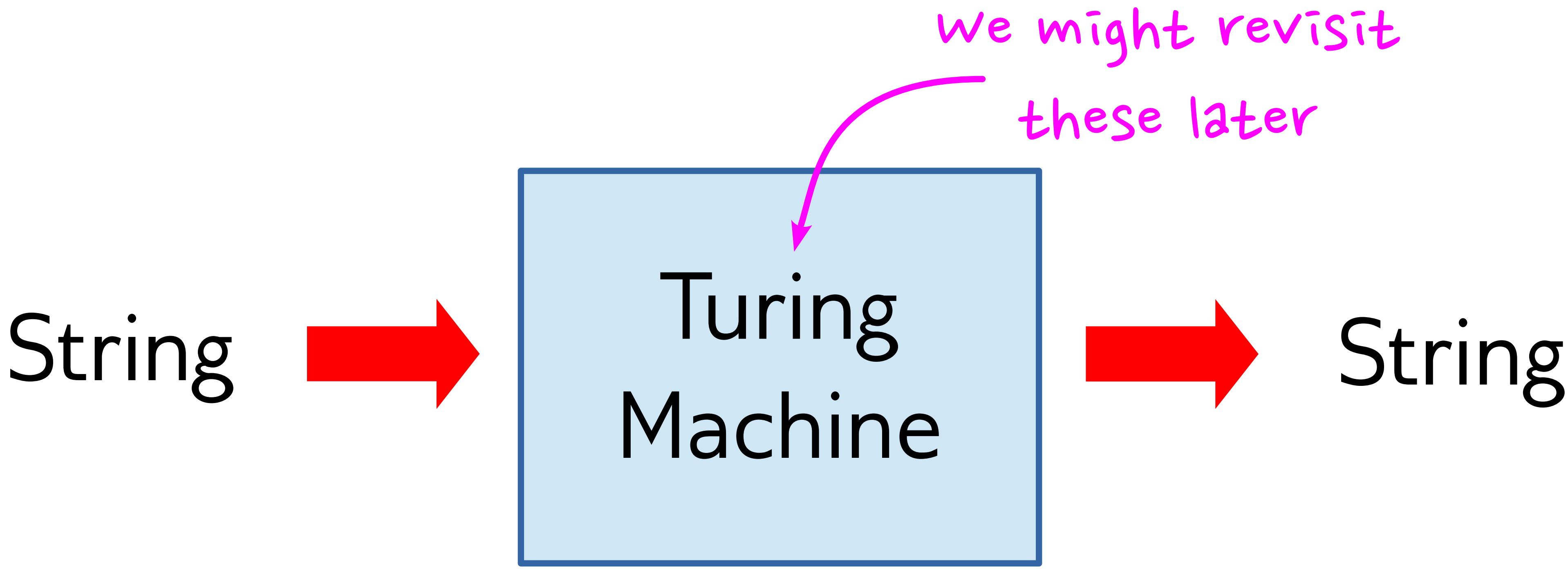
# A simplified model



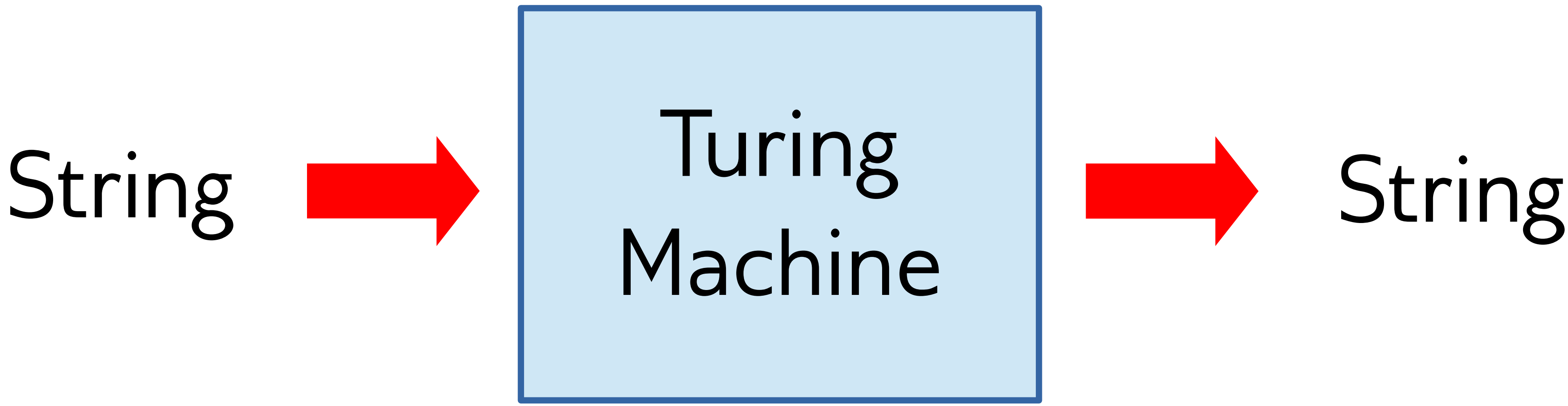
# A general-purpose computer



# A general-purpose computer



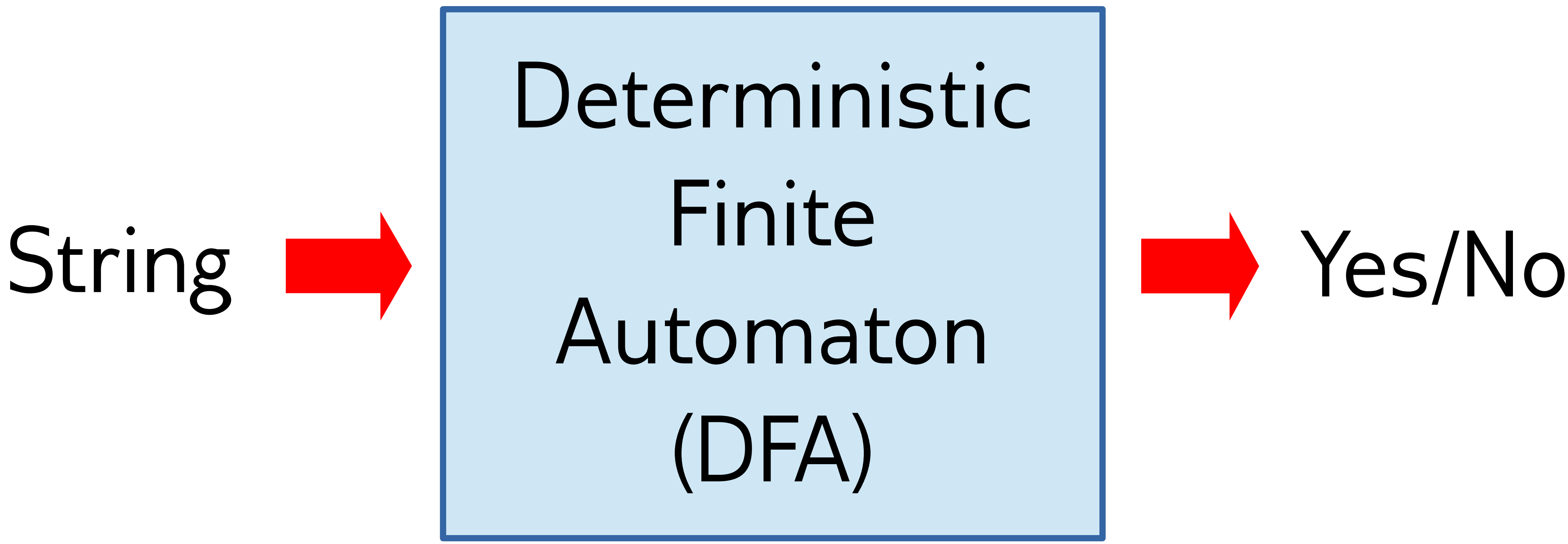
# A general-purpose computer



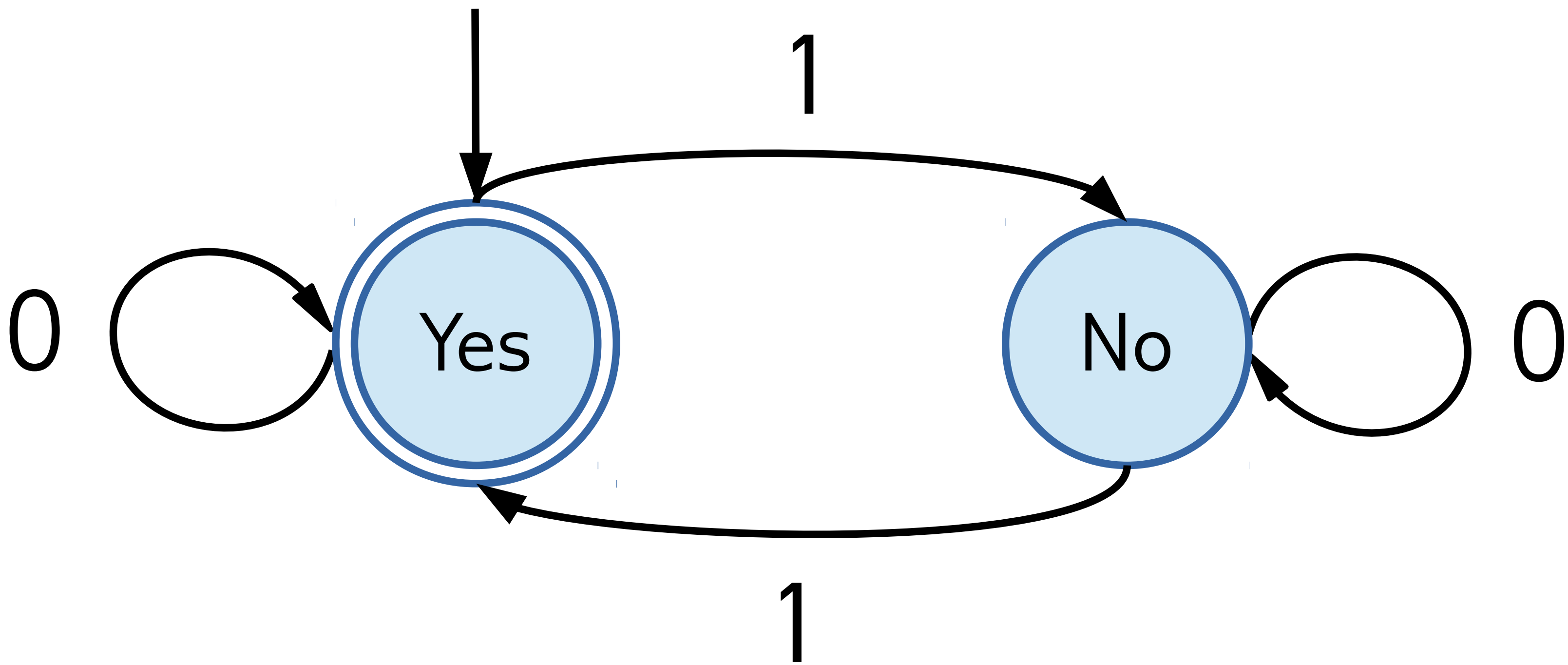

Church-Turing Thesis: Any “effective/mechanical/real-world” calculation can be carried out on a Turing machine

Alan Turing, 1912 – 1954

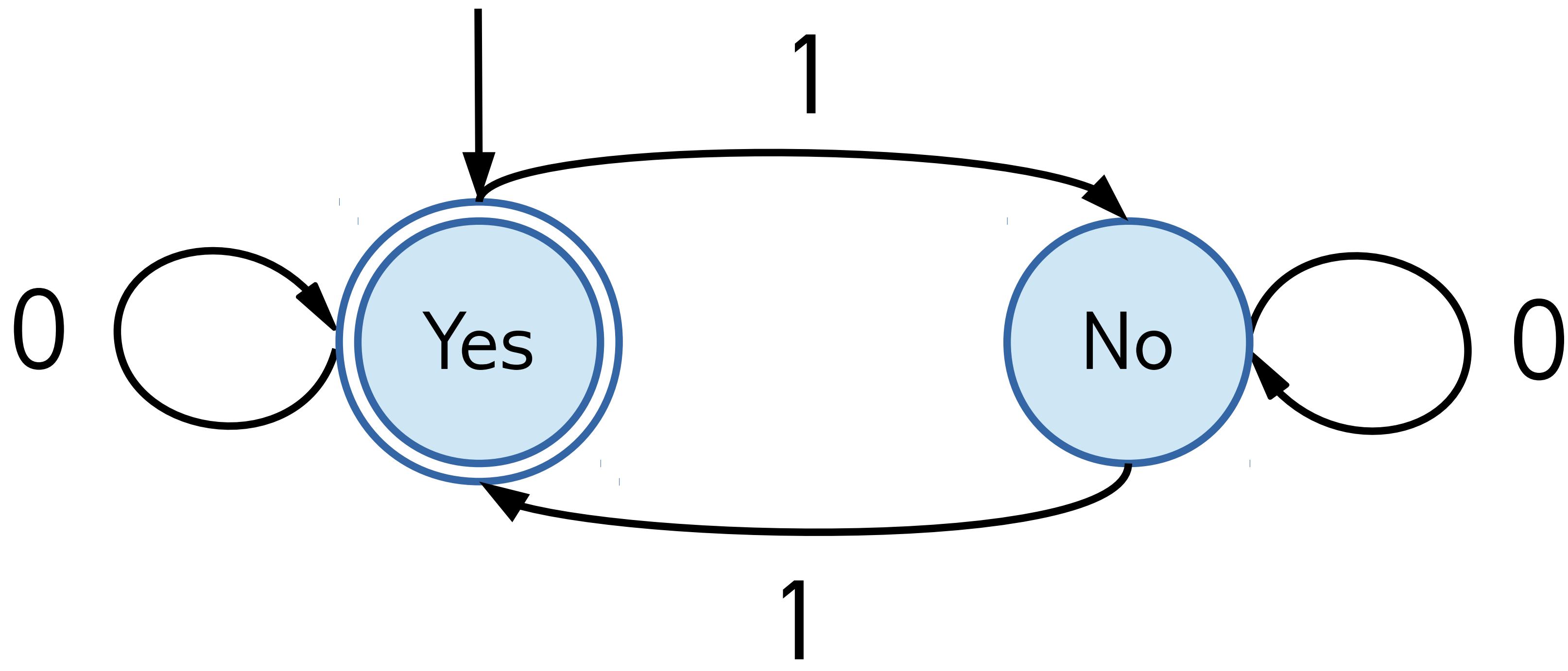
# A simple “computer”



# An example

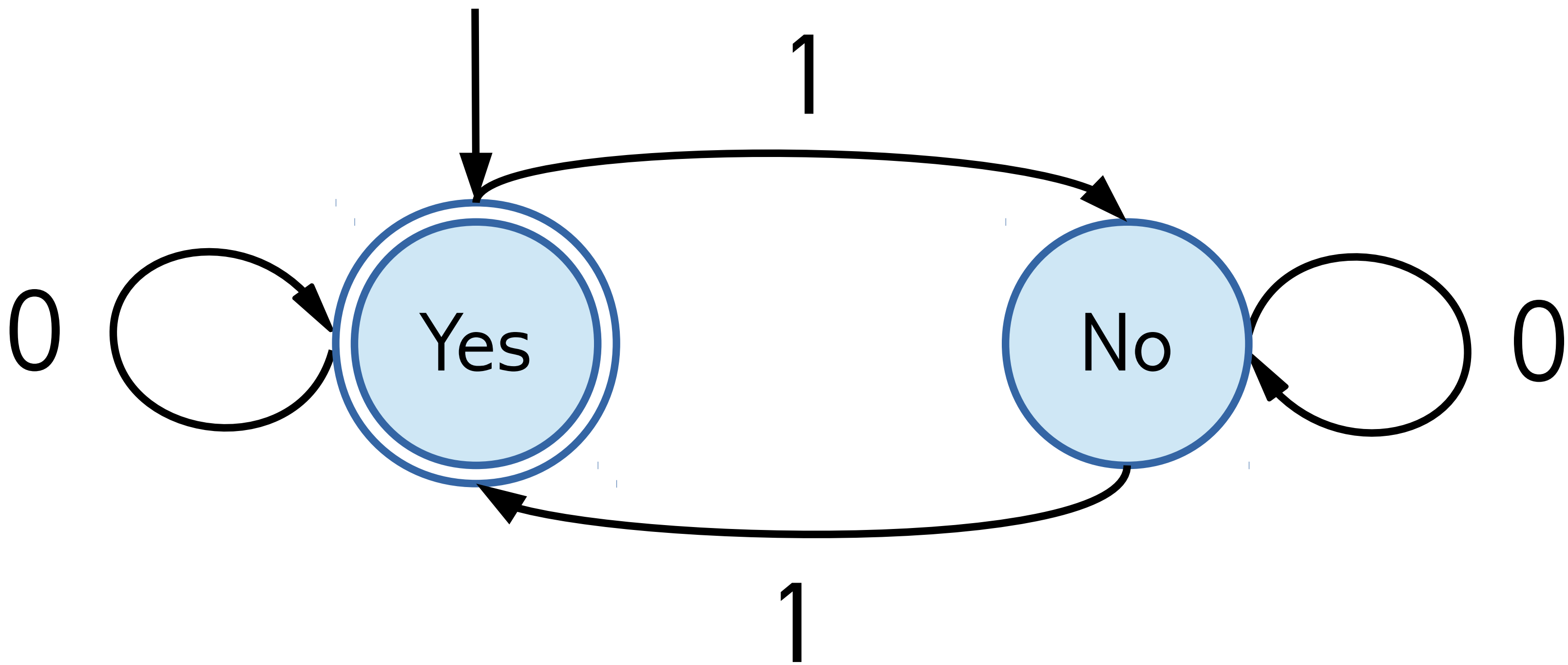


# An example



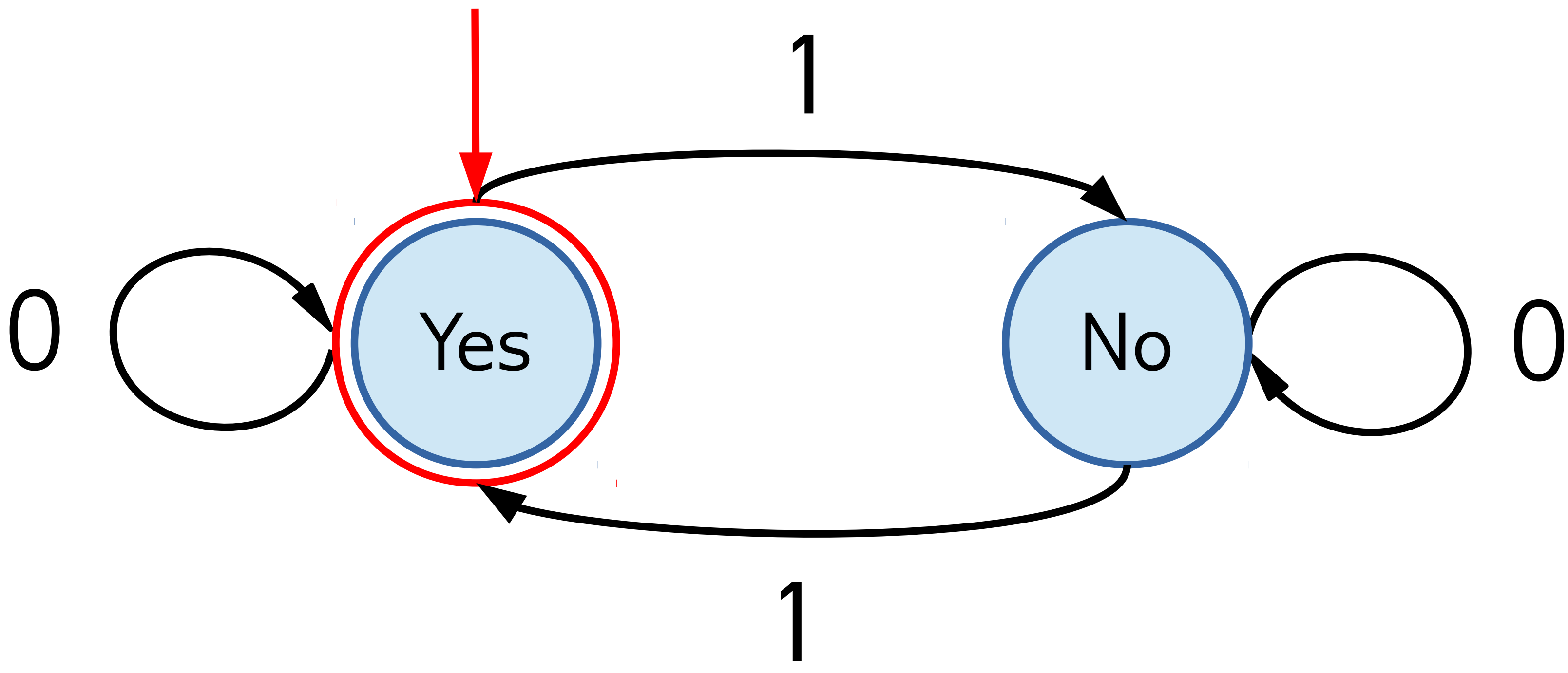

(Binary input)

# An example



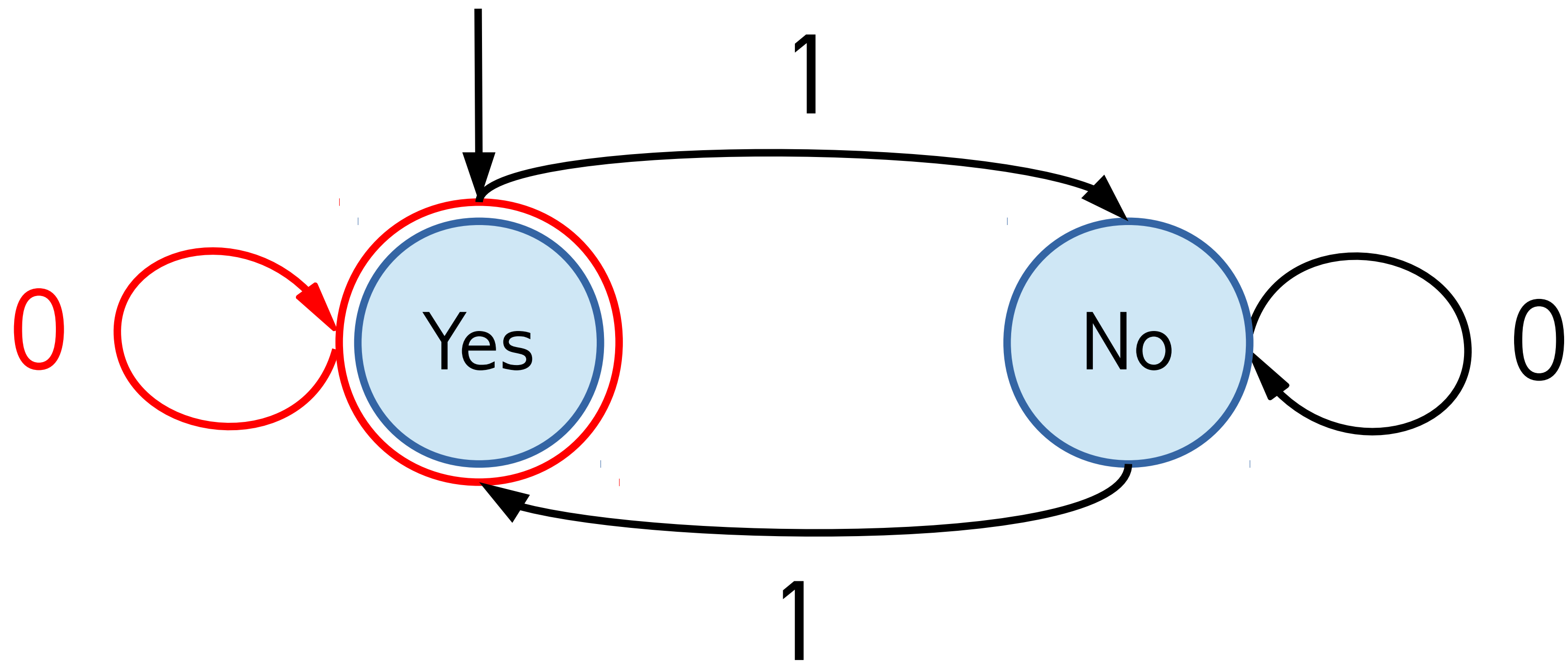

Input: 01001

# An example



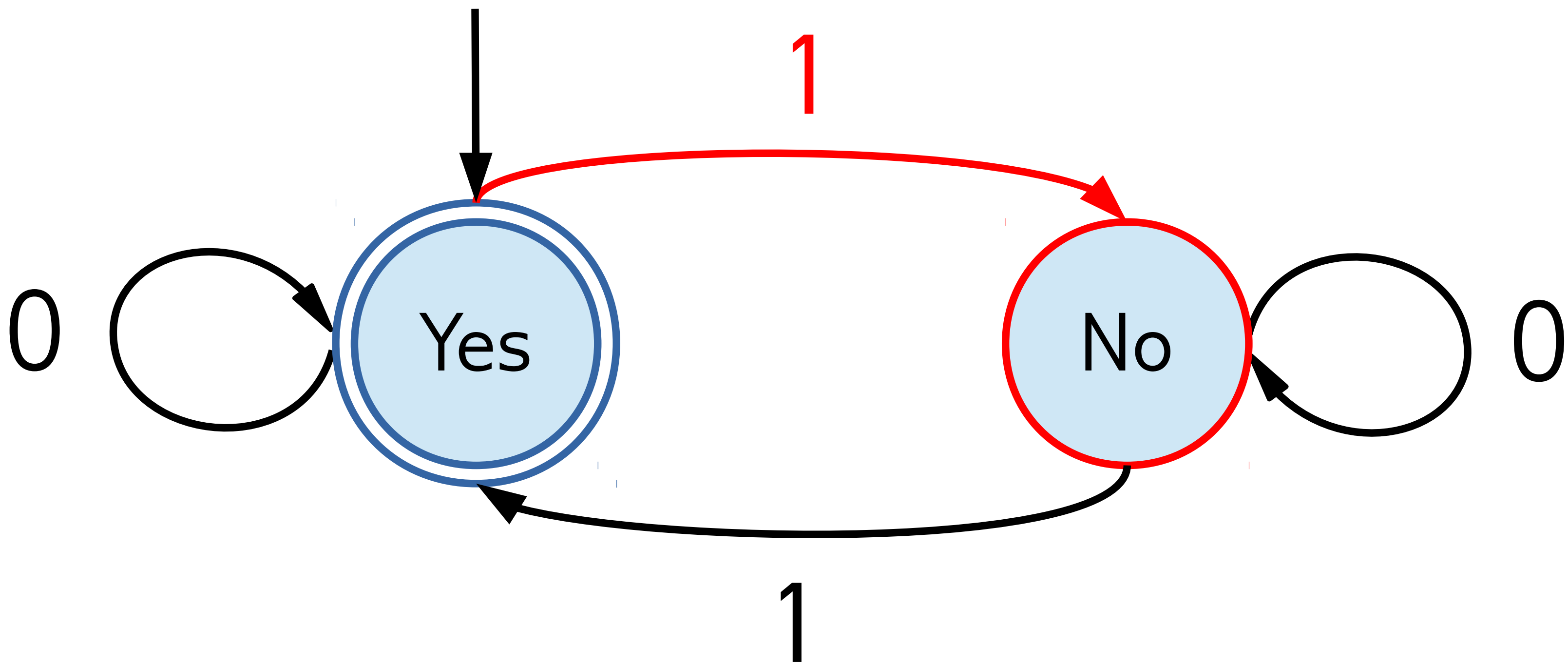

Input: 01001

# An example



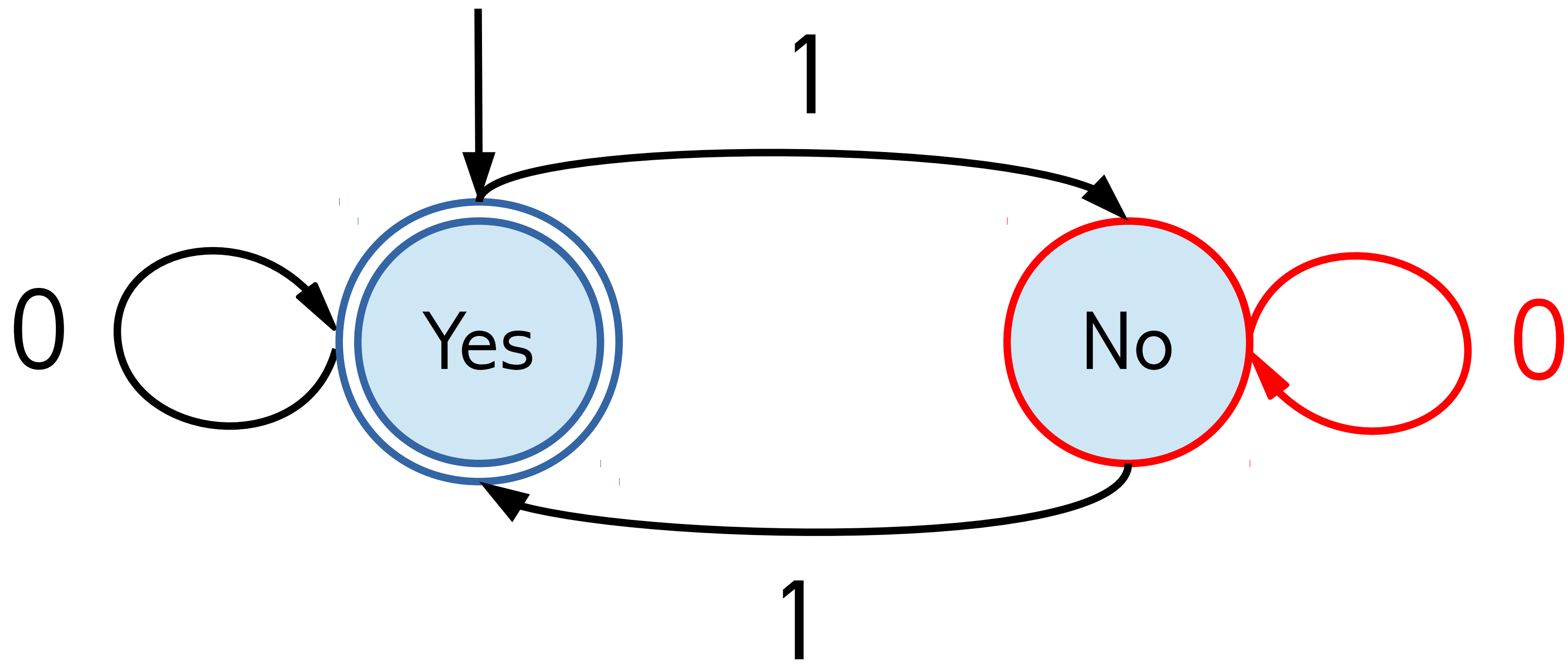

Input: 01001

# An example



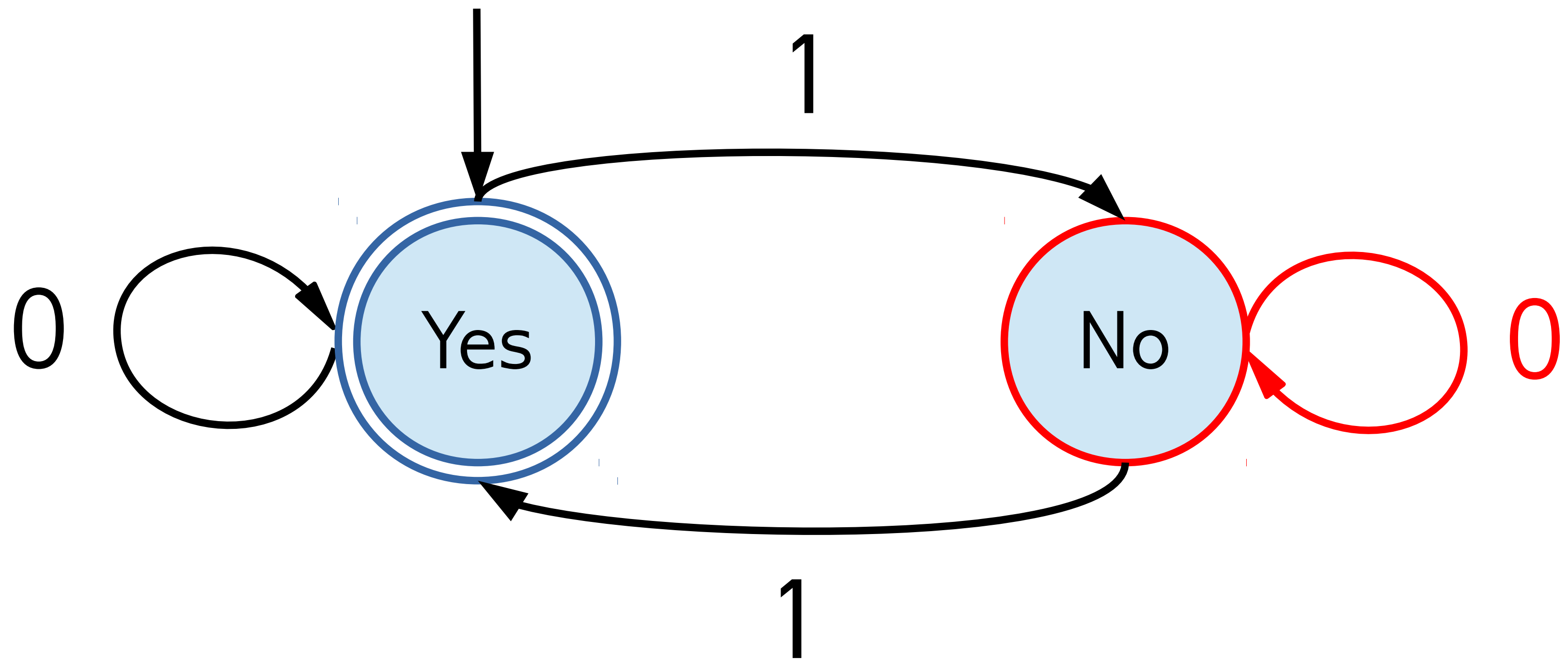

Input: 01001

# An example



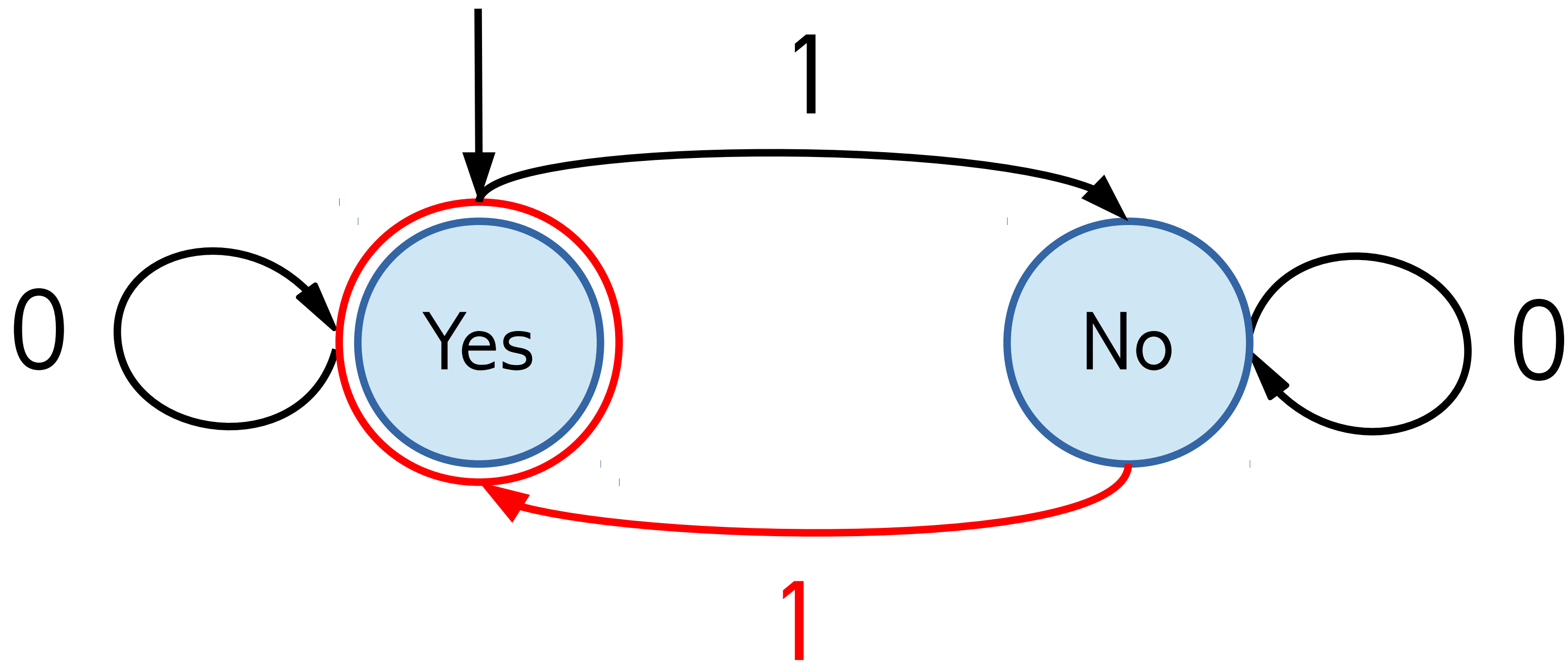

Input: 01001

# An example



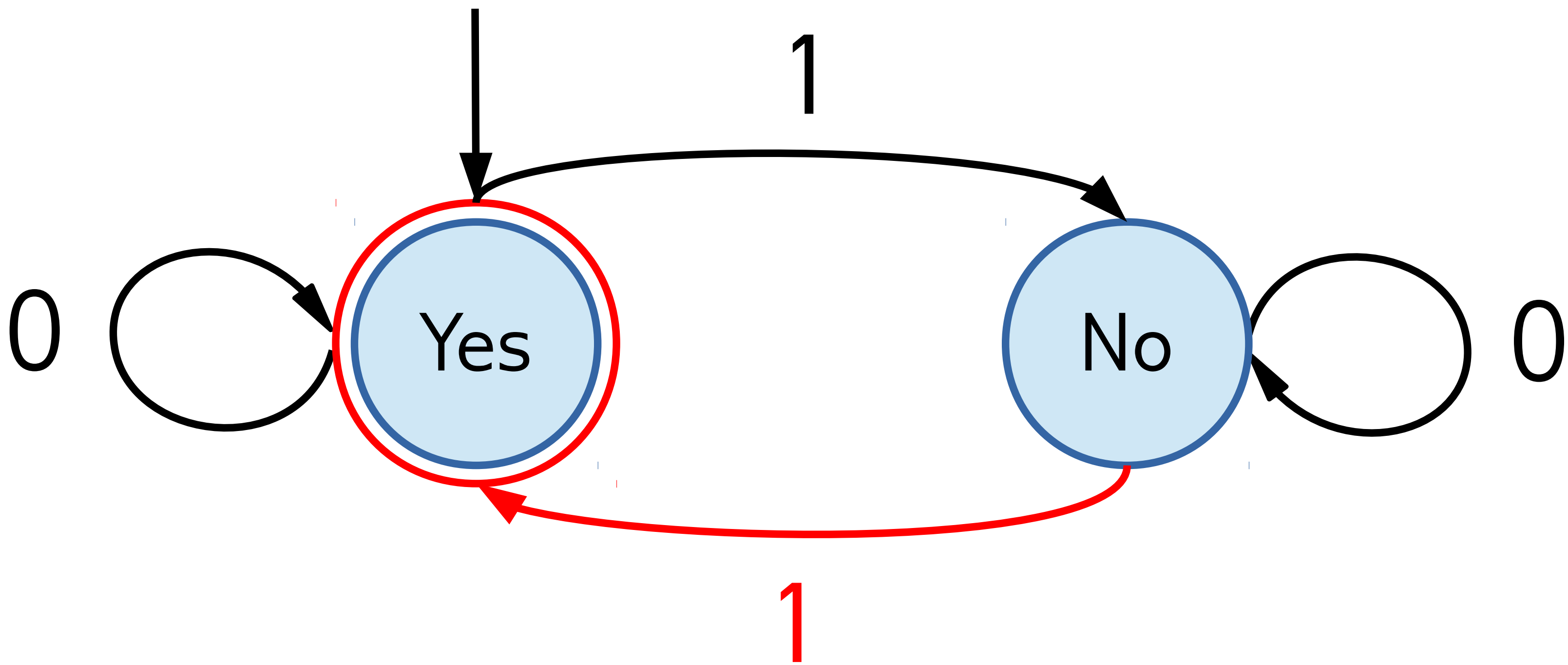

Input: 01001

# An example




Input: 01001

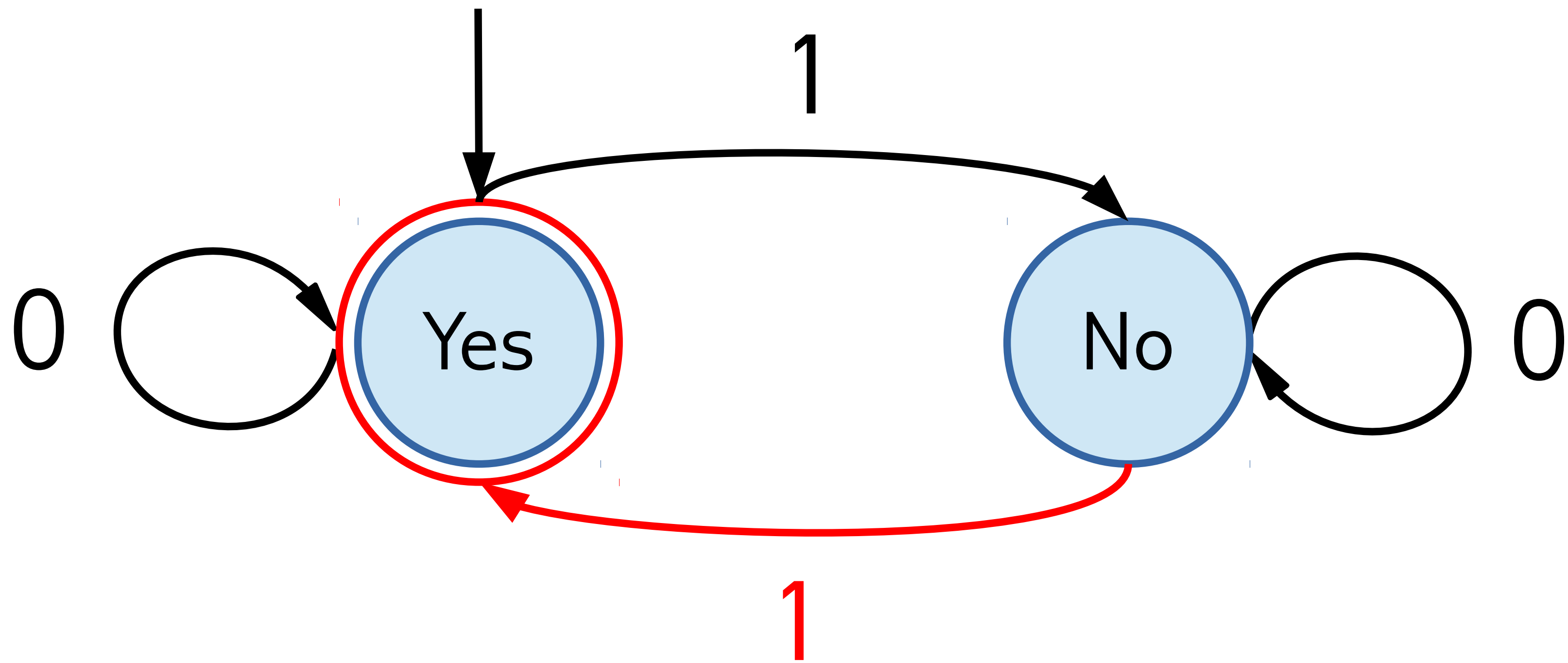
# An example




Input: 01001

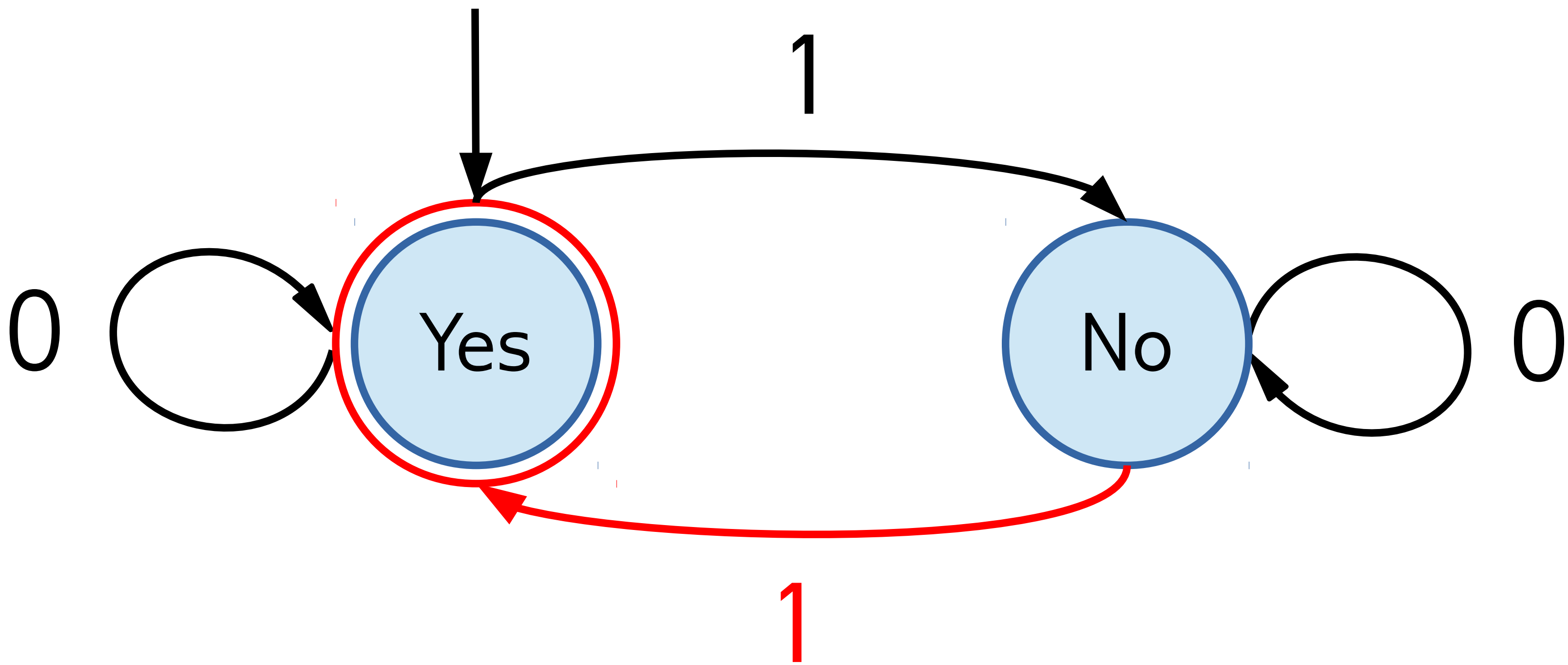
Output: Yes!

# An example




In general, on what binary strings does this DFA return Yes?

# An example




*Ans:* All strings with an even number of 1's

# Deterministic Finite Automaton

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$

# Deterministic Finite Automaton

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - $Q$ is a finite set of states

Yes

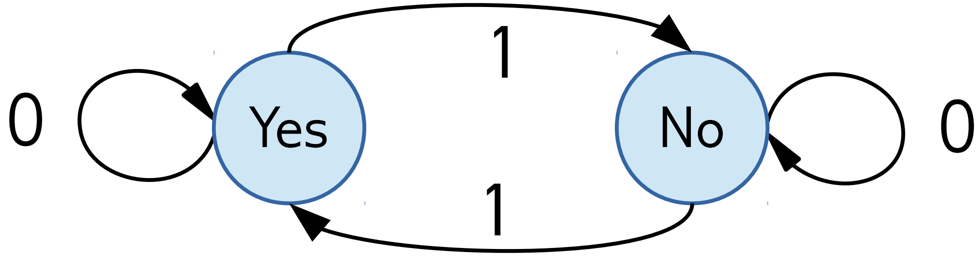
No

# Deterministic Finite Automaton

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - $Q$ is a finite set of states
  - $\Sigma$ is a finite input alphabet (e.g. $\{0, 1\}$)

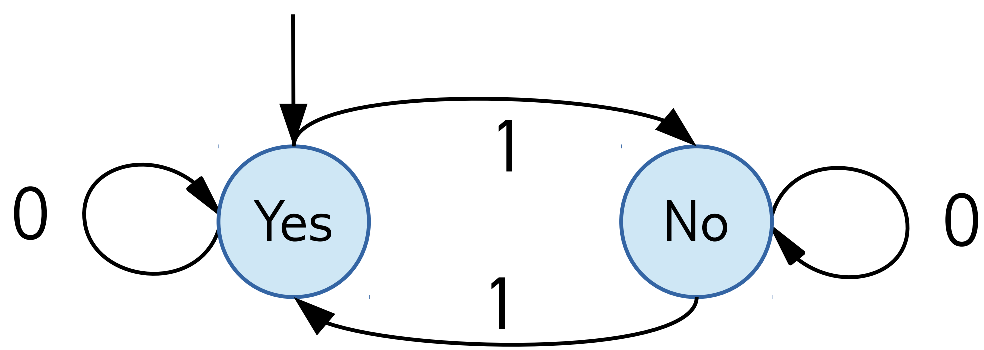
# Deterministic Finite Automaton

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - $Q$ is a finite set of states
  - $\Sigma$ is a finite input alphabet (e.g. $\{0, 1\}$)
  - $\delta$ is a transition function $\delta : Q \times \Sigma \to Q$



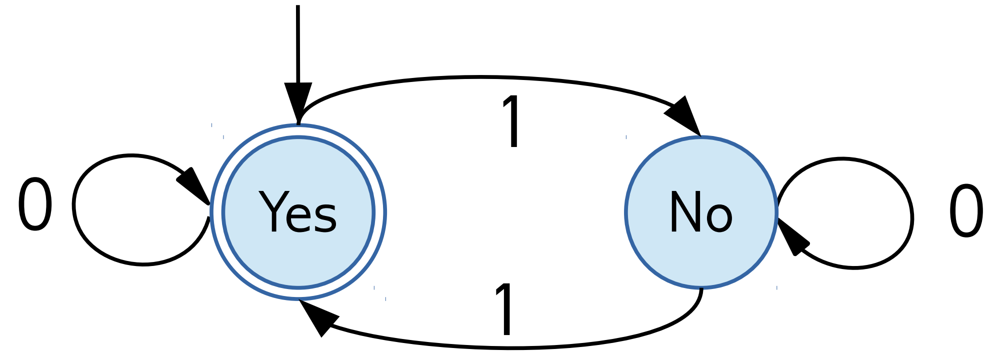
# Deterministic Finite Automaton

- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - $Q$ is a finite set of states
  - $\Sigma$ is a finite input alphabet (e.g. $\{0, 1\}$)
  - $\delta$ is a transition function $\delta : Q \times \Sigma \to Q$
  - $q_0 \in Q$ is the start/initial state

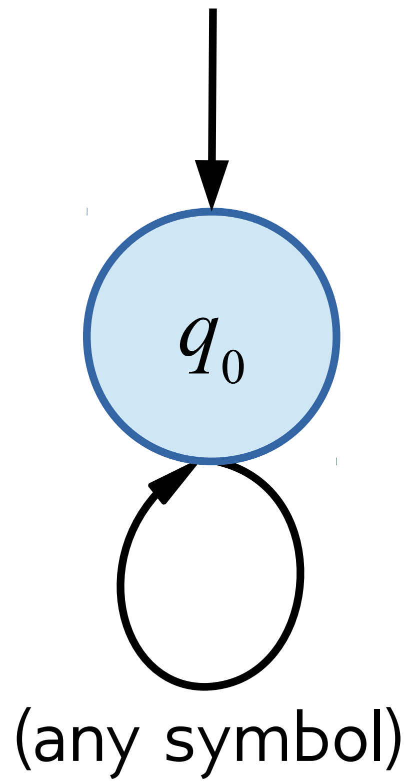


# Deterministic Finite Automaton

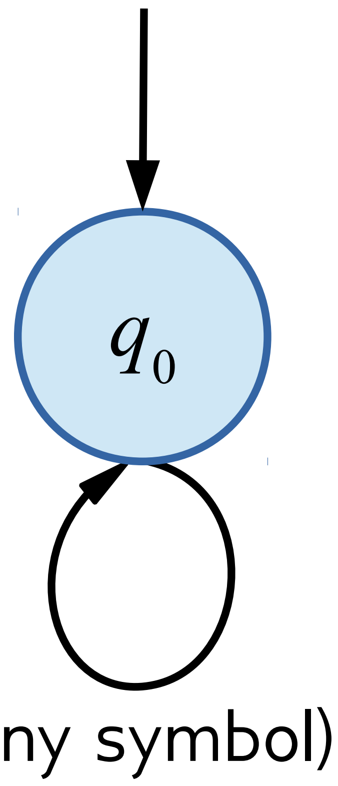
- A DFA is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$
  - $Q$ is a finite set of states
  - $\Sigma$ is a finite input alphabet (e.g. $\{0, 1\}$)
  - $\delta$ is a transition function $\delta : Q \times \Sigma \to Q$
  - $q_0 \in Q$ is the start/initial state
  - $F \subseteq Q$ is the set of final/accepting states



# What does this DFA accept?

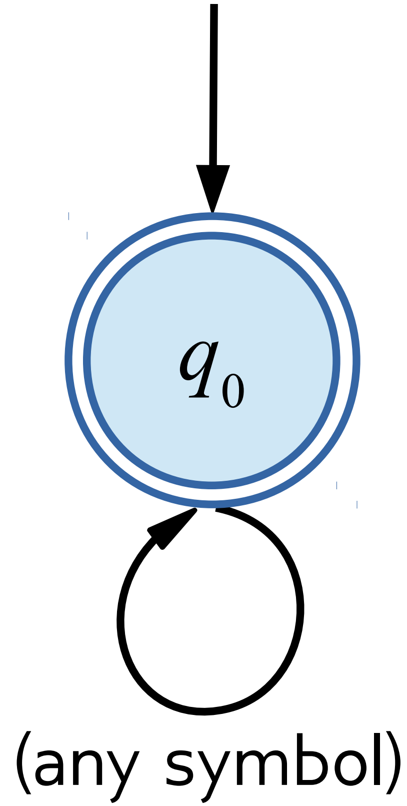


# What does this DFA accept?

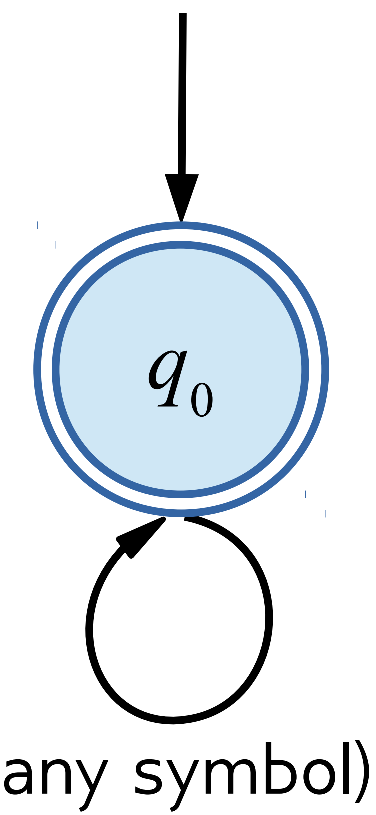



Answer: No strings

# What does this DFA accept?

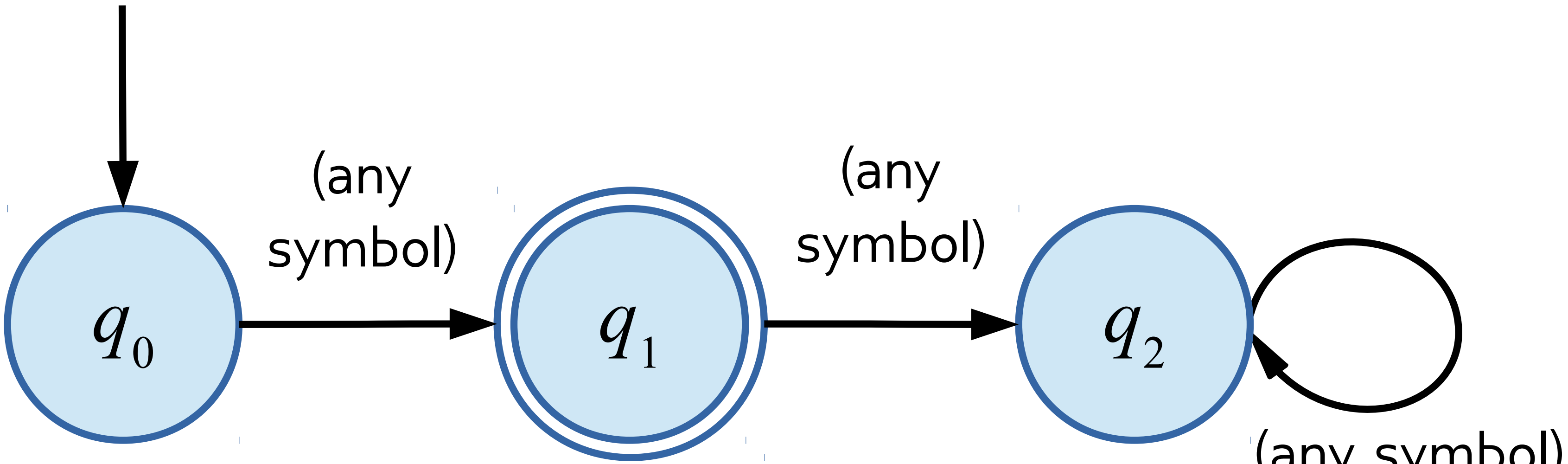


# What does this DFA accept?

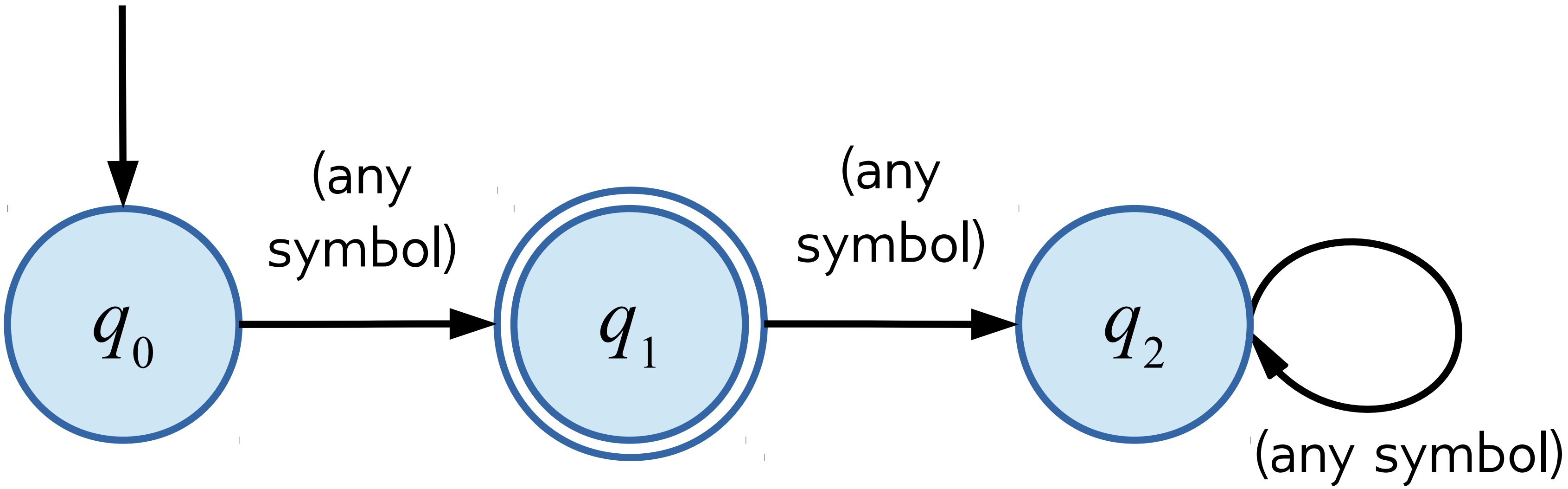



Answer: All strings

# What does this DFA accept?

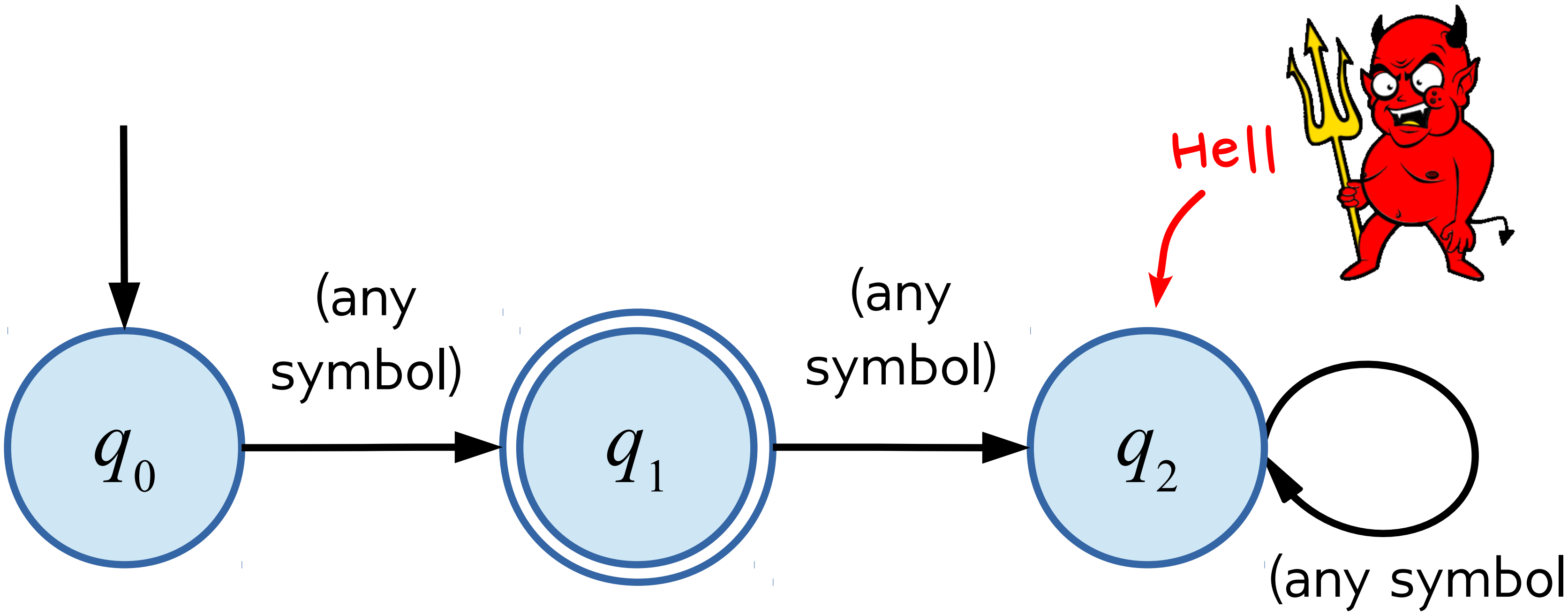


# What does this DFA accept?



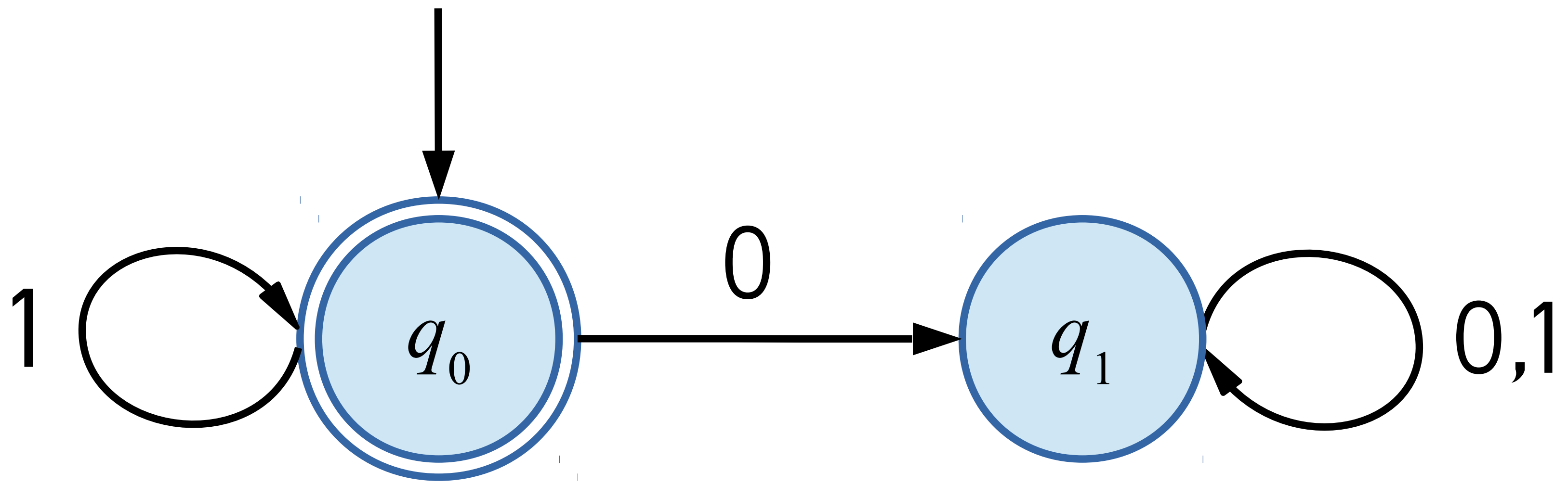

Answer: Strings of length 1

# What does this DFA accept?

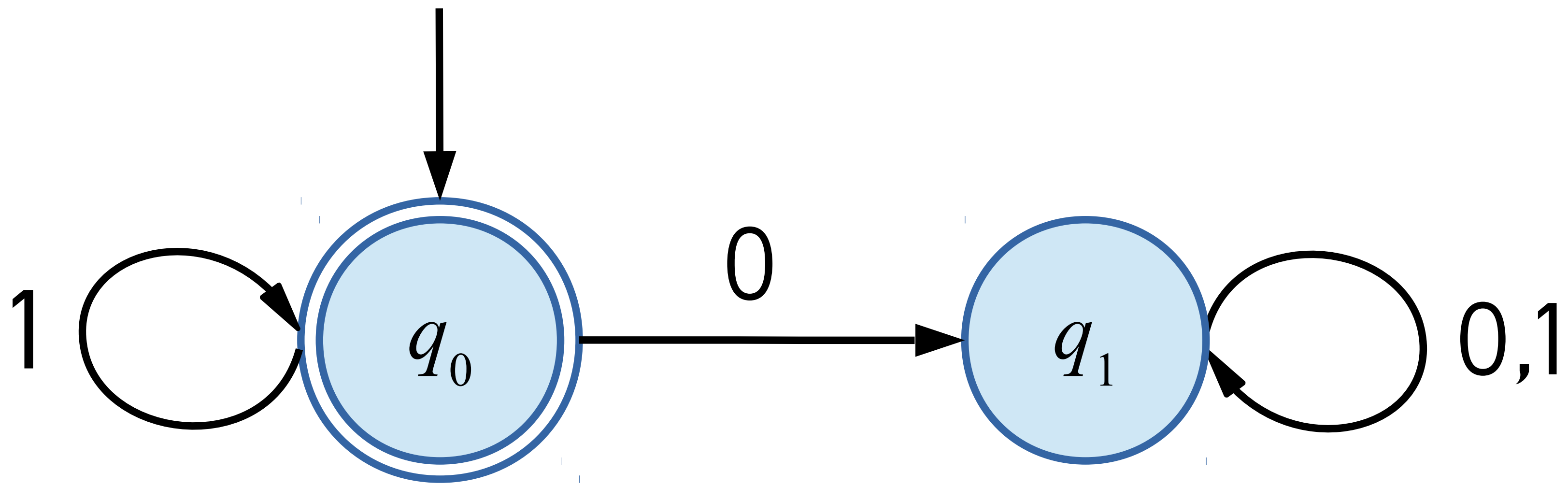



Answer: Strings of length 1

# What does this DFA accept?

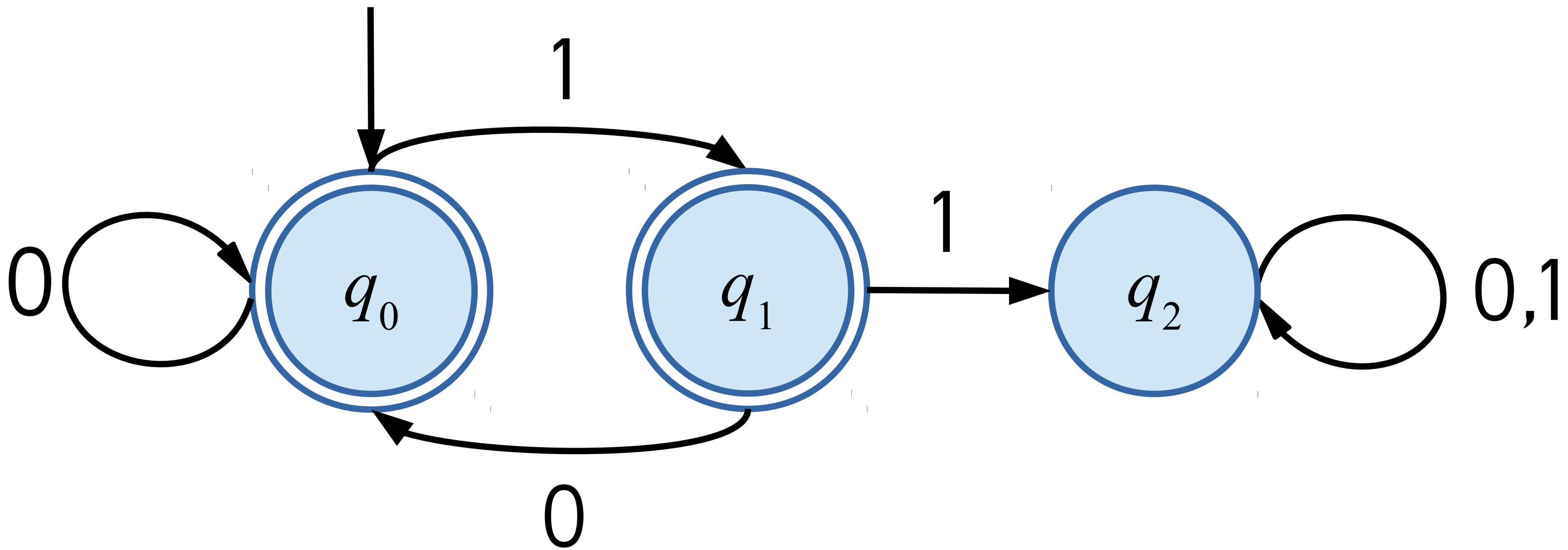


# What does this DFA accept?

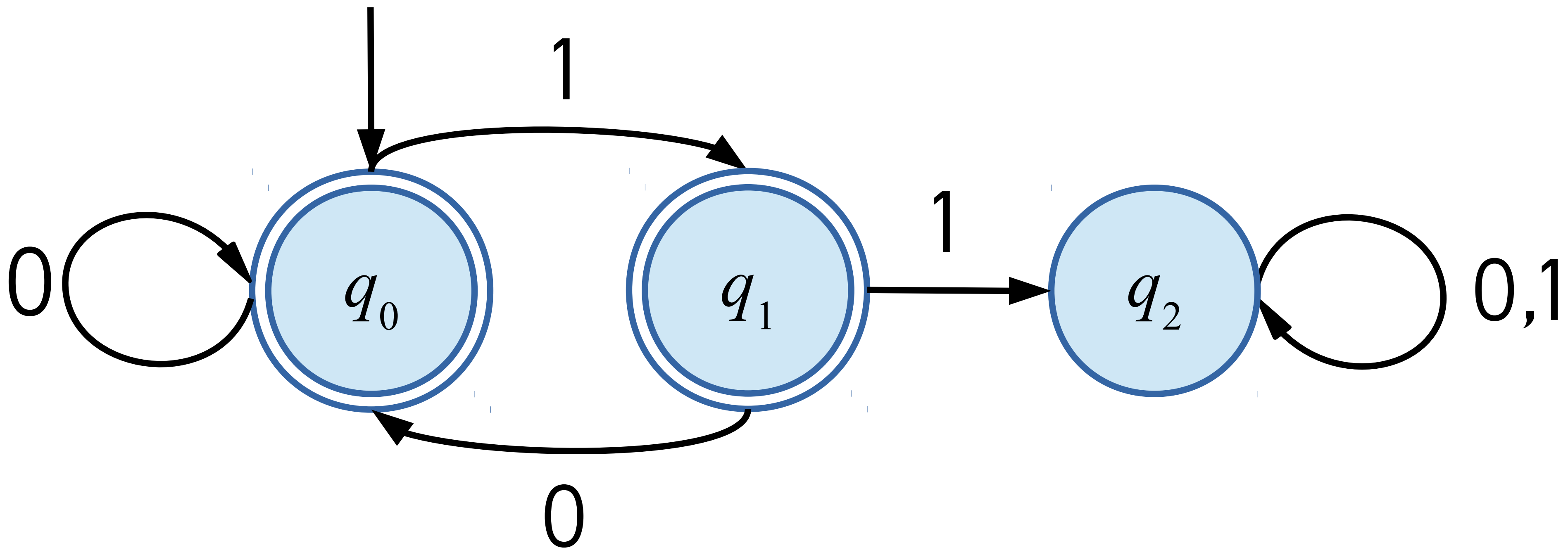



Answer: Strings containing only 1's

# What does this DFA accept?



# What does this DFA accept?




Answer: Strings containing no two consecutive 1's

# Language

- Given alphabet $\Sigma$, a language $L$ is a set of strings over the alphabet, i.e. $L \subseteq \Sigma^*$

# Language

- Given alphabet $\Sigma$, a language $L$ is a set of strings over the alphabet, i.e. $L \subseteq \Sigma^*$

Set of all possible strings over $\Sigma$

- We say a language $L$ is accepted/recognized by a DFA $M$, if $M$ accepts input string $x \in \Sigma^*$ if and only if $x \in L$

# Language

- Given alphabet $\Sigma$, a language $L$ is a set of strings over the alphabet, i.e. $L \subseteq \Sigma^*$

Set of all possible strings over $\Sigma$

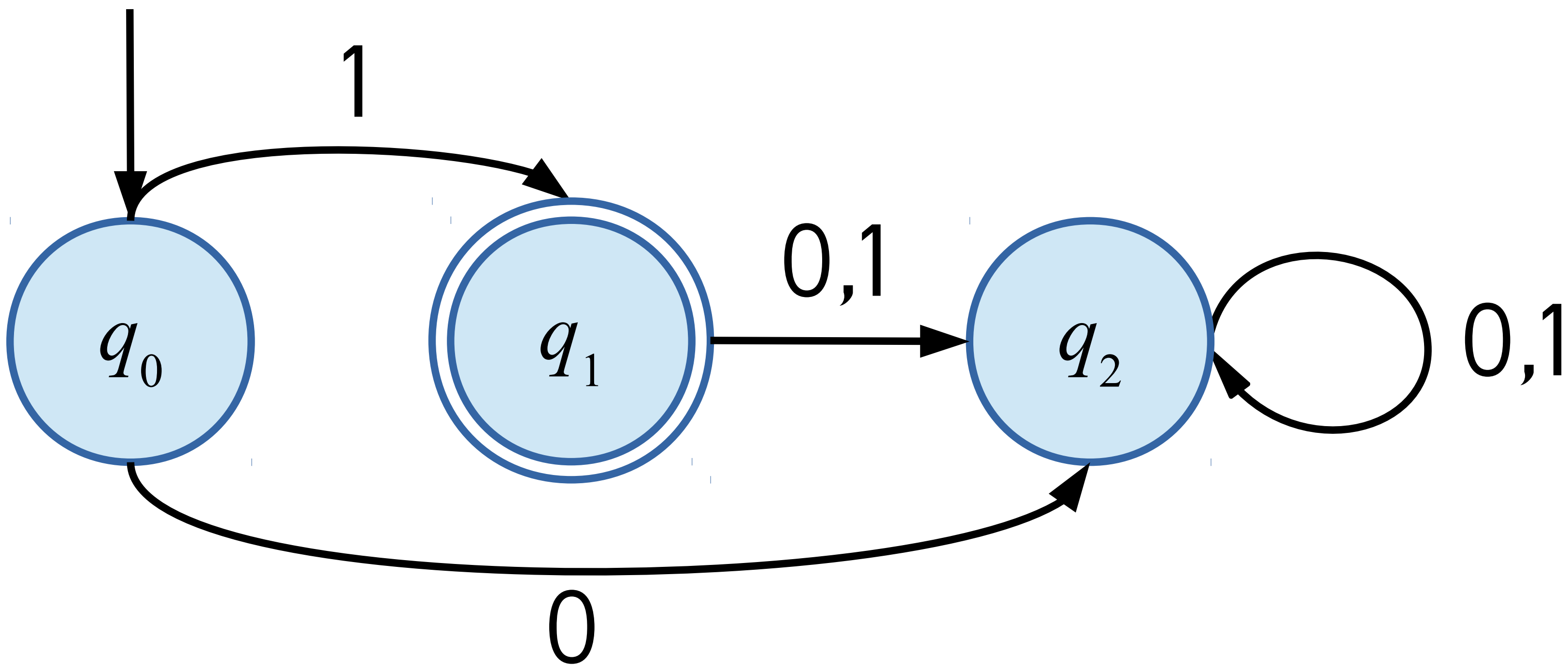
# Language

- Given alphabet $\Sigma$, a language $L$ is a set of strings over the alphabet, i.e. $L \subseteq \Sigma^*$

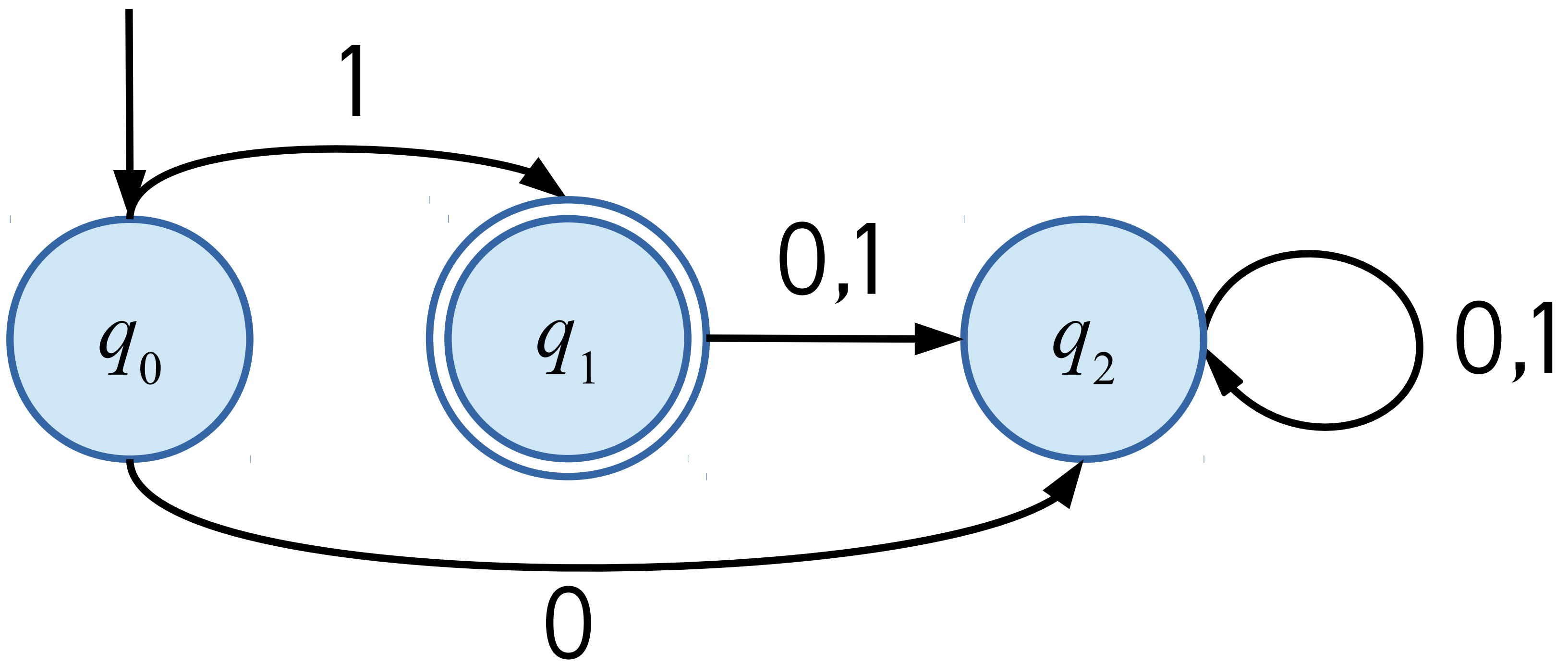
Set of all possible strings over $\Sigma$

- We say a language $L$ is accepted/recognized by a DFA $M$, if $M$ accepts input string $x \in \Sigma^*$ if and only if $x \in L$

# What language does this DFA accept?

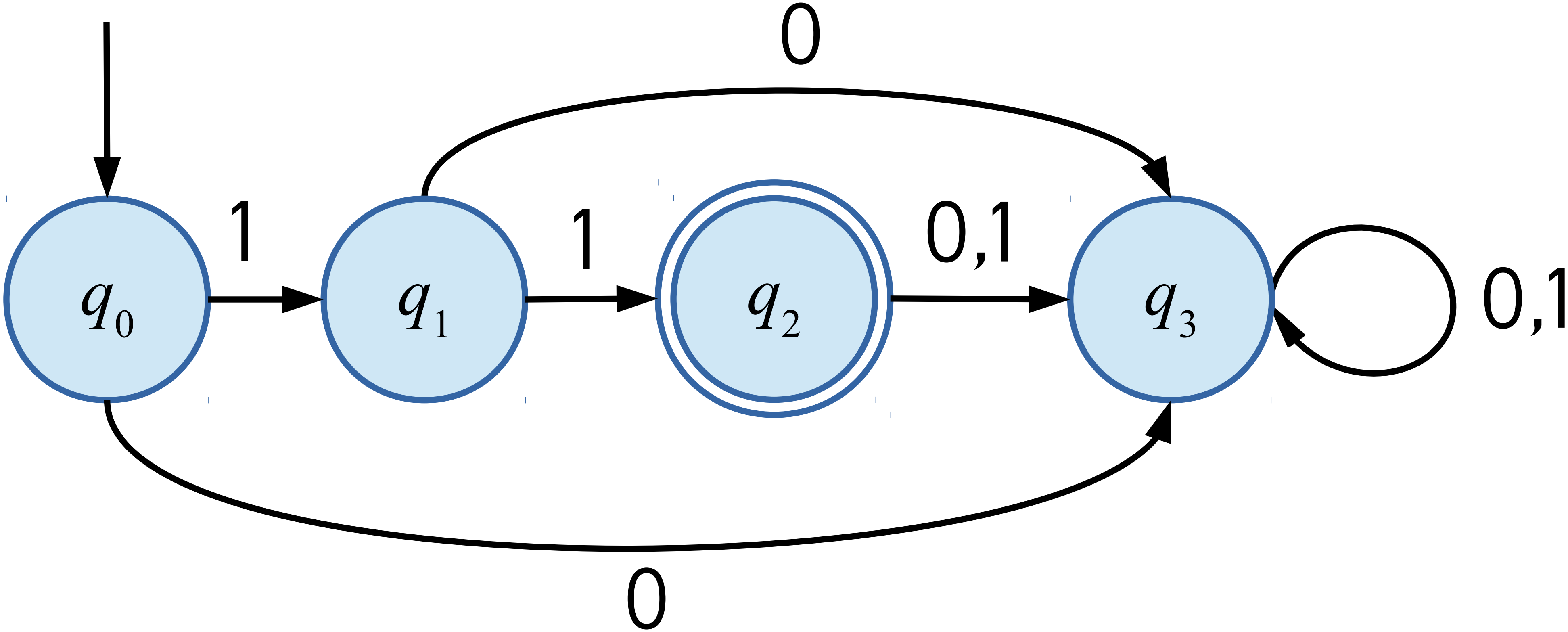


# What language does this DFA accept?

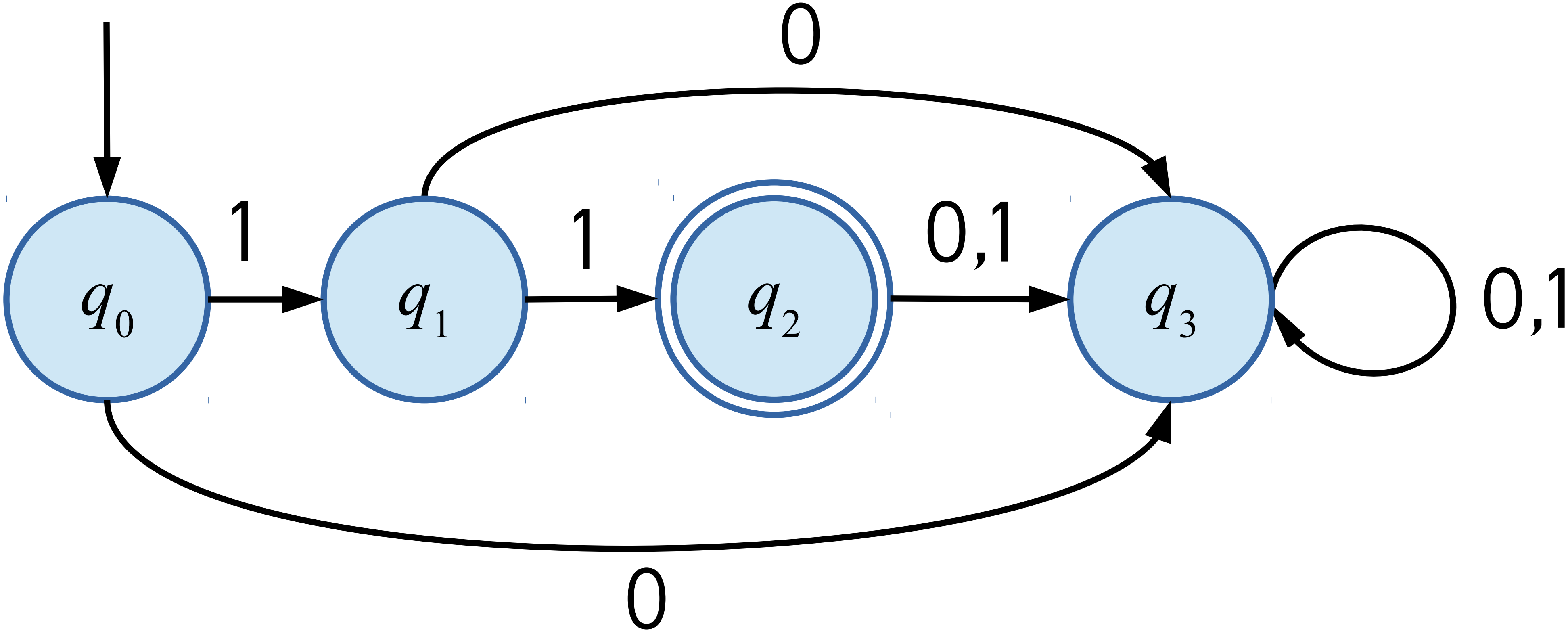



Answer: Only the string 1

# What language does this DFA accept?



# What language does this DFA accept?




Answer: Only the string 11

# DFA's find it difficult to count

- A DFA that recognizes the language $\{1^c\}$ (the single string of $c$ 1's) must have at least $c$ states

# DFA's find it difficult to count

- A DFA that recognizes the language $\{1^c\}$ (the single string of $c$ 1's) must have at least $c$ states
  - The parent alphabet is irrelevant (but must of course contain 1)

(Proof discussion to be completed next class)