Finite Automata

Lecture 4

October 17, 2021

Objectives

By the end of this lecture, you should be able to

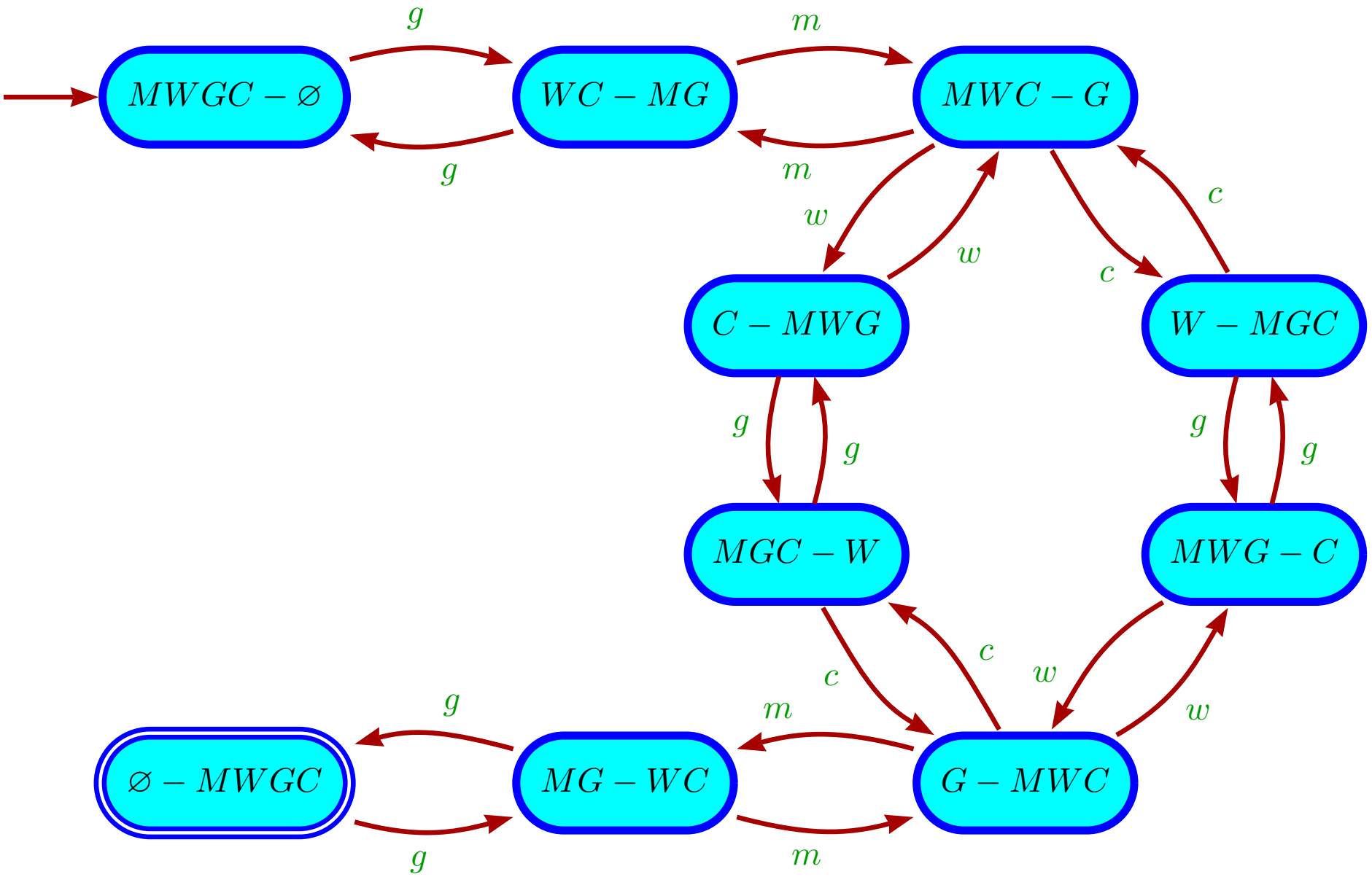
- Design simple finite automata.
- Identify the language recognized by a finite automaton.
- Formally define a finite automaton.

The Ferryman and His Companions

Once upon a time there was a man on the left bank of a river. Along with the man, there was a wolf, goat, and cabbage. The man had a boat large enough to hold him with only one of his three companions. The man wishes to ferry the wolf, goat, and cabbage across the river, one at a time. The problem is that if the man leaves the wolf and goat unattended, on either shore, the wolf will eat the goat. If he leaves the goat and cabbage unattended, the goat will eat the cabbage.

Can all four safely cross the river?

Sketch of a solution

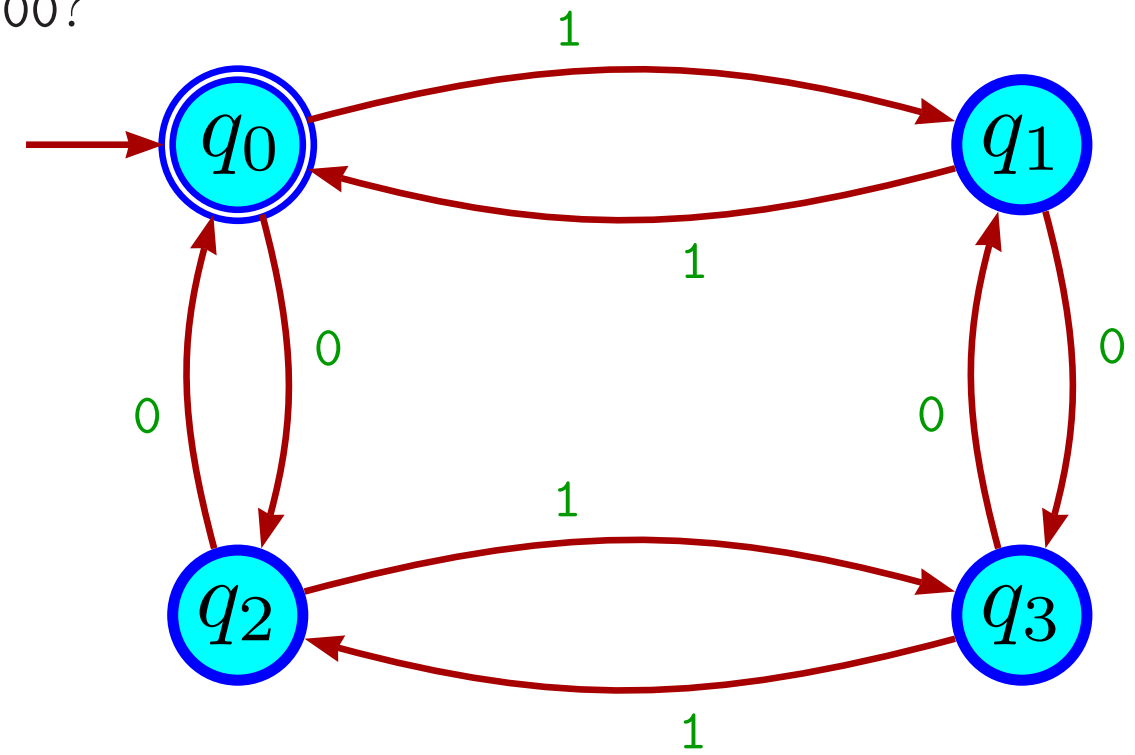


Some Reflections

- Consider the sequences of actions of the ferryman as strings.
- Only some of those strings are acceptable. (Actually, *infinitely some*).
- Those acceptable strings define a language.
- It is the language *recognized* by the automaton (machine) depicted in the previous slide.
- The automaton has a finite set of states that it can enter.
  - It is called a (deterministic) **finite automaton** (a.k.a. finite state machine).
  - We shall use the acronym **DFA**.

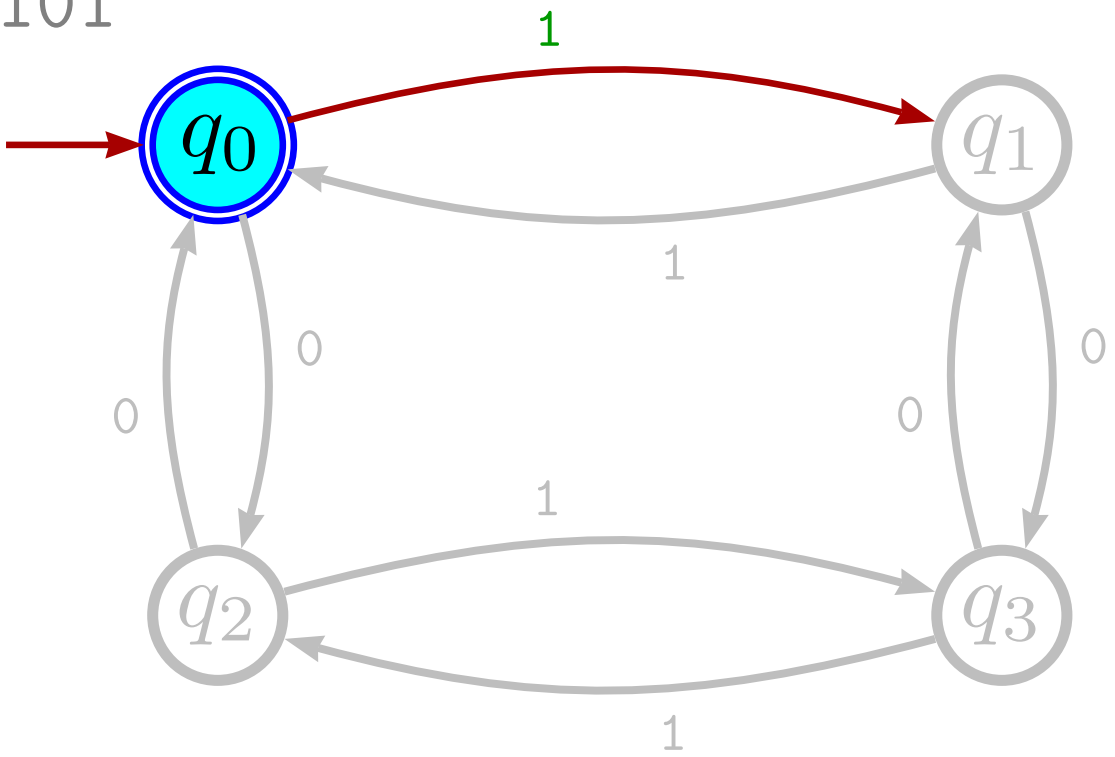
Tracing DFA

- You need to have no doubts about how DFA work.
- **Example**: How would the following automaton behave when given the inputs
  1. 101101
  2. 0100?



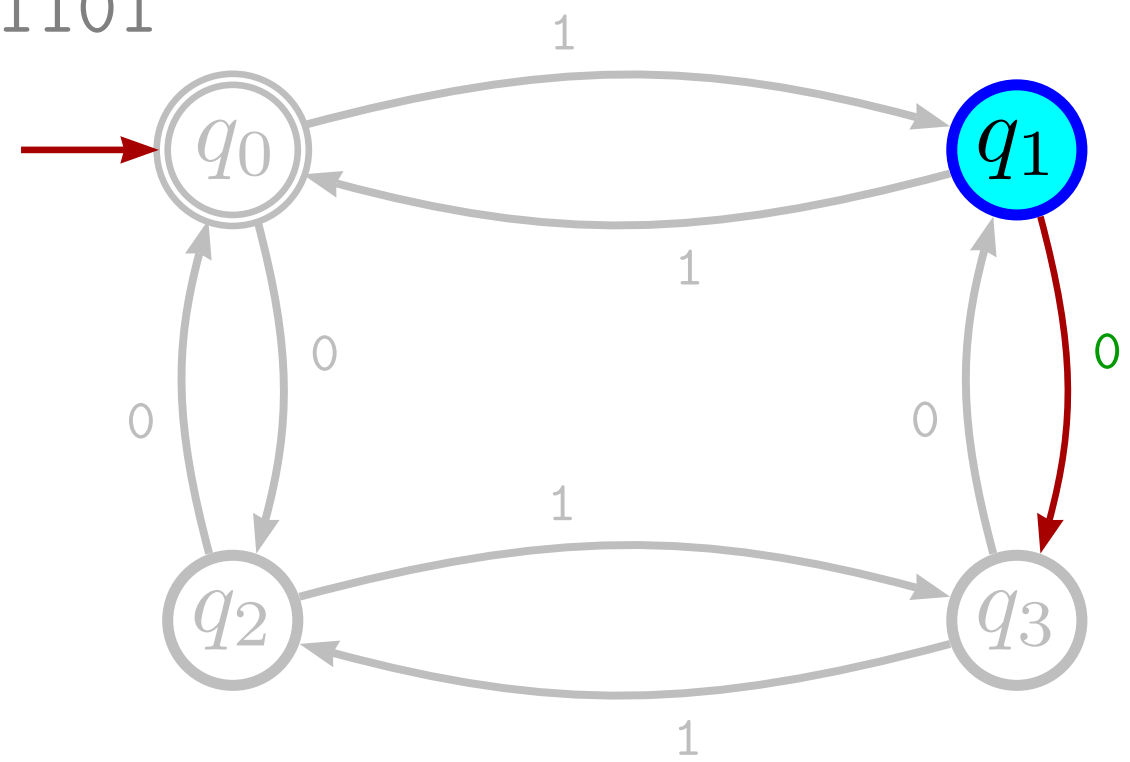
Tracing DFA

- 1 01101



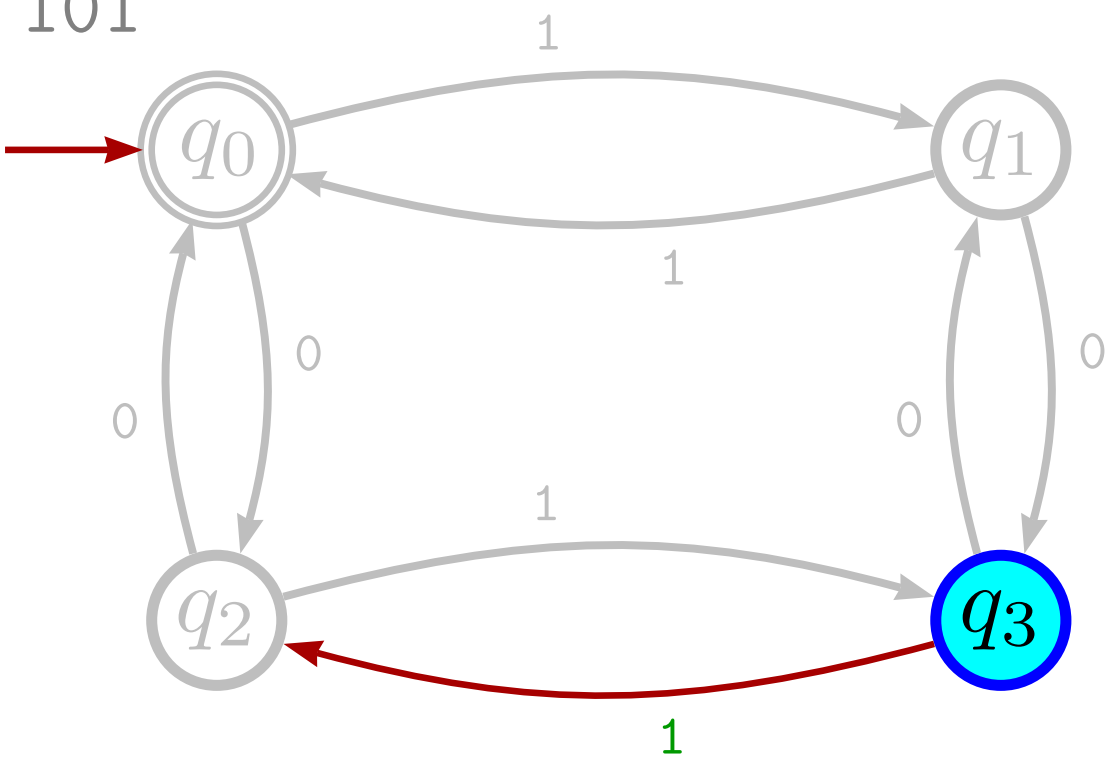
Tracing DFA

- 1 0 1101



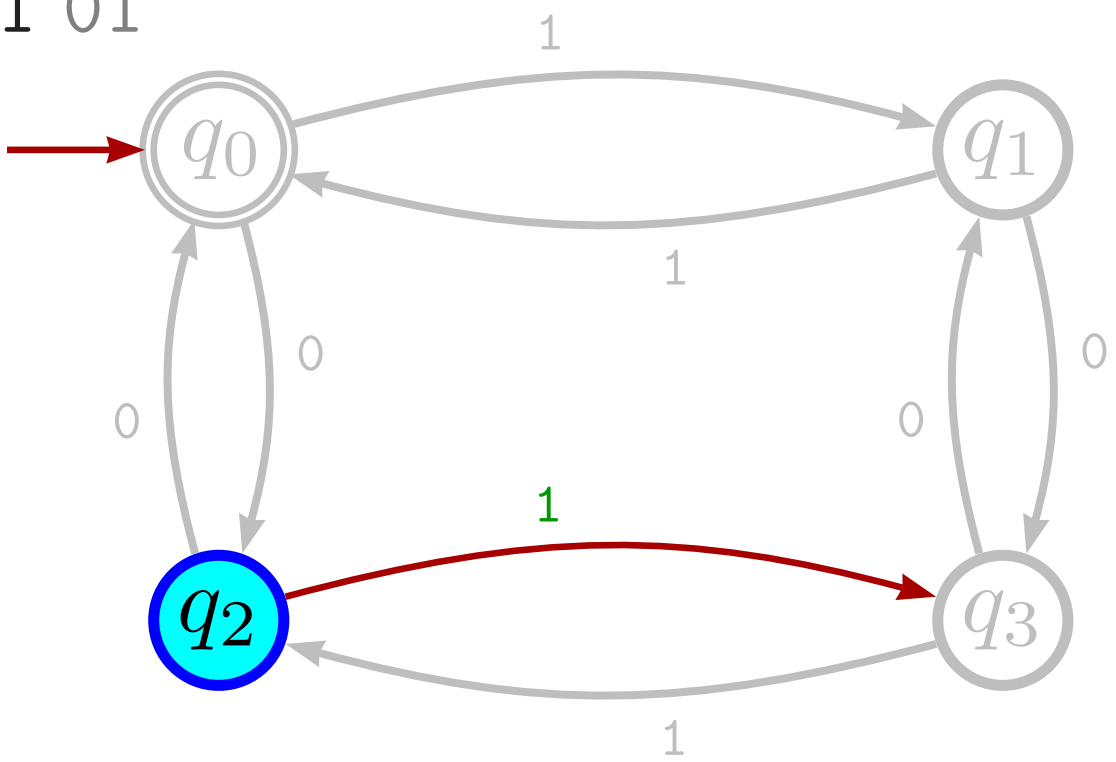
Tracing DFA

- 10 1 101



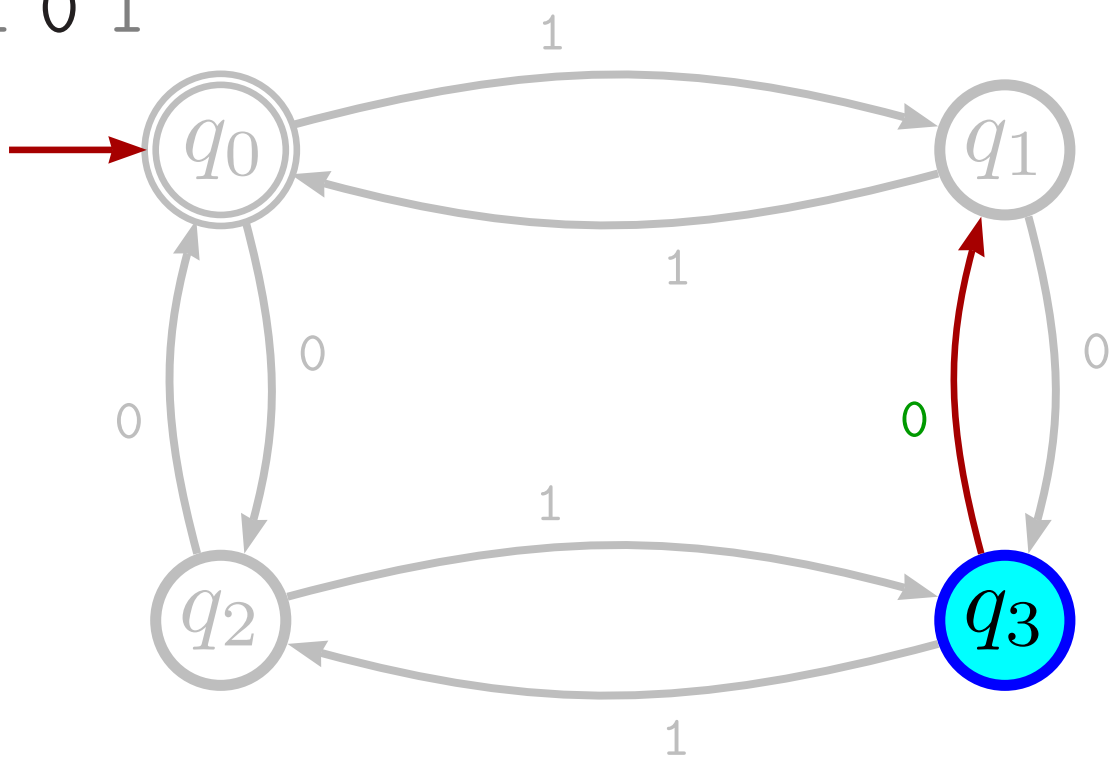
Tracing DFA

- 101 1 01



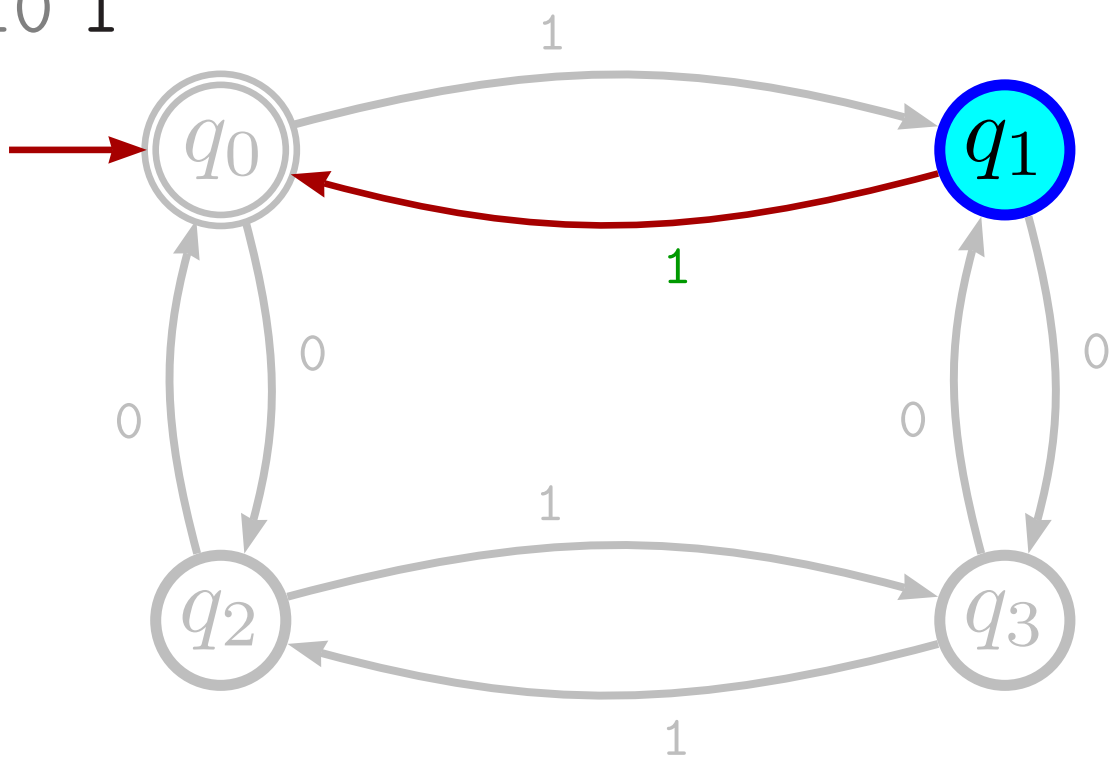
Tracing DFA

- 1011 0 1



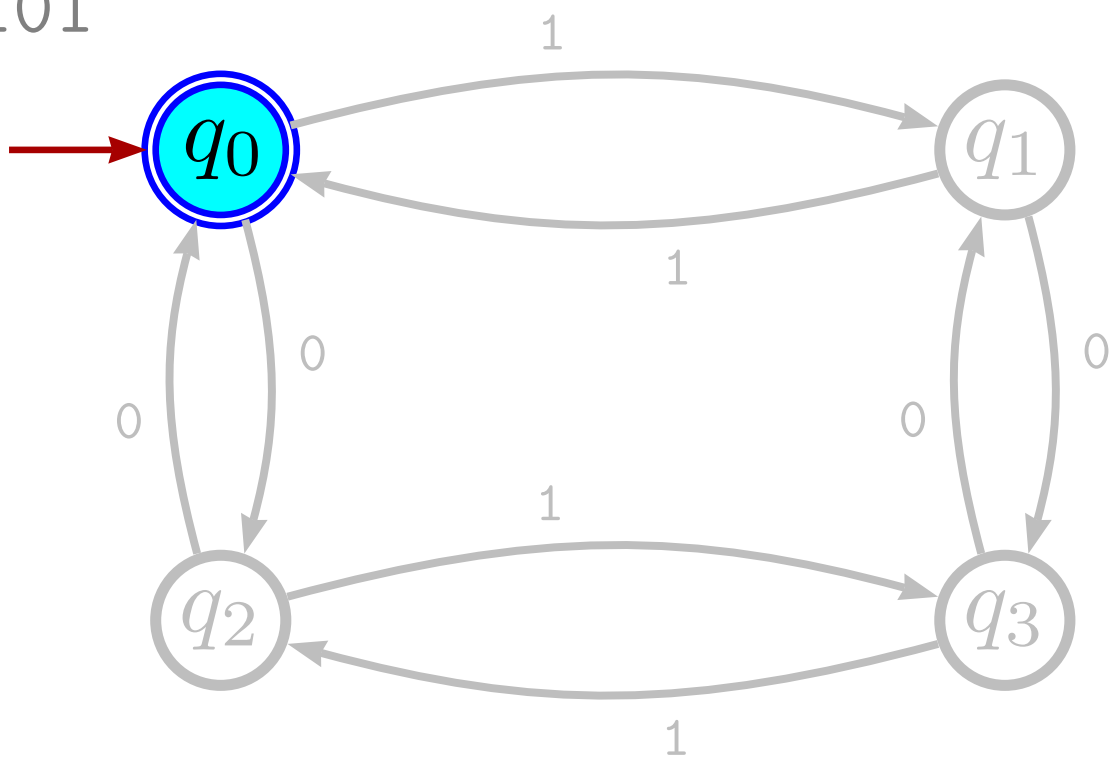
Tracing DFA

- 10110 1



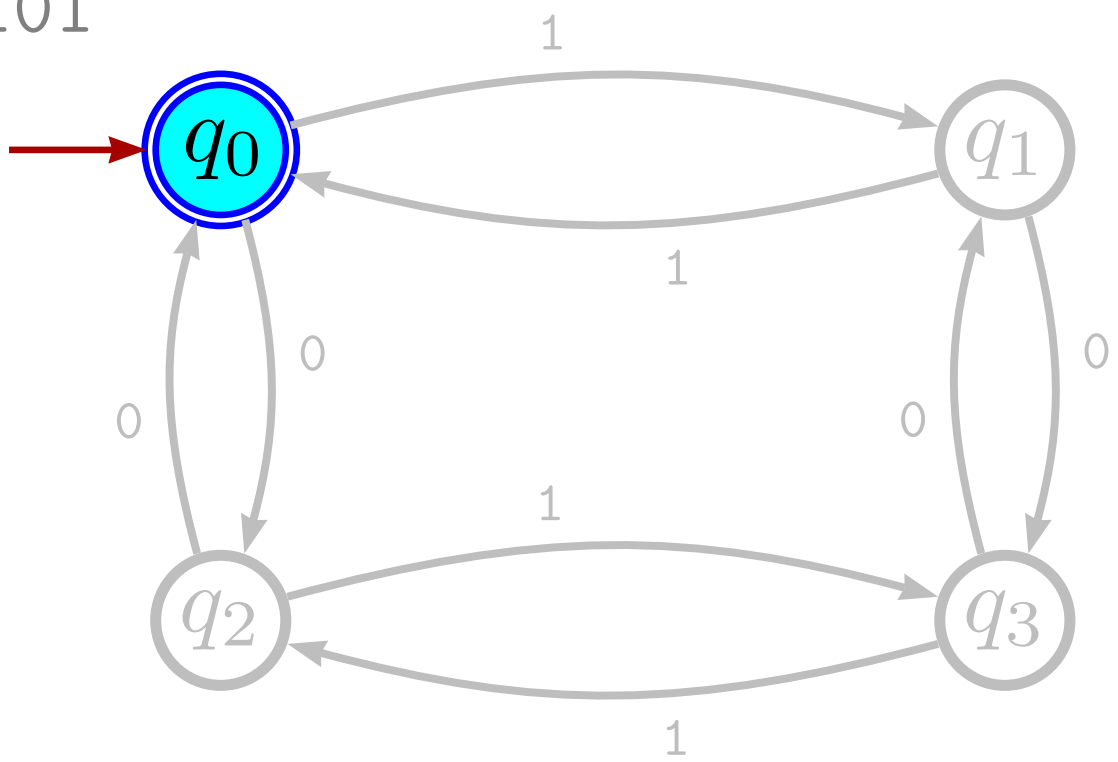
Tracing DFA

- 101101



Tracing DFA

- 101101

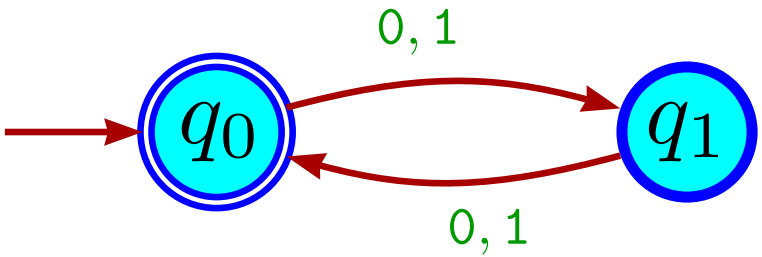


- What is the language recognized by this automaton?

Example

- Design a DFA that recognizes the language $L = \{w | w \in \{0, 1\}^*$ and $|w|$ is even $\}$.

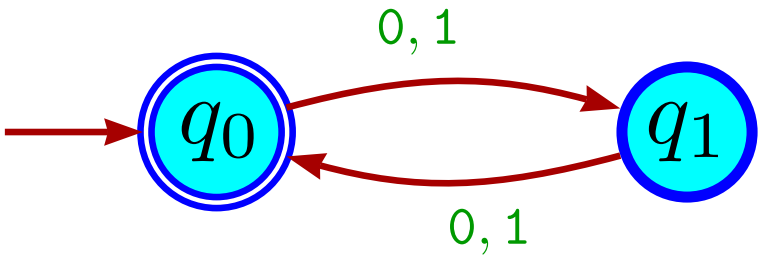
Solution



Remarks

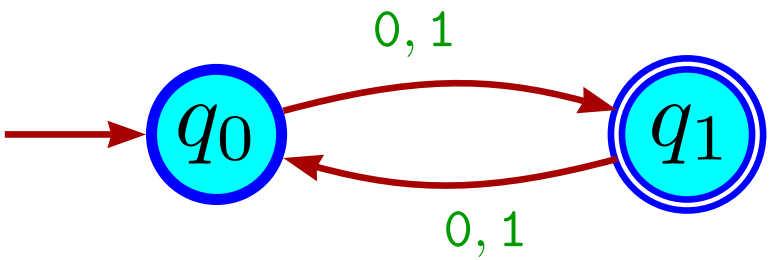
- Each state has a meaning.
- You should be able to write a statement indicating what each state encodes.
- **Accept states** should encode a statement that is satisfied by all and only those strings in the language to be recognized.

Solution



- q_0: length of string scanned so far is even.
- q_1: length of string scanned so far is odd.

Example



- Describe the language recognized by the above DFA. What does each state encode?
- Do it yourself.

DFA Design

- Imagine that you are the automaton.
- You can see the input, one symbol at a time.
- You cannot remember the whole string.
- How many pieces of information do you really need to remember?
- Note that you have to be ready to accept or reject at any time; you do not know when you will run out of input.
- All relevant, distinguishable pieces of information are represented by states.

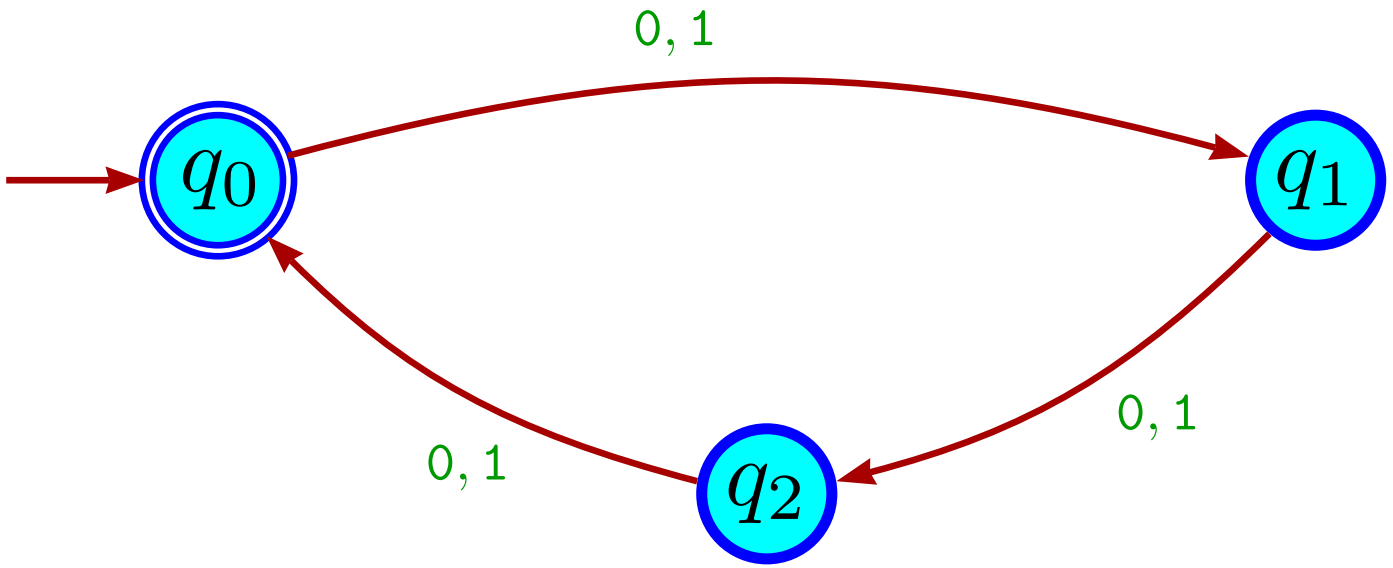
Example

- Design a DFA that recognizes the language $L = \{w | w \in \{0, 1\}^*$ and $|w|$ is a multiple of 3 $\}$.

Questions to Ask

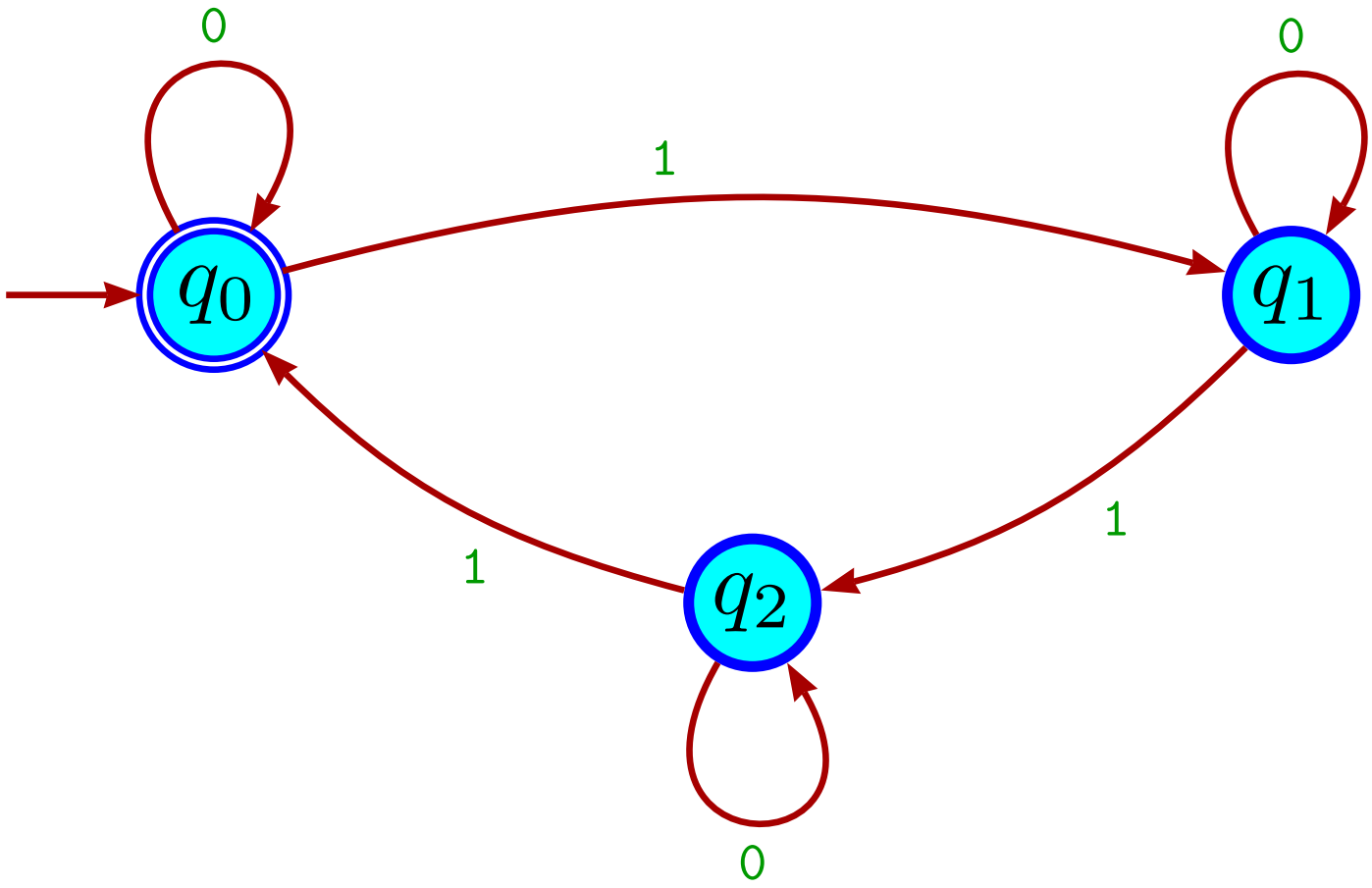
- How many states do I have to distinguish?
- What does each state encode?
- Which states are accept states?
- How should the automaton react to different inputs at different states?

Solution



- What do the different states encode?

Example

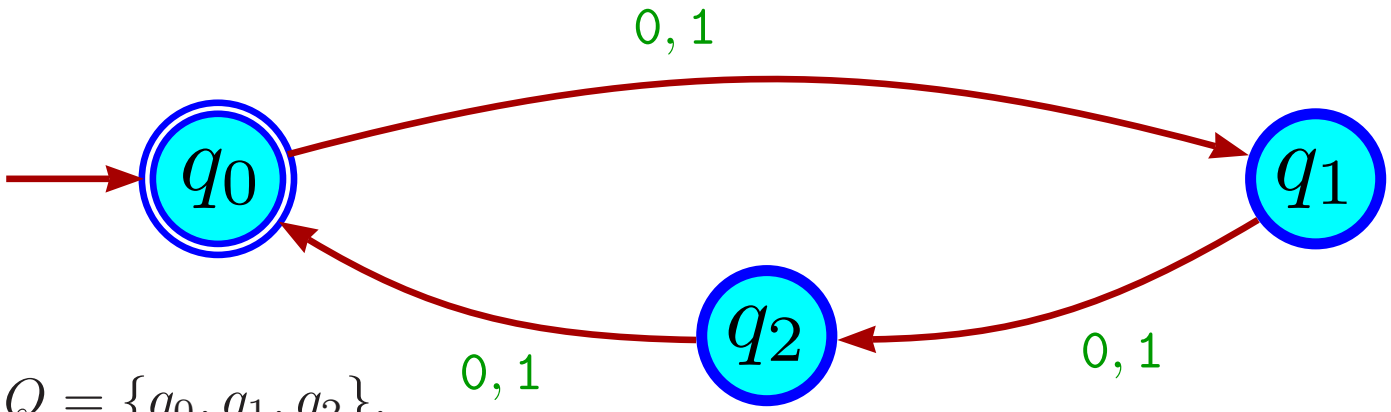


- What language is recognized by the above DFA?

Formal Definition of a DFA

- A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where
  1. Q is a finite set of states,
  2. Σ is an alphabet,
  3. $\delta : Q \times \Sigma \longrightarrow Q$ is the transition function,
  4. $q_0 \in Q$ is the start state, and
  5. $F \subseteq Q$ is the set of accept state.

Example



1. $Q = \{q_0, q_1, q_2\}$.
2. $\Sigma = \{0, 1\}$.
3. $\delta(q_0, 0) = q_1 \;\; \delta(q_0, 1) = q_1$
   $\delta(q_1, 0) = q_2 \;\; \delta(q_1, 1) = q_2$
   $\delta(q_2, 0) = q_0 \;\; \delta(q_2, 1) = q_0$
4. q_0 is the start state.
5. $F = \{q_0\}$.

The Language of a DFA

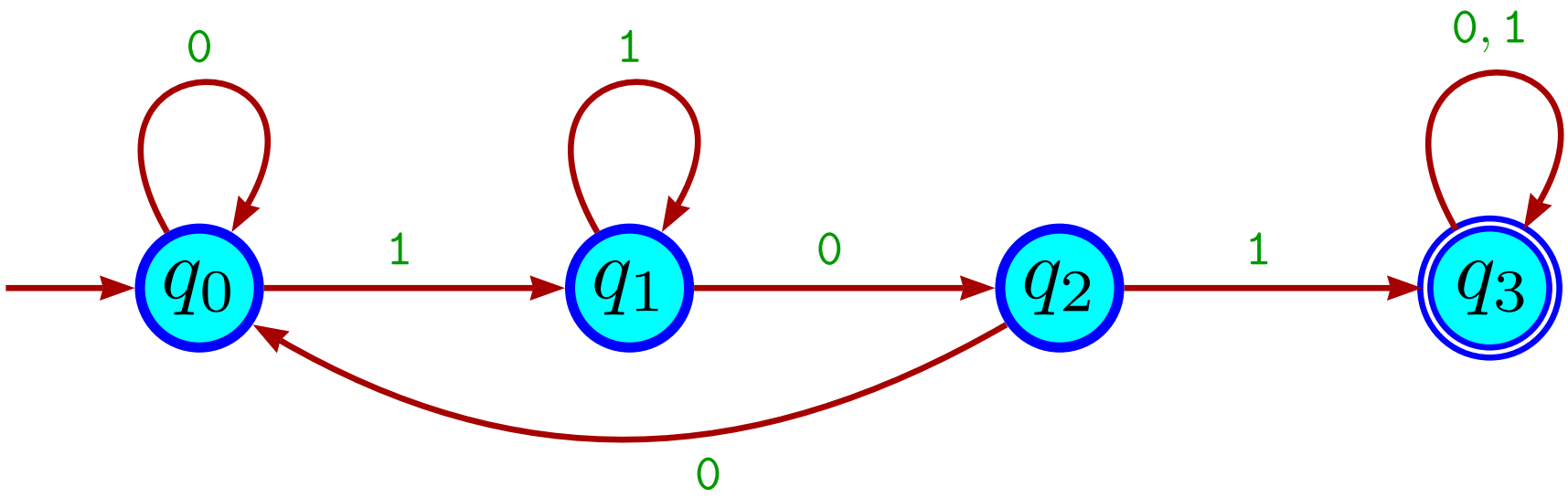
- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.
- Let $w = w_1 w_2 \cdots w_n$ be a string in Σ^*.
- M **accepts** w if there is a sequence of states $r_0, r_1, \cdots, r_n$ in Q such that
  1. $r_0 = q_0$,
  2. $\delta(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, 1, \ldots n-1$, and
  3. $r_n \in F$.
- The **language of** M (or the language recognized by M) is the set

$$L(M) = \{w |\ M \text{ accepts } w\}$$

Example

- Design a DFA that recognizes the language $L = \{w | w \in \{0, 1\}^*$ and w has 101 as a substring $\}$.

Solution



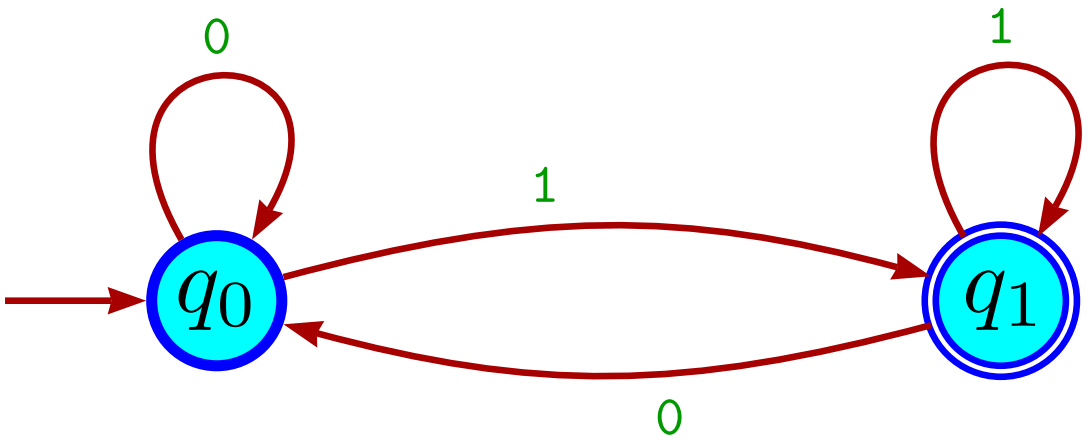
- Write the formal definition yourself.

Example

- Design a DFA that recognizes the language
  $L = \{w |\ w \in \{0, 1\}^*$ and the last symbol in w is a $1\}$

Example

- Design a DFA that recognizes the language
  $L = \{w |\ w \in \{0, 1\}^*$ and the last symbol in w is a $1\}$

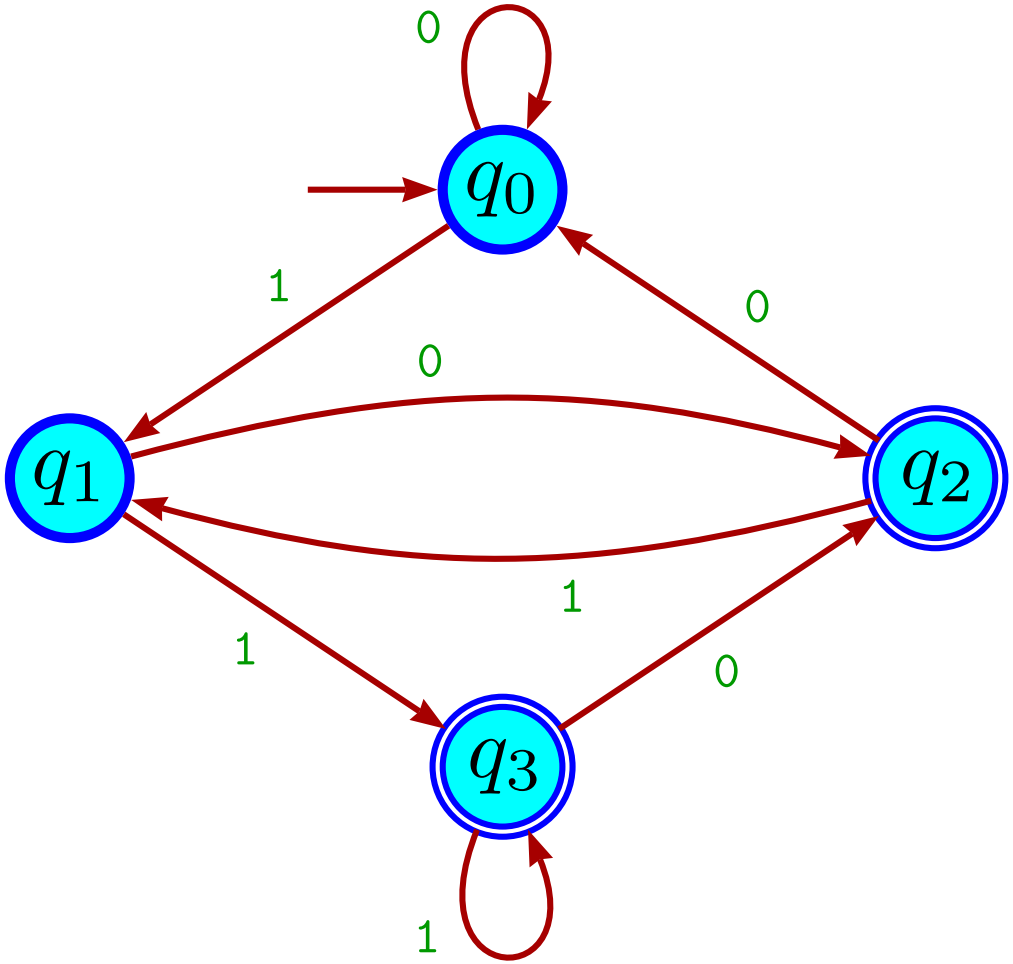


One Final Example

- Design a DFA that recognizes the language
  $L = \{w |\ w \in \{\mathtt{0}, \mathtt{1}\}^*$ and the symbol before last in w is a $\mathtt{1}\}$

One Final Example

- Design a DFA that recognizes the language
  $L = \{w |\ w \in \{0,1\}^*$ and the symbol before last in w is a $1\}$



Points to take home

- DFA.
- Language of a DFA.
- Designing DFA.

Next time

- Regular languages.
- Non-deterministic Finite Automata.