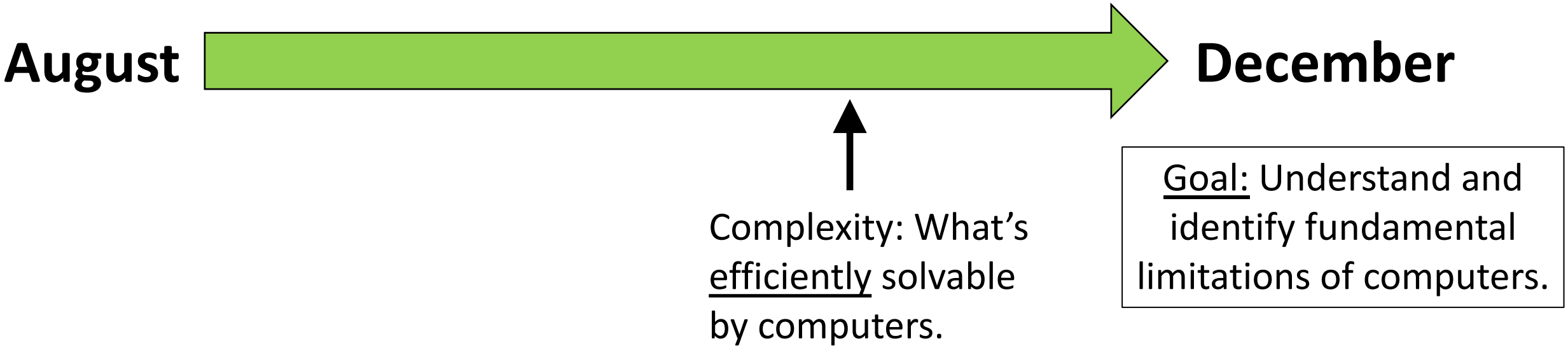
Finite Automata
CSCI 338

CSCI 338

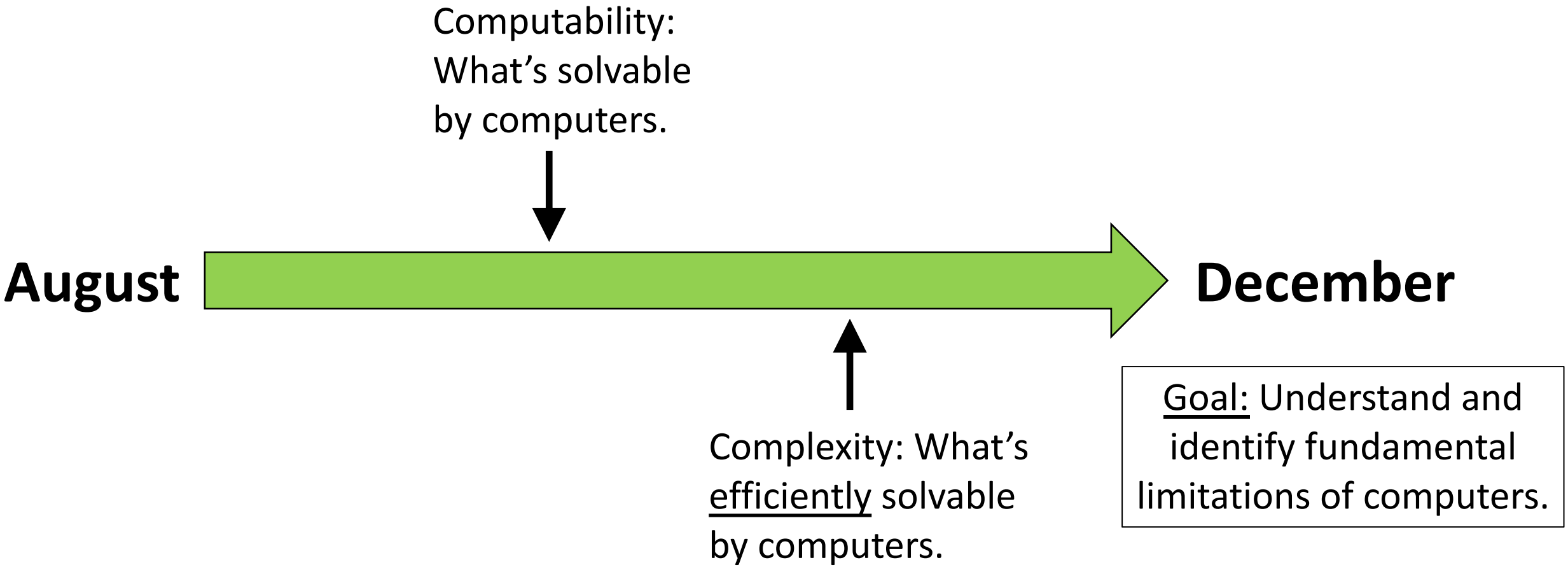
August  **December**

Goal: Understand and identify fundamental limitations of computers.

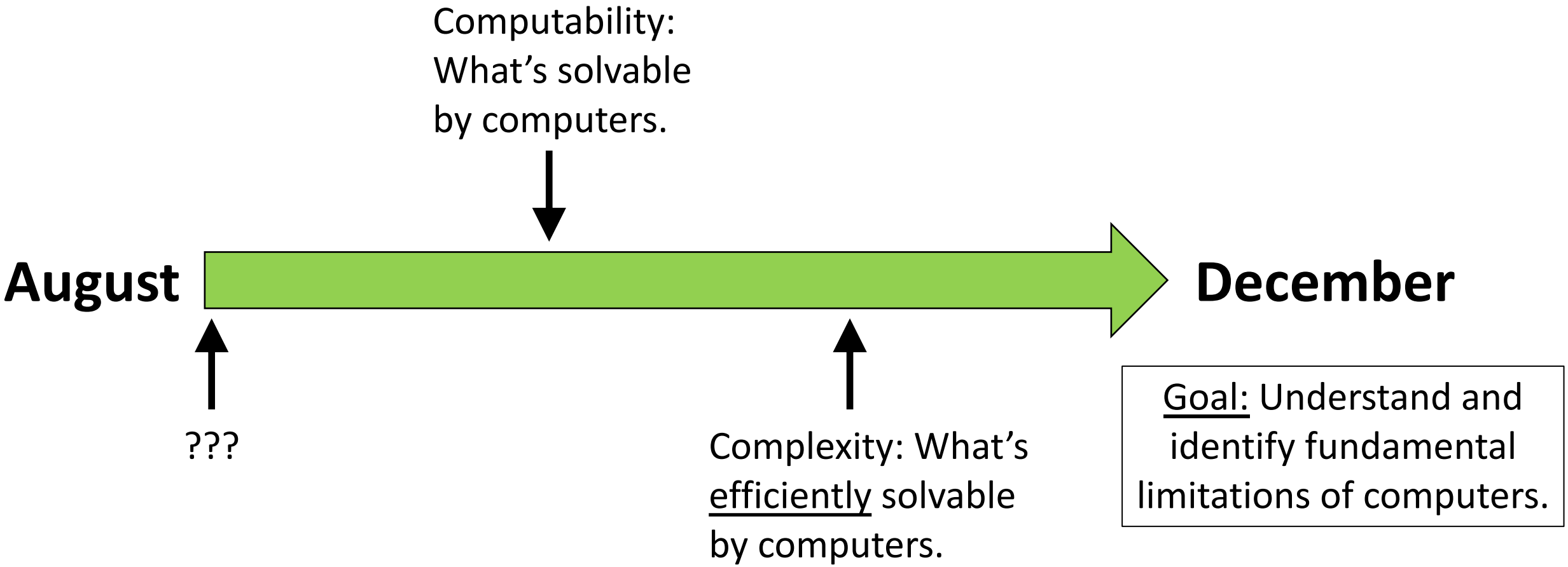
CSCI 338


August
December
Complexity: What's efficiently solvable by computers.
Goal: Understand and identify fundamental limitations of computers.

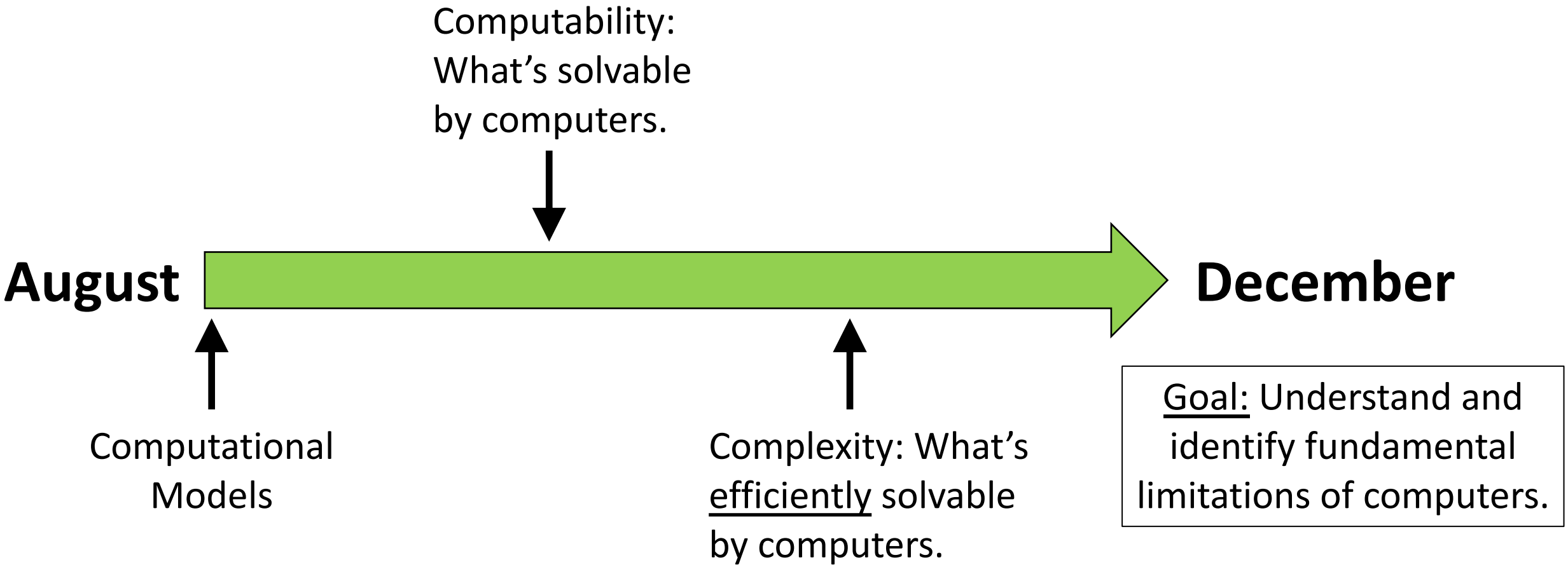
CSCI 338


Computability:
What's solvable
by computers.
August
December
Complexity: What's
efficiently solvable
by computers.
Goal: Understand and
identify fundamental
limitations of computers.

CSCI 338


Computability:
What's solvable
by computers.
August
December
???
Complexity: What's
efficiently solvable
by computers.
Goal: Understand and
identify fundamental
limitations of computers.

CSCI 338


Computability:
What's solvable
by computers.
August
December
Computational
Models
Complexity: What's
efficiently solvable
by computers.
Goal: Understand and
identify fundamental
limitations of computers.

Finite State Machine

Finite State Machine

Consider a car.

Finite State Machine

States:

??

Consider a car.

Finite State Machine

States:

- Moving
- Stopped

Consider a car.

Finite State Machine

States:

- Moving
- Stopped

Consider a car.

Actions:

??

Finite State Machine

States:

- Moving
- Stopped

Consider a car.

Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped

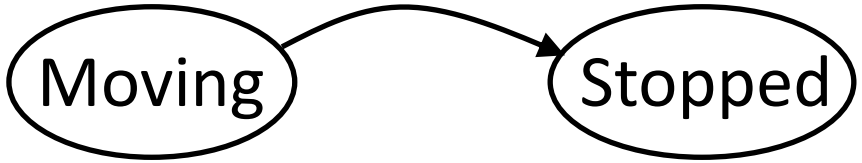
Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped



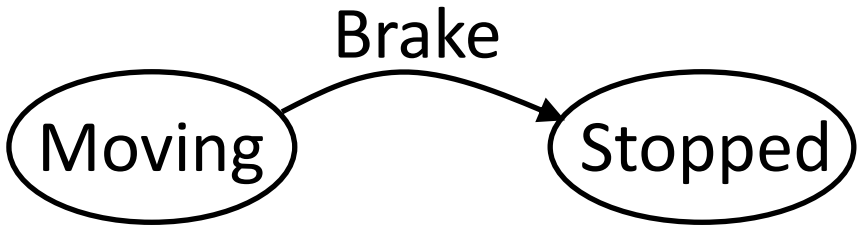
Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped



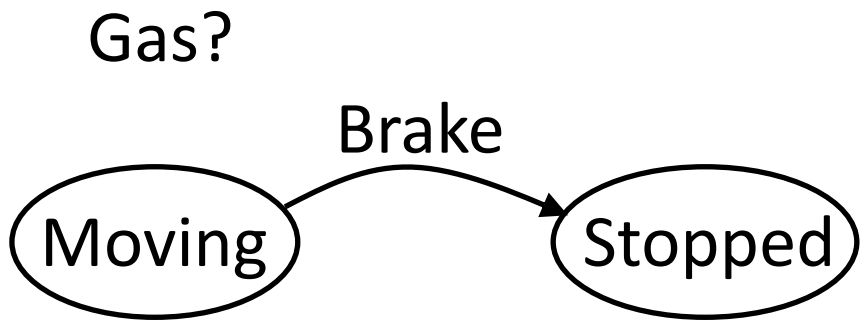
Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped



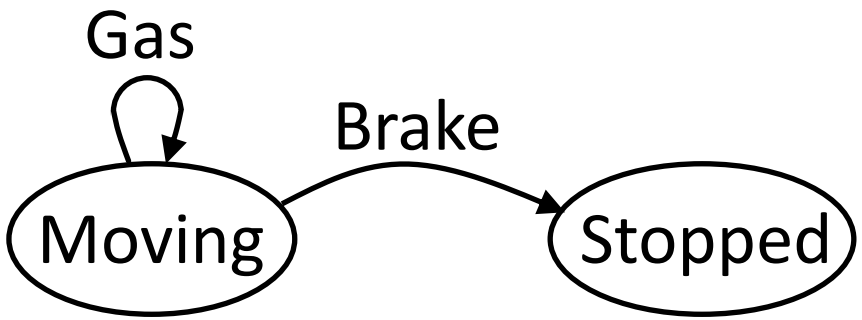
Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped



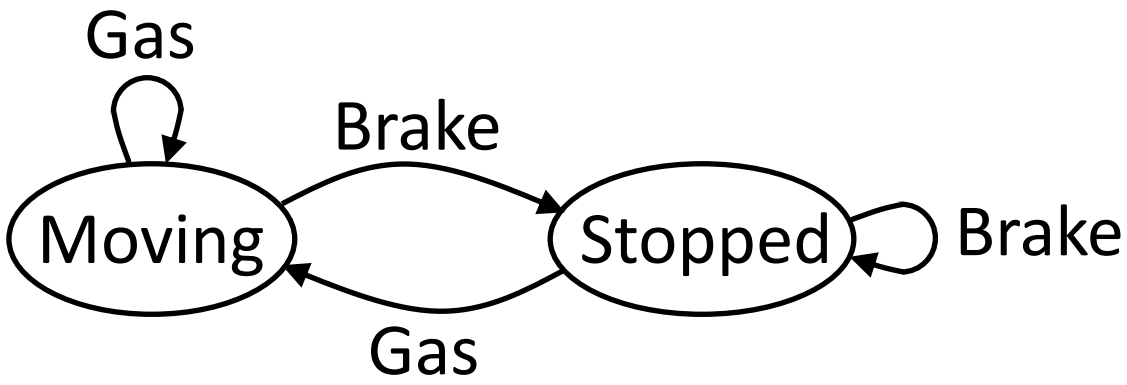
Actions:

- Gas
- Brake

Finite State Machine

States:

- Moving
- Stopped



Actions:

- Gas
- Brake

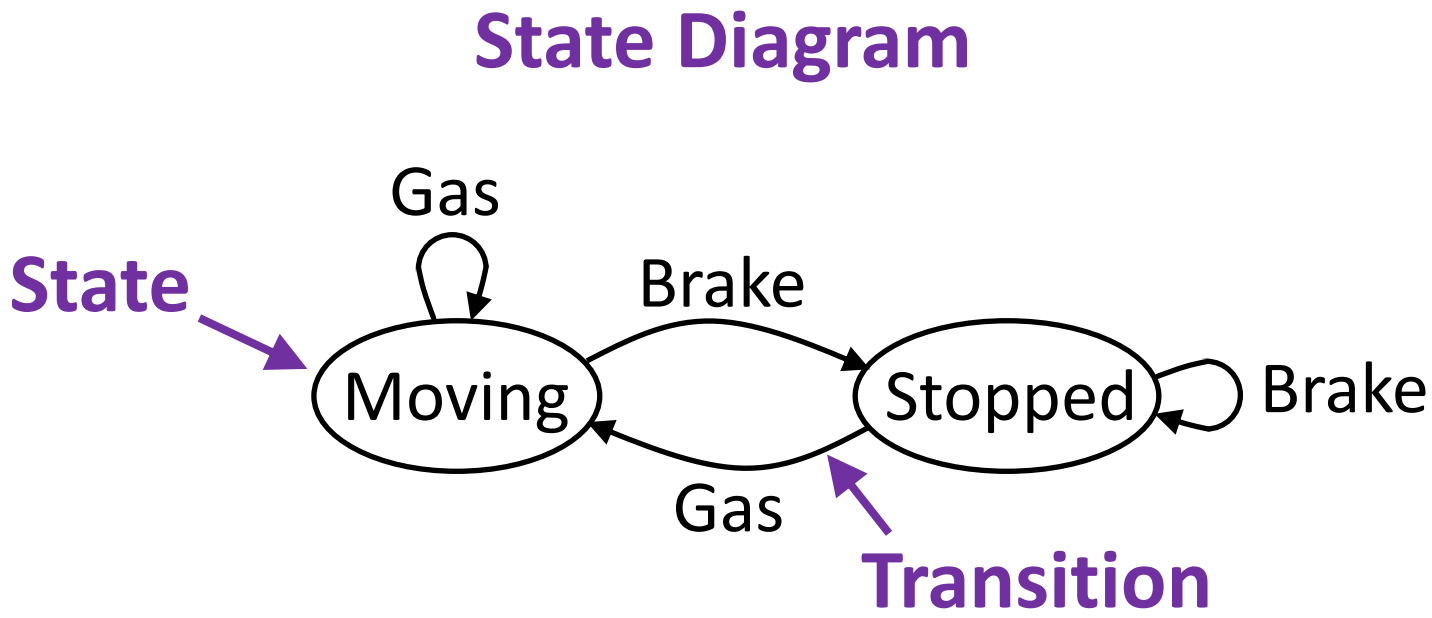
Finite State Machine

States:

- Moving
- Stopped

Actions:

- Gas
- Brake



Computational Question

Is the string $\omega = 01101001$ of the form:

$\{\omega: \omega$ contains an even number of 0's and even number of 1's$\}$?

Computational Question

Is the string $\omega = 01101001$ of the form:

$\{\omega: \omega$ contains an even number of 0's and even number of 1's$\}$?

Set of strings with a specific format (pattern).

Computational Question

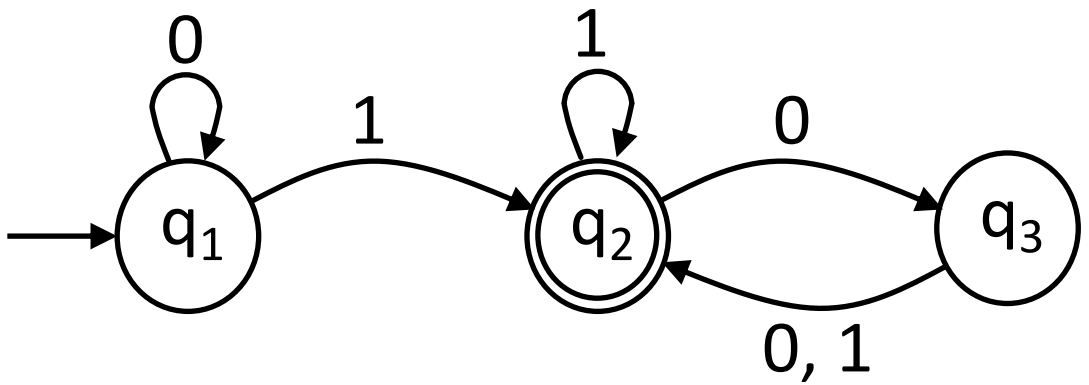
Is the string $\omega = 01101001$ of the form:

$\{\omega: \omega$ contains an even number of 0's and even number of 1's$\}$?

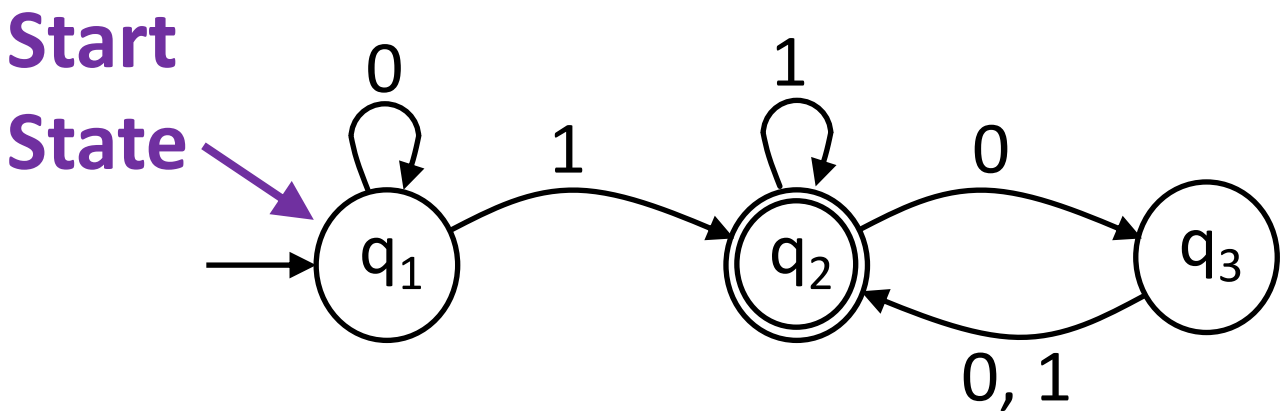
Set of strings with a specific format (pattern).

Deterministic Finite Automaton (DFA): An abstract model (machine) that determines (yes or no) if a string has a specific format.

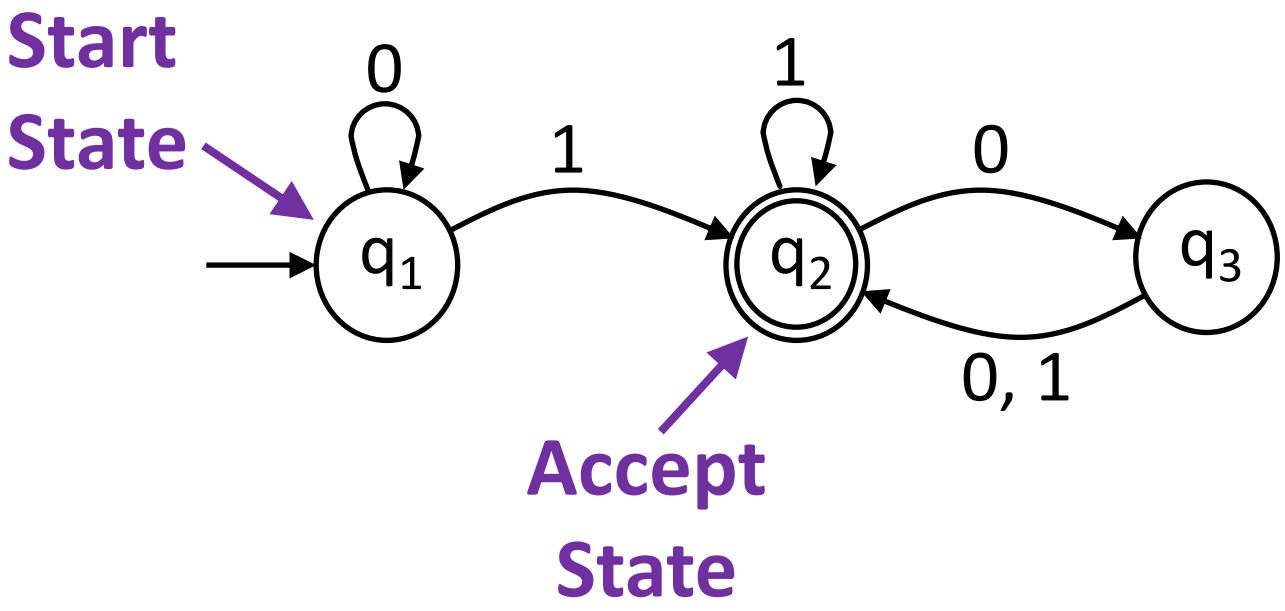
Deterministic Finite Automaton (DFA)



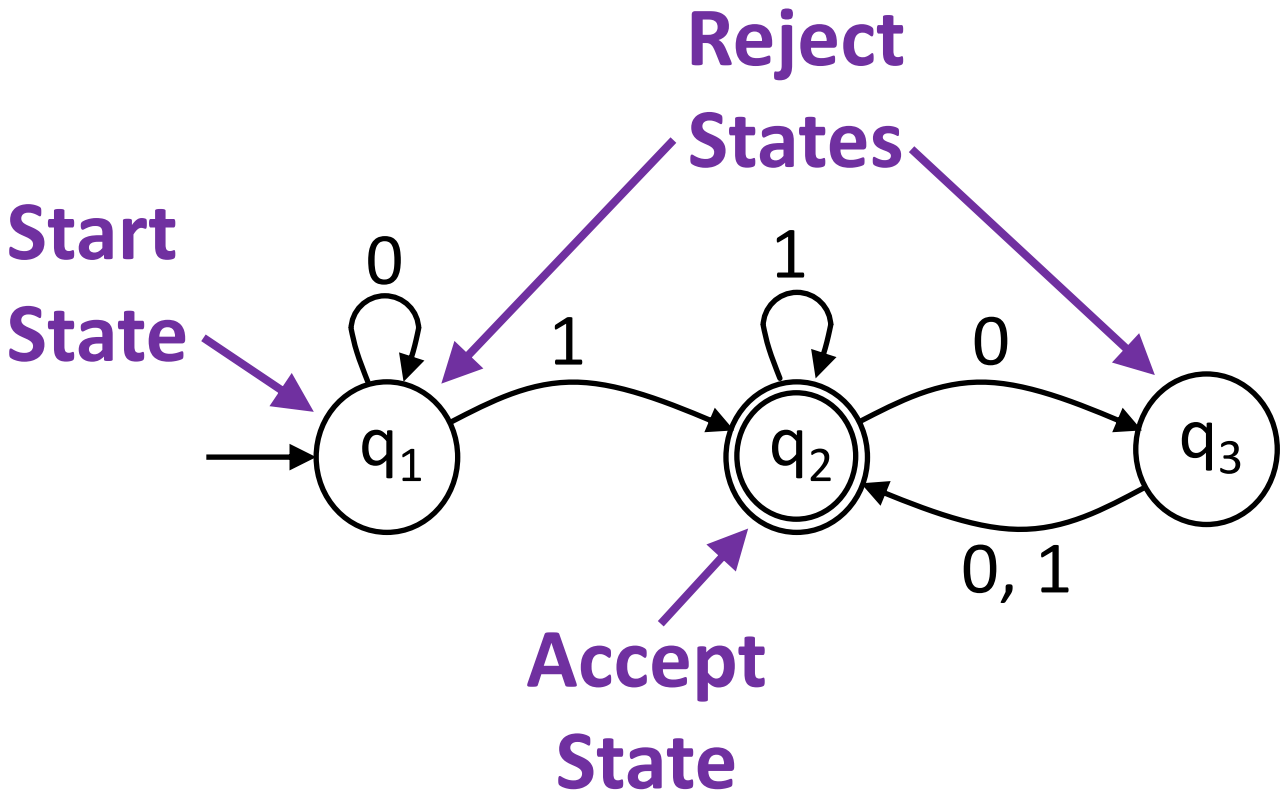
Deterministic Finite Automaton (DFA)



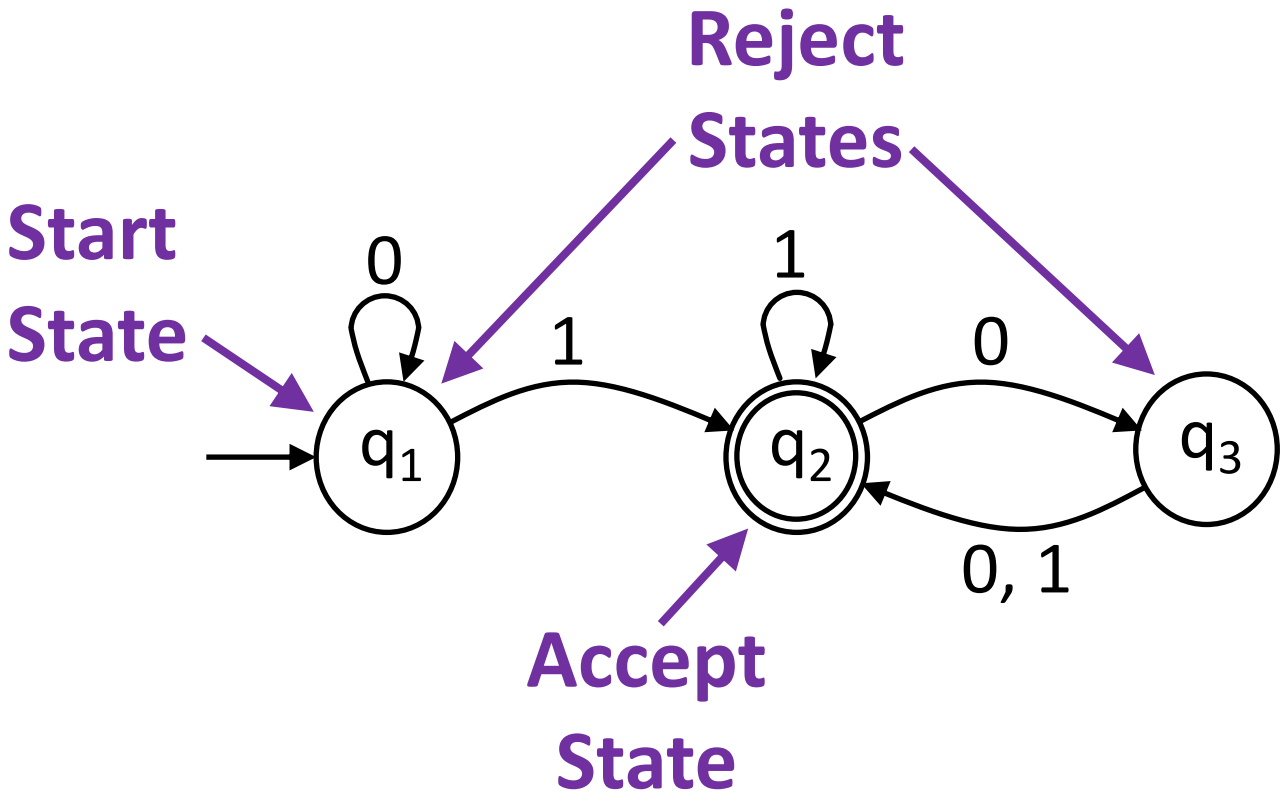
Deterministic Finite Automaton (DFA)



Deterministic Finite Automaton (DFA)

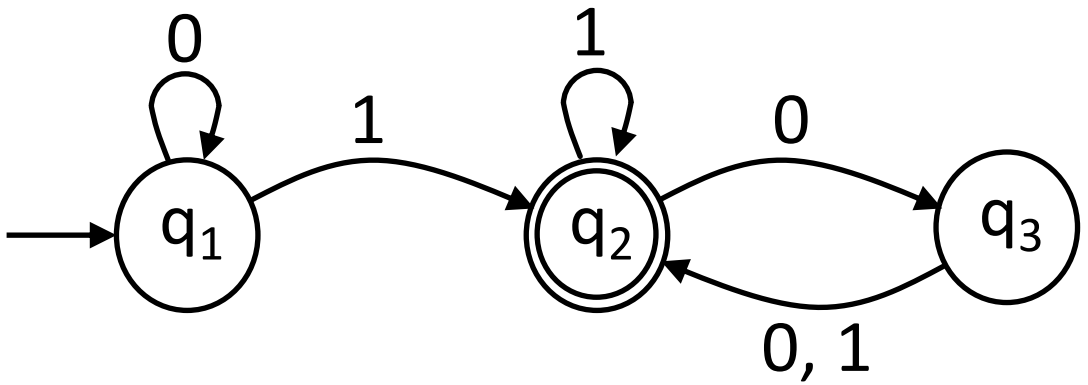


Deterministic Finite Automaton (DFA)



DFAs either accept or reject strings.

Deterministic Finite Automaton (DFA)



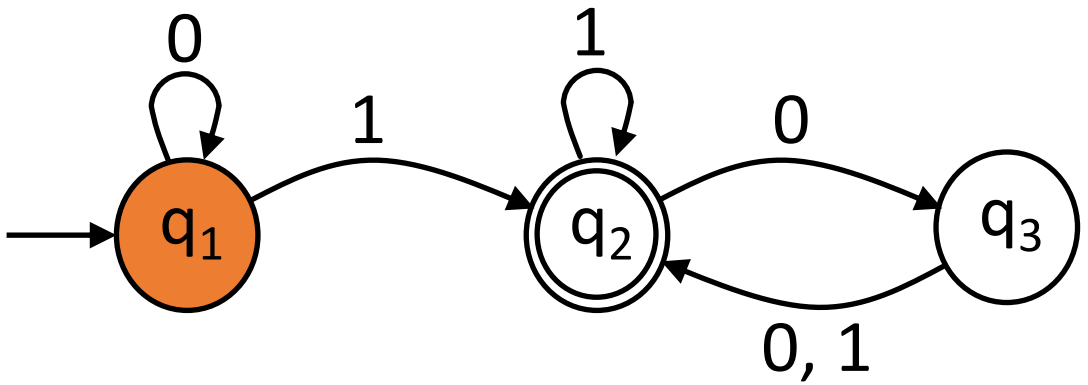
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.



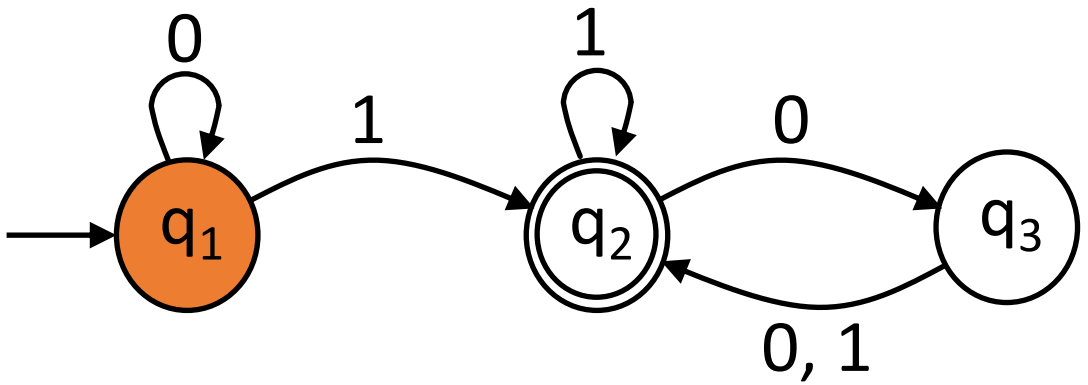
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.



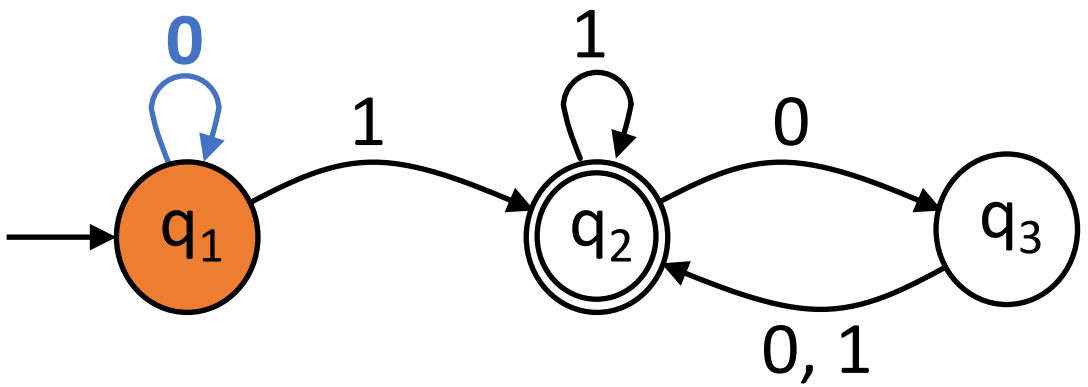
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.



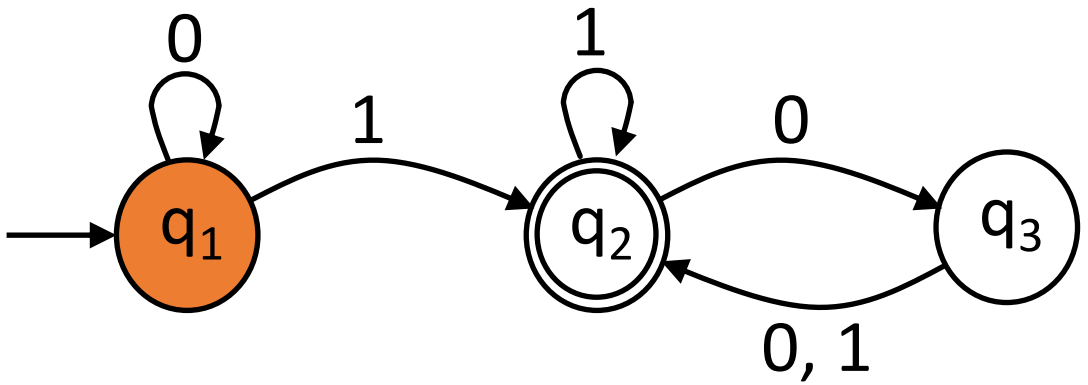
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.



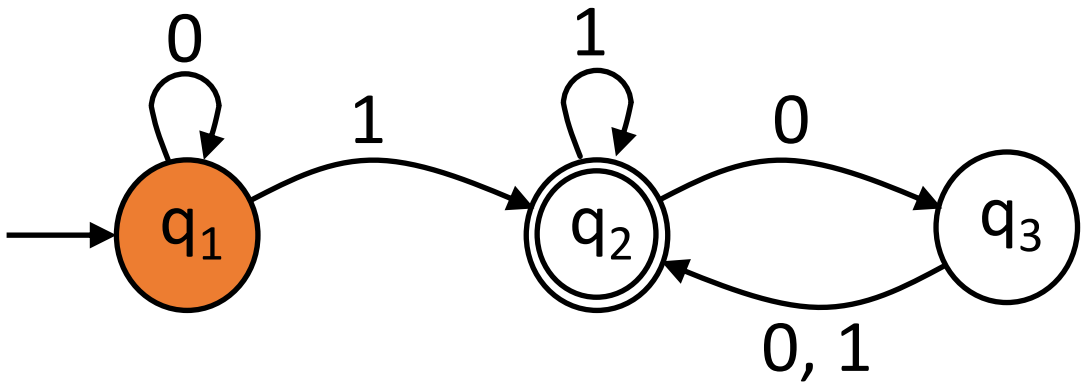
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



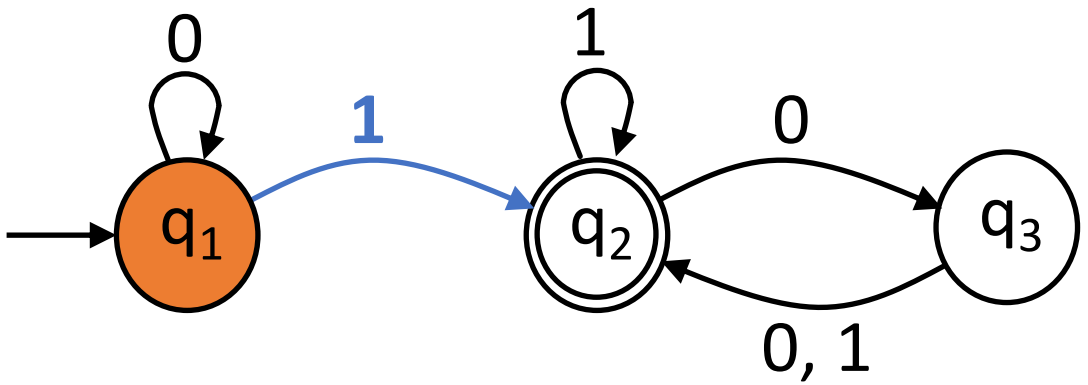
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. **Update state by following transition that corresponds to character.**
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



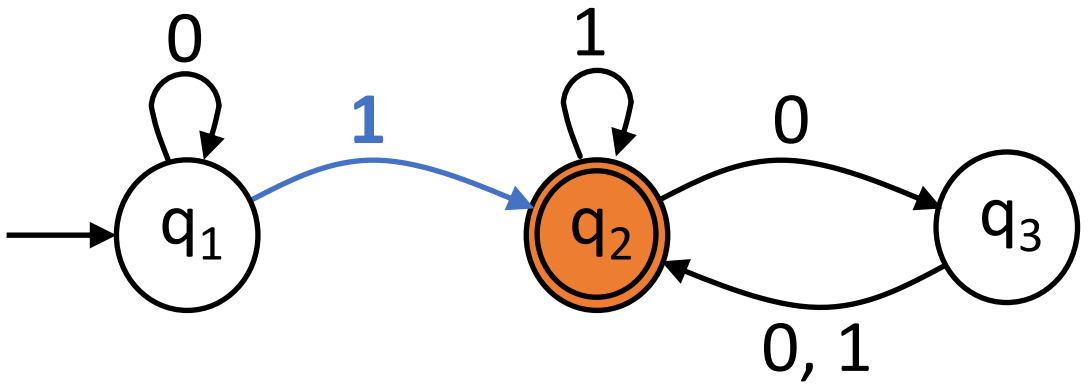
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. **Update state by following transition that corresponds to character.**
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



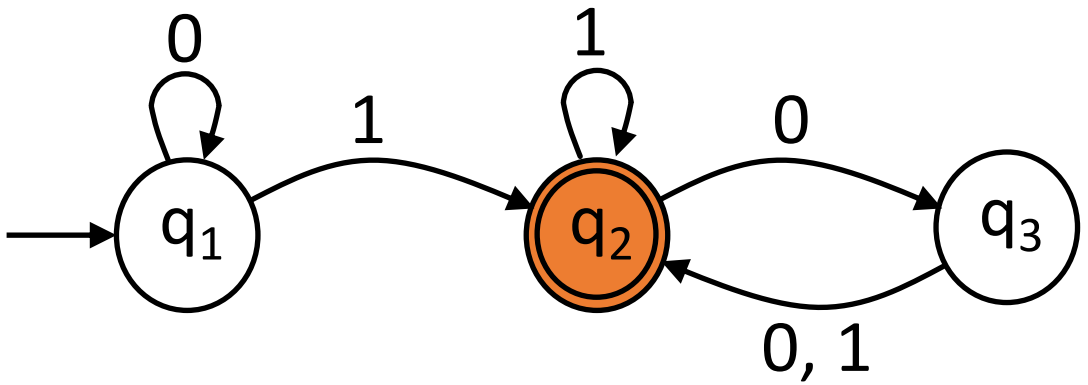
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. **Select next character in string.**
5. Repeat step 3 and 4 until no characters remain.



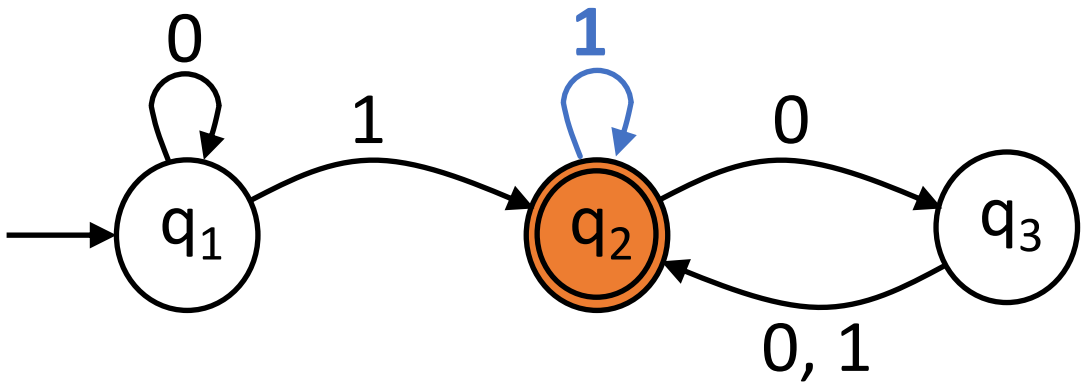
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. **Update state by following transition that corresponds to character.**
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



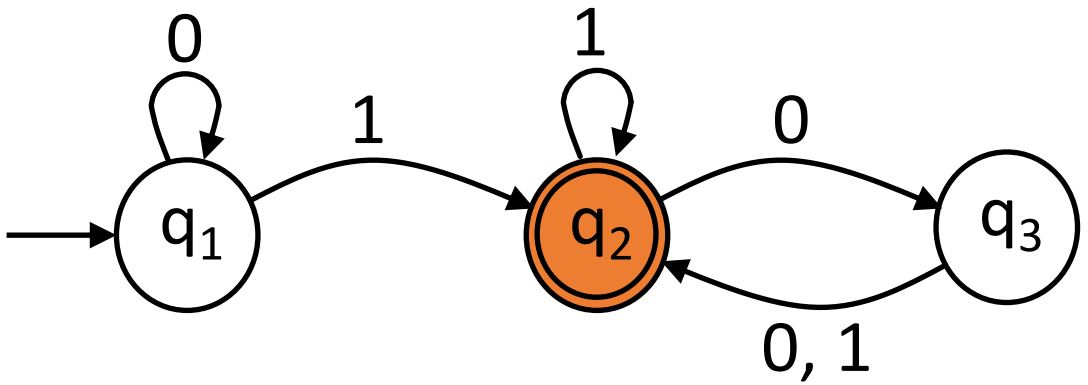
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. **Select next character in string.**
5. Repeat step 3 and 4 until no characters remain.



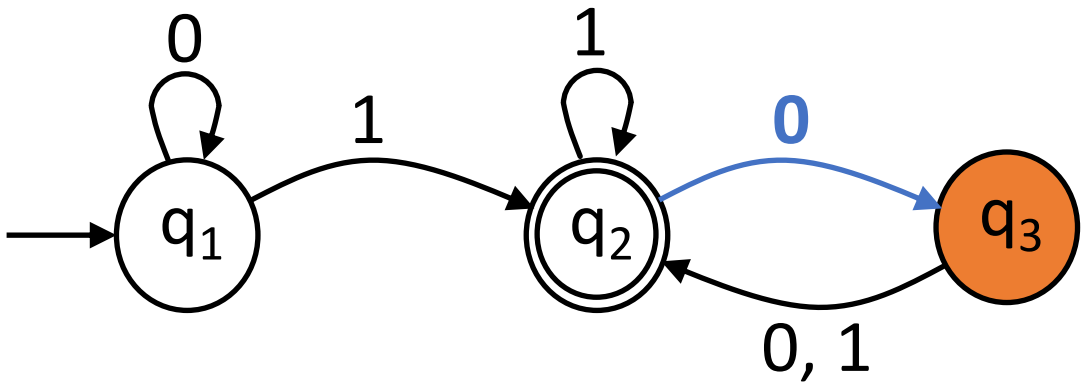
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. **Update state by following transition that corresponds to character.**
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



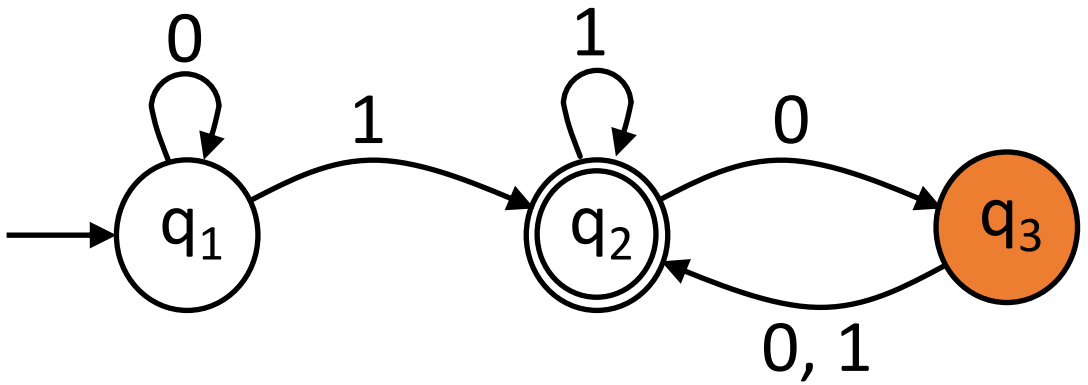
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. **Select next character in string.**
5. Repeat step 3 and 4 until no characters remain.



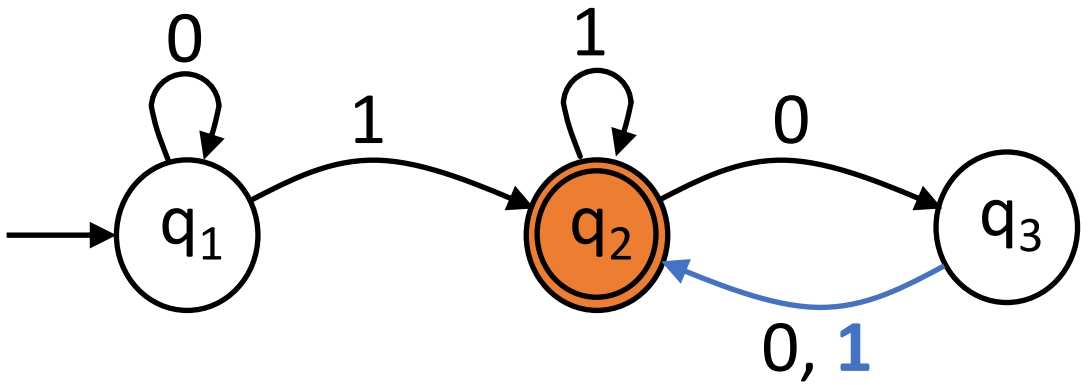
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. **Update state by following transition that corresponds to character.**
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.



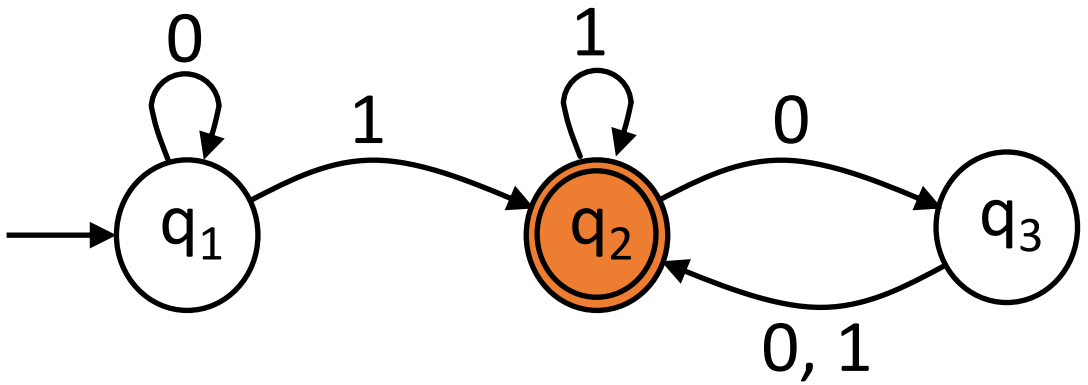
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until **no characters remain**.



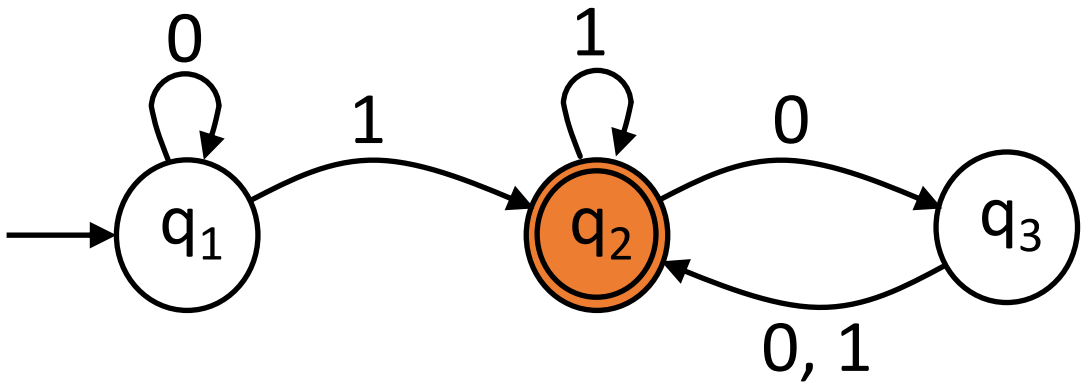
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



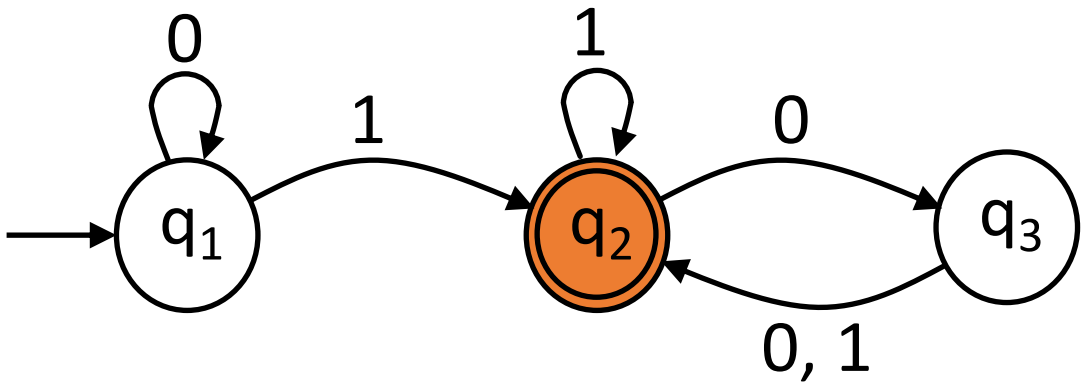
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



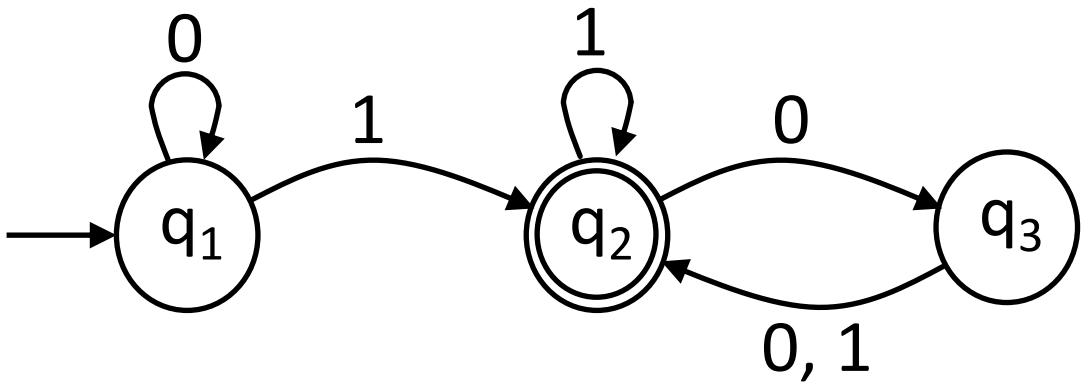
DFAs either accept or reject strings.

Given string $\omega = 01101$, does this DFA **accept** or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



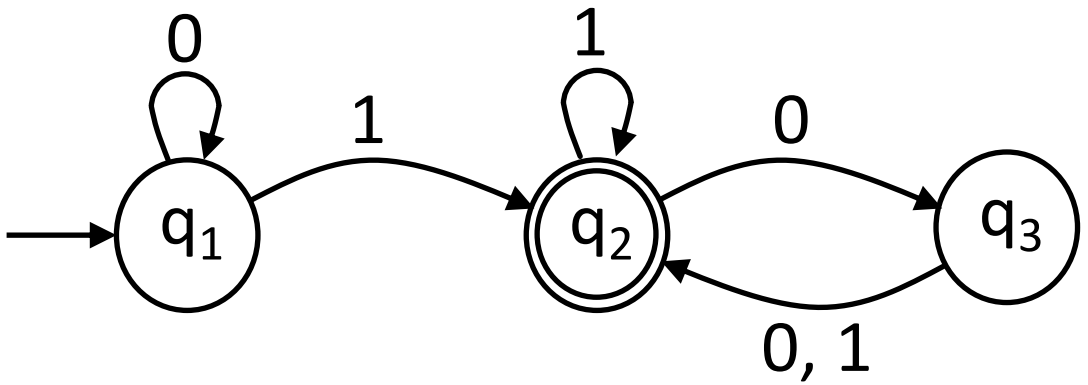
DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or reject?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



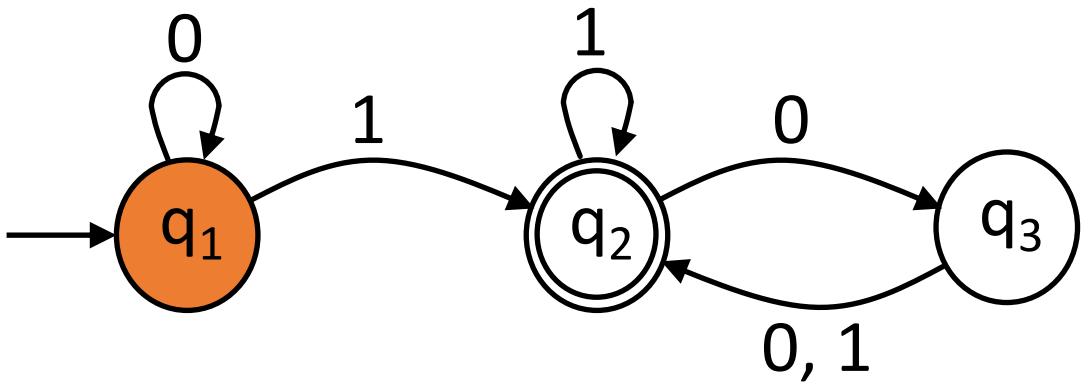
DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or **reject**?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



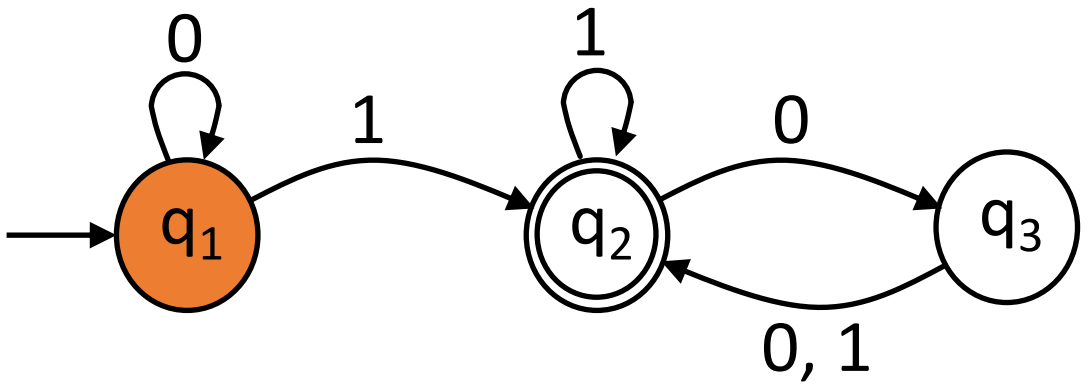
DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or **reject**?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



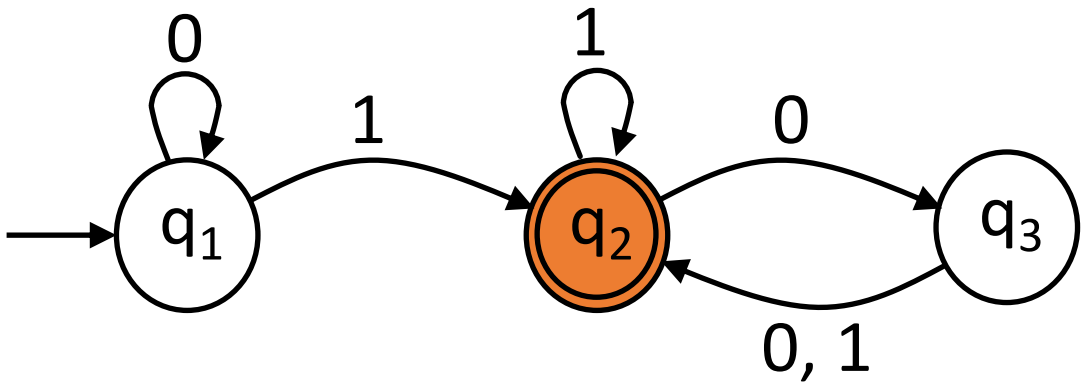
DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or **reject**?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.



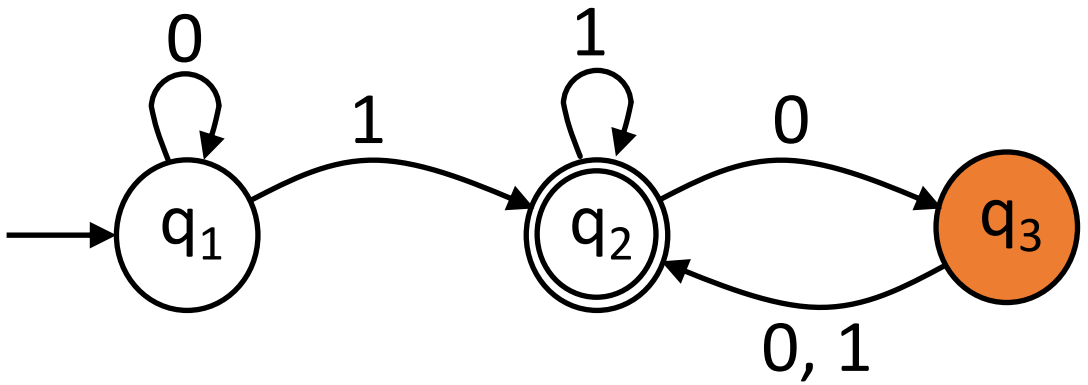
DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or **reject**?

Deterministic Finite Automaton (DFA)

DFA string processing:

1. Start at start state.
2. Select first character in string.
3. Update state by following transition that corresponds to character.
4. Select next character in string.
5. Repeat step 3 and 4 until no characters remain.
6. If final state is accept state, accept. Otherwise, reject.

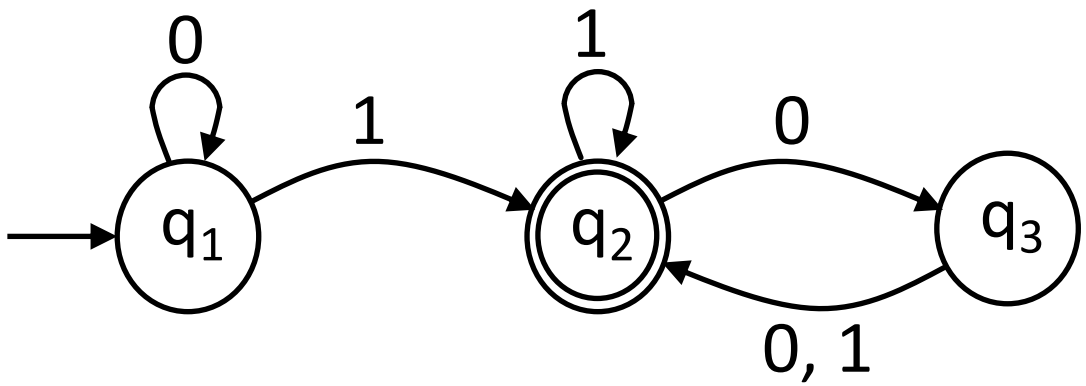


DFAs either accept or reject strings.

Given string $\omega = 010$, does this DFA accept or **reject**?

DFA Formal Definition

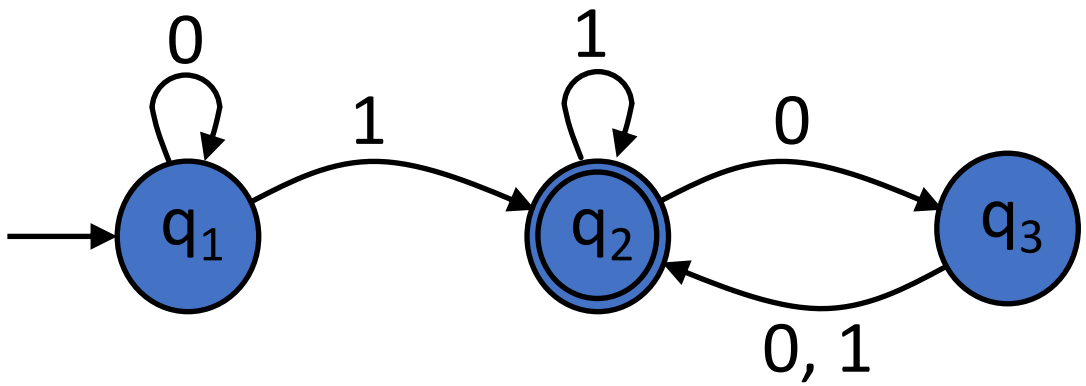
DFAs consist of:



DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.

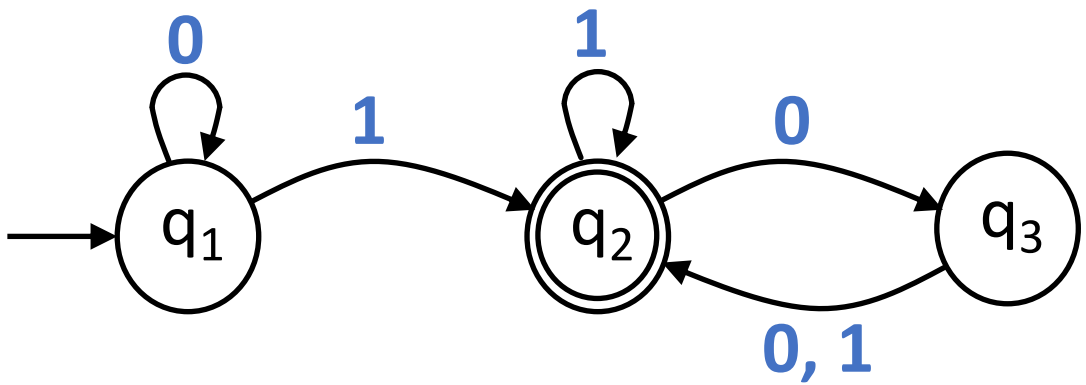


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, $\boldsymbol{\Sigma}$.

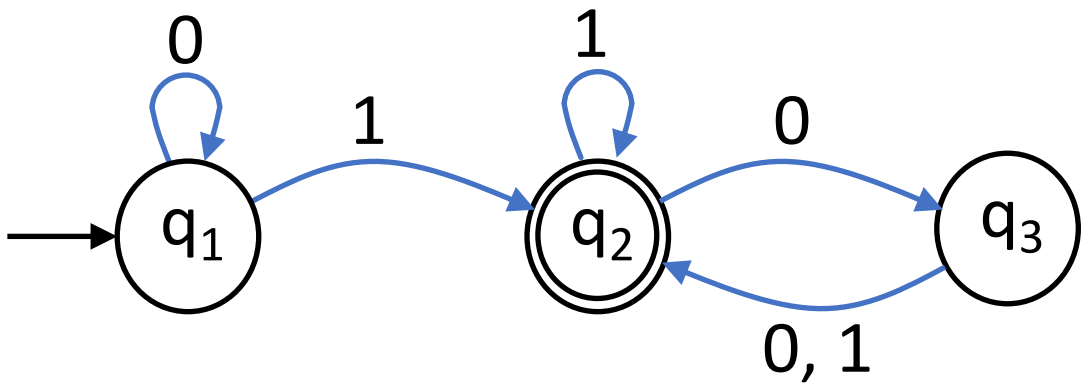
Σ consists of the transition characters (i.e. characters in the strings the DFA processes).



DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\boldsymbol{\delta: Q \times \Sigma \rightarrow Q}$.

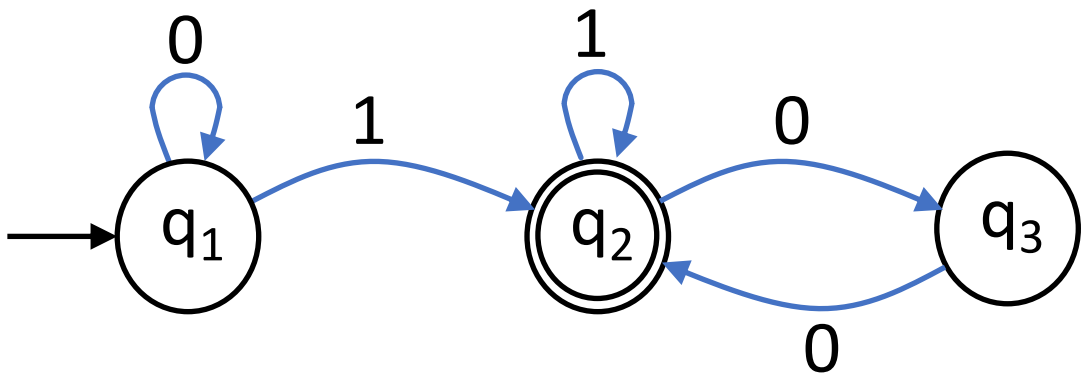


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\boldsymbol{\delta: Q \times \Sigma \rightarrow Q}$.

$\delta: Q \times \Sigma \rightarrow Q$ means that ***every*** state needs to handle ***every*** element of the alphabet.

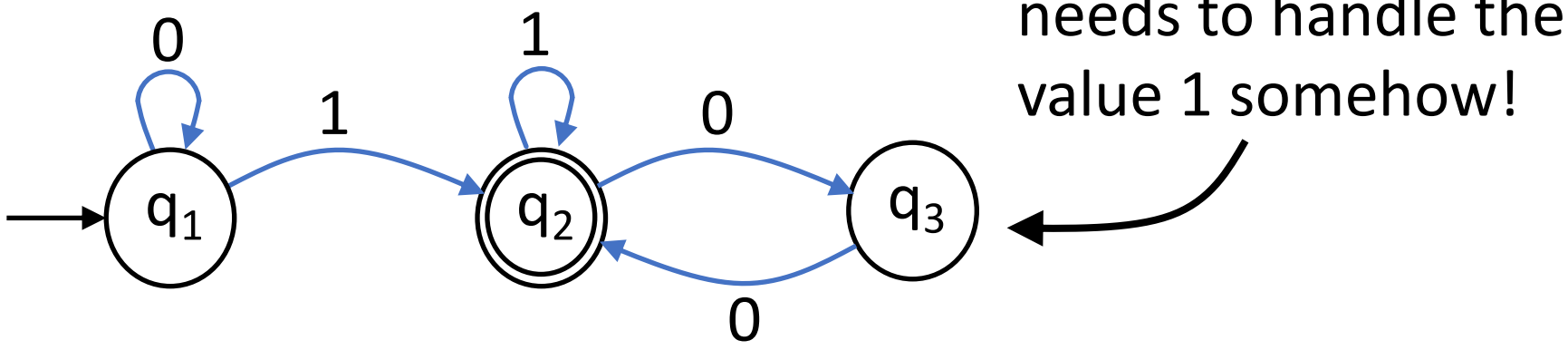


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\boldsymbol{\delta: Q \times \Sigma \rightarrow Q}$.

$\delta: Q \times \Sigma \rightarrow Q$ means that ***every*** state needs to handle ***every*** element of the alphabet.



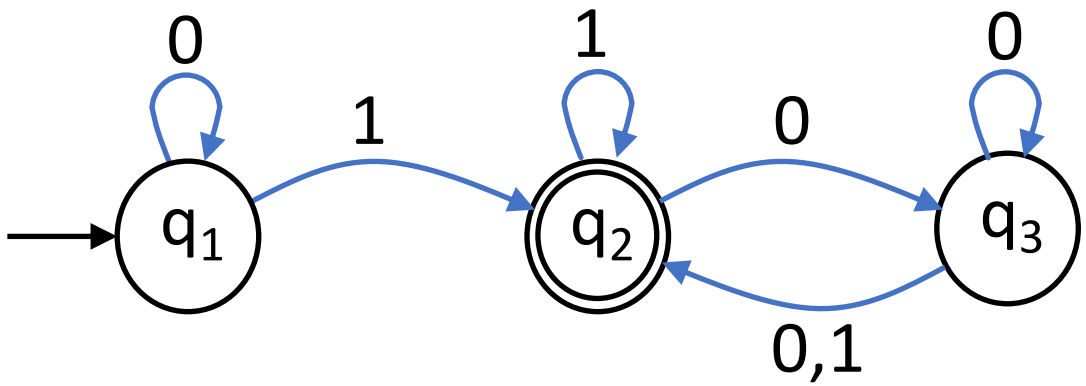
DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\boldsymbol{\delta: Q \times \Sigma \rightarrow Q}$.

$\delta: Q \times \Sigma \rightarrow Q$ means that ***every*** state needs to handle ***every*** element of the alphabet.

??

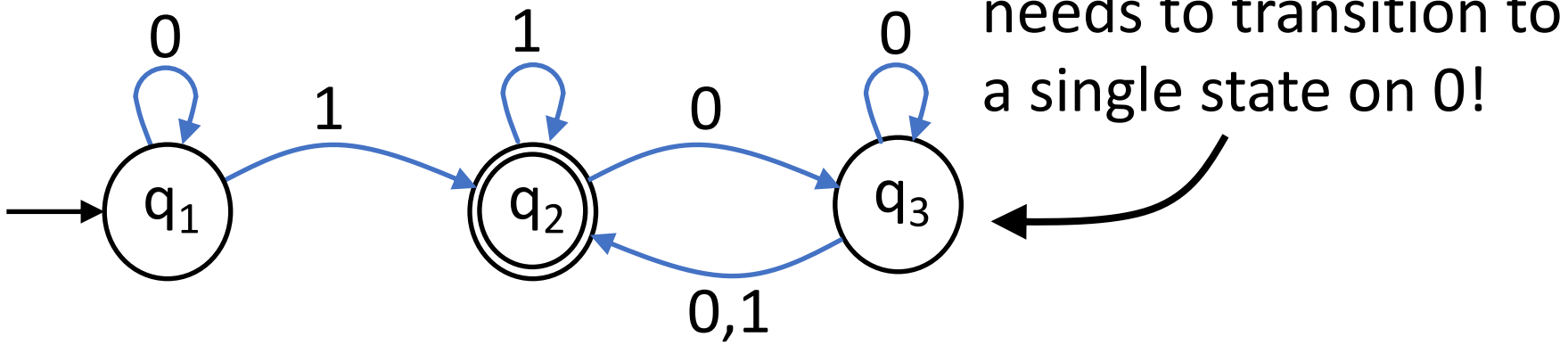


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\boldsymbol{\delta: Q \times \Sigma \rightarrow Q}$.

$\delta: Q \times \Sigma \rightarrow Q$ means that ***every*** state needs to handle ***every*** element of the alphabet.

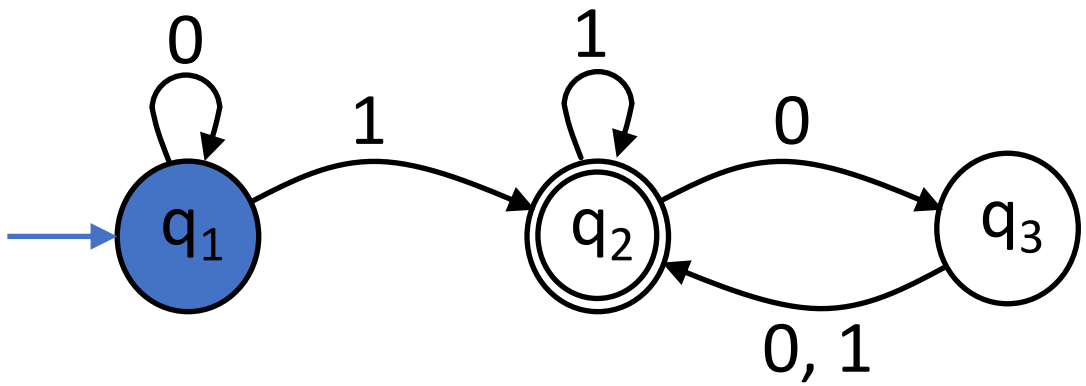


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times \Sigma \rightarrow Q$.
4. Start state, $\boldsymbol{q_0} \in Q$.

Exactly one start state needed.

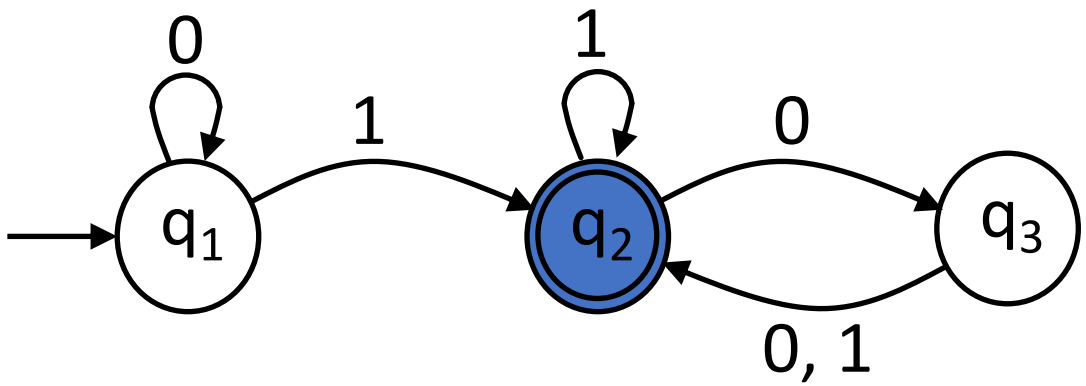


DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times \Sigma \rightarrow Q$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $\boldsymbol{F} \subseteq Q$.

F is allowed to equal Q or be empty.



DFA Formal Definition

DFAs consist of:

1. Finite set of states, Q.
2. Finite alphabet, Σ.
3. Transition function, $\delta: Q \times \Sigma \rightarrow Q$.
4. Start state, $q_0 \in Q$.
5. Set of accept states, $F \subseteq Q$.

$Q = \{q_1, q_2, q_3\}$

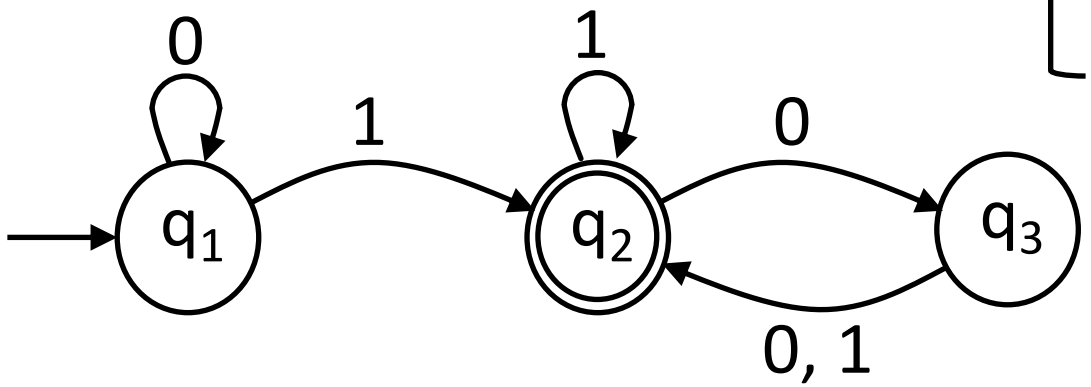
$\Sigma = \{0,1\}$

δ:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Start state = q_1

$F = \{q_2\}$



DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

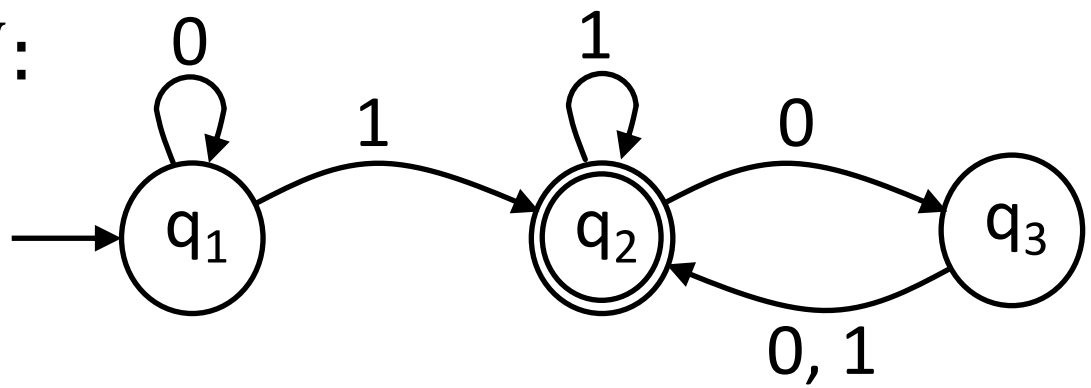
DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

M:



$L(M) =$?

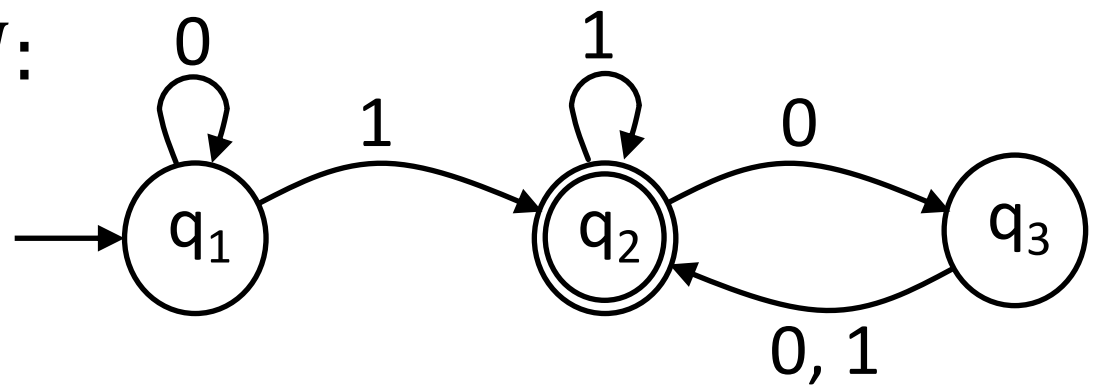
DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

M:



$L(M) = \{\omega: \omega$ contains at least one 1 and an even number of 0s following the final 1$\}$

DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

A language is called a regular language if some DFA recognizes it.

DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

A language is called a regular language if some DFA recognizes it.

How do you prove a language is regular?

DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

A language is called a regular language if some DFA recognizes it.

How do you prove a language is regular?

Make a DFA that recognizes it.

DFA Language

Definitions:

The set of all strings A that a DFA M accepts is called its language, $L(M) = A$.

M recognizes A.

A language is called a regular language if some DFA recognizes it.

How do you prove a language is regular?

Make a DFA that recognizes it.

Set of regular languages are “things we can do” with DFAs.

DFA Practice

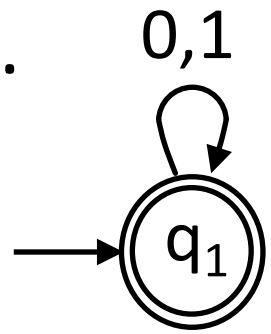
Prove that the following languages are regular:

1. Set of all strings over {0,1}.

DFA Practice

Prove that the following languages are regular:

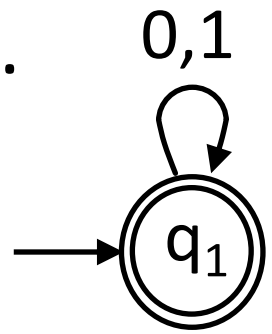
1. Set of all strings over {0,1}.



DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.

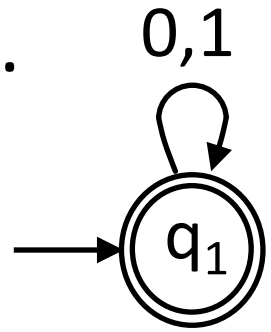


2. Set of all strings with an even number of 0s.

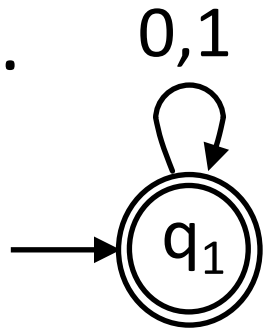
DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.



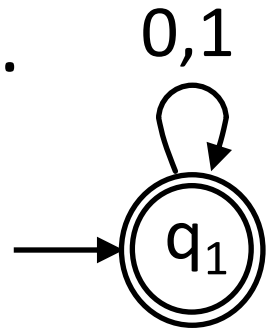
2. Set of all strings with an even number of 0s.



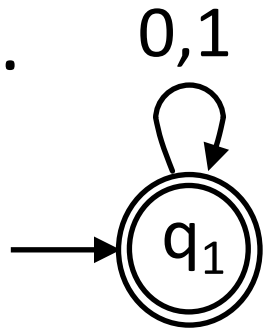
DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.



2. Set of all strings with an even number of 0s.

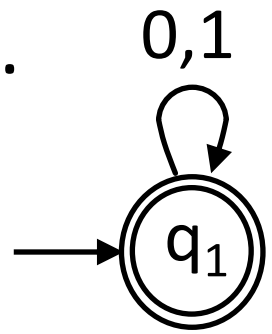


The set of **all** strings A that a DFA M accepts is called its language, $L(M) = A$.

DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.

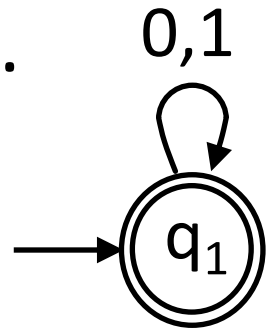


2. Set of all strings with an even number of 0s.

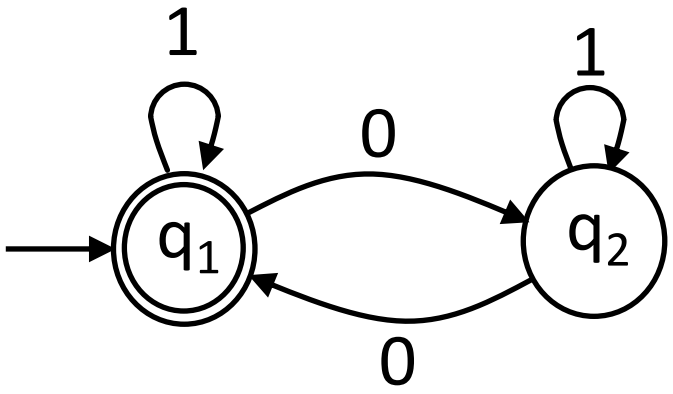
DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.



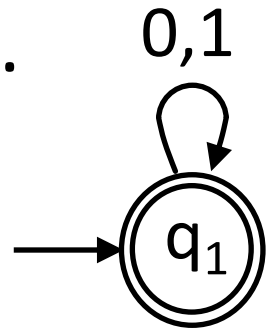
2. Set of all strings with an even number of 0s.



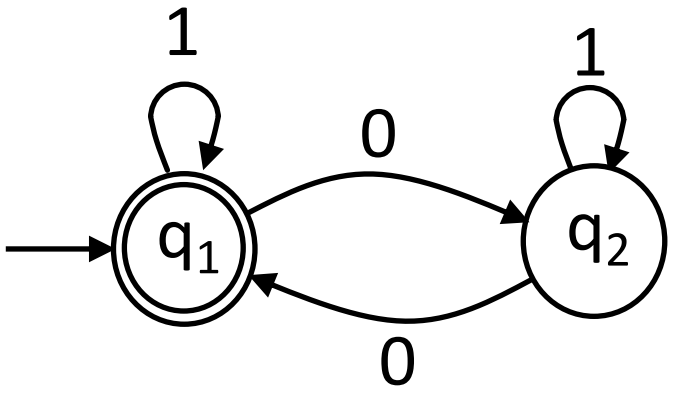
DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.



2. Set of all strings with an even number of 0s.

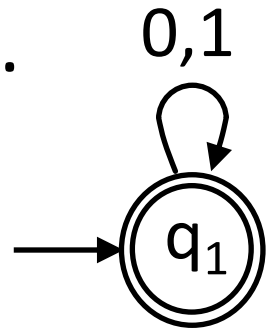


3. Set of all strings that contain the substring: 10.

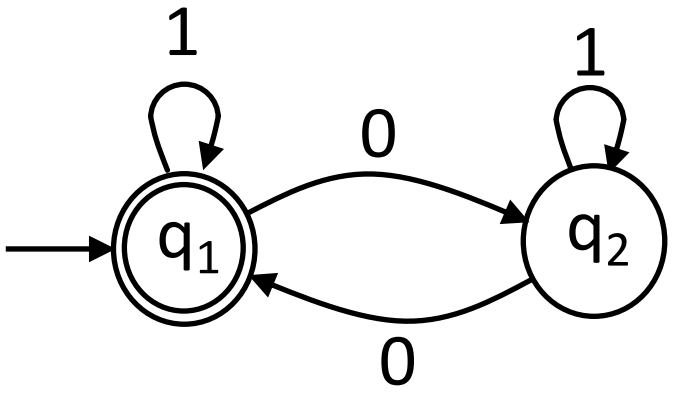
DFA Practice

Prove that the following languages are regular:

1. Set of all strings over {0,1}.



2. Set of all strings with an even number of 0s.



3. Set of all strings that contain the substring: 10.