Deterministic Finite Automata

Arjun Chandrasekhar

Deterministic Finite Automata

Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

- Several states

Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

- Several states
  - One *start state*

Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

- Several states
  - One *start state*
  - Some states are labelled as *accept states*

Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

- Several states
  - One *start state*
  - Some states are labelled as *accept states*
- Transitions (arrows) between states

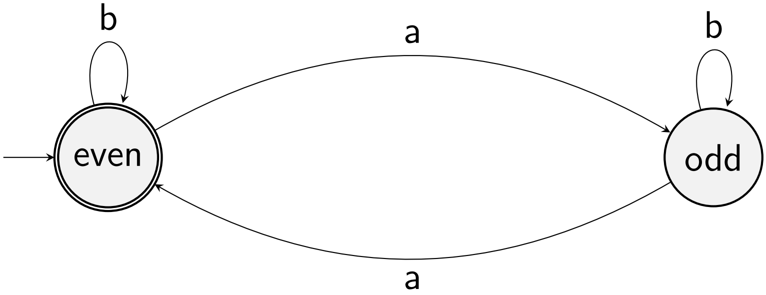
Deterministic Finite Automata

A model of computation that uses a constant amount of memory to recognize a language.

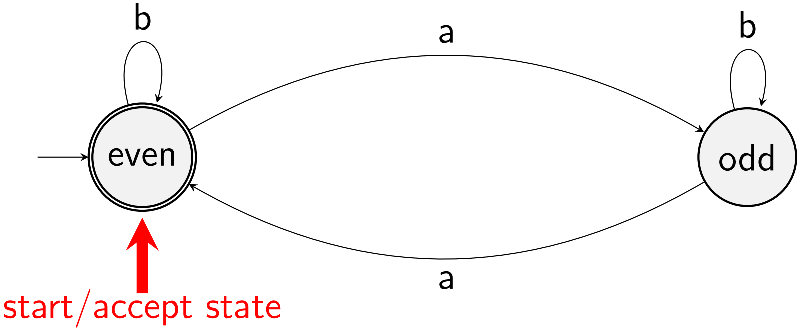
- Several states
  - One *start state*
  - Some states are labelled as *accept states*
- Transitions (arrows) between states
  - Each transition is labelled by a character from the alphabet

Deterministic Finite Automata

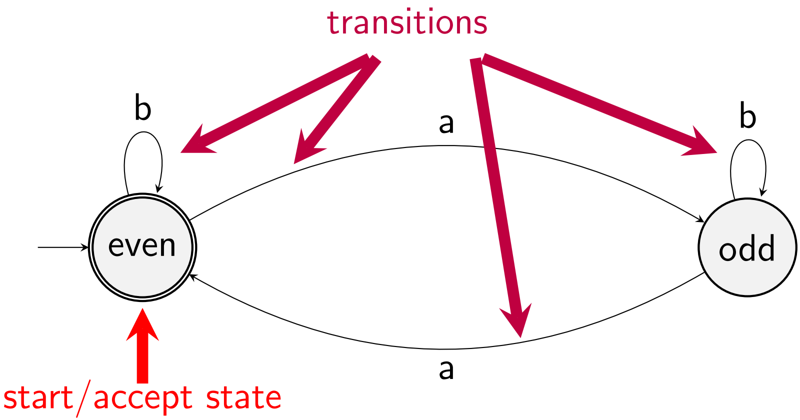
Deterministic Finite Automata



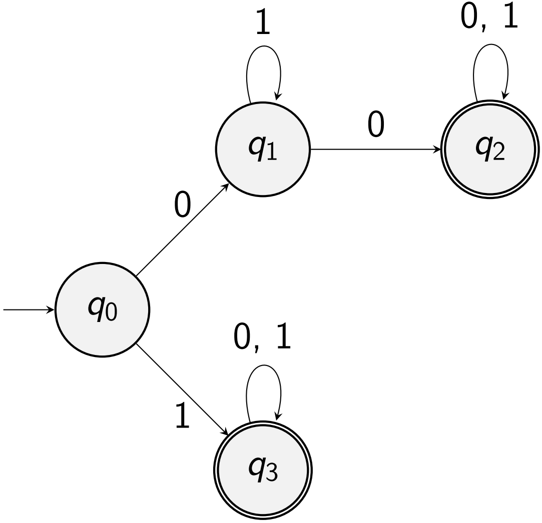
Deterministic Finite Automata



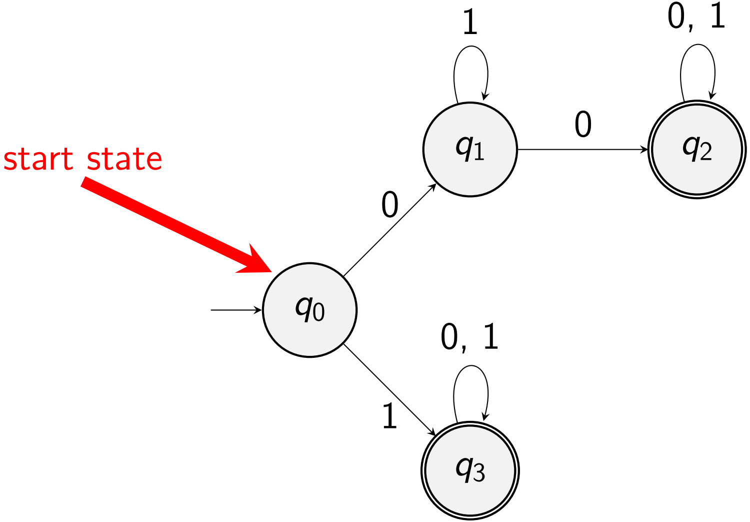
Deterministic Finite Automata



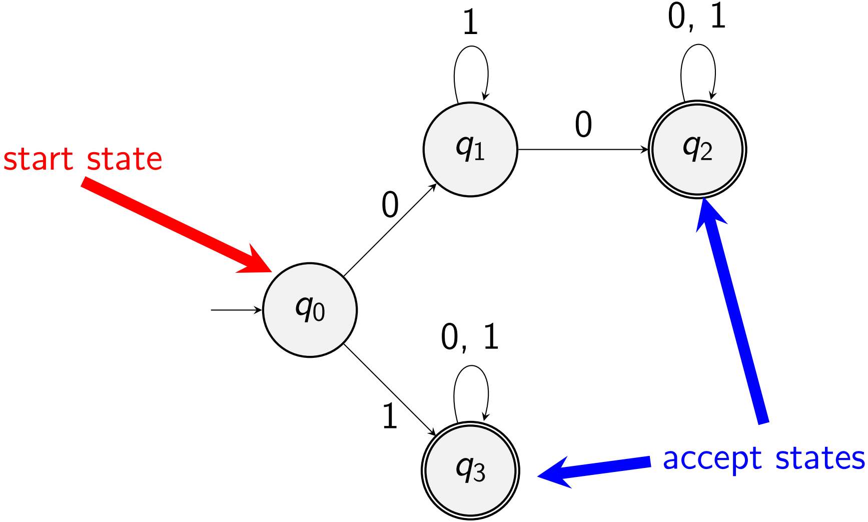
Deterministic Finite Automata



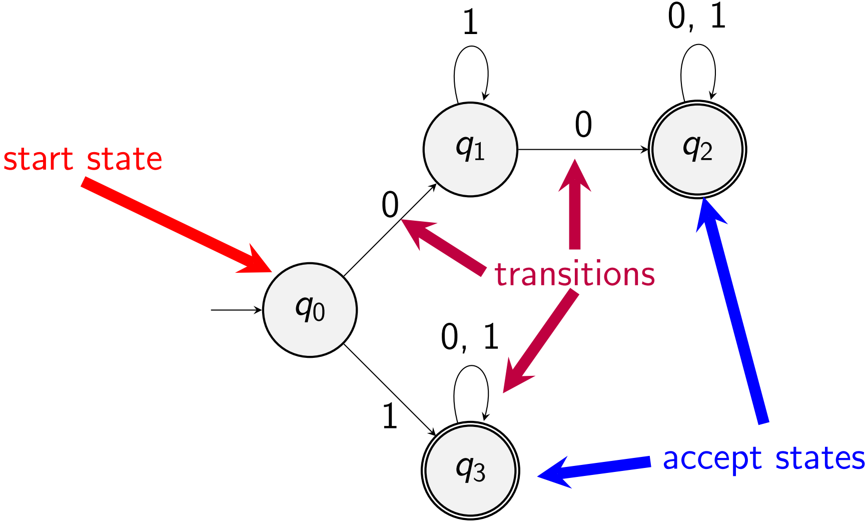
Deterministic Finite Automata



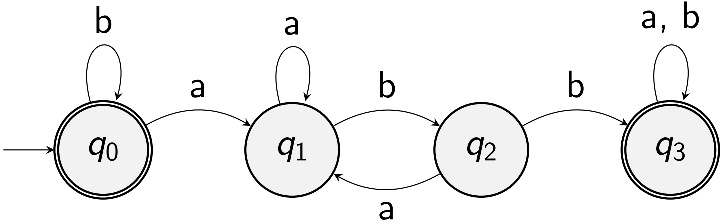
Deterministic Finite Automata



Deterministic Finite Automata



Deterministic finite automata



What is the start state of this DFA?

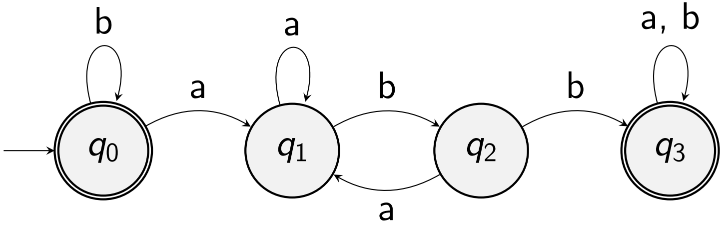
A. q_0

B. q_1

C. q_2

D. q_3

Deterministic finite automata



What is the start state of this DFA?

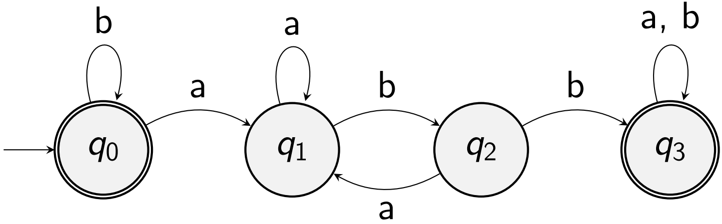
A. q_0 ✓

B. q_1

C. q_2

D. q_3

Deterministic finite automata



What are the accept states of this DFA?

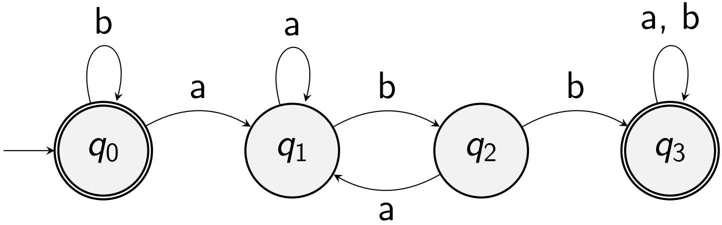
A. q_0

B. q_1

C. q_2

D. q_3

Deterministic finite automata



What are the accept states of this DFA?

A. q_0 ✓

B. q_1

C. q_2

D. q_3 ✓

Computation on a DFA

Computation on a DFA

- Start in the start state

Computation on a DFA

- Start in the start state
- Scan the symbols in the input one by one

Computation on a DFA

- Start in the start state
- Scan the symbols in the input one by one
- For each symbol σ scaned:

Computation on a DFA

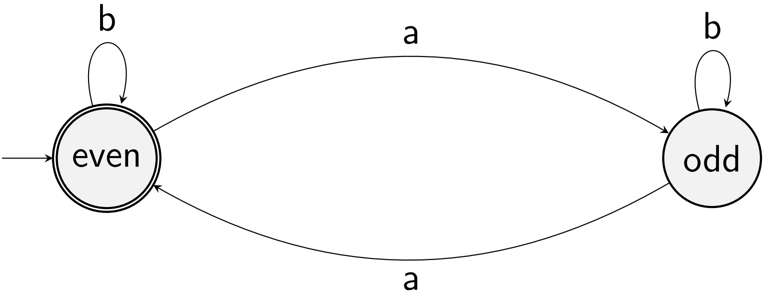
- Start in the start state
- Scan the symbols in the input one by one
- For each symbol σ scaned:
  - Go to the next state by following the arrow with the label σ

Computation on a DFA

- Start in the start state
- Scan the symbols in the input one by one
- For each symbol σ scaned:
  - Go to the next state by following the arrow with the label σ
- After scanning all of the input, if the DFA is in an accept state, the input is accepted. Otherwise the input is rejected

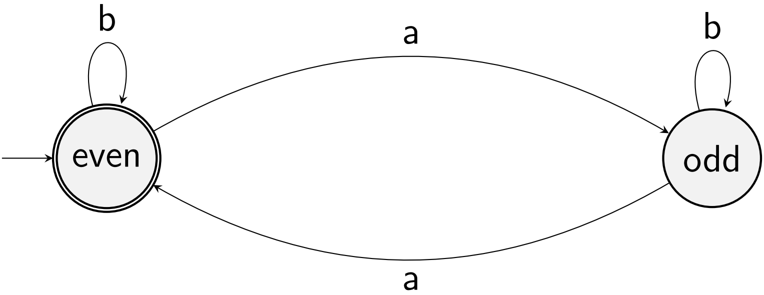
Deterministic finite automata

What does this DFA do on input aaaa?



Deterministic finite automata

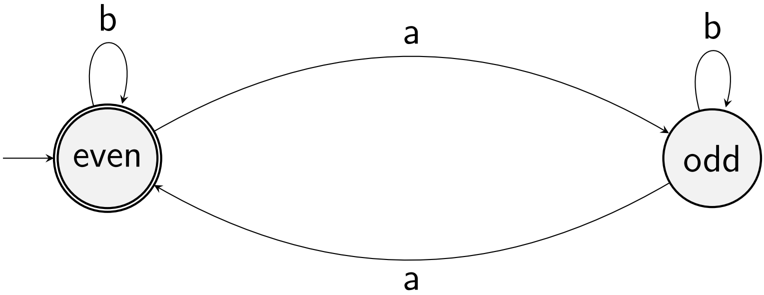
What does this DFA do on input aaaa?



$$\text{even} \xrightarrow{a} \text{odd} \xrightarrow{a} \text{even} \xrightarrow{a} \text{odd} \xrightarrow{a} \text{even} \rightarrow \text{ACCEPT}$$

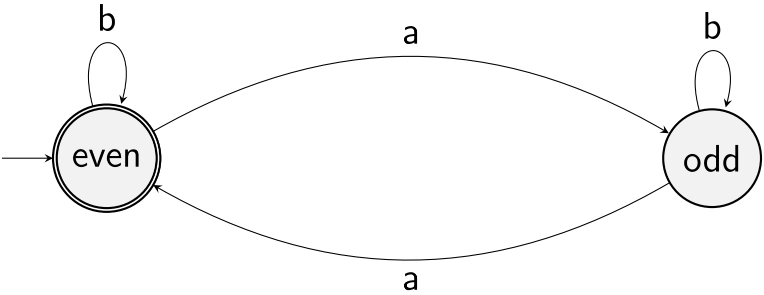
Deterministic finite automata

What does this DFA do on input b?



Deterministic finite automata

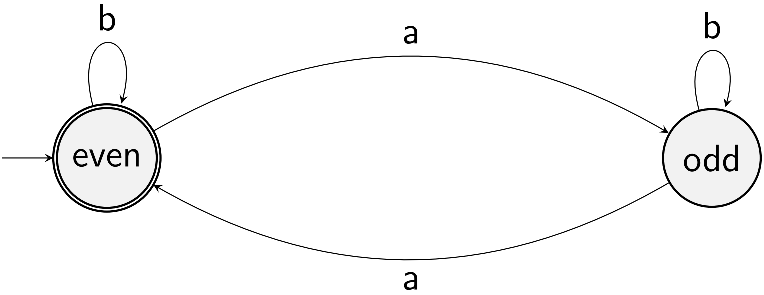
What does this DFA do on input b?



$$\text{even} \xrightarrow{b} \text{even} \rightarrow \text{ACCEPT}$$

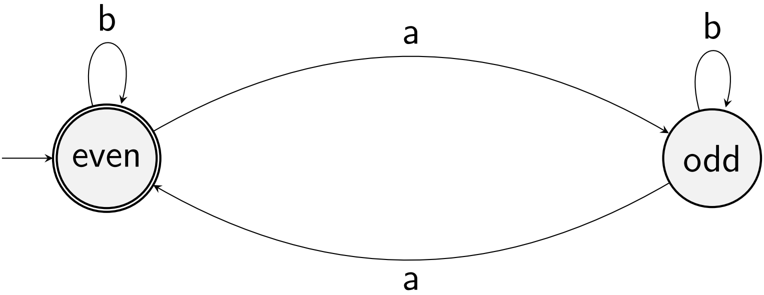
Deterministic finite automata

What does this DFA do on input abb?



Deterministic finite automata

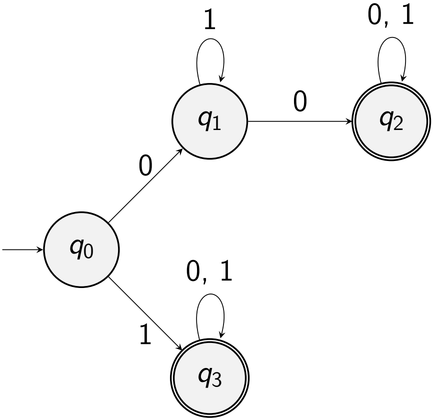
What does this DFA do on input abb?



$$\text{even} \xrightarrow{a} \text{odd} \xrightarrow{b} \text{odd} \xrightarrow{b} \text{odd} \rightarrow \textsf{REJECT}$$

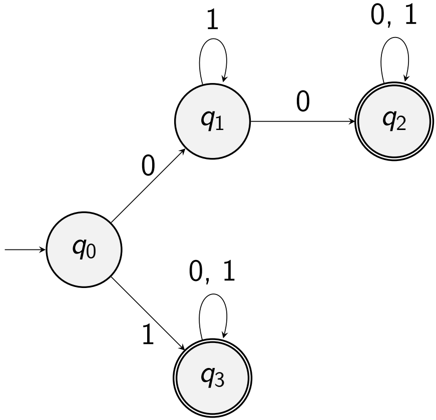
Deterministic Finite Automata

What does this DFA do on input 0110?



Deterministic Finite Automata

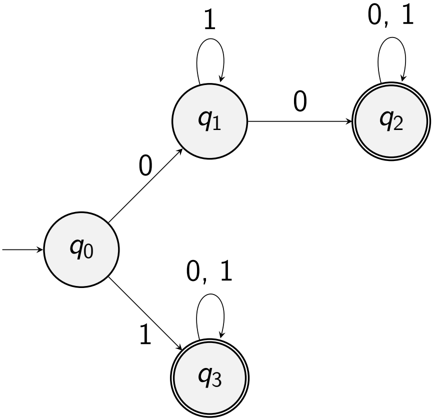
What does this DFA do on input 0110?



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \rightarrow \text{ACCEPT}$$

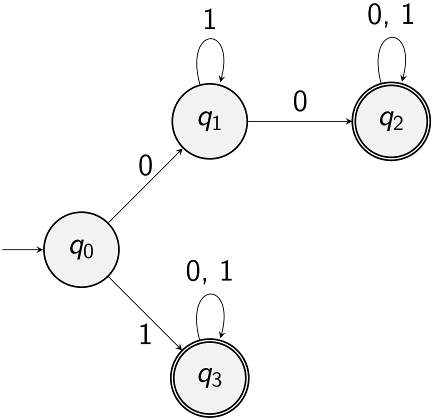
Deterministic Finite Automata

What does this DFA do on input 1011?



Deterministic Finite Automata

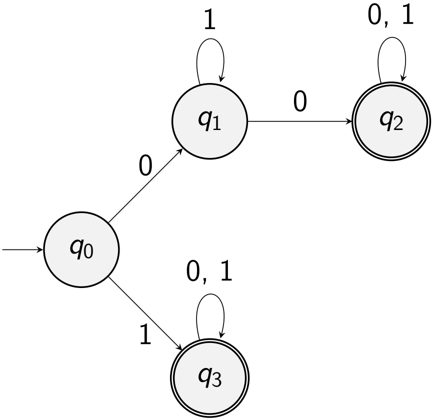
What does this DFA do on input 1011?



$$q_0 \xrightarrow{1} q_3 \xrightarrow{0} q_3 \xrightarrow{1} q_3 \xrightarrow{1} q_3 \rightarrow \text{ACCEPT}$$

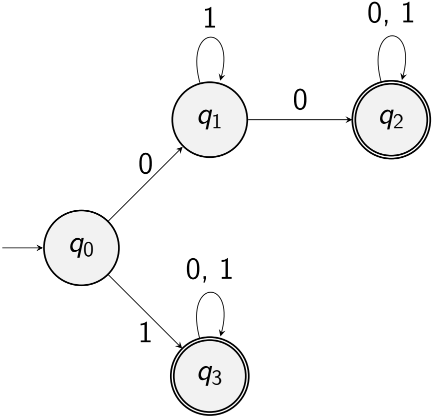
Deterministic Finite Automata

What does this DFA do on input 0111?



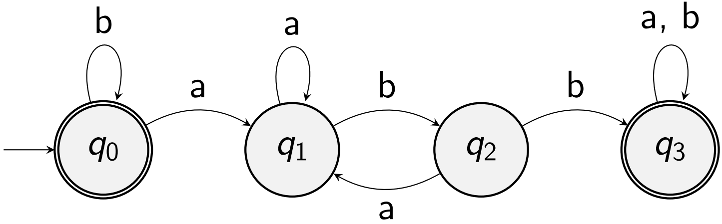
Deterministic Finite Automata

What does this DFA do on input 0111?



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \rightarrow \text{REJECT}$$

Deterministic Finite Automata



What strings are accepted by this DFA?

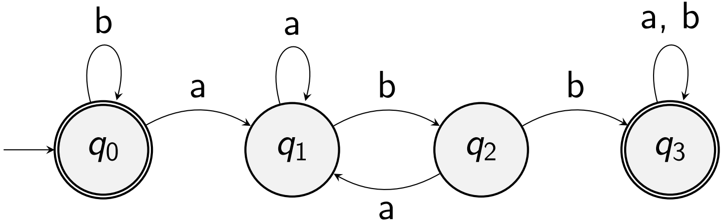
A. ϵ

B. *abbbb*

C. *bbbba*

D. $b^{100} = \underbrace{b \ldots b}_{100}$

Deterministic Finite Automata



What strings are accepted by this DFA?

A. ϵ ✓

B. *abbbb* ✓

C. *bbbba*

D. $b^{100} = \underbrace{b \ldots b}_{100}$ ✓

Formal definition of a DFA

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \to Q$ - the transition function

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \to Q$ - the transition function
  - The rules for how to move between states

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \to Q$ - the transition function
  - The rules for how to move between states
  - **Input:** Current state & next symbol

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \rightarrow Q$ - the transition function
  - The rules for how to move between states
  - **Input:** Current state & next symbol
  - **Output:** Next state

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \rightarrow Q$ - the transition function
  - The rules for how to move between states
  - **Input:** Current state & next symbol
  - **Output:** Next state
- q_s - the starting state

Formal definition of a DFA

Def: A **Deterministic Finite Automata (DFA)** is a 5-tuple $(Q, \Sigma, \delta, q_s, F)$

- Q - The set of states in the DFA
- Σ - The alphabet of characters that the DFA can read
- $\delta : Q \times \Sigma \to Q$ - the transition function
  - The rules for how to move between states
  - **Input:** Current state & next symbol
  - **Output:** Next state
- q_s - the starting state
- F - the set of accepting states

Formal defintion DFA computation

Formal defintion DFA computation

We say a DFA $D = (Q, \Sigma, q_s, \delta, F)$ accepts a string $w = \sigma_1\sigma_2 \ldots \sigma_n$ if there exists a sequence of states $q_0, q_1, \ldots, q_n$ such that:

Formal defintion DFA computation

We say a DFA $D = (Q, \Sigma, q_s, \delta, F)$ accepts a string $w = \sigma_1\sigma_2\ldots\sigma_n$ if there exists a sequence of states $q_0, q_1, \ldots, q_n$ such that:

- $q_0 = q_s$ (start in the start state)

Formal defintion DFA computation

We say a DFA $D = (Q, \Sigma, q_s, \delta, F)$ accepts a string $w = \sigma_1\sigma_2\ldots\sigma_n$ if there exists a sequence of states $q_0, q_1, \ldots, q_n$ such that:

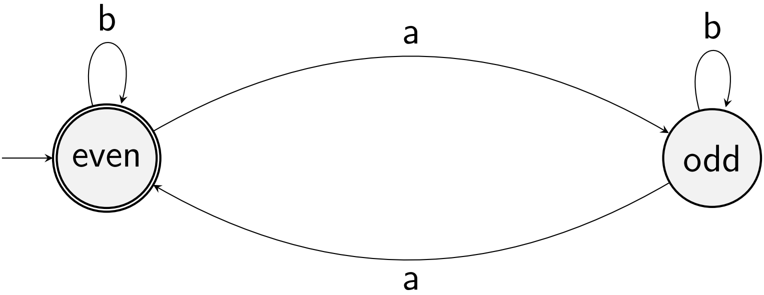
- $q_0 = q_s$ (start in the start state)
- $q_i = \delta(q_{i-1}, \sigma_i)$ for all i (all transitions are valid)

Formal defintion DFA computation

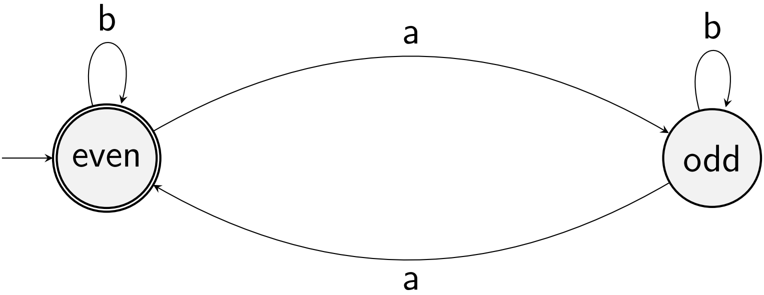
We say a DFA $D = (Q, \Sigma, q_s, \delta, F)$ accepts a string $w = \sigma_1\sigma_2 \ldots \sigma_n$ if there exists a sequence of states $q_0, q_1, \ldots, q_n$ such that:

- $q_0 = q_s$ (start in the start state)
- $q_i = \delta(q_{i-1}, \sigma_i)$ for all i (all transitions are valid)
- $q_n \in F$ (finish in an accept state)

Formal definition example

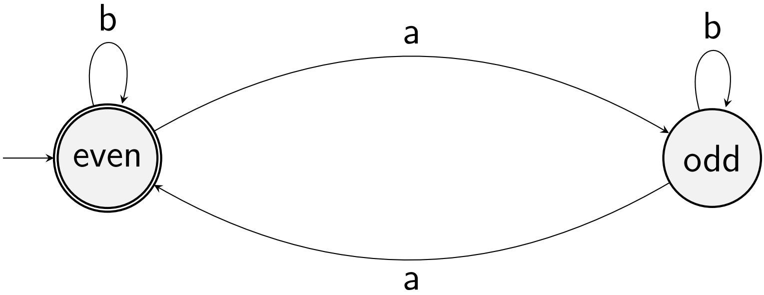


Formal definition example



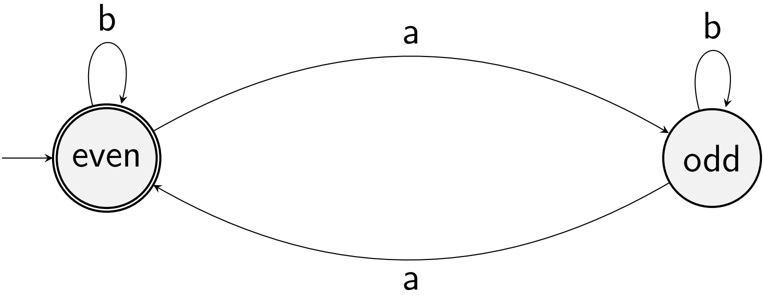
- $Q = \{\text{even}, \text{odd}\}$

Formal definition example



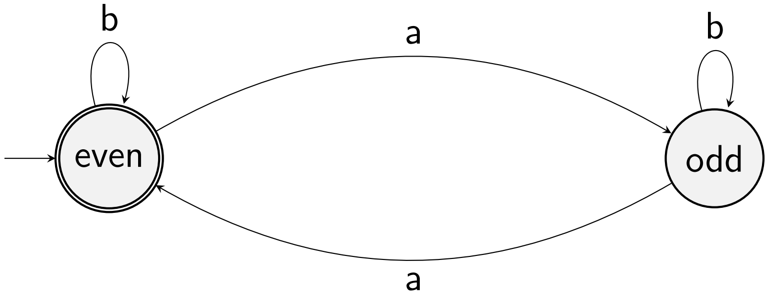
- Q = {even, odd}
- $\Sigma = \{a, b\}$

Formal definition example



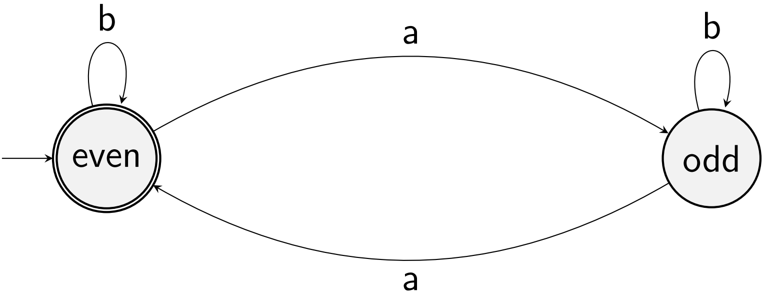
- $Q = \{\text{even}, \text{odd}\}$
- $\Sigma = \{a, b\}$
- $q_s = \text{even}$

Formal definition example



- $Q = \{\text{even}, \text{odd}\}$
- $\Sigma = \{a, b\}$
- $q_s = \text{even}$
- $F = \{\text{even}\}$

Formal definition example

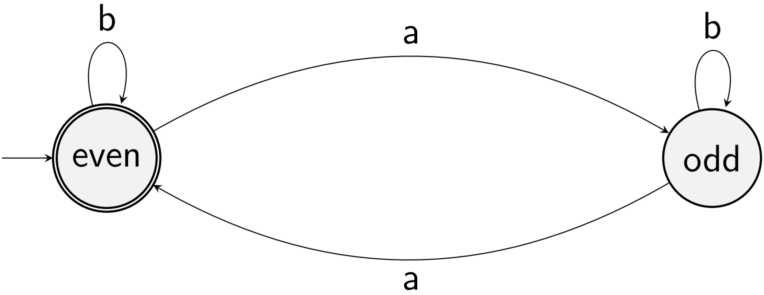


- $Q = \{\text{even, odd}\}$
- $\Sigma = \{a, b\}$
- $q_s = \text{even}$
- $F = \{\text{even}\}$

- $\delta =$

	a	b

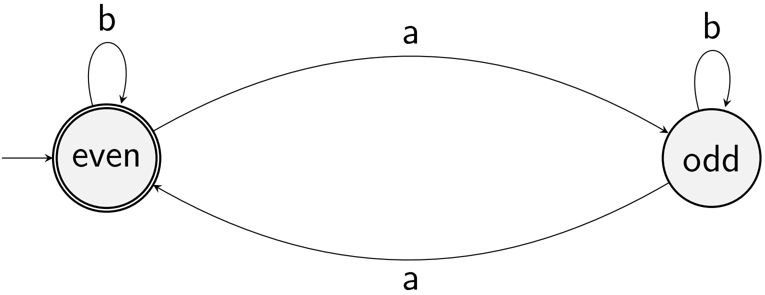
Formal definition example



- $Q = \{\text{even, odd}\}$
- $\Sigma = \{a, b\}$
- $q_s = \text{even}$
- $F = \{\text{even}\}$
- $\delta =$

	a	b
even	odd	even

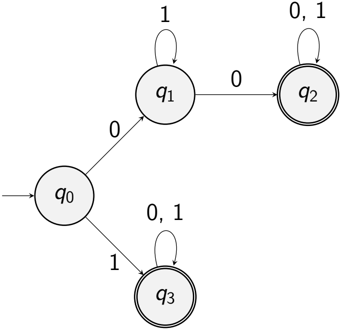
Formal definition example



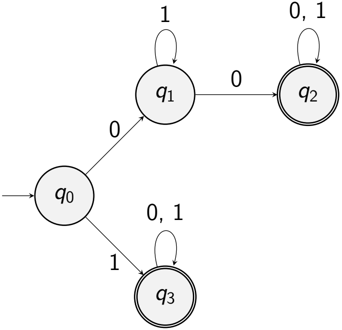
- $Q = \{\text{even, odd}\}$
- $\Sigma = \{a, b\}$
- $q_s = \text{even}$
- $F = \{\text{even}\}$
- $\delta =$

	a	b
even	odd	even
odd	even	odd

Write the formal definition

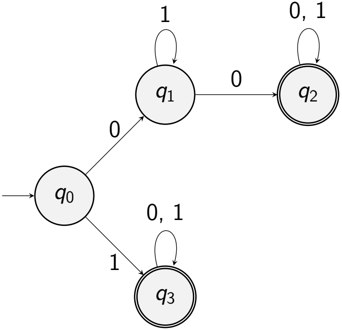


Write the formal definition



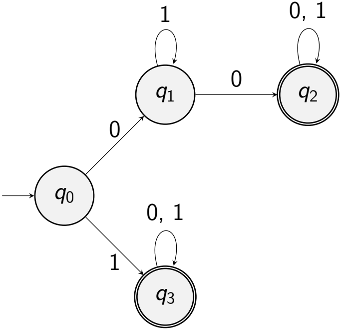
- $Q = \{q_0, q_1, q_2, q_3\}$

Write the formal definition



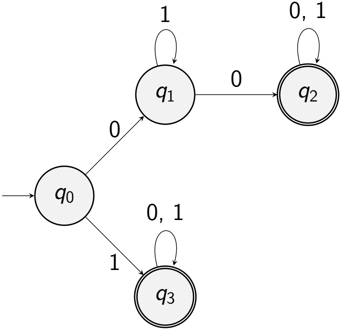
- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$

Write the formal definition



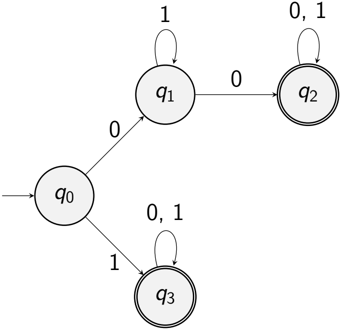
- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$

Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$

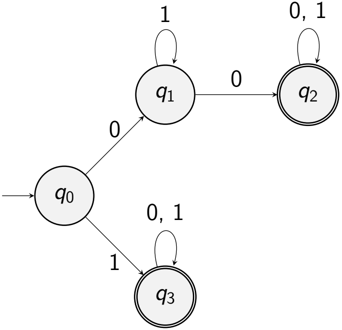
Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$
- $\delta =$

	0	1

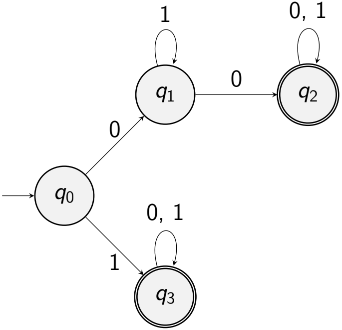
Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$
- $\delta =$

	0	1
q_0	q_1	q_3

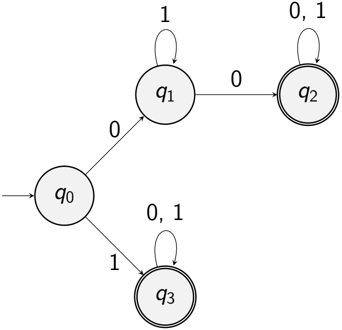
Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$
- $\delta =$

	0	1
q_0	q_1	q_3
q_1	q_2	q_1

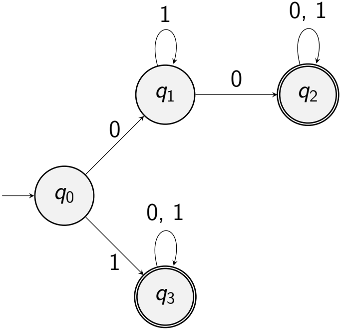
Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$
- $\delta =$

	0	1
q_0	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_2

Write the formal definition



- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1\}$
- $q_s = q_0$
- $F = \{q_2, q_3\}$
- $\delta =$

	0	1
q_0	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_2
q_3	q_3	q_3

The language of a DFA

The language of a DFA

- Let D be a DFA

The language of a DFA

- Let D be a DFA
- We say a DFA D **recognizes** a language L if for all $w \in \Sigma^*$:

The language of a DFA

- Let D be a DFA
- We say a DFA D **recognizes** a language L if for all $w \in \Sigma^*$:
  - If $w \in L$, then D accepts w

The language of a DFA

- Let D be a DFA
- We say a DFA D **recognizes** a language L if for all $w \in \Sigma^*$:
  - If $w \in L$, then D accepts w
  - If $w \notin L$, then D rejects w

The language of a DFA

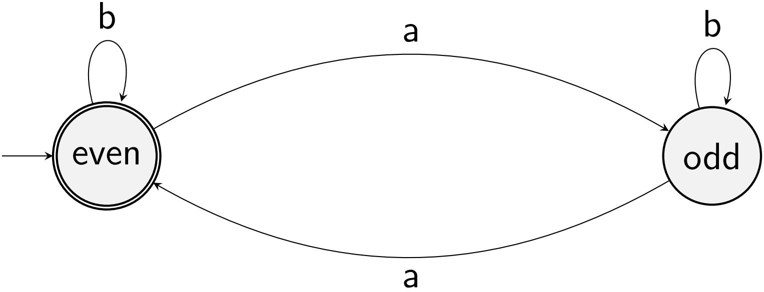
- Let D be a DFA
- We say a DFA D **recognizes** a language L if for all $w \in \Sigma^*$:
  - If $w \in L$, then D accepts w
  - If $w \notin L$, then D rejects w
  - Do we ever have to worry about D looping?

The language of a DFA

- Let D be a DFA
- We say a DFA D **recognizes** a language L if for all $w \in \Sigma^*$:
  - If $w \in L$, then D accepts w
  - If $w \notin L$, then D rejects w
  - Do we ever have to worry about D looping?
- The **language of D**, denoted $L(D)$ is the (unique) language that D recognizes - that is, the set of all strings that D accepts

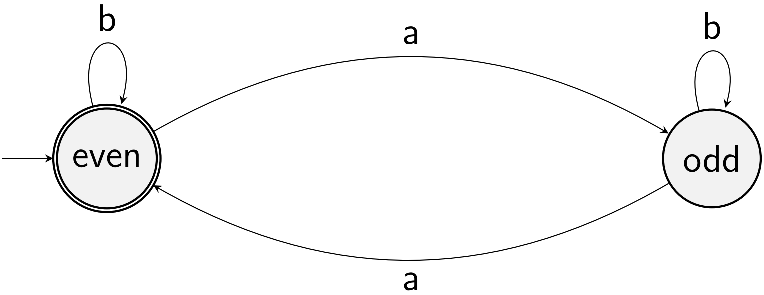
The language of a DFA

What the language of this DFA?



The language of a DFA

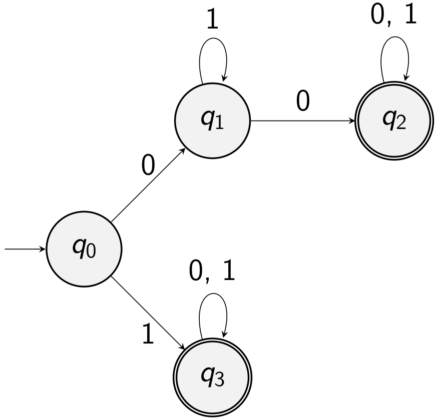
What the language of this DFA?



$L(D) = \{\text{w} \mid \text{w has an even number of a's}\}$

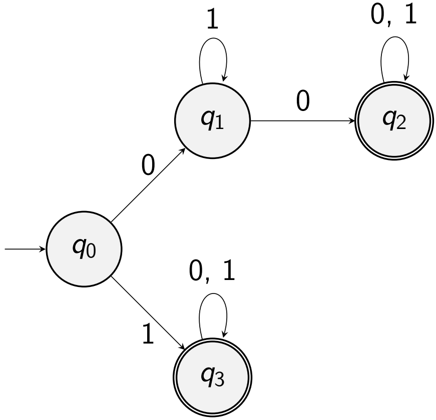
The language of a DFA

What the language of this DFA?



The language of a DFA

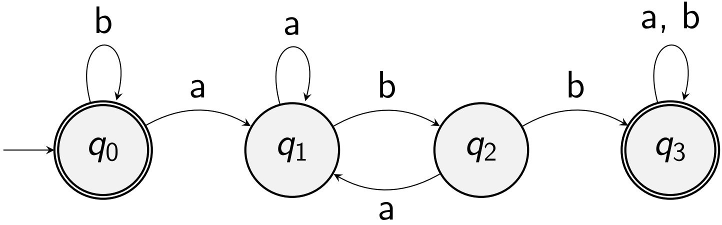
What the language of this DFA?



$L(D) = \{$w | w starts with 1 or has at least two 0's$\}$

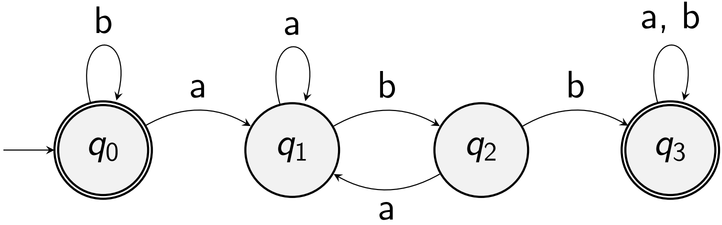
The language of a DFA

What the language of this DFA?



The language of a DFA

What the language of this DFA?



$L(Q) = \{\text{w} \mid \text{w contains only b's or contains abb}\}$

Deterministic Finite Automata

Some comments:

Deterministic Finite Automata

Some comments:

- A transition must be defined for every symbol in the alphabet, for every state in the DFA.

Deterministic Finite Automata

Some comments:

- A transition must be defined for every symbol in the alphabet, for every state in the DFA.
- There can be more than one accepting state.

Deterministic Finite Automata

Some comments:

- A transition must be defined for every symbol in the alphabet, for every state in the DFA.
- There can be more than one accepting state.
- The set of languages that can be recognized by some DFA is called the **regular languages**.

Deterministic Finite Automata

Some comments:

- A transition must be defined for every symbol in the alphabet, for every state in the DFA.
- There can be more than one accepting state.
- The set of languages that can be recognized by some DFA is called the **regular languages**.
    - A language L is regular if and only if some DFA D recognizes L

Design a DFA

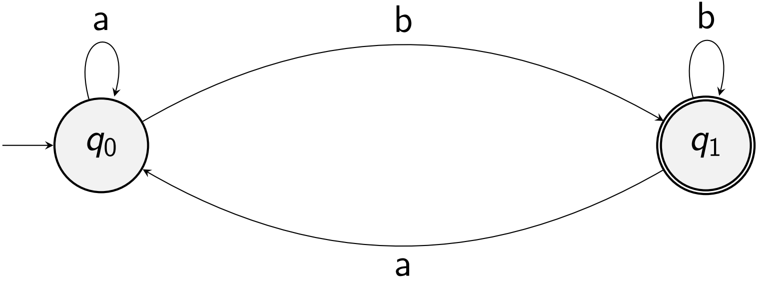
Let $\Sigma = \{a, b\}$. Let's show that the following language is regular:

$$\text{L} = \{\text{w} \mid \text{w ends with b}\}$$

Design a DFA

Let $\Sigma = \{a, b\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ ends with b}\}$$



Design a DFA

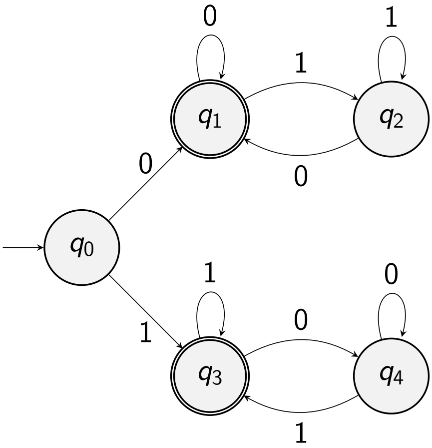
Let $\Sigma = \{0, 1\}$. Let's show that the following language is regular:

$\text{L} = \{\text{w} \mid \text{w starts and ends with the same symbol}\}$

Design a DFA

Let $\Sigma = \{0, 1\}$. Let's show that the following language is regular:

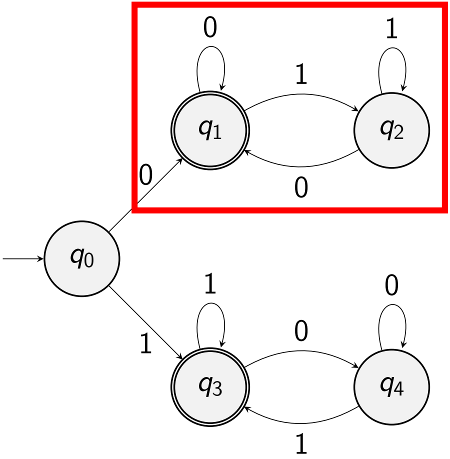
L $= \{$w $\mid$ w starts and ends with the same symbol$\}$



Design a DFA

Let $\Sigma = \{0, 1\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ starts and ends with the same symbol}\}$$

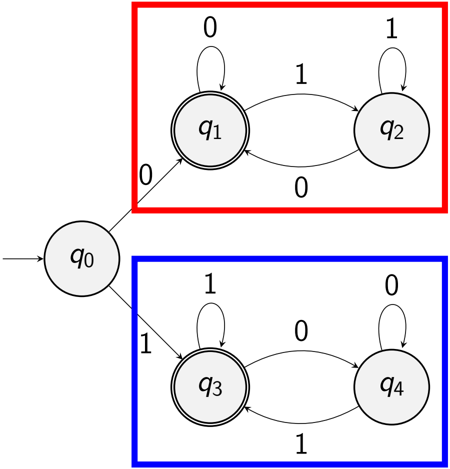


starts/ends with 0

Design a DFA

Let $\Sigma = \{0, 1\}$. Let's show that the following language is regular:

$$\text{L} = \{\text{w} \mid \text{w starts and ends with the same symbol}\}$$



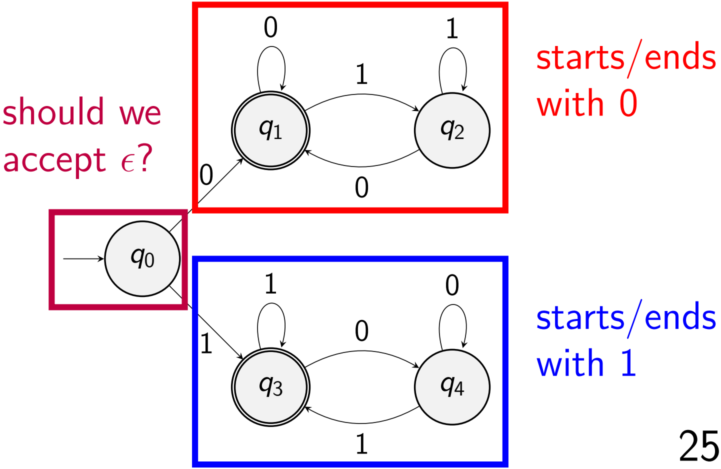
starts/ends with 0

starts/ends with 1

Design a DFA

Let $\Sigma = \{0, 1\}$. Let's show that the following language is regular:

L $= \{$w $|$ w starts and ends with the same symbol$\}$



Design a DFA

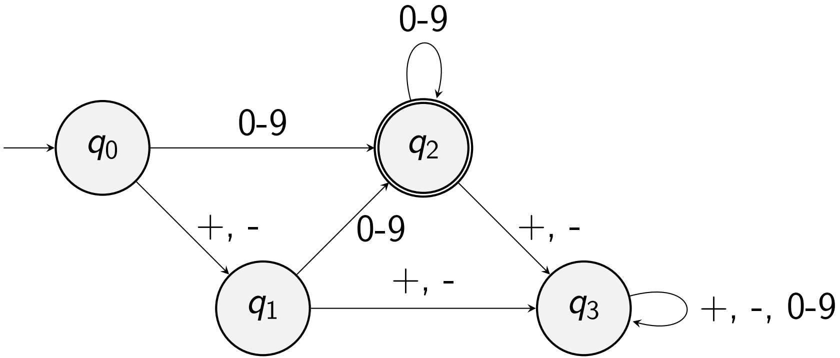
Let $\Sigma = \{+, -, 0, 1, \ldots, 9\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ is a valid integer literal}\}$$

Design a DFA

Let $\Sigma = \{+, -, 0, 1, \ldots, 9\}$. Let's show that the following language is regular:

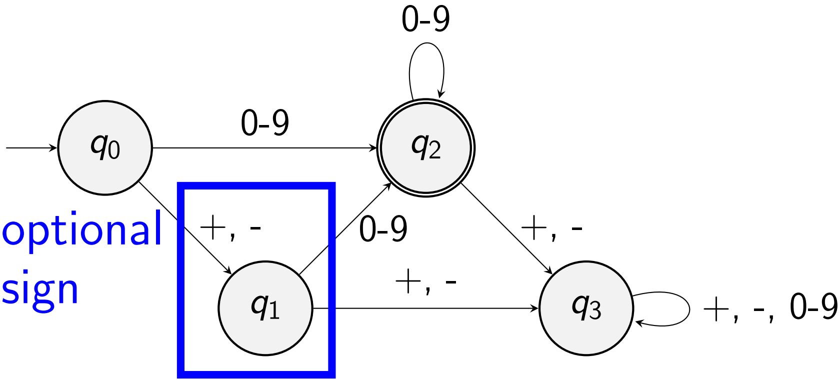
$$L = \{w \mid w \text{ is a valid integer literal}\}$$



Design a DFA

Let $\Sigma = \{+, -, 0, 1, \ldots, 9\}$. Let's show that the following language is regular:

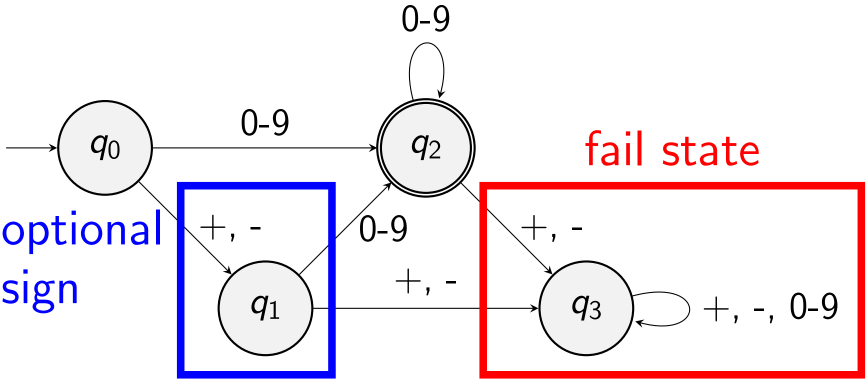
$$L = \{w \mid w \text{ is a valid integer literal}\}$$



Design a DFA

Let $\Sigma = \{+, -, 0, 1, \ldots, 9\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ is a valid integer literal}\}$$



Design a DFA

Let $\Sigma = \{a, b\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ ends in bba}\}$$

Design a DFA

Let $\Sigma = \{a, b\}$. Let's show that the following language is regular:

$$L = \{w \mid w \text{ ends in bba}\}$$