Theoretical Computer Science

Tutorial Week 3

Prof. Andrey Frolov

Innopolis University

Agenda

Finite State Automaton (FSA)

- What is a FSA?
- Formal Definition
- Languages accepted by FSAs

What is a FSA?

Models of computations

- finite automata
- pushdown automata
- Turing machines
- ...

What is a FSA?

What is a Finite State Automaton intuitively?

Let's see movies!

What is a FSA?

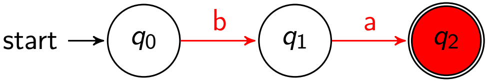
What is a Finite State Automaton intuitively?

Let's see movies!

Finite State Automaton $\Leftrightarrow$ "Mechanical" Machine
(likes Save Lock)

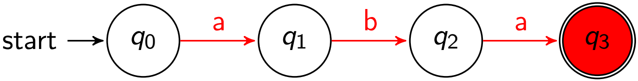
What is a FSA?

Example 1: ba



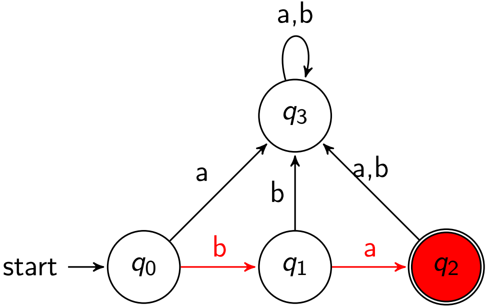
What is a FSA?

Example 2: aba



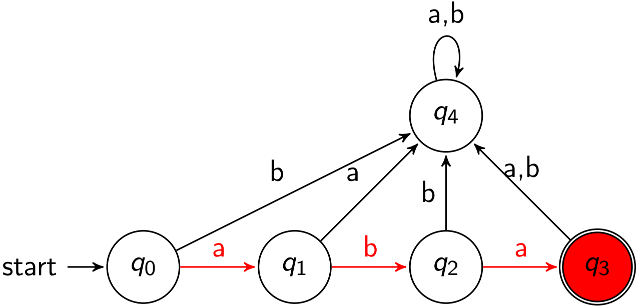
What is a FSA?

Example 1: "Trap" State



What is a FSA?

Example 2: “Trap” State



Agenda

Finite State Automaton (FSA)

- What is a FSA?
- **Formal Definition**
- Languages accepted by FSAs

FSA (Formal definition)

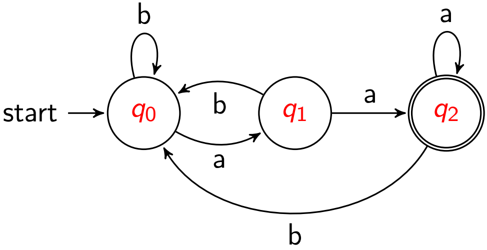
Definition

A (complete) Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where

Q is a finite set of *states*;
Σ is a finite *input alphabet*;
$q_0 \in Q$ is the *initial* state;
$A \subseteq Q$ is the set of *accepting* states;
$\delta : Q \times \Sigma \to Q$ is a (total) *transition* function.

FSA (Formal definition)

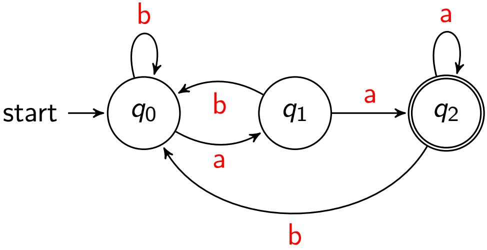
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



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

FSA (Formal definition)

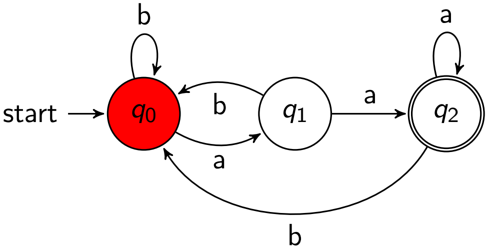
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



$\Sigma = \{a, b\}$

FSA (Formal definition)

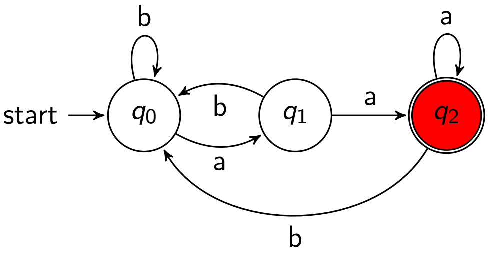
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



q_0 is the initial state

FSA (Formal definition)

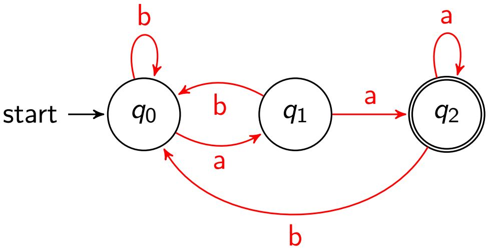
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



$A = \{q_2\}$ is the set of *accepting* states

FSA (Formal definition)

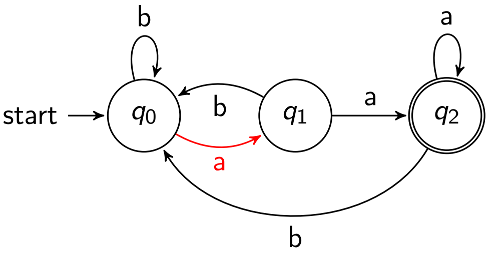
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



$\delta : Q \times \Sigma \to Q$ is a *transition* function

FSA (Formal definition)

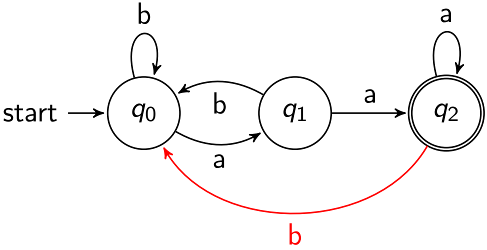
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



$\delta(q_0, a) = q_1$

FSA (Formal definition)

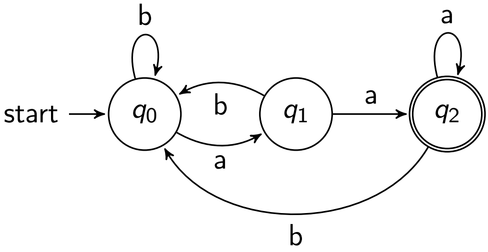
A Finite State Automaton is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where



$\delta(q_2, b) = q_0$

FSA (Formal definition)

$\delta : Q \times \Sigma \rightarrow Q$ is a *transition* function



δ	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_0

Agenda

Finite State Automaton (FSA)

- What is a FSA?
- Formal Definition
- **Languages accepted by FSAs**
  - Extended Transition
  - Languages

The extended transition

Definition

Let $M =< Q, \Sigma, q_0, A, \delta >$ be a complete finite state automaton. We define the extended transition function

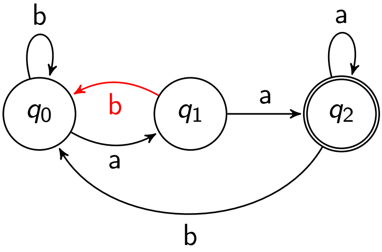
$$\delta^* : Q \times \Sigma^* \to Q$$

as follows:

1. For every $q \in Q$, $\delta^*(q, \epsilon) = q$
2. For every $q \in Q$, every $y \in \Sigma^*$, and every $\sigma \in \Sigma$,

$$\delta^*(q, y\sigma) = \delta(\delta^*(q, y), \sigma)$$

The extended transition (Example)

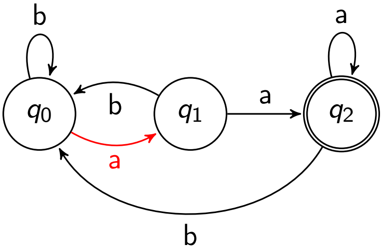


$$\begin{aligned}
\delta^*(q_1, bab) &= \delta(\delta^*(q_1, ba), b) = \\
&= \delta(\delta(\delta^*(q_1, b), a), b) = \\
&= \delta(\delta(\delta(\delta^*(q_1, \epsilon), b), a), b) = \\
&= \delta(\delta(\delta(q_1, b), a), b) = \\
&= \delta(\delta(q_0, a), b) = \\
&= \delta(q_1, b) = q_0
\end{aligned}$$

Intuitively:

$$\begin{aligned}
\delta(q_1, b) &= q_0 \\
\delta(q_0, a) &= q_1 \\
\delta(q_1, b) &= q_0
\end{aligned}$$

The extended transition (Example)



$$
\begin{aligned}
\delta^*(q_1, b\textcolor{red}{a}b) &= \delta(\delta^*(q_1, ba), b) = \\
&= \delta(\delta(\delta^*(q_1, b), a), b) = \\
&= \delta(\delta(\delta(\delta^*(q_1, \epsilon), b), a), b) = \\
&= \delta(\delta(\delta(q_1, b), a), b) = \\
&= \delta(\delta(q_0, a), b) = \\
&= \delta(q_1, b) = q_0
\end{aligned}
$$

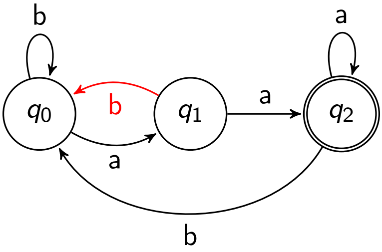
Intuitively:

$\delta(q_1, b) = q_0$

$\textcolor{red}{\delta(q_0, a) = q_1}$

$\delta(q_1, b) = q_0$

The extended transition (Example)



$$
\begin{aligned}
\delta^*(q_1, bab) \quad &= \delta(\delta^*(q_1, ba), b) = \\
&= \delta(\delta(\delta^*(q_1, b), a), b) = \\
&= \delta(\delta(\delta(\delta^*(q_1, \epsilon), b), a), b) = \\
&= \delta(\delta(\delta(q_1, b), a), b) = \\
&= \delta(\delta(q_0, a), b) = \\
&= \delta(q_1, b) = q_0
\end{aligned}
$$

Intuitively:

$\delta(q_1, b) = q_0$

$\delta(q_0, a) = q_1$

$\delta(q_1, b) = q_0$

Agenda

Finite State Automaton (FSA)

- What is a FSA?
- Formal Definition
- Languages accepted by FSAs
  - Extended Transition
  - **Languages**

Languages accepted by FSAs

Let $M =< Q, \Sigma, q_0, A, \delta >$ be a FSA.

Definition

The string $x \in \Sigma^*$ is accepted by M if

$$\delta^*(q_0, x) \in A$$

and it is rejected by M, otherwise.

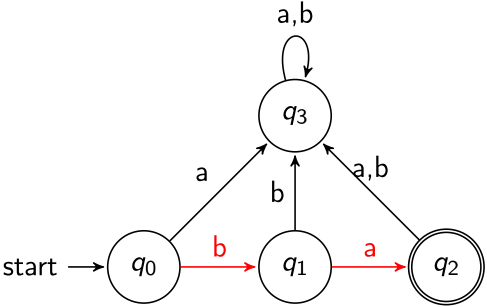
Definition

The language accepted by M is the set

$$L = \{x \in \Sigma^* \mid x \text{ is accepted by } M\}$$

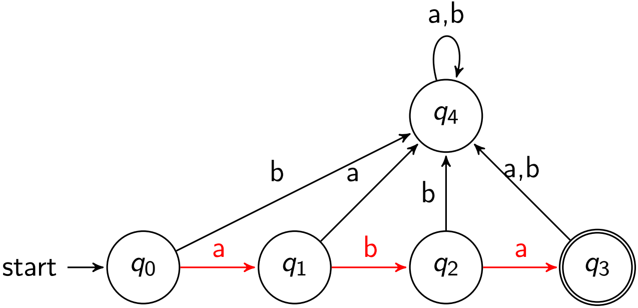
Languages accepted by FSAs

Example 1: $L = \{ba\}$



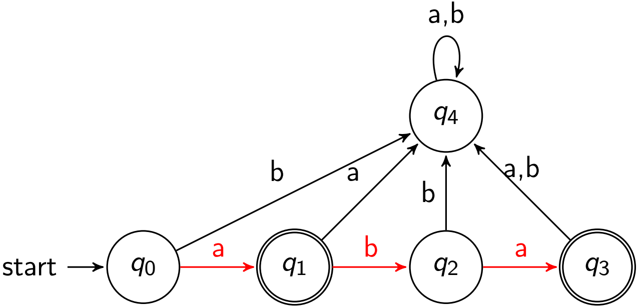
Languages accepted by FSAs

Example 2: $L = \{aba\}$



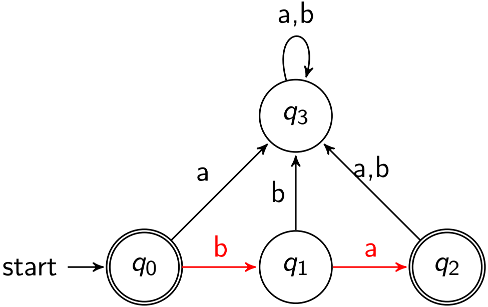
Languages accepted by FSAs

Example 3: $L = \{a, aba\}$



$A = \{q_1, q_3\}$ is the set of *accepting* states

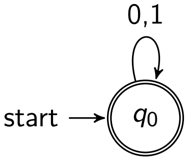
Example 4: $L = \{\epsilon, ba\}$



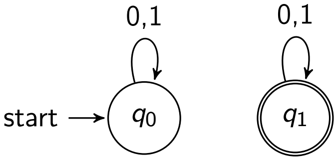
$A = \{q_0, q_2\}$ is the set of *accepting* states

Languages accepted by FSAs

Example 5: $L = \Sigma^*$, where $\Sigma = \{0, 1\}$



Example 6: $L = \emptyset$, where $\Sigma = \{0, 1\}$



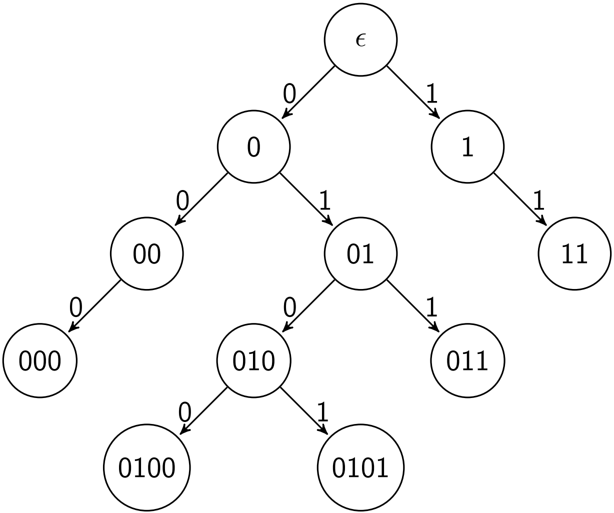
Finite languages

What about finite languages?

$L = \emptyset$, what else?

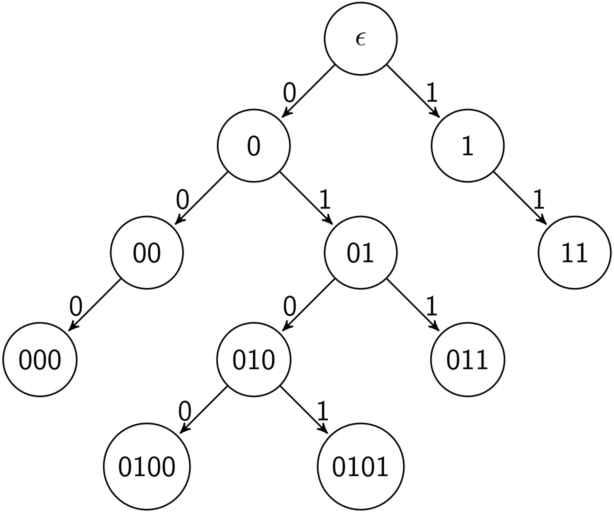
Finite languages

Binary Tree



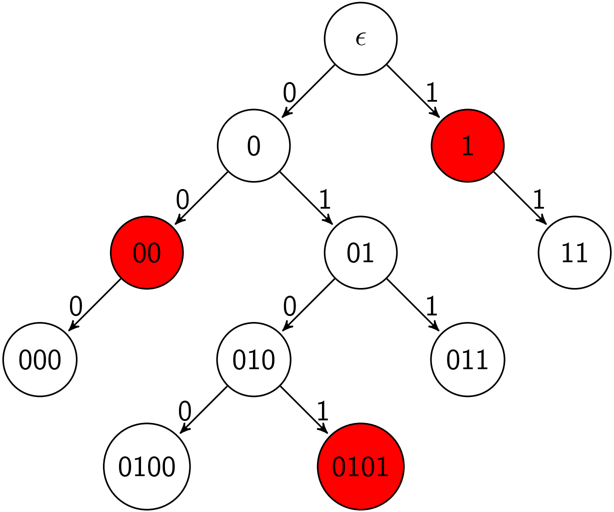
Finite languages

$L = \{1, 00, 0101\}$



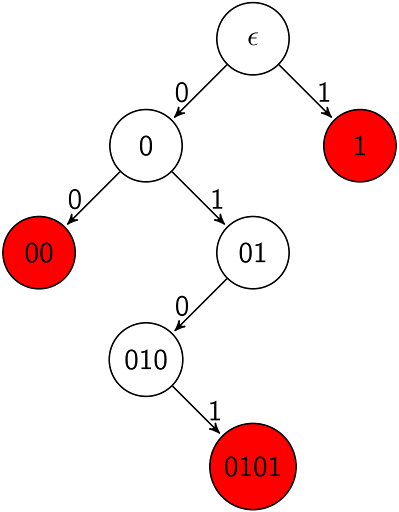
Finite languages

$L = \{1, 00, 0101\}$



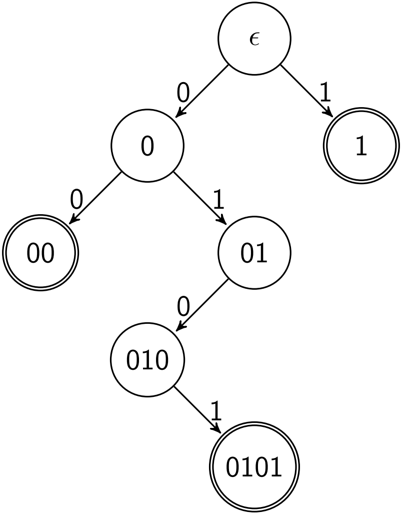
Finite languages

$L = \{1, 00, 0101\}$



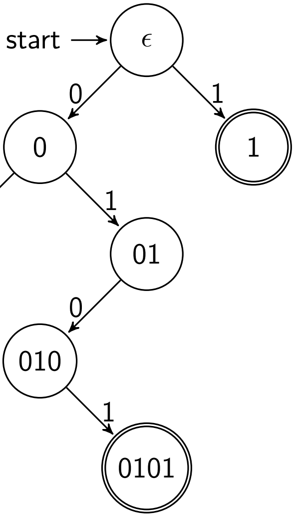
Finite languages

$L = \{1, 00, 0101\}$



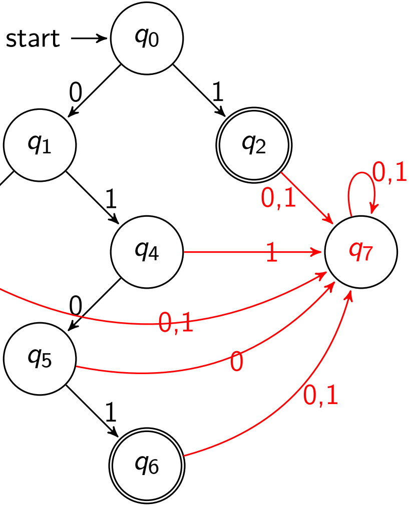
Finite languages

$L = \{1, 00, 0101\}$



Finite languages

$L = \{1, 00, 0101\}$



Languages accepted by FSAs

What about infinite languages?

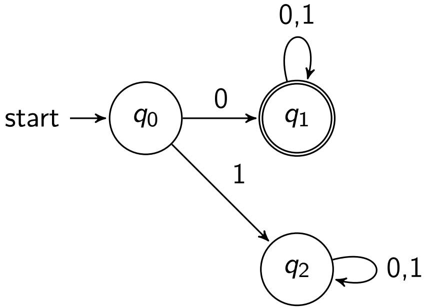
$$\begin{aligned} L &= \{\epsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\} \\ &= \{a^k b^k \mid k \in \mathbb{N}\} \end{aligned}$$

is not accepted by any FSA!

Languages accepted by FSAs

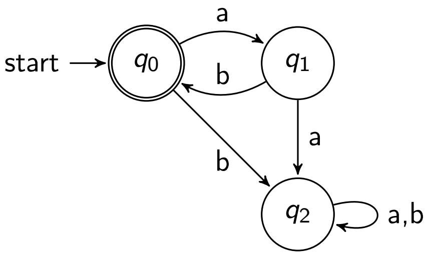
Example 1: $L = \{0x \mid x \in \Sigma^*\}$, where $\Sigma = \{0, 1\}$, i.e.,

$$L = \{w \in \Sigma^* \mid w \text{ starts with } 0\}$$



Languages accepted by FSAs

Example 2: $L = \{(ab)^k \mid k \in \mathbb{N}\}$, where $\Sigma = \{a, b\}$



Thank you for your attention!