Theory of Computer Science B3. Finite Automata

Gabriele Röger

University of Basel

March 11, 2024

Gabriele Röger (University of Basel)

Gabriele Röger (University of Basel)

B3. Finite Automata

Theory of Computer Science

March 11, 2024

Introduction

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024 2

March 11, 2024

B3.1 Introduction

Theory of Computer Science March 11, 2024 — B3. Finite Automata

B3.1 Introduction

B3.2 DFAs

B3.3 NFAs

Gabriele Röger (University of Basel)

B3. Finite Automata Introduction Content of the Course finite automata grammars closure & decidability automata theory & regular formal languages languages regular expressions computability & context-free decidability languages pumping lemma context-sensitive complexity theory and general languages

Theory of Computer Science

Theory of Computer Science March 11, 2024 3 / 32

B3. Finite Automata Introduction

A Controller for a Turnstile



CC BY-SA 3.0, author: Stolbovsky

- simple access control
- card reader and push sensor
- card can either be valid or invalid

push, validcard, invalidcard invalidcard locked unlocked push

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024 5 / 32

B3. Finite Automata

B3. Finite Automata DFAs

B3.2 DFAs

Gabriele Röger (University of Basel) Theory of Computer Science March 11, 2024 B3. Finite Automata Content of the Course DFA finite automata NFA grammars closure & decidability automata theory & regular formal languages languages regular expressions computability & context-free decidability languages pumping lemma complexity context-sensitive theory and general languages March 11, 2024 Gabriele Röger (University of Basel) Theory of Computer Science

Finite automata are a good model for computers

► We will not consider automata that run forever but that process a finite input sequence and

Where can the turnstile controller store information

with very limited memory.

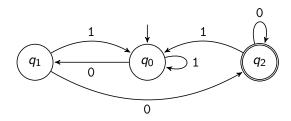
about what it has seen in the past?

then classify it as accepted or not.

Gabriele Röger (University of Basel) Theory of Computer Science March 11, 2024 7 / 32

B3. Finite Automata

Finite Automaton: Example



When reading the input 01100 the automaton visits the states $q_0, q_1, q_0, q_0, q_1, q_2.$

Gabriele Röger (University of Basel)

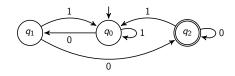
Theory of Computer Science

March 11, 2024

March 11, 2024

B3. Finite Automata

Finite Automata: Terminology and Notation



- ▶ states $Q = \{q_0, q_1, q_2\}$
- $\delta(q_0,0)=q_1$
- q_1
- ▶ input alphabet $\Sigma = \{0, 1\}$ $\delta(q_0, 1) = q_0$

ightharpoonup transition function δ

► accept states {*q*₂}

- $\delta(q_1,0)=q_2$
- $q_2 q_0$ q_2 $q_2 q_0$

table form of δ

 \triangleright start state q_0

- $\delta(q_1,1)=q_0$
- $\delta(q_2,0)=q_2$
- $\delta(q_2, 1) = q_0$

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

B3. Finite Automata

Deterministic Finite Automaton: Definition

Definition (Deterministic Finite Automata)

A deterministic finite automaton (DFA) is a 5-tuple $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ where

- Q is the finite set of states
- \triangleright Σ is the input alphabet
- $lackbox{} \delta: Q \times \Sigma \rightarrow Q$ is the transition function
- $ightharpoonup q_0 \in Q$ is the start state
- $ightharpoonup F \subset Q$ is the set of accept states (or final states)

B3. Finite Automata

DFA: Accepted Words

Intuitively, a DFA accepts a word if its computation terminates in an accept state.

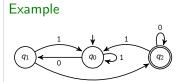
Definition (Words Accepted by a DFA)

DFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ accepts the word $w = a_1 \dots a_n$ if there is a sequence of states $q'_0, \ldots, q'_n \in Q$ with

- $q_0' = q_0,$
- $g_n' \in F.$

B3. Finite Automata DFA:

Example



accepts: 00 10010100

01000

does not accept:

1001010 010001

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

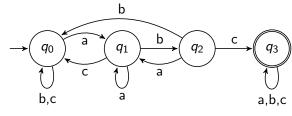
13 / 32

D3. I lilite Automata

B3. Finite Automata DFAs

Exercise (slido)

Consider the following DFA:



Which of the following words does it accept?

- ▶ abc
- ababcb
- ► babbc



Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

14 / 32

B3. Finite Automata

DFA: Recognized Language

Definition (Language Recognized by a DFA)

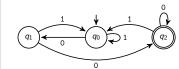
Let M be a deterministic finite automaton. The language recognized by M is defined as

 $\mathcal{L}(M) = \{ w \in \Sigma^* \mid w \text{ is accepted by } M \}.$

B3. Finite Automata

Example

Example



The DFA recognizes the language $\{w \in \{0,1\}^* \mid w \text{ ends with } 00\}.$

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

15 / 32

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

16 / 2

B3. Finite Automata DF.

A Note on Terminology

► In the literature, "accept" and "recognize" are sometimes used synonymously or the other way around.

DFA recognizes a word or accepts a language.

► We try to stay consistent using the previous definitions (following the text book by Sipser).

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

17 / 32

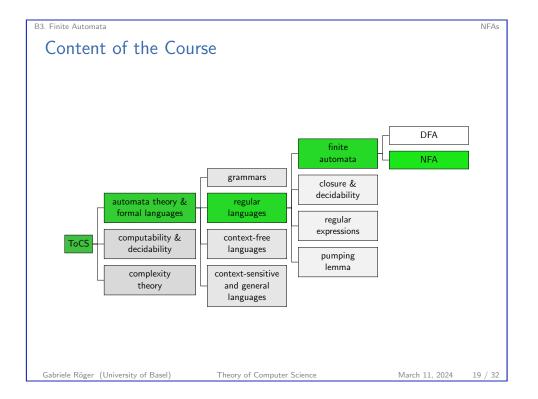
Gabriele Röger (University of Basel)

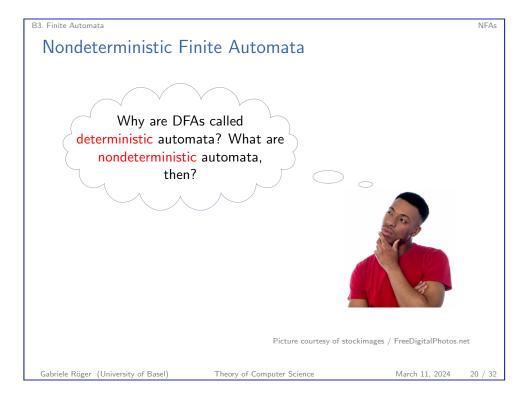
B3.3 NFAs

Theory of Computer Science

March 11, 2024

18 / 32





B3. Finite Automata

In what Sense is a DFA Deterministic?

- ► A DFA has a single fixed state from which the computation starts.
- ▶ When a DFA is in a specific state and reads an input symbol, we know what the next state will be.
- For a given input, the entire computation is determined.
- ► This is a deterministic computation.

Gabriele Röger (University of Basel)

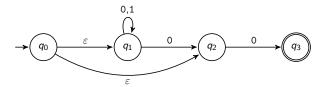
Theory of Computer Science

March 11, 2024

21 / 32

B3. Finite Automata

Nondeterministic Finite Automata: Example



differences to DFAs:

- ▶ transition function δ can lead to zero or more successor states for the same $a \in \Sigma$
- **ε-transitions** can be taken without "consuming" a symbol from the input
- ► the automaton accepts a word if there is at least one accepting sequence of states

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

B3. Finite Automata

Automata

Nondeterministic Finite Automaton: Definition

Definition (Nondeterministic Finite Automata)

A nondeterministic finite automaton (NFA) is a 5-tuple

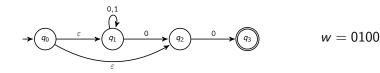
 $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ where

- Q is the finite set of states
- $ightharpoonup \Sigma$ is the input alphabet
- ▶ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ is the transition function (mapping to the power set of Q)
- ▶ $q_0 \in Q$ is the start state
- $ightharpoonup F \subseteq Q$ is the set of accept states

DFAs are (essentially) a special case of NFAs.

B3. Finite Automata

Accepting Computation: Example



Gabriele Röger (University of Basel) Theory of Computer Science

March 11, 2024

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

B3. Finite Automata

ε -closure of a State

For a state $q \in Q$, we write E(q) to denote the set of states that are reachable from q via ε -transitions in δ .

Definition (ε -closure)

For NFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ and state $q \in Q$, state p is in the ε -closure E(q) of q iff there is a sequence of states q'_0, \ldots, q'_n with

- $\mathbf{0} \ q_0' = q$
- $\mathbf{Q} \quad \mathbf{q}_i' \in \delta(\mathbf{q}_{i-1}', \varepsilon) \text{ for all } i \in \{1, \dots, n\} \text{ and } \mathbf{q}_i' \in \mathbf{q}_i'$
- **3** $q'_{p} = p$.

 $q \in E(q)$ for every state q

Gabriele Röger (University of Basel)

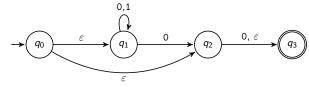
Theory of Computer Science

March 11, 2024

B3. Finite Automata

Exercise (slido)

Consider the following NFA:



Which states are in the ε -closure $E(q_0)$?

- q_0



Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

B3. Finite Automata

NFA: Accepted Words

Definition (Words Accepted by an NFA)

NFA $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ accepts the word $w = a_1 \dots a_n$ if there is a sequence of states $q_0',\ldots,q_n'\in Q$ with

- $q_0' \in E(q_0),$
- ② $q_i' \in \bigcup_{q \in \delta(q_{i-1}', a_i)} E(q)$ for all $i \in \{1, \dots, n\}$ and
- $g_n' \in F$.

B3. Finite Automata Example: Accepted Words

Example

10010100

accepts:

01000

does not accept:

1001010

010001

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

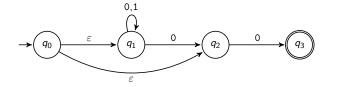
Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024

B3. Finite Automata NFAs

Exercise (slido)





Does this NFA accept input 01010?

Gabriele Röger (University of Basel)

Theory of Computer Science

March 11, 2024 29

B3. Finite Automata

NFA: Recognized Language

Definition (Language Recognized by an NFA)

Let M be an NFA with input alphabet Σ .

The language recognized by M is defined as $\mathcal{L}(M) = \{ w \in \Sigma^* \mid w \text{ is accepted by } M \}.$

Gabriele Röger (University of Basel)

Theory of Computer Science

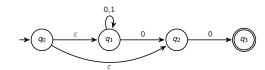
March 11, 2024

30 / 32

B3. Finite Automata

Example: Recognized Language

Example



The NFA recognizes the language $\{w \in \{0,1\}^* \mid w = 0 \text{ or } w \text{ ends with } 00\}.$

B3. Finite Automata

Summary

Summary

- ▶ DFAs are automata where every state transition is uniquely determined.
- ► NFAs can have zero, one or more transitions for a given state and input symbol.
- NFAs can have ϵ -transitions that can be taken without reading a symbol from the input.
- ► NFAs accept a word if there is at least one accepting sequence of states.

March 11, 2024 31 / 32

Theory of Computer Science

March 11, 2024

Gabriele Röger (University of Basel)

Theory of Computer Science

Gabriele Röger (University of Basel)