

# Übersicht

## 2 Reguläre Sprachen

- 2.1 Reguläre Ausdrücke
- 2.2 Endliche Automaten
- 2.3 Nichtdeterministische endliche Automaten
- 2.4 Die Potenzmengenkonstruktion
- 2.5 NFAs mit  $\epsilon$ -Übergängen
- 2.6 Minimale DFAs und der Satz von Myhill-Nerode
- 2.7 Berechnung des minimalen DFA
- 2.8 Umwandlung eines Automaten in einen regulären Ausdruck II
- 2.9 Das Pumping-Lemma
- 2.10 Entscheidungsprobleme für reguläre Sprachen

# Das Leerheitsproblem für reguläre Sprachen

## Satz 2.10.1

*Das Leerheitsproblem für reguläre Sprachen ist in polynomieller Zeit lösbar:*

*Eingabe: Ein DFA, NFA oder ein regulärer Ausdruck für Sprache  $L$*

*Frage: Ist  $L = \emptyset$ ?*

## Beweis.

DFA, NFA: Ist ein Endzustand vom Anfangszustand erreichbar?

Lineare Laufzeit.

Regulärer Ausdruck: Lineare Laufzeit. □

# Das Wortproblem für reguläre Sprachen

## Satz 2.10.2

*Das Wortproblem für reguläre Sprachen ist in polynomieller Zeit lösbar:*

*Eingabe: Ein DFA, NFA oder ein regulärer Ausdruck für Sprache  $L$*

*Frage: Ist  $w \in L$ ?*

## Beweis.

DFA: Algorithmus aus Vorlesung

Laufzeit:  $O(|w|)$

NFA: Algorithmus aus Vorlesung

Laufzeit:  $O(|Q| \cdot |w|)$

Regulärer Ausdruck: In NFA verwandeln.



# Das Universalitätsproblem für reguläre Sprachen

## Satz 2.10.3

*Das Universalitätsproblem für reguläre Sprachen ist algorithmisch lösbar:*

*Eingabe: DFA, NFA oder regulärer Ausdruck für Sprache  $L$*

*Frage: Ist  $L = \Sigma^*$ ?*

## Beweis.

DFA: Sind alle erreichbaren Zustände Endzustände?

Lineare Laufzeit.

NFA, regulärer Ausdruck: In DFA verwandeln.

Exponentielle Laufzeit.



# Das Endlichkeitsproblem für reguläre Sprachen

## Satz 2.10.4

*Das Endlichkeitsproblem für reguläre Sprachen ist in Linearzeit lösbar:*

*Eingabe: DFA, NFA oder regulärer Ausdruck für Sprache  $L$*

*Frage: Ist  $L$  endlich?*

## Beweis.

NFA, DFA: Gibt es

- 1 eine starke Zusammenhangskomponente, die vom Startzustand erreichbar ist, und
- 2 von der aus ein Endzustand erreichbar ist?

Lineare Laufzeit.

Regulärer Ausdruck: Lineare Laufzeit.



# Das Äquivalenzproblem für reguläre Sprachen

## Satz 2.10.5

*Das Äquivalenzproblem für reguläre Sprachen ist algorithmisch lösbar:*

*Eingabe: Zwei DFAs, NFAs oder reguläre Ausdrücke für Sprachen  $L_1$  und  $L_2$*

*Frage: Ist  $L_1 = L_2$ ?*

## Beweis.

DFA: Minimiere beide und teste ob sie dann isomorph sind.

NFA, Regulärer Ausdruck: In DFA verwandeln. □