

# Übung 2: Eindeutigkeit und Automaten

## Theoretische Informatik Sommersemester 2014

Markus Kaiser

26. April 2014

## Definition (kontextfreie Linksableitung)

Eine Ableitung

$$S \rightarrow^* xAz \rightarrow x\beta z \rightarrow^* w$$

heißt (kontextfreie) **Linksableitung**, wenn für jede Anwendung jeder Produktion  $A \rightarrow \beta$  gilt, dass in  $x$  kein Nichtterminal vorkommt.

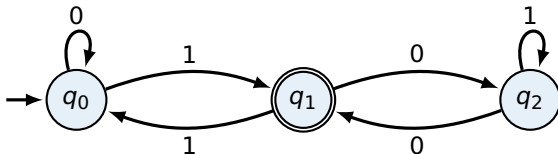
## Definition (Eindeutigkeit)

- Eine Grammatik heißt **eindeutig**, wenn es für jedes Wort genau eine Linksableitung gibt.
- Eine Sprache heißt **eindeutig**, wenn es für sie eine eindeutige Grammatik gibt.

## Definition (Deterministischer endlicher Automat)

Ein DFA ist ein Tupel  $M = (Q, \Sigma, \delta, q_0, F)$  aus einer/einem

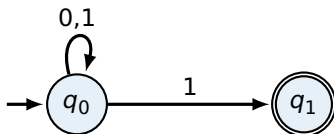
- endlichen Menge von Zuständen  $Q$
- endlichen Eingabealphabet  $\Sigma$
- totalen Übergangsfunktion  $\delta : Q \times \Sigma \rightarrow Q$
- Startzustand  $q_0 \in Q$
- Menge von Endzuständen  $F \subseteq Q$



## Definition (Nicht-Deterministischer endlicher Automat)

Ein **NFA** ist ein Tupel  $N = (Q, \Sigma, \delta, q_0, F)$  mit

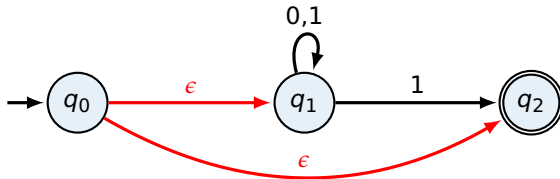
- $Q, \Sigma, q_0, F$  wie ein DFA
- **Übergangsfunktion**  $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$



## Definition (NFA mit $\epsilon$ -Übergängen)

Ein  $\epsilon$ -NFA ist ein Tupel  $N = (Q, \Sigma, \delta, q_0, F)$  mit

- $Q, \Sigma, q_0, F$  wie ein DFA
- Übergangsfunktion  $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$



## Übergangsfunktionen

Die Automaten  $A = (Q, \Sigma, \delta, q_0, F)$  unterscheiden sich nur durch ihre Übergangsfunktionen.

**DFA**  $\delta : Q \times \Sigma \rightarrow Q$

**NFA**  $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

**$\epsilon$ -NFA**  $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathcal{P}(Q)$

## Satz

***DFA**, **NFA** und  **$\epsilon$ -NFA** sind gleich mächtig und lassen sich ineinander umwandeln.*

## Potenzmengenkonstruktion

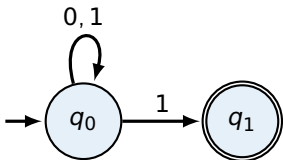
Konstruiere einen Automaten, der **alle möglichen Pfade** gleichzeitig berücksichtigt. Gegeben ein NFA  $(Q, \Sigma, \delta, q_0, F)$ , konstruiere einen DFA mit Zuständen aus  $\mathcal{P}(Q)$ .

- Starte in  $\{q_0\}$
- Die Übergangsfunktion speichert **alle möglichen Schritte**

$$\bar{\delta}: \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$$

$$(S, a) \mapsto \bigcup_{q \in S} \delta(q, a)$$

- $S$  ist Endzustand wenn  $F \cap S \neq \emptyset$



## Potenzmengenkonstruktion

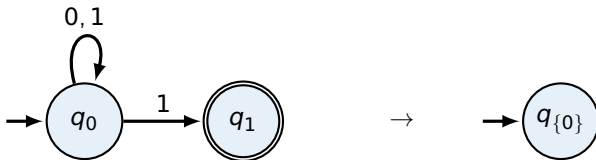
Konstruiere einen Automaten, der **alle möglichen Pfade** gleichzeitig berücksichtigt. Gegeben ein NFA  $(Q, \Sigma, \delta, q_0, F)$ , konstruiere einen DFA mit Zuständen aus  $\mathcal{P}(Q)$ .

- Starte in  $\{q_0\}$
- Die Übergangsfunktion speichert **alle möglichen Schritte**

$$\bar{\delta}: \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$$

$$(S, a) \mapsto \bigcup_{q \in S} \delta(q, a)$$

- $S$  ist Endzustand wenn  $F \cap S \neq \emptyset$





## Potenzmengenkonstruktion

Konstruiere einen Automaten, der **alle möglichen Pfade** gleichzeitig berücksichtigt. Gegeben ein NFA  $(Q, \Sigma, \delta, q_0, F)$ , konstruiere einen DFA mit Zuständen aus  $\mathcal{P}(Q)$ .

- Starte in  $\{q_0\}$
- Die Übergangsfunktion speichert **alle möglichen Schritte**

$$\bar{\delta}: \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$$

$$(S, a) \mapsto \bigcup_{q \in S} \delta(q, a)$$

- $S$  ist Endzustand wenn  $F \cap S \neq \emptyset$



## Potenzmengenkonstruktion

Konstruiere einen Automaten, der **alle möglichen Pfade** gleichzeitig berücksichtigt. Gegeben ein NFA  $(Q, \Sigma, \delta, q_0, F)$ , konstruiere einen DFA mit Zuständen aus  $\mathcal{P}(Q)$ .

- Starte in  $\{q_0\}$
- Die Übergangsfunktion speichert **alle möglichen Schritte**

$$\bar{\delta}: \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$$

$$(S, a) \mapsto \bigcup_{q \in S} \delta(q, a)$$

- $S$  ist Endzustand wenn  $F \cap S \neq \emptyset$

