Übung 2: Eindeutigkeit und Automaten

Theoretische Informatik Sommersemester 2014

Markus Kaiser

26. April 2014

Definition (Linksableitung)

Eine Ableitung

$$S \to^* x\alpha z \to x\beta z \to^* w$$

heißt Linksableitung, wenn für jede Anwendung jeder Produktion $\alpha \to \beta$ gilt, dass sich keine Regel in einem echten Präfix von $x\alpha$ anwenden lässt.

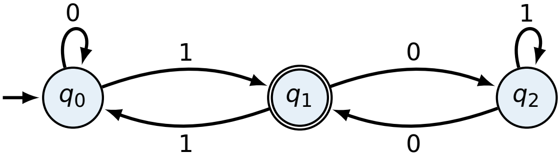
Definition (Eindeutigkeit)

- 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

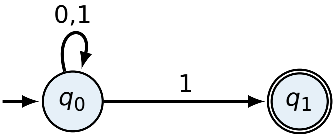
- endlichen Menge von Zuständen Q
- endlichen Eingabealphabet Σ
- totalen Übergangsfunktion $\delta : Q \times \Sigma \to Q$
- Startzustand $q_0 \in Q$
- Menge von Endzuständen $F \subseteq Q$



Definition (Nicht-Deterministischer endlicher Automat)

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

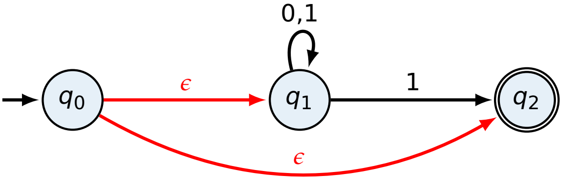
- Q, Σ, q_0, F wie ein DFA
- Übergangsfunktion $\delta : Q \times \Sigma \to P(Q)$



Definition (NFA mit ϵ-Übergängen)

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

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



Übergangsfunktionen

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

$$
\begin{aligned}
\text{DFA} \quad & \delta : Q \times \Sigma \to Q \\
\text{NFA} \quad & \delta : Q \times \Sigma \to P(Q) \\
\epsilon\text{-NFA} \quad & \delta : Q \times (\Sigma \cup \{\epsilon\}) \to P(Q)
\end{aligned}
$$

Satz

DFA, NFA und ϵ-NFA sind gleich mächtig und lassen sich ineinander umwandeln.

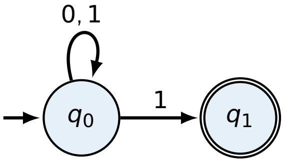
Potenzmengenkonstruktion

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 $F \cap S \neq \emptyset$



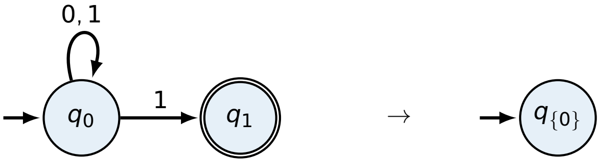
Potenzmengenkonstruktion

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 $F \cap S \neq \emptyset$



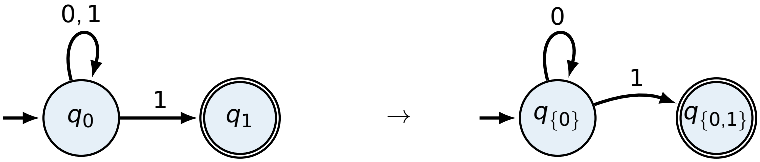
Potenzmengenkonstruktion

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

$$\begin{aligned} \overline{\delta} : \mathcal{P}(Q) \times \Sigma &\rightarrow \mathcal{P}(Q) \\ (S, a) &\mapsto \bigcup_{q \in S} \delta(q, a) \end{aligned}$$

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



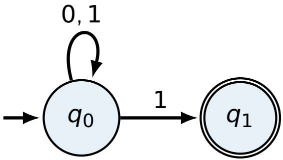
Potenzmengenkonstruktion

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 $F \cap S \neq \emptyset$



$\rightarrow$