3a
Regularität von deterministischen endlichen Automaten und nichtdeterministische endliche Automaten

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 10. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

Wiederholung: DFAs

Definition

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) ist ein 5-Tupel $M = (Z, \Sigma, \delta, z_0, E)$, wobei

- Z ist eine endliche Menge von Zuständen
- Σ ist das (endliche) Eingabealphabet mit $Z \cap \Sigma = \emptyset$
- $\delta : Z \times \Sigma \to Z$ ist die (totale) Überführungsfunktion
- $z_0 \in Z$ ist der Startzustand
- $E \subseteq Z$ ist die Menge der Endzustände.

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis Wir konstruieren für M eine reguläre Grammatik G (mit beiden Sonderregeln), sodass $L(G) = L(M)$.

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis Wir konstruieren für M eine reguläre Grammatik G (mit beiden Sonderregeln), sodass $L(G) = L(M)$.

Sei $G = (V, \Sigma, P, S)$ die Grammatik mit $V := Z$, $S := z_0$ und

$$P := \{z_i \rightarrow az_j \mid \delta(z_i, a) = z_j\}$$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis Wir konstruieren für M eine reguläre Grammatik G (mit beiden Sonderregeln), sodass $L(G) = L(M)$.

Sei $G = (V, \Sigma, P, S)$ die Grammatik mit $V := Z$, $S := z_0$ und

$$P := \{z_i \to az_j \mid \delta(z_i, a) = z_j\} \\ \cup \{z_i \to \varepsilon \mid z_i \in E\}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$P = \{$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$P = \{z_0 \to az_1,$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$P = \{z_0 \to az_1,\ z_0 \to bz_0,$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2, \end{aligned}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2,\ z_1 \to bz_0, \end{aligned}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \rightarrow az_1,\ z_0 \rightarrow bz_0, \\ & z_1 \rightarrow az_2,\ z_1 \rightarrow bz_0, \\ & z_2 \rightarrow az_2, \end{aligned}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{z_0 &\to az_1,\ z_0 \to bz_0, \\ z_1 &\to az_2,\ z_1 \to bz_0, \\ z_2 &\to az_2,\ z_2 \to bz_2, \end{aligned}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1, z_0 \to bz_0, \\ & z_1 \to az_2, z_1 \to bz_0, \\ & z_2 \to az_2, z_2 \to bz_2, \\ & z_2 \to \varepsilon \} \end{aligned}$$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{z_0 &\to az_1,\ z_0 \to bz_0, \\ z_1 &\to az_2,\ z_1 \to bz_0, \\ z_2 &\to az_2,\ z_2 \to bz_2, \\ z_2 &\to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2,\ z_1 \to bz_0, \\ & z_2 \to az_2,\ z_2 \to bz_2, \\ & z_2 \to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

z_0

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$
\begin{array}{lll}
\delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\
\delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2
\end{array}
$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$
\begin{aligned}
P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\
& z_1 \to az_2,\ z_1 \to bz_0, \\
& z_2 \to az_2,\ z_2 \to bz_2, \\
& z_2 \to \varepsilon \}
\end{aligned}
$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{z_0 &\rightarrow az_1, z_0 \rightarrow bz_0, \\ z_1 &\rightarrow az_2, z_1 \rightarrow bz_0, \\ z_2 &\rightarrow az_2, z_2 \rightarrow bz_2, \\ z_2 &\rightarrow \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{z_0 &\to az_1,\ z_0 \to bz_0, \\ z_1 &\to az_2,\ z_1 \to bz_0, \\ z_2 &\to az_2,\ z_2 \to bz_2, \\ z_2 &\to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1, \; z_0 \to bz_0, \\ & z_1 \to az_2, \; z_1 \to bz_0, \\ & z_2 \to az_2, \; z_2 \to bz_2, \\ & z_2 \to \varepsilon \} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0 \Rightarrow abbaz_1$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2,\ z_1 \to bz_0, \\ & z_2 \to az_2,\ z_2 \to bz_2, \\ & z_2 \to \varepsilon \} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0 \Rightarrow abbaz_1 \Rightarrow abbaaz_2$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{z_0 &\to az_1,\ z_0 \to bz_0, \\ z_1 &\to az_2,\ z_1 \to bz_0, \\ z_2 &\to az_2,\ z_2 \to bz_2, \\ z_2 &\to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0 \Rightarrow abbaz_1 \Rightarrow abbaaz_2 \Rightarrow abbaaaz_2$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2,\ z_1 \to bz_0, \\ & z_2 \to az_2,\ z_2 \to bz_2, \\ & z_2 \to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0 \Rightarrow abbaz_1 \Rightarrow abbaaz_2 \Rightarrow abbaaaz_2 \Rightarrow abbaaa$

Beispiel für die Konstruktion einer regulären Grammatik

DFA $M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, z_0, \{z_2\})$ mit

$$\begin{array}{lll} \delta(z_0, a) = z_1 & \delta(z_1, a) = z_2 & \delta(z_2, a) = z_2 \\ \delta(z_0, b) = z_0 & \delta(z_1, b) = z_0 & \delta(z_2, b) = z_2 \end{array}$$

Grammatik dazu: $G = (\{z_0, z_1, z_2\}, \{a, b\}, P, z_0)$ mit

$$\begin{aligned} P = \{ & z_0 \to az_1,\ z_0 \to bz_0, \\ & z_1 \to az_2,\ z_1 \to bz_0, \\ & z_2 \to az_2,\ z_2 \to bz_2, \\ & z_2 \to \varepsilon\} \end{aligned}$$

Ableitung der Eingabe $abbaaa \in L(M)$:

$$z_0 \Rightarrow az_1 \Rightarrow abz_0 \Rightarrow abbz_0 \Rightarrow abbaz_1 \Rightarrow abbaaz_2 \Rightarrow abbaaaz_2 \Rightarrow abbaaa \in L(G)$$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$, d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$$a_1 \cdots a_m \in L(M)$$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$
g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

$a_1 \cdots a_m \in L(G)$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$
g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

$z_0 \Rightarrow_G^* a_1 \cdots a_m$
g.d.w. $a_1 \cdots a_m \in L(G)$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

es gibt $z_1, \ldots, z_m \in Z$ mit $a_1 \cdots a_{i-1} z_{i-1} \Rightarrow_G a_1 \cdots a_i z_i$ für $i \in \{1, \ldots, m\}$
und $a_1 \cdots a_m z_m \Rightarrow_G a_1 \cdots a_m$

g.d.w. $z_0 \Rightarrow_G^* a_1 \cdots a_m$

g.d.w. $a_1 \cdots a_m \in L(G)$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $a_1 \cdots a_{i-1} z_{i-1} \Rightarrow_G a_1 \cdots a_i z_i$ für $i \in \{1, \ldots, m\}$
und $a_1 \cdots a_m z_m \Rightarrow_G a_1 \cdots a_m$

g.d.w. $z_0 \Rightarrow_G^* a_1 \cdots a_m$

g.d.w. $a_1 \cdots a_m \in L(G)$

Die akzeptierte Sprache von DFAs ist regulär

Theorem

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA. Dann ist $L(M)$ regulär.

Beweis (Fortsetzung) Wir zeigen, dass $L(M) = L(G)$,
d.h. $w \in L(M)$ g.d.w. $w \in L(G)$ für ein beliebiges w.

Sei $w = a_1 \cdots a_m$ mit $m \in \mathbb{N}$.

$a_1 \cdots a_m \in L(M)$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $\delta(z_{i-1}, a_i) = z_i$ für $i \in \{1, \ldots, m\}$ und $z_m \in E$

g.d.w. es gibt $z_1, \ldots, z_m \in Z$ mit $a_1 \cdots a_{i-1} z_{i-1} \Rightarrow_G a_1 \cdots a_i z_i$ für $i \in \{1, \ldots, m\}$
und $a_1 \cdots a_m z_m \Rightarrow_G a_1 \cdots a_m$

g.d.w. $z_0 \Rightarrow_G^* a_1 \cdots a_m$

g.d.w. $a_1 \cdots a_m \in L(G)$

Daher gilt $L(M) = L(G)$ und somit ist $L(M)$ regulär. $\square$

Wird jede reguläre Sprache durch einen DFA akzeptiert?

- Der vorherige Beweis konstruiert:
  „für jeden DFA gibt es eine äquivalente reguläre Grammatik“
- Für die andere Richtung wäre notwendig:
  „für jede reguläre Grammatik gibt es einen äquivalenten DFA“

Wird jede reguläre Sprache durch einen DFA akzeptiert?

- Der vorherige Beweis konstruiert:
  „für jeden DFA gibt es eine äquivalente reguläre Grammatik“
- Für die andere Richtung wäre notwendig:
  „für jede reguläre Grammatik gibt es einen äquivalenten DFA“

Problem:

- Produktionen $A \to aA_1$ und $A \to aA_2$ können in Grammatiken vorkommen.
- Die Konstruktion des DFA ist zunächst unklar.

Wird jede reguläre Sprache durch einen DFA akzeptiert?

- Der vorherige Beweis konstruiert:
  „für jeden DFA gibt es eine äquivalente reguläre Grammatik“
- Für die andere Richtung wäre notwendig:
  „für jede reguläre Grammatik gibt es einen äquivalenten DFA“

Problem:

- Produktionen $A \to aA_1$ und $A \to aA_2$ können in Grammatiken vorkommen.
- Die Konstruktion des DFA ist zunächst unklar.

Daher: Der Beweis für die andere Richtung erfolgt auf Umwegen und verwendet nichtdeterministische endliche Automaten (NFAs).

Nichtdeterministische endliche Automaten

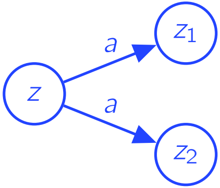
Informelle Kurzfassung:

- Der Zustandswechsel ist nicht eindeutig, sondern nichtdeterministisch in einen von mehreren (oder gar keinen) möglichen Zuständen.
- Der Automat darf sozusagen raten, welchen Nachfolgezustand er wählt.

Nichtdeterministische endliche Automaten

Informelle Kurzfassung:

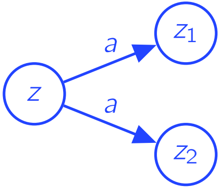
- Der Zustandswechsel ist nicht eindeutig, sondern nichtdeterministisch in einen von mehreren (oder gar keinen) möglichen Zuständen.
- Der Automat darf sozusagen raten, welchen Nachfolgezustand er wählt.
- Im Zustandsgraph erlaubt:



Nichtdeterministische endliche Automaten

Informelle Kurzfassung:

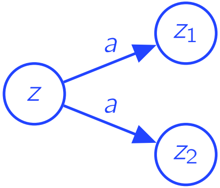
- Der Zustandswechsel ist nicht eindeutig, sondern nichtdeterministisch in einen von mehreren (oder gar keinen) möglichen Zuständen.
- Der Automat darf sozusagen raten, welchen Nachfolgezustand er wählt.
- Im Zustandsgraph erlaubt:



- Technisch:
  - Ein DFA hat $\delta : Z \times \Sigma \to Z$ und einen Startzustand.
  - Ein NFA hat $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ und eine Menge von Startzuständen.

Nichtdeterministische endliche Automaten

Informelle Kurzfassung:

- Der Zustandswechsel ist nicht eindeutig, sondern nichtdeterministisch in einen von mehreren (oder gar keinen) möglichen Zuständen.
- Der Automat darf sozusagen raten, welchen Nachfolgezustand er wählt.
- Im Zustandsgraph erlaubt: 

- Technisch:
  - Ein DFA hat $\delta : Z \times \Sigma \to Z$ und einen Startzustand.
  - Ein NFA hat $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ und eine Menge von Startzuständen.
- NFAs machen manche Konstruktionen einfacher.

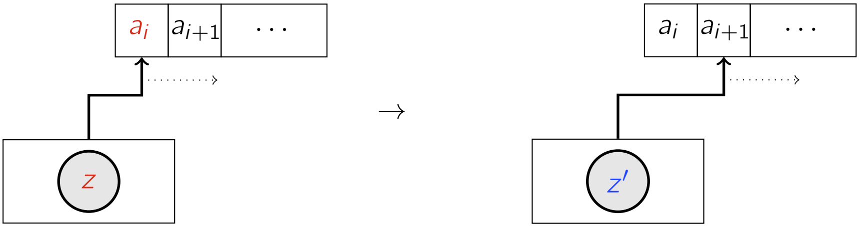
Definition eines NFA

Definition

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $M = (Z, \Sigma, \delta, S, E)$, wobei

- Z ist eine endliche Menge von Zuständen
- Σ ist das (endliche) Eingabealphabet mit $Z \cap \Sigma = \emptyset$
- $\delta : Z \times \Sigma \to \mathcal{P}(Z)$ ist die Überführungsfunktion
- $S \subseteq Z$ ist die Menge der Startzustände
- $E \subseteq Z$ ist die Menge der Endzustände.

Illustration des Zustandsübergangs



$z' \in \delta(z, a_i)$ bedeutet:
Im Zustand z bei Eingabe a_i darf der NFA in z' wechseln.

Beispiel für einen NFA

NFA $M = (\{z_0, z_1, z_2, z_3\}, \{a, b, c\}, \delta, \{z_0, z_3\}, \{z_3\})$ mit

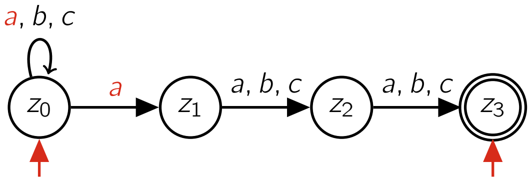
$$
\begin{array}{llll}
\delta(z_0, a) = \{z_0, z_1\} & \delta(z_1, a) = \{z_2\} & \delta(z_2, a) = \{z_3\} & \delta(z_3, a) = \emptyset \\
\delta(z_0, b) = \{z_0\} & \delta(z_1, b) = \{z_2\} & \delta(z_2, b) = \{z_3\} & \delta(z_3, b) = \emptyset \\
\delta(z_0, c) = \{z_0\} & \delta(z_1, c) = \{z_2\} & \delta(z_2, c) = \{z_3\} & \delta(z_3, c) = \emptyset
\end{array}
$$

Beispiel für einen NFA

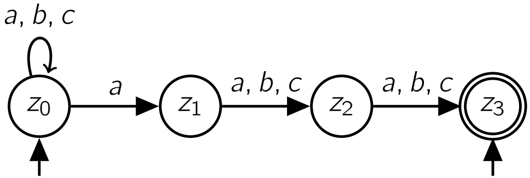
NFA $M = (\{z_0, z_1, z_2, z_3\}, \{a, b, c\}, \delta, \{z_0, z_3\}, \{z_3\})$ mit

$$\begin{array}{llll}
\delta(z_0, a) = \{z_0, z_1\} & \delta(z_1, a) = \{z_2\} & \delta(z_2, a) = \{z_3\} & \delta(z_3, a) = \emptyset \\
\delta(z_0, b) = \{z_0\} & \delta(z_1, b) = \{z_2\} & \delta(z_2, b) = \{z_3\} & \delta(z_3, b) = \emptyset \\
\delta(z_0, c) = \{z_0\} & \delta(z_1, c) = \{z_2\} & \delta(z_2, c) = \{z_3\} & \delta(z_3, c) = \emptyset
\end{array}$$

Zustandsgraph zu M:

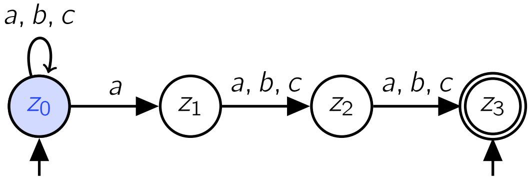


Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

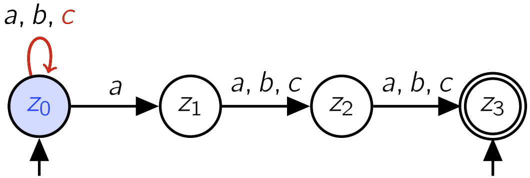
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe *caaba*:

- Starte in z_0.

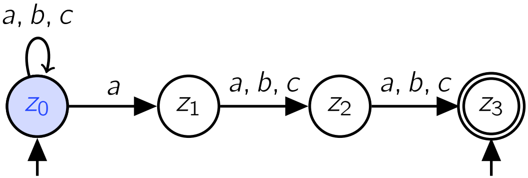
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c

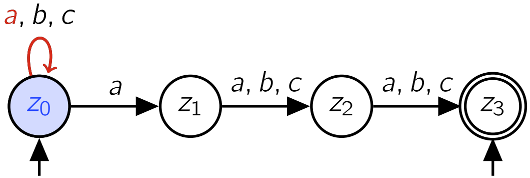
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.

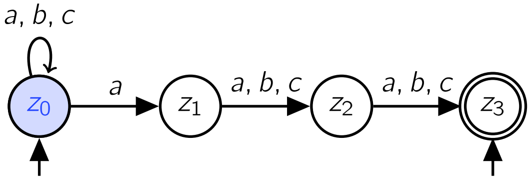
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a

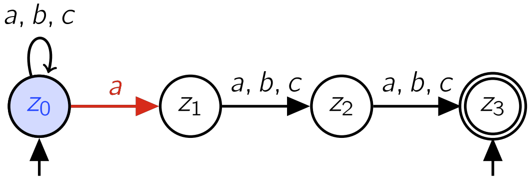
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.

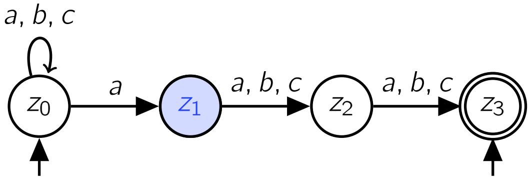
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a

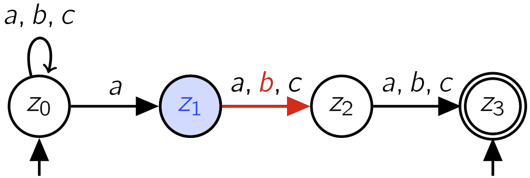
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.

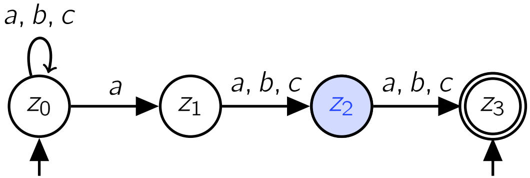
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies b

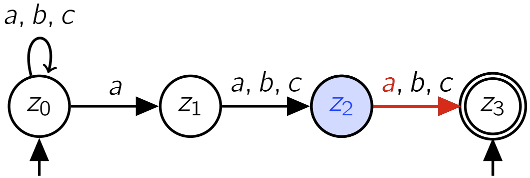
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies b und wechsle in z_2.

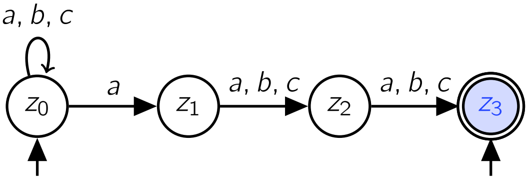
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies b und wechsle in z_2.
- Lies a

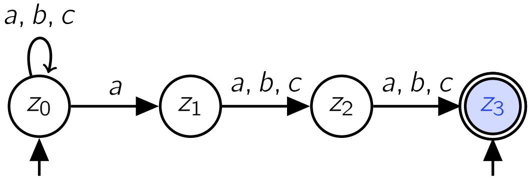
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies b und wechsle in z_2.
- Lies a und wechsle in z_3.

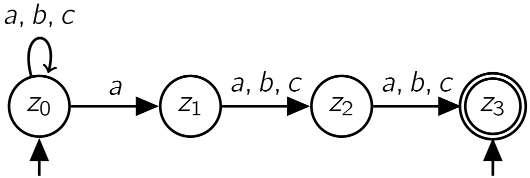
Beispiel für einen akzeptierenden Lauf



Abarbeitung der Eingabe *caaba*:

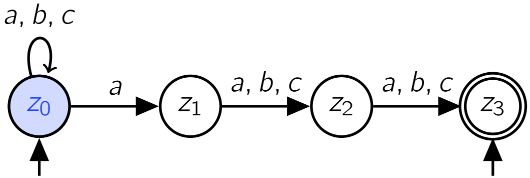
- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies b und wechsle in z_2.
- Lies a und wechsle in z_3.
- Akzeptiere.

Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

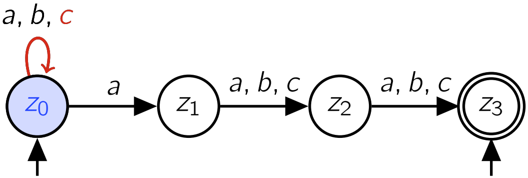
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.

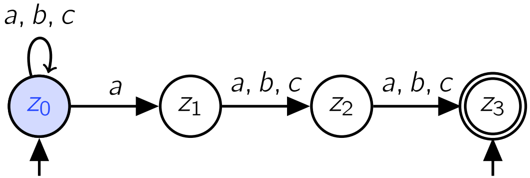
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c

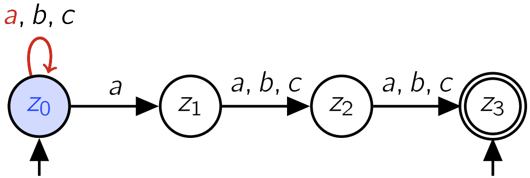
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.

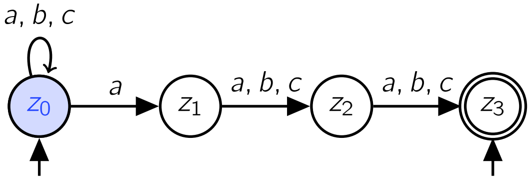
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a

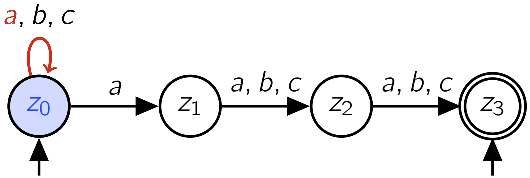
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.

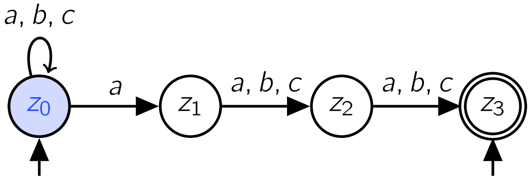
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a

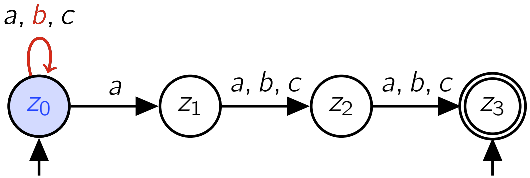
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.

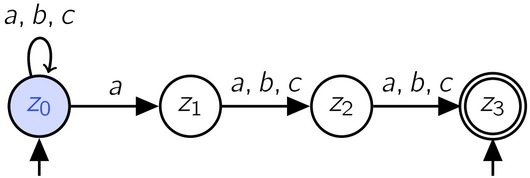
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies b

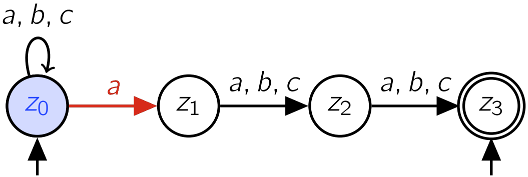
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies b und wechsle in z_0.

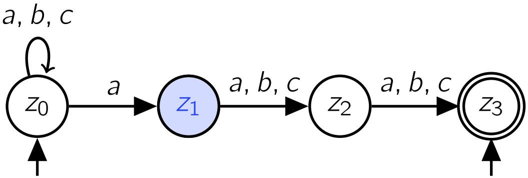
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies b und wechsle in z_0.
- Lies a

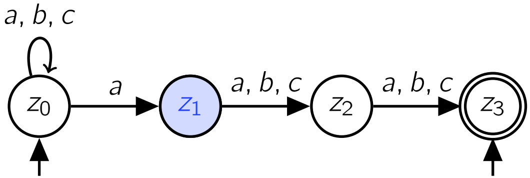
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies b und wechsle in z_0.
- Lies a und wechsle in z_1.

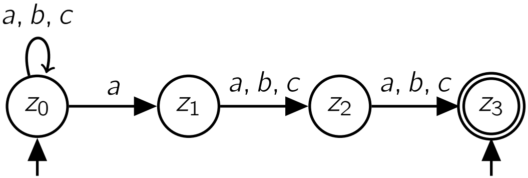
Beispiel für einen verwerfenden Lauf



Abarbeitung der Eingabe $caaba$:

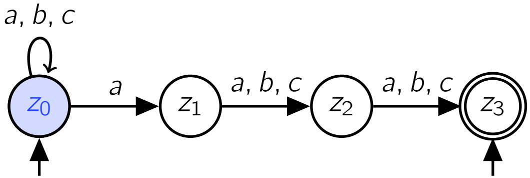
- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies a und wechsle in z_0.
- Lies b und wechsle in z_0.
- Lies a und wechsle in z_1.
- Verwerfe.

Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

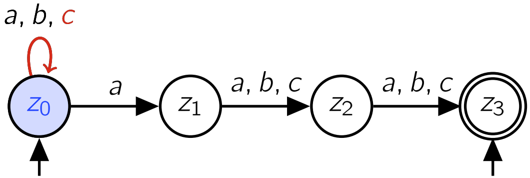
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.

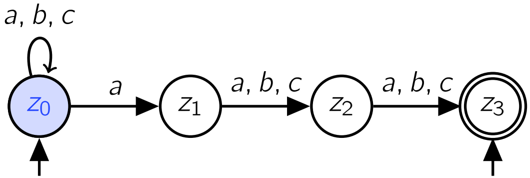
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c

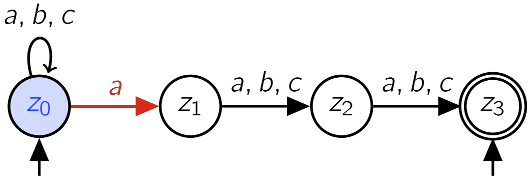
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.

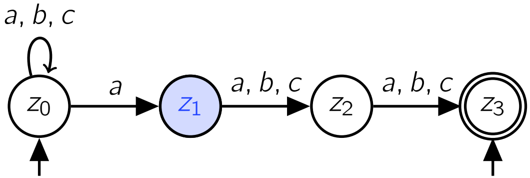
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a

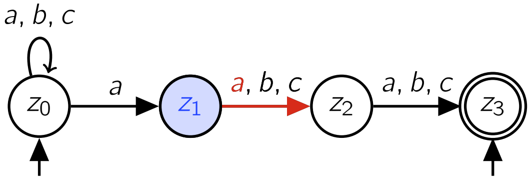
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.

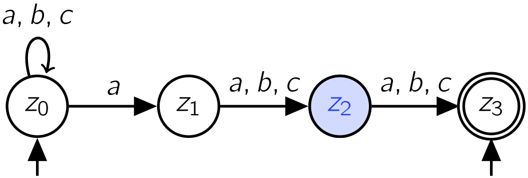
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a

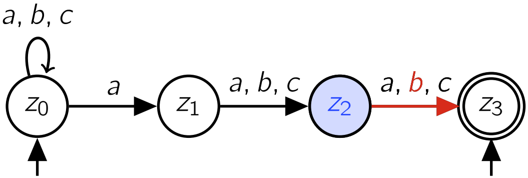
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.

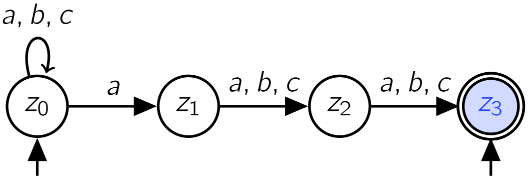
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.
- Lies b

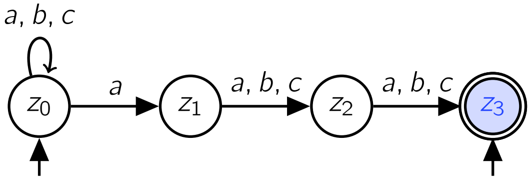
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.
- Lies b und wechsle in z_3.

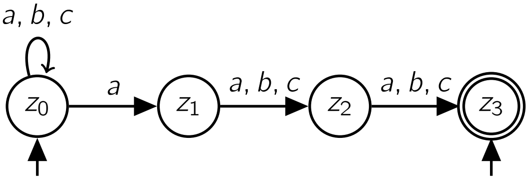
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.
- Lies b und wechsle in z_3.
- Lies a und ...?

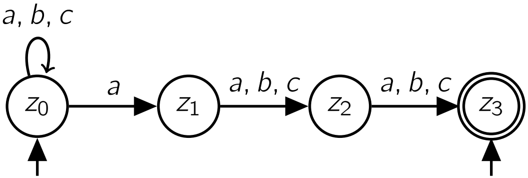
Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.
- Lies b und wechsle in z_3.
- Lies a und ...?

Beispiel für einen steckengebliebenen Lauf



Abarbeitung der Eingabe $caaba$:

- Starte in z_0.
- Lies c und wechsle in z_0.
- Lies a und wechsle in z_1.
- Lies a und wechsle in z_2.
- Lies b und wechsle in z_3.
- Lies a und ...?
- Verwerfe.

Akzeptanz bei NFAs

Ein Wort w wird vom NFA akzeptiert, wenn es mindestens einen Pfad von einem Startzustand zu einem Endzustand entlang w gibt.

Akzeptanz bei NFAs

Ein Wort w wird vom NFA akzeptiert, wenn es mindestens einen Pfad von einem Startzustand zu einem Endzustand entlang w gibt.

Definition

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA.
Wir definieren $\widetilde{\delta} : \mathcal{P}(Z) \times \Sigma^* \to \mathcal{P}(Z)$ rekursiv durch

$$\begin{aligned} \widetilde{\delta}(X, \varepsilon) &:= X \\ \widetilde{\delta}(X, aw) &:= \widetilde{\delta}\Big(\bigcup_{z \in X} \delta(z, a), w\Big) \end{aligned}$$

Akzeptanz bei NFAs

Ein Wort w wird vom NFA akzeptiert, wenn es mindestens einen Pfad von einem Startzustand zu einem Endzustand entlang w gibt.

Definition

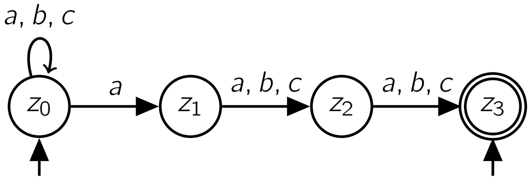
Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA.
Wir definieren $\widetilde{\delta} : \mathcal{P}(Z) \times \Sigma^* \to \mathcal{P}(Z)$ rekursiv durch

$$\begin{aligned}\widetilde{\delta}(X, \varepsilon) &:= X \\ \widetilde{\delta}(X, aw) &:= \widetilde{\delta}\Big(\bigcup_{z \in X} \delta(z, a), w\Big)\end{aligned}$$

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \widetilde{\delta}(S, w) \cap E \neq \emptyset\}$$

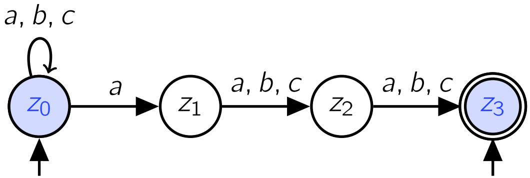
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) =$$

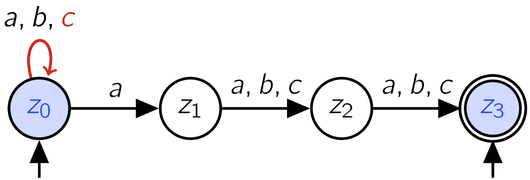
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\bigcup_{z \in \{z_0, z_3\}} \delta(z, c), aaba)$$

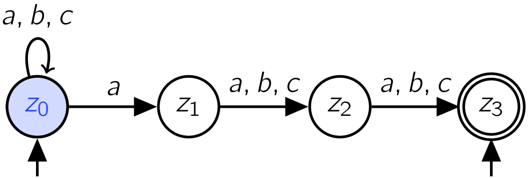
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\delta(z_0, c) \cup \delta(z_3, c), aaba)$$

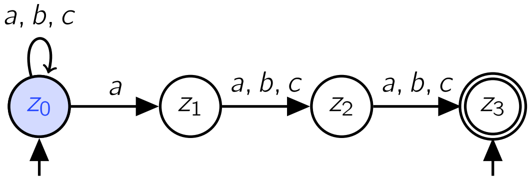
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0\} \cup \emptyset, aaba)$$

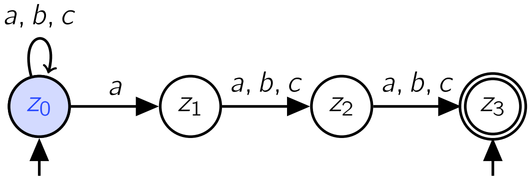
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0\}, aaba)$$

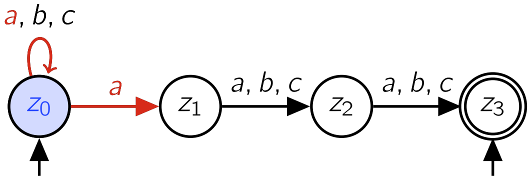
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\bigcup_{z \in \{z_0\}} \delta(z, a), aba)$$

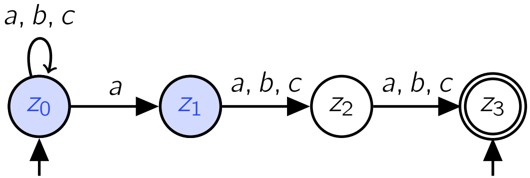
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\delta(z_0, a), aba)$$

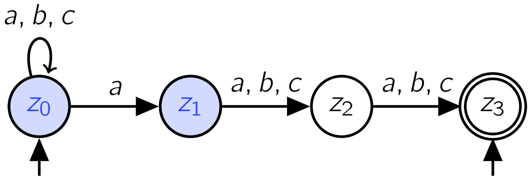
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_1\}, aba)$$

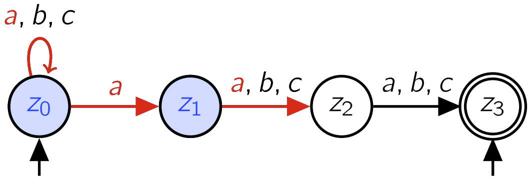
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\bigcup_{z\in\{z_0,z_1\}} \delta(z, a), ba)$$

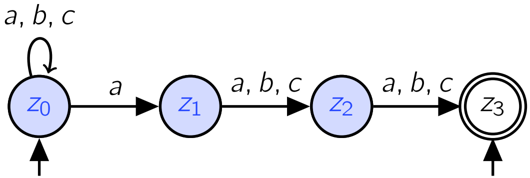
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\delta(z_0, a) \cup \delta(z_1, a), ba)$$

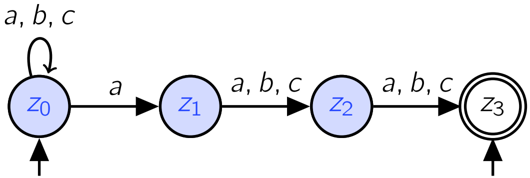
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_1\} \cup \{z_2\}, ba)$$

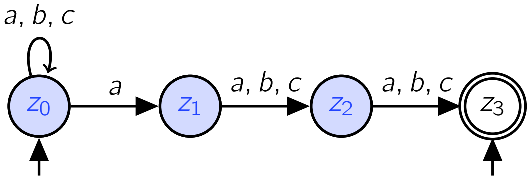
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_1, z_2\}, ba)$$

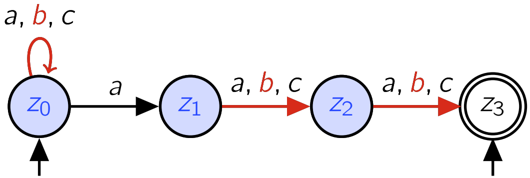
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\bigcup_{z \in \{z_0, z_1, z_2\}} \delta(z, b), a)$$

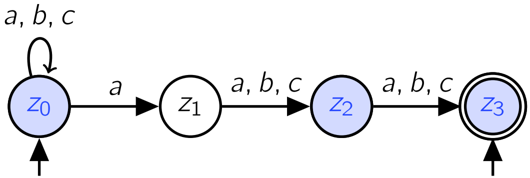
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\delta(z_0, b) \cup \delta(z_1, b) \cup \delta(z_2, b), a)$$

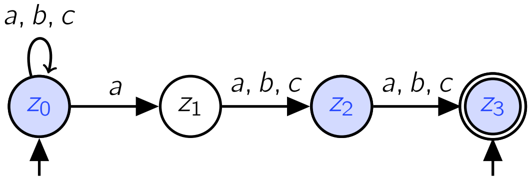
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0\} \cup \{z_2\} \cup \{z_3\}, a)$$

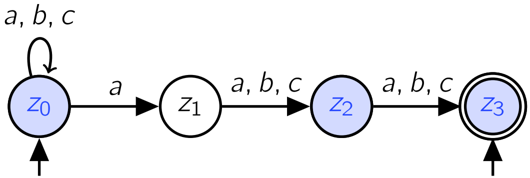
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_2, z_3\}, a)$$

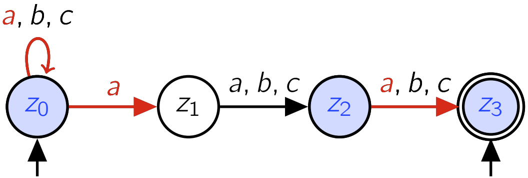
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe *caaba*:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\bigcup_{z\in\{z_0,z_2,z_3\}} \delta(z, a), \varepsilon)$$

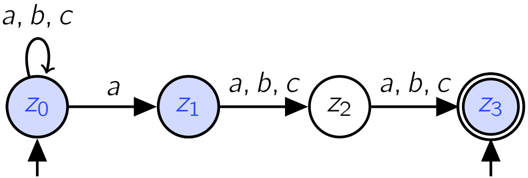
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\delta(z_0, a) \cup \delta(z_2, a) \cup \delta(z_3, a), \varepsilon)$$

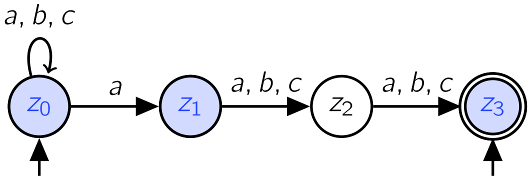
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_1\} \cup \{z_3\} \cup \emptyset, \varepsilon)$$

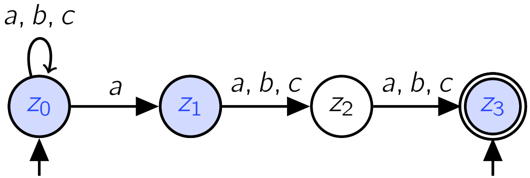
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \widetilde{\delta}(\{z_0, z_1, z_3\}, \varepsilon)$$

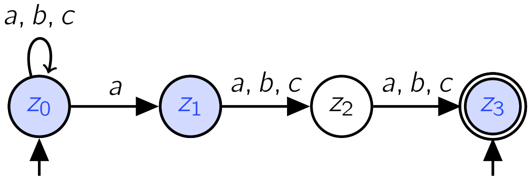
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \{z_0, z_1, z_3\}$$

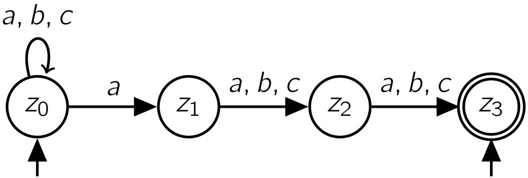
Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $caaba$:

$$\widetilde{\delta}(\{z_0, z_3\}, caaba) = \{z_0, z_1, z_3\} \cap \{z_3\} \neq \emptyset$$

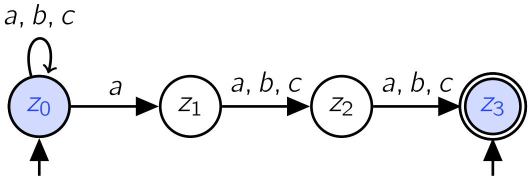
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) =$$

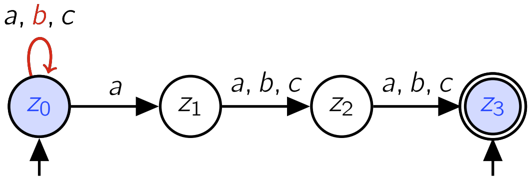
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\bigcup_{z \in \{z_0, z_3\}} \delta(z, b), ab)$$

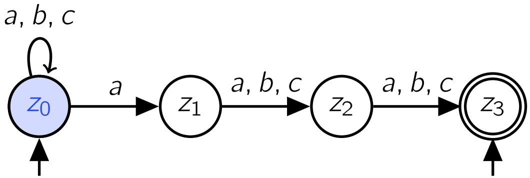
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\delta(z_0, b) \cup \delta(z_3, b), ab)$$

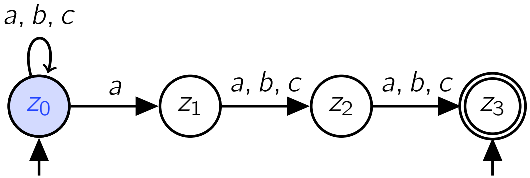
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\{z_0\} \cup \emptyset, ab)$$

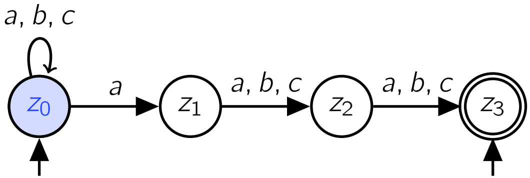
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\{z_0\}, ab)$$

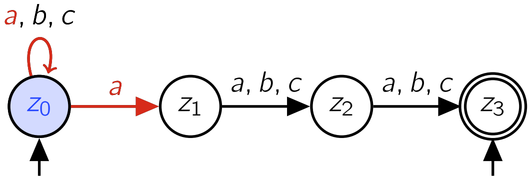
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\bigcup_{z \in \{z_0\}} \delta(z, a), b)$$

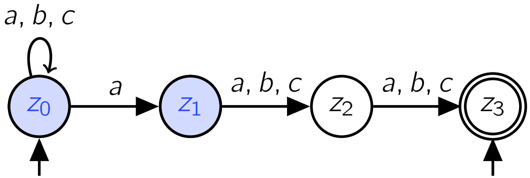
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\delta(z_0, a), b)$$

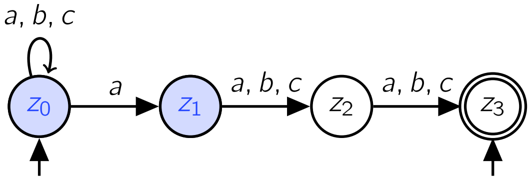
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\{z_0, z_1\}, b)$$

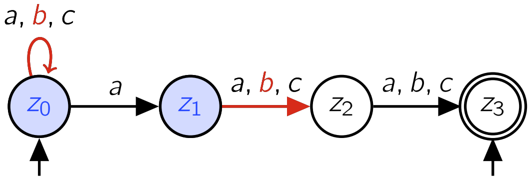
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe $ba\textcolor{red}{b}$:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\bigcup_{z \in \{z_0, z_1\}} \delta(z, \textcolor{red}{b}), \varepsilon)$$

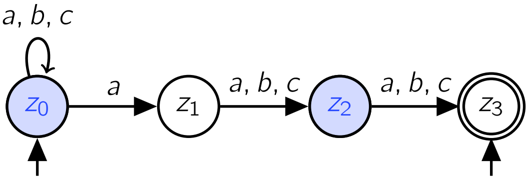
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\delta(z_0, b) \cup \delta(z_1, b), \varepsilon)$$

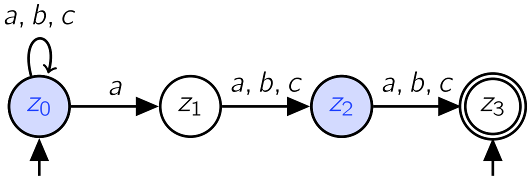
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\{z_0\} \cup \{z_2\}, \varepsilon)$$

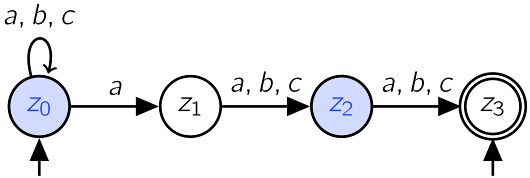
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \widetilde{\delta}(\{z_0, z_2\}, \varepsilon)$$

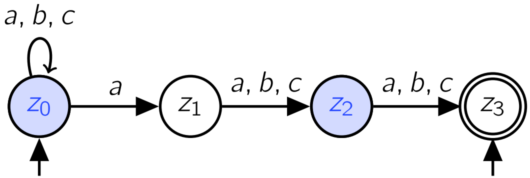
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \{z_0, z_2\}$$

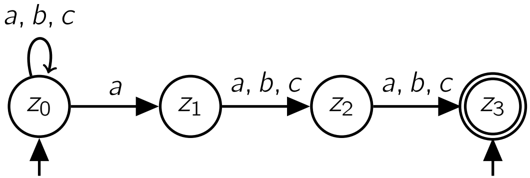
Weiteres Beispiel für die $\widetilde{\delta}$-Funktion



Abarbeitung der Eingabe bab:

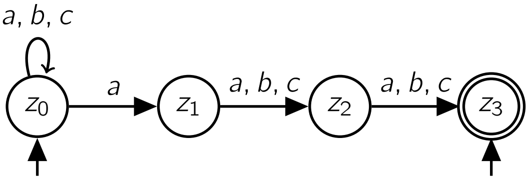
$$\widetilde{\delta}(\{z_0, z_3\}, bab) = \{z_0, z_2\} \cap \{z_3\} = \emptyset$$

Akzeptierte Sprache des Beispiels



Akzeptierte Sprache = ?

Akzeptierte Sprache des Beispiels



Akzeptierte Sprache $= \{\varepsilon\} \cup \{uav \mid u \in \{a, b, c\}^*, v \in \{a, b, c\}^2\}$

Läufe von NFAs

Definition

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA und $w \in \Sigma^*$ ein Wort der Länge n.
Eine Folge von Zuständen $z_0, \ldots, z_n$ mit $z_0 \in S$ und $z_i \in \delta(z_{i-1}, w[i])$ für $i \in \{1, \ldots, n\}$ ist ein Lauf von M für w.

Läufe von NFAs

Definition

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA und $w \in \Sigma^*$ ein Wort der Länge n.
Eine Folge von Zuständen $z_0, \ldots, z_n$ mit $z_0 \in S$ und $z_i \in \delta(z_{i-1}, w[i])$ für $i \in \{1, \ldots, n\}$ ist ein Lauf von M für w.

Für einen Lauf schreiben wir auch

$$z_0 \xrightarrow{w[1]} z_1 \xrightarrow{w[2]} \cdots \xrightarrow{w[n-1]} z_{n-1} \xrightarrow{w[n]} z_n$$

Läufe von NFAs

Definition

Sei $M = (Z, \Sigma, \delta, S, E)$ ein NFA und $w \in \Sigma^*$ ein Wort der Länge n.
Eine Folge von Zuständen $z_0, \dots, z_n$ mit $z_0 \in S$ und $z_i \in \delta(z_{i-1}, w[i])$ für $i \in \{1, \dots, n\}$ ist ein Lauf von M für w.

Für einen Lauf schreiben wir auch

$$z_0 \xrightarrow{w[1]} z_1 \xrightarrow{w[2]} \cdots \xrightarrow{w[n-1]} z_{n-1} \xrightarrow{w[n]} z_n$$

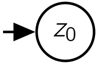
Während es bei DFAs genau einen Lauf pro Wort gibt, kann es bei NFAs mehrere geben (oder gar keine).

Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{a, b\}$, der alle Wörter akzeptiert,
die mit *abaa* beginnen und mit *bab* enden (z.B. *abaabab*, *abaaaabab*, *abaababab*):

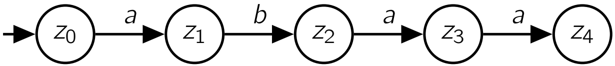
Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{a, b\}$, der alle Wörter akzeptiert,
die mit *abaa* beginnen und mit *bab* enden (z.B. *abaabab*, *abaaaabab*, *abaababab*):



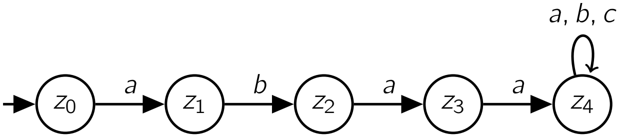
Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{a, b\}$, der alle Wörter akzeptiert,
die mit *abaa* beginnen und mit *bab* enden (z.B. *abaabab*, *abaaaabab*, *abaababab*):



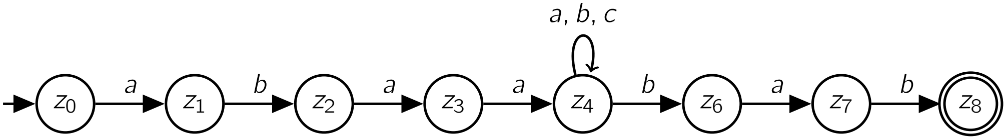
Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{a, b\}$, der alle Wörter akzeptiert, die mit *abaa* beginnen und mit *bab* enden (z.B. *abaabab*, *abaaaabab*, *abaababab*):



Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{a, b\}$, der alle Wörter akzeptiert,
die mit *abaa* beginnen und mit *bab* enden (z.B. *abaabab*, *abaaaabab*, *abaababab*):



Weiteres Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über
$\{+, -, ., 0, \ldots, 9\}$,
der alle Wörter akzeptiert,
die Gleitkommazahlen
darstellen
(z.B. $+27$, -3.14, $.666$):

Weiteres Beispiel für die Konstruktion eines NFA

Konstruiere einen NFA über $\{+, -, ., 0, \dots, 9\}$, der alle Wörter akzeptiert, die Gleitkommazahlen darstellen (z.B. $+27$, -3.14, $.666$):