



Theoretische Grundlagen der Informatik

Vorlesung am 05.11.2020

Torsten Ueckerdt | 5. November 2020

Kontextfreie Grammatiken

Gegeben ist eine Regel $l \rightarrow r$.

- Wenn in einem Wort w das Zeichen l vorhanden ist, darf l in w durch r ersetzt werden.
- Wir schreiben $w \rightarrow z$, wenn w durch Anwendung *einer* Ableitungsregel in z umgewandelt werden kann.
- Wir schreiben $w \xrightarrow{*} z$, wenn w durch Anwendung einer, keiner oder *mehrerer* Ableitungsregeln in z umgewandelt werden kann.

Die von einer Grammatik $G = (\Sigma, V, S, R)$ erzeugte Sprache $L(G)$ ist die Menge aller Wörter $z \in \Sigma^*$ für die gilt $S \xrightarrow{*} z$.

Beispiel Kontextfreie Grammatiken

Welche Grammatik erzeugt die Sprache über dem Alphabet $\Sigma = \{0, 1\}$ deren vorletztes Zeichen 0 ist?

Beispiel Kontextfreie Grammatiken

Welche Grammatik erzeugt die Sprache über dem Alphabet $\Sigma = \{0, 1\}$ deren vorletztes Zeichen 0 ist?

Grammatik $G = (\Sigma, V, S, R)$ mit

$$\Sigma := \{0, 1\}$$

$$V := \{S, A, B\}$$

$$R := \{S \rightarrow A0|A1, \quad A \rightarrow B0, \quad B \rightarrow \varepsilon|B0|B1\}$$

Ableitung:

$$S \rightarrow A0|A1 \rightarrow B00|B01 \rightarrow \dots$$

Kontextfreie Grammatiken

In TGI werden wir Grammatiken kennenlernen, deren Produktionen *mächtiger* sind, als die von kontextfreien Grammatiken.

Wir werden auch Sprachen kennenlernen, die nicht von Grammatiken erzeugt werden können.

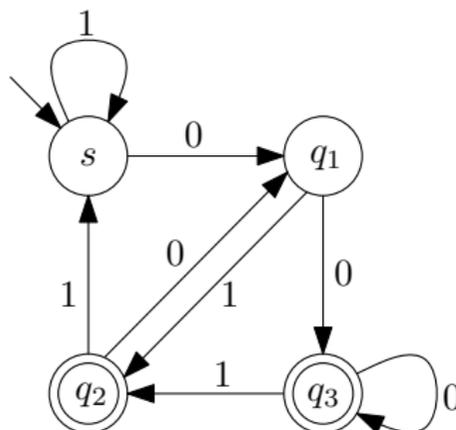
Endliche Automaten und Reguläre Sprachen

$$\Sigma = \{0, 1\}$$

$$L' = \{w0z \mid w \in \Sigma^*, z \in \Sigma\}$$

Sprache aller Wörter über $\Sigma = \{0, 1\}$,
deren vorletztes Zeichen eine 0 ist.

$L' = (\{0\} \cup \{1\})^* \cdot \{0\} \cdot (\{0\} \cup \{1\})$
ist regulär.



Ein DEA der L' erkennt.

- Gibt es zu jeder regulären Sprache einen endlichen Automaten, der diese erkennt?
- Falls ja: Wie kann man diesen konstruieren?
- Welche Sprachen werden durch endliche Automaten erkannt?

Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

Erinnerung:

Eine Sprache $L \subseteq \Sigma^*$ heißt **regulär**, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

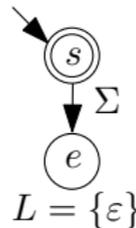
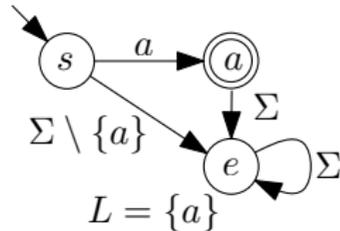
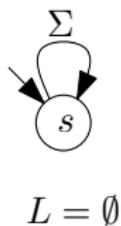
- Verankerung:
 - $L = \{a\}$ mit $a \in \Sigma$ oder
 - $L = \{\varepsilon\}$ oder
 - $L = \emptyset$
- Induktion: Es gibt reguläre Sprachen L_1, L_2 sodass
 - $L = L_1 \cdot L_2$ oder
 - $L = L_1 \cup L_2$ oder
 - $L = L_1^*$

Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

- Induktion über $n =$ Anzahl der Schritte in induktiver Definition von L
- **Induktionsanfang** $n = 1$:



- **Induktionsannahme:** Für alle L , die durch weniger als n Schritte induktiv definiert sind, existiert akzeptierender DEA

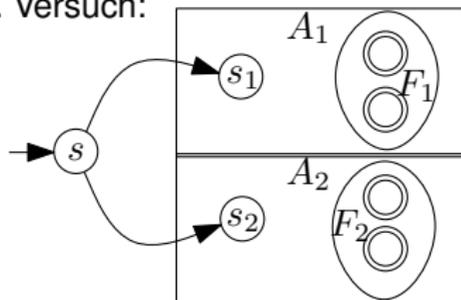
Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

■ Induktionsschluss:

- Für $n > 1$ gilt $L = L_1 \cup L_2$ oder $L = L_1 \cdot L_2$ oder $L = L_1^*$, mit L_1, L_2 regulär.
- **Fall 1:** $L = L_1 \cup L_2$
- Sei A_1 DEA der L_1 akzeptiert und A_2 DEA der L_2 akzeptiert
- Idee: „Baue“ aus A_1 und A_2 Automaten der L akzeptiert
- 1. Versuch:



Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

■ Induktionsschluss:

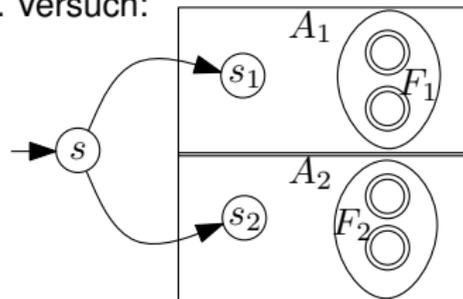
■ Für $n > 1$ gilt $L = L_1 \cup L_2$ oder $L = L_1 \cdot L_2$ oder $L = L_1^*$, mit L_1, L_2 regulär.

■ **Fall 1:** $L = L_1 \cup L_2$

■ Sei A_1 DEA der L_1 akzeptiert und A_2 DEA der L_2 akzeptiert

■ Idee: „Baue“ aus A_1 und A_2 Automaten der L akzeptiert

■ 1. Versuch:



■ Brauche Übergang ohne Lesen eines Zeichens mit **Wahlmöglichkeit**

■ DEA's erlauben das nicht

■ **Rest des Beweises: später!**

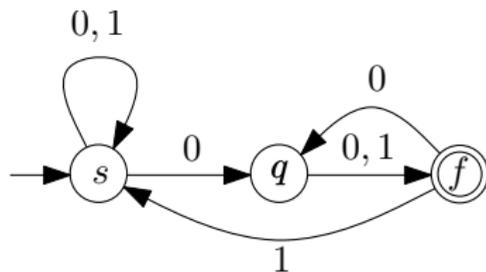
Nichtdeterministische endliche Automaten

- Ein **nichtdeterministischer endlicher Automat** (NEA) besteht aus:
 - Q , einer endlichen Zustandsmenge;
 - s , einem Startzustand;
 - Σ , einem endlichen Alphabet;
 - F , einer Menge von Endzuständen;
 - δ , einer Übergangsfunktion
 $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$
- Interpretation von δ :
 - Bei Abarbeitung eines Symbols a von Zustand q kann der Automat sich **nichtdeterministisch „aussuchen“** in welchen Zustand aus der Teilmenge $\delta(q, a)$ er übergeht
 - Auch möglich: Sprünge spontan ohne Lesen eines Zeichens aus Σ in neuen Zustand aus Teilmenge $\delta(q, \varepsilon)$ (ε -Übergang)

z.B.: $\delta(q, a) = \{q_1, q_3, q_{12}\}; \quad \delta(q, \varepsilon) = \{q_3, q_7\}; \quad \delta(q, b) = \{q_1\}$

Beispiele für NEA's

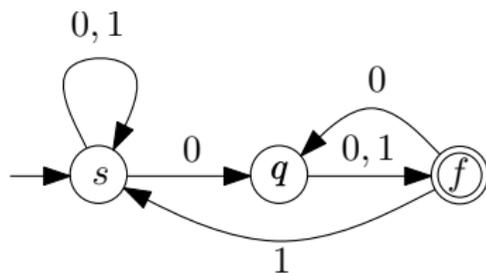
Beispiel 1:



- $\delta(s, 0) = \{s, q\}$
- beim Lesen von 0 im Zustand s kann NEA zu Zustand q oder s wechseln

Beispiele für NEA's

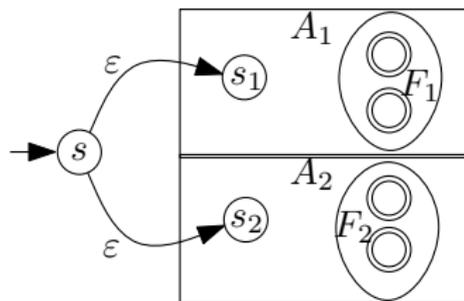
Beispiel 1:



- $\delta(s, 0) = \{s, q\}$
 - beim Lesen von 0 im Zustand s kann NEA zu Zustand q oder s wechseln
 - $w = 10$ wird nicht akzeptiert
 - $w = 01$ wird akzeptiert
 - Sprache aller Wörter über $\{0, 1\}$, deren vorletztes Zeichen 0 ist
- Ein NEA **akzeptiert** ein Wort $w \in \Sigma^*$, wenn es eine Folge von Übergängen gibt, so dass er bei Eingabe von w in einen Endzustand gelangt, d.h. bei Eingabe von w ein **Endzustand erreichbar** ist.

Beispiele für NEA's

Beispiel 2:



- $\delta(s, \varepsilon) = \{s_1, s_2\}$
- mit ε -Übergang kann der NEA ohne Lesen eines Zeichens zu s_1 oder s_2 wechseln
- Automat akzeptiert die Sprache $L(A_1) \cup L(A_2)$

- Ein NEA **akzeptiert** ein Wort $w \in \Sigma^*$, wenn es eine Folge von Übergängen gibt, so dass er bei Eingabe von w in einen Endzustand gelangt, d.h. bei Eingabe von w ein **Endzustand erreichbar** ist.

Äquivalenz von NEA's und DEA's

Definition.

Zwei endliche Automaten, die dieselbe Sprache akzeptieren, heißen *äquivalent*.

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.

Äquivalenz von NEA's und DEA's

Definition.

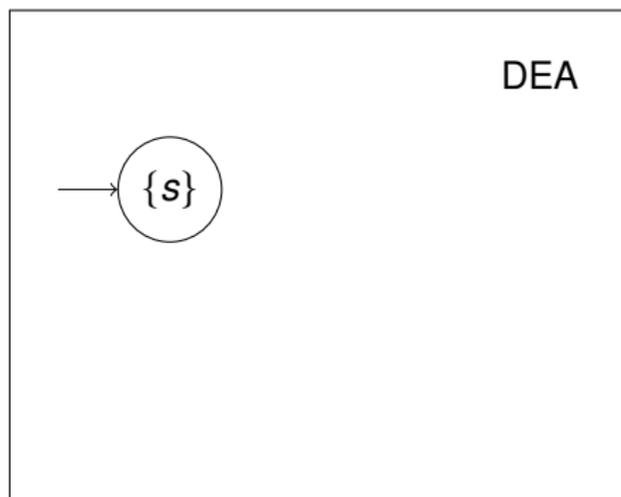
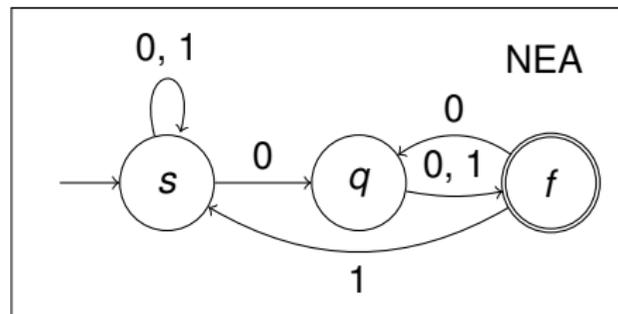
Zwei endliche Automaten, die dieselbe Sprache akzeptieren, heißen *äquivalent*.

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.

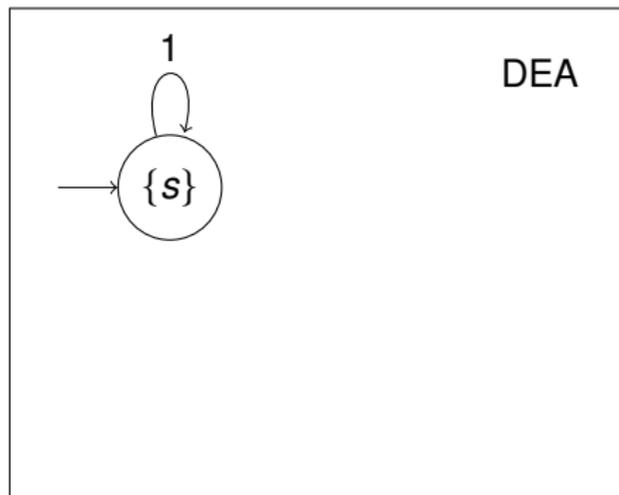
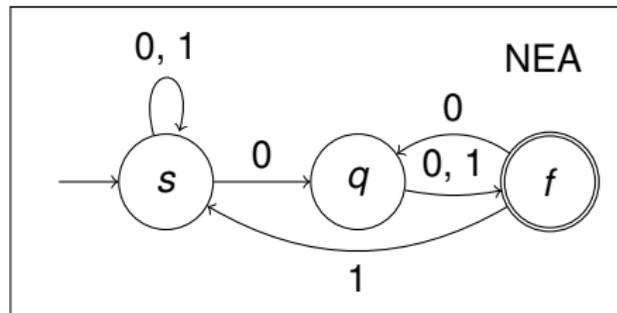
Beweis: durch [Potenzmengenkonstruktion](#)

Beispiel Potenzmengenkonstruktion



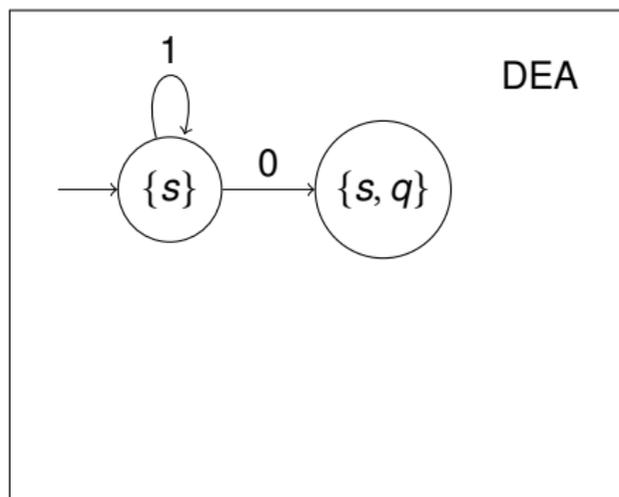
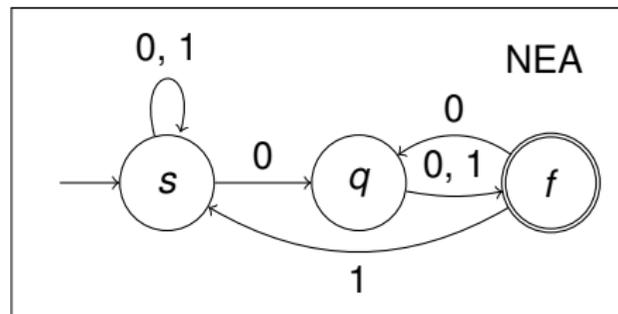
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



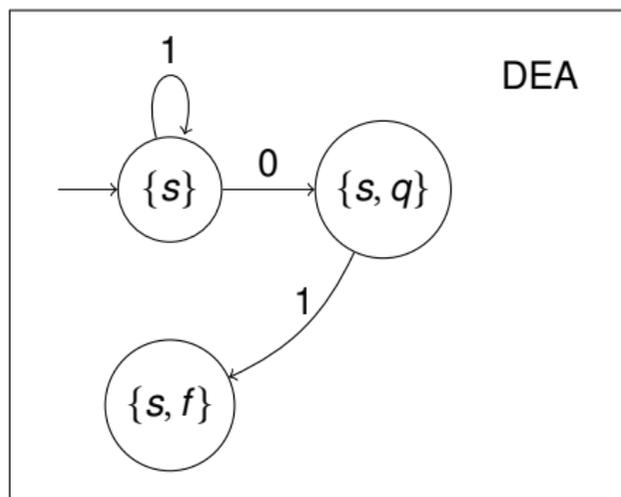
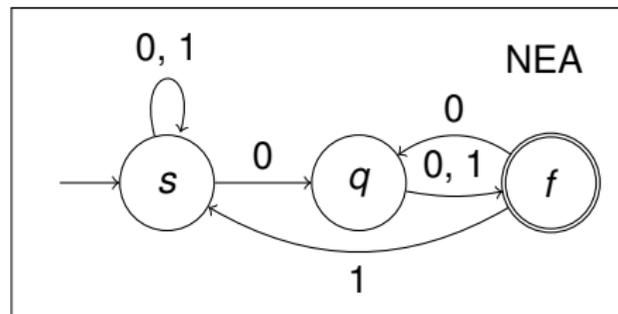
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



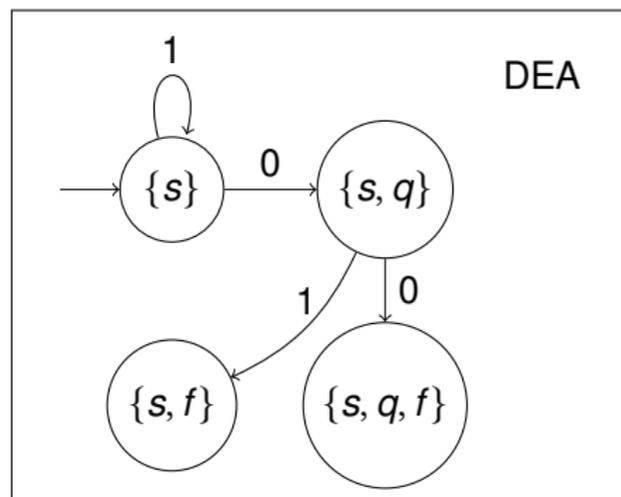
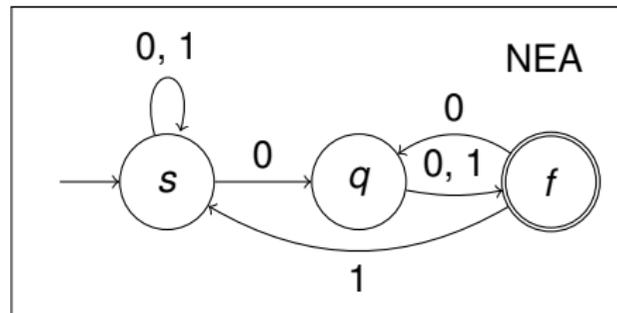
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



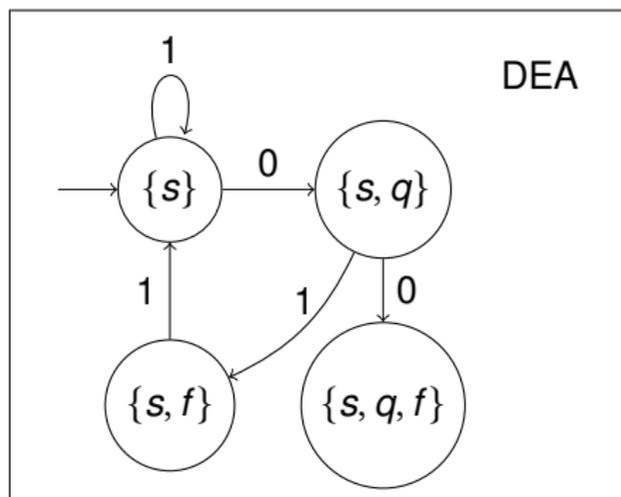
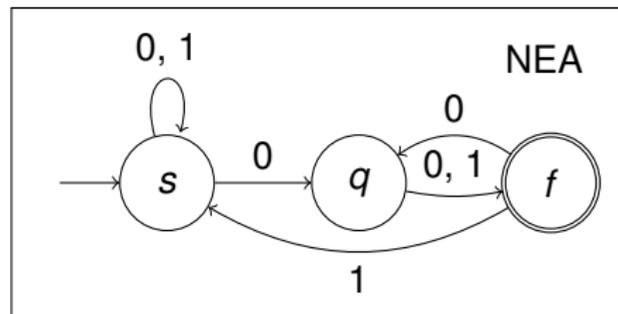
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



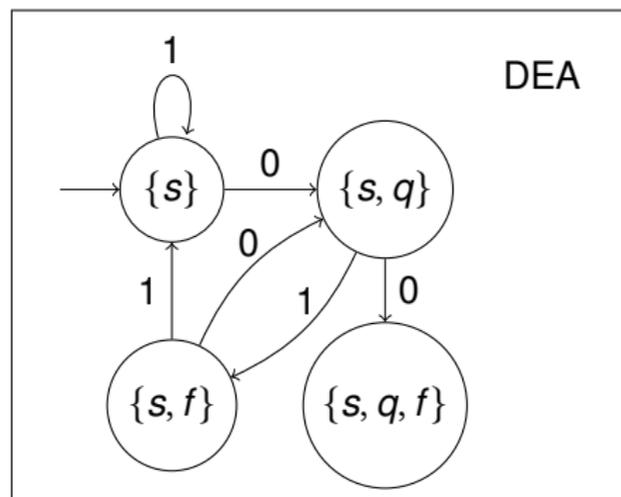
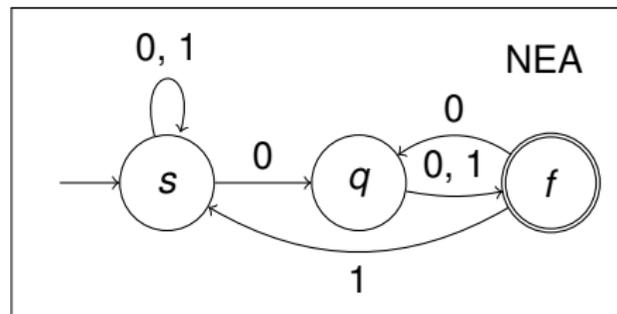
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



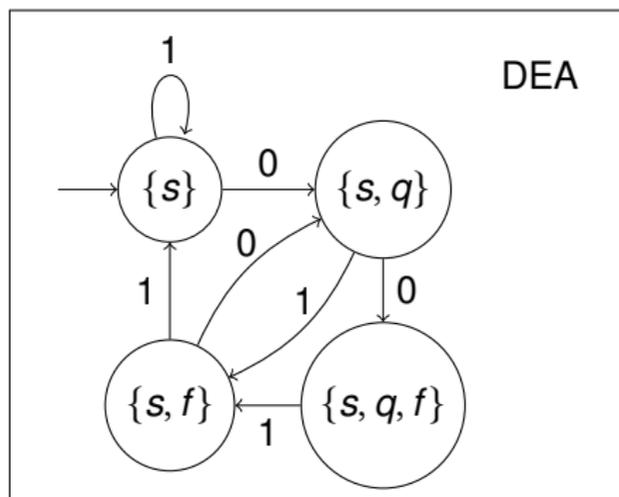
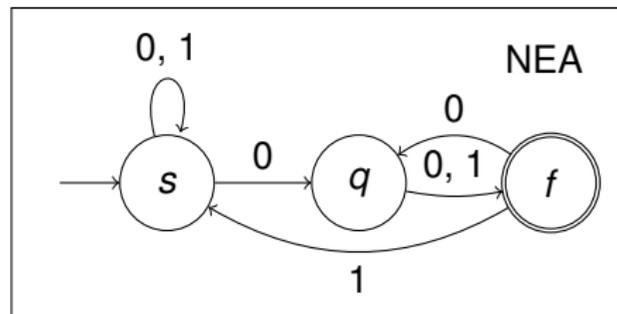
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



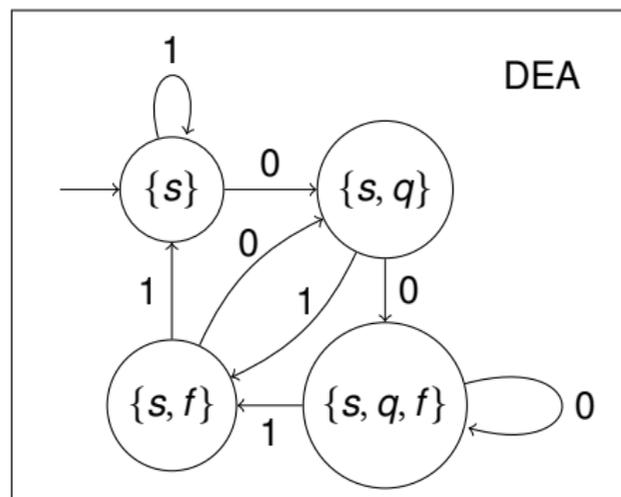
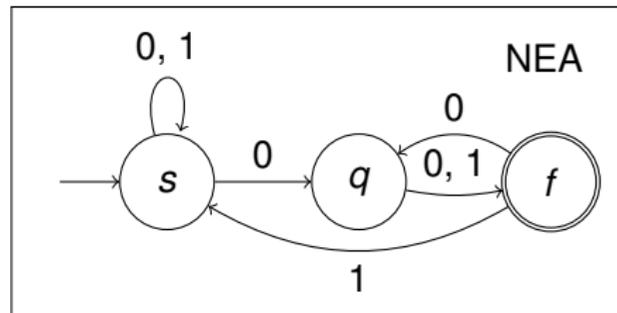
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



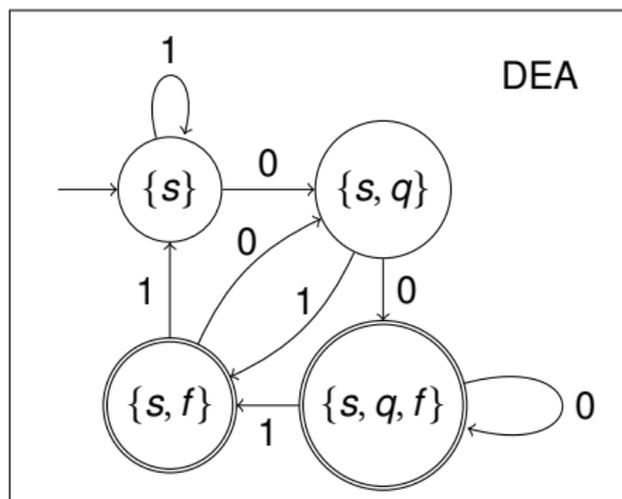
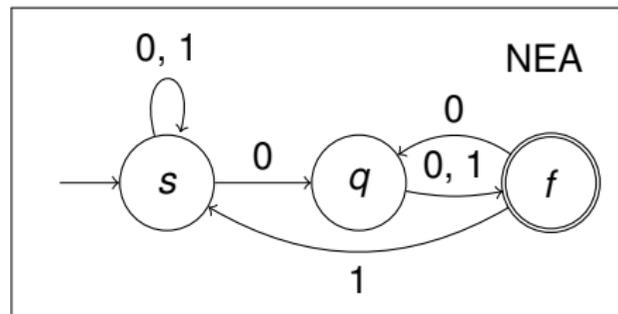
- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Beispiel Potenzmengenkonstruktion



- Zustände des DEA \leftrightarrow Mengen von Zuständen des NEA
- Übergänge des DEA \leftrightarrow Mengen von Übergängen des NEA

Erweiterung von δ

Definition.

Für einen Zustand $q \in Q$ ist der ε -**Abschluss** $E(q)$ wie folgt definiert:

$$E(q) := \{p \in Q \mid p \text{ ist von } q \text{ durch } \varepsilon\text{-Übergänge erreichbar}\}$$

Beachte, dass gilt:

- $E(q) \subseteq Q$, $E(q) \in 2^Q$
- $q \in E(q)$, d.h. die Folge von ε -Übergängen darf auch leer sein

$E(q_1)$



Erweiterung $\bar{\delta}$ von δ

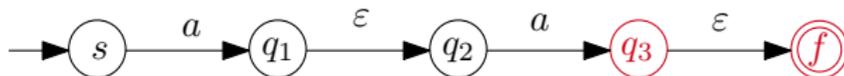
Idee: Berücksichtige ε -Übergänge bei der Übergangsfunktion

- Beim Lesen eines einzelnen Zeichens

$$\bar{\delta}: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$$

$$\bar{\delta}(q, a) = \begin{cases} E(q) & \text{falls } a = \varepsilon \\ \bigcup_{p \in E(q)} \left(\bigcup_{r \in \delta(p, a)} E(r) \right) & \text{für } a \in \Sigma \end{cases}$$

$$\bar{\delta}(q_1, a)$$



Erweiterung $\bar{\delta}$ von δ

Idee: Berücksichtige ε -Übergänge bei der Übergangsfunktion

- Erweiterung auf Mengen von Zuständen :

$$\bar{\delta}: 2^Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$$

$$\bar{\delta}(P, a) = \begin{cases} \bigcup_{p \in P} E(p) & \text{falls } a = \varepsilon \\ \bigcup_{p \in P} \bar{\delta}(p, a) & \text{für } a \in \Sigma \end{cases}$$

$$\bar{\delta}(\{s, q_1\}, a)$$



Erweiterung $\bar{\delta}$ von δ

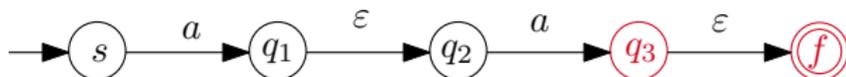
Idee: Berücksichtige ε -Übergänge bei der Übergangsfunktion

- Induktive Erweiterung auf Wörter:

$$\bar{\delta}: Q \times \Sigma^* \rightarrow 2^Q$$

$$\bar{\delta}(q, w) = \begin{cases} E(q) & \text{falls } w = \varepsilon \\ \bar{\delta}(q, w) & \text{falls } w = a \in \Sigma \\ \bigcup_{p \in \bar{\delta}(q, v)} \bar{\delta}(p, a) & \text{falls } w = va, a \in \Sigma, |v| > 0 \end{cases}$$

$$\bar{\delta}(s, w = aa)$$



- Es gilt: $w \in L(\mathcal{A}) \Leftrightarrow \bar{\delta}(s, w) \cap F \neq \emptyset !$

Erweiterung $\bar{\delta}$ von δ

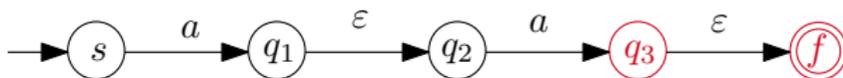
Idee: Berücksichtige ε -Übergänge bei der Übergangsfunktion

- Analog für Mengen von Zuständen:

$$\bar{\delta}: 2^Q \times \Sigma^* \rightarrow 2^Q$$

$$\bar{\delta}(P, w) = \bigcup_{p \in P} \bar{\delta}(p, w)$$

$$\bar{\delta}(\{s, q_1\}, w = aa)$$



Äquivalenz von NEA's und DEA's

Definition.

Zwei endliche Automaten, die dieselbe Sprache akzeptieren, heißen *äquivalent*.

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.

Äquivalenz von NEA's und DEA's

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten \mathcal{A} gibt es einen äquivalenten deterministischen endlichen Automaten.

Beweisidee:

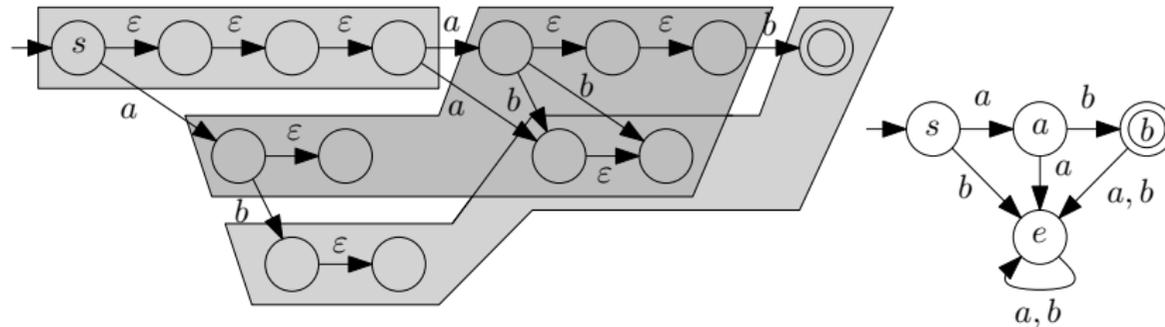
- Falls \mathcal{A} Eingabe w akzeptiert, so gibt es eine Abarbeitung, die in einem Endzustand endet
- Falls ein Zustand q bei der Eingabe von w erreichbar ist, so sind das auch alle Zustände in $E(q)$
- „Simuliere“ \mathcal{A} durch einen DEA
- Zustände des DEA sind Mengen von Zuständen von \mathcal{A}
- Endzustände des DEA sind alle Mengen, die Endzustände von \mathcal{A} enthalten

Äquivalenz von NEA's und DEA's

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten \mathcal{A} gibt es einen äquivalenten deterministischen endlichen Automaten.

Beispiel: Mögliche Abarbeitungen von $w = ab$ in einem NEA mit $ab \in L$, $a \notin L$ und äquivalenter DEA



Äquivalenz von NEA's und DEA's

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten \mathcal{A} gibt es einen äquivalenten deterministischen endlichen Automaten.

Potenzmengenkonstruktion: Gegeben sei ein NEA $\mathcal{A} := (Q, \Sigma, \delta, s, F)$. Wir konstruieren daraus einen DEA $\tilde{\mathcal{A}} := (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$:

- $\tilde{Q} = 2^Q$, d.h. die Zustände des DEA sind Mengen von Zuständen des NEA.
- $\tilde{\delta}: \tilde{Q} \times \Sigma \rightarrow \tilde{Q}$ mit $\tilde{\delta}(\tilde{q}, a) = \bar{\delta}(\tilde{q}, a)$ für $a \in \Sigma$. Es ist also $\tilde{q} \subseteq Q$ und jeder Zustand wird mit seinem ε -Abschluss im NEA identifiziert.
- $\tilde{s} := E(s)$
- $\tilde{F} := \{\tilde{q} \in \tilde{Q} \mid \tilde{q} \cap F \neq \emptyset\}$

Äquivalenz von NEA's und DEA's

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten \mathcal{A} gibt es einen äquivalenten deterministischen endlichen Automaten.

- Zeige per Induktion über $n = |w|$, dass für alle $w \in \Sigma^*$ gilt:

$$\tilde{\delta}(\tilde{s}, w) = \bar{\delta}(s, w)$$

- Dann gilt:

$$\begin{aligned} w \in L(\tilde{\mathcal{A}}) &\Leftrightarrow \tilde{\delta}(\tilde{s}, w) \in \tilde{F} \Leftrightarrow \tilde{\delta}(\tilde{s}, w) \cap F \neq \emptyset \\ &\Leftrightarrow \bar{\delta}(s, w) \cap F \neq \emptyset \Leftrightarrow w \in L(\mathcal{A}) \end{aligned}$$

- **Induktionsanfang** $n = 0$, d.h. $w = \varepsilon$:

$$\tilde{\delta}(\tilde{s}, \varepsilon) = \bar{\delta}(E(s), \varepsilon) = \cup_{p \in E(s)} E(p) = E(s) = \bar{\delta}(s, \varepsilon)$$

Äquivalenz von NEA's und DEA's

Satz. NEA \Rightarrow DEA

Zu jedem nichtdeterministischen endlichen Automaten \mathcal{A} gibt es einen äquivalenten deterministischen endlichen Automaten.

- **Induktionsannahme:** Für alle Wörter w' mit $|w'| \leq n$ gilt:

$$\tilde{\delta}(\tilde{s}, w') = \bar{\delta}(s, w')$$

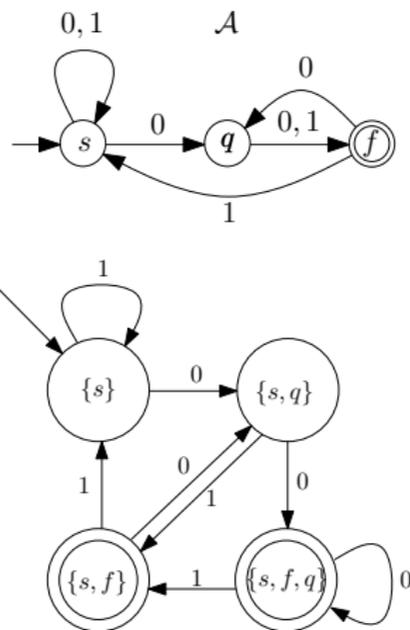
- **Induktionsschluss** $|w| = n + 1$:

Sei $w = w'a$ mit $|w'| = n$ und $a \in \Sigma$. Dann gilt:

$$\begin{aligned} \tilde{\delta}(\tilde{s}, w) &= \tilde{\delta}(\tilde{\delta}(\tilde{s}, w'), a) \\ &\stackrel{(IV)}{=} \tilde{\delta}(\bar{\delta}(s, w'), a) \\ &\stackrel{\text{Def } \tilde{\delta}}{=} \bar{\delta}(\bar{\delta}(s, w'), a) = \bigcup_{p \in \bar{\delta}(s, w')} \bar{\delta}(p, a) = \bar{\delta}(s, w) \end{aligned}$$

Beispiel Potenzmengenkonstruktion

- $L = (0 \cup 1)^* 0(0 \cup 1)$
- NEA \mathcal{A} akzeptiert L
- Konstruiere DEA für L :
 - $\tilde{s} = E(s) = \{s\}$
 - $\tilde{\delta}(\{s\}, 0) = \{s, q\}$
 - $\tilde{\delta}(\{s\}, 1) = \{s\}$
 - $\tilde{\delta}(\{s, q\}, 0) = \{s, q, f\}$
 - $\tilde{\delta}(\{s, q\}, 1) = \{s, f\}$
 - $\tilde{\delta}(\{s, q, f\}, 0) = \{s, q, f\}$
 - $\tilde{\delta}(\{s, q, f\}, 1) = \{s, f\}$
 - $\tilde{\delta}(\{s, f\}, 0) = \{s, q\}$
 - $\tilde{\delta}(\{s, f\}, 1) = \{s\}$
 - Alle anderen Zustände können gestrichen werden



Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

Erinnerung:

Eine Sprache $L \subseteq \Sigma^*$ heißt **regulär**, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

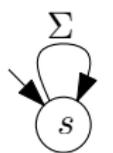
- Verankerung:
 - $L = \{a\}$ mit $a \in \Sigma$ oder
 - $L = \{\varepsilon\}$ oder
 - $L = \emptyset$
- Induktion: Es gibt reguläre Sprachen L_1, L_2 sodass
 - $L = L_1 \cdot L_2$ oder
 - $L = L_1 \cup L_2$ oder
 - $L = L_1^*$

Reguläre Sprache \Rightarrow Endlicher Automat

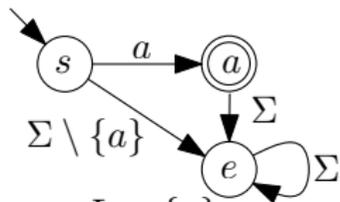
Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

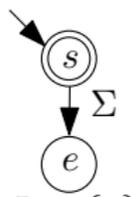
- Induktion über $n =$ Anzahl der Schritte in induktiver Definition von L
- **Induktionsanfang** $n = 1$:



$L = \emptyset$



$L = \{a\}$



$L = \{\varepsilon\}$

- **Induktionsannahme:** Für alle L , die durch weniger als n Schritte induktiv definiert sind, existiert akzeptierender DEA

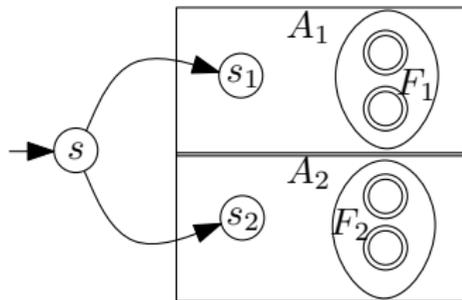
Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

■ Induktionsschluss:

- Für $n > 1$ gilt $L = L_1 \cup L_2$ oder $L = L_1 \cdot L_2$ oder $L = L_1^*$, mit L_1, L_2 regulär.
- **Fall 1:** $L = L_1 \cup L_2$
- Sei A_1 DEA, der L_1 akzeptiert und A_2 DEA, der L_2 akzeptiert
- Idee: „Baue“ aus A_1 und A_2 Automaten, der L akzeptiert



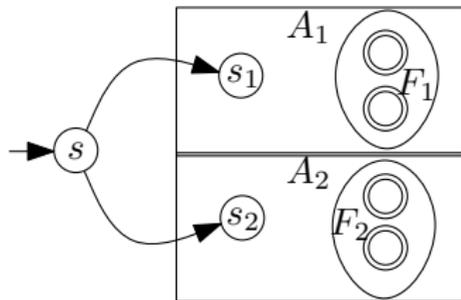
Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

■ Induktionsschluss:

- Für $n > 1$ gilt $L = L_1 \cup L_2$ oder $L = L_1 \cdot L_2$ oder $L = L_1^*$, mit L_1, L_2 regulär.
- **Fall 1:** $L = L_1 \cup L_2$
- Sei A_1 DEA, der L_1 akzeptiert und A_2 DEA, der L_2 akzeptiert
- Idee: „Baue“ aus A_1 und A_2 Automaten, der L akzeptiert



- NEA's können Übergang ohne Lesen eines Zeichens mit Wahlmöglichkeit
- NEA's können in DEA's umgebaut werden

Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

Beweis (Fortsetzung):

- **Fall 1:** $L = L_1 \cup L_2$
- NEA $\mathcal{A} := (Q, \Sigma, \delta, s, F)$, der $L_1 \cup L_2$ erkennt:
 - $Q := Q_1 \cup Q_2 \cup \{s\}$ ($s \notin Q_i$)
 - $F = F_1 \cup F_2$
 - δ :

$$\delta(q, a) := \begin{cases} \{\delta_i(q, a)\} & , q \in Q_i, a \in \Sigma \\ \emptyset & , q \in Q \setminus \{s\}, a = \varepsilon \\ \{s_1, s_2\} & , q = s, a = \varepsilon \\ \emptyset & , q = s, a \neq \varepsilon \end{cases}$$

Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

Beweis (Fortsetzung):

- **Fall 2:** $L = L_1 \cdot L_2$
- NEA $\mathcal{A} := (Q, \Sigma, \delta, s, F)$, der $L_1 \cdot L_2$ erkennt:

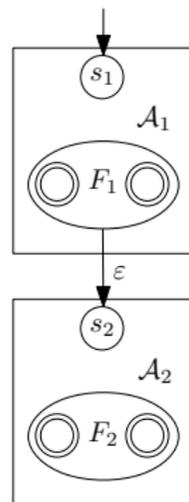
- $Q := Q_1 \cup Q_2$

- $s := s_1$

- $F := F_2$

- δ :

$$\delta(q, a) := \begin{cases} \{\delta_i(q, a)\} & , q \in Q_i, a \in \Sigma \\ \emptyset & , q \in Q \setminus F_1, a = \varepsilon \\ \{s_2\} & , q \in F_1, a = \varepsilon \end{cases}$$



Reguläre Sprache \Rightarrow Endlicher Automat

Satz.

Jede reguläre Sprache L wird von einem (deterministischen) endlichen Automaten (DEA) akzeptiert.

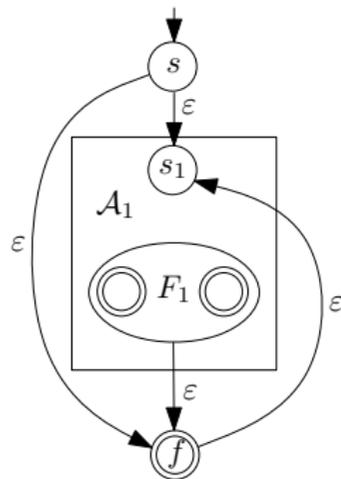
Beweis (Fortsetzung):

- **Fall 3:** $L = L_1^*$
- NEA $\mathcal{A} := (Q, \Sigma, \delta, s, F)$, der L_1^* erkennt:

- $Q := Q_1 \cup \{s, f\}$, ($s, f \notin Q_1$)

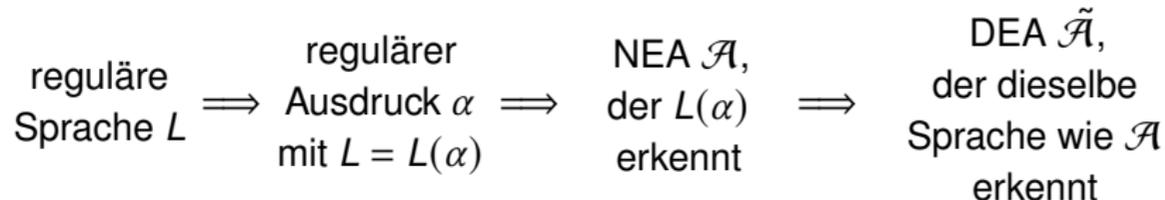
- δ :

$$\delta(q, a) := \begin{cases} \{\delta_1(q, a)\} & , q \in Q_1, a \in \Sigma \\ \emptyset & , q \in Q_1 \setminus F_1, a = \varepsilon \\ \{f\} & , q \in F_1 \cup \{s\}, a = \varepsilon \\ \emptyset & , q \in \{s, f\}, a \neq \varepsilon \\ \{s_1\} & , q \in \{s, f\}, a = \varepsilon \end{cases}$$



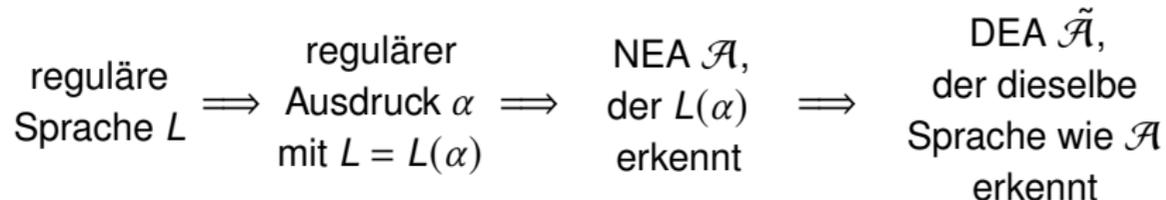
Zusammenfassung

- Reguläre Sprachen L sind induktiv definiert.
- Reguläre Ausdrücke α beschreiben reguläre Sprachen, d.h. $L = L(\alpha)$.
- Zu jedem regulären Ausdruck α gibt es einen NEA der $L(\alpha)$ erkennt.
- Zu jedem NEA gibt es einen DEA der die gleiche Sprache erkennt.



Zusammenfassung

- Reguläre Sprachen L sind induktiv definiert.
- Reguläre Ausdrücke α beschreiben reguläre Sprachen, d.h. $L = L(\alpha)$.
- Zu jedem regulären Ausdruck α gibt es einen NEA der $L(\alpha)$ erkennt.
- Zu jedem NEA gibt es einen DEA der die gleiche Sprache erkennt.



Ausblick:

- Was können NEAs mit Wahlmöglichkeiten, aber ohne ε -Übergänge?
- Gibt es Sprachen, die NEAs erkennen, aber DEAs nicht erkennen?
- Gibt es Sprachen, die NEAs nicht erkennen?

Testen Sie sich!

Testen Sie sich:

Sei $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \cup \{,\} \cup \{.\}$.

Sei $L_{\mathbb{R}} \subseteq \Sigma^*$ die Sprache aller endlichen Dezimaldarstellungen von positiven reellen Zahlen.

Beispiele: $1.024,48 \in L_{\mathbb{R}}$ und $42 \in L_{\mathbb{R}}$

aber $1024,48 \notin L_{\mathbb{R}}$ und $042,00 \notin L_{\mathbb{R}}$ und $.,.123 \notin L_{\mathbb{R}}$

↪ Finden Sie einen DEA oder NEA für $L_{\mathbb{R}}$?