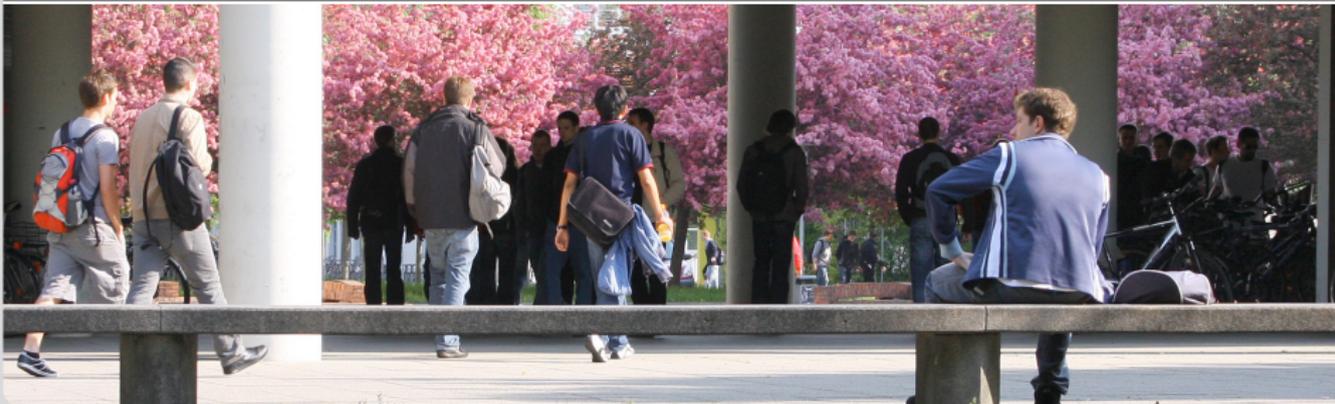


Theoretische Grundlagen der Informatik

Vorlesung am 24. Oktober 2017

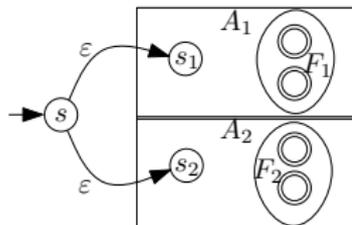
INSTITUT FÜR THEORETISCHE INFORMATIK



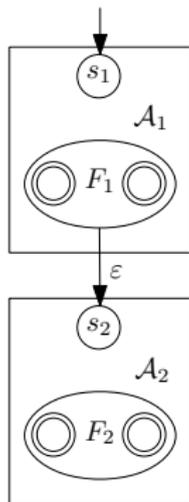
Satz:

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

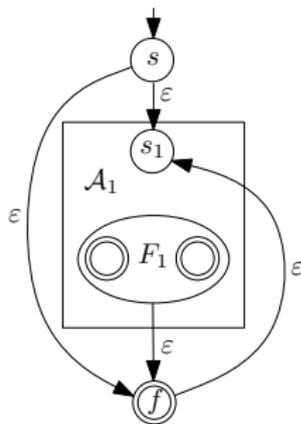
$L_1 \cup L_2$



$L_1 \cdot L_2$

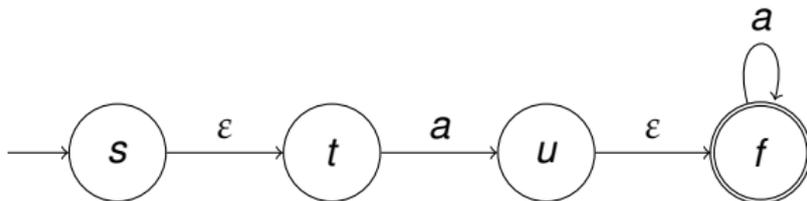


L_1^*



Satz:

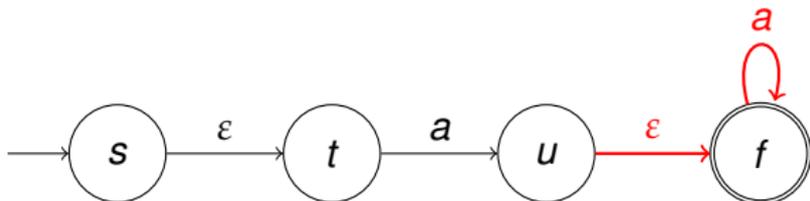
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$				$\{f\}$
$\delta(E(q), a)$				$\{f\}$

Satz:

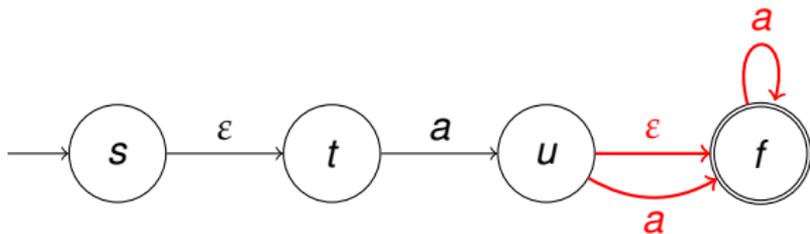
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$			$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$				$\{f\}$

Satz:

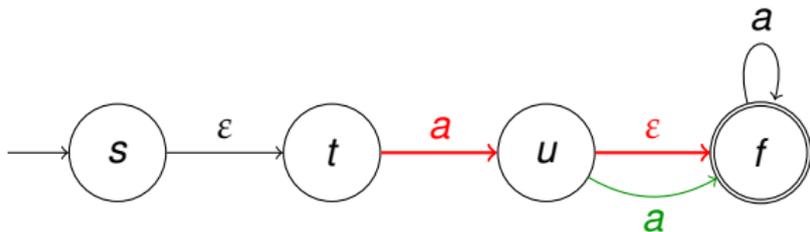
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$			$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$			$\{f\}$	$\{f\}$

Satz:

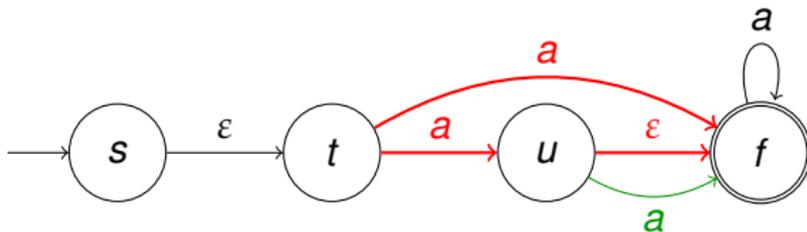
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$		$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$			$\{f\}$	$\{f\}$

Satz:

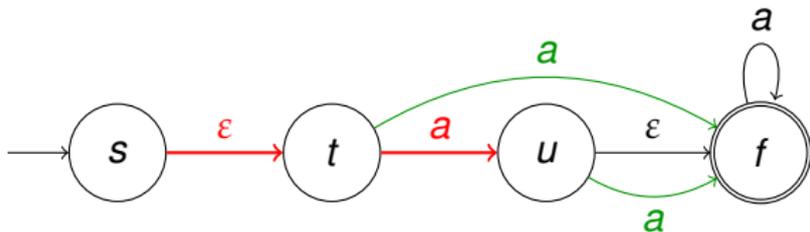
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$		$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$		$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

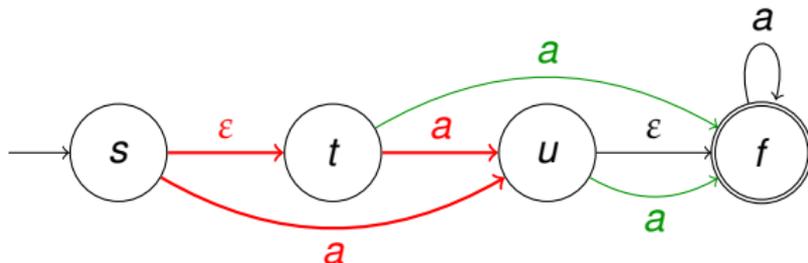
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$		$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

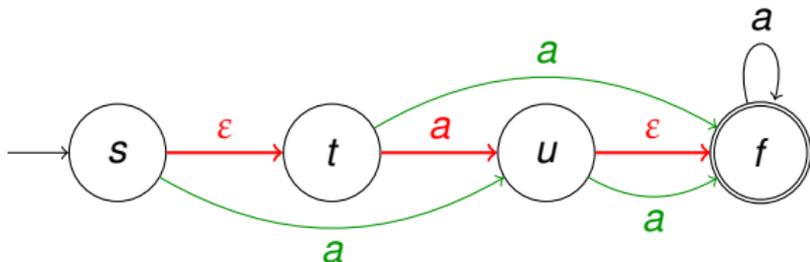
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$	$\{u\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

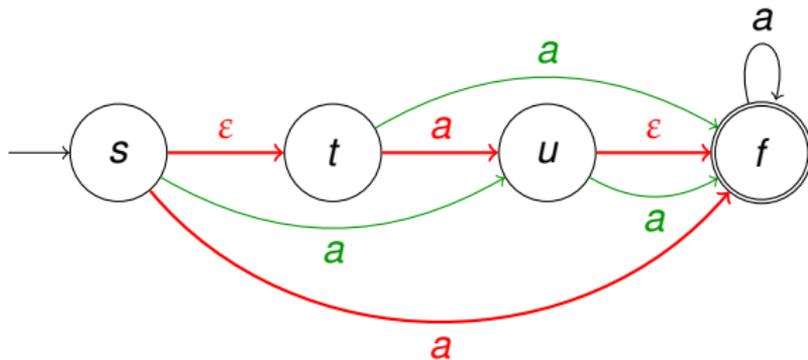
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$	$\{u\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

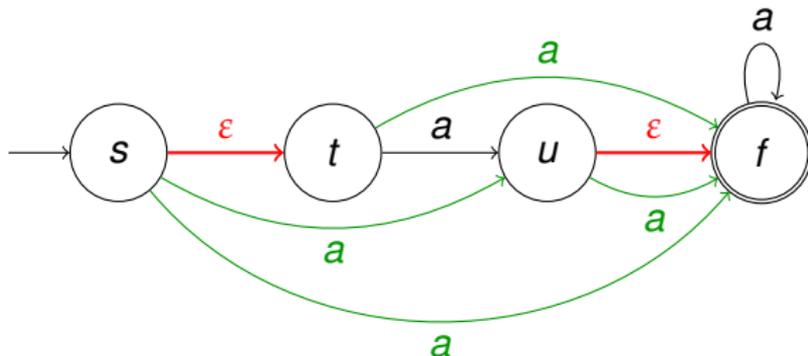
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

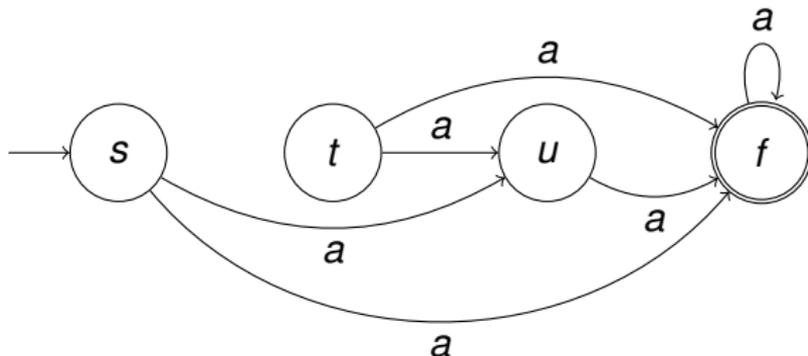
Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

Zu jedem NEA \mathcal{A} mit ϵ -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ϵ -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.



q	s	t	u	f
$E(q)$	$\{s, t\}$	$\{t\}$	$\{u, f\}$	$\{f\}$
$\delta(E(q), a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Satz:

Zu jedem NEA \mathcal{A} mit ε -Übergängen gibt es einen NEA $\tilde{\mathcal{A}}$ ohne ε -Übergänge, der dieselbe Sprache akzeptiert und nicht mehr Zustände hat.

Beweis: Sei $\mathcal{A} := (Q, \Sigma, \delta, s, F)$ ein NEA mit ε -Übergängen.

Wir konstruieren $\tilde{\mathcal{A}} := (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$ wie folgt:

- $\tilde{Q} := (Q \setminus F) \cup \tilde{F}$

- $\tilde{s} := s$



$$\tilde{\delta}(q, a) = \begin{cases} \{q\} & \text{falls } a = \varepsilon \\ \delta(E(q), a) & \text{sonst} \end{cases}$$

- $\tilde{F} := \{q \in Q \mid E(q) \cap F \neq \emptyset\}$

Damit akzeptiert $\tilde{\mathcal{A}}$ dieselbe Sprache wie \mathcal{A} , und $|\tilde{Q}| \leq |Q|$.

Satz:

Jede Sprache, die von einem endlichen Automaten erkannt wird, ist regulär.

Beweis: EA \rightarrow Regularität

- Sei DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben.
- Es ist zu zeigen, dass $L(\mathcal{A})$ regulär ist.

Es gilt:

$$L = \{w \in \Sigma^* \mid \mathcal{A} \text{ endet nach Abarbeitung von } w \text{ in einem Zustand aus } F\}$$

- Die Abarbeitung eines Wortes $w = a_1 \dots a_k$ bewirkt das Durchlaufen einer Folge von Zuständen s, q_1, \dots, q_k , wobei nicht notwendig $q_i \neq q_j$ für $i \neq j$ gilt.
- Wir suchen die Wörter, so dass der letzte Zustand in F ist.
- Betrachte für jeden Zustand $f \in F$ getrennt die Wörter, deren Abarbeitung in f endet.

Beweis: EA \rightarrow Regularität

- Sei DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben.
- Es ist zu zeigen, dass $L(\mathcal{A})$ regulär ist.

Es gilt:

$$L = \{w \in \Sigma^* \mid \mathcal{A} \text{ endet nach Abarbeitung von } w \text{ in einem Zustand aus } F\}$$

Zu $f \in F$ definiere:

$$\begin{aligned} L_f &:= \{w \in \Sigma^* \mid \mathcal{A} \text{ endet nach Abarbeitung von } w \text{ in } f\} \\ &= \{w \in \Sigma^* \mid w \text{ überführt } s \text{ in } f \text{ (im Automaten } \mathcal{A})\} \end{aligned}$$

- Damit ist $L = \bigcup_{f \in F} L_f$.
- Wenn wir zeigen können, dass für alle $f \in F$ die Sprache L_f regulär ist, so ist auch L regulär.

Beweis: EA \rightarrow Regularität

$$L_f := \{w \in \Sigma^* \mid w \text{ überführt } s \text{ in } f \text{ (im Automaten } \mathcal{A})\}$$

Ab jetzt sei $Q = \{q_1, \dots, q_n\}$.

Wir definieren zu

$$q_r, q_t \in Q: L_{q_r, q_t} := \{w \in \Sigma^* \mid w \text{ überführt } q_r \text{ in } q_t\} .$$

Insbesondere gilt also: $L_f = L_{s, f}$. Unterteile L_{q_r, q_t} :

$$L_{q_r, i, q_t} := \left\{ w \in \Sigma^* \mid \begin{array}{l} \text{Abarbeitung von } w \text{ aus } q_r \text{ nach } q_t \text{ hat nur} \\ \text{Zwischenzustände } \{q_1, \dots, q_i\} \end{array} \right\}$$

(also w bewirkt: $q_r \rightarrow \underbrace{\dots\dots\dots}_{\in \{q_1, \dots, q_i\}} \rightarrow q_t$.)

Damit gilt $L_{q_r, q_t} = L_{q_r, n, q_t}$.

Beweis: EA \rightarrow Regularität

$$L_{q_r, i, q_t} := \left\{ w \in \Sigma^* \mid \begin{array}{l} \text{Abarbeitung von } w \text{ aus } q_r \text{ nach } q_t \text{ hat nur} \\ \text{Zwischenzustände } \{q_1, \dots, q_i\} \end{array} \right\}$$

Wir zeigen, dass L_{q_r, i, q_t} für $q_r, q_t \in Q$ und $1 \leq i \leq n$ regulär sind:

- Zunächst betrachten wir direkte Überführungen, also $i = 0$:

$$L_{q_r, 0, q_t} := \left\{ w \in \Sigma^* \mid \begin{array}{l} \text{Abarbeitung von } w \text{ führt von } q_r \text{ nach } q_t \\ \text{ohne Zwischenzustand} \end{array} \right\}$$

Falls $r = t$ und somit $q_r = q_t$ ist, ist

$$L_{q_r, 0, q_t} = \{a \in \Sigma \mid \delta(q_t, a) = q_t\} \cup \{\varepsilon\}.$$

Andernfalls betrachten wir alle w mit $q_r \xrightarrow{w} q_t$, ohne Zwischenzustände, also

$$L_{q_r, 0, q_t} = \{a \in \Sigma \mid \delta(q_r, a) = q_t\}.$$

Diese Sprachen sind jeweils regulär.

Beweis: EA \rightarrow Regularität

- Betrachte nun $i = 1$:

$$L_{q_r,1,q_t} := \left\{ w \in \Sigma^* \mid \begin{array}{l} w \text{ überführt } q_r \text{ in } q_t \text{ entweder direkt oder} \\ \text{unter Benutzung nur von } q_1 \end{array} \right\}$$

Es gilt dann:

$$L_{q_r,1,q_t} = L_{q_r,0,q_t} \cup \left(L_{q_r,0,q_1} \cdot L_{q_1,0,q_1}^* \cdot L_{q_1,0,q_t} \right)$$

Also ist $L_{q_r,1,q_t}$ auch wieder regulär.

- Es gilt allgemein:

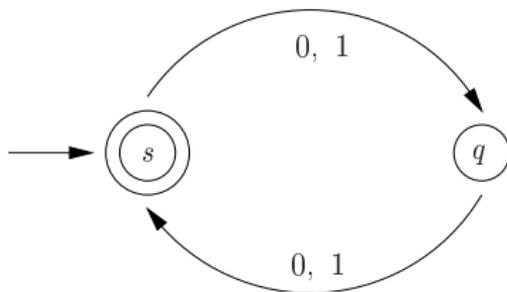
$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} \left(L_{q_{i+1},i,q_{i+1}} \right)^* L_{q_{i+1},i,q_t} \right)$$

Beweis: EA \rightarrow Regularität

- Es wurden für $L_{\cdot, i+1, \cdot}$ nur die Sprachen $L_{\cdot, i, \cdot}$ und $\cup, \cdot, *$ verwendet.
- Damit ist gezeigt (per Induktion), dass $L_{\cdot, i+1, \cdot}$ regulär ist für beliebiges i ($1 \leq i+1 \leq n$) und alle Zustandspaare aus Q^2 .
- Damit ist gezeigt, dass insbesondere $L_f = L_{s, n, f}$ regulär ist für jedes $f \in F$.

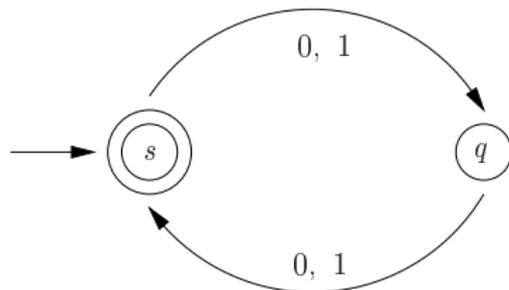
Beispiel

Sei $(Q, \Sigma, \delta, s, F)$ mit $Q := \{q_1 := s, q_2 := q\}$, $\Sigma := \{0, 1\}$, $F := \{s\}$



Gesucht: $L(Q, \Sigma, \delta, s, F)$. Es gilt $L = L_{q_1, 2, q_1}$.

Beispiel



Gesucht: $L(Q, \Sigma, \delta, s, F)$. Es gilt $L = L_{q_1, 2, q_1}$.

Dann ist

$$L_{q_i, 0, q_i} = \varepsilon$$

$$L_{q_i, 0, q_j} = (0 \cup 1) \text{ für } i, j \in \{1, 2\}, i \neq j$$

$$L_{q_1, 1, q_1} = L_{q_1, 0, q_1} \cup L_{q_1, 0, q_1} (L_{q_1, 0, q_1})^* L_{q_1, 0, q_1} = \varepsilon$$

$$L_{q_1, 1, q_2} = L_{q_1, 0, q_2} \cup L_{q_1, 0, q_1} (L_{q_1, 0, q_1})^* L_{q_1, 0, q_2} = (0 \cup 1) \cup \varepsilon \varepsilon^* (0 \cup 1) = 0 \cup 1$$

$$L_{q_2, 1, q_1} = (0 \cup 1) \cup (0 \cup 1) \varepsilon^* \varepsilon = 0 \cup 1$$

$$L_{q_2, 1, q_2} = \varepsilon \cup (0 \cup 1) \varepsilon^* (0 \cup 1) = \varepsilon \cup (0 \cup 1)(0 \cup 1)$$

$$\begin{aligned} L &= L_{q_1, 2, q_1} = L_{q_1, 1, q_1} \cup (L_{q_1, 1, q_2} (L_{q_2, 1, q_2})^* L_{q_2, 1, q_1}) \\ &= \varepsilon \cup (0 \cup 1) ((0 \cup 1)(0 \cup 1))^* (0 \cup 1) = ((0 \cup 1)(0 \cup 1))^* \end{aligned}$$

- Wir haben gezeigt, dass die von endlichen Automaten akzeptierten Sprachen genau die regulären Sprachen sind.
- Dies wird auch als der **Satz von Kleene** bezeichnet.

Satz (Satz von Kleene):

Die von endlichen Automaten akzeptierten Sprachen sind genau die regulären Sprachen.

Frage: Was können endliche Automaten nicht?

Frage: Was können endliche Automaten nicht?

Beispiel:

Die Sprache L der korrekten Klammerausdrücke über $\Sigma = \{(\,)\}$.

Etwa

$$\left(() \right), \left(() () \right) \in L \qquad ((()), (())) () \notin L$$

Frage: Was können endliche Automaten nicht?

Beispiel:

Die Sprache L der korrekten Klammerausdrücke über $\Sigma = \{ (,) \}$.

Etwa

$$\left((()) \right), \left((()) (()) \right) \in L \qquad ((()), (())) (()) \notin L$$

- Die Klammerung ist genau dann korrekt, wenn w gleich viele öffnende wie schließende Klammern enthält, und wenn man w von links nach rechts liest, so gibt es nie mehr „)“ als „(“ bis dahin.
- Ein Automat, der L erkennen kann, muss in der Lage sein, sich für ein beliebiges Wort $w \in L$ die Anzahl von (gegenüber) zu merken.
- Dies kann aber beliebig groß werden, und der Automat müsste über unendliche viele Zustände verfügen.
- Die Sprache der Klammerausdrücke ist also zwar simpel, aber wohl nicht regulär.

Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Beweis:

- Sei L eine reguläre Sprache.
- Dann existiert ein endlicher Automat, der L akzeptiert.
- Sei Q dessen Zustandsmenge und $n := |Q|$.
- Sei $w \in L$ mit $|w| > n$, etwa $w = a_1 \dots a_n \dots a_m$ mit $m > n$.

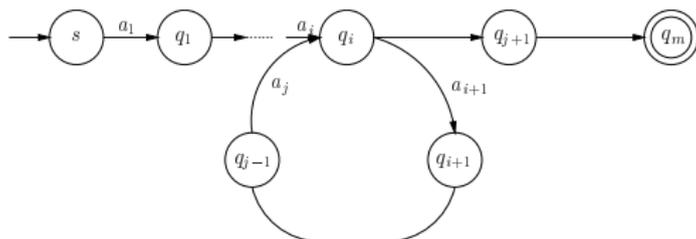
Pumping-Lemma für reguläre Sprachen

Beweis:

- Sei L eine reguläre Sprache.
- Dann existiert ein endlicher Automat, der L akzeptiert.
- Sei Q dessen Zustandsmenge und $n := |Q|$.
- Sei $w \in L$ mit $|w| > n$, etwa $w = a_1 \dots a_n \dots a_m$ mit $m > n$.

Bei der Abarbeitung von w werden dann die Zustände q_0, \dots, q_m durchlaufen mit $q_m \in F$.

Dann gibt es i, j mit $0 \leq i, j \leq n$ und $i \neq j$, so dass $q_i = q_j$. ☹ gelte $i < j$.

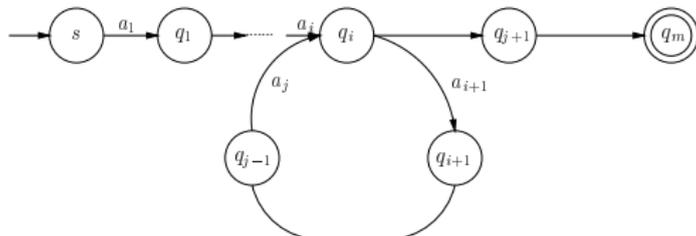


Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.



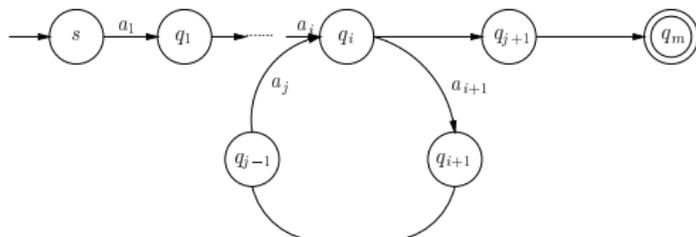
Dann kann der Zykel $q_i, q_{i+1}, \dots, q_j = q_i$ auch gar nicht oder beliebig oft bei der Abarbeitung eines Wortes aus L durchlaufen werden so dass der Zustand $q_m \in F$ erreicht wird.

Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.



Also gibt es eine Zerlegung $w = \underbrace{(a_1 \dots a_j)}_u \cdot \underbrace{(a_{j+1} \dots a_j)}_v \cdot \underbrace{(a_{j+1} \dots a_m)}_x$

mit $|uv| \leq n$ und $v \neq \varepsilon$, so dass auch $uv^i x \in L$ für alle $i \in \mathbb{N}_0$.

- Das Pumping-Lemma liefert nur eine notwendige, aber nicht hinreichende Bedingung für die Regularität von Sprachen.

Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Gegeben sei

- $\Sigma = \{0, 1\}$
- $L = \{w \in \Sigma^* \mid w \text{ enthält } 10 \text{ nicht als Teilwort}\} = 0^*1^*$

Betrachte

- $n = 1, \quad w = uvx, \quad u = \varepsilon$

Dann

- entspricht v also dem ersten Buchstaben von w
- kann $uv^i x$ auch 10 nicht als Teilwort besitzen.

Satz:

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Sei $\Sigma = \{0, 1\}$ und $L = \{0^i 1^i \mid i \geq 0\}$. Wir zeigen: L ist nicht regulär.

- Für ein n wähle $w = 0^n 1^n$
- Dann ist $|w| > n$
- Für jede Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ ist aber

$$uv^0 x = 0^l 1^n \notin L \quad (l < n)$$

Beispiel (3) zum PL

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

$$L = \left\{ w \in \Sigma^* \mid w = 1^k \ (k > 0) \text{ oder } w = 0^j 1^{k^2} \ (j \geq 1, k \geq 0) \right\}.$$

Dann erfüllt L die Darstellung des PL:

- Sei $n = 1$ und $w \in L$ mit $|w| > 1$.
- w habe eine Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$.

Setze $u = \varepsilon$ und $|v| = 1$ das erste Symbol von w .

- Falls $w = 1^k$, so ist auch $uv^i x$ vom Typ $1^\ell \in L$.
- Falls $w = 0^j 1^{k^2}$, so ist auch $uv^0 x \in L$ (für $j = 1$ ist $uv^0 x = x = 1^{k^2}$).
Für $i \geq 1$ gilt $uv^i x = 0^{j+i} 1^{k^2} \in L$.

Trotzdem ist L nicht regulär. Dies lässt sich mit dem verallgemeinertem Pumping Lemma zeigen.