Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik
Vorlesung am 23. Oktober 2018

INSTITUT FÜR THEORETISCHE INFORMATIK

neue Termine (Änderungen vorbehalten)

Dienstags		Donnerstags	
16.10.	Vorlesung	18.10.	Vorlesung
23.10.	Vorlesung	25.10.	Übung
30.10.	Vorlesung	01.11.	Feiertag
06.11.	Vorlesung	08.11.	Übung
13.11.	Vorlesung	15.11.	Vorlesung
20.11.	Übung	22.11.	Vorlesung
27.11.	Vorlesung	29.11.	Übung
04.12.	Vorlesung	06.12.	Vorlesung
11.12.	Übung	13.12.	Vorlesung
18.12.	Vorlesung	20.12.	Übung
08.01.	Vorlesung	10.01.	Vorlesung
15.01.	Übung	17.01.	Vorlesung
22.01.	Vorlesung	24.01.	Übung
29.01.	Vorlesung	31.01.	Vorlesung
05.02.	Übung	07.02.	Vorlesung

Letzte Vorlesung

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

reguläre Sprache L $\Longrightarrow$ regulärer Ausdruck α mit $L = L(\alpha)$ $\Longrightarrow$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\Longrightarrow$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Heute:

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

Letzte Vorlesung

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

reguläre Sprache L $\Longleftrightarrow$ regulärer Ausdruck α mit $L = L(\alpha)$ $\Longrightarrow$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\Longrightarrow$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Heute:

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

Letzte Vorlesung

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

reguläre Sprache L $\Longleftrightarrow$ regulärer Ausdruck α mit $L = L(\alpha)$ $\Longrightarrow$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\Longleftrightarrow$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Heute:

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

Letzte Vorlesung

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

reguläre Sprache L	$\Longleftrightarrow$	regulärer Ausdruck α mit $L = L(\alpha)$	$\stackrel{?}{\Longrightarrow}$	NEA $\mathcal{A}$, der $L(\alpha)$ erkennt	$\Longleftrightarrow$	DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Heute:

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

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.

$$
\begin{tikzcd}
\arrow[r] & s \arrow[r, "\varepsilon"] & t \arrow[r, "a"] & u \arrow[r, "\varepsilon"] & f \arrow[loop above, "a"]
\end{tikzcd}
$$

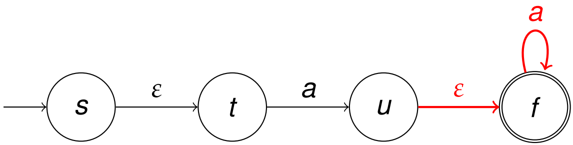
q	s	t	u	f
$\bar{\delta}(q, a)$				$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



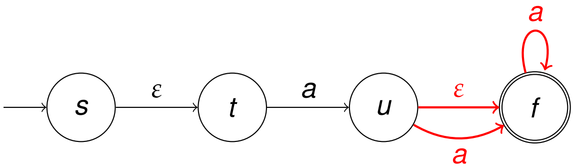
q	s	t	u	f
$\bar{\delta}(q, a)$			$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



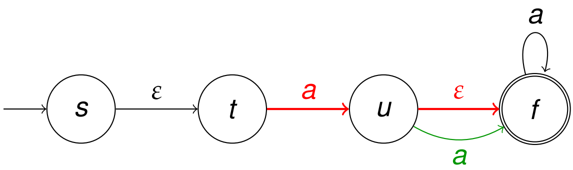
q	s	t	u	f
$\bar{\delta}(q, a)$			$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



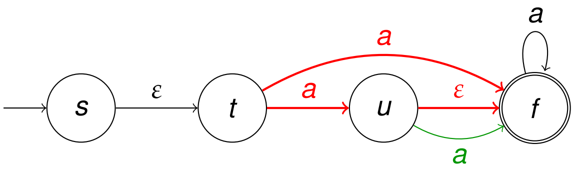
q	s	t	u	f
$\bar{\delta}(q, a)$		$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



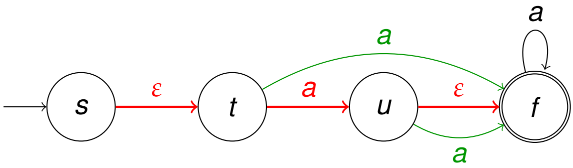
q	s	t	u	f
$\bar{\delta}(q, a)$		$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



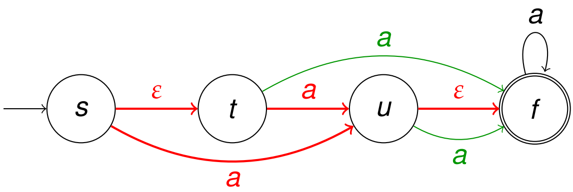
q	s	t	u	f
$\bar{\delta}(q, a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



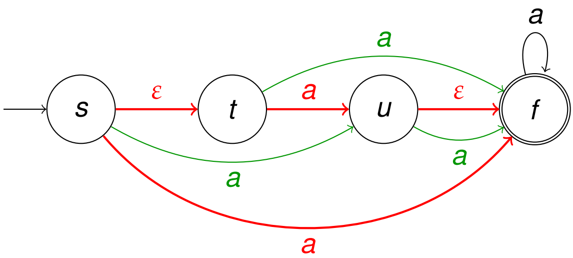
q	s	t	u	f
$\bar{\delta}(q, a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



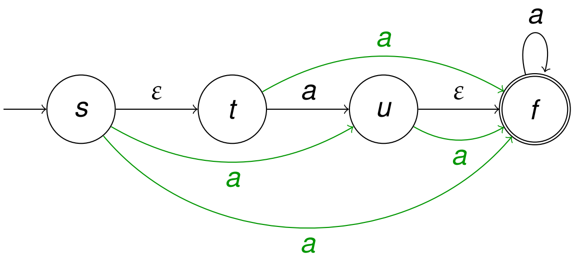
q	s	t	u	f
$\bar{\delta}(q, a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



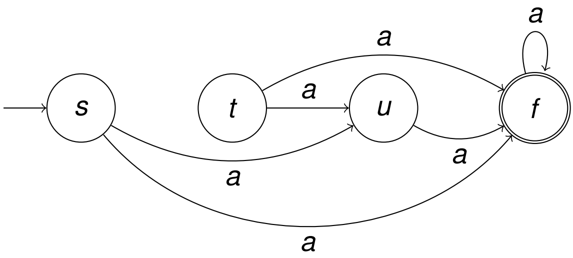
q	s	t	u	f
$\bar{\delta}(q, a)$	$\{u, f\}$	$\{u, f\}$	$\{f\}$	$\{f\}$

Entfernen von ϵ-Übergängen

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

Erinnerung an Erweiterung $\bar{\delta}$:

$p \in \bar{\delta}(q, a)$ heißt, dass p von q erreichbar ist mit beliebig vielen ε-Übergängen und genau einem a-Übergang.



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

Entfernen von ϵ-Übergängen

Satz:
Zu jedem NEA $\mathcal{A}$ mit ε-Übergängen gibt es einen NEA $\widetilde{\mathcal{A}}$ 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.
Konstruiere NEA $\widetilde{\mathcal{A}} := (\widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{s}, \widetilde{F})$ ohne ε-Übergänge wie folgt:

- $\widetilde{Q} := Q$, $\widetilde{s} := s$, $\widetilde{F} := F$ (bzw. $\widetilde{F} := F \cup \{\widetilde{s}\}$, falls $E(s) \cap F \neq \emptyset$)
-

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

Ein Übergang in $\widetilde{\mathcal{A}}$ entspricht einer Folge von Übergängen in $\mathcal{A}$ von denen genau einer kein ε-Übergang ist, und umgekehrt.

$\rightsquigarrow$ $\widetilde{\mathcal{A}}$ und $\mathcal{A}$ erkennen dieselbe Sprache, d.h. sie sind äquivalent.

Letzte Vorlesung

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

reguläre Sprache L $\iff$ regulärer Ausdruck α mit $L = L(\alpha)$ $\stackrel{?}{\Longrightarrow}$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\iff$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Heute:

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

EA $\rightarrow$ Regularität

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:

$L = \{w \in \Sigma^* \mid \mathcal{A}$ endet nach Abarbeitung von w 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 $Q = \{q_1, \ldots, q_n\}$.
Wir definieren zu

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

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

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

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

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

Beweis: EA $\rightarrow$ Regularität

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

Wir zeigen, dass L_{q_r,i,q_t} für $q_r, q_t \in Q$ und $0 \leq i \leq n$ regulär sind:

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

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

  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

Karlsruher Institut für Technologie

- Betrachte nun $i = 1$:

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

  Es gilt dann:

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

  Also ist $L_{q_r,1,q_t}$ auch wieder regulär, weil $L_{\cdot,0,\cdot}$ regulär ist.

- Wir wollen per Induktion nach i zeigen, dass L_{q_r,i,q_t} für alle $q_r, q_t \in Q$ und alle $0 \leq i \leq n$ regulär ist.
- Für den Induktionsschritt $i \geq 1$ gilt allgemein:

$$L_{q_r,i,q_t} = L_{q_r,i-1,q_t} \cup \left(L_{q_r,i-1,q_i} \cdot \left(L_{q_i,i-1,q_i} \right)^* \cdot L_{q_i,i-1,q_t} \right)$$

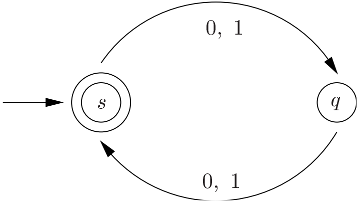
Beweis: EA $\rightarrow$ Regularität

$$L_{q_r,i,q_t} = L_{q_r,i-1,q_t} \cup \left(L_{q_r,i-1,q_i} \cdot \left(L_{q_i,i-1,q_i}\right)^* \cdot L_{q_i,i-1,q_t}\right)$$

- Es wurden für L_{q_r,i,q_t} nur die Sprachen $L_{\cdot,i-1,\cdot}$ und $\cup, \cdot, *$ verwendet.
- Per Induktion nach i können wir annehmen, dass alle Sprachen der Form $L_{\cdot,i-1,\cdot}$ regulär sind.
- Damit ist gezeigt, dass auch L_{q_r,i,q_t} regulär ist für alle Zustandspaare aus $q_r, q_t \in Q$.
- Damit ist gezeigt, dass insbesondere $L_f = L_{s,n,f}$ regulär ist für jedes $f \in F$.
- Damit ist auch $L(\mathcal{A}) = \bigcup_{f \in F} L_f$ regulär.

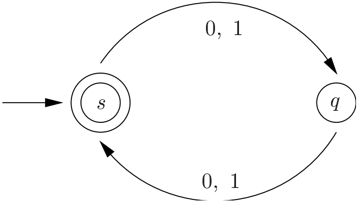
Beispiel

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

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

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

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

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

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

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

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

Satz von Kleene

- 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.

reguläre Sprache L	$\Longleftrightarrow$	regulärer Ausdruck α mit $L = L(\alpha)$	$\Longleftrightarrow$	NEA $\mathcal{A}$, der $L(\alpha)$ erkennt	$\Longleftrightarrow$	DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Satz von Kleene

- 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.

reguläre Sprache L	$\Longleftrightarrow$	regulärer Ausdruck α mit $L = L(\alpha)$	$\Longleftrightarrow$	NEA $\mathcal{A}$, der $L(\alpha)$ erkennt	$\Longleftrightarrow$	DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

- Gibt es Sprachen, die NEAs nicht erkennen?

Frage: Was können endliche Automaten nicht?

Frage: Was können endliche Automaten nicht?

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

$$\Big(()()\Big), \Bigg(()()\Big(()\Big)\Bigg) \in L \qquad ((()), (()))()(\notin L$$

Frage: Was können endliche Automaten nicht?

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

$$\Big(()()\Big), \Bigg(()()\Big(()\Big)\Bigg) \in L \qquad ((()), (()))()(\notin L$$

- 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.

Pumping-Lemma für reguläre 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 $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Pumping-Lemma für reguläre 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 $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Für alle	$\forall L \subseteq \Sigma^*$ mit L regulär
existiert	$\exists n \in \mathbb{N}$
für alle	$\forall w \in L$ mit $\|w\| > n$
existiert	$\exists u, v, x \in \Sigma^*$ mit $w = uvx, \|uv\| \leq n, v \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0:$
gilt	$uv^ix \in L$

Pumping-Lemma für reguläre 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 $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Beweis:

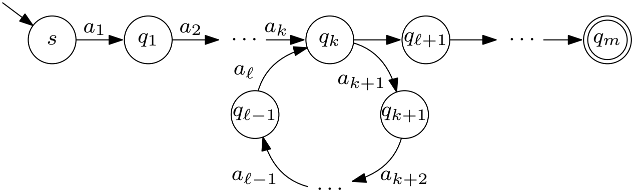
- Sei L eine reguläre Sprache.
- Dann existiert ein endlicher Automat, der L akzeptiert.
- Sei Q dessen Zustandsmenge und $n := |Q|$.
- Sei $w \in L$ beliebig mit $|w| > n$, etwa $w = a_1 \dots a_n \dots a_m$ mit $m > n$.

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 $w \in L$ beliebig mit $|w| > n$, etwa $w = a_1 \dots a_n \dots a_m$ mit $m > n$.

Bei der Abarbeitung von w werden dann Zustände $q_0, \dots, q_m$ durchlaufen mit $q_0 = s$ und $q_m \in F$.
Da $m > n$, gibt es k, ℓ mit $k \neq \ell$ und $q_k = q_\ell$.
OBdA. sei $k < \ell$ und seien k, ℓ kleinstmöglich. Also $0 \leq k < \ell \leq n$.

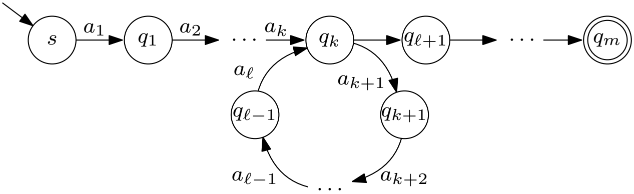


Pumping-Lemma für reguläre Sprachen

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 $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.



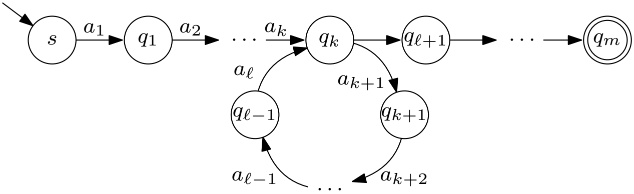
Dann kann der Zykel $q_k, q_{k+1}, \dots, q_\ell = q_k$ auch gar nicht oder beliebig oft bei der Abarbeitung eines Wortes aus L durchlaufen werden und es wird stets der Zustand $q_m \in F$ erreicht.

Pumping-Lemma für reguläre 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 $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.



Also gibt es eine Zerlegung $w = \underbrace{(a_1 \dots a_k)}_{u} \cdot \underbrace{(a_{k+1} \dots a_\ell)}_{v} \cdot \underbrace{(a_{\ell+1} \dots a_m)}_{x}$

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

Verwendung des Pumping-Lemmas

- Das Pumping-Lemma liefert eine notwendige Bedingung für die Regularität von Sprachen.
- Durch Widerlegen der Aussage des Pumping-Lemmas kann man zeigen, dass eine Sprache nicht regulär ist.

- Das Pumping-Lemma liefert keine hinreichende Bedingung für die Regularität von Sprachen.
- Durch Nachweisen der Aussage des Pumping-Lemmas kann man nichts zeigen.

Merke:
Jede reguläre Sprache erfüllt die Aussage des Pumping-Lemmas.
Eine Sprache, die die Aussage des Pumping-Lemmas nicht erfüllt, ist auch nicht regulär.

Verwendung des Pumping-Lemmas

Aussage des Pumping-Lemmas für gegebene Sprache L:

existiert	$\exists n \in \mathbb{N}$
für alle	$\forall w \in L$ mit $\lvert w \rvert > n$
existiert	$\exists u, v, x \in \Sigma^*$ mit $w = uvx, \lvert uv \rvert \leq n, v \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0:$
gilt	$uv^i x \in L$

Verwendung des Pumping-Lemmas

Aussage des Pumping-Lemmas für gegebene Sprache L:

existiert	$\exists n \in \mathbb{N}$
für alle	$\forall w \in L$ mit $\lvert w \rvert > n$
existiert	$\exists u, v, x \in \Sigma^*$ mit $w = uvx, \lvert uv \rvert \leq n, v \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0:$
gilt	$uv^ix \in L$

Widerlegen der Aussage des Pumping-Lemmas für gegebene Sprache L:

für alle	$\forall n \in \mathbb{N}$
existiert	$\exists w \in L$ mit $\lvert w \rvert > n$
für alle	$\forall u, v, x \in \Sigma^*$ mit $w = uvx, \lvert uv \rvert \leq n, v \neq \varepsilon$
existiert	$\exists i \in \mathbb{N}_0:$
gilt	$uv^ix \notin L$

Beispiel (1) zum PL

Aussage des Pumping-Lemmas für gegebene Sprache L:

existiert	$\exists n \in \mathbb{N}$
für alle	$\forall w \in L$ mit $\lvert w \rvert > n$
existiert	$\exists u, v, x \in \Sigma^*$ mit $w = uvx, \lvert uv \rvert \leq n, v \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0:$
gilt	$uv^i x \in L$

Beispiel (1)

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

“∃” Wähle $n = 1$.
“∀” Betrachte beliebiges $w \in L$, $\lvert w \rvert > n$.
“∃” Wähle Zerlegung $w = uvx$ mit $u = \varepsilon$, $\lvert v \rvert = 1$.
“∀” Für beliebiges $i \in \mathbb{N}_0$ hat $uv^i x$ nicht 10 als Teilwort; $uv^i x \in L$.

$\rightsquigarrow$ L erfüllt die Aussage des Pumping-Lemmas.

Beispiel (2) zum PL

Widerlegen der Aussage des Pumping-Lemmas für Sprache L:

für alle	$\forall n \in \mathbb{N}$
existiert	$\exists w \in L$ mit $\lvert w \rvert > n$
für alle	$\forall u, v, x \in \Sigma^*$ mit $w = uvx$, $\lvert uv \rvert \leq n$, $v \neq \varepsilon$
existiert	$\exists i \in \mathbb{N}_0$:
gilt	$uv^i x \notin L$

Beispiel (2)

- $\Sigma = \{0, 1\}$, $L = \{0^k 1^k \mid k \geq 0\} = \{\varepsilon, 01, 0011, 000111, \ldots\}$

“$\forall$” Betrachte beliebiges $n \in \mathbb{N}$.

“$\exists$” Wähle $w = 0^n 1^n$. Beachte: $\lvert w \rvert = 2n > n$ und $w \in L$.

“$\forall$” Betrachte beliebige Zerlegung $w = uvx$, $\lvert uv \rvert \leq n$, $v \neq \varepsilon$.

“$\exists$” Wähle $i = 0$. Da $v = 0^a$ für ein $a \geq 1$, ist $uv^0x = 0^{n-a}1^n \notin L$.

$\rightsquigarrow$ L erfüllt nicht Aussage des Pumping-Lemmas. $\rightsquigarrow$ L ist nicht regulär.

Beispiel (3) zum PL

$$\exists n \,\forall w \in L, |w| > n \,\exists uvx = w, |uv| \leq n, v \neq \varepsilon \,\forall i \in \mathbb{N}_0 : uv^i x \in L$$

Beispiel (3)

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

  "$\exists$" Wähle $n = 1$.
  "$\forall$" Betrachte beliebiges $w \in L$, $|w| > n$.
  "$\exists$" Wähle Zerlegung $w = uvx$ mit $u = \varepsilon$, $|v| = 1$.
  "$\forall$" Betrachte beliebiges $i \in \mathbb{N}_0$.
- Falls $w = 1^k$, so ist $uv^i x = 1^{k+i-1}$, also vom Typ $1^\ell \in L$.
- Falls $w = 0^j 1^{k^2}$ und $i = 0$, so ist $uv^0 x = 1^{k^2} \in L$ oder $uv^0 x = 0^{j-1} 1^{k^2} \in L$.
- Falls $w = 0^j 1^{k^2}$ und $i \geq 1$, so ist $uv^i x = 0^{j+i} 1^{k^2} \in L$.

$\rightsquigarrow$ *L* erfüllt die Aussage des Pumping-Lemmas. (Aber *L* ist nicht regulär!)

Zusammenfassung

- beliebiger NEA $\mathcal{A}$ $\longrightarrow$ NEA $\widetilde{\mathcal{A}}$ ohne ε-Übergänge
- beliebiger DEA $\mathcal{A}$ $\longrightarrow$ regulärer Ausdruck für $L(\mathcal{A})$

reguläre Sprache L $\iff$ regulärer Ausdruck α mit $L = L(\alpha)$ $\iff$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\iff$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

Zusammenfassung

- beliebiger NEA $\mathcal{A}$ $\longrightarrow$ NEA $\widetilde{\mathcal{A}}$ ohne ε-Übergänge
- beliebiger DEA $\mathcal{A}$ $\longrightarrow$ regulärer Ausdruck für $L(\mathcal{A})$

reguläre Sprache L $\Longleftrightarrow$ regulärer Ausdruck α mit $L = L(\alpha)$ $\Longleftrightarrow$ NEA $\mathcal{A}$, der $L(\alpha)$ erkennt $\Longleftrightarrow$ DEA $\tilde{\mathcal{A}}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt

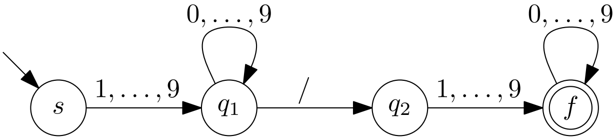
Aussage des Pumping-Lemmas:
$\exists n \,\forall w \in L, |w| > n \,\exists uvx = w, |uv| \leq n, v \neq \varepsilon \,\forall i \in \mathbb{N}_0 : uv^i x \in L \quad (\star)$

- Pumping-Lemma: L regulär $\Longrightarrow$ L erfüllt $(\star)$
- Widerlegen der Aussage des Pumping-Lemmas beweist Nicht-Regularität einer Sprache:

  L erfüllt $(\star)$ nicht $\Longrightarrow$ L ist nicht regulär $\Longleftrightarrow$ L nicht von EA erkannt

Testen Sie sich:

Sei $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \cup \{/\}$.
Sei $L = L(\mathcal{A})$ mit folgendem Automaten $\mathcal{A}$:



⇝ Finden Sie einen regulären Ausdruck für L?

⇝ Gilt die Aussage des Pumping-Lemmas für L? (Für welche n?)

Sei $\Sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \cup \{=\}$.
Sei $L' = \{w \in \Sigma^* \mid w = uvu \text{ mit } u \in \{1, \ldots, 9\}^*, v = \texttt{=}\}$.

⇝ Gilt die Aussage des Pumping-Lemmas für L'?