Theoretische Grundlagen der Informatik

Vorlesung am 10.11.2022

Torsten Ueckerdt | 10. November 2022

Letzte Vorlesung

Karlsruher Institut für Technologie

Pumping-Lemma für reguläre Sprachen

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 $w = uvx$ 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	$uv^ix \in L$

Verallgemeinertes PL für reguläre Sprachen

Karlsruher Institut für Technologie

Verallgemeinertes Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und für jede Darstellung $w = pys$ mit $|y| = n$ eine Darstellung y=uvx mit $v \neq \varepsilon$, existiert, bei der auch $puv^ixs \in L$ 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 $w = pys, \|y\| = n$
existiert	$\exists u, v, x \in \Sigma^*$ mit $y = uvx, v \neq \varepsilon$
für alle	$\forall i \in \mathbb{N}_0:$
gilt	$puv^ixs \in L$

Verallgemeinertes PL für reguläre Sprachen

Verallgemeinertes Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| \geq n$ und für jede Darstellung $w = pys$ mit $|y| = n$ eine Darstellung y=uvx mit $v \neq \varepsilon$, existiert, bei der auch $puv^ixs \in L$ ist für alle $i \in \mathbb{N}_0$.

Beweis:

- Betrachte L eine reguläre Sprache, beliebig. $\rightsquigarrow \forall L$ regulär
- Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein DEA, der L erkennt.
- Wähle $n := |Q|$. $\rightsquigarrow \exists n \in \mathbb{N}$
- Betrachte $pys \in L$ mit $|y| = n$, beliebig. $\rightsquigarrow \forall w = pys \in L, |y| = n$
- Sei $q_0, \ldots, q_n$ die Folge der $n + 1$ Zustände, die bei der Abarbeitung von y durchlaufen werden.

Verallgemeinertes PL für reguläre Sprachen

Karlsruher Institut für Technologie

- Betrachte $pys \in L$ mit $|y| = n$, beliebig. $\rightsquigarrow \forall w = pys \in L, |y| = n$
- Sei $q_0, \ldots, q_n$ die Folge der $n + 1$ Zustände, die bei der Abarbeitung von y durchlaufen werden.
- Da $n + 1 > n = |Q|$, enthält $q_0, \ldots, q_n$ mindestens einen Zykel.
- Wähle Darstellung $y = uvx$ so dass v dem Teilwort entspricht das beim Durchlaufen des Zykels abgearbeitet wird.
  $\rightsquigarrow \exists u, v, x, y = uvx$
- Insbesondere ist v nicht leer. $\rightsquigarrow v \neq \varepsilon$
- Betrachte $i \in \mathbb{N}_0$, beliebig. $\rightsquigarrow \forall i \in \mathbb{N}_0$
- Der Zykel kann auch i Mal durchlaufen werden, ohne den Endzustand zu ändern.
- Also erkennt der Automat auch puv^ixs. $\rightsquigarrow puv^ixs \in L$

Beispiel (3) - Anwendung Verallgemeinertes PL

Karlsruher Institut für Technologie

Aussage des verallgemeinerten PL für Sprache L:

$\exists n \, \forall w \in L, w = pys, |y| = n \, \exists uvx = y, v \neq \varepsilon \, \forall i \in \mathbb{N}_0 : puv^i xs \in L$

Durch **Widerlegen** der Aussage des verallgemeinerten PL für eine gegebene Sprache L zeigen wir, dass L **nicht regulär** ist.

Beispiel (3)

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

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

"$\exists$" Wähle $w = 0^n 1^{n^2}$ mit der Zerlegung $p = 0^n$, $y = 1^n$, $s = 1^{n^2 - n}$.

"$\forall$" Betrachte beliebige Zerlegung $y = uvx$, $v \neq \varepsilon$.

"$\exists$" Wähle $i = 2$.

"gilt" Da $v = 1^a$ für ein $1 \leq a \leq n$, ist $puv^2xs = 0^n 1^{n^2 + a} \notin L$.

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

$\rightsquigarrow$ L ist nicht regulär.

Kapitel

- **Minimierung von Automaten**
- **Äquivalenzklassenautomat**

Minimierung von Automaten

Frage: Kann man konstruktiv die Anzahl der Zustände eines deterministischen endlichen Automatens erheblich verringern?

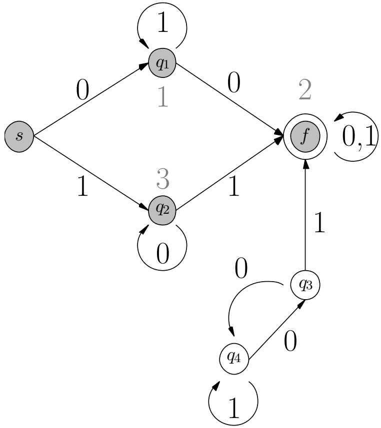
Minimierung von Automaten

Frage: Kann man konstruktiv die Anzahl der Zustände eines deterministischen endlichen Automatens erheblich verringern?

Definition.

Zustände eines (deterministischen) endlichen Automatens, die vom Anfangszustand aus nicht erreichbar sind, heißen **überflüssig**.

Beispiel



Finden nicht überflüssiger Zustände

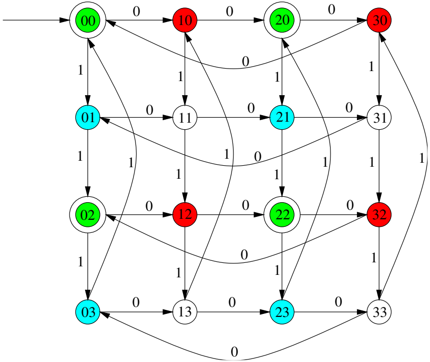
- Wir können endliche Automaten als gerichtete Graphen auffassen.
- Die überflüssigen Zustände entsprechen dann den Knoten, zu denen es vom Anfangsknoten aus keinen gerichteten Weg gibt.
- Eine Tiefensuche (**D**epth–**F**irst **S**earch, DFS) in dem Graphen liefert damit alle nicht überflüssigen Zustände.

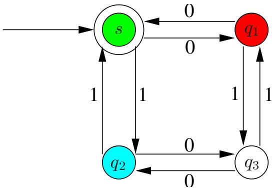
Satz.
Die Menge aller überflüssigen Zustände eines (deterministischen) endlichen Automaten kann in der Zeit $\mathcal{O}(|Q| \cdot |\Sigma|)$ berechnet werden.

Beweis: Wende DFS ab dem Startzustand an. Dies erfordert einen Aufwand proportional zu der Anzahl der Kanten in dem Graphen.

- Ein deterministischer endlicher Automat ohne überflüssige Zustände muss jedoch noch nicht minimal sein.

Beispiel





Beide Automaten akzeptieren die Sprache

$$L = \{w \in \{0,1\}^* : (|w|_0 \mod 2) = (|w|_1 \mod 2) = 0\}$$

mit $|w|_a$ = Anzahl der Vorkommen des Zeichens $a \in \Sigma$ in w.

Äquivalenz

- Zwei Zustände haben dasselbe Akzeptanzverhalten, wenn es für das Erreichen eines Endzustandes durch Abarbeiten eines Wortes w unerheblich ist, aus welchem der beiden Zustände wir starten.
- Reduktion der Anzahl der Zustände durch Zusammenlegen der Zustände mit gleichem Akzeptanzverhalten
- Letztes Beispiel: Färbung der Zustände mit gleichem Verhalten durch gleiche Farben

Äquivalenz

- Zwei Zustände haben dasselbe Akzeptanzverhalten, wenn es für das Erreichen eines Endzustandes durch Abarbeiten eines Wortes w unerheblich ist, aus welchem der beiden Zustände wir starten.
- Reduktion der Anzahl der Zustände durch Zusammenlegen der Zustände mit gleichem Akzeptanzverhalten
- Letztes Beispiel: Färbung der Zustände mit gleichem Verhalten durch gleiche Farben

Definition.

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen **äquivalent** ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Offensichtlich ist $\equiv$ eine Äquivalenzrelation. Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

Der Äquivalenzklassenautomat

Karlsruher Institut für Technologie

Äquivalenzklassenautomat

Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

- $Q^{\equiv} := \{[q] \colon q \in Q\}$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $F^{\equiv} := \{[f] \colon f \in F\}$
- $\Sigma^{\equiv} := \Sigma$
- $s^{\equiv} := [s]$

Der Äquivalenzklassenautomat

Äquivalenzklassenautomat

Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

- $Q^{\equiv} := \{[q] \colon q \in Q\}$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $F^{\equiv} := \{[f] \colon f \in F\}$
- $\Sigma^{\equiv} := \Sigma$
- $s^{\equiv} := [s]$

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu einem deterministischen endlichen Automaten $\mathcal{A}$ ist wohldefiniert.

Der Äquivalenzklassenautomat

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu einem deterministischen endlichen Automaten $\mathcal{A}$ ist wohldefiniert.

Der Äquivalenzklassenautomat

Karlsruher Institut für Technologie

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu einem deterministischen endlichen Automaten $\mathcal{A}$ ist wohldefiniert.

Beweis: Wir müssen zeigen, dass $F^{\equiv}$ und $\delta^{\equiv}$ wohldefiniert sind, der Rest ist klar. Dazu zeigen wir:

- ein Endzustand kann nur zu einem Endzustand äquivalent sein,
- δ führt äquivalente Zustände beim Lesen desselben Symbols wieder in äquivalente Zustände über.

Der Äquivalenzklassenautomat

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu einem deterministischen endlichen Automaten $\mathcal{A}$ ist wohldefiniert.

- ein Endzustand kann nur zu einem Endzustand äquivalent sein,

Es gilt

$$\delta(p, \varepsilon), \delta(q, \varepsilon) \in F \text{ genau für } p, q \in F \ .$$

Also:

$$\text{Falls } p \equiv q, \text{ dann gilt } p, q \in F \text{ oder } p, q \notin F.$$

Also ist $F^{\equiv}$ wohldefiniert.

Der Äquivalenzklassenautomat

Karlsruher Institut für Technologie

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu einem deterministischen endlichen Automaten $\mathcal{A}$ ist wohldefiniert.

- δ führt äquivalente Zustände beim Lesen desselben Symbols wieder in äquivalente Zustände über.

Sei $p \equiv q$. Dann gilt für alle $w \in \Sigma^*$

$$\delta(q, w) \in F \Leftrightarrow \delta(p, w) \in F$$

Somit gilt nach Definition von $\equiv$ auch für alle $a \in \Sigma$:

$$\delta(\delta(q, a), w) = \delta(q, aw) \in F \Leftrightarrow \delta(p, aw) = \delta(\delta(p, a), w) \in F.$$

Damit folgt $\delta(q, a) \equiv \delta(p, a)$, also ist auch $\delta^{\equiv}$ wohldefiniert.

Der Äquivalenzklassenautomat

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu $\mathcal{A}$ akzeptiert dieselbe Sprache wie $\mathcal{A}$.

Der Äquivalenzklassenautomat

Satz.

Der Äquivalenzklassenautomat $\mathcal{A}^{\equiv}$ zu $\mathcal{A}$ akzeptiert dieselbe Sprache wie $\mathcal{A}$.

Beweis:

- Sei $w \in \Sigma^*$, $q_0 := s, q_1, \ldots, q_n$ die Folge der Zustände, die von $\mathcal{A}$ bei der Abarbeitung von w durchlaufen werden.
- Es gilt nach Definition von $\delta^{\equiv}$:

$$\delta(q, a) = p \quad \Longrightarrow \quad \delta^{\equiv}([q], a) = [\delta(q, a)] = [p].$$

- Bei Abarbeitung von w in $\mathcal{A}^{\equiv}$ werden dann die Zustände $[q_0], [q_1], \ldots, [q_n]$ durchlaufen.
- $\mathcal{A}$ akzeptiert w genau dann, wenn $q_n \in F$ gilt. $\mathcal{A}^{\equiv}$ akzeptiert w genau dann, wenn $[q_n] \in F^{\equiv}$ gilt.
- Nach Definition von $\mathcal{A}^{\equiv}$ ist $q_n \in F$ genau dann, wenn $[q_n] \in F^{\equiv}$ gilt.

Frage

Wie konstruiert man $\mathcal{A}^{\equiv}$ zu $\mathcal{A}$? D.h. wie berechnet man alle Äquivalenzklassen zu den Zuständen von $\mathcal{A}$?

Beweis der Äquivalenz von zwei Zuständen p scheint aufwendig:
Nach Definition muss nachgewiesen werden, dass für alle $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \Longleftrightarrow \delta(q, w) \in F.$$

- Es gibt jedoch unendlich viele $w \in \Sigma^*$.
- Es ist einfacher für p und q zu zeigen, dass p nicht äquivalent zu q ist.
- Dafür benötigen wir *nur ein* Wort $w \in \Sigma^*$ mit

$$\begin{array}{c} \delta(p, w) \in F \\ \text{aber } \delta(q, w) \notin F \end{array} \quad \text{oder} \quad \begin{array}{c} \delta(p, w) \notin F \\ \text{aber } \delta(q, w) \in F. \end{array}$$

Zeugen für Nichtäquivalenz

Seien $p, q \in Q$, $w \in \Sigma^*$ mit

$$\begin{array}{c} \delta(p, w) \in F \\ \text{aber } \delta(q, w) \notin F \end{array} \quad \text{oder} \quad \begin{array}{c} \delta(p, w) \notin F \\ \text{aber } \delta(q, w) \in F. \end{array}$$

Notation.

Wir bezeichnen ein solches Wort w als **Zeuge** für die Nichtäquivalenz von p und q und sagen w trennt p und q.

Idee: Teste systematisch Zustandspaare auf Nichtäquivalenz

- Betrachte alle Wörter aus Σ^* in aufsteigender Länge.
- Überprüfe für jedes Wort, ob es Zeuge für Nichtäquivalenz von zwei Zuständen ist.

Abbruchkriterium

- Betrachte alle Wörter aus Σ^* in aufsteigender Länge.
- Überprüfe für jedes Wort, ob es Zeuge für Nichtäquivalenz von zwei Zuständen ist.

Wann kann dieses Verfahren abgebrochen werden?

Abbruchkriterium

- Betrachte alle Wörter aus Σ^* in aufsteigender Länge.
- Überprüfe für jedes Wort, ob es Zeuge für Nichtäquivalenz von zwei Zuständen ist.

Wann kann dieses Verfahren abgebrochen werden?

- Sei $w = aw'$ ein kürzester Zeuge für $p \not\equiv q$.
- Dann ist w' Zeuge für $p' := \delta(p, a) \not\equiv \delta(q, a) =: q'$.
- Wenn es für $p' \not\equiv q'$ einen kürzeren Zeugen w'' gäbe, so wäre aw'' ein kürzerer Zeuge für $p \not\equiv q$ als w.
- Dies wäre ein Widerspruch dazu, dass w ein kürzester Zeuge ist.

Fazit: Wenn wir alle Wörter aus Σ^* in der Reihenfolge ihrer Länge darauf testen, ob sie Zeuge sind, und für eine bestimmte Länge kein Zeuge mehr für eine Nichtäquivalenz auftritt, sind wir fertig.

Vorgehensweise

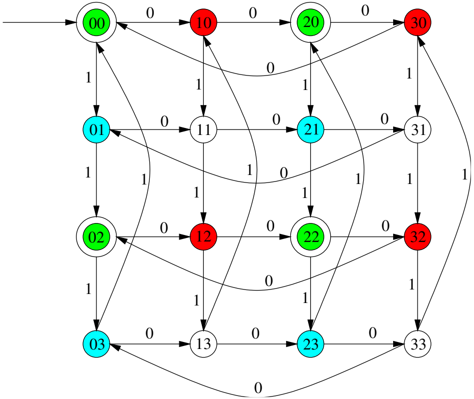
Vorgehensweise für die Konstruktion von $\mathcal{A}^{\equiv}$ aus $\mathcal{A}$

- Betrachte alle Zustandspaare und zunächst ε,
- dann alle Elemente aus Σ,
- dann alle Wörter der Länge 2 aus Σ^*,
- u.s.w.

Zunächst betrachte alle Zustände als eine Klasse.

- Dann trennt ε die Zustände aus F von denen aus $Q \setminus F$.
- Danach testen wir nur noch Paare von Zuständen aus F beziehungsweise $Q \setminus F$.
- Durch mindestens ein Wort der Länge 1 wird entweder F oder $Q \setminus F$ weiter getrennt, oder das Verfahren ist beendet.
- Dies wird iterativ so weitergeführt mit Wörtern wachsender Länge.

Beispiel zur Vorgehensweise



Vorläufige Äquivalenzklassen
vorher

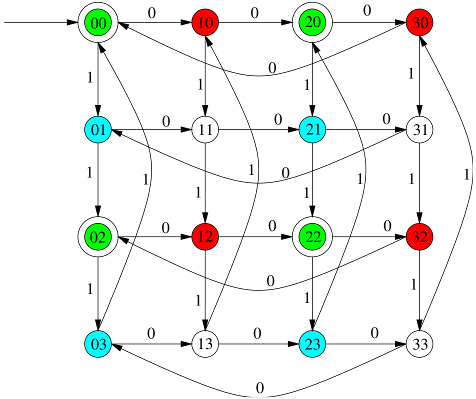
$\{00, 01, 02, 03, 10, 11, 12, 13, 20, 21, 22, 23, 30, 31, 32, 33\}$

nachher

$\{00, 02, 20, 22\}$ $\{01, 03, 10, 11, 12, 13, 21, 23, 30, 31, 32, 33\}$

ε trennt $\underbrace{\{00, 02, 20, 22\}}_{\text{grün}}$ von $\{01, 03, 10, 11, 12, 13, 21, 23, 30, 31, 32, 33\}$

Beispiel zur Vorgehensweise



Vorläufige Äquivalenzklassen

vorher

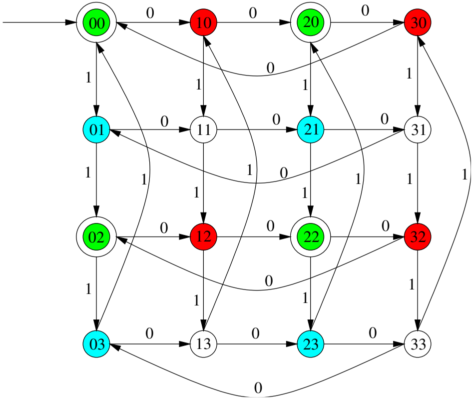
$\{00, 02, 20, 22\}$ $\{01, 03, 10, 11, 12, 13, 21, 23, 30, 31, 32, 33\}$

nachher

$\{00, 02, 20, 22\}$ $\{10, 30, 12, 32\}$ $\{01, 03, 11, 13, 21, 23, 31, 33\}$

0 trennt $\underbrace{\{10, 30, 12, 32\}}_{\text{rot}}$ von $\{01, 03, 11, 13, 21, 23, 31, 33\}$

Beispiel zur Vorgehensweise



Vorläufige Äquivalenzklassen

vorher

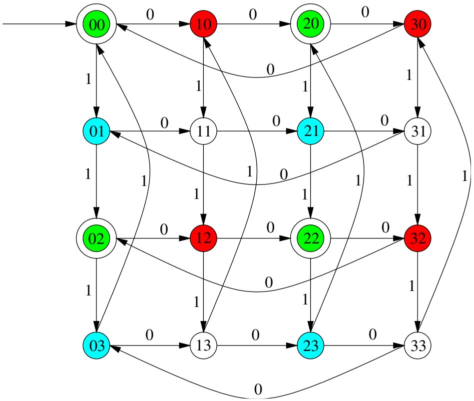
$\{00, 02, 20, 22\}$ $\{10, 30, 12, 32\}$
$\{01, 03, 11, 13, 21, 23, 31, 33\}$

nachher

$\{00, 02, 20, 22\}$ $\{10, 30, 12, 32\}$
$\{01, 03, 21, 23\}$ $\{11, 13, 31, 33\}$

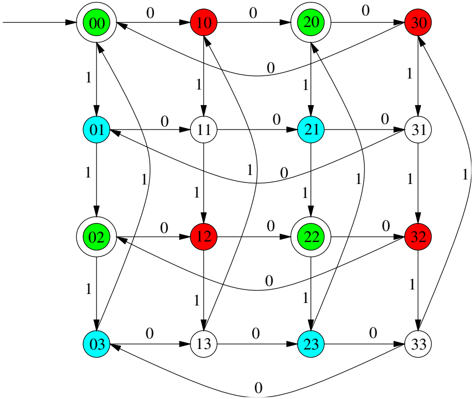
1 trennt $\underbrace{\{01, 03, 21, 23\}}_{\text{blau}}$ von $\underbrace{\{11, 13, 31, 33\}}_{\text{weiß}}$

Beispiel zur Vorgehensweise



Die Wörter 00, 01, 10, 11 trennen keine Zustandspaare mehr.

Beispiel zur Vorgehensweise



Äquivalenzklassen

$\{00, 02, 20, 22\}$ $\{10, 30, 12, 32\}$
$\{01, 03, 21, 23\}$ $\{11, 13, 31, 33\}$

Fazit: Die Äquivalenzklassen der Zustände sind:
$s = [00]$, $q_1 = [01]$, $q_2 = [10]$ und $q_3 = [11]$.

Zusammenfassung

Aussage des Verallgemeinerten Pumping-Lemmas:

$$\exists n\ \forall w \in L, w = pys, |y| = n\ \exists uvx = y, v \neq \varepsilon\ \forall i \in \mathbb{N}_0: \qquad puv^i xs \in L \qquad (\star\star)$$

- Verallgemeinertes PL: L regulär $\Longrightarrow$ L erfüllt $(\star\star)$
- Widerlegen der Aussage des Lemmas beweist Nicht-Regularität:

  L erfüllt $(\star\star)$ nicht $\Longrightarrow$ L ist nicht regulär

Zusammenfassung

Karlsruher Institut für Technologie

Aussage des Verallgemeinerten Pumping-Lemmas:

$$\exists n \, \forall w \in L, w = pys, |y| = n \, \exists uvx = y, v \neq \varepsilon \, \forall i \in \mathbb{N}_0 : \qquad puv^i xs \in L \qquad (\star\star)$$

- Verallgemeinertes PL: L regulär $\Longrightarrow$ L erfüllt $(\star\star)$
- Widerlegen der Aussage des Lemmas beweist Nicht-Regularität:

  L erfüllt $(\star\star)$ nicht $\Longrightarrow$ L ist nicht regulär

Äquivalenzklassenautomat

<u>Idee:</u> Reduziere die Anzahl der Zustände in DEA $\mathcal{A}$.
<u>Definition:</u> Äquivalente Zustände und $\mathcal{A}^{\equiv}$ zu gegebenen DEA $\mathcal{A}$.
<u>Satz:</u> $\mathcal{A}^{\equiv}$ ist wohldefiniert und $L(\mathcal{A}^{\equiv}) = L(\mathcal{A})$.
<u>Konstruktion:</u> Teste Nicht-Äquivalenz mit Wörter aufsteigender Länge.

Testen Sie sich!

Testen Sie sich:

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

Betrachte $L_1 = \{(0^+1)^{k^2} \mid k > 0\}$ und $L_2 = \{0^+1^{k^2} \mid k > 0\}$.

$\rightsquigarrow$ Gilt die Aussage des verallgemeinerten PL für L_1 und/oder L_2?

$\rightsquigarrow$ Ist L_1 und/oder L_2 regulär?

Testen Sie sich!

Karlsruher Institut für Technologie

Testen Sie sich:

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

Betrachte $L_1 = \{(0^+1)^{k^2} \mid k > 0\}$ und $L_2 = \{0^+1^{k^2} \mid k > 0\}$.

$\rightsquigarrow$ Gilt die Aussage des verallgemeinerten PL für L_1 und/oder L_2?

$\rightsquigarrow$ Ist L_1 und/oder L_2 regulär?

Sei $\mathcal{A} = (Q = \{q_0, \ldots, q_9\}, \Sigma = \{0, \ldots, 9\}, \delta, s = q_0, F = \{q_0, q_3, q_6, q_9\})$ gegeben durch:

$$\delta(q_i, a) = \begin{cases} q_{i+a} & \text{falls } i + a \leq 9 \\ q_{i+a-10} & \text{falls } i + a \geq 10. \end{cases}$$

$\rightsquigarrow$ Wieviele Zustände hat $\mathcal{A}^{\equiv}$?

Bonus: Finden Sie $\mathcal{A}^{\equiv}$? Finden Sie $L(\mathcal{A}^{\equiv})$?