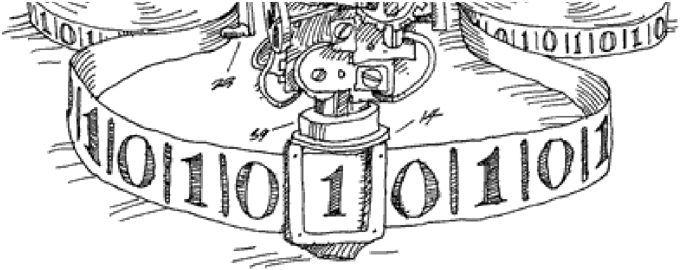
Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Tutorium 2

Institut für Theoretische Informatik



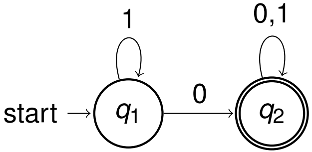
Konstruktion eines RA aus einem DEA

Wir wissen: Zu jedem DEA gibt es einen regulären Ausdruck, der genau die Sprache beschreibt, die der Automat akzeptiert. Wie konstruiert man nun diesen RA aus dem DEA?

Idee: Betrachte die Sprachen L_{q_r,i,q_t}, definiert als $w \in \Sigma^*$ mit w überführt q_r in q_t unter Benutzung der Zwischenzustände $\{q_1, \ldots, q_i\}$

- Es ist $L = \cup_{f \in F} L_{s,n,f}$
- Es ist weiterhin $L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup (L_{q_r,i,q_{i+1}}(L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t})$
- Letztlich ist $L_{q_r,0,q_t}$ immer regulär, denn das sind die Zeichen, mit denen man von q_r nach q_t kommt, ohne weitere Zustände zu verwenden (sowie ε, falls $r = t$).
- Unter Benutzung dieser Punkte kann man nun zu einem DEA einen regulären Ausdruck konstruieren.

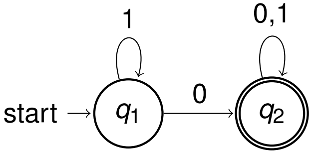
Beispiele zum Verständnis



Was sind hier jeweils:

- $L_{q_1,0,q_1}$

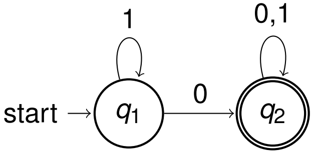
Beispiele zum Verständnis



Was sind hier jeweils:

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

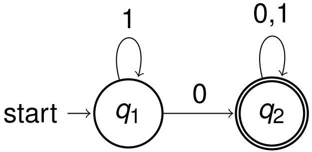
Beispiele zum Verständnis



Was sind hier jeweils:

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

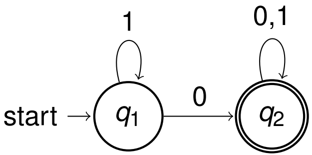
Beispiele zum Verständnis



Was sind hier jeweils:

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

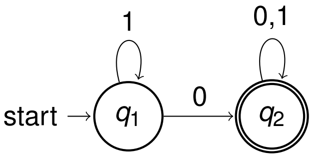
Beispiele zum Verständnis



Was sind hier jeweils:

- $L_{q_1,0,q_1} = (1 \cup \varepsilon)$
- $L_{q_1,0,q_2} = (0)$
- $L_{q_1,1,q_1} = (1^*)$
- $L_{q_2,1,q_2} = (0 \cup 1 \cup \varepsilon)$
- $L_{q_2,2,q_2}$

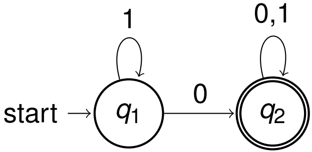
Beispiele zum Verständnis



Was sind hier jeweils:

- $L_{q_1,0,q_1} = (1 \cup \varepsilon)$
- $L_{q_1,0,q_2} = (0)$
- $L_{q_1,1,q_1} = (1^*)$
- $L_{q_2,1,q_2} = (0 \cup 1 \cup \varepsilon)$
- $L_{q_2,2,q_2} = (0 \cup 1)^*$
- $L_{q_1,2,q_2}$

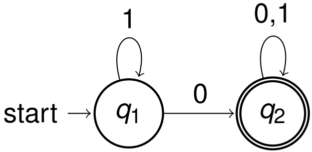
Beispiele zum Verständnis



Was sind hier jeweils:

- $L_{q_1,0,q_1} = (1 \cup \varepsilon)$
- $L_{q_1,0,q_2} = (0)$
- $L_{q_1,1,q_1} = (1^*)$
- $L_{q_2,1,q_2} = (0 \cup 1 \cup \varepsilon)$
- $L_{q_2,2,q_2} = (0 \cup 1)^*$
- $L_{q_1,2,q_2} = 1^*0(0 \cup 1)^*$

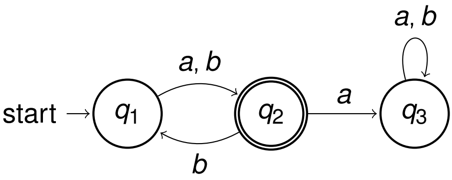
Ausführliche Konstruktion



1. $L_{q_1,2,q_2} = L_{q_1,1,q_2} \cup (L_{q_1,1,q_2}(L_{q_2,1,q_2})^* L_{q_2,1,q_2})$
2. $L_{q_1,1,q_2} = L_{q_1,0,q_2} \cup (L_{q_1,0,q_1}(L_{q_1,0,q_1})^* L_{q_1,0,q_2}) = 0 \cup ((1 \cup \varepsilon)(1 \cup \varepsilon)^* 0)$
3. $L_{q_2,1,q_2} = L_{q_2,0,q_2} \cup (L_{q_2,0,q_1}(L_{q_1,0,q_1})^* L_{q_1,0,q_2}) = (0 \cup 1 \cup \varepsilon)$
4. Also: $L_{q_1,2,q_2} =$
   $0 \cup (1 \cup \varepsilon)(1 \cup \varepsilon)^* 0 \cup ((0 \cup ((1 \cup \varepsilon)(1 \cup \varepsilon)^* 0))(0 \cup 1 \cup \varepsilon)^*(0 \cup 1 \cup \varepsilon))$
5. Vereinfacht: $1^* 0 (0 \cup 1)^*$

Aufgabe

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14 im Skript zur Vorlesung) einen regulären Ausdruck für die von folgendem Automaten erkannte Sprache. Geben Sie dabei alle benötigten Zwischenergebnisse an und lesen Sie nur Sprachen der Form $L_{r,0,t}$ direkt ab.
Hinweis: Wenn Sie frühzeitig $L_{q_3,2,q_2}$ berechnen, können Sie sich einige Rechenschritte sparen.



Pumping Lemma

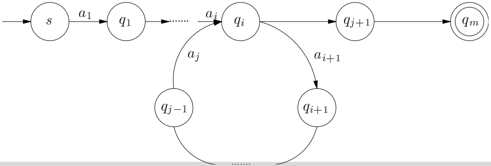
Pumping Lemma

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

$$w = uvx$$

existiert, so dass folgende Eigenschaften erfüllt sind:

1. $v \neq \varepsilon$
2. $|uv| \leq n$
3. Für alle $i \in \mathbb{N}_0$ gilt: $uv^i x \in L$



Pumping Lemma: Übersicht

- Jede reguläre Sprache erfüllt das Pumping Lemma. Aber: Nicht jede Sprache, die das Pumping Lemma erfüllt, ist regulär!
- In der Übung wird üblicherweise die Kontraposition des Pumping-Lemmas verwendet: Man zeigt für eine Sprache, dass das Pumping-Lemma *nicht* erfüllt ist, woraus folgt, dass diese Sprache *nicht* regulär sein kann.
- $\neg \left[\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \ldots \forall i \in \mathbb{N} : uv^i x \in L \right]$
  $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \ldots \exists i \in \mathbb{N} : uv^i x \notin L$
  - Finden wir für *jedes n ein w* mit $|w| > n$, so dass für *jede* Darstellung $w = uvx$ mit $v \neq \varepsilon$ sowie $|uv| \leq n$ ein $i \in \mathbb{N}_0$ existiert mit $uv^i x \notin L$, dann ist L nicht regulär.

Karlsruher Institut für Technologie

Pumping Lemma: Übersicht

- Jede reguläre Sprache erfüllt das Pumping Lemma. Aber: Nicht jede Sprache, die das Pumping Lemma erfüllt, ist regulär!
- In der Übung wird üblicherweise die Kontraposition des Pumping-Lemmas verwendet: Man zeigt für eine Sprache, dass das Pumping-Lemma *nicht* erfüllt ist, woraus folgt, dass diese Sprache *nicht* regulär sein kann.
- $\neg \left[\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \ldots \forall i \in \mathbb{N} : uv^i x \in L\right]$
  $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \ldots \exists i \in \mathbb{N} : uv^i x \notin L$
  - Finden wir für *jedes n ein w* mit $|w| > n$, so dass für *jede* Darstellung $w = uvx$ mit $v \neq \varepsilon$ sowie $|uv| \leq n$ ein $i \in \mathbb{N}_0$ existiert mit $uv^i x \notin L$, dann ist L nicht regulär.

Pumping Lemma: Übersicht

- Jede reguläre Sprache erfüllt das Pumping Lemma. Aber: Nicht jede Sprache, die das Pumping Lemma erfüllt, ist regulär!
- In der Übung wird üblicherweise die Kontraposition des Pumping-Lemmas verwendet: Man zeigt für eine Sprache, dass das Pumping-Lemma *nicht* erfüllt ist, woraus folgt, dass diese Sprache *nicht* regulär sein kann.
- $\neg\left[\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \ldots \forall i \in \mathbb{N} : uv^i x \in L\right]$
  $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \ldots \exists i \in \mathbb{N} : uv^i x \notin L$
  - Finden wir für *jedes n ein w* mit $|w| > n$, so dass für *jede* Darstellung $w = uvx$ mit $v \neq \varepsilon$ sowie $|uv| \leq n$ ein $i \in \mathbb{N}_0$ existiert mit $uv^i x \notin L$, dann ist L nicht regulär.

Beispiel

KIT Karlsruher Institut für Technologie

Sei $\Sigma = \{a, b\}$ und $L = \{a^n b^n \mid n \geq 0\}$. (Also $L = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$)

1. Sei $n \in \mathbb{N}$ beliebig, aber fest.
2. Wähle $w = a^n b^n$.
3. Es ist also $|w| > n$.
4. Nun ist aber für *jede* Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ $v = a^m$ mit $m \geq 1$. Demnach ist $uv^0x = a^l b^n \neq L$, da $l < n$.
5. Daher kann L nicht regulär sein.

Beispiel

Sei $\Sigma = \{a, b\}$ und $L = \{a^n b^n \mid n \geq 0\}$. (Also $L = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$)

1. Sei $n \in \mathbb{N}$ beliebig, aber fest.
2. Wähle $w = a^n b^n$.
3. Es ist also $|w| > n$.
4. Nun ist aber für *jede* Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ $v = a^m$ mit $m \geq 1$. Demnach ist $uv^0x = a^l b^n \neq L$, da $l < n$.
5. Daher kann L nicht regulär sein.

Beispiel

Sei $\Sigma = \{a, b\}$ und $L = \{a^n b^n \mid n \geq 0\}$. (Also $L = \{\varepsilon, ab, aabb, aaabbb, \dots\}$)

1. Sei $n \in \mathbb{N}$ beliebig, aber fest.
2. Wähle $w = a^n b^n$.
3. Es ist also $|w| > n$.
4. Nun ist aber für *jede* Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ $v = a^m$ mit $m \geq 1$. Demnach ist $uv^0x = a^l b^n \neq L$, da $l < n$.
5. Daher kann L nicht regulär sein.

Beispiel

Karlsruher Institut für Technologie

Sei $\Sigma = \{a, b\}$ und $L = \{a^n b^n \mid n \geq 0\}$. (Also $L = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$)

1. Sei $n \in \mathbb{N}$ beliebig, aber fest.
2. Wähle $w = a^n b^n$.
3. Es ist also $|w| > n$.
4. Nun ist aber für *jede* Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ $v = a^m$ mit $m \geq 1$. Demnach ist $uv^0x = a^l b^n \neq L$, da $l < n$.
5. Daher kann L nicht regulär sein.

Beispiel

Sei $\Sigma = \{a, b\}$ und $L = \{a^n b^n \mid n \geq 0\}$. (Also $L = \{\varepsilon, ab, aabb, aaabbb, \dots\}$)

1. Sei $n \in \mathbb{N}$ beliebig, aber fest.
2. Wähle $w = a^n b^n$.
3. Es ist also $|w| > n$.
4. Nun ist aber für *jede* Darstellung $w = uvx$ mit $|uv| \leq n$ und $v \neq \varepsilon$ $v = a^m$ mit $m \geq 1$. Demnach ist $uv^0x = a^l b^n \neq L$, da $l < n$.
5. Daher kann L nicht regulär sein.

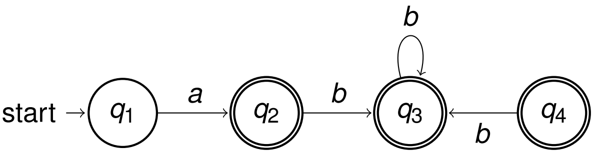
Pumping Lemma: Aufgaben

Welche der folgenden Sprachen sind regulär? Begründen Sie Ihre Antwort.

1. Die Menge aller Wörter über $\{0, 1\}$, sodass auf jede Null eine Eins folgt
2. $L_1 = \{ww^R | w \in \{a, b\}^*\}$, wobei w^R das 'Spiegelwort' zu w ist (Sprache der Palindrome gerader Länge)
3. $L_2 = \{a^i b^j c^k \mid i < j < k\}$.

Minimierung von DEAs

Beispielautomat $A = (Q, \Sigma, \delta, s, F)$

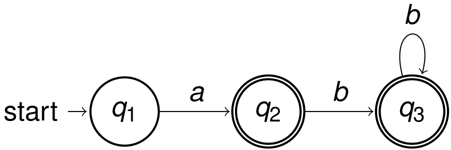


Kann A auf einen Automaten $A' = (Q', \Sigma, \delta', s', F')$ mit $L(A) = L(A')$ und $|Q'| < |Q|$ überführt werden?

Minimierung von DEAs

q_4 ist nicht vom Startzustand aus erreichbar

Entferne q_4

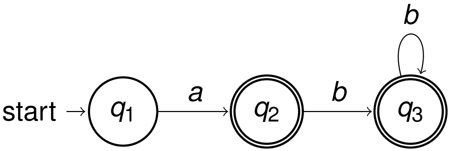


Schon minimal?

Minimierung von DEAs

q_4 ist nicht vom Startzustand aus erreichbar

Entferne q_4

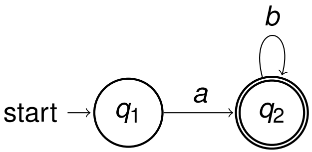


Schon minimal?

Minimierung von DEAs

Nein!

Automat $A'' = (Q'', \Sigma, \delta'', s, F'')$



Akzeptierte Sprache: $L(A) = L(A') = L(A'') = L(ab^*)$

Methodischer Ansatz

Definition

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

Vorgehen

1. Tiefensuche durchführen, um überflüssige Zustände zu finden.
2. Diese entfernen.
3. Aus restlichen Zuständen den Äquivalenzklassenautomaten bilden.

Äquivalenz

Aus der Vorlesung

- 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.

Definition (Äquivalenz):

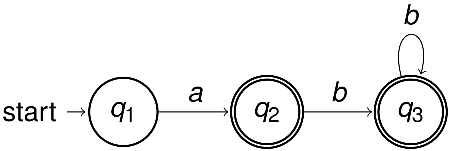
Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

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

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

Beispiel

Zurück zu Q'



Hier sind q_2 und q_3 äquivalent. Warum?

Äquivalenzklassenautomat

Definition aus der Vorlesung (Äquivalenzklassenautomat)

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

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

Konstruktion der Äquivalenzklassen

1. Fasse alle Zustände $q_i \in Q$ in einer Klasse zusammen.
2. ε trennt Zustände aus F von denen aus $Q \setminus F$.
3. Für Worte $w \in \Sigma^*$ mit wachsender Länge und Zustandspaare p, q in einer Klasse:
   3.1 Falls $[\delta(p, w)] \neq [\delta(q, w)]$, trenne die Zustände q und p (w ist Zeuge).
   3.2 Brich ab, falls sich für eine Wortlänge keine weiteren Zeugen finden.

Toller:

3. Für Zeichen $x \in \Sigma$ und Zustandspaare p, q in einer Klasse:
   3.1 Falls $[\delta(p, x)] \neq [\delta(q, x)]$, trenne die Zustände q und p .
   3.2 Brich ab, falls in einem Durchlauf kein Zeichen mehr Zustände trennt.

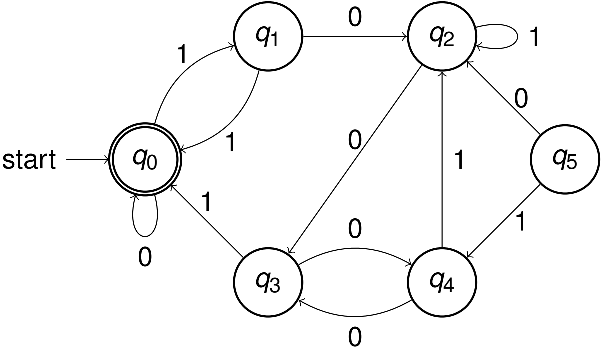
Konstruktion der Äquivalenzklassen

1. Fasse alle Zustände $q_i \in Q$ in einer Klasse zusammen.
2. ε trennt Zustände aus F von denen aus $Q \setminus F$.
3. Für Worte $w \in \Sigma^*$ mit wachsender Länge und Zustandspaare p, q in einer Klasse:
   3.1 Falls $[\delta(p, w)] \neq [\delta(q, w)]$, trenne die Zustände q und p (w ist Zeuge).
   3.2 Brich ab, falls sich für eine Wortlänge keine weiteren Zeugen finden.

Toller:

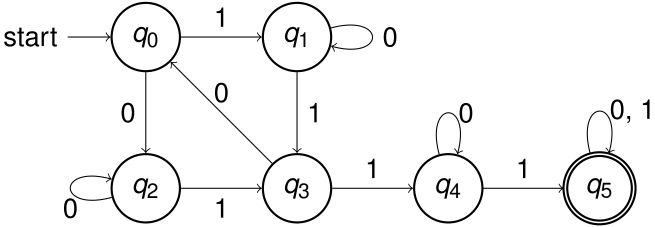
3. Für Zeichen $x \in \Sigma$ und Zustandspaare p, q in einer Klasse:
   3.1 Falls $[\delta(p, x)] \neq [\delta(q, x)]$, trenne die Zustände q und p .
   3.2 Brich ab, falls in einem Durchlauf kein Zeichen mehr Zustände trennt.

Automaten-Minimierung: Beispiel



Automaten-Minimierung: Aufgabe

Konstruiere zu folgendem DEA den Äquivalenzklassenautomaten:



Aufgabe zu DEAs

Karlsruher Institut für Technologie

Gegeben sei ein DEA (als Graph mit beschrifteten Knoten und Kanten) ohne überflüssige Zustände. Schreibe einen Algorithmus, der entscheidet, ob die zugehörige Sprache unendlich viele Wörter hat. Hinweis: Siehe Beweis zur Korrektheit des Pumping-Lemmas.

Aufgabe: Regularität

Karlsruher Institut für Technologie

Definition

Ein Wort w ist *Präfix* eines Wortes w', falls ein Wort v mit $wv = w'$ existiert. Gilt zusätzlich $w \neq w'$, so ist w ein *echtes Präfix* von w'.

Aufgabe

Sei L eine reguläre Sprache. Zeige, dass dann auch die folgenden beiden Sprachen regulär sind:

1. $\mathrm{NOPREFIX}(L) := \{w \in L \mid \text{kein } w' \in L \text{ ist echtes Präfix von } w\}$.
2. $\mathrm{NOEXTEND}(L) :=$
$$\{w \in L \mid w \text{ ist kein echtes Präfix eines Wortes } w' \in L\}.$$

Zur Veranschaulichung

$L = \{$ Katze, Kater, Katzenpfote, Katzenfutter $\}$
Was sind hier $\mathrm{NOPREFIX}(L)$ und $\mathrm{NOEXTEND}(L)$?

Aufgabe: Regularität

Definition

Ein Wort w ist *Präfix* eines Wortes w', falls ein Wort v mit $wv = w'$ existiert. Gilt zusätzlich $w \neq w'$, so ist w ein *echtes Präfix* von w'.

Aufgabe

Sei L eine reguläre Sprache. Zeige, dass dann auch die folgenden beiden Sprachen regulär sind:

1. $\text{NOPREFIX}(L) := \{w \in L \mid \text{kein } w' \in L \text{ ist echtes Präfix von } w\}$.
2. $\text{NOEXTEND}(L) :=$
   $\{w \in L \mid w \text{ ist kein echtes Präfix eines Wortes } w' \in L\}$.

Zur Veranschaulichung

$L = \{$ Katze, Kater, Katzenpfote, Katzenfutter $\}$
Was sind hier NOPREFIX(L) und NOEXTEND(L)?

Aufgabe: Regularität

Definition

Ein Wort w ist *Präfix* eines Wortes w', falls ein Wort v mit $wv = w'$ existiert. Gilt zusätzlich $w \neq w'$, so ist w ein *echtes Präfix* von w'.

Aufgabe

Sei L eine reguläre Sprache. Zeige, dass dann auch die folgenden beiden Sprachen regulär sind:

1. $\mathrm{NOPREFIX}(L) := \{w \in L \mid \text{kein } w' \in L \text{ ist echtes Präfix von } w\}$.
2. $\mathrm{NOEXTEND}(L) :=$
$$\{w \in L \mid w \text{ ist kein echtes Präfix eines Wortes } w' \in L\}.$$

Zur Veranschaulichung

$L = \{$ Katze, Kater, Katzenpfote, Katzenfutter $\}$
Was sind hier $\mathrm{NOPREFIX}(L)$ und $\mathrm{NOEXTEND}(L)$?

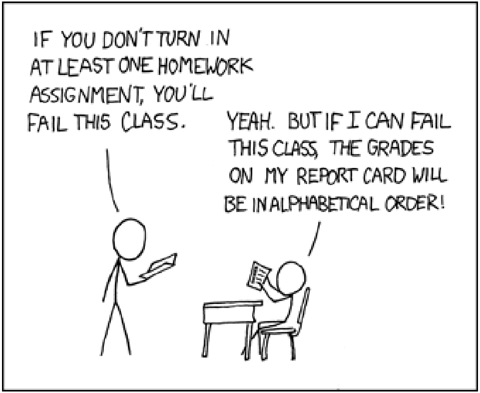
Aufgabe: Regularität

Karlsruher Institut für Technologie

Zeige, dass die folgende Sprache nicht regulär ist:

$$L = \{0^m 1^n \mid m \neq n\}$$

Bis zum nächsten Mal!



You should start giving out 'E's so I can spell FACADE or DEFACED.

Lizenzen