

Theoretische Grundlagen der Informatik

Tutorium 1

Institut für Theoretische Informatik



- **Joachim Priesner**
joachim.priesner@student.kit.edu
Montag 15:45, SR -108
- **Sebastian Ullrich**
sebasti@nullrich.de
Donnerstag 15:45, SR 131
- **Max Wagner**
max@trollbu.de
Donnerstag 15:45, SR 301

- **Abgabe:** *Handschriftlich* in Zweiergruppen
- **Schein:**
 - Klausurbonus (1 Notenschritt)
 - Ab 50% der erreichbaren Punkte
- Tutoriumsmaterial und aktueller Punktestand online
 - <http://tinyurl.com/tgi1112>
 - E-Mail-Liste geht rum für
 - Allgemeines Blabla
 - Passwort für Online-Punkteinsicht

Organisatorisches – Zum Tutorium

- Stoff soll wiederholt werden
- Dabei Fokus auf Übungsbetrieb
- Fragen/Vorschläge/Anmerkungen willkommen!

Eine *formale Sprache* L ist eine Teilmenge aller Wörter über einem endlichen Alphabet Σ . Also $L \subseteq \Sigma^*$.

Beispiele:

- $\Sigma = \{0, 1\}, L = \{w11z \mid w, z \in \Sigma^*\}$
 - Die Menge aller Wörter, die "11" enthalten.

Im Allgemeinen kann man formale Sprachen sehr frei angeben:

- $\Sigma = \{0, 1\}, L = \{w \mid w \in \Sigma^*, w \text{ hat eine gerade Anzahl an 1en}\}$
 - Die Menge aller Wörter, die eine gerade Anzahl an Einsen enthalten.

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

- Verankerung
 - $L = a$ mit $a \in \Sigma^*$ oder
 - $L = \emptyset$
- Induktion: Seien L_1, L_2 reguläre Sprachen.
 - $L = L_1 \cdot L_2$ oder
 - $L = L_1 \cup L_2$ oder
 - $L = L_1^*$

Beispiel ($\Sigma = \{a, b\}$):

- $L_1 = \{w \in \Sigma^* \mid w \text{ besteht aus einer geraden Anzahl } a\}$
- $L_2 = \{w \in \Sigma^* \mid w \text{ enthält gleich viele } a \text{ und } b\}$

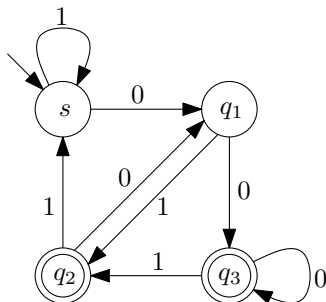
L_1 ist regulär, L_2 nicht.

Deterministische endliche Automaten

Ein deterministischer endlicher Automat M
ist ein 5-Tupel

$$M = (Q, \Sigma, \delta, s, F).$$

- Q : endliche Zustandsmenge
- Σ : endliches Alphabet
- δ : Zustandsübergangsfunktion
 $Q \times \Sigma \rightarrow Q$
- s : Startzustand $\in Q$
- F : Endzustandsmenge $\subseteq Q$

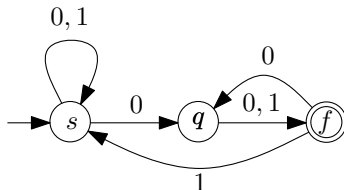


Nichtdeterministische endliche Automaten

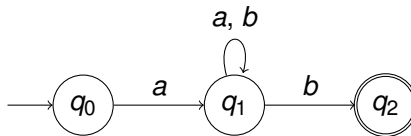
Ein nichtdeterministischer endlicher Automat M ist ein 5-Tupel

$$M = (Q, \Sigma, \delta, s, F).$$

- Q : endliche Zustandsmenge
- Σ : endliches Alphabet
- δ : Zustandsübergangsfunktion
 $Q \times (\Sigma \cup \varepsilon) \rightarrow \mathcal{P}(Q)$
- s : Startzustand $\in Q$
- F : Endzustandsmenge $\subseteq Q$



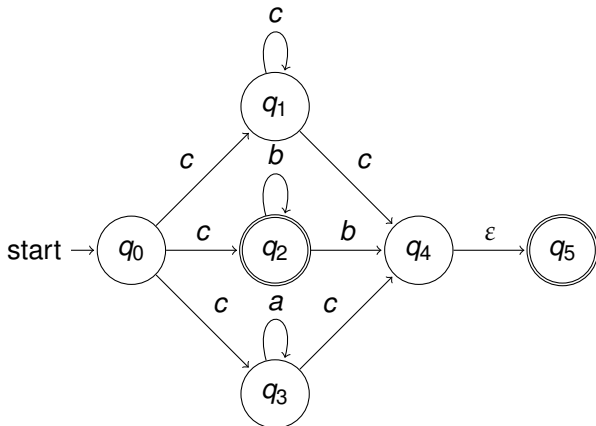
Damit der NEA ein Wort akzeptiert, muss es *einen* akzeptierenden Weg geben.



Bei Eingabe von b im Zustand q_1 gibt es mehrere Möglichkeiten.

(siehe Berechnungsbaum an der Tafel).

Welche Sprache akzeptiert der nichtdeterministische endliche Automat zu dem folgenden Zustandsgraphen?



Über dem Alphabet $\Sigma = \{a, b\}$ sei der reguläre Ausdruck

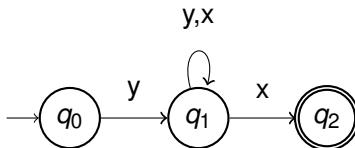
$$r := (a \cup (ab(b)^*ba))^*$$

gegeben.

Geben Sie einen NEA an, der $L(r)$ erkennt. Begründen Sie kurz die Korrektheit Ihres Automaten, ein formaler Korrektheitsbeweis ist jedoch nicht erforderlich.

(Hinweis: Es gibt einen NEA mit 3 Zuständen.)

Zu jedem nichtdeterministischen endlichen Automaten existiert ein äquivalenter deterministischer endlicher Automat.



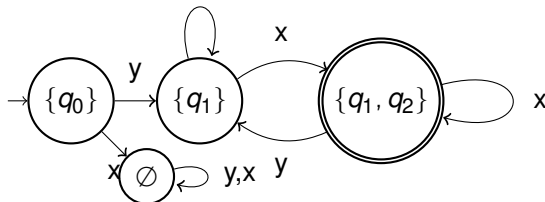
In eine Tabelle werden die Automatenzustände und ihre Folgezustände bei jeweiliger Eingabe eingetragen.

	y	x
$\{q_0\}$	$\{q_1\}$	\emptyset
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$

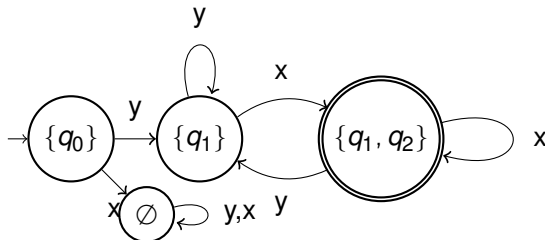
Potenzmengenkonstruktion

Ein **neuer Zustand** entsteht, wenn man von einem alten Zustand durch eine Eingabe in mehrere Zustände kommt.

	y	x
$\{q_0\}$	$\{q_1\}$	\emptyset
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_1\}$	$\{q_1, q_2\}$
\emptyset	\emptyset	\emptyset



Die Einträge der ersten Spalte sind die neuen Zustände. Alle Mengen, die einen Endzustand enthalten, sind wiederum im neuen Automaten Endzustände.



Satz 2.13 (Skript)

- Zu jedem nichtdeterministischen endlichen Automaten mit ε -Übergängen gibt es einen äquivalenten nichtdeterministischen endlichen Automaten ohne ε -Übergänge, der nicht mehr Zustände hat.
- äquivalent = akzeptiert dieselbe Sprache.

Erinnerung

Der ε -Abschluss $E(q)$ eines Zustandes q ist definiert als die Menge aller Zustände, die von q aus durch lediglich ε -Übergänge erreichbar sind (q selbst zählt auch dazu).

Konstruktion

Zu einem NEA $A := (Q, \Sigma, \delta, s, F)$ mit ε -Übergängen konstruieren wir einen äquivalenten NEA $\tilde{A} := (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{s}, \tilde{F})$ mit

- gleicher Zustandsmenge $\tilde{Q} := Q$
- gleichem Startzustand $\tilde{s} := s$
- neuer Endzustandsmenge $\tilde{F} := \{q \in Q \mid E(q) \cap F \neq \emptyset\}$
- neuer Übergangsfunktion $\tilde{\delta}(q, a) := \begin{cases} \{q\} & \text{falls } a = \varepsilon \\ \delta(E(q), a) & \text{sonst} \end{cases}$

Eigenschaften von \tilde{A}

$L(\tilde{A}) = L(A)$ und $|\tilde{Q}| = |Q|$.

Gegeben sei der NEA $\mathcal{A} = (\{s, q, f\}, \{a, b, c\}, \delta, s, \{f\})$, wobei die Übergangsfunktion δ gegeben ist durch:

	ε	a	b	c
s	$\{q, f\}$	\emptyset	$\{q\}$	$\{f\}$
q	\emptyset	$\{s\}$	$\{f\}$	$\{s, q\}$
f	\emptyset	\emptyset	\emptyset	\emptyset

1. Geben Sie zu dem Automaten \mathcal{A} den Übergangsgraphen an und eliminieren Sie die ε -Übergänge.
2. Ermitteln Sie mittels Potenzmengenkonstruktion den zu \mathcal{A} äquivalenten DEA. Geben Sie hierbei die Übergangsfunktion tabellarisch an.

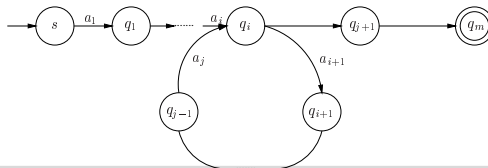
Pumping Lemma

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: $uv^i x \in L$



- 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.
- $\neg [\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \dots \forall i \in \mathbb{N} : uv^i x \in L]$
 $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \dots \exists i \in \mathbb{N} : uv^i x \notin L$
 - 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.

- 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.
- $\neg [\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \dots \forall i \in \mathbb{N} : uv^i x \in L]$
 $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \dots \exists i \in \mathbb{N} : uv^i x \notin L$
 - 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.

- 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.
- $\neg [\exists n \in \mathbb{N} : \forall w \in L, |w| > n : \exists uvx = w : \dots \forall i \in \mathbb{N} : uv^i x \in L]$
 $\Leftrightarrow \forall n \in \mathbb{N} : \exists w \in L, |w| > n : \forall uvx = w : \dots \exists i \in \mathbb{N} : uv^i x \notin L$
 - 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.

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

1. Angenommen, L sei regulär. Sei dann n wie im Pumping Lemma.
2. Wähle das Wort $w = a^n b^n$.
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.

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

1. Angenommen, L sei regulär. Sei dann n wie im Pumping Lemma.
2. Wähle das Wort $w = a^n b^n$.
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.

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

1. Angenommen, L sei regulär. Sei dann n wie im Pumping Lemma.
2. Wähle das Wort $w = a^n b^n$.
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.

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

1. Angenommen, L sei regulär. Sei dann n wie im Pumping Lemma.
2. Wähle das Wort $w = a^n b^n$.
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.

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

1. Angenommen, L sei regulär. Sei dann n wie im Pumping Lemma.
2. Wähle das Wort $w = a^n b^n$.
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.

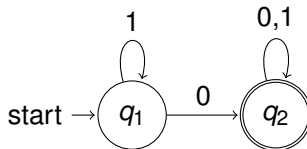
Welche der folgenden Sprachen sind regulär? Begründen Sie Ihre Antwort.

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

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?

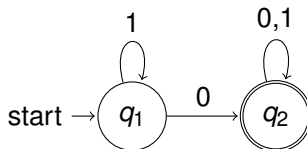
Idee: Betrachte die Sprachen L_{q_r, i, q_t} , definiert als $w \in \Sigma^*$ mit w überführt q_r in q_t unter Benutzung der Zwischenzustände $\{q_1, \dots, q_i\}$

- Es ist $L = \bigcup_{f \in F} L_{s, n, f}$
- 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.



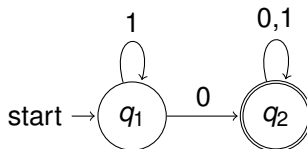
Was sind hier jeweils:

■ $L_{q_1,0,q_1}$



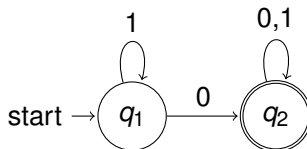
Was sind hier jeweils:

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



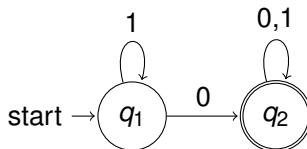
Was sind hier jeweils:

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



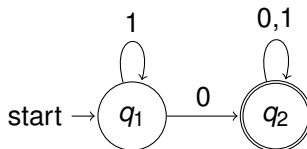
Was sind hier jeweils:

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



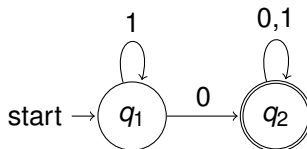
Was sind hier jeweils:

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



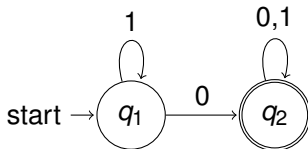
Was sind hier jeweils:

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



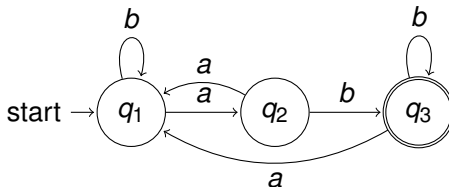
Was sind hier jeweils:

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



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: $1^* 0 (0 \cup 1)^*$

Bestimmen Sie mit dem im Beweis von Satz 2.14 verwendeten Verfahren die reguläre Sprache, die folgender deterministische endliche Automat erkennt:





Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems. – Jamie Zawinski

Some people, when confronted with a problem, think "I know, I'll quote Jamie Zawinski." Now they have two problems. – Mark Pilgrim



Dieses Werk ist unter einem "Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland"-Lizenzvertrag lizenziert. Um eine Kopie der Lizenz zu erhalten, gehen Sie bitte zu <http://creativecommons.org/licenses/by-sa/3.0/de/> oder schreiben Sie an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Davon ausgenommen sind das Titelbild, welches aus der März-April 2002 Ausgabe von American Scientist erschienen ist und ohne Erlaubnis verwendet wird, sowie das KIT Beamer Theme. Hierfür gelten die Bestimmungen der jeweiligen Urheber.