Theoretische Informatik 1
Große Übung 4

Prof. Dr. Roland Meyer
René Maseli
TU Braunschweig
Wintersemester 2025/26

1. Mit Myhill & Nerode zur Regularität

Gegeben sei die Sprache $L_1 := \{ w \in \{a, b\}^* \mid \exists n \in \mathbb{N} : n \equiv |w|_a \bmod 3 \text{ und } n \equiv |w|_b \bmod 2 \}$. Ziel ist es, zu zeigen, dass L_1 regulär ist. Dazu konstruiert man den Automaten nach Myhill & Nerode und stellt fest, dass die Zustandsmenge endlich ist. Ähnlich wie beim Potenzmengen-Automaten entdeckt man iterativ Zustände von A_{L_1}, beginnend mit $[\varepsilon]_{\equiv_{L_1}}$. Ob das Zielwort Repräsentant einer noch ungesehenen Äquivalenzklasse ist, lässt sich mit der Nerode-Rechtskongruenz prüfen.

$a \not\equiv_{L_1} \varepsilon$ (siehe ε)

$b \not\equiv_{L_1} \varepsilon$ (siehe ε) $\quad b \not\equiv_{L_1} a$ (a)

$aa \not\equiv_{L_1} \varepsilon$ (siehe ε) $\quad aa \not\equiv_{L_1} a$ (b) $\quad aa \not\equiv_{L_1} b$ (b)

$ab \not\equiv_{L_1} \varepsilon$ (siehe ab) $\quad ab \not\equiv_{L_1} a$ (ε) $\quad ab \not\equiv_{L_1} b$ (ε) $\quad ab \not\equiv_{L_1} aa$ (ε)

$ba \equiv_{L_1} ab$

$bb \equiv_{L_1} \varepsilon$

$aaa \equiv_{L_1} \varepsilon$

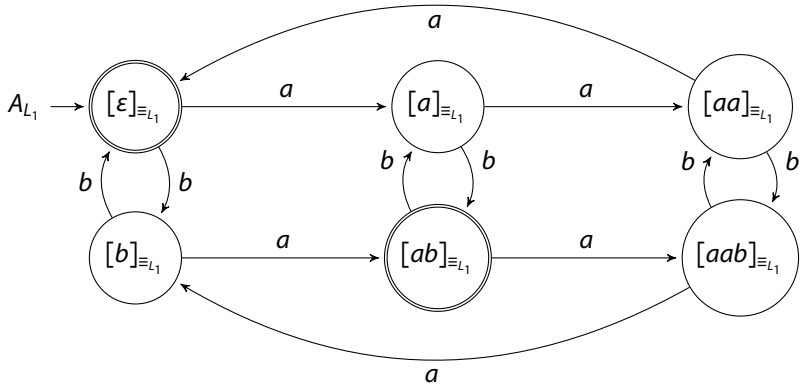
$aab \not\equiv_{L_1} \varepsilon$ (siehe ε) $\quad aab \not\equiv_{L_1} a$ (b) $\quad aab \not\equiv_{L_1} b$ (a) $\quad aab \not\equiv_{L_1} aa$ (a) $\quad aab \not\equiv_{L_1} ab$ (ε)

$aba \equiv_{L_1} aab$

$abb \equiv_{L_1} a$

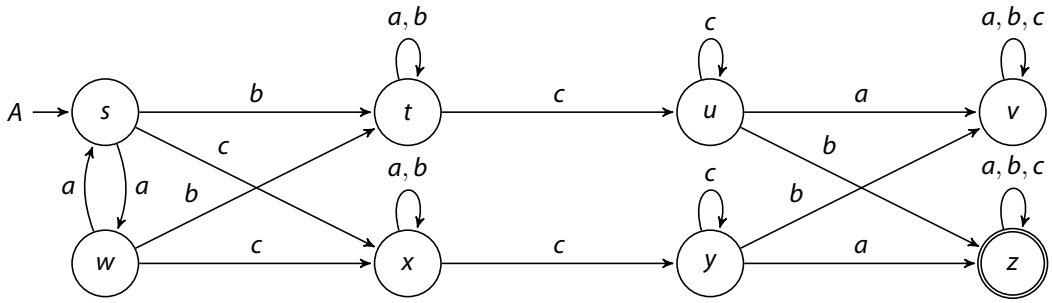
$aaba \equiv_{L_1} b$

$aabb \equiv_{L_1} aa$



2. Table-Filling-Algorithmus

Gegeben ist der folgende DFA. Ziel ist es, einen minimalen DFA zu erzeugen.

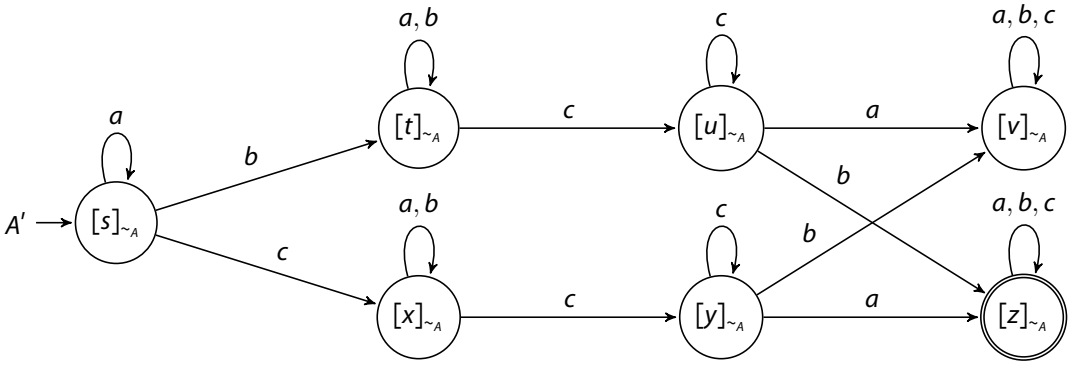


Die folgende Tabelle betrachtet ungeordnete Paare aus Zuständen von A. Die Nummern stehen für die Iteration des Algorithmus, in welcher das entsprechende Paar von Zuständen als ungleich festgestellt wurden.

	s	t	u	v	w	x	y	z
s		3	1	2		2	1	0
t			1	2	2	2	1	0
u				1	1	1	1	0
v					3	2	1	0
w						2	1	0
x							1	0
y								0
z								

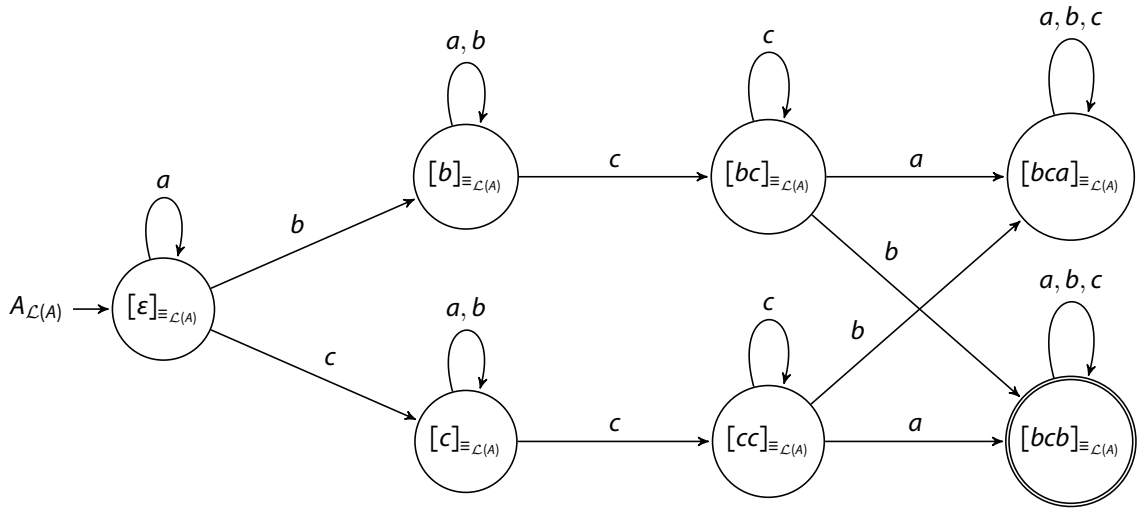
Vorgehensweise:
Initial (0) alle Q_F von $Q \setminus Q_F$ trennen.
0: z anders als der Rest
Iterativ (1,2,...) die Vorgänger frisch getrennter Zustände untersuchen. Als Hilfestellung stehen hier die für die jeweilige Iteration interessanten Transitionen, Self-Loops ausgenommen.
1: $u \xrightarrow{b} z, y \xrightarrow{a} z$ ($\Rightarrow$ 11 neue Trennungen)
2: $t \xrightarrow{c} u, x \xrightarrow{c} y$ ($\Rightarrow$ 7 neu)
3: $s \xrightarrow{b} t, w \xrightarrow{b} t, s \xrightarrow{c} x, w \xrightarrow{c} x$ ($\Rightarrow$ 2 neu)
4: $w \xrightarrow{a} s, s \xrightarrow{a} w$

Bei der vierten Iteration konnten keine neuen Paare unterschieden werden. Die leeren Zellen verraten nun, welche Zustände verschmolzen werden müssen, um einen minimalen DFA für $\mathcal{L}(A)$ zu erzeugen:



Es sollen alle Äquivalenzklassen der Nerode-Rechtskongruenz von $\mathcal{L}(A)$ aufgelistet werden. Jede dieser Klassen ist eine Sprache über dem Alphabet $\{a, b, c\}$ und ist einem der Zustände des minimalen DFA zugewiesen, nämlich den einzigartigen Zustand q mit $q_0 \xrightarrow{w} q$ für jedes Wort w

aus der Äquivalenzklasse. Die Klassen indentifizieren sich am Besten mit einem Repräsentanten, z.B. einem kürzesten Wort, das den Zustand ansteuert.



Ein minimaler Automat ist nützlich, um alle Nerode-Rechtskongruenzen zu berechnen. Betrachte dazu jeweils einen Zustand als einzigen Final-Zustand und erzeuge einen regulären Ausdruck mit Arden's Lemma. Diese Sprachen sollten paarweise disjunkt sein, und die Vereinigung sollte Σ^* sein.

$$
\begin{aligned}
[\varepsilon]_{\equiv_{\mathcal{L}(A)}} &= a^* \\
[b]_{\equiv_{\mathcal{L}(A)}} &= a^* b(a \cup b)^* \\
[c]_{\equiv_{\mathcal{L}(A)}} &= a^* c(a \cup b)^* \\
[bc]_{\equiv_{\mathcal{L}(A)}} &= a^* b(a \cup b)^* c^+ \\
[cc]_{\equiv_{\mathcal{L}(A)}} &= a^* c(a \cup b)^* c^+ \\
[bcb]_{\equiv_{\mathcal{L}(A)}} &= a^* \big(b(a \cup b)^* c^+ b \cup c(a \cup b)^* c^+ a\big)(a \cup b \cup c)^* \\
[bca]_{\equiv_{\mathcal{L}(A)}} &= a^* \big(b(a \cup b)^* c^+ a \cup c(a \cup b)^* c^+ b\big)(a \cup b \cup c)^*
\end{aligned}
$$

3. Pumping-Lemma

Das Pumping-Lemma stellt ein vergleichsweise einfaches notwendiges Kriterium an reguläre Sprachen. Jede reguläre Sprache L besitzt eine Pumping-Konstante $p_L \in \mathbb{N}$, sodass für jedes längere Wort $w \in L$ mit $|w| \geq p_L$ zerlegbar ist in $w = xyz$ mit $|xy| \leq p_L$ und $y \neq \varepsilon$, sodass für alle $i \in \mathbb{N}$ das gepumpte Wort $xy^iz \in L$ auch in der Sprache liegt.

Anders ausgedrückt, kann eine Sprache nicht regulär sein, wenn es für jede potenzielle Zustandszahl eines NFAs ein akzeptiertes Wort mit mindestens einem Schleifen-Durchlauf gibt, sodass jede potenzielle erste Schleife, die für dieses Wort durchlaufen werden muss, mindestens ein ungewolltes Wort zuviel akzeptieren lassen würde.

a) Sei $L_2 := \{ w \in \{a, b\}^* \mid |w|_a \geq |w|_b \}$. Es ist zu zeigen, dass L_2 nicht regulär ist.

Beweis
Sei $p \in \mathbb{N}$.
Betrachte $x = a^p b^p \in L_2 \cap \Sigma^{\geq p}$. (Alternativ $b^p a^p$ oder größere Wörter.)
Sei $x = u.v.w$ eine Zerlegung mit $|v| \geq 1$ und $|uv| \leq p$. Es gilt $v \in a^+$.
Betrachte $i = 0$. Es gilt $u.v^0.w = a^{p-|v|} b^p \notin L_2$, weil $|uw|_a = p - |v| \ngeq p = |uw|_b$.
Da dies für alle p gilt, ist L_2 nach dem Pumping-Lemma nicht regulär. □

b) Sei $L_3 := \{ w \in \{a, b\}^* \mid |w|_a \leq |w|_b \text{ oder } 2|w|_b \leq |w|_a \}$ Es ist zu zeigen, dass L_3 nicht regulär ist.

Beweis
Sei $p \in \mathbb{N}$.
Betrachte $x = a^{2(p+1)} b^{p+1} \in L_3 \cap \Sigma^{\geq p}$.
Sei $x = u.v.w$ eine Zerlegung mit $|v| \geq 1$ und $|uv| \leq p$. Es gilt $v \in a^{\leq p}$.
Betrachte $i = 0$. Es gilt $u.v^0.w = a^{2p+2-|v|} b^{p+1} \notin L_3$, weil $|uw|_a = 2p + 2 - |v| > p + 1 = |uw|_b$ und $2|uw|_b = 2p + 2 > 2p + 2 - |v| = |uw|_a$.
Da dies für alle p gilt, ist L_3 nach dem Pumping-Lemma nicht regulär. □

c) Sei $L_4 := \{ a^n b^m \mid n, m \in \mathbb{N} \text{ und } (n \neq 1 \text{ oder } \exists \ell \in \mathbb{N}\colon m = \ell^2) \}$.

Es kann mit dem Pumping-Lemma nicht (sofort) gezeigt werden, dass L_4 nicht regulär ist:

(Gegenbeweis)
Wähle $p_{L_4} = 3$. Sei $w = a^n b^m$ mit $n + m \geq p_{L_4}$ und $n = 1$ oder $m = \ell^2$.
Falls $n = 0$ oder $n = 2$, wähle eine Zerlegung mit $y = b$. Anderenfalls ist $n = 1$ oder $n \geq 3$.
Wähle eine Zerlegung mit $y = a$. Weder $i = 0$ noch $i \geq 2$ können $xy^iz \in L_4$ verhindern.
Nach dem Pumping-Lemma kann so keine Aussage über L_4 getroffen werden.

Um trotzdem Regularität widerlegen zu können, kann man eine Abschluss-Eigenschaften nutzen: Falls L_4 regulär ist, dann sind es auch e.g. $\overline{L_4}$, L_4^{reverse} oder $L_4 \cap ab^*$.

Beweis
Sei $p \in \mathbb{N}$.
Betrachte $x := ab^{p^2} \in L_4 \cap ab^* \cap \Sigma^{\geq p}$.
Sei $x = u.v.w$ mit $|v| \geq 1$ und $|u.v| \leq p$.
Falls $v \in ab^*$, wähle $i = 0$. E.g. folgt einfach $u.v^0.w \notin ab^* \supseteq L_4 \cap ab^*$.
Sonst ist $v \in b.b^{\leq p}$. Wähle $i = 2$.
Es gelten $p^2 < |u.v^2.w|_b$ und $|u.v^2.w|_b \leq p^2 + p < p^2 + 2p + 1 = (p + 1)^2$.
Daraus folgt $u.v^2.w = ab^{p^2+|v|} \notin L_4$.
Das gilt für alle Zerlegungen von x und alle p.
Nach dem Pumping-Lemma ist $L_4 \cap ab^*$ nicht regulär. Damit ist auch L_4 nicht regulär. □