Prof. Javier Esparza
Kevin Kappelmann, Simon Roßkopf, Lukas Stevens

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2023 – Übungsblatt 4

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Für den Rest des Semesters gilt: $\mathbb{N} \coloneqq \mathbb{N}_0 \coloneqq \{0, 1, 2, ...\}$ und $\mathbb{N}_+ \coloneqq \mathbb{N}_0 \setminus \{0\}$.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Individualaufgabe Ü4.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- Pumping Lemma
- Wortproblem
- Leerheitsproblem
- Endlichkeitsproblem
- Äquivalenzproblem
- Residualsprache
- Kanonischer Minimalautomat
- Minimierung
- $\equiv_L$

Individualaufgabe Ü4.2. (*Automata Tutor: Residualsprachen*)

Lösen Sie die Aufgaben Ü4.2 (a–f) auf Automata Tutor. **Hinweis:** Die “language of suffixes” ist die Residualsprache.

Übung und Nachbereitung

Übungsaufgabe Ü4.3. (*Pumping Lemma*)

Beweisen Sie für jede der folgenden Sprachen mithilfe des Pumping Lemmas, dass sie *nicht* regulär ist.

(a) $L_1 = \{w \in \{0, 1\}^* \mid w = w^R\}$

(b) $L_2 = \{w \in \{0, 1\}^* \mid |w|_0 \geq |w|_1\}$

(c) $L_5 = \{a^{2^i} \mid i \geq 0\}$

Lösungsskizze.

(a) Angenommen, L_1 wäre regulär.

Dann können wir das Pumping-Lemma anwenden.

Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl.

Dann gilt $z = 0^n10^n \in L_1$ und $|z| \geq n$.

Es gibt also für z eine Zerlegung $z = uvw$ mit $v \neq \varepsilon$ und $|uv| \leq n$, sodass $uv^iw \in L_1$ für alle $i \in \mathbb{N}$ $(*)$.

Durch unsere Wahl von z folgt $uv = 0^k$ für $0 < k \leq n$. Also $u = 0^i$ und $v = 0^j$ mit $i+j = k$ und $j > 0$.

Dann ist aber $uv^2w = 0^i0^{2j}0^{n-k}10^n \notin L_1$, denn $i + 2j + n - k > n$. Dies steht im Widerspruch zu $(*)$.

Das heißt unsere ursprüngliche Annahme ist falsch und L_1 nicht regulär.

(b) Angenommen, L_2 wäre regulär.

Dann können wir das Pumping-Lemma anwenden.

Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl.

Dann gilt $z = 0^n1^n \in L_2$ und $|z| \geq n$.

Es gibt also für z eine Zerlegung $z = uvw$ mit $v \neq \varepsilon$ und $|uv| \leq n$, sodass $uv^iw \in L_2$ für alle $i \in \mathbb{N}$ $(*)$.

Aus unserer Wahl von z folgt, dass $u = 0^i$ und $v = 0^j$ mit $j > 0$ gilt.

Allerdings ist $uv^0w = 0^i0^{n-i-j}1^n \notin L_2$, weil $i + (n - i - j) = n - j < n$. Dies steht im Widerspruch zu $(*)$.

Das heißt unsere ursprüngliche Annahme ist falsch und L_2 nicht regulär.

(c) Angenommen, L_5 wäre regulär.

Dann können wir das Pumping-Lemma anwenden.

Sei $n \in \mathbb{N}_+$ eine Pumping-Lemma-Zahl.

Dann gilt $a^{2^n} \in L_5$ und $|a^{2^n}| \geq n$.

Es gibt also für z eine Zerlegung $z = uvw$ mit $v \neq \varepsilon$ und $|uv| \leq n$, sodass $uv^iw \in L_5$ für alle $i \in \mathbb{N}$ $(*)$.

Deshalb $v = a^j$ für ein $0 < j \leq n$.

Dann ist aber $uv^2w = a^{2^n+j} \notin L_5$, denn

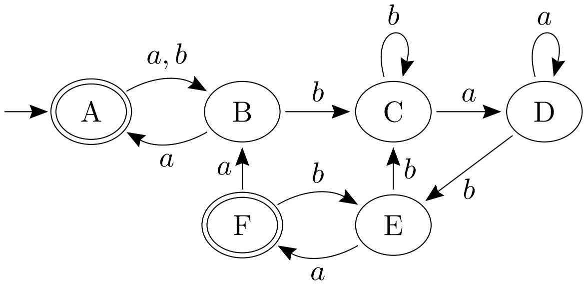
$$2^n < 2^n + j \leq 2^n + n < 2^n + 2^n = 2^{n+1}$$

Dies steht im Widerspruch zu $(*)$.

Das heißt unsere ursprüngliche Annahme ist falsch und L_4 nicht regulär.

Übungsaufgabe Ü4.4. (*Minimierungsalgorithmustuning*)

(a) Minimieren Sie den folgenden DFA.

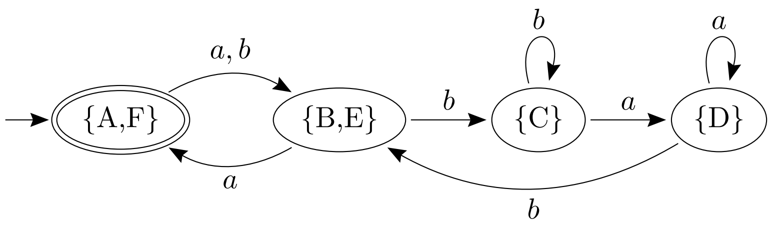


(b) Überlegen Sie sich, wie man den Minimierungsalgorithmus aus der Vorlesung abändern könnte, damit er neben einem minimalen DFA auch noch für jedes Paar an Zuständen (q_1, q_2), die nicht äquivalent sind, ein möglichst kurzes Wort w generiert, das beweist, dass q_1 und q_2 nicht äquivalent sind.
Wenden Sie den neuen Algorithmus auf den DFA aus (a) an.

Lösungsskizze.

(a) Tabelle und Automat:

	A	B	C	D	E
B	$\times$				
C	$\times$	$\times$			
D	$\times$	$\times$	$\times$		
E	$\times$	$=$	$\times$	$\times$	
F	$=$	$\times$	$\times$	$\times$	$\times$



(b) Statt in der Tabelle nur mit einem Kreuz ($\times$) zu markieren, dass zwei Zustände unterscheidbar sind, merken wir uns in der Tabelle ein Wort, das sie unterscheidet.

Gegeben ein DFA $(Q, \Sigma, q_0, \delta, F)$. Im ersten Schritt, tragen wir bei allen Paaren $q_1 \in F, q_2 \in Q \setminus F$ das leere Wort ε ein. Dann iterieren wir so lange über die Tabelle, bis sich nichts mehr ändert. Immer wenn wir zwei Zustände $q_1, q_2 \in Q$ mit dem Zeichen $x \in \Sigma$ unterscheiden können, tragen wir als Zeugen xw in die Tabelle ein, wobei $w \in \Sigma^*$ der Zeuge aus der Tabelle ist, dass $\delta(q_1, x), \delta(q_2, x)$ unterscheidbar sind. Hier die Tabelle von Aufgabe (a), die durch den neuen Minimierungsalgorithmus entsteht:

	A	B	C	D	E
B	ε				
C	ε	a			
D	ε	a	ba		
E	ε	$=$	a	a	
F	$=$	ε	ε	ε	ε

Übungsaufgabe Ü4.5. (*Residualsprachen*)

Sei $L = L((a^*b \mid c)^*a)$ über dem Alphabet $\Sigma = \{a, b, c, d\}$.

(a) Entscheiden Sie, welche der folgenden Äquivalenzen wahr sind und begründen Sie Ihre Antwort:

- $b \stackrel{?}{\equiv}_L c$
- $\varepsilon \stackrel{?}{\equiv}_L a$
- $abc \stackrel{?}{\equiv}_L cba$

(b) Sei $v = aababc$. Geben Sie ein Wort $u \neq v$ an, so dass $u \equiv_L v$.

(c) Geben Sie die Mengen L^{ab}, L^{ac} und L^{ca} an.

(d) Finden Sie nun L', sodass $c \equiv_{L'} ba$, $c \not\equiv_{L'} ab$ und $aba \equiv_{L'} bab$. Weiterhin soll $\varepsilon, aba \in L'$ gelten.

Lösungsskizze.

(a)
- $b \equiv_L c$, da $L^b = L = L^c$.
- $\varepsilon \not\equiv_L a$, da $\varepsilon \notin L^\varepsilon$ aber $\varepsilon \in L^a$.
- $abc \not\equiv_L cba$, da $\varepsilon \notin L^{abc}$ aber $\varepsilon \in L^{cba}$.

(b) ε

(c) $L^{ab} = L$, $L^{ac} = \emptyset$ und $L^{ca} = L^{a} = L(a^{*}b)L \cup \{\varepsilon\}$.

(d) $L^{'} = \{\varepsilon, aba, bab, cb\}$. (Es gibt auch andere Lösungen.)