Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – 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.

Vorbereitung ($\to$ vor der Übung selbständig zu bearbeiten)

Vorbereitungsaufgabe Ü4.1. (*Wichtige Begriffe*)

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

- Wortproblem
- Leerheitsproblem
- Endlichkeitsproblem
- Äquivalenzproblem
- Minimierung
- Quotientenautomat
- Residualsprache
- $\equiv_M$, $\equiv_L$

Vorbereitungsaufgabe Ü4.2. (*Automata Tutor: "Equivalence Classes"*)

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

Vorbereitungsaufgabe Ü4.3. (*Automata Tutor: "RE $\to$ ϵ-NFA"*)

Falls Sie den RE $\to$ ϵ-NFA-Algorithmus üben wollen, lösen Sie die Aufgaben Ü4.3 (a–c) auf Automata Tutor.

Beachten Sie: Automata Tutor überprüft nur, ob Ihr ϵ-NFA die richtige Sprache erkennt, nicht aber, ob Ihre Eingabe mit dem Verfahren der Vorlesung konstruiert wurde.

Übung und Nachbereitung

Übungsaufgabe Ü4.4. (*Ich nehm den 50:50 Joker*)

Bestimmen Sie für jede der folgenden Aussagen, ob sie wahr oder falsch ist. Falls die Aussage wahr ist, geben Sie eine kurze Begründung an. Andernfalls widerlegen Sie die Aussage, gegebenenfalls mit einem geeigneten Gegenbeispiel und Begründung, dass das Gegenbeispiel korrekt ist.

(a) Seien L_1, L_2 beliebige nichtleere Sprachen. Wenn L_1L_2 regulär ist, dann ist L_1 und L_2 regulär.

(b) Wenn $L \subseteq \Sigma^*$ regulär ist und $a \in \Sigma$, dann ist die Sprache aller Wörter aus L, die nicht mit a enden, regulär.

(c) Es gibt einen minimalen DFA M mit maximal 4 Zuständen, sodass die minimale Pumping-Lemma-Zahl für $L(M)$ kleiner als die Anzahl der Zustände von M ist.

Erinnerung: Eine Pumping-Lemma-Zahl n für eine Sprache L ist eine Zahl, sodass die Pumping-Lemma-Eigenschaft gilt. Genauer gesagt gibt es für alle $z \in L$ mit $|z| \geq n$ eine Zerlegung $z = uvw$ mit $v \neq \varepsilon$, $|uv| \leq n$ und $\forall i \geq 0.\ uv^i w \in L$.

Lösungsskizze.

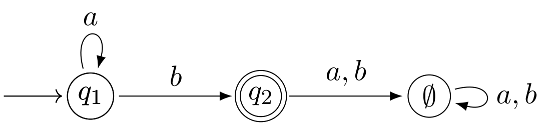
(a) Falsch. Sei z.B. $\Sigma = \{a, b\}$, $L_1 := \{a^n b^n \mid n \in \mathbb{N}\}$ und $L_2 := \Sigma^*$. Dann ist $L_1 L_2 = \Sigma^*$, da jedes $w \in \Sigma^*$ als εw mit $\varepsilon \in L_1$ und $w \in L_2$ dargestellt werden kann. $L_1 L_2$ ist regulär, L_1 aber nicht.

(b) Die Aussage ist wahr. Die Sprache $\Sigma^*\{a\}$ ist regulär, da Σ^* und $\{a\}$ regulär sind und reguläre Sprachen unter Konkatenation abgeschlossen sind. Durch die Abschlusseigenschaften der regulären Sprachen ist damit auch $L \setminus \Sigma^*\{a\}$ regulär, was die gesuchte Sprache ist.

Alternative: Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA für L. Sei $q_f \notin Q$. Der NFA $M' := (Q \cup \{q_f\}, \Sigma, \delta', q_0, \{q_f\})$ mit $\delta' := \delta \cup \{(q, x, q_f) \mid \delta(q, x) \in F \wedge x \in (\Sigma \setminus \{a\})\}$ akzeptiert die gewünschte Sprache.

Häufiger Fehler: Das Entfernen aller Transitionen (q, a, q_f) mit $q_f \in F$ ist zu restriktiv: es könnten Suffixe beginnend von q_f gelesen werden, welche nicht auf a enden und in einen weiteren Endzustand führen.

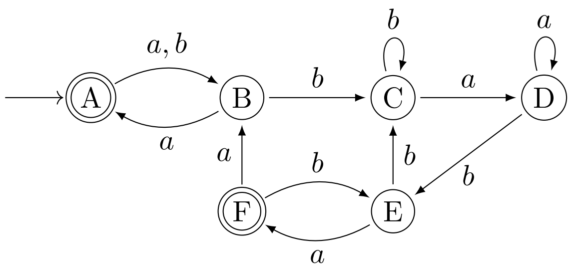
(c) Die Aussage ist wahr. Betrachte den minimalen DFA für die Sprache $\{a^n b \mid n \in \mathbb{N}\}$:



Der Automat hat 3 Zustände, aber 2 ist eine gültige Pumping-Lemma-Zahl: Für jedes Wort $a^n b$ mit $n \geq 1$ definiere $u := \epsilon, v := a, w := a^{n-1}b$. Dann gilt (1) $a^n b = \epsilon a a^{n-1} b = uvw$, (2) $v \neq \epsilon$, (3) $|uv| = 1 \leq 2$ und (4) $uv^i w = a^i a^{n-1} b = a^{n-1+i} b \in L$ für alle $i \geq 0$.

Übungsaufgabe Ü4.5. (*Minimierungsalgorithmustuning*)

(a) Minimieren Sie den folgenden DFA.

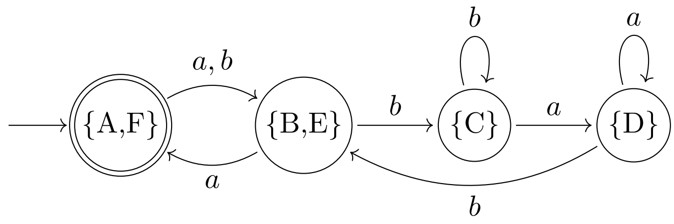


(b) Überlegen Sie sich, wie man den Minimierungsalgorithmus aus der Vorlesung abändern könnte, damit er neben einem minimalen DFA auch noch für jedes Paar an Zuständen (q_1, q_2), die nicht äquivalent sind, ein möglichst kurzes Wort w generiert, das beweist, dass q_1 und q_2 nicht äquivalent sind.
Wenden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

Lösungsskizze.

(a) Tabelle und Automat:

	A	B	C	D	E
B	×				
C	×	×			
D	×	×	×		
E	×	=	×	×	
F	=	×	×	×	×



(b) Statt in der Tabelle nur mit einem Kreuz (×) 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 Ü4.6. (*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, sodass $u \equiv_L v$.

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

(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.)