Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Peter Sanders
L. Hübschle-Schneider, T. Maier

4. Übungsblatt zu Theoretische Grundlagen der Informatik im WS 2015/16

http://algo2.iti.kit.edu/TGI2015.php
{sanders,huebschle,t.maier}@kit.edu

Musterlösungen

Aufgabe 1 (*Regularität von Sprachen, $2 + 2 + 2$ Punkte*)

Beweisen oder widerlegen sie die Regularität der folgenden Sprachen:

a) $L_1 = \{w \in L\big((a + (b \cup c))*\big) \mid n_a(w) > n_b(w) > n_c(w)\}$

b) $L_2 = \{w \in \{a,b\}^* \mid n_a(w) \bmod n_b(w) = 0\}$

c) $L_3 = \{uvvy \mid u, y \in \{a,b\}^*, v \in \{a,b\}\} \cup \{(ab)^{n^2} \mid n \in \mathbb{N}\}$

Musterlösung:

a) L_1 ist nicht regulär! Beweis mit Hilfe des Pumping Lemmas.
zZ. $\forall n \in \mathbb{N} \quad \exists w \in L_1 \mid |w| \geq n \quad \forall uvx = w \mid |uv| \leq n \quad |v| \geq 1 \quad \exists i \in \mathbb{N}_0 \mid uv^i x \notin L_1$

Für beliebiges aber festes n betrachte das Wort $(ab)^{n+1}(ac)^n$, jede Aufteilung des Wortes (nach obigen Einschränkungen) liegt in einem der folgenden Fälle:

- $v = a \quad \Rightarrow uv^0x = ux \notin L_1$ (es enthällt bb oder beginnt auf b)
- $n_b(v) > 0 \quad \Rightarrow uv^0x = ux \notin L_1$ ($n_b(ux) \leq n_c(ux)$)

b) L_2 ist nicht regulär! Beweis mit Hilfe des Pumping Lemmas.
zZ. $\forall n \in \mathbb{N} \quad \exists w \in L_2 \mid |w| \geq n \quad \forall uvx = w \mid |uv| \leq n \quad |v| \geq 1 \quad \exists i \in \mathbb{N}_0 \mid uv^i x \notin L_2$

Für beliebiges aber festes n betrachte das Wort $a^{n+1}b^{n+1} = uvx$. Durch die Einschränkungen an die Aufteilung gilt $v = a^k$ ($1 \leq k \leq n$). Daraus folgt $uv^0x = a^{n-k+1}b^{n+1} \quad \Rightarrow ux \notin L_2$.

c) L_3 ist nicht regulär! Der Beweis kann nicht mit dem Pumping Lemma vollzogen werden, weil jedes Wort der Sprache pumpbar ist.

Wir beweisen dies kurz (dies war in der Aufgabe nicht gefordert):
z.Z. $\exists n \in \mathbb{N} \quad \forall w \in L_3 \mid |w| > n \quad \exists uvx = w \mid |uv| \leq n, |v| \geq 1 \quad \forall i \in \mathbb{N}_0 \quad uv^i x \in L_3$
Sei $n = 3$, jedes Wort der Sprache (welches länger als 3 ist) lässt sich schreiben als: fg wobei $f \in \{a,b\}^3$ und $g \in \{a,b\}^*$. Wir machen die folgende Fallunterscheidung:

- $f = aaa$ sei $v = a$ nun gilt für alle $i \in \mathbb{N}_0$ dass $uv^ix \in L_3$
- $f = bbb$ analog.
- $n_a(f) = 2$ sei $v = b$ nun gilt für alle $i \in \mathbb{N}_0$ dass $uv^ix \in L_3$ (für $i = 0 \Rightarrow uv^0x = aag \in L_3$, $i = 1$ ist klar da $uvx = w \in L_3$, und $i > 1 \Rightarrow uv^ix$ enthält doppel bs)
- $n_b(f) = 2$ analog.

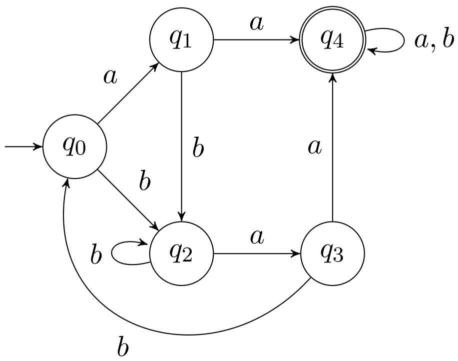
Es bleibt zu zeigen, dass L_3 nicht regulär ist (dies ist gefordert):
z.Z. L_3 hat unendlich viele Nerode-Äquivalenzklassen. Wir zeigen, dass für jedes $i \in \mathbb{N}_{\not\square}$ das Wort $(ab)^i$ seine eigene Äquivalenzklasse bildet ($[(ab)^i] = \{(ab)^i\}$). Sei $i, j \in \mathbb{N}_{\not\square}, i < j$ außerdem sei k die kleinste Quadratzahl $\geq i$ demnach gilt $(ab)^k = (ab)^i(ab)^{k-i} \in L_3$. Es gibt nun zwei Fälle:

- $(ab)^{j+k-i}$ ist keine Quadratzahl. $\Rightarrow [(ab)^i] \neq [(ab)^j]$
- $(ab)^{j+k-i}$ ist eine Quadratzahl (größer als k). Sei ℓ die nächstgrößere Quadratzahl ($\ell = (\sqrt{k}+1)^2$), dann folgt aus der Konstruktion, dass $(ab)^{j+\ell-i}$ keine Quadratzahl ist. Dies gilt, da die Abstände zwischen zwei Quadratzahlen streng monoton steigen ($\ell = (\sqrt{k}+1)^2 = k+2\sqrt{k}+1$).

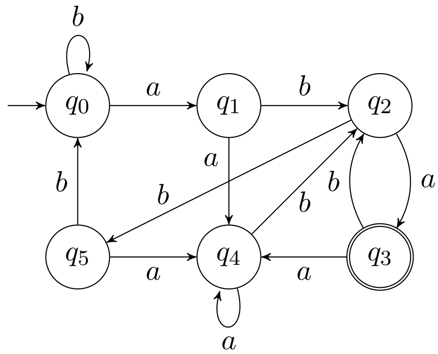
Aufgabe 2 (*Automatenminimierung,* $2+3+3$ *Punkte*)

Minimieren Sie die folgenden Automaten mit dem Verfahren aus der Vorlesung.

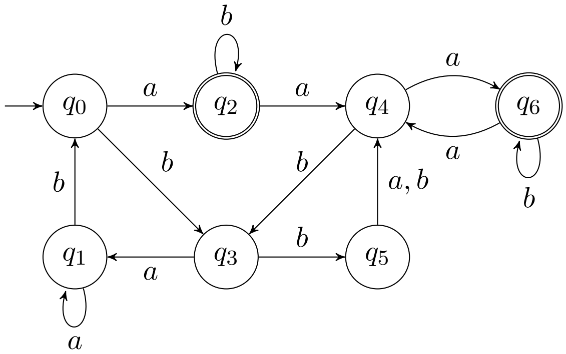
a)



b)



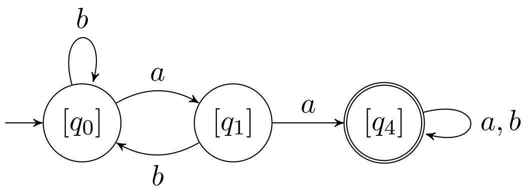
c)



Musterlösung:
Die Automaten enthalten keine Zustände, die von q_0 aus nicht erreichbar sind. Wir erhalten folgende kürzesten Zeugen für Nichtäquivalenz und folgenden Minimalautomaten:

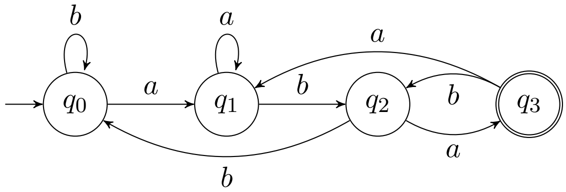
a) Der Automat erkennt die Sprache aller Wörter, die aa enthalten.

	q_0	q_1	q_2	q_3
q_1	a			
q_2		a		
q_3	a		a	
q_4	ε	ε	ε	ε



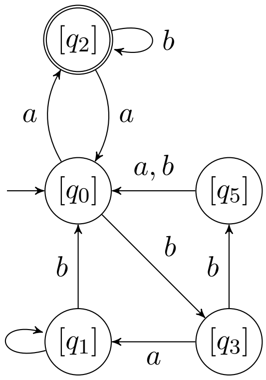
b) Der Automat erkennt die Sprache aller Wörter, die auf aba enden.

	q_0	q_1	q_2	q_3	q_4
q_1	ba				
q_2	a	a			
q_3	ε	ε	ε		
q_4	ba		a	ε	
q_5		ba	a	ε	ba



c) Der reguläre Ausdruck $(ba^+b \mid bbb \mid bba \mid ab^*a)^*ab^*$ beschreibt die Sprache des Automaten.

	q_0	q_1	q_2	q_3	q_4	q_5
q_1	a					
q_2	ε	ε				
q_3	a	aa	ε			
q_4		a	ε	a		
q_5	a	aa	ε	aa	a	
q_6	ε	ε		ε	ε	ε



Aufgabe 3 (*Chomsky-Normalform,* $2+4$ *Punkte*)

Überführen Sie die folgenden Grammatiken mit der Konstruktion aus der Vorlesung in Chomsky-Normalform.

a) $G_1 = (V_1, \Sigma_1 = \{a, b, c, d\}, P_1, S)$
$V_1 = \{S, B, D\}$

$$\begin{aligned} P_1 = \{\, & S \to aSc \mid B \mid dD, \\ & B \to bBc \mid \varepsilon, \\ & D \to dS \mid \varepsilon \,\} \end{aligned}$$

b) $G_2 = (V_2, \Sigma_2 = \{a, b, c\}, P_2, 1)$
$V_2 = \{1, 2, 3, 4, 5\}$

$$\begin{aligned} P_2 = \{\, & 1 \to 2 \mid \varepsilon \mid a34, \\ & 2 \to 1 \mid b5, \\ & 3 \to 4 \mid a12, \\ & 4 \to 5 \mid b35, \\ & 5 \to c5 \mid \varepsilon \,\} \end{aligned}$$

Musterlösung:

a) 1. Entfernen der ε-Produktionen: $G_1' = (V_1' = V_1 \cup \{S'\}, \Sigma_1, P_1', S)$ mit

$$\begin{aligned} P_1' = \{\, & S' \to \varepsilon \mid S \\ & S \to aSc \mid ac \mid B \mid dD \mid d \\ & B \to bBc \mid bc \\ & D \to dS \mid d \,\} \end{aligned}$$

2. Entfernen von zyklischen und nichtzyklischen Einheitsproduktionen: $G_1'' = (V_1', \Sigma_1, P_1'', S)$ mit

$$\begin{aligned} P_1'' = \{\, & S' \to \varepsilon \mid aSc \mid ac \mid dD \mid d \mid bBc \mid bc \\ & S \to aSc \mid ac \mid dD \mid d \mid bBc \mid bc \\ & B \to bBc \mid bc \\ & D \to dS \mid d \,\} \end{aligned}$$

3. Elimination gemischter und langer rechter Seiten: $G_1''' = (V_1''', \Sigma_1, P_1''', S)$
$V_1''' = V_1' \cup \{S_c, B_c, A_T, B_T, C_T, D_T\}$

$$\begin{aligned} P_1''' = \{\, & S' \to \varepsilon \mid A_T S_c \mid A_T C_T \mid D_T D \mid d \mid \\ & \qquad B_T B_c \mid B_T C_T \\ & S \to A_T S_c \mid A_T C_T \mid D_T D \mid d \mid \\ & \qquad B_T B_c \mid B_T C_T \\ & B \to B_T B_c \mid B_T C_T \\ & D \to D_T S \mid d \\ & S_c \to S C_T \\ & B_c \to B C_T \\ & A_T \to a \\ & B_T \to b \\ & C_T \to c \\ & D_T \to d \,\} \end{aligned}$$

b) 1. Entfernen der ε-Produktionen: $G_2' = (V_2' = V_2 \cup \{S\}, \Sigma_2, P_2', S)$ mit

$$\begin{aligned} P_2' = \{\, & S \to \varepsilon \mid 1, \\ & 1 \to 2 \mid a34 \mid a3 \mid a4 \mid a, \\ & 2 \to 1 \mid b5 \mid b, \\ & 3 \to 4 \mid a12 \mid a \mid a1 \mid a2, \\ & 4 \to 5 \mid b \mid b3 \mid b5 \mid b35, \\ & 5 \to c5 \mid c \,\} \end{aligned}$$

2. Entfernen von zyklischen und nichtzyklischen Einheitsproduktionen: $G_2'' = (V_2', \Sigma_2, P_2'', S)$ mit

$$\begin{aligned} P_2'' = \{\, & S \to \varepsilon \mid a \mid b \mid a3 \mid a34 \mid a4 \mid b5, \\ & 1 \to a \mid b \mid a3 \mid a34 \mid a4 \mid b5, \\ & 3 \to a \mid b \mid c \mid a1 \mid a11 \mid b3 \mid b5 \mid b35 \mid c5, \\ & 4 \to b \mid c \mid b3 \mid b5 \mid b35 \mid c5, \\ & 5 \to c \mid c5 \,\} \end{aligned}$$

3. Elimination gemischter und langer rechter Seiten: $G_2''' = (V_2''', \Sigma_2, P_2''', S)$
$V_2''' = V_2' \cup \{A, B, C, 1_1, 3_4, 3_5\}$

$$\begin{aligned} P_2''' = \{\, & S \to \varepsilon \mid a \mid b \mid A3 \mid A4 \mid A3_4 \mid B5, \\ & 1 \to a \mid b \mid A3 \mid A4 \mid A3_4 \mid B5, \\ & 3 \to a \mid b \mid c \mid A1 \mid A1_1 \mid B3 \mid B5 \mid \\ & \quad B3_5 \mid C5, \\ & 4 \to b \mid c \mid B3 \mid B5 \mid B3_5 \mid C5, \\ & 5 \to c \mid C5, \\ & A \to a, \\ & B \to b, \\ & C \to c, \\ & 1_1 \to 11, \\ & 3_4 \to 34, \\ & 3_5 \to 35 \,\} \end{aligned}$$

Zusatzaufgabe 1 (*Spiegelung regulärer Sprachen,* 2 *Punkte*)

Schreiben Sie einen Algorithmus, der für einen deterministischen endlichen Automaten $A = (Q, \Sigma, \delta, s, F)$ einen DEA A^r erzeugt, sodass $L(A^r) = \{w^r \mid w \in L(A)\}$.

Musterlösung:
Wir konstruieren zuerst einen ε-NEA auf Basis des DEA A:

- Füge einen neuen Startzustand S' ein mit $\delta(S', \varepsilon) = F$ (Epsilon-Übergange zu jedem Endzustand des DEA)
- Einziger neuer Endzustand ist s, der alte Startzustand
- Drehe alle Übergänge des DEA um, d.h. aus $\delta(q_1, a) = q_2$ wird $q_1 \in \delta^r(q_2, a)$

Nun überführen wir den ε-NEA mit der Potenzmengenkonstruktion in einen äquivalenten DEA A^r.

Zusatzaufgabe 2 (*Effiziente Berechnung minimaler DEAs von NEAs,* 3 *Punkte*)

Abgabe auf gesondertem Blatt und ausnahmsweise auch zu dritt möglich!
Gesucht ist ein Algorithmus, der als Eingabe einen NEA erhält und den dazugehörigen minimalen DEA ausgibt. Die Laufzeit des Algorithmus soll polynomiell in der Anzahl Zustände n_{NEA} des NEA und n_{DEA} des minimalen DEA sein. Sie darf *nicht* von der Anzahl Zustände des Potenzmengenautomaten abhängen, Sie können also nicht einfach den Automaten deterministisch machen und dann minimieren.

- Für einen Algorithmus, der das Problem löst, erhalten Sie 1 Million US-Dollar.[1]
- Für einen Beweis, dass das Problem unlösbar ist, erhalten Sie 1 Doktorarbeit.[2, 3]
- Für die Begründung, warum das Problem vermutlich unlösbar ist, erhalten Sie 3 Punkte.[4] Für besonders gute Begründungen erhalten Sie außerdem 1 Eis.[3, 4, 5]

[1] Leider nicht von uns, aber von dritter Stelle.
[2] Gerne auch bei uns. Keine Garantie.
[3] Kein Umtausch, keine Barauszahlung.
[4] Von uns.
[5] Kann Spuren von Schalenfrüchten, Erdnüssen, Ei, Weizen und Soja enthalten.

Musterlösung:
Das Problem ist *vermutlich unlösbar*: Es ist (mindestens) PSPACE-vollständig (mehr dazu später in der Vorlesung), da es PSPACE-vollständig ist, zu entscheiden, ob für einen NEA $A = (Q, \Sigma, P, S)$ gilt: $L(A) = \Sigma^*$ (Meyer, Stockmeyer, 1972, Lemma 2.3, `https://people.csail.mit.edu/meyer/rsq.pdf`). Der minimale DEA für Σ^* hat aber nur einen Zustand, also $n_{\mathrm{NEA}} + n_{\mathrm{DEA}} = n_{\mathrm{NEA}} + 1$. Wenn Sie also die Aufgabe lösen können, impliziert das, dass sie ein PSPACE-vollständiges Problem in polynomieller Zeit gelöst haben.
Da $P \subseteq NP \subseteq PSPACE$, würde dies $P = NP = PSPACE$ implizieren, zwei phänomenale Ergebnisse. Auf Ersteres hat das Clay Mathematics Institute ein Preisgeld von 1 Million US-Dollar ausgeschrieben.[6]
Da für beide Inklusionen *vermutet* wird, dass diese echt sind (also $P \subset NP \subset PSPACE$), dies aber bisher nicht bewiesen werden konnte, wäre ein Unmöglichkeitsbeweis ebenfalls ein sehr spannendes Ergebnis, zumal dies implizieren würde, dass entweder $P \neq NP$ oder $NP \neq PSPACE$.

[6] `http://www.claymath.org/millennium-problems/p-vs-np-problem`