Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 2

Vorlesung Theoretische Grundlagen der Informatik im WS 19/20

Ausgabe: 5. November 2019
Abgabe: 19. November 2019, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1 (2 + 2 = 4 Punkte)

(a) Geben Sie einen Algorithmus an, der für einen gegebenen vollständigen DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ in Zeit $\mathcal{O}(|Q||\Sigma|)$ entscheidet, ob $L(\mathcal{A}) = \emptyset$ gilt. Begründen Sie die Laufzeit!

(b) Geben Sie einen Algorithmus an, der für zwei gegebene vollständige DEAs $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ und $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ in Zeit $\mathcal{O}(|Q_1||Q_2||\Sigma|)$ entscheidet, ob $L(\mathcal{A}_1) = L(\mathcal{A}_2)$ gilt. Begründen Sie die Laufzeit!

Aufgabe 2 (4 Punkte)

Gegeben sei ein endliches Alphabet Σ. Das *Shuffle-Produkt* von zwei Wörtern $u, v \in \Sigma^*$ ist wie folgt definiert:

$$u \between v := \{u_1v_1u_2v_2 \dots u_nv_n \mid n \in \mathbb{N}_0, u_1, \dots, u_n, v_1, \dots, v_n \in \Sigma^*, u_1 \dots u_n = u, v_1 \dots v_n = v\}$$

Die beiden Wörter werden also auf (fast) beliebige Weise durchmischt, wobei nur gefordert wird, dass die interne Reihenfolge von u bzw. v erhalten bleibt.

Beispiel: $\texttt{ab} \between \texttt{cde} = \{\texttt{abcde}, \texttt{acbde}, \texttt{acdbe}, \texttt{acdeb}, \texttt{cabde}, \texttt{cadbe}, \texttt{cadeb}, \texttt{cdabe}, \texttt{cdaeb}, \texttt{cdeab}\}$

Dabei setzt sich z.B. $\texttt{cadeb}$ wie folgt zusammen: $u_1 = \varepsilon$, $v_1 = \texttt{c}$, $u_2 = \texttt{a}$, $v_2 = \texttt{de}$, $u_3 = \texttt{b}$, $v_3 = \varepsilon$.

Das Shuffle-Produkt zweier Sprachen L_1, L_2 ist dann definiert als:

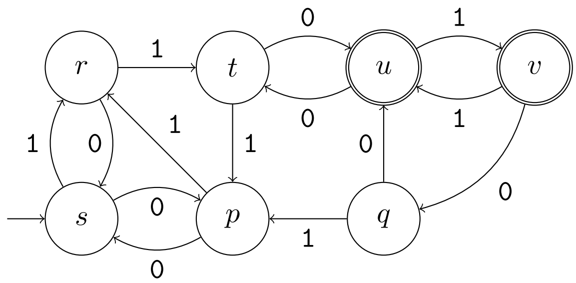
$$L_1 \between L_2 := \bigcup_{u \in L_1, v \in L_2} u \between v$$

Zeigen Sie: Wenn L_1 und L_2 regulär sind, dann ist auch $L_1 \between L_2$ regulär.

Hinweis: Erinnern Sie sich an den Kreuzproduktautomaten, der $L_1 \cap L_2$ erkennt. Passen Sie diesen geeignet an.

Aufgabe 3 (3 Punkte)

Konstruieren Sie zu folgendem Automaten den zugehörigen Äquivalenzklassenautomaten und zeichnen Sie den Übergangsgraphen.



Aufgabe 4 (2 + 3 + 2 + 1 = 8 Punkte)

Ein Kommilitone von Ihnen behauptet, dass er ein alternatives Verfahren zur Konstruktion von Äquivalenzklassenautomaten gefunden hat. Statt nach möglicherweise langen Zeugen suchen zu müssen, betrachtet er immer nur einzelne Zeichen. Zu einem deterministischen endlichen Automaten $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ geht er folgendermaßen vor. Im ersten Schritt partitioniert er die Zustandsmenge in zwei Mengen $Q \setminus F$ und F. In jedem weiteren Schritt wählt er zunächst ein Zeichen $a \in \Sigma$. Jede Menge[1] $[q]$ trennt er genau dann weiter auf, wenn für zwei Zustände $q_1, q_2 \in [q]$ nach dem vorherigen Schritt galt, dass $[\delta(q_1, a)] \neq [\delta(q_2, a)]$. Solch einen Schritt wiederholt er, bis sich bei keinem Zeichen $a \in \Sigma$ weitere Trennungen ergeben. Mit den entstandenen Mengen und dem Verfahren aus der Vorlesung konstruiert er dann den Äquivalenzklassenautomaten.

(a) Führen Sie das Verfahren für den Automaten aus Aufgabe 3 durch, indem Sie die unten stehende Tabelle ausfüllen. Finden Sie dieselben Mengen wie die Äquivalenzklassen aus Aufgabe 3?

[1] Disjunkte Mengen $\{\dots, q, \dots\} = [q]$ können durch einen Repräsentanten q identifiziert werden.

Schritt	Zeichen a	Partition nach Trennung durch a
1		$\{s, p, q, r, t\}$, $\{u, v\}$
2	0	
3		
4		
5		
6		
7		
8		
9		

(b) Folgern Sie aus $[q_1] \neq [q_2]$ induktiv die Existenz eines Zeugen w, der q_1 und q_2 trennt.

(c) Zeigen Sie, dass am Ende des Verfahrens $[q_1] = [q_2]$ impliziert, dass kein Zeuge existiert, der q_1 und q_2 trennt.

(d) Ist das Verfahren Ihres Kommilitonen korrekt?

Aufgabe 5

(2 + 2 + 2 + 2 = 8 Punkte)

Ist die Aussage des Pumping-Lemmas für folgende Sprachen erfüllt? Welche Sprachen sind regulär? Begründen Sie!

(a) $\Sigma = \{\mathtt{a}\}$, $L_a = \{\mathtt{a}^{\lfloor 10/j \rfloor !} \mid j \in \mathbb{N}\}$

(b) $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $L_b = \{\mathtt{a}^j \mathtt{b}^k \mathtt{c}^\ell \mid j, k, \ell \in \mathbb{N}_0, j < k < \ell\}$

(c) $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, $L_c = \{y\mathtt{0}z \mid y, z \in \Sigma^*, |y| = |z|\}$

(d) $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, $L_d = \{aza \mid a \in \Sigma, z \in \Sigma^*\}$

Aufgabe 6

(3 + 2 = 5 Punkte)

Gegeben seien die folgenden Sprachen über dem Alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$:

- $L_1 = \{w \in \Sigma^* \mid |w|_\mathtt{a} = |w|_\mathtt{b}\}$
- $L_2 = \{w \in \Sigma^* \mid |w|_\mathtt{a} \mod 3 = |w|_\mathtt{b} \mod 3\}$

Dabei bezeichnet $|w|_x$ die Anzahl der Vorkommen des Zeichens x in w. In dieser Aufgabe sollen Sie die gegebenen Sprachen mithilfe der Nerode-Relation untersuchen.

(a) Geben Sie für L_1 und L_2 jeweils die Äquivalenzklassen der Nerode-Relation (in Mengenschreibweise) an. Geben Sie außerdem für jede Äquivalenzklasse K die Menge S_K der gültigen Suffixe an, also die Menge der $z \in \Sigma^*$, sodass für jedes $u \in K$ gilt: $uz \in L_1$ bzw. $uz \in L_2$.

(b) Ist L_1 bzw. L_2 regulär? Wenn ja, geben Sie einen DEA mit der minimalen Anzahl von Zuständen an.