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

Übungsblatt 2

Vorlesung Theoretische Grundlagen der Informatik im WS 18/19

Ausgabe 06. November 2018
Abgabe 20. November 2018, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Bitte benutzen Sie den WebInScribe Deckblattgenerator:
`https://webinscribe.ira.uka.de/deckblatt/index.php?course=10696`

Aufgabe 1 (1 + 2 + 2 = 5 Punkte)

In der Vorlesung wurde die Quotientensprache zweier Sprachen L_1 und L_2 über dem Alphabet Σ wie folgt definiert:

$$L_1/L_2 \coloneqq \{w \in \Sigma^\star \mid \exists z \in L_2 \colon wz \in L_1\}$$

(a) Beschreiben Sie die Sprache $L_1/\Sigma^\star$ in Worten.

(b) Zeigen Sie: Falls L_1 regulär ist, ist auch $L_1/\Sigma^\star$ regulär. Sei dazu $\mathcal{A}$ ein DEA, der L_1 erkennt. Geben Sie einen DEA $\mathcal{A}'$ an, der $L_1/\Sigma^\star$ erkennt.

(c) Zeigen Sie: Falls L_1 regulär ist, ist auch L_1/L_2 regulär. Beachten Sie, dass für L_2 keine Einschränkungen gemacht werden. Sei dazu $\mathcal{A}$ ein DEA, der L_1 erkennt. Geben Sie einen DEA $\mathcal{A}'$ an, der L_1/L_2 erkennt.

Lösung:

(a) $L_1/\Sigma^\star$ ist die Sprache aller Präfixe von Wörtern aus L_1.

(b) Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein DEA, der L_1 erkennt. Dann wird $L_1/\Sigma^\star$ durch den Automaten $\mathcal{A}' = (Q, \Sigma, \delta, s, F')$ mit $F' \coloneqq \{q \in Q \mid \exists w \in \Sigma^\star \colon \delta(q, w) \in F\}$ erkannt. Es werden also alle Zustände zu Finalzuständen gemacht, von denen aus es einen Pfad zu einem ursprünglichen Finalzustand gibt.

Wir beweisen, dass $\mathcal{A}'$ $L_1/\Sigma^\star$ erkennt, indem wir für ein beliebiges Wort $w \in \Sigma^\star$ zeigen: $w \in L_1/\Sigma^\star \Leftrightarrow \delta(s, w) \in F'$.

$$\begin{aligned} w \in L_1/\Sigma^\star &\Leftrightarrow \exists z \in \Sigma^\star \colon wz \in L_1 \\ &\Leftrightarrow \exists z \in \Sigma^\star \colon \delta(s, wz) \in F \\ &\Leftrightarrow \exists z \in \Sigma^\star \colon \delta(\delta(s, w), z) \in F \\ &\Leftrightarrow \delta(s, w) \in F' \end{aligned}$$

(c) Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein DEA, der L_1 erkennt. Dann wird L_1/L_2 durch den Automaten $\mathcal{A}' = (Q, \Sigma, \delta, s, F')$ mit $F' := \{q \in Q \mid \exists w \in L_2 \colon \delta(q, w) \in F\}$ erkannt.

Wir beweisen, dass $\mathcal{A}'$ L_1/L_2 erkennt, indem wir für ein beliebiges Wort $w \in \Sigma^\star$ zeigen: $w \in L_1/L_2 \Leftrightarrow \delta(s, w) \in F'$.

$$\begin{aligned} w \in L_1/L_2 &\Leftrightarrow \exists z \in L_2 \colon wz \in L_1 \\ &\Leftrightarrow \exists z \in L_2 \colon \delta(s, wz) \in F \\ &\Leftrightarrow \exists z \in L_2 \colon \delta(\delta(s, w), z) \in F \\ &\Leftrightarrow \delta(s, w) \in F' \end{aligned}$$

Aufgabe 2

(1 + 1 + 2 = 4 Punkte)

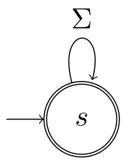
(a) Geben Sie den Zustandsgraphen eines deterministischen endlichen Automaten $\mathcal{A} = (Q, \Sigma = \{\mathtt{0}, \mathtt{1}\}, \delta, S, F)$ mit möglichst wenig Zuständen an, sodass für alle binär kodierten Primzahlen w gilt: $w \in L(\mathcal{A})$.

(b) Geben Sie den Zustandsgraphen eines deterministischen endlichen Automaten an, der die Sprache aller vierstelligen Binärzahlen, die eine Primzahl kodieren, erkennt. Dabei sollen Zahlen mit weniger als vier Stellen mit führenden Nullen aufgefüllt werden, also wird z.B. die 1 durch `0001` repräsentiert, aber nicht durch `1`, `01` oder `001`.

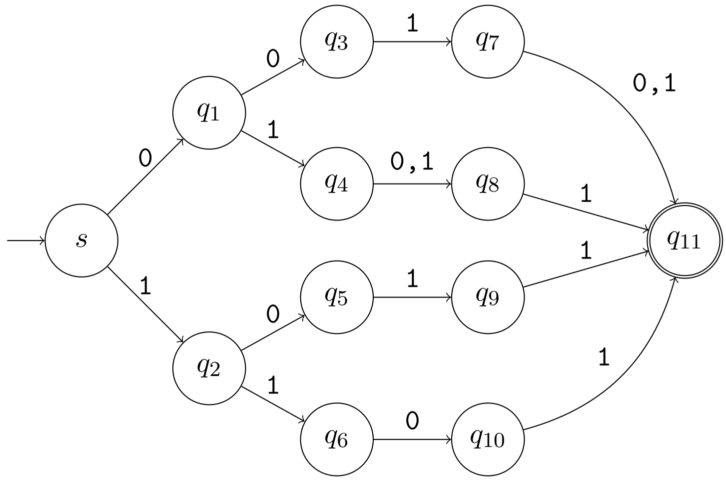
(c) Zeigen Sie, dass jede endliche Sprache L regulär ist, in dem Sie in wenigen Sätzen beschreiben, wie ein DEA konstruiert werden kann, der L akzeptiert.

Lösung:

(a)



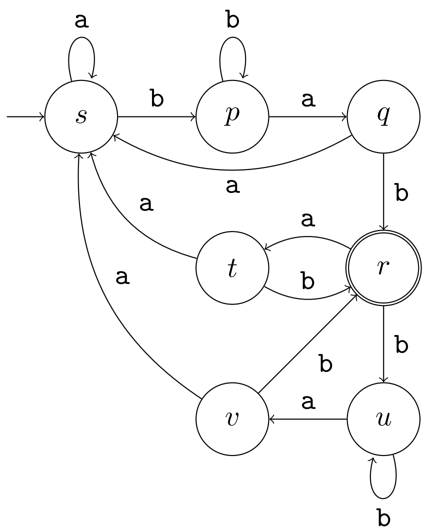
(b)



(c) Beginne mit einem NEA mit nicht akzeptierendem Startzustand s. Füge für jedes der endlich vielen Wörter $w \in L$ eine „Zustandschlange“ ein, die genau w akzeptiert, und einen ε-Übergang von s zum Anfangszustand dieser Schlange. Nutze schließlich die Potenzmengenkonstruktion, um einen zum NEA äquivalenten DEA zu erhalten.

Aufgabe 3 (3 Punkte)

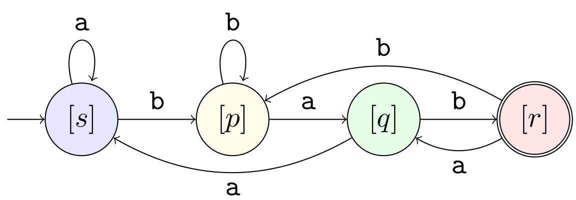
Konstruieren Sie zu folgendem Automaten den zugehörigen Äquivalenzklassenautomaten und zeichnen Sie den Übergangsgraphen.



Lösung:

Zeuge	Äquivalenzklassen
ε	$\{s,p,q,t,u,v\}$, $\{r\}$
a	$\{s,p,q,t,u,v\}$, $\{r\}$
b	$\{s,p,u\}$, $\{q,t,v\}$, $\{r\}$
aa	$\{s,p,u\}$, $\{q,t,v\}$, $\{r\}$
ab	$\{s\}$, $\{p,u\}$, $\{q,t,v\}$, $\{r\}$
ba	$\{s\}$, $\{p,u\}$, $\{q,t,v\}$, $\{r\}$
bb	$\{s\}$, $\{p,u\}$, $\{q,t,v\}$, $\{r\}$

Alle Wörter der Länge 3 trennen keine weiteren Äquivalenzklassen.



Bemerkung: Diese Automaten erkennen die Sprache aller Wörter aus $\{\texttt{a},\texttt{b}\}^\star$, die mit bab enden.

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?

Lösung:

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

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

Lösung:

Induktionsanfang: Nach dem Anfangsschritt gibt es zwei Mengen $Q \setminus F$ und F. Diese werden durch den Zeugen ε getrennt.

Induktionsannahme: Nach dem i-ten Schritt folgt aus $[q_1] \neq [q_2]$ die Existenz eines Zeugen w, der q_1 und q_2 trennt.

Induktionsschluss: Betrachte jetzt den Schritt $i + 1$ und zwei Zustände $q_1, q_2 \in Q$ mit $[q_1] \neq [q_2]$. Galt schon im Schritt i der Zusammenhang $[q_1] \neq [q_2]$, existiert nach Induktionsannahme ein Zeuge, der q_1 und q_2 trennt. Ansonsten galt im Schritt i dass $[q_1] = [q_2]$. Das bedeutet, dass a die Zustände q_1 und q_2 im Schritt $i + 1$ getrennt hat. Das ist der Fall genau dann, wenn im Schritt i galt, dass $[\delta(q_1, a)] \neq [\delta(q_2, a)]$. Nach Induktionsannahme existiert dann ein Zeuge w, der $\delta(q_1, a)$ von $\delta(q_2, a)$ trennt. Dann ist aw ein Zeuge, der im Schritt $i + 1$ die Zustände 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.

Lösung:

Sei am Ende des Verfahrens $q_1, q_2 \in Q$ mit $[q_1] = [q_2]$. Wähle ein beliebiges Wort $w = a_1 a_2 \cdots a_k$ mit $a_i \in \Sigma$ für $1 \leq i \leq k$. Es gilt $[\delta(q_1, a_1)] = [\delta(q_2, a_1)]$, denn sonst würden die Mengen $[q_1]$ und $[q_2]$ durch a_1 weiter aufgetrennt werden. Genauso gilt $[\delta(\delta(q_1), a_1), a_2)] = [\delta(\delta(q_2, a_1), a_2)]$, denn sonst würden die Mengen $[\delta(q_1, a_1)]$ und $[\delta(q_2, a_1)]$ durch a_2 weiter aufgetrennt werden. Es folgt $[\delta(\ldots, a_k)] = [\delta(\ldots, a_k)]$. Da schon im ersten Schritt akzeptierende von nicht akzeptierenden Zuständen getrennt wurden, kann w also q_1 und q_2 nicht trennen. Da w beliebig gewählt ist, kann also kein Zeuge existieren, der q_1 und q_2 trennt.

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

(d) Ist das Verfahren Ihres Kommilitonen korrekt?

Lösung:
Ja, denn aus (b) folgt, dass nicht-äquivalente Zustände in unterschiedlichen Mengen liegen, und aus (c) folgt, dass Zustände, die in derselben Menge liegen, äquivalent sind. Das Verfahren berechnet also genau die (eindeutigen) Äquivalenzklassen und ist ansonsten identisch zu dem Verfahren aus der Vorlesung.

Aufgabe 5

(1 + 2 + 2 + 2 + 1 = 8 Punkte)

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

(a) $\Sigma = \{\texttt{A}, \dots, \texttt{Z}\} \cup \{\texttt{␣}, \texttt{-}\} \cup \{\texttt{0}, \dots, \texttt{9}\}$, $L_a = \{\text{mögliches Autokennzeichen in Karlsruhe}\}$
Beispielsweise gilt `KA-RT␣911` $\in L_a$, aber `KA-RT-OFFEL` $\notin L_a$.

(b) $\Sigma = \{\texttt{0}, \texttt{1}\}$, $L_b = \{w \in \Sigma^\star \mid w_0 \neq w_{\lfloor |w|/2 \rfloor}\}$

(c) $\Sigma = \{a, b, c\}$, $L_c = \{w \in \Sigma^\star \mid |w|_a + |w|_b = |w|_c\}$

(d) Für eine beliebige natürliche Zahl $k \in \mathbb{N}$ definieren wir $L_k = \{w \in \{1, \dots, k\}^\star \mid w \text{ ist eine monoton aufsteigende Zahlenfolge}\}$. Das Wort $4\,4\,7\,8$ ist beispielsweise in L_8 enthalten, das Wort $8\,8\,4\,8$ hingegen nicht. Wir unterscheiden Zahlen aus $\{1, \dots, k\}$ von Wörtern (Zahlenfolgen) aus L_k, indem wir die Zahlen mit einem Leerzeichen trennen. Sie können also die Zahlenfolge $8\,8\,4\,8$ von der Zahl 8848 unterscheiden.

(e) Ist die Menge $L_\infty = \{w \in \mathbb{N}^\star \mid w \text{ ist eine monoton aufsteigende Zahlenfolge}\}$ eine reguläre Sprache?

Lösung:

(a) Für alle Wörter der Länge $n > 10$ ist die Aussage des Pumping-Lemmas erfüllt, da die zugrunde liegende Menge leer ist. Die Sprache L_a ist endlich und somit regulär.

(b) Wir zeigen, dass die Aussage des Pumping-Lemmas nicht erfüllt ist und somit L_b nicht regulär sein kann.

Beweis: Sei $n \in \mathbb{N}$ beliebig aber fest. Wir wählen $w = \texttt{0}^n\texttt{1}^n$. Sei $w = uvx$ eine beliebige Zerlegung mit $|uv| \leq n$, $v \neq \varepsilon$. Dann gilt $u = \texttt{0}^i$, $v = \texttt{0}^j$ für Zahlen i, j mit $i + j \leq n, j > 0$. Das Wort $uv^2x = \texttt{0}^{n+j}\texttt{1}^n$ liegt nicht in L_b, da $(uv^2x)_0 = (uv^2x)_{\lfloor |uv^2x|/2 \rfloor} = \texttt{0}$. Somit gibt es für jedes $n \in \mathbb{N}$ ein Wort w in L_b, für das keine Zerlegung entsprechend dem Pumping-Lemma existiert.

(c) Wir zeigen, dass die Aussage des Pumping-Lemmas nicht erfüllt ist und somit L_b nicht regulär sein kann.

Beweis: Sei $n \in \mathbb{N}$ beliebig aber fest. Wir wählen $w = a^n b^n c^{2n}$. Sei $w = uvx$ eine beliebige Zerlegung mit $|uv| \leq n$, $v \neq \varepsilon$. Dann gilt $u = a^i$, $v = a^j$ für Zahlen i, j mit $i + j \leq n, j > 0$. Das Wort uv^0x liegt nicht in L_c, da $|uv^0x|_a = |uvx|_a - j < n, |uv^0x|_b = n, |uv^0x|_c = 2n \neq |uv^0x|_a + |uv^0x|_b$. Somit gibt es für jedes $n \in \mathbb{N}$ ein Wort w in L_c, für das keine Zerlegung entsprechend dem Pumping-Lemma existiert.

(d) Wir zeigen, dass die Aussage des Pumping-Lemmas für jedes $k \in \mathbb{N}$ erfüllt ist. Sei $w \in L_k$ mit $|w| > n = 1$ mit $w = uvx$ und $u = \varepsilon$ und $v = w_0$. Das Wort $uv^ix = v^ix$ enthält also i Kopien des ersten Zeichens am Wortanfang. Die Zeichenfolge in v^ix bleibt monoton aufsteigend und das Wort ist somit in L_k enthalten.

Wir konstruieren induktiv für jedes $k \in \mathbb{N}$ einen DEA $\mathcal{A}_k$, der L_k akzeptiert, und zeigen somit, dass L_k regulär ist.

Verankerung, $k = 1$:
$\mathcal{A}_1 = (Q_1 = \{q_1\}, \delta_1, q_1, F_1 = Q_1\}$ mit $\delta_1(q_1, 1) = q_1$
Induktive Forsetzung:
$\mathcal{A}_{k+1} = (Q_{k+1} = Q_k \cup \{q_{k+1}\}, \delta_{k+1}, F_{k+1} = F_k \cup \{q_{k+1}\})$

$$\delta_{k+1}(q, x) = \begin{cases} \delta(q, x) & q \in Q_k, x \in \mathbb{N}_k \\ q_{k+1} & x = k + 1 \end{cases}$$

(e) Die Aussage des Pumping-Lemmas ist auch für $k = \infty$ erfüllt, die Menge L_∞ ist allerdings keine formale Sprache (und somit auch keine reguläre Sprache), da das zugrunde liegende Alphabet nicht endlich ist.

Aufgabe 6

(1 + 3 + 1 = 5 Punkte)

Gegeben seien zwei Sprachen über dem Alphabet $\Sigma = \{\texttt{(}, \texttt{)}\}$: L_1 ist die Sprache, in der auf jede öffnende Klammer direkt eine schließende folgt. L_2 ist die Sprache der korrekten Klammerausdrücke (siehe Vorlesung 3, Folie 10).

(a) Geben Sie L_2 in Mengenschreibweise an.

Hinweis: Für ein Wort w können Sie das Teilwort von w, das vom i-ten Zeichen bis zum j-ten Zeichen geht, mit $w_{i,j}$ bezeichnen.

(b) 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 $z \in \Sigma^\star$, sodass für jedes $u \in K$ gilt: $uz \in L_1$ bzw. $uz \in L_2$.

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

Lösung:

(a) $L_2 = \{w \in \Sigma^\star \mid |w|_{\texttt{(}} = |w|_{\texttt{)}}, \forall\, 0 \leq i < |w| \colon |w_{0,i}|_{\texttt{)}} \leq |w_{0,i}|_{\texttt{(}}\}$

(b) L_1 hat drei Äquivalenzklassen:

- $[\varepsilon] = L_1$, $S_{[\varepsilon]} = L_1$.
- $[\texttt{(}] = \{w\texttt{(} \mid w \in L_1\}$, $S_{[\texttt{(}]} = \{\texttt{)}w \mid w \in L_1\}$.
- $[\texttt{((}] = \{u\texttt{((}v \mid u, v \in \Sigma^\star\}$, $S_{[\texttt{((}]} = \emptyset$.

L_2 hat unendlich viele Äquivalenzklassen:

- Für jedes $i \in \mathbb{N}_0$ bildet $[(^i] = \{w \in \Sigma^\star \mid |w|_(- |w|_) = i, \forall 0 \leq i < |w|\colon |w_{0,i}|_) \leq |w_{0,i}|_(\}$ eine eigene Äquivalenzklasse mit $S_{[(^i]} = \{w \in \Sigma^\star \mid |w|_) - |w|_(= i, \forall 0 \leq i < |w|\colon |w_{i,|w|-1}|_(\leq |w_{i,|w|-1}|_)\}$.
- Die Menge $[)] = \{uv \mid u, v \in \Sigma^\star, |u|_) > |u|_(\}$ bildet eine Äquivalenzklasse mit $S_{[)]} = \emptyset$.

(c) L_2 ist nicht regulär, da es unendliche viele Äquivalenzklassen gibt. L_1 hat 3 Äquivalenzklassen und ist somit regulär. Der Minimalautomat sieht wie folgt aus: