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

Übungsblatt 2

Vorlesung Theoretische Grundlagen der Informatik im WS 20/21

Ausgabe: 24. November 2020
Abgabe: 8. Dezember 2020, 11:30 Uhr (digital im ILIAS)

Aufgabe 1

(1 + 2 + 4 = 7 Punkte)

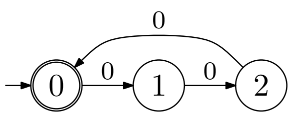
(a) Gegeben sei die Sprache $L_1 = \{w \in \{0\}^* \mid |w| \text{ ist durch 3 teilbar}\}$. Geben Sie einen deterministischen endlichen Automaten in Graphdarstellung an, der L_1 erkennt.

(b) Gegeben sei die Sprache $L_2 = \{w \in \{0,1\}^* \mid \text{bin}(w) \text{ ist durch 3 teilbar}\}$, wobei $\text{bin}(w)$ die natürliche Zahl ist, deren Binärdarstellung w ist. Der Einfachheit halber sei $\text{bin}(\varepsilon) = 0$, also durch k teilbar. Geben Sie einen deterministischen endlichen Automaten in Graphdarstellung an, der L_2 erkennt.

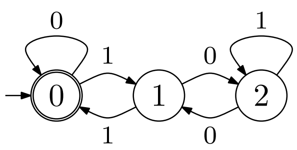
(c) Es seien $k, b \in \{1, 2, \dots\}$ und $\Sigma = \{0, 1, \dots, b-1\}$. Gegeben ist die Sprache $L_3 = \{w \in \Sigma^* \mid b\text{-adi}(w) \text{ ist durch } k \text{ teilbar}\}$, wobei b-adi(w) die natürliche Zahl ist, deren b-adische Darstellung w ist. Der Einfachheit halber sei $b\text{-adi}(\varepsilon) = 0$, also durch k teilbar. Geben Sie einen deterministischen endlichen Automaten an, der L_3 erkennt. Beweisen Sie die Korrektheit Ihrer Konstruktion.

Lösung:

(a)



(b)



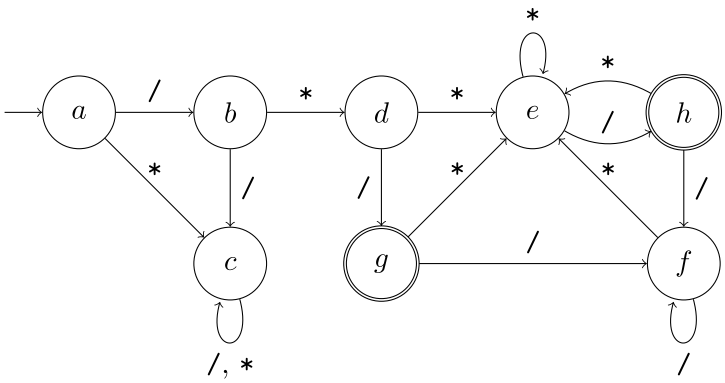
(c) Als Zustandsmenge wählen wir $Q = \{0, 1, \dots, k-1\}$. Die Idee ist, dass sich der Automat nach Abarbeiten eines (Teil-)Worts w im Zustand $b\text{-adi}(w) \bmod k$ befindet. Entsprechend ist der Startzustand 0 und die Menge der akzeptierenden Zustände $F = \{0\}$. Sei also $\mathcal{A}$ nach Abarbeiten des Wortes w im Zustand $q = b\text{-adi}(w) \bmod k$. Betrachte das Wort wa mit $a \in \Sigma$. Es gilt $b\text{-adi}(wa) = b \cdot b\text{-adi}(w) + a$. Nach den Gesetzen der modularen Arithmetik gilt $b \cdot b\text{-adi}(w) + a \mod k = b \cdot q + a \mod k$. Definiere also für alle $q \in Q, a \in \Sigma$ den Übergang $\delta(q, a) = b \cdot q + a \bmod k$. Damit ist $\mathcal{A}$ nach Abarbeiten des Wortes wa im Zustand $b\text{-adi}(wa) \bmod k$.

Man beachte, dass die Konstruktionen aus (a) und (b) Instanzen dieser allgemeineren Konstruktion sind.

Aufgabe 2

(3 Punkte)

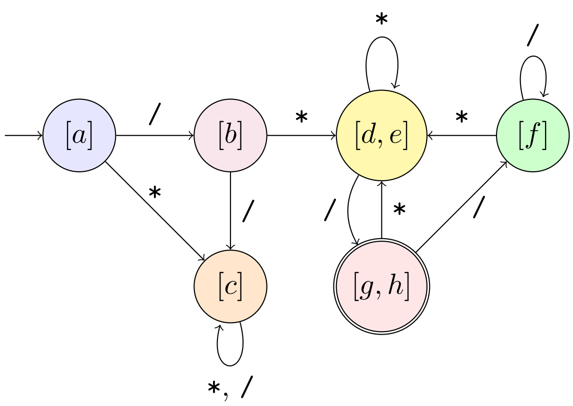
Konstruieren Sie zu folgendem Automaten den zugehörigen Äquivalenzklassenautomat. Geben Sie in jedem Schritt alle Äquivalenzklassen an, und durch welche Zeugen sie getrennt werden. Zeichnen Sie den Übergangsgraphen des Äquivalenzklassenautomaten.



Lösung:

Zeuge	Äquivalenzklassen
ε	$\{a,b,c,d,e,f\}$, $\{g,h\}$
/	$\{a,b,c,f\}$, $\{d,e\}$, $\{g,h\}$
*	$\{a,b,c,f\}$, $\{d,e\}$, $\{g,h\}$
//	$\{a,b,c,f\}$, $\{d,e\}$, $\{g,h\}$
/*	$\{a,b,c,f\}$, $\{d,e\}$, $\{g,h\}$
*/	$\{a,c\}$, $\{b,f\}$, $\{d,e\}$, $\{g,h\}$
**	$\{a,c\}$, $\{b,f\}$, $\{d,e\}$, $\{g,h\}$
///	$\{a,c\}$, $\{b,f\}$, $\{d,e\}$, $\{g,h\}$
//*	$\{a,c\}$, $\{b,f\}$, $\{d,e\}$, $\{g,h\}$
/*/	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
/**	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
*//	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
/	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
**/	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
***	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$

Alle Wörter der Länge 4 trennen keine weiteren Äquivalenzklassen.



Aufgabe 3 (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 2 durch, indem Sie die unten stehende Tabelle ausfüllen. Finden Sie dieselben Mengen wie die Äquivalenzklassen aus Aufgabe 2?

Lösung:

Schritt	Zeichen a	Partition nach Trennung durch a
1		$\{a,b,c,d,e,f\}$, $\{g,h\}$
2	/	$\{a,b,c,f\}$, $\{d,e\}$, $\{g,h\}$
3	*	$\{a,c\}$, $\{b,f\}$, $\{d,e\}$, $\{g,h\}$
4	/	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
5	*	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$
6	/	$\{a\}$, $\{c\}$, $\{b\}$, $\{f\}$, $\{d,e\}$, $\{g,h\}$

(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? Begründen Sie!

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 4

(1 + 1 + 1 + 2 + 1 = 6 Punkte)

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

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

(b) $\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}\}, L_b = \{w \in \Sigma^* \mid |w|_\texttt{a} + |w|_\texttt{b} = |w|_\texttt{c}\}$. Dabei bezeichnet $|w|_x$ die Anzahl der Vorkommen des Symbols x in w.

(c) $\Sigma = \{\texttt{0}, \texttt{1}\}$. Für ein Wort $w \in \Sigma^*$ sei $\text{neg}(w)$ die bitweise Negation von w, d.h. $\text{neg}(w)_i = 1 - w_i$, wobei w_i das i-te Zeichen von w beschreibt. $L_c = \{uv \in \Sigma^* \mid v = \text{neg}(u)\}$.

(d) Für eine beliebige natürliche Zahl $k \in \mathbb{N}$ definieren wir $L_k = \{w \in \{1, \dots, k\}^* \mid w \text{ ist eine monoton aufsteigende Zahlenfolge}\}$. 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 $L_\infty = \{w \in \mathbb{N}^* \mid w \text{ ist eine monoton aufsteigende Zahlenfolge}\}$ 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 $w = \texttt{a}^n\texttt{b}^n\texttt{c}^{2n}$. Offensichtlich gilt $w \in L_b$ mit $|w| > n$. Dann gilt $u = \texttt{a}^i$, $v = \texttt{a}^j$ für Zahlen i, j mit $i + j \leq n, j > 0$. Das Wort $uv^0x = \texttt{a}^{n-j}\texttt{b}^n\texttt{c}^{2n}$ liegt nicht in L_b, da $|uv^0x|_\texttt{a} = n - j < n, |uv^0x|_\texttt{b} = n, |uv^0x|_\texttt{c} = 2n$. Somit gibt es für jedes $n \in \mathbb{N}$ ein Wort 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_c nicht regulär sein kann.

Beweis: Sei $n \in \mathbb{N}$ beliebig, aber fest. Wir wählen $w = \texttt{0}^n\texttt{1}^n$. Offensichtlich gilt $w \in L_c$ mit $|w| > n$. Dann gilt $u = \texttt{0}^i$, $v = \texttt{0}^j$ für Zahlen i, j mit $i + j \leq n, j > 0$. Eine notwendige Bedingung dafür, dass ein Wort w in L_c enthalten ist, ist $|w|_\texttt{0} = |w|_\texttt{1}$. Das Wort $uv^0x = \texttt{0}^{n-j}\texttt{1}^n$ liegt daher nicht in L_c, da $|uv^0x|_\texttt{0} = n - j < n = |uv^0x|_\texttt{1}$. Somit gibt es für jedes $n \in \mathbb{N}$ ein Wort 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 $|w| > n = 1$. Wir wählen die Zerlegung $w = uvx$ mit $u = \varepsilon$ und $v = w_0$. Das Wort $uv^i x = v^i x$ enthält also i Kopien des ersten Zeichens am Wortanfang. Die Zeichenfolge in $v^i x$ 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 5

(2 + 2 + 3 = 7 Punkte)

In dieser Aufgabe betrachten wir Sprachen über dem Alphabet $\Sigma = \{0, 1, 2\} \subseteq \mathbb{N}$.

(a) Geben Sie für $L_1 = \{w = w_1 \dots w_n \in \Sigma^* \mid \forall i = 2, \dots, n : w_i \geq w_{i-1}\}$ alle 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: $uz \in L_1$.

(b) Geben Sie für L_1 einen DEA mit minimaler Anzahl von Zuständen an.

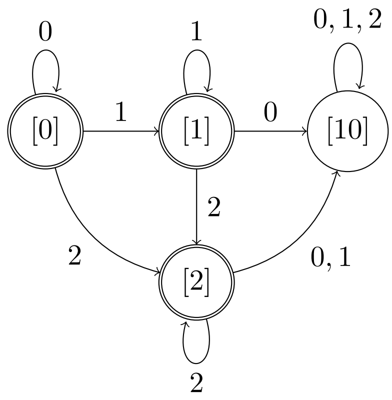
(c) Für ein Wort $w = w_1 \dots w_n$ definieren wir $q(w) := \sum_{i=1}^{n} w_i$. Wenn wir w als Zahl zur Basis 3 interpretieren, ist $q(w)$ also die Quersumme dieser Zahl. Zeigen Sie mithilfe der Nerode-Relation, dass $L_2 = \{w \in \Sigma^* \mid q(w) = |w|\}$ nicht regulär ist.

Lösung:

(a) L_1 ist regulär und hat 4 Äquivalenzklassen:

- $[0] = \{0\}^*$ und $S_{[i]} = L_1$.
- $[1] = \{w1 \mid w \in L_1 \cap \{0, 1\}^*\}$ und $S_{[1]} = L_1 \cap \{1, 2\}^*$.
- $[2] = \{w2 \mid w \in L_1\}$ und $S_{[2]} = \{2\}^*$.
- $[10] = L_1^c$ und $S_{[10]} = \emptyset$.

(b) Der Automat der Nerode-Relation ist minimal:



(c) Betrachte für eine beliebige Wortlänge n und $0 \leq i \leq n$ die Wörter $w_i = 0^i 1^{n-i}$ und $x_i = 2^i 1^{n-i}$. Es gilt $q(w_i) = n - i$ und $q(x_i) = n + i$. Dann sind für alle $i \neq j$ die Wörter w_i und w_j in verschiedenen Äquivalenzklassen, denn $w_i x_i \in L_2$ wegen $q(w_i x_i) = (n-i) + (n+j) = 2n = |w_i x_i|$, aber $w_j x_i \notin L_2$ wegen $q(w_j x_i) = 2n - j + i \neq 2n = |w_j x_i|$. Es gibt also mindestens n verschiedene Äquivalenzklassen. Da n beliebig gewählt war und L_2 Wörter beliebiger Länge erlaubt, gibt es unendlich viele Äquivalenzklassen und somit ist L_2 nicht regulär.

Aufgabe 6 (Bonus: 2 + 2 + 1 + 1 + 2 + 2 + 2 = 12 Punkte)

Wir betrachten typischerweise Sprachen über *endlichen* Alphabeten. Man kann sich aber auch überlegen, was passiert, wenn man Sprachen über *unendlichen* Alphabeten definiert. Genau das werden wir in dieser Aufgabe tun.

Sei also Σ_∞ eine (möglicherweise unendliche) Menge von Zeichen. Die Begriffe „Sprache" und „deterministischer endlicher Automat" übertragen sich direkt. Sei L_∞ eine Sprache über dem Alphabet Σ_∞, die von einem deterministischen endlichen Automaten $\mathcal{A}_\infty = (Q, \Sigma_\infty, \delta_\infty, s, F)$ erkannt wird. Beachten Sie, dass Q weiterhin eine endliche Menge sein muss. Wir definieren nun eine zweistellige Relation $\sim \subseteq \Sigma_\infty \times \Sigma_\infty$ mit folgender Vorschrift:

$$a \sim b \quad :\Longleftrightarrow \quad \forall q \in Q\colon \delta_\infty(q, a) = \delta_\infty(q, b)$$

Vergegenwärtigen Sie sich, dass $\sim$ eine Äquivalenzrelation ist. Zwei Zeichen a und b sind also genau dann bezüglich $\sim$ äquivalent, wenn sie von allen Zustandsübergängen in $\mathcal{A}_\infty$ gleich behandelt werden.

(a) Zeigen Sie, dass $\sim$ endlich viele Äquivalenzklassen hat.

(b) Zeigen Sie, dass ein endliches Alphabet Σ und eine Funktion $f : \Sigma_\infty \to \Sigma$ existieren, sodass die Sprache $L = \{f(w_1)f(w_2)\dots f(w_n) \mid w_1 w_2 \dots w_n \in L_\infty\}$ über dem Alphabet Σ von einem deterministischen endlichen Automaten erkannt wird.

Den Begriff einer regulären Sprache passen wir folgendermaßen an. Eine Sprache $L \subseteq \Sigma_\infty^*$ heißt ∞-*regulär*, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

- Verankerung:
  - $L \subseteq \Sigma_\infty$ oder

  - $L = \{\varepsilon\}$

- Induktion: Es gibt ∞-reguläre Sprachen L_1, L_2 sodass
  - $L = L_1 \cdot L_2$ oder
  - $L = L_1 \cup L_2$ oder
  - $L = L_1^*$

(c) Zeigen Sie, dass diese Definition zu der Definition von regulären Sprachen aus der Vorlesung äquivalent ist, wenn Σ_∞ endlich ist.

Sei L_∞ jetzt eine ∞-reguläre Sprache. Vergegenwärtigen Sie sich, dass die induktive Definition von L_∞ nur auf endlich viele Verankerungen zurückgreift. Bezeichne diese mit $L_1, L_2, \ldots, L_n$. Im Allgemeinen sind diese Mengen nicht paarweise disjunkt.

(d) Zeigen Sie, dass Sie L_∞ auch so definieren können, dass die Verankerungen paarweise disjunkt sind.

(e) Zeigen Sie, dass ein endliches Alphabet Σ und eine Funktion $f : \Sigma_\infty \to \Sigma$ existieren, sodass die Sprache $L = \{f(w_1)f(w_2)\ldots f(w_n) \mid w_1w_2\ldots w_n \in L_\infty\}$ regulär gemäß der Definition aus der Vorlesung ist.

Sie sind nun bereit zu zeigen, dass deterministische endliche Automaten mit möglicherweise unendlichem Eingabealphabet genau die ∞-regulären Sprachen erkennen können.

(f) Sei L_∞ eine Sprache, die von einen deterministischen endlichen Automaten $\mathcal{A}_\infty$ mit Eingabealphabet Σ_∞ erkannt wird. Zeigen Sie, dass L_∞ ∞-regulär ist.

    Verwenden Sie das entsprechende Ergebnis für endliche Automaten aus der Vorlesung und das Ergebnis aus Teilaufgabe (b).

(g) Sei L_∞ eine ∞-reguläre Sprache über dem Alphabet Σ_∞. Zeigen Sie, dass es einen deterministischen endlichen Automaten $\mathcal{A}_\infty$ mit Eingabealphabet Σ_∞ gibt, der L_∞ erkennt.

    Verwenden Sie das entsprechende Ergebnis für reguläre Sprachen aus der Vorlesung und das Ergebnis aus Teilaufgabe (e).

Lösung:

(a) Sei $Q = \{q_1, q_2, \ldots, q_n\}$. Für jedes $a \in \Sigma_\infty$ definieren wir das n-Tupel $\pi(a) := (\delta_\infty(q_1, a), \delta_\infty(q_2, a), \ldots, \delta_\infty(q_n, a)) \in Q^n$. Für $a, b \in \Sigma_\infty$ gilt $a \sim b$ genau dann, wenn $\pi(a) = \pi(b)$ gilt. Also hat $\sim$ höchstens $|Q^n| = n^n$ viele Äquivalenzklassen.

(b) Wähle Σ als Menge der Äquivalenzklassen von $\sim$. Nach (a) ist Σ endlich. Wähle ferner $f : a \mapsto [a]$. Definiere für jedes $q \in Q$ und $[a] \in \Sigma$ den Übergang $\delta(q, [a]) = \delta_\infty(q, a)$. Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$.

    Zeige nun, dass $\mathcal{A}$ gerade L erkennt. Weil s der Startzustand von $\mathcal{A}$ und $\mathcal{A}_\infty$ ist, gilt $\delta(s, \varepsilon) = \delta_\infty(s, \varepsilon)$. Nach Definition von δ gilt für alle $q \in Q, a \in \Sigma_\infty$ dass $\delta_\infty(q, a) = \delta(q, [a])$. Induktiv folgt für jedes $w_1w_2\ldots w_n \in L_\infty$, dass $\delta_\infty(s, w_1w_2\ldots w_n) = \delta(s, f(w_1)f(w_2)\ldots f(w_n))$. Also erkennt $\mathcal{A}$ die Sprache L.

(c) Offenbar ist jede reguläre Sprache wie in der Vorlesung definiert auch ∞-regulär. Betrachte umgekehrt eine ∞-reguläre Sprache mit endlichem Σ_∞. Jede Verankerung $L \subseteq \Sigma_\infty$ lässt sich durch endlich viele Vereinigungen $L = \bigcup_{a \in L}\{a\}$ erzeugen. Dann ist eine ∞-reguläre Sprache auch regulär gemäß der Definition aus der Vorlesung.

(d) Es gibt eine Partition von Σ_∞ in endlich viele Mengen $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$ so, dass jedes L_i eine Vereinigung von Elementen aus $\mathcal{B}$ ist. Die Definition von L lässt sich dann so anpassen, dass eine Verankerung an L_i durch endlich häufiges Vereinigen und Verankern an Elementen in $\mathcal{B}$ ersetzt wird.

(e) Gehe durch (d) motiviert o.B.d.A. davon aus, dass die Verankerungen $L_1, L_2, \ldots, L_n$ paarweise disjunkt sind. Definiere ferner $L_{n+1} = \Sigma_\infty \setminus \bigcup_{1 \le i \le n} L_i$. Sei $\sim$ eine Äquivalenzrelation, deren Äquivalenzklassen gerade $L_1, L_2, \ldots, L_{n+1}$ sind. Wähle wie oben $\Sigma = \{L_1, L_2, \ldots, L_{n+1}\}$ und $f : a \mapsto [a]$. Ersetze in der induktiven Definition von L_∞ eine Verankerung an L_i durch eine Verankerung an $\{L_i\}$ – man beachte $L_i \in \Sigma$ – um die induktive Definition von L zu erhalten.

(f) Sei L_∞ eine Sprache, die von einem deterministischen endlichen Automaten $\mathcal{A}_\infty$ mit Eingabealphabet Σ_∞ erkannt wird. Nach (b) existieren ein endliches Alphabet Σ und eine Funktion $f : \Sigma_\infty \to \Sigma$, sodass die Sprache $L = \{f(w_1)f(w_2)\ldots f(w_n) \mid w_1w_2\ldots w_n \in L_\infty\}$ über dem endlichen Alphabet Σ von einem deterministischen endlichen Automaten erkannt wird. Aus der Vorlesung ist ferner bekannt, dass L dann regulär gemäß der Definition aus Vorlesung ist. Ersetze in dieser Definition jede Verankerung $\{a\}$ durch die Verankerung $f^{-1}(a) \subseteq \Sigma_\infty$, um die Definition von L_∞ zu erhalten. Damit ist gezeigt, dass L_∞ ∞-regulär ist.

(g) Sei L_∞ eine ∞-reguläre Sprache über dem Alphabet Σ_∞. Nach (e) existieren ein endliches Alphabet Σ und eine Funktion $f : \Sigma_\infty \to \Sigma$, sodass die Sprache $L = \{f(w_1)f(w_2)\ldots f(w_n) \mid w_1w_2\ldots w_n \in L_\infty\}$ regulär gemäß der Definition aus der Vorlesung ist. Aus der Vorlesung ist ferner bekannt, dass dann auch ein endlicher deterministischer Automat $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ existiert, der L erkennt. Definiere nun $\mathcal{A}_\infty = (Q, \Sigma_\infty, \delta_\infty, s, F)$, wobei für alle $q \in Q, a \in \Sigma_\infty$ gelte, dass $\delta_\infty(q, a) = \delta(q, f(a))$.