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

Übungsblatt 1

Vorlesung Theoretische Grundlagen der Informatik im WS 18/19

Ausgabe 18. Oktober 2018
Abgabe 6. November 2018, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

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

Gegeben seien zwei Sprachen $L_1, L_2 \subseteq \Sigma^\star$ über dem Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Dabei sei L_1 die Sprache der Wörter, deren erstes und letztes Zeichen übereinstimmen, und L_2 die Sprache der Wörter, in denen auf jedes a ein b folgt. Geben Sie reguläre Ausdrücke für folgende Sprachen an.

(a) $L_1 \cup L_2$

(b) $L_1 \cdot L_2$

(c) $L_1 \cap L_2$

(d) $L_2 \setminus L_1$

(e) L_1^c

(f) L_2^c

Lösung:
Für $\alpha = \varepsilon \cup \mathsf{a} \cup \mathsf{b} \cup (\mathsf{a}(\mathsf{a} \cup \mathsf{b})^\star\mathsf{a}) \cup (\mathsf{b}(\mathsf{a} \cup \mathsf{b})^\star\mathsf{b})$ und $\beta = \mathsf{b}^\star(\mathsf{ab}^+)^\star$ gilt $L_1 = L(\alpha)$ und $L_2 = L(\beta)$. Anders ausgedrückt: α und β beschreiben die Sprachen L_1 beziehungsweise L_2. Demnach gilt:

(a) $L_1 \cup L_2 = L(\alpha \cup \beta)$

(b) $L_1 \cdot L_2 = L(\alpha \cdot \beta)$

(c) $L_1 \cap L_2 = L(\mathsf{b}^+(\mathsf{ab}^+)^\star \cup \varepsilon)$

(d) $L_2 \setminus L_1 = L((\mathsf{ab}^+)^+)$

(e) $L_1^c = L((\mathsf{a}(\mathsf{a} \cup \mathsf{b})^\star\mathsf{b}) \cup (\mathsf{b}(\mathsf{a} \cup \mathsf{b})^\star\mathsf{a}))$

(f) $L_2^c = L((\mathsf{a} \cup \mathsf{b})^\star\mathsf{aa}(\mathsf{a} \cup \mathsf{b})^\star \cup (\mathsf{a} \cup \mathsf{b})^\star\mathsf{a})$

Aufgabe 2

(2 + 2 = 4 Punkte)

In dieser Aufgabe sollen Sie zeigen, dass die Menge der regulären Sprachen unter der Komplementbildung und Schnittbildung abgeschlossen ist. Dazu seien ein beliebiges Alphabet Σ und zwei beliebige reguläre Sprachen L_1, L_2 gegeben. Für $i \in \{1, 2\}$ sei $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$ der deterministische endliche Automat, der L_i erkennt.

(a) Wandeln Sie $\mathcal{A}_1$ in einen deterministischen endlichen Automaten $\bar{\mathcal{A}}$ um, der die Sprache L_1^c erkennt.

(b) Geben Sie einen deterministischen endlichen Automaten $\mathcal{A}_\cap$ an, der die Sprache $L_1 \cap L_2$ erkennt.

Beweisen Sie jeweils, dass Ihr Automat die gewünschte Sprache erkennt.

Lösung:

(a) Der Automat $\bar{\mathcal{A}} = (Q_1, \Sigma, \delta_1, s_1, Q_1 \setminus F_1)$, in dem akzeptierende und nicht akzeptierende Zustände vertauscht wurden, erkennt L_1^c. Dies lässt sich wie folgt zeigen. Für beliebige Wörter $w \in \Sigma^\star$ gilt: $w \in L_1 \Leftrightarrow \delta_1^{|w|}(s_1, w) \in F_1$, wobei $\delta_1^{|w|}$ die $|w|$-fache Hintereinanderausführung von δ_1 ist. Durch Negation ergibt sich gerade $w \in L_1^c \Leftrightarrow w \notin L_1 \Leftrightarrow \delta_1^{|w|}(s_1, w) \in Q_1 \setminus F_1$.

(b) Es gibt zwei Wege, $\mathcal{A}_\cap$ zu konstruieren: durch Bildung des Produktautomaten oder durch Anwendung der ersten De Morganschen Regel.

Durch Anwendung der ersten De Morganschen Regel ergibt sich: $L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$. In der Vorlesung wurde gezeigt, wie sich ein nichtdeterministischer Automat konstruieren lässt, der die Vereinigung zweier Sprachen erkennt. Dieser lässt sich durch Potenzmengenkonstruktion deterministisch machen. Zusammen mit der Komplementbildung aus Aufgabenteil a) kann so ein Automat konstruiert werden, der $L_1 \cap L_2$ erkennt.

Die zweite Möglichkeit ist, einen Automaten zu konstruieren, der in jedem Überführungschritt „parallel" je einen Überführungsschritt der beiden ursprünglichen Automaten simuliert. Dies lässt sich durch den sogenannten Produktautomaten erreichen, bei dem die Zustandsmenge das Kreuzprodukt der ursprünglichen Zustandsmengen ist: $\mathcal{A}_\cap = (Q_1 \times Q_2, \Sigma, \delta_\cap, (s_1, s_2), F_1 \times F_2))$, wobei $\delta_\cap((q_1, q_2), w) = (\delta_1(q_1, w), \delta_2(q_2, w))$.

Ein Wort $w \in \Sigma^\star$ wird von $\mathcal{A}_\cap$ genau dann akzeptiert, wenn $\delta_\cap^{|w|}((s_1, s_2), w) \in F_1 \times F_2$ gilt. Durch wiederholte Anwendung der Definition von $\delta_\cap$ ergibt sich $\delta_\cap^{|w|}((s_1, s_2), w) = (\delta_1(s_1, w), \delta_2(s_2, w))$. Dementsprechend gilt: $\delta_\cap^{|w|}((s_1, s_2), w) \in F_1 \times F_2 \Leftrightarrow \delta_1(s_1, w) \in F_1 \wedge \delta_2(s_2, w) \in F_2$. Also wird w von $\mathcal{A}_\cap$ genau dann akzeptiert, wenn es sowohl von $\mathcal{A}_1$ als auch von $\mathcal{A}_2$ akzeptiert wird.

Aufgabe 3

(1 + 1 = 2 Punkte)

Gegeben seien zwei Grammatiken $G_i = (\Sigma, V, S, R_i), i = 1, 2$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}, V = \{S, A, B\}$ und folgenden Produktionsregeln

(a) $R_1 = \{S \to ABS \mid \varepsilon, A \to \mathsf{ac}A \mid \mathsf{ac}, B \to \mathsf{bc}B \mid \mathsf{bc}\}$

(b) $R_2 = \{S \to AB, A \to \mathsf{a}A\mathsf{bb} \mid \varepsilon, B \to \mathsf{b}B\mathsf{c} \mid \mathsf{bc}\}$

Ist $L(G_i)$ eine reguläre Sprache? Wenn ja, geben Sie einen regulären Ausdruck an. Wenn nicht, begründen Sie dies kurz (informell) und geben sie eine formale Definition der Sprache in Mengenschreibweise an.

Lösung:

(a) $((\mathsf{ac})^+(\mathsf{bc})^+)^\star$

(b) $L(G_2) = \{\mathsf{a}^i\mathsf{b}^{2i+j}\mathsf{c}^j \mid i \in \mathbb{N}_0, j \in \mathbb{N}\}$. Nein, DEA können nicht zählen.

Aufgabe 4

(2 + 1 + 4 = 7 Punkte)

Gegeben sei das Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Für ein Wort $v \in \Sigma^\star$ sei L_v die Sprache der Wörter über Σ, die mit v enden.

(a) Geben Sie einen deterministischen endlichen Automaten an, der L_{bab} erkennt. Verwenden Sie genau 4 Zustände.

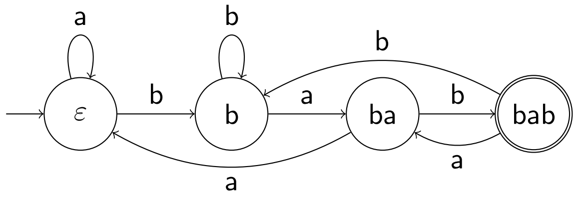
Sei nun ein beliebiges Wort $v \in \Sigma^\star$ der Länge n gegeben.

(b) Geben Sie eine formale Definition von L_v in Mengenschreibweise an.

(c) Geben Sie formal einen deterministischen endlichen Automaten $\mathcal{A}_v$ an, der L_v erkennt. Verwenden Sie genau $n+1$ Zustände. Begründen Sie, warum Ihr Automat deterministisch ist.

Lösung:

(a)



(b) $L_v = \{uv \mid u \in \Sigma^\star\}$

(c) Sei $\mathcal{A}_v = (Q, \Sigma, \delta, s, F)$. Wähle Zustandsmenge $Q = \{u \in \Sigma^* \mid u \text{ ist Präfix von } v\}$, Startzustand $s = \varepsilon$ und akzeptierende Zustandsmenge $F = \{v\}$. Wähle $\delta(q, \sigma)$ als das längste Wort $q' \in Q$, das ein Suffix von $q\sigma$ ist. Weil die Wörter in Q paarweise unterschiedliche Länge haben, ist die Überführungsvorschrift eindeutig und der Automat somit deterministisch.

Aufgabe 5 (1 + 2 = 3 Punkte)

(a) Offensichtlich lässt sich jeder deterministische endliche Automat (DEA) durch einen nichtdeterministischen endlichen Automaten (NEA) simulieren, indem der Nichtdeterminismus in der Überführungsfunktion nicht ausgenutzt wird. Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein NEA, der wie ein DEA arbeitet. Welche Bedingungen müssen für die Überführungsfunktion $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ gelten? Formulieren Sie die Bedingungen als prädikatenlogische Formel.

(b) Für NEA wurde in der Vorlesung die erweiterte Überführungsfunktion $\bar{\delta}\colon Q \times \Sigma^\star \to 2^Q$ eingeführt, die einen Zustand q und ein Wort w auf die Menge der Zustände abbildet, die von q bei der Abarbeitung von w erreichbar sind. Mithilfe der Simulation aus dem vorigen Aufgabenteil lässt sich $\bar{\delta}$ auch bei DEA anwenden. Kreuzen Sie in der folgenden Tabelle an, welche Aussage für *alle* NEA bzw. DEA $\mathcal{A}$ mit $L(\mathcal{A}) \neq \emptyset$ und $L(\mathcal{A}) \neq \Sigma^\star$ gelten.

Aussage	NEA	DEA
$\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		
$\exists w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$		
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$		
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$		
$\forall w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		
$\forall w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \notin F$		

Lösung:

(a) $\forall q \in Q\colon (\delta(q, \varepsilon) = \emptyset \land \forall x \in \Sigma\colon |\delta(q, x)| = 1)$

(b)

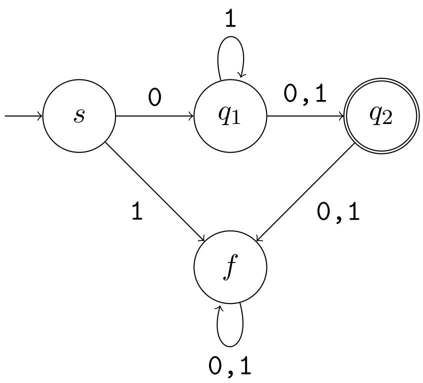
Aussage	NEA	DEA
$\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		×
$\exists w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$		×
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$	×	×
$\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$		
$\forall w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$		×
$\forall w \notin L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \notin F$	×	×

Aufgabe 6 (1 + 2 = 3 Punkte)

(a) Geben Sie für die Sprache über dem Alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, die von dem unten abgebildeten Automaten erkannt wird, einen regulären Ausdruck an.

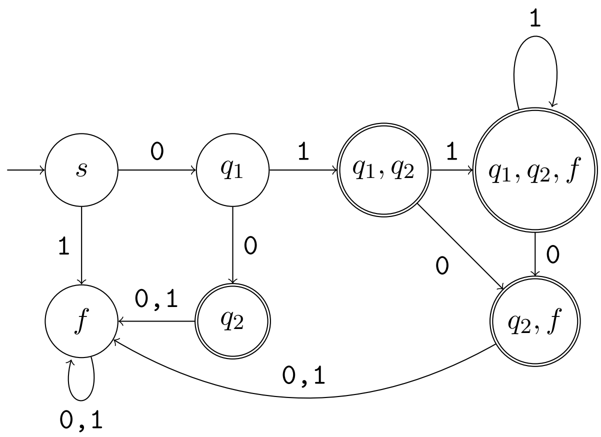
(b) Nutzen Sie die Potenzmengenkonstruktion aus der Vorlesung, um den Automaten in einen deterministischen endlichen Automaten zu überführen.

Lösung:



(a) $01^*(1 \cup 0)$

(b)



Aufgabe 7

(1 + 3 + 2 = 6 Punkte)

Der ebenso geniale, wie auch vergessliche Wissenschaftler und Superbösewicht Doktor Meta ist im Baufieber. Der Hauptzugang zu seinem unterirdischen Geheimlabor soll von einer unüberwindbaren Stahltür geschützt werden, die sich nur durch Eingabe eines gültigen Passworts öffnen lässt. Bei einer falschen Eingabe wird der Eindringling stattdessen durch eine Falltür im Boden den Haien vorgeworfen. Da sich Doktor Meta, aufgrund seiner Vergesslichkeit, nur schlecht an Passwörter erinnern kann, hat er die Tür so eingestellt, dass Sie jedes Wort aus einer zuvor festgelegten formalen Sprache L über einem endlichen Alphabet Σ akzeptiert. Dieses - sicherheitstechnisch fragwürdige, aber für Doktor Meta sehr praktische - Verfahren soll intern durch einen deterministischen endlichen Automaten realisiert werden. Da Sie als Doktor Metas neuer Sicherheitsexperte eingestellt worden sind, sollen Sie ihm bei einigen letzten Konfigurationen helfen:

(a) Kann der Öffnungsmechanismus der Falltür für jede formale Sprache L durch einen deterministischen endlichen Automaten realisiert werden? Warum?

(b) Sie haben sich mit Doktor Meta auf eine geeignete Sprache L geeignet. Leider hat einer ihrer Bauarbeiter, namentlich Ingo N. Kompetenz, das Schloss falsch verbaut. Die Kontrolleinheit der Tür bekommt das Passwort genau umgedreht zur Überprüfung. Da die Bauarbeiten bereits weit fortgeschritten sind, beauftragt Doktor Meta Sie, sich – natürlich erst nachdem Sie Herrn Kompetenz den Haien vorgeworfen haben – um eine Lösung des Problems zu bemühen. Wie könnten Sie den Automaten mit möglichst wenigen Änderungen umbauen, sodass er dennoch die gleiche Passwortmenge erkennt? Wieso funktioniert Ihr Ansatz? Ist Ihr Automat deterministisch oder nichtdeterministisch? Begründen Sie!

(c) Doktor Meta möchte einige Mitarbeiter unauffällig verschwinden lassen. Dafür hat er die Passwörter, die sie normalerweise eingeben, studiert und Muster[1] darin gefunden. Er beauftragt sie nun, die Kontrolleinheit so umzubauen, dass gerade die Muster der Leute, die er loswerden möchte von der Kontrolleinheit abgelehnt werden. Der Automat darf also nur noch eine Teilmenge der alten Sprache L akzeptieren. Können Sie den Automaten für jedes Muster entsprechend umbauen? Begründen Sie Ihre Antwort!

Lösung:

(a) Nein. Dafür muss L regulär sein.

(b) Sei $\mathcal{A} = \{Q, \Sigma, \delta, s, F\}$ ein deterministischer Automat, der L akzeptiert. Konstruiere NEA $\mathcal{A}' = \{Q', \Sigma', \delta', s', F'\}$ wie folgt:

- $Q' = Q \cup \{s'\}$ mit $s' \notin Q$, $\Sigma' = \Sigma$
- Für jedes $a \in \Sigma$ und alle $q_1, q_2 \in Q$, wenn $\delta(q_1, a) = q_2$, dann $q_1 \in \delta'(q_2, a)$. Folglich werden alle Zustandsübergänge *umgedreht*. So kann es zu nichtdeterministischen Übergängen kommen.
- Der Startzustand von $\mathcal{A}$ wird der akzeptierende Zustand von $\mathcal{A}'$: $F' = \{s\}$
- s' ist der Startzustand von $\mathcal{A}'$ mit $\delta'(s', \epsilon) = F$. Es gibt also ϵ-Übergänge.

Dieser NEA kann natürlich mit der Potenzmengenkonstruktion in einen DEA überführt werden.

(c) Nein. Denn die Sprache, die durch die Muster beschrieben wird, muss nicht regulär sein. Zum Beispiel könnten Muster der Form $\left\{a^i b^i \mid i \in \mathbb{N}\right\}$, also eine Teilmenge der regulären Sprache a^+b^+ ist, von einem DEA nicht erkannt werden kann.

[1] Muster sind beliebige Mengen von Wörtern über dem Alphabet Σ.