Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme

21. 07. 2014

Dr. Dominik D. Freydenberger
Dipl.-Inf. Joachim Bremer

Theoretische Informatik 2

Sommersemester 2014

Übungsblatt 13 (mit Lösungen)

Abgabe: *entfällt – für dieses Blatt gibt es auch keine Punkte.*

Aufgabe 1: **(0 Punkte)**

Sei $\Sigma := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Wir betrachten die DTD $D := (\Sigma, \mathtt{a}, \tau)$, wobei die Funktion τ folgendermaßen definiert sei:

$$\mathtt{a} \to \mathtt{bb} \mid \mathtt{c} \mid \varepsilon, \qquad \mathtt{b} \to \mathtt{c} \mid \varepsilon, \qquad \mathtt{c} \to (\mathtt{a} \cdot \mathrm{DATA})^+ \mid \mathrm{DATA}\,.$$

Welche der folgenden XML-Bäume über Σ sind unter D gültig?

Lösung: Wir gehen die Bäume von links nach rechts durch. In der Aufgabenstellung war zwar keine Begründung der Antwort gefragt; um eventuelle Unklarheiten auszuräumen werden die Antworten dennoch begründet:

(a) Die Wurzel ist mit $\rho = \mathtt{a}$ beschriftet, und da diese keine Kinder hat, muss $\varepsilon \in \mathcal{L}(\tau(\mathtt{a}))$ gelten. Dies ist der Fall; somit sind alle Bedingungen erfüllt, und der Baum ist *gültig*.

(b) Der mit $\mathtt{c}$ beschriftete Knoten hat keine Kinder. Damit dieser Baum gültig ist, müsste also $\varepsilon \in \mathcal{L}(\tau(\mathtt{c}))$ gelten. Allerdings ist $\tau(\mathtt{c}) = (\mathtt{a} \cdot \mathrm{DATA})^+ \mid \mathrm{DATA}$, und somit $\varepsilon \notin \mathcal{L}(\tau(\mathtt{c}))$. Dieser Baum ist also *nicht gültig*.

(c) Die Wurzel ist korrekt mit $\mathtt{a}$ beschriftet; an den einzelnen Knoten bilden die Kinder die folgenden Wörter:

$$\begin{aligned} \mathtt{c} &\in \mathcal{L}(\tau(\mathtt{a})) = \mathcal{L}(\mathtt{bb} \mid \mathtt{c} \mid \varepsilon), \\ \mathtt{a} \cdot \mathrm{DATA} &\in \mathcal{L}(\tau(\mathtt{c})) = \mathcal{L}((\mathtt{a} \cdot \mathrm{DATA})^+ \mid \mathrm{DATA}), \\ \varepsilon &\in \mathcal{L}(\tau(\mathtt{a})) = \mathcal{L}(\mathtt{bb} \mid \mathtt{c} \mid \varepsilon). \end{aligned}$$

Da alle Bedingungen erfüllt sind, ist der Baum *gültig*.

(d) Hier ergeben die Kinder des mit $\mathtt{c}$ beschrifteten Knoten das Wort $\mathrm{DATA} \cdot \mathrm{DATA} \notin \mathcal{L}(\tau(\mathtt{c}))$, Da $\tau(\mathtt{c}) = (\mathtt{a} \cdot \mathrm{DATA})^+ \mid \mathrm{DATA}$. Dieser Baum ist also *nicht gültig*.

(e) Bei diesem Baum ist die Wurzel korrekt mit $\mathtt{a}$ beschriftet, es gilt

$$\mathtt{bb} \in \mathcal{L}(\tau(\mathtt{a})) = \mathcal{L}(\mathtt{bb} \mid \mathtt{c} \mid \varepsilon),$$
$$\varepsilon \in \mathcal{L}(\tau(\mathtt{b})) = \mathcal{L}(\mathtt{c} \mid \varepsilon).$$

Dieser Baum ist daher *gültig*.

(f) Dieser Baum ist mit $\mathtt{b}$ beschriftet (statt mit $\rho = \mathtt{a}$), also ist er *nicht gültig*.

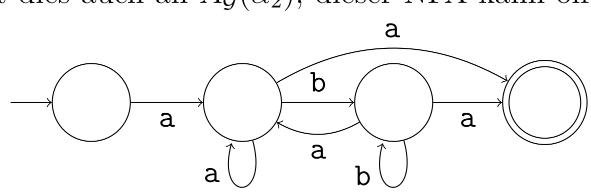
Aufgabe 2: **(0 Punkte)**

Sei $\Sigma := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Welche dieser regulären Ausdrücke sind deterministisch, welche nicht?

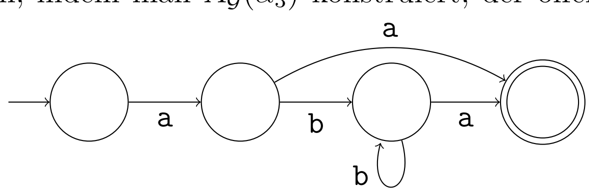
$$\alpha_1 := (\mathtt{a}^*\mathtt{b}^*)^+, \qquad \alpha_2 := (\mathtt{a}^*\mathtt{b}^*)^*\mathtt{a}, \qquad \alpha_3 := \mathtt{ab}^*\mathtt{a},$$
$$\alpha_4 := ((\mathtt{a} \mid \varepsilon)\mathtt{b}) \mid \mathtt{b}, \qquad \alpha_5 := (\mathtt{ab} \mid \mathtt{cb}), \qquad \alpha_6 := (\mathtt{ba} \mid \mathtt{bc}).$$

Lösung: Wir betrachten die einzelnen Ausdrücke. Auch hier war keine Begründung gefordert; als zusätzliche Hilfestellung wird diese dennoch angegeben.

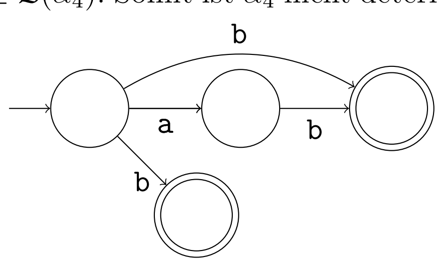
- α_1 ist *deterministisch*. Dies ist leicht zu sehen: Da jedes Terminal nur ein einziges Mal in α_1 vorkommt, muss dieser Ausdruck deterministisch sein (gemäß der Definition von nicht-deterministischen regulären Ausdrücken).
- α_2 ist *nicht deterministisch*. Wir betrachten dazu den indizierten Ausdruck $\widetilde{\alpha_2} = (\mathtt{a}_1^*\mathtt{b}_1^*)^*\mathtt{a}_2$. Nun gilt: $a_1a2 \in \mathcal{L}(\widetilde{\alpha_2})$, und $a_1a2 \in \mathcal{L}(\widetilde{\alpha_2})$. Für $x := \varepsilon$, $z_1 := \varepsilon$ und $z_2 := \mathtt{a}_2$ ist $x\mathtt{a}_1z_1 \in \mathcal{L}(\widetilde{\alpha_2})$ $\mathtt{a}_1z_2 \in \mathcal{L}(\widetilde{\alpha_2})$. Man sieht dies auch an $A_{\mathcal{G}}(\alpha_2)$; dieser NFA kann offensichtlich nicht als DFA interpretiert werden:



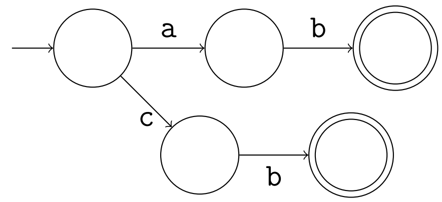
- α_3 ist *deterministisch*. Jedes Wort aus $\mathcal{L}(\alpha_3)$ enthält genau zwei Vorkommen von $\mathtt{a}$; und zwar als ersten und letzten Buchstaben. Der erste Buchstabe muss dabei vom linken Vorkommen von $\mathtt{a}$ in α_3 erzeugt werden, der letzte vom rechten Vorkommen. Daher ist es auch unmöglich, Wörter aus $\mathcal{L}(\widetilde{\alpha_3})$ so zu zerlegen, dass gegen die Kriterien von deterministischen regulären Ausdrücken verstoßen wird. Man kann auch zu dieser Erkenntnis kommen, indem man $A_{\mathcal{G}}(\alpha_3)$ konstruiert, der offensichtlich als DFA interpretiert werden kann:



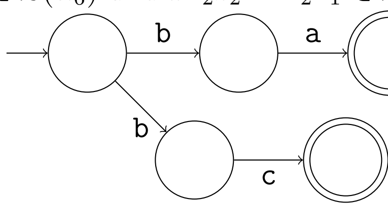
- α_4 ist *nicht deterministisch*. Sei $x := z_1 := z_2 := \varepsilon$. Dann ist $x\mathtt{b}_1z_1 = \mathtt{b}_1 \in \mathcal{L}(\widetilde{\alpha_4})$ und $x\mathtt{b}_2z_2 = \mathtt{b}_2 \in \mathcal{L}(\widetilde{\alpha_4})$. Somit ist α_4 nicht deterministisch. Stattdessen kann man auch $A_{\mathcal{G}}(\alpha_4)$ betrachten:



- α_5 ist *deterministisch*. Hier kann, ähnlich wie bei α_3, die eindeutige Zerlegbarkeit der Wörter aus $\mathcal{L}(\widetilde{\alpha_3})$ festgestellt werden. Oder man betrachtet den Glushkov-Automaten $A_{\mathcal{G}}(\alpha_5)$, der als DFA interpretiert werden kann:



- α_6 ist *nicht deterministisch*. Wir wählen $x := \varepsilon$, $z_1 := \mathtt{a}_1$, $z_2 := \mathtt{c}_1$. Dann gilt: $x\mathtt{b}_1 z_1 = \mathtt{b}_1\mathtt{a}_1 \in \mathcal{L}(\widetilde{\alpha_6})$ und $x\mathtt{b}_2 z_2 = \mathtt{b}_2\mathtt{c}_1 \in \mathcal{L}(\widetilde{\alpha_6})$. Alternativ dazu können wir auch $A_{\mathcal{G}}(\alpha_6)$ betrachten:



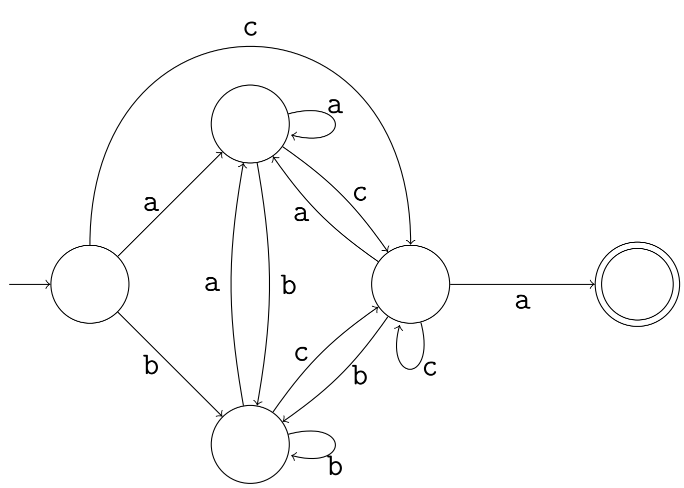
Aufgabe 3: **(0 Punkte)**

Über dem Alphabet $\Sigma := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ sei der reguläre Ausdruck α definiert als

$$\alpha := ((\mathtt{a} \mid \mathtt{b})^*\mathtt{c})^+\mathtt{a}.$$

Geben Sie einen NFA A an, so dass $\mathcal{L}(A) = \mathcal{L}(\alpha)$. Wie Sie den NFA konstruieren ist Ihnen überlassen; aber beachten Sie, dass A keine ε-Übergänge enthalten darf.

Lösung: Sie können A konstruieren, indem Sie α in einen ε-NFA umwandeln und aus diesem dann die ε-Übergänge entfernen. Alternativ dazu können Sie auch versuchen, A direkt zu konstruieren, da $\mathcal{L}(\alpha)$ eine vergleichsweise einfache Sprache ist. Auf diese Lösungen gehen wir hier nicht weiter ein, da dies schon in der ersten Hälfte des Semesters geübt wurde. Eine weitere Lösungsmöglichkeit ist die Konstruktion des Glushkov-Automaten $A_{\mathcal{G}}(\alpha)$:



Aufgabe 4: **(0 Punkte)**

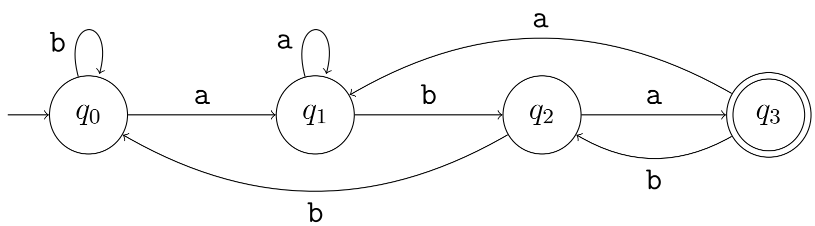
Geben Sie jeweils einen minimalen DFA für die folgenden Sprachen an:

$$L_1 := \Sigma_1^* \cdot \{\mathtt{aba}\}, \qquad L_2 := \Sigma_2^* \cdot \{\mathtt{abc}\}, \qquad L_3 := \Sigma_2^* \cdot \{\mathtt{abaac}\}.$$

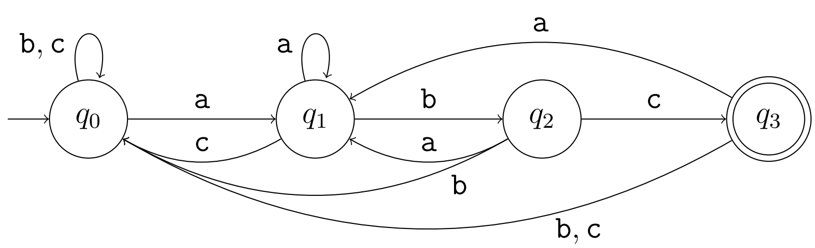
wobei $\Sigma_1 := \{\mathtt{a}, \mathtt{b}\}$ und $\Sigma_2 := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

Lösung:

Für L_1:



Für L_2:



Für L_3: