Technische Universität München
Fakultät für Informatik
Prof. Tobias Nipkow, Ph.D.
Sascha Böhme, Lars Noschinski

Sommersemester 2011
Lösungsblatt 3
6. Juni 2011

Einführung in die Theoretische Informatik

Hinweis: *Bitte beachten Sie unbedingt die Hinweise zum Übungsablauf und zu den Aufgabentypen auf der THEO-Website (`http://theo.in.tum.de/`).*

Hausaufgabe 1 (5 Punkte)

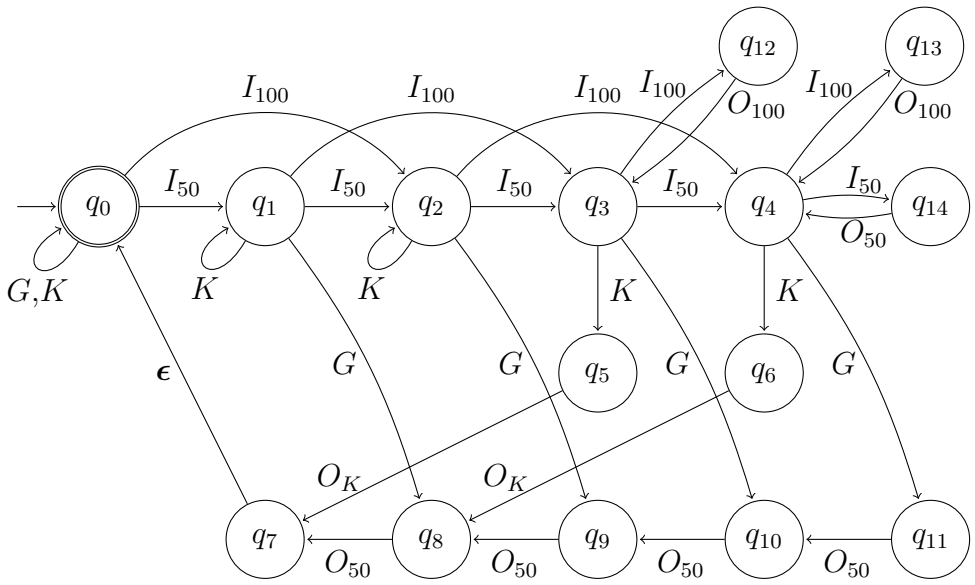
Wir betrachten einen Kaffeeautomaten mit folgender Spezifikation:

- Der Automat hat unendlich viel Wechselgeld und Zutaten zur Verfügung.
- Eingeworfen werden können 50-Cent- und 1-Euro-Münzen (I_{50}, I_{100}). Der Automat speichert ein Guthaben von höchstens zwei Euro. Wird durch den Einwurf einer weiteren Münze dieser Betrag überschritten, so wird diese Münze direkt zurückgegeben (O_{50}, O_{100}).
- Außer der Münzeingabe hat der Automat noch Knöpfe für Geldrückgabe (G) und einen extra großen Kaffee für 1,50 EUR (K). Verlangt der Benutzer einen Kaffee und ist genug Guthaben im Automaten, so wird zunächst das gewünschte Getränk ausgegeben (O_K) und danach das Restguthaben ausgezahlt. Wurde nicht genug Geld eingeworfen, so passiert nichts.

  Bei einem Druck auf Geldrückgabe wird das eingeworfene Guthaben ausgezahlt.
- Die Auszahlung eines (Rest-)Guthabens erfolgt immer in 50-Cent-Münzen.

Modellieren Sie nun einen endlichen Automaten, der genau die Wörter akzeptiert, die eine gültige Interaktionssequenz mit dem Kaffeeautomaten darstellen und bei denen kein Guthaben im Automaten verbleibt.
Verwenden Sie dazu das Alphabet $\Sigma = \{I_{50}, I_{100}, O_{50}, O_{100}, G, K, O_K\}$.

Lösungsvorschlag



Hausaufgabe 2 (5 Punkte)

Die *Dropout-Wörter* zu einem Wort w sind alle Wörter, die durch das Entfernen genau eines Buchstabens aus w entstehen. Für eine Sprache L nennen wir die Sprache L_D die aus den Dropout-Wörtern der Wörter aus L besteht, *Dropout-Sprache*. Formal schreiben wir: $L_D = \{uv \mid \exists a.\ uav \in L\}$.

1. Geben Sie je einen regulären Ausdruck für die Dropout-Sprachen zu $(ab)^*$ und $a((a|b)b)^*$ an.

2. Zeigen Sie, dass L_D regulär ist für jede reguläre Sprache L.

Lösungsvorschlag

1. - Zu $(ab)^*$: $(ab)^*(a|b)(ab)^*$.
   - Zu $a((a|b)b)^*b$: $((a|b)b)^* \mid a((a|b)b)^*(a|b)((a|b)b)^*$

2. Wir zeigen die Regularität von L_D, indem wir aus einem NFA $M = (Q, \Sigma, \delta, q_0, F)$ für L einen ϵ-NFA für L_D konstruieren.

   Der Automat für L_D soll M simulieren, aber genau einmal die Möglichkeit bieten, auf einen Nachfolgezustand zu "springen" ohne ein Zeichen zu lesen. Dazu müssen wir uns im Zustand das Automaten merken, ob wir bereits einmal gesprungen sind. Wir setzen zu diesem Zweck $Q' = Q \times \{1, 2\}$. Dabei entspricht $(q, 1)$ einem Zustand im ursprünglichen Automaten, vor dem Sprung, und $(q, 2)$ dem selben Zustand im ursprünglichen Automaten, jedoch nach dem Sprung. Sowohl für $Q \times \{1\}$ als auch für $Q \times \{2\}$ simulieren wir zunächst den Ursprungsautomaten:

$$\delta'((q, i), a) = \{(q', i) \mid q' \in \delta(q, a)\} \qquad i \in \{1, 2\}$$

Wenn wir ein Zeichen überspringen, wechseln wir damit gleichzeitig in die zweite Hälfte des Automaten. Damit stellen wir sicher, dass höchstens einmal gesprungen werden kann:

$$\delta'((q,1),\boldsymbol{\epsilon}) = \{(q',2) \mid a \in \Sigma \wedge q' \in \delta(q,a)\}$$

Um sicherzustellen, dass auch mindestens einmal gesprungen wird, wählen wir den Startzustand in der ersten und die Endzustände in der zweiten Hälfte:

$$M' = \{Q', \{\boldsymbol{\epsilon}\} \cup \Sigma, \delta', (q_0,1), F \times \{2\}\}$$

Nun bleibt noch zu zeigen, dass M' wirklich L_D akzeptiert. Zunächst einmal stellen wir fest, dass $\hat{\delta'}((q_0,1),w) \cap F \neq \varnothing$ nur dann gelten kann, wenn $w \in \{\boldsymbol{\epsilon}\} \cup \Sigma^*$ von der Form $w = u\boldsymbol{\epsilon}v$ für $u, v \in \Sigma^*$ ist. Denn der Startzustand liegt in der ersten Hälfte, die Endzustände in der zweiten, und ein Wechsel ist nur mit einer $\boldsymbol{\epsilon}$-Transition möglich, da alle Transitionen, die nicht von der ersten in die zweiten Hälfte führen, mit Zeichen aus Σ^* beschriftet sind.

Wir haben

$$\begin{aligned}
\hat{\delta'}((q_0,1),u\boldsymbol{\epsilon}v) &= \hat{\overline{\delta'}}\left(\overline{\delta'}(\hat{\delta'}((q_0,1),u),\boldsymbol{\epsilon}),v\right) \\
&= \hat{\overline{\delta'}}\left(\overline{\delta'}\left(\{(q,1) \mid q \in \hat{\delta}(q_0,u)\},\boldsymbol{\epsilon}\right),v\right) \\
&= \hat{\overline{\delta'}}\left(\left\{(q',2) \mid q \in \hat{\delta}(q_0,u) \wedge a \in \Sigma \wedge q' \in \delta(q,a)\right\},v\right) \\
&= \hat{\overline{\delta'}}\left(\left\{(q,2) \mid a \in \Sigma \wedge q \in \hat{\delta}(q_0,ua)\right\},v\right) \\
&= \left\{(q',2) \mid a \in \Sigma \wedge q \in \hat{\delta}(q_0,ua) \wedge q' \in \hat{\delta}(q,v)\right\} \\
&= \left\{(q,2) \mid a \in \Sigma \wedge q \in \hat{\delta}(q_0,uav)\right\} \\
&= \left(\bigcup_{a\in\Sigma} \hat{\delta}(q_0,uav)\right) \times \{2\}
\end{aligned}$$

Damit gilt $\hat{\delta'}((q_0,1),u\boldsymbol{\epsilon}v) \cap F \times \{2\} \neq \varnothing$ genau dann, wenn ein $a \in \Sigma$ existiert, so dass $\hat{\delta}(q,uav) \cap F = \varnothing$. Zusammen mit den Vorüberlegungen folgt also, dass die von M' akzeptierte Sprache genau L_D ist.

Hausaufgabe 3 (3 Punkte)

Wir bezeichnen die Menge er regulären Ausdrücke über einem Alphabet Σ mit RE_Σ.

1. Sei $a \in \Sigma$. Definieren Sie eine rekursive Funktion $\mathit{contains}_a : \mathrm{RE}_\Sigma \to \{\mathrm{true}, \mathrm{false}\}$, so dass gilt:

$$\mathit{contains}_a(\alpha) \iff L(\alpha) \neq \varnothing \wedge L(\alpha) \subseteq \{uav \mid u, v \in \Sigma^*\}$$

2. Definieren Sie zwei rekursive Funktionen

$$\mathit{even} : \mathrm{RE}_\Sigma \to \{\mathrm{true}, \mathrm{false}\} \quad \text{und} \quad \mathit{odd} : \mathrm{RE}_\Sigma \to \{\mathrm{true}, \mathrm{false}\}$$

so dass $\mathit{even}(\alpha) = \mathrm{true}$ gilt gdw. alle Wörter in $L(\alpha)$ gerade Länge haben und so dass $\mathit{odd}(\alpha) = \mathrm{true}$ gilt gdw. alle Wörter in $L(\alpha)$ ungerade Länge haben.

Lösungsvorschlag

1. Wir definieren:

$$
\begin{aligned}
\mathit{contains}_a(\emptyset) &= \text{false} \\
\mathit{contains}_a(\epsilon) &= \text{false} \\
\mathit{contains}_a(b) &= \begin{cases} \text{true} & \text{falls } a = b \\ \text{false} & \text{sonst} \end{cases} \\
\mathit{contains}_a(\alpha\beta) &= \neg \mathit{empty}(\alpha) \wedge \neg \mathit{empty}(\beta) \wedge (\mathit{contains}_a(\alpha) \vee \mathit{contains}_a(\beta)) \\
\mathit{contains}_a(\alpha|\beta) &= \mathit{contains}_a(\alpha) \wedge \mathit{contains}_a(\beta) \\
\mathit{contains}_a(\alpha^*) &= \text{false}
\end{aligned}
$$

   wobei:

$$
\begin{aligned}
\mathit{empty}(\emptyset) &= \text{true} \\
\mathit{empty}(\epsilon) &= \text{false} \\
\mathit{empty}(a) &= \text{false} \\
\mathit{empty}(\alpha\beta) &= \mathit{empty}(\alpha) \vee \mathit{empty}(\beta) \\
\mathit{empty}(\alpha|\beta) &= \mathit{empty}(\alpha) \wedge \mathit{empty}(\beta) \\
\mathit{empty}(\alpha^*) &= \text{false}
\end{aligned}
$$

2. Wir definieren:

$$
\begin{aligned}
\mathit{even}(\emptyset) &= \text{true} \\
\mathit{even}(\epsilon) &= \text{true} \\
\mathit{even}(a) &= \text{false} \\
\mathit{even}(\alpha\beta) &= \mathit{even}(\alpha) \wedge \mathit{even}(\beta) \vee \mathit{odd}(\alpha) \wedge \mathit{odd}(\beta) \\
\mathit{even}(\alpha|\beta) &= \mathit{even}(\alpha) \wedge \mathit{even}(\beta) \\
\mathit{even}(\alpha^*) &= \mathit{even}(\alpha)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{odd}(\emptyset) &= \text{true} \\
\mathit{odd}(\epsilon) &= \text{false} \\
\mathit{odd}(a) &= \text{true} \\
\mathit{odd}(\alpha\beta) &= \mathit{even}(\alpha) \wedge \mathit{odd}(\beta) \vee \mathit{odd}(\alpha) \wedge \mathit{even}(\beta) \\
\mathit{odd}(\alpha|\beta) &= \mathit{odd}(\alpha) \wedge \mathit{odd}(\beta) \\
\mathit{odd}(\alpha^*) &= \text{false}
\end{aligned}
$$

Hausaufgabe 4 (3 Punkte)

Die Permutationssprache L_P zu einer Sprache L besteht aus allen Permutationen aller Wörter in L. Formal setzen wir:

$$L_P = \{u_{\pi(1)} \cdots u_{\pi(k)} \mid k \in \mathbb{N},\ \pi \text{ Permutation und } u_1 \cdots u_k \in L\}$$

1. Geben Sie eine alternative, möglichst einfache Beschreibung der Permutationssprache zu der Sprache $L = L((ab)^*)$ an.

2. Verwenden Sie das Pumping-Lemma, um anhand dieser Sprache zu zeigen, dass reguläre Sprachen nicht abgeschlossen sind unter Permutation (d.h. für eine reguläre Sprache L ist L_P nicht zwangsläufig regulär).

Lösungsvorschlag

1. L_P ist die Sprache der Wörter, die aus gleich vielen a und b bestehen.

2. Angenommen, L_P sei regulär. Nach dem Pumping-Lemma können wir nun ein $n > 0$ wählen, so dass für alle $z \in L_P$ mit $|z| \geq n$ eine Zerlegung $z = uvw$ existiert mit $v \neq \epsilon$, $|uv| \leq n$ und $uv^iw \in L_P$ für alle $i \geq 0$. Wir nennen n eine Pumping-Lemma-Zahl.

   Es gilt $a^nb^n \in L_P$. Sei also $z = a^nb^n$. Für jede Zerlegung $z = uvw$ mit $|uv| \leq n$ und $v \neq \epsilon$ gilt $v = a^m$ für ein $m > 0$. Dann besteht aber uv^2w aus $m + n$ Vorkommen von a und n Vorkommen von b und ist damit nicht in L_P enthalten.

   Somit erhalten wir einen Widerspruch zur Regularität von L_P.

Hausaufgabe 5 (4 Punkte)

Widerlegen Sie die Regularität der folgenden Sprachen (mit Hilfe des Pumping-Lemmas):

1. $L_1 = \{w \in \{0,1\}^* \mid w^R = w\}$

2. $L_2 = \{w \in \{0,1\}^* \mid |w|_0 \geq |w|_1\}$

Dabei bedeutet w^R die Umkehrung von w ("rückwärtslesen"), und $|w|_a$ gibt die Anzahl der Zeichen a in einem Wort w an.

Lösungsvorschlag

1. Wir nehmen an, dass L_1 regulär ist und leiten mittels des Pumping-Lemmas daraus einen Widerspruch her.

   Sei $n \in \mathbb{N}$ eine Pumping-Lemma-Zahl. Dann ist sicherlich $0^n10^n \in L_1$. Es gibt also für z eine Zerlegung uvw mit $v \neq \epsilon$ und $|uv| \leq n$, so dass $(*)$ $uv^iw \in L_1$ für alle $i \in \mathbb{N}$. Offensichtlich muss $uv = 0^k$ für $0 < k \leq n$ gelten, also $u = 0^i$ und $v = 0^j$ mit $i + j = k$ und $j > 0$. Dann ist aber $uv^2w = 0^i0^{2j}0^{n-k}10^n \notin L_1$, denn $i + 2j + n - k > n$. Dies steht im Widerspruch zu $(*)$, d.h. unsere ursprüngliche Annahme, dass L_1 regulär ist, kann nicht stimmen.

2. Angenommen, L_2 wäre regulär. Dann können wir das Pumping-Lemma anwenden. Sei also $n \in \mathbb{N}$ eine Pumping-Lemma-Zahl. Das Wort $z = 0^n1^n$ ist in L_2. Es gibt daher eine Zerlegung uvw mit $v \neq \epsilon$, $|uv| \leq n$ und $uv^iw \in L_2$ für alle $i \in \mathbb{N}$. Aus unserer Wahl von z folgt, dass $u = 0^i$ und v^j mit $j > 0$ gilt. Allerdings ist $uv^0w = 0^i0^{n-i-j}1^n \notin L_2$, ein Widerspruch zum Pumping-Lemma. Also kann L_2 nicht regulär sein.

Quiz 1

Beantworten Sie kurz die folgenden Fragen:

1. Wenn L regulär ist und $L' \subseteq L$ gilt, ist dann auch L' regulär?
2. Finden Sie ein Verfahren, welches für einen gegebenen NFA M entscheidet, ob $|L(M)| \leq 100$ gilt.
3. Ist die Sprache $L = \{a^n b^m \mid n+m \text{ gerade}\}$ regulär? Lässt sich das mit dem Pumping-Lemma zeigen?

Lösungsvorschlag

1. Nein, denn $L(a^*b^*)$ ist regulär, nicht aber $L(a^n b^n)$, obwohl letzteres eine Teilsprache von ersterem ist.
2. Zunächst testet man, ob $L(M)$ endlich ist. Wenn nein, sind wir fertig. Wenn ja, gibt es einen längsten Pfad (der Länge n), der von einem Startzustand zu einem Endzustand führt. Jetzt zählt man alle Wörter bis zur Länge n auf und testet, ob der Automat sie akzeptiert.
3. Die Sprache ist regulär, denn $L = L\left((aa)^*(\epsilon|aa|ab|bb)(bb)^*\right)$. Mit dem Pumping-Lemma lässt sich jedoch nicht zeigen, dass eine Sprache regulär ist, sondern nur, dass sie *nicht* regulär ist.

Tutoraufgabe 1

1. Sei $\Sigma = \{a, b, c, d, e\}$.
   Zeigen Sie, dass die Sprache $L = \{ab^{2i}cd^i e \mid i \in \mathbb{N}\}$ nicht regulär ist.
2. Sei $\Sigma = \{1\}$. Zeigen Sie, dass die Sprache $P = \{1^p \mid p \text{ prim}\}$ nicht regulär ist.

Lösungsvorschlag

1. Wir nehmen an, dass L regulär ist, und leiten wie folgt einen Widerspruch her.

   Zunächst folgt für eine Pumping-Lemma Zahl $n \in \mathbb{N}$ mit $n > 0$ für alle $z \in L$ mit $|z| \geq n$ die Eigenschaft

   $$\begin{aligned} \exists u, v, w \in \Sigma^* : \quad & 1. \quad z = uvw\,, \\ & 2. \quad v \neq \epsilon\,, \\ & 3. \quad |uv| \leq n\,, \\ & 4. \quad uv^i w \in L, \forall i \geq 0\,. \end{aligned}$$

   Dass diese Eigenschaft nicht für alle $z \in L$ mit $|z| \geq n$ gelten kann, zeigt man z. B. mit dem Wort $z = ab^{2n}cd^n e$.

   Für dieses Wort z ist zunächst klar, dass $|z| \geq n$. Und es ist klar, dass z in L enthalten ist. Deshalb kann man eine Zerlegbarkeit von z annehmen, so dass Eigenschaften 1. bis 4. gelten.

Sei also $z = uvw$ mit $v \neq \epsilon$, $|uv| \leq n$ und $uv^i w \in L, \forall i \geq 0$.

Wegen $|uv| \leq n$ kann v kein c, d oder e enthalten. Dann muss also v aus a's und/oder b's bestehen, und, da v nicht leer ist, folglich mit a oder b enden. Beide Fälle führen wir wie folgt zum Widerspruch.

Im Folgenden bezeichnen wir mit $|w|_x$ die Anzahl der Buchstaben x, die in w enthalten sind.

Falls v mit a endet, dann gilt $|uv^2w|_a > 1$. Daraus folgt $uv^2w \notin L$ im Widerspruch zu Eigenschaft 4. für $i = 2$.

Falls v mit b endet, dann gilt $|uv^0w|_b < 2n$ und $|uv^0w|_d = n$.

Daraus folgt ebenfalls $uv^0w \notin L$ im Widerspruch zu Eigenschaft 4. für $i = 0$.

Man beachte, dass im Beweis alle Eigenschaften 1. bis 4. verwendet wurden.

2. Wir nehmen an, dass $L = \{1^p \mid p \text{ prim}\}$ regulär ist. Sei n eine Pumping-Lemma-Zahl.

   Da die Menge der Primzahlen bekanntlich unendlich ist, gibt es eine Primzahl p bzw. ein $z = 1^p \in L$, so dass z mindestens n Zeichen enthält, also $|z| \geq n$.

   Sei also $z \in L$ mit $|z| \geq n$ und sei $z = uvw$ mit $|uv| \leq n$, v nicht leer und $uv^i w \in L$ für alle $i \geq 0$.

   Wegen $uw = uv^0w \in L$ ist $|uw|$ eine Primzahl.

   Wir setzen $x = uv^{|uw|}w$. Es gilt $x \in L$, d. h. $|x|$ ist eine Primzahl.

   Es folgt aber auch $|x| = |uw| + |uw||v| = |uw|(1 + |v|)$, d. h. $|x|$ ist wegen $|v| \neq 0$ keine Primzahl. Widerspruch!

Tutoraufgabe 2

Seien A, B und X Sprachen über Σ mit $\epsilon \notin A$. Beweisen oder widerlegen Sie die Umkehrung von Arden's Lemma, d.h. $X = A^*B \Longrightarrow X = AX \cup B$.

Lösungsvorschlag

Wir beweisen die Umkehrung von Arden's Lemma mit Hilfe von bereits bewiesenen oder elementaren Eigenschaften von Mengenoperationen. Sei $X = A^*B$.

$$\begin{aligned} AX \cup B &= A\,(A^*B) \cup B \\ &= (AA^*)\,B \cup \{\epsilon\}B \\ &= (AA^* \cup \{\epsilon\})\,B \\ &= A^*B \\ &= X \end{aligned}$$

Tutoraufgabe 3

Wir betrachten den DFA $A = (Q, \Sigma, \delta, q_0, \{q_1, q_2\})$, dessen Zustandsübergangsfunktion δ gegeben ist durch

q_i	$\delta(q_i, 0)$	$\delta(q_i, 1)$
q_0	q_1	q_2
q_1	q_0	q_1
q_2	q_1	q_0

Stellen Sie zur Berechnung der Sprachen $L_i = \{w \in \Sigma^* \mid \hat{\delta}(q_i, w) \in \{q_1, q_2\}\}$ ein Gleichungssystem mit entsprechenden Unbekannten X_i für die Sprachen L_i auf und berechnen Sie mit Hilfe von Ardens Lemma alle X_i als reguläre Ausdrücke.

Lösungsvorschlag

Aufstellen des Gleichungssystems:

$$\begin{aligned} X_0 &\equiv 0X_1 \mid 1X_2 && (1)\\ X_1 &\equiv 0X_0 \mid 1X_1 \mid \boldsymbol{\epsilon} && (2)\\ X_2 &\equiv 0X_1 \mid 1X_0 \mid \boldsymbol{\epsilon} && (3) \end{aligned}$$

Einsetzen von Gleichung (1) in (2, 3):

$$\begin{aligned} X_1 &\equiv 0(0X_1 \mid 1X_2) \mid 1X_1 \mid \boldsymbol{\epsilon}\\ &\equiv (00 \mid 1)X_1 \mid 01X_2 \mid \boldsymbol{\epsilon} && (4)\\ X_2 &\equiv 0X_1 \mid 1(0X_1 \mid 1X_2) \mid \boldsymbol{\epsilon}\\ &\equiv (0 \mid 10)X_1 \mid 11X_2 \mid \boldsymbol{\epsilon} && (5) \end{aligned}$$

Auflösen von Gleichung (4) nach X_1 (Ardens Lemma):

$$X_1 \equiv (00 \mid 1)^*(01X_2 \mid \boldsymbol{\epsilon}) \quad (6)$$

Einsetzen in Gleichung (5):

$$\begin{aligned} X_2 &\equiv \underbrace{(0 \mid 10)(00 \mid 1)^*}_{\alpha}(01X_2 \mid \boldsymbol{\epsilon}) \mid 11X_2 \mid \boldsymbol{\epsilon}\\ &\equiv (\alpha 01 \mid 11)X_2 \mid \alpha \mid \boldsymbol{\epsilon} \end{aligned}$$

Auflösen nach X_2 (Ardens Lemma):

$$X_2 \equiv (\alpha 01 \mid 11)^*(\alpha \mid \boldsymbol{\epsilon})$$

Durch Einsetzen in die Gleichungen (6) und (1) erhalten wir schließlich X_1 und X_0:

$$\begin{aligned} X_1 &\equiv (00 \mid 1)^*(01(\alpha 01 \mid 11)^*(\alpha \mid \boldsymbol{\epsilon}) \mid \boldsymbol{\epsilon})\\ X_0 &\equiv 0(00 \mid 1)^*(01(\alpha 01 \mid 11)^*(\alpha \mid \boldsymbol{\epsilon}) \mid \boldsymbol{\epsilon}) \mid 1(\alpha 01 \mid 11)^*(\alpha \mid \boldsymbol{\epsilon}) \end{aligned}$$

Strategische Hinweise: Um bei größer werdenden Ausdrücken nicht den Überblick zu verlieren, empfiehlt es sich, Abkürzungen für variablenfreie Teilausdrücke einzuführen, wie oben das α. Das verhindert auch viele Abschreibfehler. Natürlich ist α keine neue Variable, sondern nur eine Abkürzung für einen festen Ausdruck.
Gleichungen der Form $X_i \equiv R$, wobei X_i nicht in R vorkommt, setzt man am besten sofort in die restlichen Gleichungen ein, womit man sofort eine Variable (oben: X_0) aus dem System eliminiert hat.