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

Sommersemester 2011
Lösungsblatt 2
30. Mai 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)

1. Beweisen oder widerlegen Sie für Sprachen $A, B \subseteq \Sigma^*$:

$$(AB)^* = (AB \cup B)^* \qquad\qquad (A^*B^*)^* = (A \cup B)^*$$

Gilt eine Gleichung nicht, so geben Sie möglichst einfache Eigenschaften für A und B an, so dass alle Fälle erfasst sind, in denen diese Gleichung gilt.

2. Beweisen oder widerlegen Sie für reguläre Ausdrücke α und β:

$$(\alpha\beta|\alpha)^*\alpha \equiv \alpha(\beta\alpha|\alpha)^* \qquad\qquad (\alpha\beta|\alpha)^* \equiv (\alpha\beta)^*|\alpha^*$$

Lösungsvorschlag

1.
   - Die Gleichung gilt nicht, denn z.B. mit $A = \{a\}$ und $B = \{b\}$ ist b ein Wort der Sprache auf der rechten Seite, nicht aber in der Sprache auf der linken Seite enthalten. Die Gleichung gilt aber, falls $B = \emptyset$ oder $B = \{\epsilon\}$ oder $\epsilon \in A$.
   - Wir zeigen die Aussage, indem wir die Teilmengenbeziehung für beide Richtungen zeigen. Beachten Sie, dass Konkatenation und Stern monoton sind, d.h. es gilt $A \subseteq C \Rightarrow AB \subseteq CB$, $B \subseteq C \Rightarrow AB \subseteq AC$ und $A \subseteq B \Rightarrow A^* \subseteq B^*$ für alle Sprachen A, B, C. Damit gilt nun:

$$\begin{aligned}
(A^*B^*)^* &\subseteq ((A \cup B)^*B^*)^* && \text{da } A \subseteq (A \cup B)\\
&\subseteq ((A \cup B)^*(A \cup B)^*)^* && \text{da } B \subseteq (A \cup B)\\
&= ((A \cup B)^*)^* && \text{da } C^* = C^*C^* \text{ für beliebige } C\\
&= (A \cup B)^* && \text{da } C^* = (C^*)^* \text{ für beliebige } C
\end{aligned}$$

   und umgekehrt:

$$\begin{aligned}
(A \cup B)^* &\subseteq ((A \cup B) \cup (A^*B^*))^* && \text{da } (A \cup B) \subseteq (A \cup B) \cup C \text{ für beliebige } C\\
&= (A^*B^*)^* && \text{da } A \cup B \subseteq A^*B^*
\end{aligned}$$

2.
   - Die Ausdrücke sind äquivalent:

$$\begin{aligned}
(\alpha\beta|\alpha)^*\alpha &\equiv (\alpha\beta|\alpha\epsilon)^*\alpha && \text{mit Lemma 2.24(3)}\\
&\equiv (\alpha(\beta|\epsilon))^*\alpha && \text{mit Distributivität}\\
&\equiv \alpha((\beta|\epsilon)\alpha)^* && \text{mit TA3.1}\\
&\equiv \alpha(\beta\alpha|\alpha)^* && \text{mit Distributivität}
\end{aligned}$$

- Mit $\alpha = a$ und $\beta = b$ haben wir $aab \in L((\alpha\beta|\alpha)^*)$ aber $aab \notin L((\alpha\beta)^*|\alpha^*)$ und damit sind die beiden Ausdrücke nicht äquivalent.

Hausaufgabe 2 (7 Punkte)

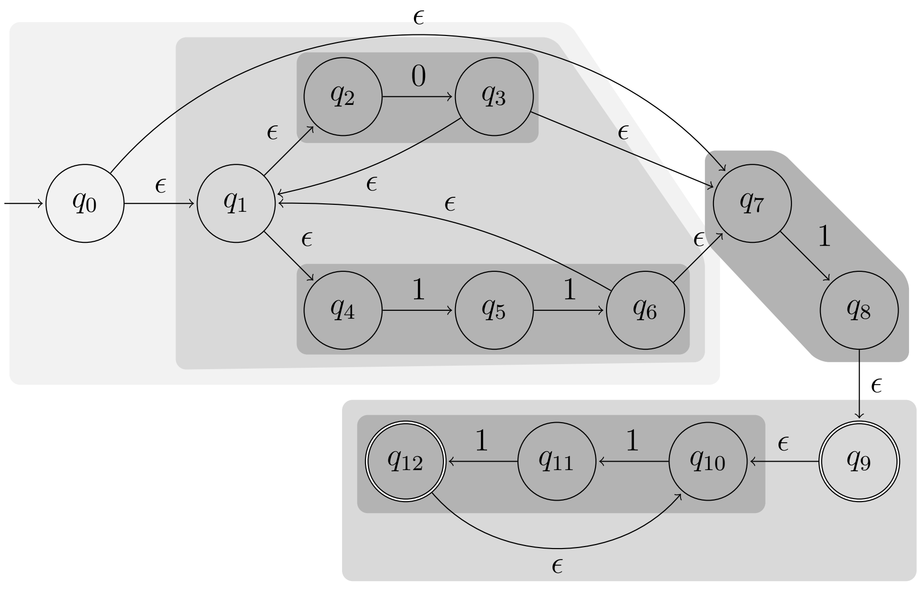
Wir betrachten das Alphabet $\Sigma = \{0, 1\}$.

1. Finden Sie einen regulären Ausdruck α, der folgende Sprache beschreibt: Jedes Wort w in $L(\alpha)$ endet in einer Sequenz ungerader Länge von Einsen und alle anderen Sequenzen von Einsen in w sind von gerader Länge. Wandeln Sie diesen regulären Ausdruck mit dem Standardverfahren aus der Vorlesung in einen ϵ-NFA M um, so dass $L(M) = L(\alpha)$, und geben sie dabei sowohl den Graphen als auch die Übergangsrelation δ an.

2. Geben Sie einen NFA M_1 an, der die Menge aller Wörter beschreibt, die entweder aus einer ungeraden Anzahl von Nullen bestehen oder aber mit einer Null anfangen, die von beliebig vielen Teilwörtern 01 gefolgt wird. Nutzen Sie den Nichtdeterminismus aus. Wandeln Sie diesen NFA in einen DFA M_2 um, so dass $L(M_2) = L(M_1)$. Geben Sie für beide Automaten sowohl den Graphen als auch die Übergangsrelation an.

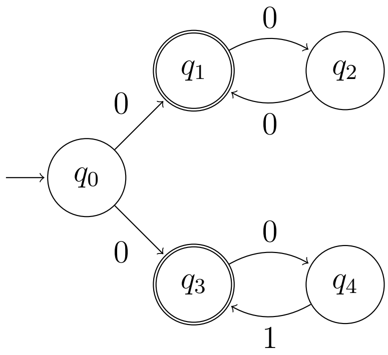
Lösungsvorschlag

1. Der gesuchte reguläre Ausdruck ist $(0|11)^*1(11)^*$. Entsprechend dem Standardverfahren aus der Vorlesung ergibt sich folgender ϵ-NFA:



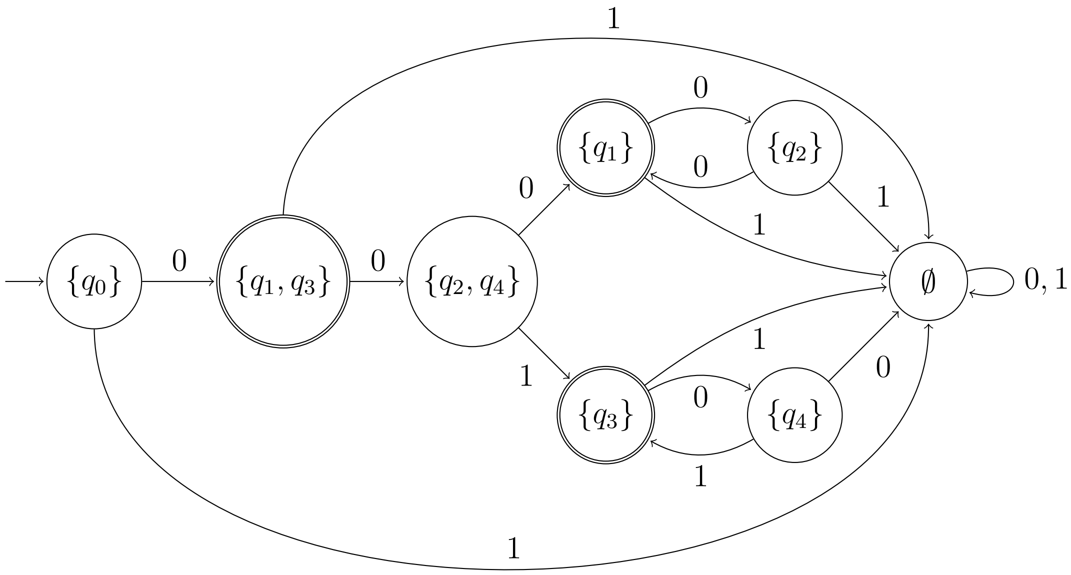
$\delta(q_0, \epsilon) = \{q_1, q_7\}$	$\delta(q_4, 1) = \{q_5\}$	$\delta(q_7, 1) = \{q_8\}$	$\delta(q_{10}, 1) = \{q_{11}\}$
$\delta(q_1, \epsilon) = \{q_2, q_4\}$	$\delta(q_5, 1) = \{q_6\}$	$\delta(q_8, \epsilon) = \{q_9\}$	$\delta(q_{11}, 1) = \{q_{12}\}$
$\delta(q_2, 0) = \{q_3\}$	$\delta(q_6, \epsilon) = \{q_1, q_7\}$	$\delta(q_9, \epsilon) = \{q_{10}\}$	$\delta(q_{12}, \epsilon) = \{q_{10}\}$
$\delta(q_3, \epsilon) = \{q_1, q_7\}$			

2. Ein möglicher NFA M_1 ist der folgende:



$$\delta(q_0, 0) = \{q_1, q_3\}$$
$$\delta(q_1, 0) = \{q_2\}$$
$$\delta(q_2, 0) = \{q_1\}$$
$$\delta(q_3, 1) = \{q_4\}$$
$$\delta(q_4, 0) = \{q_3\}$$

Der daraus durch Determinierung erhaltene DFA M_2 ist wie folgt:



Wir listen nur die von $\{q_0\}$ erreichbaren Zustände auf:

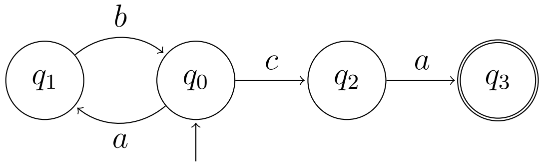
q	$\delta(q,0)$	$\delta(q,1)$
$\{q_0\}$	$\{q_1, q_3\}$	$\emptyset$
$\{q_1, q_3\}$	$\{q_2, q_4\}$	$\emptyset$
$\{q_2, q_4\}$	$\{q_1\}$	$\{q_3\}$
$\{q_1\}$	$\{q_2\}$	$\emptyset$
$\{q_2\}$	$\{q_1\}$	$\emptyset$
$\{q_3\}$	$\{q_4\}$	$\emptyset$
$\{q_4\}$	$\emptyset$	$\{q_3\}$
$\emptyset$	$\emptyset$	$\emptyset$

Hausaufgabe 3 (4 Punkte)

Ein *Fragment* von $w \in \Sigma$ ist ein Wort, dass aus w durch Entfernen beliebiger Buchstaben entstanden ist. Zum Beispiel sind ϵ, a, b, aa, ab, ba, bb, aaa, aab, aba, baa. bab, bba, $baaa$, $baab$, $baba$ und $baaba$ alle Fragmente von $baaba$.

Für eine Sprache L bezeichen wir mit Fragment(L) die Sprache aller Fragmente aller Wörter in L.

1. Sei L die von dem folgenden Automaten akzeptierte Sprache.



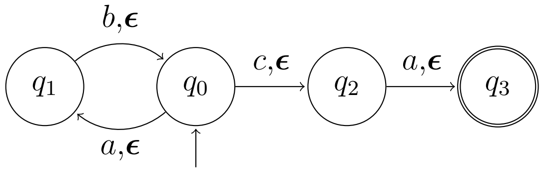
   Konstruieren Sie daraus einen endlichen Automaten, der Fragment(L) akzeptiert.

2. Zeigen Sie, dass für alle regulären Sprachen L auch Fragment(L) regulär ist. Formulieren Sie dazu einen Algorithmus, der zu einem NFA M einen NFA M' mit $L(M') = \text{Fragment}(L(M))$ konstruiert, und begründen Sie seine Korrektheit.

3. Sei $\Sigma = \{a, b\}$. Die Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$ ist nicht regulär. Ist Fragment(L) dennoch regulär? Begründen Sie.

Lösungsvorschlag

1. Ein $\boldsymbol{\epsilon}$-NFA, der Fragment(L) akzeptiert ist der folgende:



2. Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein NFA. Zunächst konstruieren wir einen $\boldsymbol{\epsilon}$-NFA $M'_{\boldsymbol{\epsilon}}$ mit $M'_{\boldsymbol{\epsilon}} = (Q, \Sigma \cup \{\boldsymbol{\epsilon}\}, \delta'_{\boldsymbol{\epsilon}}, q_0, F)$ und $\delta'_{\boldsymbol{\epsilon}} = \delta \cup \{(q, \boldsymbol{\epsilon}, q') \mid \exists a \in \Sigma.\, (q, a, q') \in \delta\}$.

   Man sieht leicht ein, dass dieser Automat Fragment($L(M)$) akzeptiert: $M'_{\boldsymbol{\epsilon}}$ enthält zu jeder Kante, die mit einem $a \in \Sigma$ beschriftet ist, eine parallele Kante, die mit $\boldsymbol{\epsilon}$ beschriftet ist. Daher ist ein akzeptierender Lauf durch den Automaten für ein Wort w auch gleichzeitig ein akzeptierender Lauf für jedes Fragment von w. Dazu wird für jeden ausgelassenen Buchstaben die $\boldsymbol{\epsilon}$-Kante (statt eine der Kanten aus δ) gewählt

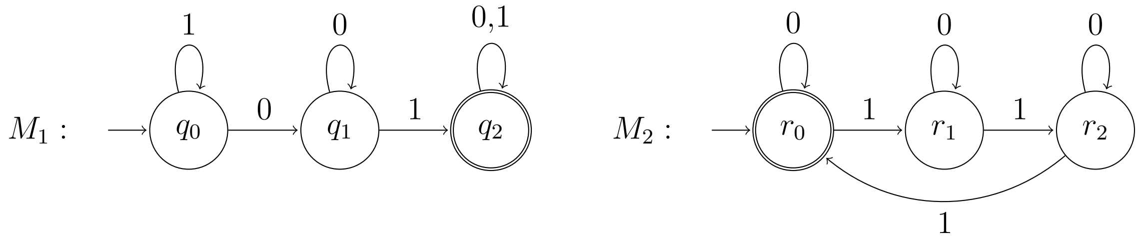
   Wie in der Vorlesung erhalten wir nun aus M'_ϵ den gesuchten NFA M'.

3. Es gilt Fragment(L) $= L(a^*b^*)$. Damit ist Fragment(L) regulär.

Hausaufgabe 4 (4 Punkte)

1. Gegeben seien zwei DFAs M_1 und M_2 über dem gleichen Eingabealphabet. Geben Sie ein direktes Verfahren an (ohne Umweg über reguläre Ausdrücke oder NFAs), um einen DFA M mit $L(M) = L(M_1) \cup L(M_2)$ zu konstruieren.

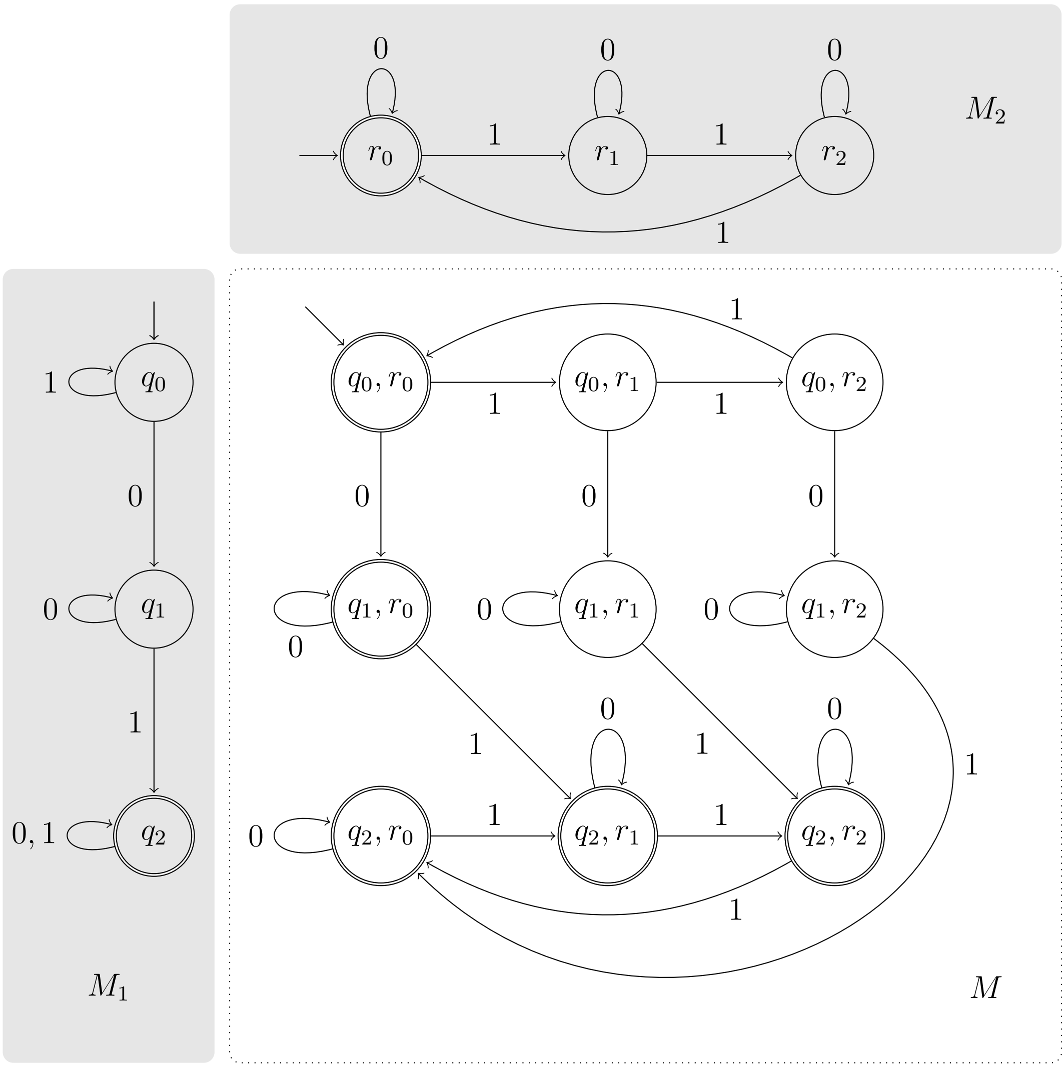
2. Wenden Sie dieses Verfahren auf die beiden folgenden Automaten an.



Lösungsvorschlag

1. Seien $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ und $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$. Dann gilt für den Automaten
$$M = (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times Q_2 \cup Q_1 \times F_2)$$
mit
$$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a))$$
die geforderte Bedingung $L(M) = L(M_1) \cup L(M_2)$, denn:
$$\begin{aligned}
& w \in L(M) \\
\text{gdw. } & \hat{\delta}((q_1, q_2), w) \in F_1 \times Q_2 \cup Q_1 \times F_2 \\
\text{gdw. } & \hat{\delta}((q_1, q_2), w) \in F_1 \times Q_2 \;\text{ oder }\; \hat{\delta}((q_1, q_2), w) \in Q_1 \times F_2 \\
\text{gdw. } & (\hat{\delta_1}(q_1, w), \hat{\delta_2}(q_2, w)) \in F_1 \times Q_2 \;\text{ oder }\; (\hat{\delta_1}(q_1, w), \hat{\delta_2}(q_2, w)) \in Q_1 \times F_2 \\
\text{gdw. } & \hat{\delta_1}(q_1, w) \in F_1 \;\text{ oder }\; \hat{\delta_2}(q_2, w) \in F_2 \\
\text{gdw. } & w \in L(M_1) \;\text{ oder }\; w \in L(M_2) \\
\text{gdw. } & w \in L(M_1) \cup L(M_2)
\end{aligned}$$
wobei $\hat{\delta}((q_1, q_2), w) = (\hat{\delta_1}(q_1, w), \hat{\delta_2}(q_2, w))$ induktiv über die Länge von w gezeigt werden kann.

2. Wir konstruieren den Vereinigungsautomaten aus M_1 und M_2 durch Ausnutzen der Tabellennotation:



Quiz 1

Beantworten Sie kurz die folgenden Fragen:

1. Geben sie eine Sprache A über dem Alphabet $\Sigma = \{0, 1\}$ an, so dass $01 \in A$ und $A^*A = A$.
2. Wann genau ist die von einem endlichen Automaten erzeugte Sprache endlich?
3. Finden Sie einen regulären Ausdruck über dem Alphabet $\Sigma = \{0, 1\}$, der die Menge aller Wörter beschreibt, die mit 00 beginnen und in denen 1 genau dreimal vorkommt.

Lösungsvorschlag

1. Eine Lösung ist $A = L((01)^*)$.
2. Genau dann, wenn es keinen erreichbaren und produktiven (d.h. ein Endzustand ist erreichbar) Zustand gibt, der auf einem Kreis liegt.
3. Eine Lösung ist $000^*10^*10^*10^*$.

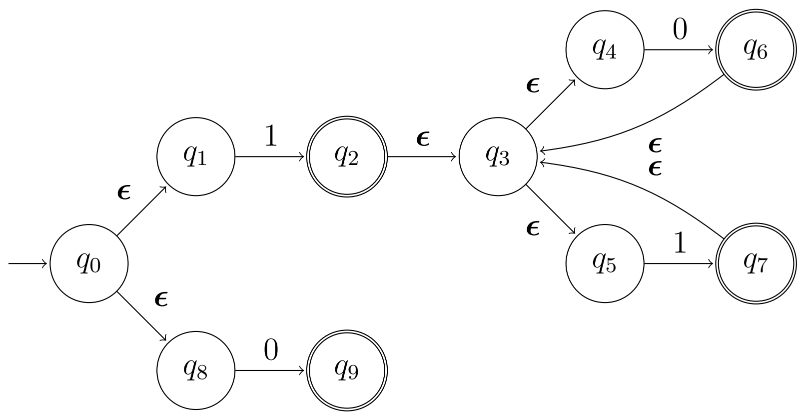
Tutoraufgabe 1

Wir betrachten den regulären Ausdruck $\alpha = (1(0|1)^*)|0$.

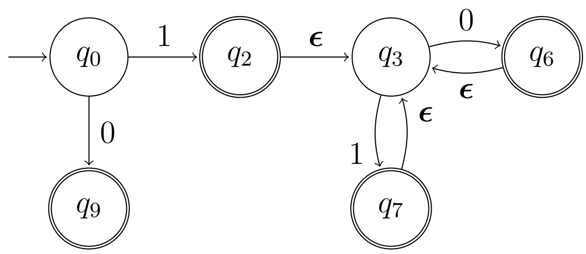
1. Konstruieren Sie für α mit dem Standardverfahren aus der Vorlesung einen ϵ-NFA A, so dass $L(\alpha) = L(A)$ gilt.
2. Wandeln Sie den erhaltenen Automaten in einen äquivalenten NFA ohne ϵ-Übergänge um.
3. Konstruieren Sie durch Anwendung des Potenzmengenverfahrens einen DFA, der die Sprache des Ausdrucks α akzeptiert.

Lösungsvorschlag

1. Die Konstruktion ist in Satz 2.17 der Vorlesung beschrieben mit dem Hilfszeichen $\boldsymbol{\epsilon}$. Man erhält dann den folgenden ϵ-NFA $A = (Q, \Sigma \cup \{\boldsymbol{\epsilon}\}, q_0, F)$:



2. Wir vereinfachen zunächst den obigen ϵ-NFA durch Beseitigung „offensichtlich unnötiger“ Zustände. Das sind Zustände, die keine Endzustände sind, und die genau eine eingehende und eine ausgehende Kante haben, von denen mindestens eine mit $\boldsymbol{\epsilon}$ beschriftet ist. Diese Zustände können offensichtlich entfernt werden und die Übergänge zusammengefasst. Im konkreten Beispiel entfernen wir also die Zustände q_1, q_4, q_5 und q_8 und erhalten folgenden vereinfachten ϵ-NFA:



Wir konstruieren nun den zugehörigen NFA $A' = (Q, \Sigma, \delta', q_0, F')$ ohne $\boldsymbol{\epsilon}$-Übergänge. Dieser hat dieselbe Zustandsmenge aber eine geänderte Übergangsfunktion δ':

$$\delta'(q,a) = \hat{\delta}(q, \boldsymbol{\epsilon}^* a \boldsymbol{\epsilon}^*) = \bigcup_{x \in \boldsymbol{\epsilon}^* a \boldsymbol{\epsilon}^*} \hat{\delta}(q,x)\,.$$

Es ist sinnvoll, zunächst die sogenannten $\boldsymbol{\epsilon}$-Hüllen

$$\hat{\delta}(q, \boldsymbol{\epsilon}^*) = \bigcup_{x \in \boldsymbol{\epsilon}^*} \hat{\delta}(q,x)$$

aller Zustände q zu berechnen. Wir erhalten

$$\begin{array}{lll}
\hat{\delta}(q_0, \boldsymbol{\epsilon}^*) = \{q_0\}\,, & \hat{\delta}(q_2, \boldsymbol{\epsilon}^*) = \{q_2, q_3\}\,, & \hat{\delta}(q_3, \boldsymbol{\epsilon}^*) = \{q_3\}\,, \\
\hat{\delta}(q_6, \boldsymbol{\epsilon}^*) = \{q_3, q_6\}\,, & \hat{\delta}(q_7, \boldsymbol{\epsilon}^*) = \{q_3, q_7\}\,, & \hat{\delta}(q_9, \boldsymbol{\epsilon}^*) = \{q_9\}\,.
\end{array}$$

Die Übergänge δ' bei Eingabe von Buchstaben aus Σ ergeben sich aus der Formel

$$\delta'(q,a) = \hat{\delta}(\hat{\delta}(\hat{\delta}(q, \boldsymbol{\epsilon}^*), a), \boldsymbol{\epsilon}^*)$$

wie folgt für $a \in \{0,1\}$:

$$\begin{array}{rcl}
\delta'(q_0, 0) & = & \hat{\delta}(\hat{\delta}(\{q_0\}, 0), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_9\}, \boldsymbol{\epsilon}^*) = \{q_9\}\,, \\
\delta'(q_0, 1) & = & \hat{\delta}(\hat{\delta}(\{q_0\}, 1), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_2\}, \boldsymbol{\epsilon}^*) = \{q_2, q_3\}\,, \\
\delta'(q_2, 0) & = & \hat{\delta}(\hat{\delta}(\{q_2, q_3\}, 0), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_6\}, \boldsymbol{\epsilon}^*) = \{q_3, q_6\}\,, \\
 & \vdots &
\end{array}$$

Die übrigen Ergebnisse sind in der untenstehenden Tabelle angegeben, die die Übergangsrelation des neuen Automaten beschreibt. Da in unserem Beispiel offenbar das leere Wort ϵ nicht in L ist, d. h. $\epsilon \notin L$, kann die Menge der Endzustände unverändert übernommen werden. Damit ist $F' = \{q_2, q_6, q_7, q_9\}$.

q_i	$\delta'(q_i, 0)$	$\delta'(q_i, 1)$
q_0	$\{q_9\}$	$\{q_2, q_3\}$
q_2	$\{q_3, q_6\}$	$\{q_3, q_7\}$
q_3	$\{q_3, q_6\}$	$\{q_3, q_7\}$
q_6	$\{q_3, q_6\}$	$\{q_3, q_7\}$
q_7	$\{q_3, q_6\}$	$\{q_3, q_7\}$
q_9	$\emptyset$	$\emptyset$

3. Das Potenzmengenverfahren liefert einen zu A' äquivalenten DFA $A'' = (\mathcal{P}(Q), \Sigma, \delta'', \{q_0\}, F'')$. Wir listen aber die Übergangsfunktion nur für die von q_0 aus erreichbaren Zustände auf. Im Folgenden schreiben wir abkürzend $q_iq_jq_k$ für $\{q_i, q_j, q_k\}$.

s_i	$\delta''(s_i, 0)$	$\delta''(s_i, 1)$
q_0	q_9	q_2q_3
q_2q_3	q_3q_6	q_3q_7
q_3q_6	q_3q_6	q_3q_7
q_3q_7	q_3q_6	q_3q_7
q_9	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

Alle weiteren Zustände und Übergänge sind von $\{q_0\}$ aus nicht erreichbar und daher für die Sprache von A'' irrelevant. Die Endzustandsmenge ergibt sich als $F'' = \{q_2q_3, q_3q_6, q_3q_7, q_9\}$.

Bemerkung: Es gibt allerdings einen äquivalenten DFA mit nur 4 Zuständen.

Tutoraufgabe 2

Das sogenannte *Shuffle-Produkt* spielt in der Theorie der nebenläufigen Systeme eine wichtige Rolle. Für zwei Sprachen L_1 und L_2 bezeichnet $L_1 \parallel L_2$ die Menge der Wörter, die man erhält, inde man zwei Wörter $v \in L_1$ und $u \in L_2$ beliebig miteinander verschränkt. Dabei können sich Teile aus v und w beliebig abwechseln, wobei die Reihenfolge der Zeichen aus v und w jedoch erhalten bleibt. Das kann man sich gut als das Ineinanderschieben zweier Kartenstapel veranschaulichen.
Formal definieren wir $L_1 \parallel L_2$ wie folgt:

$$L_1 \parallel L_2 = \{v_1 w_1 v 2_w 2 \cdots v_n w_n \,| v_1, \ldots, v_n, w_1, \ldots, w_n \in \Sigma^*, \\ v_1 v_2 \cdots v_n \in L_1 \text{ und } w_1 w_2 \cdots w_n \in L_2\}$$

1. Versuchen Sie, eine einfache Beschreibung von $L((01)^*) \parallel L((10)^*)$ zu finden.
2. Begründen Sie: Wenn L_1 und L_2 regulär sind, dann ist auch $L_1 \| L_2$ regulär. *Hinweis: Konstruieren Sie einen NFA für $L_1 \parallel L_2$.*
3. Führen Sie die Konstruktion konkret für die Sprachen $L((01)^*)$ und $L((10)^*)$ durch und geben Sie einen regulären Ausdruck an, der die Sprache beschreibt.

Lösungsvorschlag

1. Es ist auf Anhieb nicht klar, wie die Sprache $L((01)^*) \parallel L((10)^*)$ aussieht. Die erste Hypothese, dass es sich um $\{0,1\}^*$ handeln könnte ist falsch, denn in beiden Teilsprachen ist die Anzahl der Einsen und der Nullen stets gleich, also muss das auch für das Shuffle-Produkt gelten. Andererseits ist z.B. das Wort 000111 nicht möglich, da drei Nullen hintereinander nicht vorkommen können. Man sieht also, das eine kompakte Beschreibung des Shuffle-Produkts unter Umständen schwierig sein kann. Die Automatenkonstruktion im nächsten Aufgabenteil hilft uns aber weiter.

2. Aus zwei NFAs $N_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ und $N_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ für L_1 bzw. L_2 konstruieren wir einen NFA, der das Shuffle-Produkt akzeptiert. Wir verwenden einen Produktautomaten, dessen Zustände genau Paare von Zuständen der ursprünglichen Automaten sind:

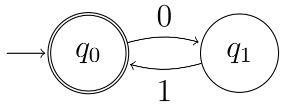
$$N_\parallel = (Q_1 \times Q_2, \Sigma, \delta_\parallel, (q_{01}, q_{02}), F_1 \times F_2)$$

Ein Schritt im neuen Automaten besteht darin, dass einer der beiden "Teilautomaten" einen Schritt macht:

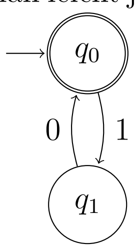
$$\delta_\parallel((q_1, q_2), a) = \{(q', q_2) \mid q' \in \delta_1(q_1, a)\} \cup \{(q_1, q') \mid q' \in \delta_2(q_2, a)\}$$

3. Zu den regulären Ausdrücken $(01)^*$ und $(10)^*$ kann man leicht je einen NFA angeben:

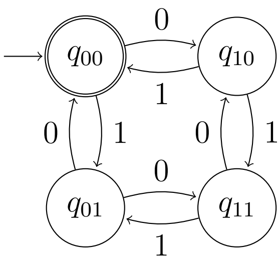
N_1:



N_2:



Der Automat $N_\parallel$ für das Shuffle-Produkt hat dementsprechend vier Zustände:



Hier kann man jetzt auch etwas leichter ablesen, wie die Produktsprache aussieht: Sie kann z.B. durch den regulären Ausdruck $(01|10)^*$ beschrieben werden.

Tutoraufgabe 3

Beweisen oder widerlegen Sie für reguläre Ausdrücke α und β:

1. $(\alpha\beta)^*\alpha \equiv \alpha(\beta\alpha)^*$

2. $\alpha|\alpha^* \equiv \alpha^*$

Falls möglich, verwenden Sie für einen Beweis die Regeln zum Rechnen mit regulären Ausdrücken aus der Vorlesung.

Lösungsvorschlag

1. Diese Gleichung lässt sich mit den Rechenregeln aus der Vorlesung nicht zeigen. Dennoch sind diese beiden Ausdrücke äquivalent. Dafür zeigen wir zunächst, für zwei beliebige Sprachen A und B die Gleichung

$$(AB)^n A = A(BA)^n$$

per Induktion:

- $n = 0$: Es gilt $(AB)^0 A = A = A(BA)^0$.
- $n \to n+1$: Es gilt

$$\begin{aligned}(AB)^{n+1} A &= AB(AB)^n A \\ &= ABA(BA)^n && \text{mit Ind.hyp.} \\ &= A(BA)^{n+1}\end{aligned}$$

Damit sind die regulären Ausdrücke äquivalent, denn es folgt:

$$\begin{aligned}L((\alpha\beta)^*\alpha) &= (L(\alpha)L(\beta))^* L(\alpha) \\ &= \bigcup_{n\geq 0} (L(\alpha)L(\beta))^n L(\alpha) \\ &= \bigcup_{n\geq 0} L(\alpha)(L(\beta)L(\alpha))^n \\ &= L(\alpha)(L(\beta)L(\alpha))^* \\ &= L(\alpha(\beta\alpha)^*)\end{aligned}$$

2. Mit den Regeln aus der Vorlesung ergibt sich folgende Gleichungskette:

$$\begin{aligned}\alpha|\alpha^* &\equiv \alpha|(\epsilon|\alpha\alpha^*) && \text{mit Lemma 2.26(1)} \\ &\equiv (\alpha|\epsilon)|\alpha\alpha^* && \text{mit Assoziativität} \\ &\equiv (\epsilon|\alpha)|\alpha\alpha^* && \text{mit Kommutativität} \\ &\equiv \epsilon|(\alpha|\alpha\alpha^*) && \text{mit Assoziativität} \\ &\equiv \epsilon|(\alpha\epsilon|\alpha\alpha^*) && \text{mit Lemma 2.24(3)} \\ &\equiv \epsilon|\alpha(\epsilon|\alpha^*) && \text{mit Distributivität} \\ &\equiv \epsilon|\alpha\alpha^* && \text{mit Beispiel 2.27} \\ &\equiv \alpha^* && \text{mit Lemma 2.26(1)}\end{aligned}$$