

---

## Einführung in die Theoretische Informatik

---

### Hausaufgabe 1 (5 Punkte)

Sei  $\Sigma = \{a, b\}$ . Geben Sie Sprachen  $L_1$  bzw.  $L_2$  an, so dass die folgenden Eigenschaften gelten:

1.  $ab \in L_1$  und  $L_1^* L_1 = L_1$ .
2.  $L_2$  ist endlich und  $L_2^* = L_2$

Zeigen Sie für alle Sprachen  $A \subseteq \Sigma^*$ :

3.  $(A^*)^* = A^*$ .

### Lösungsvorschlag

1. Z.B.  $L_1 = \{ab\}^*$  oder  $L_1 = \Sigma^*$ .
2.  $L_2 = \{\epsilon\}$  ist die einzige Lösung. Gäbe es ein nichtleeres Wort  $w \in L_2$ , dann wäre  $\{w^n \mid n \in \mathbb{N}\} \subseteq L_2^*$  und damit  $L_2 = L_2^*$  unendlich. Das leere Wort ist immer in  $L_2^*$ .
3. Für die Richtung  $(A^*)^* \subseteq A^*$  genügt es zu zeigen, dass für alle  $n \in \mathbb{N}$ ,  $(A^*)^n \subseteq A^*$  gilt. Wir induzieren über  $n$ . Im Fall  $n = 0$  gilt offenbar  $(A^*)^0 = \{\epsilon\} \subseteq A^*$ . Im Induktionsschritt gilt  $(A^*)^{n+1} = A^*(A^*)^n \subseteq A^*A^* = A^*$ , wobei die mittlere Inklusion aus der Induktionshypothese folgt, und die hintere Gleichung aus TA1 von Blatt 1. Die Gegenrichtung ist offensichtlich, da  $A^* = (A^*)^1 \subseteq \bigcup_{n \in \mathbb{N}} (A^*)^n = (A^*)^*$ .

### Hausaufgabe 2 (5 Punkte)

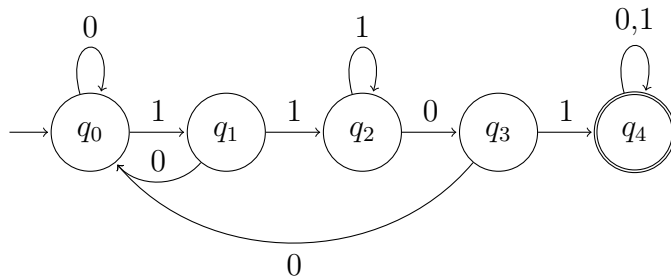
Geben Sie einen deterministischen endlichen Automaten (als Graph und Übergangsrelation) an, der über dem Alphabet  $\Sigma = \{0, 1\}$  die folgende Sprache akzeptiert.

1. Die Menge aller Wörter, die das Teilwort 1101 enthalten.
2. Die Menge aller Wörter, bei denen die Anzahl der Nullen durch 4 teilbar ist.

## Lösungsvorschlag

1. Die Menge aller Wörter, die das Teilwort 1101 enthalten, werden akzeptiert vom Automaten  $A_1 = (\{q_0, q_1, q_2, q_3, q_4\}, \Sigma, \delta, q_0, \{q_4\})$ .

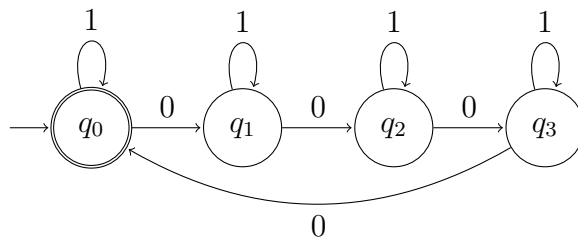
Graph und Übergangsrelation:



$q_i$	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$q_0$	$q_0$	$q_1$
$q_1$	$q_0$	$q_2$
$q_2$	$q_3$	$q_2$
$q_3$	$q_0$	$q_4$
$q_4$	$q_4$	$q_4$

2. Die Menge aller Wörter, bei denen die Anzahl der Nullen durch 4 teilbar ist, werden akzeptiert vom Automaten  $A_2 = (\{q_0, q_1, q_2, q_3\}, \Sigma, \delta, q_0, \{q_0\})$ .

Graph und Übergangsrelation:



$q_i$	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$q_0$	$q_1$	$q_0$
$q_1$	$q_2$	$q_1$
$q_2$	$q_3$	$q_2$
$q_3$	$q_0$	$q_3$

## Hausaufgabe 3 (5 Punkte)

Zeigen Sie

$$\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v) \quad \text{für alle } u, v \in \Sigma^*.$$

## Lösungsvorschlag

Wir induzieren über  $u$ :

Basis:  $u = \epsilon$ .  $\hat{\delta}(q, \epsilon v) = \hat{\delta}(q, v) = \hat{\delta}(\hat{\delta}(q, \epsilon), v)$

Schritt:  $u = aw$  mit  $a \in \Sigma$  und  $w \in \Sigma^*$ .

Wir haben die Induktionshypothese  $\hat{\delta}(q, wv) = \hat{\delta}(\hat{\delta}(q, w), v) \quad \forall q \in Q$ .

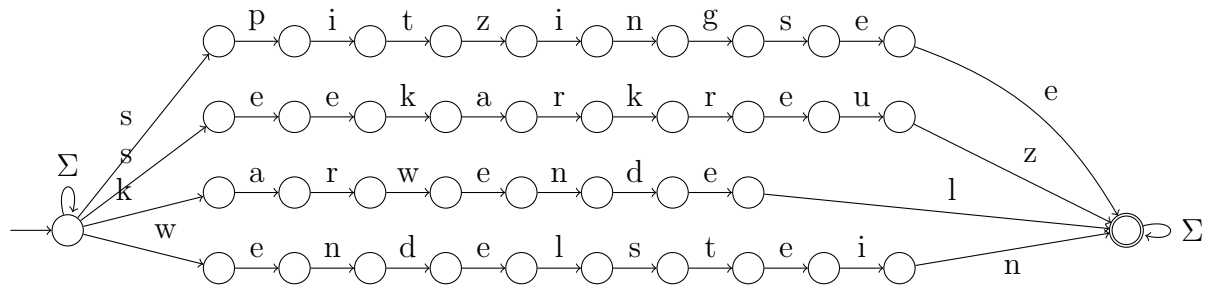
$$\hat{\delta}(q, awv) = \hat{\delta}(\delta(q, a), wv) \stackrel{\text{(IH)}}{=} \hat{\delta}(\hat{\delta}(\delta(q, a), w), v) = \hat{\delta}(\hat{\delta}(q, aw), v)$$

## Hausaufgabe 4 (5 Punkte)

Sei  $\Sigma = \{a, \dots, z\}$ . Geben Sie einen NFA an, der die Sprache der Wörter akzeptiert, die (mindestens) eines der folgenden Wörter als Teilwort enthalten: **spitzingsee**, **seekarkreuz**, **karwendel**, **wendelstein**.

Machen Sie sinnvollen Gebrauch von Nichtdeterminismus, so dass Ihre Konstruktion möglichst einfach ist, und sich leicht auf andere Suchwörter verallgemeinern lässt.

### Lösungsvorschlag

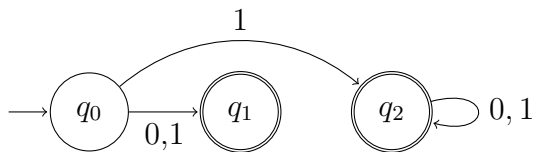


Hier ist der Nichtdeterminismus besonders nützlich, da wir uns nicht um mögliche Überlappungen in den Suchwörtern kümmern müssen. Die direkte Konstruktion eines DFA hingegen ist wegen dieser Überlappungen deutlich schwieriger.

Mit dem Potenzmengenverfahren kann man allerdings auf automatisierte Weise einen DFA erhalten, woraus sich auch direkt ein effiziente Implementierung eines Matchers ergibt.

## Vorbereitung 1

Wir betrachten einen nichtdeterministischen Automaten  $A = (Q, \Sigma, \delta, q_0, F)$ , der durch die folgende Grafik gegeben ist.



1. Geben Sie einen regulären Ausdruck  $R$  an, dessen Sprache von  $A$  akzeptiert wird.
2. Bestimmen Sie  $\bar{\delta}(\{q_0, q_1\}, 1)$  und  $\bar{\delta}(\{q_1\}, 1)$

## Lösungsvorschlag

1.  $L(A)$  ist die Vereinigung von  $\{0, 1\}$  und der Menge aller Wörter, die mit 1 beginnen und ansonsten beliebig fortgesetzt sind. Dabei wird die einzelne 1 vom zweiten Fall subsumiert.

Wir setzen  $R$  also aus den 3 Ausdrücken für  $\{0\}$ ,  $\{1\}$  und  $\{0, 1\}^*$  zusammen. Dies sind 0 bzw. 1 bzw.  $(0|1)^*$ . Wir erhalten z. B.

$$R = 0 \mid (1(0|1)^*).$$

2.  $\bar{\delta}(\{q_0, q_1\}, 1) = \{q_1, q_2\}$ , und  $\bar{\delta}(\{q_1\}, 1) = \emptyset$ .

## Vorbereitung 2

Für zwei Wörter  $v, w \in \Sigma^*$  definieren wir die Sprache  $v \parallel w \subseteq \Sigma^*$  wie folgt:

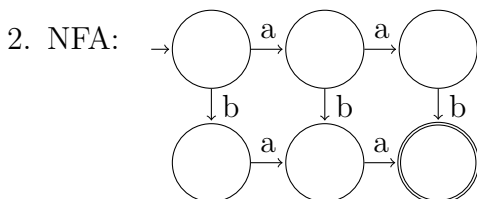
$$\begin{aligned} v \parallel \epsilon &= \{v\} \\ \epsilon \parallel w &= \{w\} \\ xv \parallel yw &= \{x\}(v \parallel yw) \cup \{y\}(xv \parallel w) \end{aligned}$$

Intuitiv beschreibt  $v \parallel w$  genau die Menge aller Verschränkungen von  $v$  und  $w$  in denen sich Fragmente 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.

1. Sei  $\Sigma = \{a, b, A, B\}$ . Bestimmen Sie  $ab \parallel AB$ .
2. Geben Sie einen NFA an, der die Sprache  $aa \parallel b$  akzeptiert.

## Lösungsvorschlag

1.  $ab \parallel AB = \{abAB, aAbB, aABb, AaBb, ABab, AabB\}$ .



## Tutoraufgabe 1

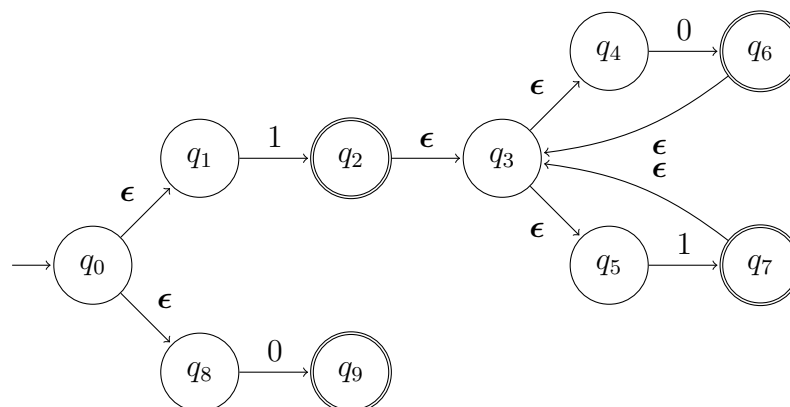
Wir betrachten den regulären Ausdruck  $R = (1(0|1)^*)|0$ .

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

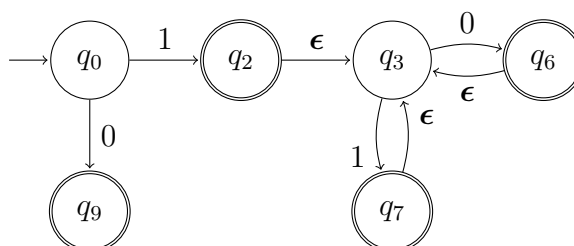
## Lösungsvorschlag

1. Der Zusammenbau eines Automaten nach dem Schema eines regulären Ausdrucks geschieht mit einem Hilfszeichen  $h$ , das die Übergänge zwischen den Komponentenautomaten regelt. Man erhält zunächst einen NFA über dem erweiterten Alphabet  $\Sigma \cup \{h\}$ . Die Sprache dieses Automaten projiziert man dann auf die gesuchte Sprache  $L(R)$  durch Entfernung des Hilfszeichens aus allen Wörtern. Diese Entfernung kann formal als Interpretation von  $h$  mit dem leeren Wort  $\epsilon$  dargestellt werden.

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



2. Wir vereinfachen zunächst den obigen  $\epsilon$ -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  $\epsilon$  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  $\epsilon$ -NFA:



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

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

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

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

aller Zustände  $q$  zu berechnen. Wir erhalten

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

Die Übergänge  $\delta'$  bei Eingabe von Buchstaben aus  $\Sigma$  ergeben sich aus der Formel

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

wie folgt für  $a \in \{0, 1\}$ :

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

Die übrigen Ergebnisse sind in der untenstehenden Tabelle angegeben, die die Übergangsrelation des neuen Automaten beschreibt. Da in unserem Beispiel offenbar das leere Wort  $\epsilon$  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  $q_i q_j q_k$  für  $\{q_i, q_j, q_k\}$ .

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

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

Sei  $v \parallel w$  für  $v, w \in \Sigma^*$  wie in Vorbereitungsaufgabe 2 definiert. Für zwei Sprachen  $L_1$  und  $L_2$  definieren wir nun:

$$L_1 \parallel L_2 = \bigcup_{\substack{u \in L_1 \\ v \in L_2}} (u \parallel v)$$

Man kann dieses sogenannte *Shuffle-Produkt* als Parallelkomposition von zwei unabhängigen Abläufen betrachten.

1. Versuchen Sie, eine einfache Beschreibung von  $L((01)^*) \parallel L((10)^*)$  zu finden.
2. Zeigen Sie: Wenn  $L_1$  und  $L_2$  regulär sind, dann ist auch  $L_1 \parallel L_2$  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

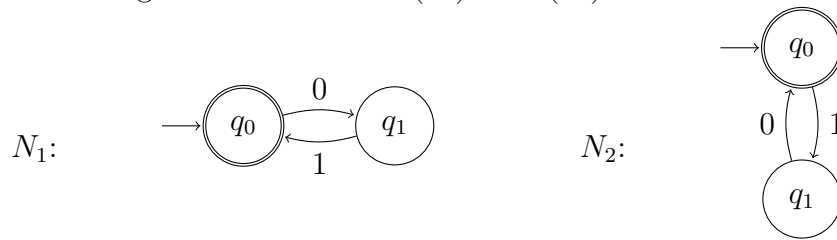
1. Es ist auf Anhieb nicht klar, wie die Sprache  $L((01)^*) \parallel L((10)^*)$  aussieht. Die erste Hypothese, dass es sich um  $\{0, 1\}^*$  handeln könnte ist falsch, denn in beiden Teilsprachen ist die Anzahl der Einsen und der Nullen stets gleich, also muss das auch für das Shuffle-Produkt gelten. Andererseits ist z.B. das Wort 000111 nicht möglich, da drei Nullen hintereinander nicht vorkommen können. Man sieht also, dass 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:

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

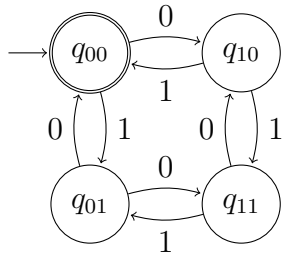
Ein Schritt im neuen Automaten besteht darin, dass einer der beiden "Teilautomaten" einen Schritt macht:

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

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



Der Automat  $N_{||}$  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.