
Einführung in die Theoretische Informatik

Hausaufgabe 1 (5 Punkte)

Wir betrachten wieder die Operation $quot$ (vgl. Tutoraufgabe 1, Blatt 3):

$$quot(a, L) = \{w \in \Sigma^* \mid aw \in L\}$$

Diese wird auch als Ableitung der Sprache L nach a bezeichnet.

1. Finden Sie geeignete rechte Seiten für die folgenden Ableitungsregeln:

- $quot(a, \{\epsilon\}) = \dots$
- $quot(a, \{a'\}) = \dots$ für $a' \in \Sigma$,
- $quot(a, L_1 \cup L_2) = \dots$

2. Zeigen Sie die Korrektheit der Regel für die Vereinigung

3. Stellen Sie auch die Regel für die Konkatenation auf, und unterscheiden Sie dabei die Fälle $\epsilon \in L$ und $\epsilon \notin L$. Begründen Sie informell, warum die Regel korrekt ist.

Lösungsvorschlag

1.
 - $quot(a, \{\epsilon\}) = \emptyset$
 - $quot(a, \{a'\}) = \begin{cases} \{\epsilon\} & \text{falls } a' = a \\ \emptyset & \text{sonst} \end{cases}$
 - $quot(a, L_1 \cup L_2) = quot(a, L_1) \cup quot(a, L_2)$
2. Sei $w \in \Sigma^*$ beliebig. Dann gilt.

$$\begin{aligned} w \in quot(a, L_1 \cup L_2) & \\ \iff aw \in L_1 \cup L_2 & \\ \iff aw \in L_1 \vee aw \in L_2 & \\ \iff w \in quot(a, L_1) \vee w \in quot(a, L_2) & \\ \iff w \in quot(a, L_1) \cup quot(a, L_2) & \end{aligned}$$

Damit sind beide Sprachen gleich.

3.

$$quot(a, L_1 L_2) = \begin{cases} quot(a, L_1) L_2 \cup quot(a, L_2) & \text{falls } \epsilon \in L_1 \\ quot(a, L_1) L_2 & \text{falls } \epsilon \notin L_1 \end{cases}$$

Begründung: Für $w \in quot(a, L_1 L_2)$, also $aw \in L_1 L_2$ gibt es zwei Fälle. Entweder gehören das a (und ein Teil von w) zu L_1 , dann ist $w \in quot(a, L_1) L_2$. Oder ganz aw liegt in L_2 . Dann ist $w \in quot(a, L_2)$. Dann muss aber $\epsilon \in L_1$ sein.

Hausaufgabe 2 (5 Punkte)

Stellen Sie die folgenden Sprachen jeweils durch einen regulären Ausdruck dar.

1. Die Sprache L_1 aller Wörter über $\Sigma = \{0, 1, /, \#\}$, in denen das Teilwort $\#/$ nicht vorkommt.
2. Die Sprache L_2 aller Wörter über $\{0, 1, /, \#\}$, der Form $\/#w\#/\$, so dass $w \in \Sigma^*$ kein Teilwort $\#/$ enthält.

Lösungsvorschlag

1. Folgen des Symbols $\#$ können am Ende stehen oder sie sind gefolgt von einem der Symbole $0, 1$. Wir schreiben abkürzend $\overline{\#}$ für den regulären Ausdruck $(0 \mid 1 \mid /)$. Dann gilt $L_1 = L(\alpha_1)$ mit

$$\alpha_1 = (\overline{\#} \mid \#^+(0 \mid 1))^* \#^* .$$

2. Wir bemerken, dass einerseits $\/#\#/\in L_2$ und andererseits $\/#\#/\notin L_2$ gelten. Nun folgt leicht, dass $L_2 = L(\alpha_2)$ mit

$$\alpha_2 = \/# \alpha_1 \#/\ .$$

Insbesondere der erste Ausdruck ist durch bloßes Hinschauen nicht leicht zu finden. Ein DFA für L_1 lässt sich aber sehr leicht konstruieren, und somit könnte man auch einen äquivalenten Ausdruck aus diesem DFA erhalten.

Hausaufgabe 3 (5 Punkte)

Konstruieren Sie durch Lösung entsprechender Gleichungen (siehe TA 3, Blatt 3) einen regulären Ausdruck für die Sprache R_{q_4, q_0}^5 (Def. siehe Kleene'scher Satz 2.17) des DFA $M = (\{q_0, \dots, q_4\}, \{0, 1\}, \delta, q_0, \{q_4\})$ mit folgender Übergangstabelle:

| q | $\delta(q, 0)$ | $\delta(q, 1)$ |
|-------|----------------|----------------|
| q_0 | q_1 | q_2 |
| q_1 | q_0 | q_4 |
| q_2 | q_2 | q_2 |
| q_3 | q_1 | q_0 |
| q_4 | q_3 | q_3 |

Lösungsvorschlag

Man erklärt q_4 zum Startzustand und q_0 zum Endzustand und sucht nun den Ausdruck X_4 für $L(q_4)$.

Man stellt zunächst das Gleichungssystem auf:

$$\begin{aligned} X_0 &\equiv 0X_1 \mid 1X_2 \mid \epsilon \\ X_1 &\equiv 0X_0 \mid 1X_4 \\ X_2 &\equiv 0X_2 \mid 1X_2 \\ X_3 &\equiv 0X_1 \mid 1X_0 \\ X_4 &\equiv 0X_3 \mid 1X_3 \end{aligned}$$

Der Zustand q_2 ist ein Fangzustand, von dem aus man nicht in einen Endzustand kommen kann, weshalb $X_2 \equiv \emptyset$ sein muss, wobei \emptyset der Ausdruck für die leere Sprache sein soll. Eine Sonderüberlegung braucht man dafür allerdings nicht, denn diese Tatsache ergibt sich bereits aus der sturen Anwendung von Ardens Lemma:

$$X_2 \equiv 0X_2 \mid 1X_2 \equiv \underbrace{(0 \mid 1)}_A X_2 \mid \underbrace{\emptyset}_B \implies X_2 \equiv (0 \mid 1)^* \emptyset \equiv \emptyset$$

Damit erhalten wir $X_0 \equiv 0X_1 \mid \epsilon$ und setzen in die Gleichung für X_3 ein:

$$\begin{aligned} X_3 &\equiv 0X_1 \mid 1\underbrace{(0X_1 \mid \epsilon)}_{X_0} \\ &\equiv 0X_1 \mid 10X_1 \mid 1 \\ &\equiv (0 \mid 10)X_1 \mid 1, \end{aligned}$$

womit sich X_4 wie folgt ergibt.

$$\begin{aligned} X_4 &\equiv (0 \mid 1)((0 \mid 10)X_1 \mid 1) \\ &\equiv (0 \mid 1)(0 \mid 10)X_1 \mid (01 \mid 11). \end{aligned}$$

Nun erhalten wir für X_1 die folgenden Gleichungen.

$$\begin{aligned} X_1 &\equiv 0(0X_1 \mid \epsilon) \mid 1(0 \mid 1)X_3 \\ &\equiv 0(0X_1 \mid \epsilon) \mid 1(0 \mid 1)((0 \mid 10)X_1 \mid 1) \\ &\equiv 00X_1 \mid 0 \mid 1(0 \mid 1)(0 \mid 10)X_1 \mid 1(0 \mid 1)1 \\ &\equiv (00 \mid 1(0 \mid 1)(0 \mid 10))X_1 \mid (0 \mid 1(0 \mid 1)1). \end{aligned}$$

Wir lösen mit Arden's Lemma.

$$X_1 \equiv [(00 \mid 1(0 \mid 1)(0 \mid 10))]^*(0 \mid 1(0 \mid 1)1)$$

und setzen ein

$$X_4 \equiv (0 \mid 1)(0 \mid 10)[[(00 \mid 1(0 \mid 1)(0 \mid 10))]^*(0 \mid 1(0 \mid 1)1)] \mid (01 \mid 11).$$

Hausaufgabe 4 (5 Punkte)

Sei $\Sigma = \{a, b\}$.

1. Sei $L_1 = \{(a^2b^n)^n \mid n \in \mathbb{N}\}$.
Zeigen Sie mithilfe des Pumping-Lemmas, dass L_1 nicht regulär ist.
2. Sei $L_2 = \{a^i b^j \mid i < j\}$.
Nutzen Sie die Abschlusseigenschaften der regulären Sprachen und die bekannte Tatsache, dass $L = \{a^i b^i \mid i \in \mathbb{N}\}$ nicht regulär ist, um zu zeigen, dass L_2 nicht regulär ist.
3. Geben Sie ein Beispiel dafür, dass die Schlussweise „ L_1 ist nicht regulär und L_2 ist nicht regulär, also ist $L_1 \cap L_2$ nicht regulär“ im Allgemeinen falsch ist.

Lösungsvorschlag

1. Angenommen L ist regulär, dann erhalten wir wie folgt einen Widerspruch.

Sei $n > 0$ eine Pumping-Lemma-Zahl für L .

Wir betrachten $z = (a^2b^n)^n$.

Dann gilt $|z| \geq n$ und offenbar $z \in L$.

Nach dem Pumping-Lemma folgt für gewisse $u, v, w \in \Sigma^*$ dann

$z = uvw$ mit $|uv| \leq n$, $|v| > 0$ und $uv^iw \in L$ für alle $i \geq 0$.

Wegen $|uv| \leq n$ und $|v| > 0$ gilt $uv = a$ oder $uv = a^2b^k$ mit $0 \leq k \leq n - 2$.

Falls v ein a enthält, dann enthält uv^0w am Wortanfang weniger als 2 a 's.

Da jedes nichtleere Wort in L am Wortanfang genau 2 a 's enthält ist die Bedingung $uv^0w \in L$ verletzt wenn $uv^0w \neq \epsilon$. Im Fall $uv^0w = \epsilon$ wären aber $u = w = \epsilon$ und somit $uv^2w = z^2 = (a^2b^n)^{2n} \in L$. Das ist unmöglich.

Falls aber v nur b 's enthält, dann kann $uv^2w \in L$ nicht gelten, weil nun das Anfangswort (Präfix) $a^2b^l a^2$ mehr b 's enthält als das Suffix a^2b^n am Ende des Wortes uv^2w .

Damit haben wir einen Widerspruch hergeleitet.

2. Wir nehmen an, dass L_2 regulär ist, um daraus einen Widerspruch zu erhalten.

Nach Satz 2.19 der Vorlesung muss dann auch $L_2^R = \{b^j a^i \mid i, j \in \mathbb{N}, i < j\}$ regulär sein. Aus einer einfachen Umbenennung von Zeichen ergibt sich, dass die Sprache $L' = \{a^i b^j \mid i, j \in \mathbb{N}, i > j\}$ ebenfalls regulär ist. Bekanntlich ist $L = \{a^i b^j \mid i, j \in \mathbb{N}\}$ regulär. Also sind auch $L \setminus L_2$ und $L \setminus L'$ regulär, mithin ist

$$(L \setminus L_2) \cap (L \setminus L') = \{a^i b^i \mid i \in \mathbb{N}\}$$

regulär. Widerspruch!

3. Wir haben gezeigt, dass L_2 und L' nicht regulär sind. Es gilt aber $L_2 \cap L' = \emptyset$ und \emptyset ist regulär.

Vorbereitung 1

Die Minimierung deterministischer Automaten beruht auf 2 Minimierungsprinzipien, und zwar

- Entfernen der vom Startzustand aus unerreichbaren Zustände eines Automaten A .
- Konstruktion des Quotienten A/\equiv_A eines Automaten A .

Der Quotientenautomat A/\equiv_A wird dabei in den 2 aufeinanderfolgenden Schritten

- a) Berechnung der äquivalenten Zustandspaare (Relation \equiv_A über Q), und
- b) „Kollabierung“ des Automaten durch Klassenbildung äquivalenter Zustände, gewonnen.

1. Zeigen Sie, dass bei jeder Anwendung eines der obigen Minimierungsprinzipien die akzeptierte Sprache unverändert bleibt.
2. Machen Sie sich klar, dass man stets isomorphe Automaten erhält, wenn man nacheinander mindestens beide Minimierungsprinzipien in beliebiger Reihenfolge anwendet. Die beiden Minimierungsprinzipien sind also in gewissem Sinne unabhängig voneinander.

Lösungsvorschlag

1. Minimierung durch Entfernen:

Man darf nur dann eine Teilmenge von Zuständen eines Automaten entfernen, wenn sichergestellt ist, dass die Übergangsfunktion δ weiterhin definiert ist. Das Entfernen von unerreichbaren Zuständen eines Automaten $A = (Q, \Sigma, \delta, q_0, F)$ führt mindestens dann zu einem definierten Automaten $A' = (Q', \Sigma, \delta', q_0, F')$, wenn man gleichzeitig alle unerreichbaren Zustände entfernt.

Sei U gleich der Menge der unerreichbaren Zustände in A . Dann setzen wir $Q' = Q \setminus U$. Natürlich gilt $q_0 \in Q'$.

Wir definieren $F' = F \cap Q'$.

Die Übergangsfunktion δ wird eingeschränkt auf $Q' \times \Sigma$, d. h. es wird definiert $\delta' = \delta \cap ((Q' \times \Sigma) \times Q)$. Die Funktionswerte werden dabei nicht verändert.

Nun ist nachzuweisen, dass δ' die Funktionalität $Q' \times \Sigma \rightarrow Q'$ hat, d. h. $\delta'(Q', \Sigma) \subseteq Q'$. Andernfalls wäre der Automat A' nicht wohldefiniert.

Seien $q \in Q', x \in \Sigma$. Dann gibt es ein Wort $w \in \Sigma^*$, so dass $\hat{\delta}(q_0, w) = q$. Mithin gilt $\hat{\delta}(q_0, wx) = \delta(q, x) = \delta'(q, x)$. Damit ist $\delta'(q, x)$ erreichbar, d. h. $\delta'(q, x) \in Q'$. A' ist also wohldefiniert.

Nun zeigen wir $L(A') = L(A)$ wie folgt.

Zunächst gilt $\hat{\delta}(q_0, w) = \hat{\delta}'(q_0, w)$. Dann rechnen wir

$$\begin{aligned} L(A') &= \{w \in \Sigma^* \mid \hat{\delta}'(q_0, w) \in F'\} \\ &= \{w \in \Sigma^* \mid \hat{\delta}'(q_0, w) \in F \cap Q'\} \\ &= \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} \\ &= \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} \\ &= L(A). \end{aligned}$$

Minimierung durch Quotientenbildung:

Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat. Zunächst beobachten wir, dass die Definitionen 2.59 bzw. 2.62 der Vorlesung für die Äquivalenz \equiv_A (im Folgenden kurz mit \equiv bezeichnet) von Zuständen bzw. für den Quotientenautomaten $(A/\equiv) = (Q/\equiv, \Sigma, \delta', [q_0]_{\equiv}, F/\equiv)$ unabhängig sind von der Existenz unerreichbarer Zustände. Wir setzen also voraus, dass diese Definitionen für beliebige DFA's gelten. Nun beweisen wir ganz allgemein das Lemma 2.63 der Vorlesung, dessen Beweis dort den Übungen überlassen worden war:

$$L(A/\equiv) = L(A).$$

Beweis: Der Beweis stützt sich auf die folgende Implikation:

$$f \equiv g \wedge g \in F \implies f \in F.$$

Nach Definition folgt aus $f \equiv g$ für $u = \epsilon$ die Äquivalenz $\hat{\delta}(f, u) \in F \Leftrightarrow \hat{\delta}(g, u) \in F$. Wegen $\hat{\delta}(g, u) = g \in F$ folgt also $f \in F$.

Wir tragen noch die Definition von F/\equiv nach:

$$F/\equiv = \{[f]_{\equiv} \mid f \in F\}.$$

Nun rechnen wir

$$\begin{aligned} w \in L(A/\equiv) &\iff \hat{\delta}'([q_0]_{\equiv}, w) \in F/\equiv \\ &\iff [\hat{\delta}(q_0, w)]_{\equiv} \in F/\equiv \\ &\iff \exists f. \hat{\delta}(q_0, w) \equiv f \in F \\ &\iff \hat{\delta}(q_0, w) \in F \\ &\iff w \in L(A). \end{aligned}$$

2. Eine mehrfache Hintereinanderausführung der Minimierung durch Entfernen ergibt offenbar stets den gleichen Automaten, weil die Entfernung unerreichbarer Zustände, die Eigenschaft der Erreichbarkeit anderer Zustände unberührt läßt.

Eine mehrfache Hintereinanderausführung der Minimierung durch Quotientenbildung führt zwar stets zu neuen Automaten, weil Klassen von Klassen nie identisch mit den ursprünglichen Klassen sind. Jede nachfolgende Klassenbildung ergibt aber nur einelementige Klassen und kann als Umbenennung der Zustände interpretiert werden.

Es bleibt die Möglichkeit der Vertauschung beider Minimierungen. Sei E die Minimierung durch Entfernen und Q die Quotientenbildung. Dann stellt sich die Frage, ob der Automat $E(Q(A))$ isomorph ist zum Automaten $Q(E(A))$.

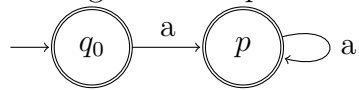
Zum Beweis der Isomorphie von $E(Q(A))$ und $Q(E(A))$ zeigt man, dass die Zuordnung $[p]_{E(Q(A))} \leftrightarrow [p]_{Q(E(A))}$ für erreichbare p eine isomorphe Abbildung ist zwischen den Klassen aus $E(Q(A))$ und den Klassen aus $Q(E(A))$.

Vorbereitung 2

Konstruieren Sie einen DFA $A = (Q, \Sigma, \delta, q_0, F)$, der einen zu q_0 äquivalenten und erreichbaren Zustand $p \neq q_0$ besitzt.

Lösungsvorschlag

Ein mögliches Beispiel ist der folgende Automat über $\Sigma = \{a\}$:



Vorbereitung 3

Skizzieren Sie, wie man für einen gegebenen NFA $N = (Q, \Sigma, \delta, q_0, F)$ und ein $n \in \mathbb{N}$ entscheiden kann, ob ein Wort $w \in L(N)$ existiert mit $|w| \geq n$.

Lösungsvorschlag

Z.B. kann man $\bigcup_{w \in \Sigma^n} \hat{\delta}(q_0, w)$ berechnen und überprüfen, ob von einem dieser Zustände ein Endzustand erreichbar ist.

Oder man konstruiert einen Automaten für die Sprache $\Sigma^n \Sigma^*$ und testet dann den Schnitt beider Automaten auf Leerheit.

Tutoraufgabe 1

Gegeben sei der deterministische endliche Automat

$A = (\{0, 1, 2, 3, 4, 5, 6\}, \{a, b\}, \delta, 0, \{6\})$ mit folgender Übergangsfunktion δ :

| q | $\delta(q, a)$ | $\delta(q, b)$ | q | $\delta(q, a)$ | $\delta(q, b)$ |
|-----|----------------|----------------|-----|----------------|----------------|
| 0 | 1 | 0 | 4 | 1 | 4 |
| 1 | 3 | 5 | 5 | 6 | 0 |
| 2 | 2 | 2 | 6 | 6 | 2 |
| 3 | 6 | 4 | | | |

Konstruieren Sie einen zu A äquivalenten DFA mit einer für $L(A)$ minimalen Anzahl von Zuständen.

Lösungsvorschlag

Da wir vermuten, dass es keine vom Startzustand aus unerreichbaren Zustände gibt, berechnen wir zunächst den Quotientenautomaten in Schritten a) und b). Im Schritt c) überzeugen wir uns abschließend, dass alle Zustände erreichbar sind. Dieses Vorgehen ist gerechtfertigt durch VA 1.

- a) Wir wenden den $O(n^4)$ -Algorithmus aus der Vorlesung an zur Berechnung der äquivalenten Zustandspaare (Relation \equiv_A über Q) wie folgt.

Wir markieren nun in einem ersten Schritt alle Paare von Zuständen, bei denen einer ein Endzustand und der andere kein Endzustand ist. Nach Definition sind diese Paare sicher nicht in der Relation \equiv_A). Es genügt, die Paare unterhalb der Diagonalen zu nehmen, d. h. (x, y) mit $y < x$.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|------------|------------|------------|------------|------------|------------|---|
| 0 | – | – | – | – | – | – | – |
| 1 | | – | – | – | – | – | – |
| 2 | | | – | – | – | – | – |
| 3 | | | | – | – | – | – |
| 4 | | | | | – | – | – |
| 5 | | | | | | – | – |
| 6 | \times_0 | \times_0 | \times_0 | \times_0 | \times_0 | \times_0 | – |

Nun werden Zustandspaare (p, q) , solange das möglich ist, markiert, falls $(\delta(p, x), \delta(q, x))$ mit gewisser Eingabe x bereits markiert ist. Die Reihenfolge ist zwar beliebig, aber wir versuchen, die Reihenfolge günstig zu wählen im Hinblick auf den Arbeitsaufwand.

1. Markierung von $(3, 0), (3, 1), (3, 2), (4, 3), (5, 0), (5, 1), (5, 2), (5, 4)$ wegen:

$$\{3, 0\} \xrightarrow{a} \{6, 1\}, \quad \{3, 1\} \xrightarrow{a} \{6, 3\}, \quad \{3, 2\} \xrightarrow{a} \{6, 2\},$$

$$\{4, 3\} \xrightarrow{a} \{1, 6\},$$

$$\{5, 0\} \xrightarrow{a} \{6, 1\}, \quad \{5, 1\} \xrightarrow{a} \{6, 3\}, \quad \{5, 2\} \xrightarrow{a} \{6, 2\}, \quad \{5, 4\} \xrightarrow{a} \{6, 1\}.$$

2. Markierung von $(1, 0), (2, 1), (4, 1)$ wegen:

$$\{1, 0\} \xrightarrow{a} \{1, 3\}, \quad \{2, 1\} \xrightarrow{b} \{5, 2\}, \quad \{4, 1\} \xrightarrow{a} \{3, 1\}.$$

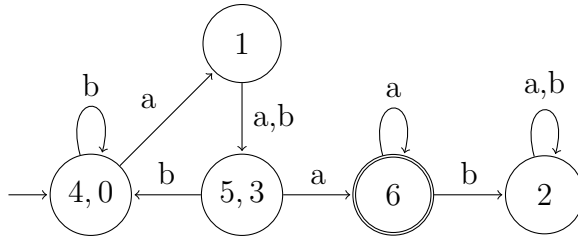
3. Markierung von $(2, 0), (4, 2)$ wegen:

$$\{2, 0\} \xrightarrow{a} \{2, 1\}, \quad \{4, 2\} \xrightarrow{a} \{1, 2\}.$$

Es verbleiben die Paare $(4, 0)$ und $(5, 3)$, die sich nur in unmarkierte Paare transformieren lassen. $(4, 0)$ und $(5, 3)$ sind also in der Relation \equiv_A . Ergebnis:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|------------|------------|------------|------------|------------|------------|---|
| 0 | — | — | — | — | — | — | — |
| 1 | \times_2 | — | — | — | — | — | — |
| 2 | \times_3 | \times_2 | — | — | — | — | — |
| 3 | \times_1 | \times_1 | \times_1 | — | — | — | — |
| 4 | \equiv_A | \times_2 | \times_3 | \times_1 | — | — | — |
| 5 | \times_1 | \times_1 | \times_1 | \equiv_A | \times_1 | — | — |
| 6 | \times_0 | \times_0 | \times_0 | \times_0 | \times_0 | \times_0 | — |

- b) „Kollabierung“ des Automaten durch Identifizierung äquivalenter Zustände (Klassenbildung).

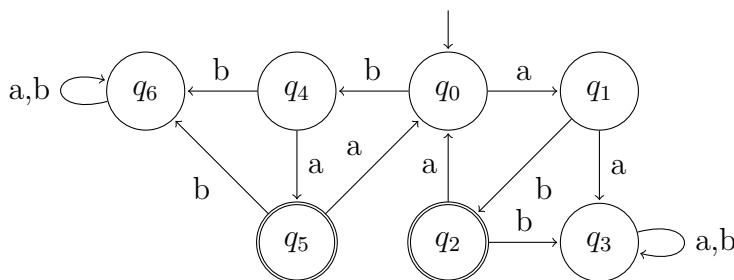


- c) Erreichbare Zustände.

Offensichtlich sind alle Zustände erreichbar. Es müssen also keine Zustände zwecks Minimierung entfernt werden.

Tutoraufgabe 2

1. Sei $\Sigma = \{a, b\}$. Zeigen Sie, dass im untenstehenden DFA $q_3 \not\equiv_M q_4$ und $q_3 \equiv_M q_6$ gelten:



2. Sei $L = L(a^*b^*)$. Zeigen oder widerlegen Sie die folgenden Äquivalenzbeziehungen:

$$aa \equiv_L \epsilon, \quad ab \equiv_L aa, \quad aba \equiv_L abba, \quad aba \equiv_L \epsilon.$$

3. Sei nun $\Sigma_n = \{a_1, \dots, a_n\}$ und

$$L_n = \{w \in \Sigma_n^* \mid \forall a \in \Sigma_n. \#_a(w) = 1\}.$$

Zeigen Sie, dass jeder DFA, der L_n akzeptiert, mindestens $2^n + 1$ Zustände haben muss. Betrachten Sie dazu die Äquivalenzklassen von \equiv_{L_n} .

Lösungsvorschlag

1. $\hat{\delta}(q_3, a) = q_3 \notin F$, aber $\hat{\delta}(q_4, a) = q_5 \in F$. Damit ist die Definition von Äquivalenz nicht erfüllt.

2. $aa \equiv_L \epsilon$:

$$aaw \in L \iff aaw \in a^*b^* \iff aaw = x_1x_2 \dots x_my_1y_2 \dots y_n \text{ mit } x_i = a, \\ y_j = b \text{ für gewisse } m \geq 2 \text{ und } n \geq 0 \iff w = x_3x_4 \dots x_my_1y_2 \dots y_n \text{ mit } x_i = a, \\ y_j = b \text{ für gewisse } m \geq 2 \text{ und } n \geq 0 \iff w \in a^*b^* \iff w \in L \iff \\ \epsilon w \in L.$$

$ab \not\equiv_L aa$:

Für $w = a$ gilt $abw = aba \notin L$ und $aaw = aaa \in L$.

$aba \equiv_L abba$:

Für alle w gilt $abaw \notin L$ und $abbaw \notin L$.

$aba \not\equiv_L \epsilon$:

Für $w = \epsilon$ gilt $abaw = aba \notin L$ und $\epsilon w = \epsilon \in L$.

3. Betrachte Äquivalenzklassen von \equiv_{L_n} . Für jede Teilmenge $A \subseteq \Sigma$ sei w_A ein beliebiges Wort, welches jedes Zeichen aus A genau einmal enthält.

Für beliebige $A, A' \subseteq \Sigma$ mit $A \neq A'$ und, o.B.d.A., mit einem $x \in A', x \notin A$ gilt nun einerseits $w_A w_{\Sigma \setminus A} \in L_n$, aber andererseits $w_{A'} w_{\Sigma \setminus A} \notin L_n$. Damit ist $w_A \not\equiv_{L_n} w_{A'}$. Wir erhalten also für jede der 2^n Teilmengen von Σ eine Äquivalenzklasse in \equiv_{L_n} . Eine weitere Äquivalenzklasse hat den Repräsentanten aa ($a \in \Sigma$), welches zu keinem der w_A äquivalent ist.

Da die Äquivalenzklassen die Zustände des kanonischen Minimalautomaten sind, ergibt sich daraus die untere Schranke $2^n + 1$ für die Anzahl der Zustände.

Tutoraufgabe 3

Skizzieren Sie ein Entscheidungsverfahren, welches für einen gegebenen DFA $M = (Q, \Sigma, \delta, q_0, F)$ feststellt, ob $|L(M)| \leq 100$.

Lösungsvorschlag

Wir entfernen zunächst alle nicht erreichbaren Zustände, sowie alle Zustände, von denen kein Endzustand mehr erreichbar ist. Diese Transformationen ändert die akzeptierte Sprache nicht, allerdings ist der resultierende Automat ein NFA, den wir mit N bezeichnen. Nun testen wir, ob N einen Zyklus hat. Falls das der Fall ist, so wissen wir, dass $L(N)$ unendlich ist und insbesondere $|L(M)| = |L(N)| > 100$.

Hat N keine Zyklen, so kann er keine Wörter akzeptieren, die länger sind als der längste Pfad durch den Automaten. Insbesondere müssen die Wörter kürzer sein als $|Q|$.

Wir können also einfach naiv alle Wörter bis Länge $|Q|$ aufzählen und jeweils testen, ob der Automat sie akzeptiert. So erhalten wir die gesamte Sprache $L(M)$ und prüfen zuletzt, ob sie mindestens 100 Wörter enthält.