Technische Universität München
Fakultät für Informatik
Lehrstuhl Informatik XIV
Prof. Dr. Ernst W. Mayr
Dr. Werner Meixner

Sommersemester 2015
Lösungsblatt 3
11. Mai 2015

Theoretische Informatik

Hausaufgabe 1 (5 Punkte)

Sei M eine nichtleere Menge. Wir nennen eine binäre Relation $R \subseteq M \times M$ über einer Menge M vollständig, falls die folgende Implikation (Regel) gilt:

$$(Av) \qquad (x,y) \in R \wedge (u,v) \in R \implies (x,v) \in R \wedge (u,y) \in R\,.$$

1. Sei V eine Menge von vollständigen Relationen über M. Zeigen Sie, dass dann der Durchschnitt aller $R \in V$ wieder vollständig ist, d.h.

   $$\bigcap_{R\in V} R \quad \text{ist vollständig.}$$

   Dabei gelte $\bigcap_{R\in V} R = M \times M$, falls V leer ist.

2. Sei R eine binäre Relation über M. Wir definieren die vollständige Hülle R^{v} von R durch

   $$R^{\mathrm{v}} := \bigcap_{(S \supseteq R \text{ und } S \text{ erfüllt } Av)} S\,.$$

   Seien $M = \mathbb{N}$ und $R = \{(x,y) \in \mathbb{N} \times \mathbb{N}\,;\, y = x + 10 \wedge x \in [10]\}$. Berechnen Sie R^{v} als Mengenprodukt $A \times B$.

Lösung

Ein ähnlicher Begriff der Vollständigkeit stammt aus der Graphentheorie (siehe DS).

1. Sei $U = \bigcap_{R\in V} R$.

   Seien $(x,y) \in U$ und $(u,v) \in U$. Wir zeigen, dass $(x,v) \in U$ und $(u,y) \in U$ folgt.

   Aus $(x,y) \in U$ und $(u,v) \in U$ folgt für alle $R \in V$, dass $(x,y) \in R$ und $(u,v) \in R$ gelten. Da alle $R \in V$ vollständig sind, folgt mit Regel Av für alle $R \in V$, dass $(x,v) \in R$ gilt Analog folgt, dass $(u,y) \in R$ gilt. Demzufolge liegen (x,v) und (u,y) auch im Durchschnitt aller $R \in V$ und damit in U. (2P)

2. Offenbar gilt $R = \{(1,11),(2,12),\ldots,(9,19),(10,20)\}$.

   Wir setzen $A = \{x \in M\,;\, \exists y : (x,y) \in R\}$ und $B = \{y \in M\,;\, \exists x : (x,y) \in R\}$.

   Erinnerung DS im WS14/15: A bzw. B heißen Urbild bzw. Bild der Relation R.

   Dann gelten

   (a) $A \times B$ ist vollständig und $R \subseteq A \times B$,

(b) $A \times B \subseteq S$ für alle vollständigen Relationen S, die R enthalten, d.h. mit $R \subseteq S$. (Minimalität)

Beweis:

$A \times B$ ist vollständig:

Seien $(x,y) \in A \times B$ und $(u,v) \in A \times B$. Dann folgen $x,u \in A$ und $y,v \in B$. Nach Definition des Cartesischen Produkts gelten dann auch $(x,v) \in A \times B$ und $(y,v) \in A \times B$.

$R \subseteq A \times B$:

Sei $(x,y) \in R$. Nach Definition gelten dann $x \in A$ und $y \in B$. Es folgt $(x,y) \in A \times B$.

Minimalität:

Sei S eine vollständige Relation mit $R \subseteq S$. Seien $(x,y) \in A \times B$. Wir zeigen $(x,y) \in S$ wie folgt:

Es gelten $x \in A, y \in B$. Nach Definition von A gibt es ein v mit $(x,v) \in R$. Entsprechend gibt es ein u mit $(u,y) \in R$.

Da R in S enthalten ist, folgt $(x,v) \in S$ und $(u,y) \in S$. Da S vollständig ist, folgt mindestens $(x,y) \in S$, was zu zeigen war.

Es folgt $R^{\mathrm{v}} = A \times B$. (2P)

Berechnung von R^{v}:

Es gilt $A = [10]$ und $B = [11,20] = \{11,12,\ldots 20\}$. Dann gilt, wie oben gezeigt wurde:

$$R^{\mathrm{v}} = A \times B = [10] \times [11,20]\,. \qquad \text{(1P)}$$

Hausaufgabe 2 (5 Punkte)

Sei $\Sigma = \{(,)\}$ ein Alphabet, bestehend aus einer öffnenden bzw. schließenden Klammer. Sei $G = (\{S\}, \Sigma, P, S)$ eine Grammatik mit Axiom S und den Produktionen

$$S \underset{P}{\rightarrow} \epsilon\,, \qquad S \underset{P}{\rightarrow} (S)\,, \qquad S \underset{P}{\rightarrow} SS\,.$$

1. Konstruieren Sie nach Lemma 12 der Vorlesung eine kontextfreie Grammatik $G' = (V', \Sigma, P', S)$, die $L(G)$ erzeugt.
2. Zeigen Sie, dass es für $z' = ()(())$ eine Ableitung von S in G' der Länge 10 gibt.
3. Weisen Sie nach, dass $w = ())($ nicht in $L(G')$ liegt, indem Sie nach Satz 20 für hinreichend großes m die Folge der Mengen T_m^4 der in m Schritten aus S ableitbaren Wörter w der Länge höchstens 4 in extensionaler Form bestimmen und die Aussage $w \notin T_m^4$ durch Rechnung entscheiden.

Lösung

1. Sei $P' = \emptyset$.
   Im 1. Schritt nach Lemma 12 werden folgende Produktionen zu P' hinzugefügt: $S \underset{P}{\rightarrow} ()$, $S \underset{P}{\rightarrow} S$ und wir erhalten
   $$P': \qquad S \underset{P}{\rightarrow} (S)\,, \qquad S \underset{P}{\rightarrow} SS\,, \qquad S \underset{P}{\rightarrow} ()\,, \qquad S \underset{P}{\rightarrow} S\,.$$
   Im 2.Schritt wird S auf den rechten Seiten beseitigt.

   Es werden folgende Produktionen zu P' hinzugefügt:

   $S \underset{P}{\rightarrow} \epsilon$, $S \underset{P}{\rightarrow} T$.

   Ersetzung S durch T in allen rechten Seiten:

   $S \underset{P}{\rightarrow} (T)$, $S \underset{P}{\rightarrow} TT$, $S \underset{P}{\rightarrow} ()$, $S \underset{P}{\rightarrow} T$.

   In dieser letzten Zeile wird S nun vollständig entfernt:

   $T \underset{P}{\rightarrow} (T)$, $T \underset{P}{\rightarrow} TT$, $T \underset{P}{\rightarrow} ()$, $T \underset{P}{\rightarrow} T$.

   Ergebnis:

   $P': \qquad S \underset{P}{\rightarrow} \epsilon$, $S \underset{P}{\rightarrow} T$, $T \underset{P}{\rightarrow} (T)$, $T \underset{P}{\rightarrow} TT$, $T \underset{P}{\rightarrow} ()$, $T \underset{P}{\rightarrow} T$. (2P)

2. Da die Produktion $T \underset{P}{\rightarrow} T$ beliebig oft anwendbar ist, gilt

   $S \underset{G}{\rightarrow} T \underset{G}{\rightarrow} T \underset{G}{\rightarrow} \dots \underset{G}{\rightarrow} T \underset{G}{\rightarrow} TT \underset{G}{\rightarrow} ()T \underset{G}{\rightarrow} ()(T) \underset{G}{\rightarrow} ()(())\,.$ (1P)

3. Induktiver Aufbau der Mengen T^4_m wie folgt:
   $$\begin{aligned} T^4_0 &= \{S\}\,, \\ T^4_1 &= T^4_0 \cup \{\epsilon, T\}\,, \\ T^4_2 &= T^4_1 \cup \{(T), TT, ()\}\,, \\ T^4_3 &= T^4_2 \cup \{(T)T, T(T), (TT), TTT, (()), ()T, T()\}\,, \\ T^4_4 &= T^4_3 \cup \{TTTT, ()TT, TT(), ()()\}\,, \\ T^4_5 &= T^4_4 \cup \{\}\,, \\ \vdots &= \vdots \\ T^4_m &= T^4_4\,. \end{aligned}$$
   Offenbar gilt $(()(\notin T^4_m$ für alle m, mithin $(()(\notin L(G')$. (2P)

   Bemerkung: Prinzipiell kann man $x \in L(G)$ entscheiden, falls G monoton ist. Es gibt im Einzelfall meist aber effizientere Überlegungen.

Hausaufgabe 3 (5 Punkte)

Nach Beispiel 16 der Vorlesung wird die Sprache $L = \{a^n b^n c^n\,;\, n \in \mathbb{N}\}$ von der Grammatik $G = (\{S, X, Y\}, \{a, b, c\}, P, S)$ mit folgenden Produktionen erzeugt.

$$\begin{array}{rclcrcl} S & \rightarrow & aSXY \mid abY\,, & & & & \\ YX & \rightarrow & XY\,, & & bX & \rightarrow & bb\,, \\ bY & \rightarrow & bc\,, & & cY & \rightarrow & cc\,. \end{array}$$

1. Geben Sie eine Ableitung für $a^2b^2c^2$ von S in G an.

2. Modifizieren Sie G so zu einer Grammatik G', dass G' die Sprache $L' = L \cup \{\epsilon\}$ erzeugt.

3. Konstruieren Sie eine kontexsensitive Grammatik G'' die L' erzeugt.

Lösung

1. $S \underset{G}{\rightarrow} aSXY \underset{G}{\rightarrow} aabYXY \underset{G}{\rightarrow} aabXYY \underset{G}{\rightarrow} aabbYY \underset{G}{\rightarrow} aabbcY \underset{G}{\rightarrow} aabbcc\,.$

   (1P)

2. Sei $G' = (\{T, S, X, Y\}, \{a, b, c\}, P', T)$ mit $P' = P \cup \{T \rightarrow S, T \rightarrow \epsilon\}$.
   Dann gilt $L(G') = L'$. (2P)

3. Es gibt mehrere Möglichkeiten, die Anforderungen zu erfüllen. Sei $G'' = (\{T, S, X, Y, X_1, Y_1\}, \{a, b, c\}, P', T)$.

   Wir definieren P'', indem wir die Regel $YX \rightarrow XY$ aus P'' entfernen und die folgenden Regeln einfügen:

   $$\begin{array}{rclcrcl} YX & \rightarrow & Y_1X\,, & \quad & Y_1X & \rightarrow & Y_1X_1\,, \\ Y_1X_1 & \rightarrow & XX_1\,, & \quad & XX_1 & \rightarrow & XY\,. \end{array}$$

   (2P)

Hausaufgabe 4 (5 Punkte)

Seien $G_1 = (V_1, \Sigma_1, P_1, S_1)$ und $G_2 = (V_2, \Sigma_2, P_2, S_2)$ Grammatiken des gleichen Typs $k \in \{0, 1, 2, 3\}$.

1. Geben Sie eine Grammatik G vom gleichen Typ wie G_1 an, so dass gilt

   $$L(G) = L(G_1) \cup \{\epsilon\}.$$

2. Geben Sie eine Grammatik H vom gleichen Typ wie G_1 an, so dass gilt

   $$L(H) = L(G_1) \cup L(G_2).$$

Lösung

1. Falls $\epsilon \in L(G_1)$, dann können wir $G = G_1$ setzen. Andernfalls sei T eine Variable, die in V_1 nicht vorkommt. Dann definieren wir

   $$G = (\, V_1 \cup \{T\}\,,\, \Sigma_1\,,\, P_1 \cup \{T \rightarrow S_1,\, T \rightarrow \epsilon\}\,,\, T\,)\,.$$

   (2P)

2. Wir betrachten laut Vorlesung Grammatiken, in denen alle Produktionen in ihrer linken Seite mindestens eine Variable enthalten. Typ 3 Grammatiken nehmen wir als rechtslinear an. Gegebenenfalls wird eine linkslineare Grammatik äquivalent in eine rechtslineare umgeschrieben.

Zunächst ist durch Umbezeichung der Variablen dafür zu sorgen, dass $V_1 \cap V_2 = \emptyset$ gilt. Sei nun T eine Variable, die weder in V_1 noch V_2 vorkommt. Dann definieren wir

$$H' = (\, V_1 \cup V_2 \cup \{T\}\,,\ \Sigma_1 \cup \Sigma_2\,,\ P_1 \cup P_2 \cup \{T \to S_1,\ T \to S_2\}\,,\ T\,)\,.$$

Falls G_1 oder G_2 vom Typ 1 sind und eine von beiden erzeugten Sprachen das Wort ϵ enthält, dann wird die Produktion $T \to \epsilon$ zu den Produktionen von H' hinzugefügt und gleichzeitig jede Produktion $S_1 \to \epsilon$ oder $S_2 \to \epsilon$ gestrichen. Das Ergebnis sei dann H. Andernfalls ist nichts zu tun und wir setzen $H = H'$.

(3P)

Zusatzaufgabe 2 (Wird nicht korrigiert.)

Um Klammerausdrücke zu zählen, beschränkt man sich auf Ausdrücke ohne Variable und einen einzigen Klammertyp von öffnenden und schließenden Klammern „(“ bzw. „)“. Man betrachtet also eine Menge $K \subseteq \Sigma^*$ von Wörtern über der Zeichenmenge $\Sigma = \{(,)\}$, die mit folgenden Regeln erzeugt werden kann. Dabei bezeichne ε das leere Wort, und die Konkatenation von Wörtern wird durch Nebeneinanderschreiben notiert. K heißt Menge der korrekten Klammerausdrücke über Σ.

- $\varepsilon \in K$,
- $w \in K \implies (w) \in K$,
- $w_1, w_2 \in K \implies w_1 w_2 \in K$.

Wir nennen ein Wort $w \in \Sigma^*$ semipositiv, falls $|u|_(\geq |u|_)$ für alle Anfangsteilwörter u von w gilt. Wir setzen im Folgenden voraus, dass $w \in \Sigma^*$ genau dann ein korrekter Klammerausdruck ist, falls w semipositiv ist und $|w|_(= |w|_)$ gilt.

Die Anzahl von korrekten Klammerausdrücken mit $n \in \mathbb{N}_0$ öffnenden Klammern heißt Catalan-Zahl C_n. Beweisen Sie die folgende Aussage.

$$C_n = \frac{1}{n+1}\binom{2n}{n}\,.$$

Lösung

Wir bezeichnen die Menge der korrekten Klammerausdrücke $w \in K$ der Gesamtlänge $|w| = |w|_(+ |w|_) = n$ für alle $n \in \mathbb{N}_0$ mit K_n, d. h. $K_n = \{w \in K\,;\, |w| = n\}$.
Für alle $w \in K_n$ entsteht durch Voraustellung einer öffnenden Klammer ein Wort $w' = (w$. Wir definieren die Menge $K'_n = (\,K_n = \{(w\,;\, w \in K_n\}$.
Es gilt $|K'_n| = |K_n|$.
Die Bestimmung von $|K'_n|$ ist Spezialfall des Ballot-Problems mit $a = |w|_($ und $b = |w|_)$ und $w \in K'_n$. Danach gilt zunächst $a - b = 1$ und $a + b = n + 1$, und insbesondere

$$|K'_n| = \frac{a-b}{a+b} \cdot \binom{a+b}{a}\,.$$

Im Folgenden ist zu beachten, dass die Catalan-Zahl C_n die Menge der korrekten Klammerausdrücke der Länge $2n$ abzählt. Damit gelten im Folgenden für die Parameter a und

b die Gleichungen $a = n + 1$, $b = n$ und insbesondere $a + b = 2n + 1$.

$$
\begin{aligned}
C_n \;=\; |K_{2n}| \;&=\; |K'_{2n}| \\
&= \frac{1}{2n+1} \cdot \binom{2n+1}{n+1} \\
&= \frac{1}{2n+1} \cdot \frac{(2n+1)^{\underline{n+1}}}{(n+1)!} \;=\; \frac{(2n)^{\underline{n}}}{(n+1)!} \;=\; \frac{1}{n+1} \cdot \frac{(2n)^{\underline{n}}}{(n)!} \\
&= \frac{1}{n+1} \cdot \binom{2n}{n}.
\end{aligned}
$$

Hinweis: *Die Vorbereitungsaufgaben bereiten die Tutoraufgaben vor und werden in der Zentralübung unterstützt. Tutoraufgaben werden in den Übungsgruppen bearbeitet. Hausaufgaben sollen selbstständig bearbeitet und zur Korrektur und Bewertung abgegeben werden.*

Vorbereitung 1

Reguläre Ausdrücke stellen Elemente einer Mengenalgebra dar. Die Operationen der Mengenalgebra sind die Konkatenation, Vereinigung („|“), Sternbildung („*“) und die Operationen $\emptyset$, ϵ, sowie eine endliche Anzahl von Konstanten, d.h. nullstelligen Operationen $a, b, \ldots$, die eineindeutig den Elementen eines Alphabets Σ entsprechen. Reguläre Ausdrücke sind also nichts anderes als algebraische Ausdrücke ohne Variable über einer Mengenalgebra.
Studieren Sie die Definition des Begriffs des regulären Ausdrucks und beantworten Sie kurz die folgenden Fragen:

1. Welche Mengen stellen die Ausdrücke $\emptyset$ bzw. ϵ bzw. a, b dar?
2. Geben sie einen regulären Ausdruck an, der eine Sprache A über dem Alphabet $\Sigma = \{0, 1\}$ mit den Eigenschaften $01 \in A$ und $A^*A = A$ darstellt.
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.

Lösung

Reguläre Ausdrücke gehorchen den Gesetzen algebraischer Ausdrücke. Die Sprache der regulären Ausdrücke wird nicht vollständig formalisiert, insbesondere greifen wir auf bekannte Gesetze der Klammerbildung bei algebraischen Ausdrücken zurück.

1. Wir benützen eine drucktechnische Unterscheidung durch Fettdruck.

   $L(\boldsymbol{\emptyset}) = \emptyset$, $L(\boldsymbol{\epsilon}) = \{\epsilon\}$, $L(\boldsymbol{a}) = \{a\}$, $L(\boldsymbol{b}) = \{b\}$.

2. $A = \{0, 1\}^*$ erfüllt die Anforderungen an die Sprache. (Beweis?)

   Sei $\alpha = (\mathbf{0}|\mathbf{1})^*$ ein regulärer Ausdruck.

   Es gilt $A = L(\alpha)$. (Beweis?)

   Die mengentheoretisch kleinste Sprache, für die die Anforderungen erfüllt sind, ist

   $A' := \{01\}^*\{01\}$, d.h. $A' = L((\mathbf{01})^*(\mathbf{01}))$.

   Beweis:

   Jede Sprache A, für die die Anforderungen erfüllt sind, enthält A':

   Falls $A^*A = A$ und $01 \in A$, folgt $\{01\} \subseteq A$ mithin

   $\{01\}^*\{01\} \ \subseteq \ A$

   d.h. $A' \subseteq A$.

   Umgekehrt gelten für $A' = \{01\}^*\{01\}$ die Anforderungen,

   d.h. die Implikationen

   $01 \in A'$ und $A'^*A' \subseteq A'$.

Algebraischer Beweis von $A'^*A' = A'$:

$$
\begin{aligned}
A'^*A' &= (\{01\}^*\{01\})^*\{01\}^*\{01\} \\
&= (\{01\}^*)^*\{01\}^*\{01\}^*\{01\} && \text{wg. } (XY)^* = X^*Y^* \\
&= \{01\}^*\{01\}^*\{01\}^*\{01\} && \text{wg. } (X^*)^* = X^* \\
&= \{01\}^*\{01\} && \text{wg. } (XY)^* = X^*Y^* \\
&= A' \,.
\end{aligned}
$$

3. Eine Lösung ist $000^*10^*10^*10^*$.

Vorbereitung 2

Reguläre Ausdrücke α, β heißen äquivalent, i.Z. $\alpha \equiv \beta$, genau dann, wenn $L(\alpha) = L(\beta)$ gilt.

Beweisen Sie für alle regulären Ausdrücke α und β: $(\alpha\beta)^*\alpha \equiv \alpha(\beta\alpha)^*$.

Lösung

Die Äquivalenz $r \equiv s$ von regulären Ausdrücken r und s ist als Gleichheit der zugeordneten Sprachen $L(r)$ und $L(s)$ definiert. Für diese Äquivalenzgleichungen gelten einfache Regeln.

Wir zeigen zunächst für zwei beliebige Sprachen A und B die Gleichung

$$
(AB)^n\, A = A\,(BA)^n \tag{1}
$$

per Induktion:

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

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

Nun wird die Gleichung definitionsgemäß semantisch mit den zugeordneten Sprachen interpretiert und die für Mengen von Wörtern, d.h. Sprachen, geltenden Rechenregeln angewandt:

$$
\begin{aligned}
L\left((\alpha\beta)^*\alpha\right) &= \left(L(\alpha)L(\beta)\right)^*\, L(\alpha) \\
&= \bigcup_{n\geq 0} \left(L(\alpha)L(\beta)\right)^n\, L(\alpha) \\
&= \bigcup_{n\geq 0} L(\alpha)\,\left(L(\beta)L(\alpha)\right)^n && \text{(mit Gl. 1)} \\
&= L(\alpha)\,\left(L(\beta)L(\alpha)\right)^* \\
&= L\left(\alpha(\beta\alpha)^*\right) \,.
\end{aligned}
$$

Vorbereitung 3

Wann genau ist die von einem endlichen Automaten erzeugte Sprache endlich? Beantworten Sie diese Frage anhand der Länge der Pfade, die es in dem Übergangsgraphen eines Automaten gibt.

Lösung

Antwort:
Genau dann, wenn es keinen erreichbaren und „produktiven" (d.h., von dort aus ist ein Endzustand erreichbar) Zustand gibt, der auf einem Kreis liegt.

Vorbereitung 4

Wahr oder falsch? Begründen Sie Ihre Antwort!

1. Zu jedem ϵ-NFA $N = (Q, \Sigma, \delta, \{q_0\}, F)$ gibt es einen äquivalenten (ϵ-freien) NFA $N' = (Q', \Sigma, \delta', \{q_0\}', F')$, so dass $|Q'| \leq |Q|$ gilt.
2. Für jeden NFA $N = (Q, \Sigma, \delta, \{q_0\}, F)$ gilt: Wenn $F = Q$, dann ist $L(N) = \Sigma^*$.
3. $a(ab)^*(ba)^*b \equiv a(ab|ba)^*b$.
4. Wenn $L \subseteq \Sigma^*$ regulär ist und $\Gamma \subseteq \Sigma$,
   dann ist $L' = \{w \in L \mid w \text{ enthält nur Zeichen aus } \Gamma\}$ ebenfalls regulär.

Lösung

1. Wahr! Die Konstruktion in der Vorlesung liefert $Q = Q'$.
2. Falsch! Gegenbsp.: $\Sigma = \{a\}, Q = F = \{q_0\}, \delta(q_0, a) = \emptyset$. Dann ist $L(N) = \{\epsilon\}$.
3. Falsch! $w = aabbaabb \in L(a(ab|ba)^*b)$, aber $w \notin L(a(ab)^*(ba)^*b$.
4. Wahr! Abschluss unter Schnitt: $L' = L \cap \Gamma^*$.

Tutoraufgabe 1 (Myhill-Verfahren)

Wir betrachten einen nichtdeterministischen Automaten

$$N = (Q, \Sigma, \delta, S, F) = (\{s_0, s_1\}, \{a, b\}, \delta, \{s_0\}, \{s_1\})$$

mit $\delta(s_0, a) = \{s_1\}$ und $\delta(q, x) = \emptyset$ für $(q, x) \neq (s_0, a)$.

1. Konstruieren Sie mit dem Myhill-Verfahren einen endlichen deterministischen Automaten M, für den $L(M) = L(N)$ gilt.

2. Konstruieren Sie nach Vorlesungsmethodik eine reguläre Grammatik G mit $L(G) = L(M)$.

Lösung

Die Graphdarstellung der Übergangsfunktion des gegebenen, sehr einfachen Automaten enthält, abgesehen von den Markierungen von Startzustand und Endzustand, nur 2 Knoten und eine Verbindungskante. Die akzeptierte Sprache $L(N)$ ist gleich $\{a\}$. Der Automat ist nur deshalb nicht deterministisch, weil nicht für alle Eingaben Übergänge definiert sind. Man könnte die Übergänge durch eine sogenannte „partiell definierte“ Funktion $Q\times\Sigma \to Q$ beschreiben, was aber im Konzept des DFA nicht vorgesehen ist.
Bei einem NFA dagegen ist die Modellierung von nicht definierten Übergängen sehr einfach durch die leere Menge als Funktionswert der Übergangsfunktion zu beschreiben. Das angegebene Beispiel ist also nichts anderes als die Beschreibung eines deterministischen, aber nur partiell definierten Automaten durch einen NFA.

Bekanntlich kann man aber deterministische endliche Automaten A mit partieller Übergangsfunktion durch Einführung eines zusätzlichen Zustands „undefiniert“ zu einem DFA B mit gleicher akzeptierter Sprache erweitern. Die vorliegende Aufgabe zeigt, dass diese Erweiterung elegant mit der Myhill-Konstruktion oder Teilmengenkonstruktion gemacht werden kann. Organisiert man die Teilmengenkonstruktion durch geschickte Tabellensortierung so, dass vom Anfangszustand aus durch Eingabe eines Wortes unerreichbare Zustände am Ende der Tabelle liegen und sofort gestrichen werden können, dann liefert diese Konstruktion tatsächlich eine „kleinste“ Erweiterung, die nur aus der Hinzunahme eines einzigen Zustands besteht.
Ein weiteres Thema der Aufgabe ist Satz 28 der Vorlesung, d. h. die Konstruktion einer regulären Grammatik, die die akzeptierte Sprache eines NFA bzw. DFA erzeugt.

1. Nach Satz 33 der Vorlesung konstruieren wir den gesuchten DFA M wie folgt.

   Sei $M = (Q', \Sigma, \delta', q'_0, F')$ mit

$$\begin{aligned}
Q' &:= \{\emptyset, \{s_0\}, \{s_1\}, \{s_0, s_1\}\}\,,\\
\delta'(X, a) &:= \{r \in \delta(q, a)\ ;\ q \in X\} \quad \text{für alle} \quad X \in Q', a \in \Sigma\,,\\
q'_0 &:= \{s_0\}\,,\\
F' &:= \{\{s_1\}, \{s_0, s_1\}\}\,.
\end{aligned}$$

Wir berechnen die Übergangstabelle von δ' in geeigneter Sortierung nach Bedarf, beginnend beim Startzustand wie folgt.

$q \in Q'$	$\delta'(q,a)$	$\delta'(q,b)$
$\{s_0\}$	$\{s_1\}$	$\emptyset$
$\{s_1\}$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$
$\{s_0, s_1\}$	$\{s_1\}$	$\emptyset$

Da der Zustand $\{s_0, s_1\}$ nicht erreichbar ist, kann er aus dem Automat bzw. der Übergangsfunktion gestrichen werden, ohne die akzeptierte Sprache zu verändern.

2. Wir gehen von dem in Teilaufgabe 1 erhaltenen DFA mit 3 Zuständen aus. Die zugehörende Zustandsmenge sei $Q'' = \{\emptyset, \{s_0\}, \{s_1\}\}$.

   Zur Abkürzung bezeichnen wir $q_0 := \{s_0\}, q_1 := \{s_1\}, q_2 := \emptyset$. Nach Satz 28 der Vorlesung ist eine reguläre Grammatik G mit $L(G) = L(M)$ gegeben durch

$$\begin{aligned} G &= (\{q_0, q_1, q_2\}, \Sigma, P, q_0)\,, \\ P &= \{q_0 \to aq_1, q_0 \to bq_2, q_1 \to aq_2, q_1 \to bq_2, q_2 \to aq_2, q_2 \to bq_2\} \\ &\quad \cup \{q_0 \to a\}\,. \end{aligned}$$

Tutoraufgabe 2 (Äquivalente Darstellungen regulärer Sprachen)

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

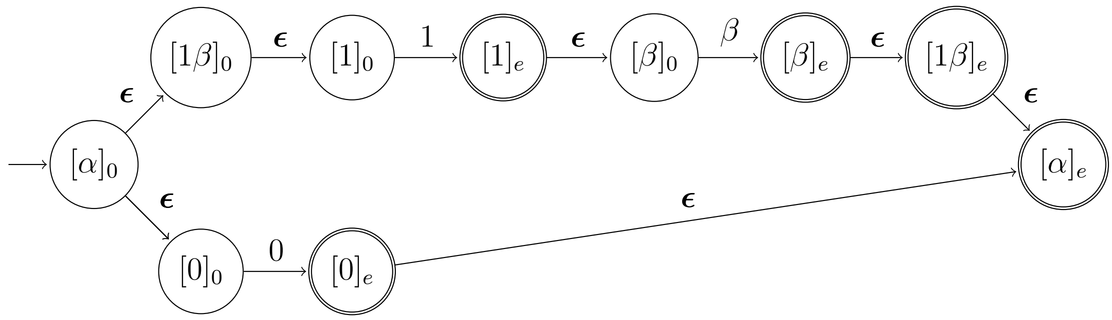
1. Konstruieren Sie mit dem Standardverfahren aus der Vorlesung einen ϵ-NFA A, so dass $L(\alpha) = L(A)$ gilt.
2. Wandeln Sie den erhaltenen Automat in einen äquivalenten NFA ohne ϵ-Übergänge.
3. Konstruieren Sie durch Anwendung des Potenzmengenverfahrens einen DFA, der die Sprache des Ausdrucks α akzeptiert.

Lösung

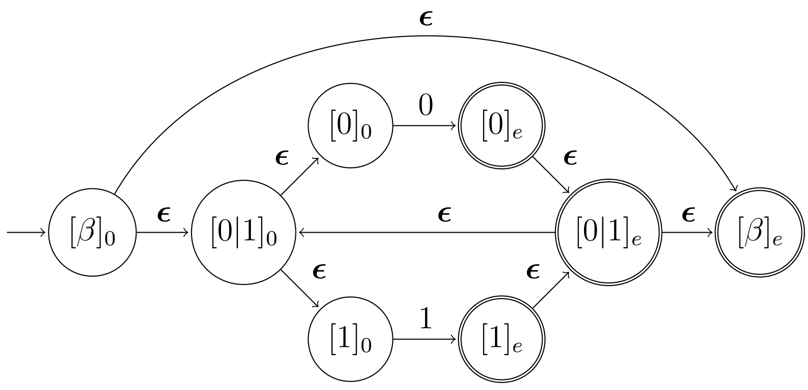
Der Zusammenbau eines Automaten nach dem Schema eines regulären Ausdrucks geschieht mit einem Hilfszeichen, z. B. $\boldsymbol{\epsilon}$ (Fett-Epsilon), das die Übergänge zwischen den Komponentenautomaten regelt. Man erhält zunächst einen NFA über dem erweiterten Alphabet $\Sigma \cup \{\boldsymbol{\epsilon}\}$.

1. Die Konstruktion ist in Satz 43 der Vorlesung beschrieben mit dem Hilfszeichen $\boldsymbol{\epsilon}$. Wir bilden der Schachtelstruktur dieser Konstruktion die Bezeichnung von Start- und Endzuständen nach. Wir bezeichnen den Start- bzw. Endzustand für den Automaten des regulären Ausdrucks α mit $[\alpha]_0$ bzw. $[\alpha]_e$. Außerdem erweitern wir die Übergangsnotation, indem wir zulassen, dass nicht nur Zeichen, sondern mit regulären Ausdrücken beschriebene Sprachen einen Übergang verursachen können.

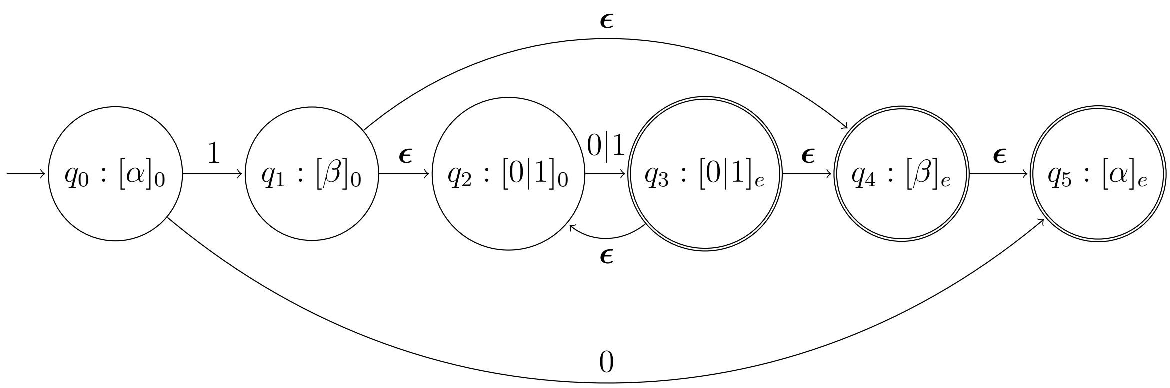
   Man erhält dann den folgenden ϵ-NFA $A = (Q, \Sigma \cup \{\boldsymbol{\epsilon}\}, \{q_0\}, F)$. Wir definieren dabei den regulären Ausdruck $\beta = (0|1)^*$.



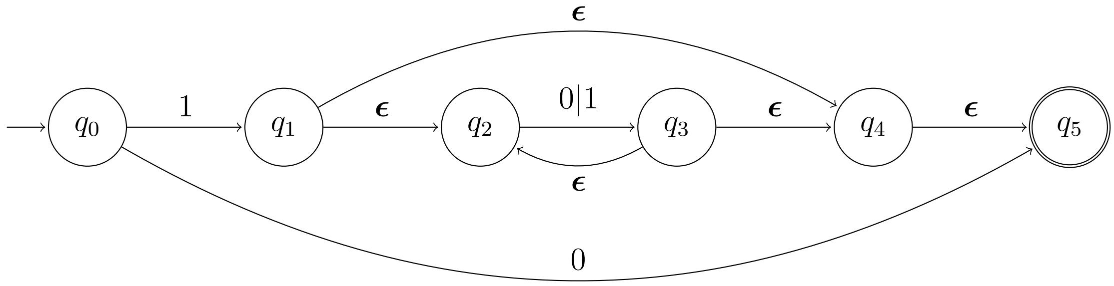
Es folgt der Automat für β:



2. Wir vereinfachen zunächst den obigen ϵ-NFA durch Beseitigung „offensichtlich unnötiger“ Zustände. Das sind Zustände, die keine Endzustände für α 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 und die Übergänge zusammengefasst werden.



D.h.



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

$$\begin{array}{lll} \hat{\delta}(q_0, \boldsymbol{\epsilon}^*) = \{q_0\}\,, & \hat{\delta}(q_1, \boldsymbol{\epsilon}^*) = \{q_1, q_2, q_4, q_5\}\,, & \hat{\delta}(q_2, \boldsymbol{\epsilon}^*) = \{q_2\}\,, \\ \hat{\delta}(q_3, \boldsymbol{\epsilon}^*) = \{q_2, q_3, q_4, q_5,\}\,, & \hat{\delta}(q_4, \boldsymbol{\epsilon}^*) = \{q_4, q_5\}\,, & \hat{\delta}(q_5, \boldsymbol{\epsilon}^*) = \{q_5\}\,. \end{array}$$

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 $a \in \{0, 1\}$:

$$\begin{array}{rcl} \delta'(q_0, 0) & = & \hat{\delta}(\hat{\delta}(\{q_0\}, 0), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_5\}, \boldsymbol{\epsilon}^*) = \{q_5\}\,, \\ \delta'(q_0, 1) & = & \hat{\delta}(\hat{\delta}(\{q_0\}, 1), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_1\}, \boldsymbol{\epsilon}^*) = \{q_1, q_2, q_4, q_5\}\,, \\ \delta'(q_1, 0) & = & \hat{\delta}(\hat{\delta}(\{q_1, q_2, q_4, q_5\}, 0), \boldsymbol{\epsilon}^*) = \hat{\delta}(\{q_3\}, \boldsymbol{\epsilon}^*) = \{q_2, q_3, q_4, q_5\}\,, \\ \delta'(q_1, 1) & = & \ldots \; = \; \{q_2, q_3, q_4, q_5\} \\ & \vdots & \end{array}$$

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 $F' = \{q_5\}$.

q_i	$\delta'(q_i, 0)$	$\delta'(q_i, 1)$
q_0	$\{q_5\}$	$\{q_1, q_2, q_4, q_5\}$
q_1	$\{q_2, q_3, q_4, q_5\}$	$\{q_2, q_3, q_4, q_5\}$
q_2	$\{q_2, q_3, q_4, q_5\}$	$\{q_2, q_3, q_4, q_5\}$
q_3	$\{q_2, q_3, q_4, q_5\}$	$\{q_2, q_3, q_4, q_5\}$
q_4	$\emptyset$	$\emptyset$
q_5	$\emptyset$	$\emptyset$

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 z.B. $q_iq_jq_k$ für $\{q_i, q_j, q_k\}$.

s_i	$\delta''(s_i, 0)$	$\delta''(s_i, 1)$
q_0	q_5	$q_1q_2q_4q_5$
q_5	$\emptyset$	$\emptyset$
$q_1q_2q_4q_5$	$q_2q_3q_4q_5$	$q_2q_3q_4q_5$
$q_2q_3q_4q_5$	$q_2q_3q_4q_5$	$q_2q_3q_4q_5$
$\emptyset$	$\emptyset$	$\emptyset$

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 $F'' = \{q_5, q_1q_2q_4q_5, q_2q_3q_4q_5\}$.

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

Tutoraufgabe 3 (Zyklische Zustandsänderungen im DFA)

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit einer Anzahl n von Zuständen. Zeigen Sie:

1. Sei $w = w_1w_2 \dots w_{2n} \in \Sigma^*$ mit $|w_i| = 1$ ein Wort der Länge $2n$ und sei $q_0, q_1, \dots, q_{2n}$ die Folge der Zustände, die M ausgehend von q_0 bei Eingabe von w annimmt. Dann gibt es k, l mit $k < l$, so dass $q_k = q_l$.

2. Falls es ein Wort w der Länge $2n$ gibt mit $w \in L(M)$, dann gibt es unendlich viele Wörter, die der Automat M akzeptiert.

3. Finden Sie ein Verfahren, welches für einen gegebenen NFA M entscheidet, ob $|L(M)| \leq 100$ gilt.

Lösung

Den hier beschriebenen Sachverhalt sollte jeder Prüfling parat haben. Er behandelt den Beweis des Pumping Lemmas für reguläre Sprachen.

1. Der Beweis ist eine unmittelbare Anwendung des Schubfachprinzips. Denn die genannte Folge ist eine Abbildung von $\{0, 1, \dots, 2n\}$ in Q. Wegen $|\{0, 1, \dots, 2n\}| = 2n+1 > n = |Q|$ folgt, dass mindestens 2 Zahlen auf den gleichen Zustand abgebildet werden.

2. Wir gehen von den gleichen Bezeichungen aus wie in 1. Dann geht M bei Eingabe von $u = w_1w_2 \dots w_k$ in den Zustand q_k über. Die weitere Eingabe von $v = w_{k+1} \dots w_l$ führt zum Zustand q_l. Schließlich führt die Eingabe von $x = w_{l+1} \dots w_{2n}$ in den akzeptierenden Endzustand q_{2n} über. Wegen $q_k = q_l$ kann aber v beliebig oft eingegeben und abschließend mit x in den akzeptierenden Zustand q_{2n} gegangen werden. Damit gilt $uv^ix \in L(M)$ für alle $i \geq 0$. Wegen $v \neq \epsilon$ enthält also $L(M)$ unendlich viele Elemente.

3. 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.