2. Übungsblatt (mit Lösungen)

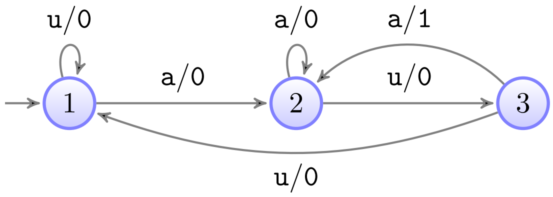
3.0 VU Formale Modellierung

Gernot Salzer, Marion Scholz

25.November 2015

Aufgabe 1 (0.4 Punkte)

Sei $\mathcal{A}$ der folgende Mealy-Automat.



(a) Geben Sie die Ausgabe zur Eingabe $\mathtt{uuaauauaaauaaa}$ an.

(b) Beschreiben Sie $\mathcal{A}$ als 6-Tupel; legen Sie die Übergangsfunktion δ sowie die Ausgabefunktion γ durch eine Tabelle fest.

(c) Berechnen Sie schrittweise $\delta^*(1, \mathtt{uauau})$ und $\gamma^*(1, \mathtt{uauau})$.

(d) Beschreiben Sie die Übersetzungsfunktion $[\mathcal{A}]$.

Lösung

(a) $\gamma^*(1, \mathtt{uuaauauaaauaaa}) = \mathtt{0000010100100}$

(b) $\mathcal{A} = \langle \{1, 2, 3\}, \{\mathtt{a}, \mathtt{u}\}, \{\mathtt{0}, \mathtt{1}\}, \delta, \gamma, 1\rangle$, wobei δ und γ durch die folgenden Tabellen festgelegt werden.

δ	a	u
1	2	1
2	2	3
3	2	1

γ	a	u
1	0	0
2	0	0
3	1	0

(c) $$\begin{aligned}\delta^*(1,\mathtt{uauau}) &= \delta^*(\delta(1,\mathtt{u}),\mathtt{auau})\\ &= \delta^*(1,\mathtt{auau})\\ &= \delta^*(\delta(1,\mathtt{a}),\mathtt{uau})\\ &= \delta^*(2,\mathtt{uau})\\ &= \delta^*(\delta(2,\mathtt{u}),\mathtt{au})\\ &= \delta^*(3,\mathtt{au})\\ &= \delta^*(\delta(3,\mathtt{a}),\mathtt{u})\\ &= \delta^*(2,\mathtt{u})\\ &= \delta^*(\delta(2,\mathtt{u}),\varepsilon)\\ &= \delta^*(3,\varepsilon)\\ &= 3\end{aligned}$$

$$\begin{aligned}\gamma^*(1,\mathtt{uauau}) &= \gamma(1,\mathtt{u})\cdot\gamma^*(\delta(1,\mathtt{u}),\mathtt{auau})\\ &= \mathtt{0}\cdot\gamma^*(1,\mathtt{auau})\\ &= \mathtt{0}\cdot\gamma(1,\mathtt{a})\cdot\gamma^*(\delta(1,\mathtt{a}),\mathtt{uau})\\ &= \mathtt{0}\cdot\mathtt{0}\cdot\gamma^*(2,\mathtt{uau})\\ &= \mathtt{00}\cdot\gamma(2,\mathtt{u})\cdot\gamma^*(\delta(2,\mathtt{u}),\mathtt{au})\\ &= \mathtt{00}\cdot\mathtt{0}\cdot\gamma^*(3,\mathtt{au})\\ &= \mathtt{000}\cdot\gamma(3,\mathtt{a})\cdot\gamma^*(\delta(3,\mathtt{a}),\mathtt{u})\\ &= \mathtt{000}\cdot\mathtt{1}\cdot\gamma^*(2,\mathtt{u})\\ &= \mathtt{0001}\cdot\gamma(2,\mathtt{u})\cdot\gamma^*(\delta(2,\mathtt{u}),\varepsilon)\\ &= \mathtt{0001}\cdot\mathtt{0}\cdot\gamma^*(3,\varepsilon)\\ &= \mathtt{00010}\cdot\varepsilon = \mathtt{00010}\end{aligned}$$

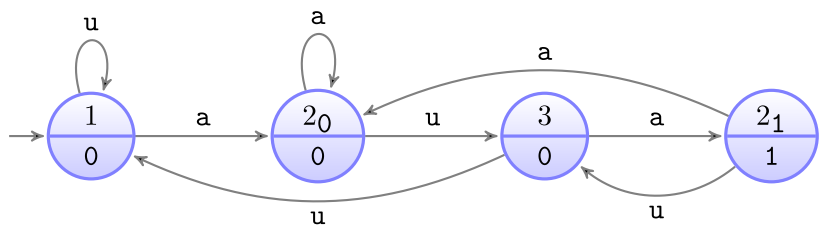
(d) Der Mealy-Automat ist ein `aua`-Detektor: Immer wenn die letzten drei Eingabesymbole das Wort `aua` bilden, wird das Symbol `1` ausgegeben, sonst `0`.

Aufgabe 2 **(0.2 Punkte)**

Finden Sie einen Moore-Automaten, der äquivalent zum Mealy-Automaten aus Aufgabe 1 ist. Geben Sie ein Verfahren an, mit dem sich zu jedem Mealy-Automaten ein äquivalenter Moore-Automat konstruieren lässt.

Lösung

Ein Moore-Automat zeichnet sich dadurch aus, dass sämtliche Übergänge, die zu einem Zustand führen, dieselbe Ausgabe erzeugen. Der Mealy-Automat aus Aufgabe 1 erfüllt diese Bedingung beinahe. Nur zum Zustand 2 führen Übergänge mit den unterschiedlichen Ausgaben `0` und `1`. Wir verdoppeln daher diesen Zustand samt den Übergängen, die vom Zustand wegführen. Die zum Zustand 2 führenden Übergänge teilen wir je nach Ausgabe auf die beiden neuen Zustände $2_{\mathtt{0}}$ und $2_{\mathtt{1}}$ auf.



Im Allgemeinen kann ein Mealy-Automat folgendermaßen in einen Moore-Automaten umgewandelt werden.

- Jeder Zustand q wird in n neue Zustände q_{o_1}, …, q_{o_n} aufgespaltet, wobei o_1, …, o_n die Ausgabesymbole sind, die im Mealy-Automaten bei Übergängen nach q auftreten können.

- Jeder Übergang im Mealy-Automaten vom Zustand q nach q' mit Eingabesymbol s und Ausgabesymbol o führt zu n Übergängen im Moore-Automaten, die von jedem der Zustände $q_{o_1}, \ldots, q_{o_n}$ nach q'_o führen (mit Eingabe s und Ausgabe o).
- Als Startzustand kann ein beliebiger jener Zustände gewählt werden, die aus dem Startzustand des Mealy-Automaten entstanden sind.

Zur formalen Beschreibung des Verfahrens gehen wir von einem beliebigen Mealy-Automaten $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, \gamma, q_0 \rangle$ aus. Einen äquivalenten Moore-Automaten erhalten wir durch

$$\bar{\mathcal{A}} = \langle Q \times \Gamma, \Sigma, \Gamma, \bar{\delta}, \bar{\gamma}, (q_0, o) \rangle$$

wobei o ein beliebiges Ausgabesymbol aus Γ ist. Die neue Ausgabe- und Übergangsfunktion wird definiert durch

$$\begin{aligned} \bar{\gamma}((q,o)) &= o && \text{für alle } q \in Q \text{ und } o \in \Gamma \\ \bar{\delta}((q,o),s) &= (\delta(q,s), \gamma(q,s)) && \text{für alle } q \in Q,\ o \in \Gamma \text{ und } s \in \Sigma \end{aligned}$$

Der so entstehende Moore-Automat kann nicht-erreichbare Zustände enthalten, die man entfernen kann.

Aufgabe 3 (0.3 Punkte)

Zahlen werden im Ternärsystem (Dreiersystem) durch Folgen der Zeichen `0`, `1` und `2` dargestellt, wobei die einzelnen Stellen mit Potenzen von 3 gewichtet werden. Etwa repräsentiert das Ternärnumeral `1201`

- die Zahl $1 \cdot 3^0 + 2 \cdot 3^1 + 0 \cdot 3^2 + 1 \cdot 3^3 = 34$, wenn das Numeral mit der niedrigstwertigen Stelle beginnt, bzw.
- die Zahl $1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 46$, wenn das Numeral mit der höchstwertigen Stelle beginnt.

(a) Geben Sie einen Mealy-Automaten an, der die Summe zweier Ternärnumerale gleicher Länge berechnet und wieder als Ternärnumeral ausgibt. Nehmen Sie an, dass die Numerale mit der *niedrigstwertigen* Stelle beginnen.

Beispiel: Die Eingabe der Addition $\substack{\texttt{122}\\\texttt{120}}$ führt zur Ausgabe `210`, da: $1 + 1 = 2$ ohne Übertrag; $2 + 2 = 1$ mit Übertrag 1; $2 + 0 +$ Übertrag $= 0$ mit Übertrag 1. Folgt als weitere Spalte $\substack{\texttt{0}\\\texttt{0}}$, wird wegen des Übertrags der bisherigen Rechnung als nächstes Symbol `1` ausgegeben.

Welche Bedeutung besitzen die Zustände sowie die Ein- und Ausgabesymbole Ihres Automaten?

Wieviele Zustände benötigen Sie im Allgemeinen, wenn der Automat zwei Numerale zur Basis $n > 3$ addieren soll? Begründen Sie Ihre Antwort.

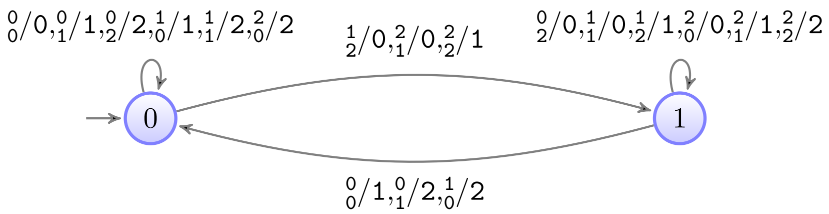
(b) Geben Sie einen Transducer an, der die Summe von Ternärnumeralen gleicher Länge berechnet, die mit der *höchstwertigen* Stelle beginnen. Beispielsweise sind die Paare $(\binom{\mathtt{102}}{\mathtt{020}}, \mathtt{122})$ und $(\binom{\mathtt{122}}{\mathtt{120}}, \mathtt{1012})$ in der Übersetzungsrelation des Transducers enthalten, da $\mathtt{102}_3 + \mathtt{020}_3 = \mathtt{122}_3$ bzw. $\mathtt{122}_3 + \mathtt{120}_3 = \mathtt{1012}_3$ gilt.

Welche Zustände benötigen Sie nun, welche Bedeutung haben sie?

Hinweis: Transducer dürfen in jeder Hinsicht indeterministisch sein: sie können mehrere Folgezustände für eine Eingabe besitzen, und es ist auch das Leerwort als Ein- bzw. Ausgabe erlaubt.

Lösung

(a)

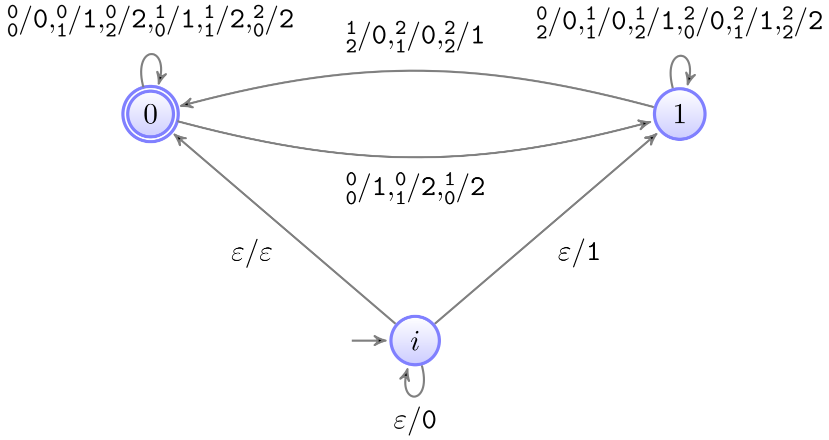


Die Zustände repräsentieren den Übertrag: in Zustand 0 liegt kein Übertrag vor, in Zustand 1 ergibt die bisherige Rechnung einen Übertrag von 1, der bei der nächsten Spalte berücksichtigt werden muss.

Die Eingabesymbole $\genfrac{}{}{0pt}{}{\mathtt{0}}{\mathtt{0}}, \ldots, \genfrac{}{}{0pt}{}{\mathtt{2}}{\mathtt{2}}$ entsprechen den Spalten der Addition, also den Ziffern der beiden Summanden. Die Ausgabesymbole $\mathtt{0}$, $\mathtt{1}$ und $\mathtt{2}$ entsprechen den Ziffern der berechneten Summe.

Unabhängig vom Zahlensystem kann bei der Addition zweier Zahlen immer nur ein Übertrag von höchstens 1 auftreten: Der maximale Wert einer Ziffer ist $n-1$, der maximale Übertrag der letzten Spalte ist 1. Die Summe daraus ergibt $(n-1)+(n-1)+1 = 2n-1$. Den Übertrag erhält man durch ganzzahlige Division durch n, also $(2n-1)/n = (n+(n-1))/n = 1+(n-1)/n = 1+0 = 1$.

(b)



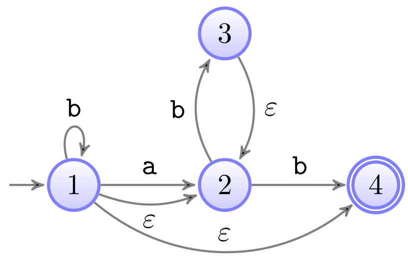
Im Prinzip sind wieder nur zwei Zustände für die beiden möglichen Überträge notwendig. Die Bedeutung ändert sich aber ein wenig: Der Übertrag ist jener, der von der noch ausständigen Rechnung erwartet wird. Demnach befindet sich der Transducer auch nur dann in einem Endzustand, wenn kein Übertrag für die letzte Ein-/Ausgabe erforderlich ist.

Zusätzlich benötigen wir einen Anfangszustand i, um führende Nuller bzw. einen führenden Einser (entstanden durch einen Übertrag) zu berücksichtigen.

Verglichen mit dem Mealy-Automaten von vorhin sind die Übergänge alle gespiegelt. Dadurch entsteht Indeterminismus; etwa kann im Zustand 0 die Eingabe $\genfrac{}{}{0pt}{}{\texttt{0}}{\texttt{0}}$ die Ausgabe `0` oder `1` generieren, je nachdem, ob von der weiteren Addition ein Übertrag erwartet wird (Wechsel in den Zustand 1) oder nicht (Verbleib im Zustand 0).

Aufgabe 4 (0.2 Punkte)

Sei $\mathcal{A}$ der folgende Automat:



Konstruieren Sie mit Hilfe des in der Vorlesung besprochenen Determinisierungsverfahrens einen deterministischen Automaten $\mathcal{A}'$, der äquivalent zu $\mathcal{A}$ ist.
Hinweis: Beginnen Sie mit einer Tabelle für die Werte $\delta^*(q, s)$, wobei $q \in \{1, 2, 3, 4\}$ und $s \in \{\texttt{a}, \texttt{b}\}$ gilt. Beachten Sie, dass man etwa vom Zustand 3 mit dem Symbol `b` in einem oder mehreren Schritten (wir berechnen ja δ^*) nicht nur in den Zustand 4, sondern auch zurück in den Zustand 3 und anschließend über die ε-Kante nach 2 gelangen kann, d.h., $\delta^*(3, \texttt{b}) = \{2, 3, 4\}$.

Lösung

Wir konstruieren als Zwischenschritt eine Tabelle mit den Werten von $\delta^*(q, s)$ für alle Zustände q und alle Symbole s. ε-Übergänge werden dadurch berücksichtigt, dass vor und nach einem Symbol beliebig viele Leerwörter auftreten können. Etwa sind vom Zustand 1 aus alle Zustände mit dem Symbol `b` erreichbar (d.h., $\delta^*(1, \texttt{b}) = \{1, 2, 3, 4\}$), obwohl der einzige `b`-Übergang bei 1 nur nach 1 führt:

$$\begin{array}{l}
1 \xrightarrow{\texttt{b}} 1 \\
1 \xrightarrow{\texttt{b}} 1 \xrightarrow{\varepsilon} 2 \quad \text{oder} \quad 1 \xrightarrow{\varepsilon} 2 \xrightarrow{\texttt{b}} 3 \xrightarrow{\varepsilon} 2 \\
1 \xrightarrow{\varepsilon} 2 \xrightarrow{\texttt{b}} 3 \\
1 \xrightarrow{\varepsilon} 2 \xrightarrow{\texttt{b}} 4
\end{array}$$

Aus der Tabelle für $\delta^*(q,s)$ ergibt sich jene für die Übergangsfunktion $\hat{\delta}$ des deterministischen Automaten durch Vereinigung der entsprechenden Zeilen.

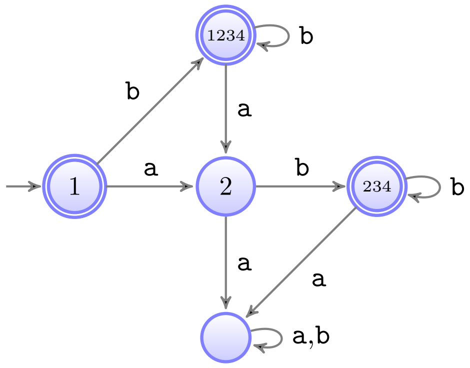
δ^*	a	b
1	$\{2\}$	$\{1,2,3,4\}$
2	$\{\}$	$\{2,3,4\}$
3	$\{\}$	$\{2,3,4\}$
4	$\{\}$	$\{\}$

$\hat{\delta}$	a	b
$\{1\}$	$\{2\}$	$\{1,2,3,4\}$
$\{2\}$	$\{\}$	$\{2,3,4\}$
$\{1,2,3,4\}$	$\{2\}$	$\{1,2,3,4\}$
$\{\}$	$\{\}$	$\{\}$
$\{2,3,4\}$	$\{\}$	$\{2,3,4\}$

Startzustand des deterministischen Automaten ist $\{1\}$, Endzustände sind alle Zustände, die den ursprünglichen Endzustand 4 enthalten. Außerdem kann man vom Startzustand aus mit dem Leerwort einen Endzustand erreichen, da $\delta^*(1,\varepsilon) = 4$ gilt, daher ist auch der Startzustand ein Endzustand. Der gesuchte deterministische Automat ist somit gegeben durch

$$\mathcal{A}' = \langle\ \{\{\},\{1\},\{2\},\{2,3,4\},\{1,2,3,4\}\},\ \{\mathtt{a},\mathtt{b}\},\ \hat{\delta},\ \{1\},\ \{\{1\},\{2,3,4\},\{1,2,3,4\}\}\ \rangle\ .$$

In graphischer Darstellung sieht der deterministische Automat wie folgt aus.



Aufgabe 5 (0.5 Punkte)

Vereinfachen Sie die folgenden Ausdrücke.

(a) $\{\mathtt{a},\mathtt{b},\mathtt{ab}\} \cup \{\}$

(b) $\{\mathtt{a},\mathtt{b},\mathtt{ab}\} \cdot \{\}$

(c) $\{\mathtt{a}\}^* \cdot (\{\mathtt{a}\} \cup \{\varepsilon\})$

(d) $(\{\mathtt{ab}\}^* \cdot \{\mathtt{a}\} \cdot \{\mathtt{ba}\}^* \cdot \{\mathtt{b}\}) \cdot \{\mathtt{a}\}^*$

(e) $((\{\mathtt{a},\mathtt{ab},\mathtt{cc}\} \cup \{\varepsilon\}) \cdot (\{\varepsilon,\mathtt{bb}\} \cup \{\mathtt{b}\})) \cup \{\mathtt{c}\}$

Lösung

(a) $\{\mathtt{a}, \mathtt{b}, \mathtt{ab}\} \cup \{\} = \{\mathtt{a}, \mathtt{b}, \mathtt{ab}\}$

(b) $\{\mathtt{a}, \mathtt{b}, \mathtt{ab}\} \cdot \{\} = \{\}$

(c) $\{\mathtt{a}\}^* \cdot (\{\mathtt{a}\} \cup \{\varepsilon\}) = \{\mathtt{a}\}^+ \cup \{\mathtt{a}\}^* = \{\mathtt{a}\}^*$

(d) $(\{\mathtt{ab}\}^* \cdot \{\mathtt{a}\} \cdot \{\mathtt{ba}\}^* \cdot \{\mathtt{b}\}) \cdot \{\mathtt{a}\}^*$
$= \{\mathtt{ab}\}^* \cdot \{\mathtt{ab}\}^* \cdot \{\mathtt{ab}\} \cdot \{\mathtt{a}\}^*$
$= \{\mathtt{ab}\}^* \cdot \{\mathtt{ab}\} \cdot \{\mathtt{a}\}^*$
$= \{\mathtt{ab}\}^+ \cdot \{\mathtt{a}\}^*$

(e) $((\{\mathtt{a}, \mathtt{ab}, \mathtt{cc}\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, \mathtt{bb}\} \cup \{\mathtt{b}\})) \cup \{\mathtt{c}\}$
$= (\{\varepsilon, \mathtt{a}, \mathtt{ab}, \mathtt{cc}\} \cdot \{\varepsilon, \mathtt{b}, \mathtt{bb}\}) \cup \{\mathtt{c}\}$
$= \{\varepsilon, \mathtt{a}, \mathtt{ab}, \mathtt{abb}, \mathtt{abbb}, \mathtt{b}, \mathtt{bb}, \mathtt{cc}, \mathtt{ccb}, \mathtt{ccbb}\} \cup \{\mathtt{c}\}$
$= \{\varepsilon, \mathtt{a}, \mathtt{ab}, \mathtt{abb}, \mathtt{abbb}, \mathtt{b}, \mathtt{bb}, \mathtt{c}, \mathtt{cc}, \mathtt{ccb}, \mathtt{ccbb}\}$

Aufgabe 6 **(0.2 Punkte)**

Geben Sie endliche Automaten an, die dieselbe Sprache beschreiben wie die folgenden regulären Ausdrücke in algebraischer Notation.

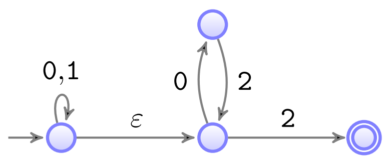
(a) $(\mathtt{0} + \mathtt{1})^*(\mathtt{02})^*\mathtt{2}$

(b) $(\mathtt{0} + \mathtt{1})^* + (\mathtt{02})^* + \mathtt{2}$

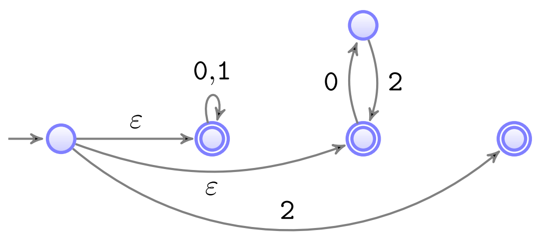
Lösung

Die gesuchten Automaten können mit dem allgemeinen Verfahren konstruiert werden, enthalten dann aber in der Regel viel mehr Zustände und ε-Kanten als notwendig. Bei den folgenden Automaten wurden die Automaten für die Teilausdrücke durch Hinschauen bzw. Erfahrung konstruiert (etwa für $(\mathtt{0} + \mathtt{1})^*$ und $(\mathtt{02})^*$) und erst diese mit dem allgemeinen Verfahren hintereinander bzw. parallel geschaltet.

(a)

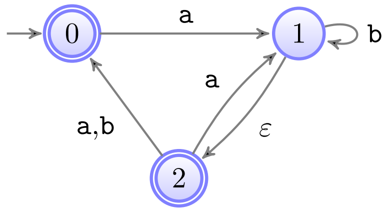


(b)



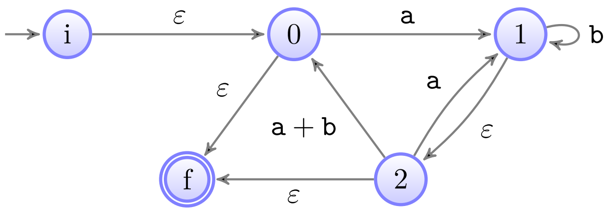
Aufgabe 7 (0.4 Punkte)

Konstruieren Sie zu folgendem endlichen Automaten einen regulären Ausdruck. Orientieren Sie sich am Algorithmus, der in der Vorlesung besprochen wurde und geben Sie den Automaten nach jeder Zustandselimination an!



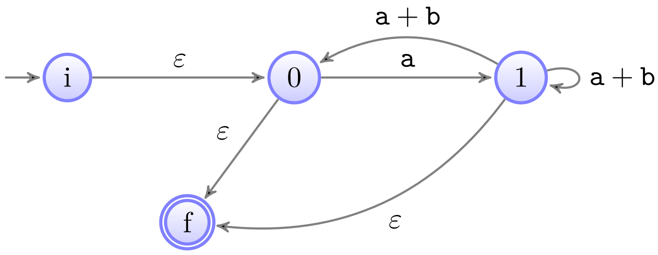
Lösung

Neuer Anfangs- und Endzustand:

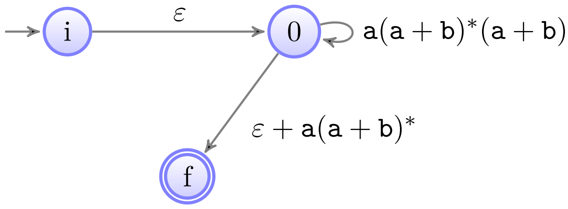


Wir eliminieren die Zustände in der Reihenfolge 2, 1 und 0; die anderen Reihenfolgen sind ebenfalls möglich.

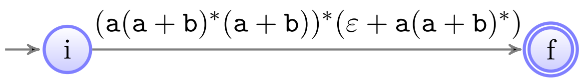
Elimination von Zustand 2:



Elimination von Zustand 1:



Elimination von Zustand 0:



Dieser Ausdruck ist bereits eine Lösung der Aufgabe, er lässt sich aber noch vereinfachen:

$$\begin{aligned}
&(\mathtt{a}(\mathtt{a}+\mathtt{b})^*(\mathtt{a}+\mathtt{b}))^*(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*)\\
&=(\mathtt{a}(\mathtt{a}+\mathtt{b})^+)^*(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*)\\
&=(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^+ + \mathtt{a}(\mathtt{a}+\mathtt{b})^+\mathtt{a}(\mathtt{a}+\mathtt{b})^+ + \cdots)(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*)\\
&=(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^+)(\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*)\\
&=\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^* + \mathtt{a}(\mathtt{a}+\mathtt{b})^+ + \mathtt{a}(\mathtt{a}+\mathtt{b})^+\mathtt{a}(\mathtt{a}+\mathtt{b})^*\\
&=\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*
\end{aligned}$$

Die Sprache des ursprünglichen Automaten wird also durch den Ausdruck $\varepsilon+\mathtt{a}(\mathtt{a}+\mathtt{b})^*$ beschrieben; sie ist die Menge aller Wörter über $\{\mathtt{a},\mathtt{b}\}$, die nicht mit dem Symbol `b` beginnen.

Aufgabe 8 (0.3 Punkte)

Beschreiben Sie E-Mail-Adressen des Providers gmx mit folgendem Aufbau. Vor dem @-Zeichen steht eine Mischung aus Buchstaben, Ziffern und Punkten, die immer mit einem Buchstaben beginnt. Unmittelbar vor und nach jedem Punkt muss mindestens ein Buchstabe oder eine Ziffer stehen. Die Punkte sind also nur zur Trennung der Buchstaben und Ziffern gedacht, mehrere Punkte nacheinander oder Punkte am Beginn oder Ende des vorderen Teils können nicht vorkommen. Nach dem @-Zeichen steht immer *gmx.at* als Domainangabe.

(a) Geben Sie einen regulären Ausdruck in algebraischer Notation an.

(b) Geben Sie einen regulären Ausdruck in `egrep`-Notation an. (Gesucht sind alle Zeilen, die *ausschließlich* eine derartige E-Mail-Adresse enthalten.)

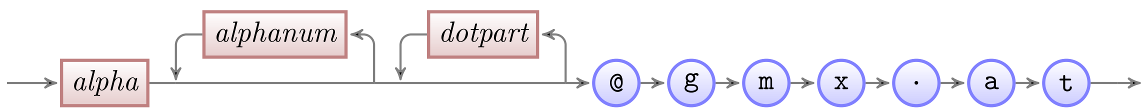
(c) Zeichnen Sie das Syntaxdiagramm, das Ihrem regulären Ausdruck aus Teil a entspricht.

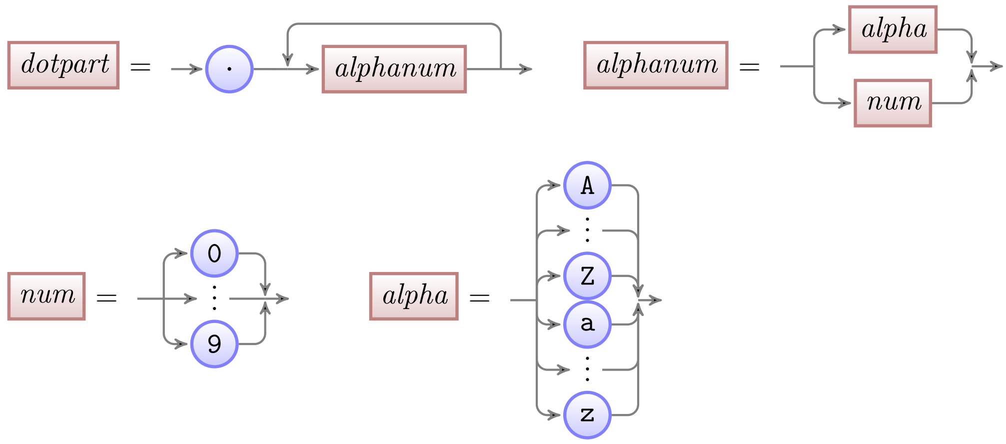
Lösung

(a) $\mathit{alpha}(\mathit{alpha}+\mathit{num})^*(\mathtt{.}(\mathit{alpha}+\mathit{num})(\mathit{alpha}+\mathit{num})^*)^*\mathtt{@gmx.at}$
mit den Abkürzungen
$\mathit{alpha} := \mathtt{A}+\cdots+\mathtt{Z}+\mathtt{a}+\cdots+\mathtt{z}$
$\mathit{num} := \mathtt{0}+\cdots+\mathtt{9}$

(b) `^[a-zA-Z][a-zA-Z0-9]*(\.[a-zA-Z0-9]+)*@gmx\.at$` oder
`^[:alpha:][:alnum:]*(\.[:alnum:]+)*@gmx\.at$`

(c) Syntaxdiagramm zum Ausdruck $\mathit{alpha}(\mathit{alpha}+\mathit{num})^*(\mathtt{.}(\mathit{alpha}+\mathit{num})(\mathit{alpha}+\mathit{num})^*)^*\mathtt{@gmx.at}$:





Aufgabe 9 (0.3 Punkte)

In einem wissenschaftlichen Artikel ist folgende Darstellung zu finden:

> Eine Maschine $\mathcal{M}$ wird beschrieben durch ein 5-Tupel $\langle P, b, E, A, S \rangle$, wobei P eine endliche Menge von Positionen ist, $b \in P$ die Beginnposition und $E \subseteq P$ die Endpositionen bezeichnen und A eine endliche Menge von Aktionen ist. Die Steuerungsfunktion $S \colon P \times P \mapsto 2^A$ gibt zu jedem Paar von Positionen all jene Aktionen an, mit denen die Maschine von der ersten Position in die zweite gelangen kann.
>
> Die erweiterte Steuerungsrelation $\hat{S} \subseteq P \times P \times A^*$ ist die kleinste Menge, für die gilt:
>
> - (p, p, ε) liegt in $\hat{S}$ (für alle $p \in P$).
> - (p_1, p_2, a) liegt in $\hat{S}$, falls $a \in S(p_1, p_2)$ gilt (für alle $p_1, p_2 \in P$ und $a \in A$).
> - Wenn (p_1, p_2, u) und (p_2, p_3, v) in $\hat{S}$ liegen, dann liegt auch (p_1, p_3, uv) darin.
>
> Eine Aktionsfolge $w \in A^*$ ist für die Maschine $\mathcal{M}$ zulässig, wenn es eine Endposition $e \in E$ gibt, sodass (b, e, w) in $\hat{S}$ liegt.

(a) Geben Sie ein konkretes Beispiel für eine Maschine mit 3 Positionen und 2 Aktionen an, die zwei Endpositionen besitzt. Beschreiben Sie die für Ihre Maschine zulässigen Aktionsfolgen.

(b) Beschreiben Sie, wie sich Maschinen als endliche Automaten darstellen lassen, wenn man die Menge der zulässigen Aktionsfolgen als die Sprache des Automaten betrachtet.

(c) Geben Sie einen endlichen Automaten an, der Ihrer Maschine aus der ersten Teilaufgabe entspricht.

Lösung

(a) Ein Beispiel für eine Maschine ist $\mathcal{M}_1 = \langle\{1,2,3\}, 1, \{2,3\}, \{\mathtt{a},\mathtt{b}\}, S_1\rangle$, wobei die Steuerungsfunktion S_1 definiert ist durch

S_1	1	2	3
1	$\{\mathtt{a}\}$	$\{\mathtt{b}\}$	$\{\}$
2	$\{\}$	$\{\mathtt{a}\}$	$\{\mathtt{b}\}$
3	$\{\mathtt{b}\}$	$\{\}$	$\{\mathtt{a}\}$

Die zulässigen Aktionsfolgen sind genau jene, bei denen die Anzahl der $\mathtt{b}$'s nicht durch drei teilbar ist.

Ein einfacheres Beispiel ist die triviale Maschine $\mathcal{M}_2 = \langle\{1,2,3\}, 1, \{2,3\}, \{\mathtt{a},\mathtt{b}\}, S_2\rangle$, wobei S_2 definiert ist durch $S_2(p_1,p_2) = \{\}$ für alle $p_1, p_2 \in P$. Diese Maschine besitzt keine zulässigen Aktionsfolgen.

(b) Die Positionen entsprechen den Zuständen, die Aktionen dem Eingabealphabet, die Anfangs- und Endpositionen den Anfangs- und Endzuständen und die Steuerungsfunktion der Übergangsfunktion bzw. -relation. Zu beachten ist allerdings, dass die Steuerungsfunktion zu jedem Positionspaar die möglichen Aktionen liefert, während die Übergangsfunktion bzw. -relation den Folgezustand zum momentanen Zustand und zum gelesenen Eingabesymbol angibt.

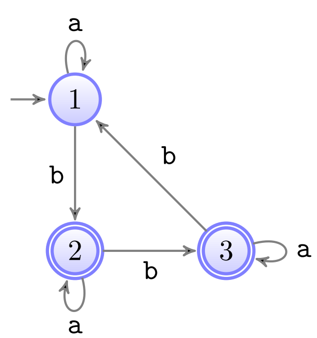
Der einer Maschine $\mathcal{M} = \langle P, b, E, A, S\rangle$ entsprechende (in der Regel indeterministische) Automat $\mathcal{A}_\mathcal{M}$ lässt sich beschreiben durch $\langle P, A, \delta, b, E\rangle$, wobei die Übergangsrelation δ für alle $p_1, p_2 \in P$ und alle $a \in A$ definiert ist durch

$$(p_1, a, p_2) \in \delta \quad \text{genau dann, wenn } a \in S(p_1, p_2).$$

(c) $\mathcal{A}_{\mathcal{M}_1} = \langle\{1,2,3\}, \{\mathtt{a},\mathtt{b}\}, \delta_1, 1, \{2,3\}\rangle$, wobei die Übergangsrelation definiert ist durch

δ_1	$\mathtt{a}$	$\mathtt{b}$
1	$\{1\}$	$\{2\}$
2	$\{2\}$	$\{3\}$
3	$\{3\}$	$\{1\}$

oder graphisch



$\mathcal{A}_{\mathcal{M}_2} = \langle\{1,2,3\}, \{\mathtt{a},\mathtt{b}\}, \delta_2, 1, \{2,3\}\rangle$, wobei $\delta_2 = \{\}$.

Aufgabe 10 (0.4 Punkte)

Songtexte enthalten gelegentlich Füllworte wie „Schubidubidu“ oder „Schubischubidubiduaaah“. Die Grammatik $G = \langle V, T, P, A \rangle$ erzeugt solche Füllwörter, wobei

$$\begin{aligned}
V &= \{A, B, C, D\} \\
T &= \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{h}, \mathtt{i}, \mathtt{s}, \mathtt{u}\} \\
P &= \{\, A \to B\, D \ , \\
&\qquad B \to \texttt{"schu"}\, C\, B\, \texttt{"du"}\, C \mid \varepsilon \\
&\qquad C \to \texttt{"bi"} \mid \texttt{"aaah"} \ , \\
&\qquad D \to \texttt{"du"} \mid \varepsilon \ \}
\end{aligned}$$

Überprüfen Sie für die nachfolgenden Wörter, ob sie in der von der Grammatik G spezifizierten Sprache $\mathcal{L}(G)$ liegen. Falls ja, geben Sie eine Ableitung an. Falls nein, argumentieren Sie, warum nicht.

(a) `schubidubidu`

(b) `schubischubidubidu`

(c) `schubischubidubiduaaah`

Überlegen Sie weiters:

(d) Ist es möglich, die Sprache $\mathcal{L}(G)$ auch durch einen endlichen Automaten zu beschreiben? Falls ja, geben Sie einen derartigen Automaten an. Fall nein, begründen Sie, warum das nicht geht.

Lösung

(a) Ja, das Wort liegt in der Sprache $\mathcal{L}(G)$.

$$\begin{aligned}
A &\Rightarrow B\, D \\
&\Rightarrow \texttt{"schu"}\, C\, B\, \texttt{"du"}\, C\, D \\
&\Rightarrow \texttt{"schubi"}\, B\, \texttt{"du"}\, C\, D \\
&\Rightarrow \texttt{"schubidu"}\, C\, D \\
&\Rightarrow \texttt{"schubidubi"}\, D \\
&\Rightarrow \texttt{"schubidubidu"}
\end{aligned}$$

(b) Das Wort liegt nicht in der Sprache $\mathcal{L}(G)$. Das lässt sich zum Beispiel über die Gesamtzahl der `bi` und `aaah` Silben argumentieren. Jedes Wort in $\mathcal{L}(G)$ hat eine gerade Anzahl davon, im Wort `schubischubidubidu` kommen aber 3 `bi`- und 0 `aaah`-Silben vor, also in Summe eine ungerade Anzahl.

Dass die Zahl der `bi`’s und `aaah`’s in Summe gerade sein muss, lässt sich folgendermaßen argumentieren. Diese beiden Silben können nur aus dem Nonterminal C erzeugt werden, welches wiederum nur durch Anwendung der Produktion

$B \to$ "schu" $C\,B$ "du" C eingeführt werden kann. Jede Anwendung dieser Produktion erzeugt aber zwei Vorkommnisse des Nonterminals C, die Anzahl muss daher gerade sein.

(c) Ja, das Wort liegt in der Sprache $\mathcal{L}(G)$:

$$\begin{aligned}
A &\Rightarrow B\,D \\
&\Rightarrow \texttt{"schu"}\,C\,B\,\texttt{"du"}\,C\,D \\
&\Rightarrow \texttt{"schu"}\,C\,\texttt{"schu"}\,C\,B\,\texttt{"du"}\,C\,\texttt{"du"}\,C\,D \\
&\Rightarrow \texttt{"schu"}\,C\,\texttt{"schu"}\,C\,\varepsilon\,\texttt{"du"}\,C\,\texttt{"du"}\,C\,D \\
&= \texttt{"schu"}\,C\,\texttt{"schu"}\,C\,\texttt{"du"}\,C\,\texttt{"du"}\,C\,D \\
&\Rightarrow \texttt{"schubischu"}\,C\,\texttt{"du"}\,C\,\texttt{"du"}\,C\,D \\
&\Rightarrow \texttt{"schubischubidu"}\,C\,\texttt{"du"}\,C\,D \\
&\Rightarrow \texttt{"schubischubidubidu"}\,C\,D \\
&\Rightarrow \texttt{"schubischubidubiduaaah"}\,D \\
&\Rightarrow \texttt{"schubischubidubiduaaah"}\,\varepsilon \\
&= \texttt{"schubischubidubiduaaah"}
\end{aligned}$$

(d) Nein, die Sprache $\mathcal{L}(G)$ kann nicht durch einen endlichen Automaten beschrieben werden, da sie nicht regulär ist. Das lässt sich über die `schu`'s und `du`'s argumentieren. Abgesehen vom optionalen Schluss-`du` (eingeführt durch das Nonterminal D) muss die Anzahl der beiden Silben übereinstimmen. Da die `schu`'s links von B und die `du`'s rechts davon erzeugt werden, können die Wörter der Sprache eine beliebig große Zahl von `schu`'s enthalten, ehe die `du`'s folgen. Ein endlicher Automat müsste sich die Zahl der `schu`'s in seinen Zuständen merken. Da diese Zahl nicht begrenzt werden kann, reicht eine endliche Zahl von Zuständen nicht aus.

Aufgabe 11 (0.3 Punkte)

Der Lambdakalkül (oder λ-Kalkül) ist eines der Modelle für Berechenbarkeit. Jede mit einem Computer berechenbare Funktion lässt sich durch einen Ausdruck im Lambdakalkül (einen sogenannten λ-Term) darstellen und berechnen.
Es gibt drei Arten von λ-Termen: Variablen, λ-Abstraktionen und Applikationen. Jede *Variable* ist bereits ein zulässiger λ-Term. Entscheiden Sie selbst, wie Variablen aussehen müssen, es sollen aber unendlich viele zur Auswahl stehen. Weiters können mittels λ-*Abstraktion* anonyme (d.h., unbenannte) Funktionen definiert werden, die eine Variable als Parameter besitzen; diese wird später bei der Abarbeitung im zugehörigen λ-Term ersetzt. λ-Abstraktionen werden mit dem Zeichen „λ“ eingeleitet, es folgt die Variable, dann ein Punkt und ein λ-Term. Eine *Applikation* steht für die Anwendung einer Funktion auf ein Argument und besteht aus zwei aufeinanderfolgenden λ-Termen, einem für die Funktion und einem für das Argument; getrennt werden die beiden Terme durch ein Leerzeichen. λ-Abstraktionen und Applikationen werden geklammert.

Beispiele zulässiger λ-Terme:

$(\lambda x.x)$	λ-Abstraktion
$(x\ y)$	Applikation, bei der x auf y angewendet wird.
$((x\ y)\ (\lambda z.z))$	Applikation, bei der das Ergebnis von $(x\ y)$ auf die Funktion $\lambda z.z$ angewendet wird.
$(\lambda x.(\lambda x.(x\ y)))$	λ-Term mit zwei λ-Abstraktionen und einer Applikation.

(a) Geben Sie eine induktive Definition für die Menge der λ-Terme an.

(b) Geben Sie eine kontextfreie Grammatik für die Menge der λ-Terme an. Verwenden Sie EBNF-Notationen, um die Grammatik übersichtlich zu halten.

(c) Zeigen Sie, dass der λ-Term $(\lambda x.(\lambda y.((x\ (\lambda z.y))\ z)))$ in der Sprache Ihrer Grammatik liegt.

Lösung

(a) Sei $\mathcal{V}$ die Menge der Variablen, etwa $\mathcal{V} = \{x, x_0, x_1, x_2, \ldots, y, y_0, y_1, \ldots, z, z_0, z_1, \ldots\}$. Dann lässt sich die Menge Λ der λ-Terme definieren als die kleinste Menge mit folgenden Eigenschaften:

- $\mathcal{V} \subseteq \Lambda$
- Wenn v eine Variable und l ein λ-Term ist, dann ist auch $(\lambda v\,.\,l)$ ein λ-Term.
  Anders ausgedrückt: Aus $v \in \mathcal{V}$ und $l \in \Lambda$ folgt $(\lambda v\,.\,l) \in \Lambda$.
- Wenn l_1 und l_2 λ-Terme sind, dann ist auch $(l_1\ l_2)$ ein λ-Term.
  Anders ausgedrückt: Aus $l_1, l_2 \in \Lambda$ folgt $(l_1\ l_2) \in \Lambda$.

(b) $\langle V, T, P, \mathit{Lambda}\rangle$ mit

$$\begin{aligned}
V &= \{\mathit{Lambda}, \mathit{Variable}, \mathit{Abstraktion}, \mathit{Anwendung}, \mathit{Buchstabe}, \mathit{Ziffer}\}\\
T &= \{\texttt{"0"}, \ldots, \texttt{"9"}, \texttt{"("}, \texttt{"}\lambda\texttt{"}, \texttt{"."}, \texttt{"␣"}, \texttt{")"}\}\\
P &= \{\begin{array}[t]{lcl}
\mathit{Lambda} & \to & \mathit{Variable} \mid \mathit{Abstraktion} \mid \mathit{Anwendung}\ ,\\
\mathit{Variable} & \to & \mathit{Buchstabe}\,\{\mathit{Ziffer}\}\ ,\\
\mathit{Abstraktion} & \to & \texttt{"("}\ \texttt{"}\lambda\texttt{"}\ \mathit{Variable}\ \texttt{"."}\ \mathit{Lambda}\ \texttt{")"}\ ,\\
\mathit{Anwendung} & \to & \texttt{"("}\ \mathit{Lambda}\ \texttt{"␣"}\ \mathit{Lambda}\ \texttt{")"}\ ,\\
\mathit{Buchstabe} & \to & \texttt{"x"} \mid \texttt{"y"} \mid \texttt{"z"}\ ,\\
\mathit{Ziffer} & \to & \texttt{"0"} \mid \cdots \mid \texttt{"9"}\ \}
\end{array}
\end{aligned}$$

Die hier gewählte Definition der Variablen orientiert sich an der vorigen Teilaufgabe; es gibt natürlich auch andere Möglichkeiten, eine unbeschränkte Menge von Variablennamen zu generieren.

(c) Um zu zeigen, dass der gegebene Term in der Sprache der Grammatik liegt, geben wir eine Parallelableitung für ihn an.

$$
\begin{aligned}
\mathit{Lambda} &\Rightarrow_p \mathit{Abstraktion} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{"}\ \mathit{Variable}\ \texttt{"."}\ \mathit{Lambda}\ \texttt{")"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x."}\ \mathit{Abstraktion}\ \texttt{")"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{"}\ \mathit{Variable}\ \texttt{"."}\ \mathit{Lambda}\ \texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y."}\ \mathit{Anwendung}\ \texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.("}\ \mathit{Lambda}\ \texttt{"␣"}\ \mathit{Lambda}\ \texttt{")))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.("}\ \mathit{Anwendung}\ \texttt{"␣"}\ \mathit{Variable}\ \texttt{")))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.(("}\ \mathit{Lambda}\ \texttt{"␣"}\ \mathit{Lambda}\ \texttt{")␣z)))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.(("}\ \mathit{Variable}\ \texttt{"␣"}\ \mathit{Abstraktion}\ \texttt{")␣z)))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.((x␣(}\lambda\texttt{"}\ \mathit{Variable}\ \texttt{"."}\ \mathit{Lambda}\ \texttt{"))␣z)))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.((x␣(}\lambda\texttt{z."}\ \mathit{Variable}\ \texttt{"))␣z)))"} \\
&\Rightarrow_p \texttt{"(}\lambda\texttt{x.(}\lambda\texttt{y.((x␣(}\lambda\texttt{z.y))␣z)))"}
\end{aligned}
$$

Aufgabe 12 **(0.3 Punkte)**

Wir betrachten die folgenden prädikatenlogischen Formeln:

$$
\begin{aligned}
F_1 &:= \neg\forall x\,(\mathit{Mensch}(x) \supset \forall y\,(\mathit{Mensch}(y) \supset \mathit{Kennt}(x,y))) \\
F_2 &:= \exists x\,\exists y\,(\mathit{Mensch}(x) \wedge \mathit{Mensch}(y) \wedge \neg\mathit{Kennt}(x,y)) \\
F_3 &:= \exists y\,\exists x\,\neg\forall y\,(\neg\mathit{Mensch}(x) \vee \exists z\,(\neg\mathit{Mensch}(y) \vee \mathit{Kennt}(y,x)))
\end{aligned}
$$

(a) Übersetzen Sie die Formeln F_1 und F_2 in natürliche Sprache, wobei $\mathit{Mensch}(x)$ für „x ist Mensch“ und $\mathit{Kennt}(x, y)$ für „x kennt y“ steht.

(b) Zeigen Sie mit Hilfe der algebraischen Gesetze für logische Operatoren und Quantoren, dass die beiden Formeln F_1 und F_2 äquivalent sind.

(c) Zeigen Sie mit Hilfe der algebraischen Gesetze für logische Operatoren und Quantoren, dass die beiden Formeln F_2 und F_3 äquivalent sind.

Lösung

(a) Formel F_1 entspricht der Aussage „Nicht alle Menschen kennen alle Menschen“ oder „Es ist nicht der Fall, dass jeder Mensch jeden Menschen kennt“.

Formel F_2 entspricht der Aussage „Es gibt Menschen, bei denen der eine den anderen nicht kennt“.

(b) $\neg\forall x\,(\mathit{Mensch}(x) \supset \forall y\,(\mathit{Mensch}(y) \supset \mathit{Kennt}(x,y)))$

$= \neg\forall x\,(\neg\mathit{Mensch}(x) \lor \forall y\,(\neg\mathit{Mensch}(y) \lor \mathit{Kennt}(x,y)))$	$F \supset G = \neg F \lor G$ (zweimal)
$= \neg\forall x\,\forall y\,(\neg\mathit{Mensch}(x) \lor \neg\mathit{Mensch}(y) \lor \mathit{Kennt}(x,y))$	$F \lor \forall z G = \forall z(F \lor G)$ wenn z nicht in F vorkommt
$= \exists x\,\exists y\,\neg(\neg\mathit{Mensch}(x) \lor \neg\mathit{Mensch}(y) \lor \mathit{Kennt}(x,y))$	$\neg\forall z F = \exists z \neg F$ (zweimal)
$= \exists x\,\exists y\,(\neg\neg\mathit{Mensch}(x) \land \neg\neg\mathit{Mensch}(y) \land \neg\mathit{Kennt}(x,y))$	$\neg(F \lor G) = \neg F \land \neg G$ (zweimal)
$= \exists x\,\exists y\,(\mathit{Mensch}(x) \land \mathit{Mensch}(y) \land \neg\mathit{Kennt}(x,y))$	$\neg\neg F = F$ (zweimal)

(c) $\exists y\,\exists x\,\neg\forall y\,(\neg\mathit{Mensch}(x) \lor \exists z\,(\neg\mathit{Mensch}(y) \lor \mathit{Kennt}(y,x)))$

$= \exists x\,\neg\forall y\,(\neg\mathit{Mensch}(x) \lor \neg\mathit{Mensch}(y) \lor \mathit{Kennt}(y,x))$	$\exists z F = F$ wenn z nicht frei in F vorkommt (zweimal)
$= \exists x\,\exists y\,\neg(\neg\mathit{Mensch}(x) \lor \neg\mathit{Mensch}(y) \lor \mathit{Kennt}(y,x))$	$\neg\forall z F = \exists z \neg F$
$= \exists x\,\exists y\,(\neg\neg\mathit{Mensch}(x) \land \neg\neg\mathit{Mensch}(y) \land \neg\mathit{Kennt}(y,x))$	$\neg(F \lor G) = \neg F \land \neg G$ (zweimal)
$= \exists x\,\exists y\,(\mathit{Mensch}(x) \land \mathit{Mensch}(y) \land \neg\mathit{Kennt}(y,x))$	$\neg\neg F = F$ (zweimal)
$= \exists x\,\exists z\,(\mathit{Mensch}(x) \land \mathit{Mensch}(z) \land \neg\mathit{Kennt}(z,x))$	Umbenennung quantifizierter Variable
$= \exists y\,\exists z\,(\mathit{Mensch}(y) \land \mathit{Mensch}(z) \land \neg\mathit{Kennt}(z,y))$	Umbenennung quantifizierter Variable
$= \exists y\,\exists x\,(\mathit{Mensch}(y) \land \mathit{Mensch}(x) \land \neg\mathit{Kennt}(x,y))$	Umbenennung quantifizierter Variable
$= \exists x\,\exists y\,(\mathit{Mensch}(y) \land \mathit{Mensch}(x) \land \neg\mathit{Kennt}(x,y))$	$\exists y \exists x F = \exists x \exists y F$
$= \exists x\,\exists y\,(\mathit{Mensch}(x) \land \mathit{Mensch}(y) \land \neg\mathit{Kennt}(x,y))$	$F \land G = G \land F$

Aufgabe 13 **(0.6 Punkte)**

Seien *Spielt*/2, *Vater*/1 und *Kind*/1 Prädikatensymbole sowie *albrecht* und *frieda* Konstantensymbole mit folgender Bedeutung:

$\mathit{Spielt}(x,y)$	… x spielt mit y	*albrecht*	… Albrecht
$\mathit{Vater}(x)$	… x ist ein Vater	*frieda*	… Frieda
$\mathit{Kind}(x)$	… x ist ein Kind		

Verwenden Sie diese Symbole, um die folgenden Sätze in prädikatenlogische Formeln zu übersetzen.

(a) Alle Väter, die mit Albrecht spielen, spielen auch mit Frieda.

(b) Manche Kinder spielen mit allen Vätern.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Albrecht}, \text{Bogdan}, \text{Erich}, \text{Frieda}, \text{Kathrin}, \text{Nina}, \text{Tamara}\}\\
I(\mathit{Vater}) &= \{\text{Bogdan}, \text{Erich}\}\\
I(\mathit{Kind}) &= \{\text{Albrecht}, \text{Frieda}, \text{Nina}\}\\
I(\mathit{Spielt}) &= \{(\text{Albrecht}, \text{Frieda}), (\text{Bogdan}, \text{Albrecht}), (\text{Erich}, \text{Albrecht}), (\text{Erich}, \text{Frieda}),\\
&\qquad (\text{Frieda}, \text{Nina}), (\text{Frieda}, \text{Kathrin}), (\text{Frieda}, \text{Albrecht}), (\text{Nina}, \text{Frieda}),\\
&\qquad (\text{Tamara}, \text{Erich}), (\text{Tamara}, \text{Bogdan}), (\text{Tamara}, \text{Albrecht})\},\\
I(\mathit{albrecht}) &= \text{Albrecht} \qquad I(\mathit{frieda}) = \text{Frieda}
\end{aligned}$$

Übersetzen Sie die nachfolgenden Formeln in natürliche Sprache. Geben Sie an, ob die Formeln in der Interpretation I wahr oder falsch sind. Begründen Sie Ihre Antwort; es ist keine formale Auswertung erforderlich.

(c) $\forall x\,(\mathit{Vater}(x) \supset \exists y\,(\mathit{Kind}(y) \wedge \mathit{Spielt}(x,y)))$

(d) $\forall x\,(\mathit{Vater}(x) \wedge \mathit{Spielt}(x, \mathit{albrecht}))$

(e) $\exists x\,(\mathit{Kind}(x) \wedge \forall y\,(\mathit{Kind}(y) \supset \mathit{Spielt}(x,y)))$

(f) $\forall x\,(\mathit{Spielt}(x, \mathit{frieda}) \not\equiv \mathit{Spielt}(x, \mathit{albrecht}))$

Lösung

(a) $\forall x\,((\mathit{Vater}(x) \wedge \mathit{Spielt}(x, \mathit{albrecht})) \supset \mathit{Spielt}(x, \mathit{frieda}))$

(b) $\exists x\,(\mathit{Kind}(x) \wedge \forall y\,(\mathit{Vater}(y) \supset \mathit{Spielt}(x,y)))$ oder
$\exists x \forall y\,(\mathit{Kind}(x) \wedge (\mathit{Vater}(y) \supset \mathit{Spielt}(x,y)))$

(c) Übersetzung: Alle Väter spielen mit (mindestens) einem Kind.
Diese Aussage ist wahr in I, da Vater Bogdan mit Kind Albrecht und Vater Erich mit Kind Frieda spielt.

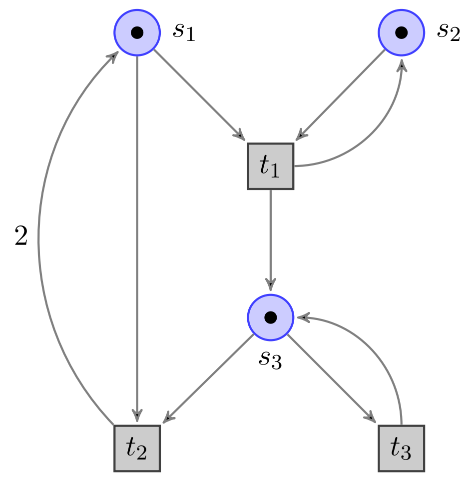
(d) Übersetzung: Alle sind Väter und spielen mit Albrecht.
Diese Aussage ist falsch, da etwa Albrecht keine Vater ist.

(e) Übersetzung: Es gibt (mindestens) ein Kind, das mit allen Kindern spielt.
Diese Aussage ist falsch in I, da es die drei Kinder Albrecht, Frieda und Nina gibt und gemäß $I(\mathit{Spielt})$ Albrecht nicht mit Nina, Frieda nicht mit Frieda und Nina nicht mit Albrecht spielt.

(f) Übersetzung: Alle spielen entweder mit Frieda oder Albrecht (aber nicht mit beiden).
Diese Aussage ist falsch in I, da etwa Erich sowohl mit Frieda als auch mit Albrecht spielt.

Aufgabe 14 (0.3 Punkte)

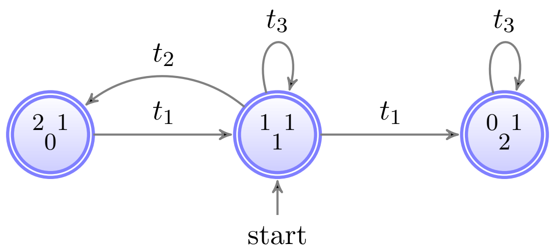
Gegeben sei das folgende Petri-Netz mit Anfangsmarkierung.



Beschreiben Sie alle möglichen Reihenfolgen, in der die Transitionen feuern und die Markierungen auftreten können. Am einfachsten lässt sich das durchführen, wenn Sie die Namen der Transitionen als Alphabet und die Markierungen als Zustände auffassen und einen Automaten bzw. einen regulären Ausdruck verwenden.

Lösung

Die Abläufe im Petrinetz lassen sich durch einen Automaten mit (in diesem Fall) endlich vielen Zuständen beschreiben. Die Markierungen bilden die Zustände, die Transitionen das Alphabet. Erhält man aus einer Markierung m durch Feuern einer Transition t eine Markierung m', dann gibt es einen Übergang beschriftet mit t vom Zustand für m zu jenem für m'.
Wir stellen jede Markierung durch drei Zahlen ${}^{n_1}{}_{n_3}{}^{n_2}$ dar, wobei n_i die Anzahl der Marken in der Stelle s_i angibt. Die endlichen Reihenfolgen, in denen die Transitionen feuern können, entsprechen der Sprache des folgenden Automaten.



Diese Reihenfolgen lassen sich auch durch den folgenden regulären Ausdruck in algebraischer Notation beschreiben:

$$(t_3 + t_2t_1)^*(\varepsilon + t_2 + t_1t_3^*)$$

Aufgabe 15 (0.3 Punkte)

Eisenbahnstrecken werden aus Sicherheitsgründen in Abschnitte unterteilt, in denen sich jeweils höchstens ein Zug befinden darf. Jeder Abschnitt wird durch eine Ampel gesichert, die nur dann grün ist, wenn der Abschnitt frei ist. Ein Zug darf nur dann in den nächsten Abschnitt einfahren, wenn die Ampel grün ist.

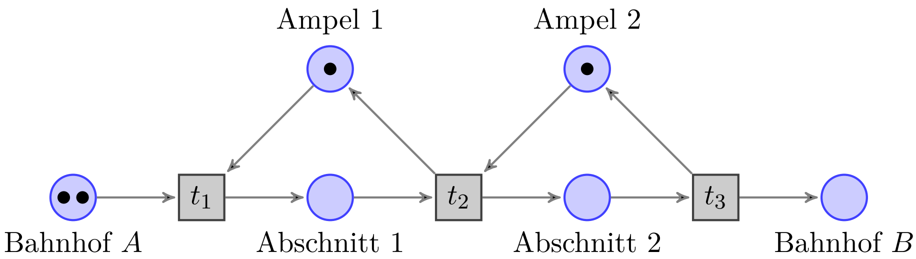
(a) Geben Sie ein Petri-Netz an, das eine Eisenbahnstrecke zwischen zwei Bahnhöfen A und B modelliert, die in zwei solche Abschnitte unterteilt ist. Nehmen Sie an, dass sich zu Beginn zwei Zügen im Bahnhof A befinden.

Erklären Sie die Bedeutung der Stellen und der Transitionen Ihres Petri-Netzes.

(b) Geben Sie alle möglichen Markierungen an, die von der Startmarkierung aus erreichbar sind, sowie ihre Reihenfolge. (Verwenden Sie eine geeignete Kurznotation für die Markierungen; es ist nicht notwendig, das vollständige Petri-Netz zu kopieren.)

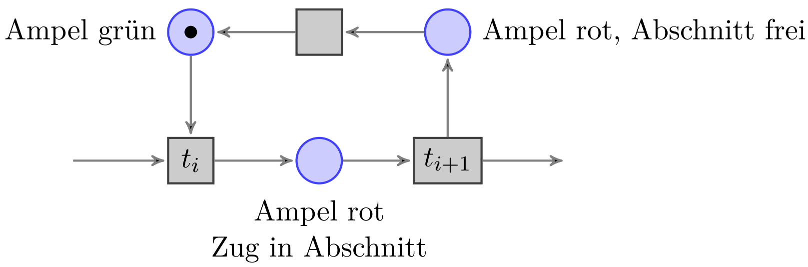
Lösung

(a)

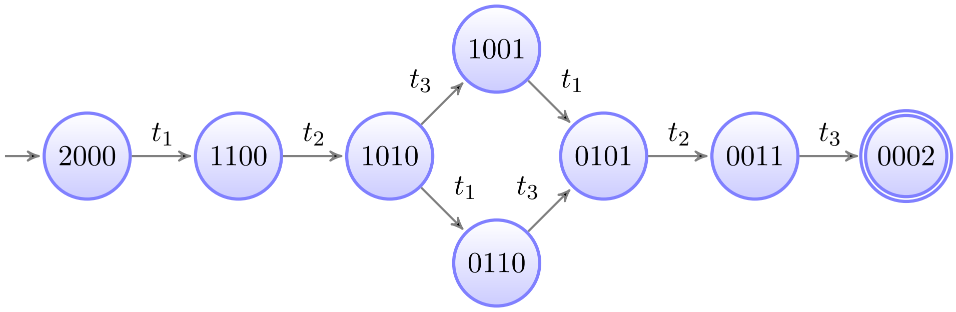


Eine Markierung in den Stellen „Bahnhof A/B" bzw. „Abschnitt 1/2" bedeutet, dass sich ein Zug im jeweiligen Bahnhof bzw. Abschnitt befindet. Eine Markierung in den Stellen „Ampel 1/2" bedeutet, dass die entsprechende Ampel auf grün steht; andernfalls steht sie auf rot. Transition t_1 kann feuern, wenn sich ein Zug in Bahnhof A befindet und Ampel 1 grün ist; danach ist Ampel 1 rot und der Zug befindet sich in Abschnitt 1. Transition t_2 kann feuern, wenn sich ein Zug in Abschnitt 1 befindet und Ampel 2 grün ist; danach befindet sich der Zug in Abschnitt 2, Abschnitt 1 ist wieder frei (und Ampel 1 ist wieder grün), und Ampel 2 ist rot. Transition t_3 kann feuern, wenn sich ein Zug in Abschnitt 2 befindet; danach befindet sich der Zug in Bahnhof B und Abschnitt 2 ist wieder frei (und Ampel 2 ist wieder grün).

Die einzelnen Abschnitte können auch aufwändiger modelliert werden. Soll die Ampel auch rot sein können, ohne dass sich ein Zug im Abschnitt befindet, könnte jeder Abschnitt so aussehen:



(b) Die Markierungen im ersten, einfacheren Petri-Netz können durch Tupel aa_1a_2b aus vier Zahlen beschrieben werden, wobei a bzw. b die Zahl der Markierungen in der Stelle Bahnhof A bzw. B und a_1 bzw. a_2 die Zahl der Markierungen in der Stelle Abschnitt 1 bzw. 2 angibt. Die Zahl der Markierungen in den Ampel-Stellen müssen wir nicht gesondert angeben, da sie eindeutig durch die Markierungen in Abschnitt 1 bzw. 2 festgelegt sind. Wir erhalten folgenden Automaten für die Abläufe:



Wenn wir den Zustand 0002 als Endzustand markieren, entspricht die Sprache des Automaten all jenen Transitionsfolgen, die in jene Markierung führen, in der beide Züge im Bahnhof B angelangt sind. Das sind in diesem Fall nur zwei Wörter, nämlich $t_1t_2t_3t_1t_2t_3$ und $t_1t_2t_1t_3t_2t_3$.