2. Übungsblatt (mit Lösungen)

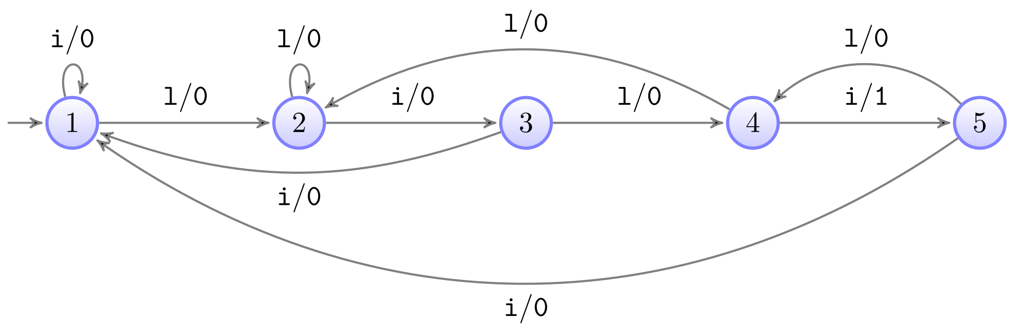
3.0 VU Formale Modellierung

Gerald Berger

27. April 2016

Aufgabe 1 (0.4 Punkte)

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



(a) Geben Sie die Ausgabe zur Eingabe `iililiililili` an.

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

(c) Finden Sie einen zu $\mathcal{A}$ äquivalenten Automaten $\mathcal{A}'$ mit vier Zuständen. Beschreiben Sie $\mathcal{A}'$ zusätzlich als 6-Tupel; legen Sie die Übergangsfunktion δ sowie die Ausgabefunktion γ durch eine Tabelle fest.

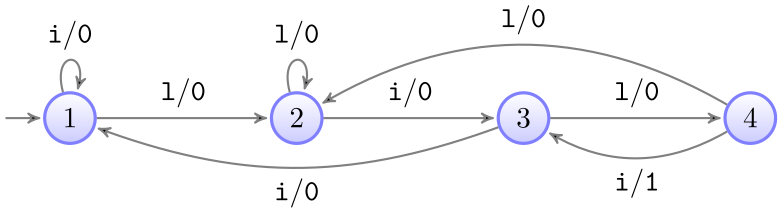
(d) Berechnen Sie schrittweise $\delta^*(1, \texttt{lilil})$ und $\gamma^*(1, \texttt{lilil})$.

Lösung

(a) $\gamma^*(1, \texttt{iililiililili}) = \texttt{0000010000101}$

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

(c) Sei $\mathcal{A}'$ der folgende Mealy-Automat:



$\mathcal{A}' = \langle\{1,2,3,4\}, \{\mathtt{l},\mathtt{i}\}, \{\mathtt{0},\mathtt{1}\}, \delta, \gamma, 1\rangle$, wobei δ und γ durch die folgenden Tabellen festgelegt werden.

δ	l	i
1	2	1
2	2	3
3	4	1
4	2	3

γ	l	i
1	0	0
2	0	0
3	0	0
4	0	1

(d)

$$\begin{aligned}
\delta^*(1,\mathtt{lilil}) &= \delta^*(\delta(1,\mathtt{l}),\mathtt{ilil}) \\
&= \delta^*(2,\mathtt{ilil}) \\
&= \delta^*(\delta(2,\mathtt{i}),\mathtt{lil}) \\
&= \delta^*(3,\mathtt{lil}) \\
&= \delta^*(\delta(3,\mathtt{l}),\mathtt{il}) \\
&= \delta^*(4,\mathtt{il}) \\
&= \delta^*(\delta(4,\mathtt{i}),\mathtt{l}) \\
&= \delta^*(5,\mathtt{l}) \\
&= \delta^*(\delta(5,\mathtt{l}),\varepsilon) \\
&= \delta^*(4,\varepsilon) \\
&= 4
\end{aligned}$$

$$\begin{aligned}
\gamma^*(1,\mathtt{lilil}) &= \gamma(1,\mathtt{l})\cdot\gamma^*(\delta(1,\mathtt{l}),\mathtt{ilil}) \\
&= \mathtt{0}\cdot\gamma^*(2,\mathtt{ilil}) \\
&= \mathtt{0}\cdot\gamma(2,\mathtt{i})\cdot\gamma^*(\delta(2,\mathtt{i}),\mathtt{lil}) \\
&= \mathtt{0}\cdot\mathtt{0}\cdot\gamma^*(3,\mathtt{lil}) \\
&= \mathtt{00}\cdot\gamma(3,\mathtt{l})\cdot\gamma^*(\delta(3,\mathtt{l}),\mathtt{il}) \\
&= \mathtt{00}\cdot\mathtt{0}\cdot\gamma^*(4,\mathtt{il}) \\
&= \mathtt{000}\cdot\gamma(4,\mathtt{i})\cdot\gamma^*(\delta(4,\mathtt{i}),\mathtt{l}) \\
&= \mathtt{000}\cdot\mathtt{1}\cdot\gamma^*(5,\mathtt{l}) \\
&= \mathtt{0001}\cdot\gamma(5,\mathtt{l})\cdot\gamma^*(\delta(5,\mathtt{l}),\varepsilon) \\
&= \mathtt{0001}\cdot\mathtt{0}\cdot\gamma^*(4,\varepsilon) \\
&= \mathtt{00010}\cdot\varepsilon = \mathtt{00010}
\end{aligned}$$

Aufgabe 2 **(0.3 Punkte)**

Sei L die Sprache

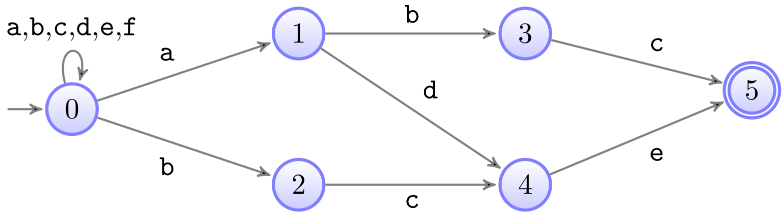
$$\{\, w \in \{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},\mathtt{e},\mathtt{f}\}^* \mid w \text{ endet mit der Zeichenfolge } \mathtt{abc}, \mathtt{bce} \text{ oder } \mathtt{ade} \,\} \ .$$

Geben Sie einen *deterministischen* Automaten für L an. Gehen Sie folgendermaßen vor:

1. Konstruieren Sie einen indeterministischen Automaten für diese Sprache.
2. Wandeln Sie den indeterministischen Automaten mit Hilfe des in der Vorlesung besprochenen Verfahrens in einen deterministischen um.

Lösung

Wir konstruieren zunächst auf möglichst direktem Weg einen beliebigen Automaten für die gesuchte Sprache. Dieser ist im Zustand 0 indeterministisch: Für das Symbol `a` und für das Symbol `b` gibt es jeweils zwei mögliche Folgezustände. Zusätzlich zur graphischen Darstellung geben wir die Übergangsfunktion auch als Tabelle an, da diese bei der Determinisierung hilft.



δ^*	a	b	c	d	e	f
0	$\{0,1\}$	$\{0,2\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
1	$\{\}$	$\{3\}$	$\{\}$	$\{4\}$	$\{\}$	$\{\}$
2	$\{\}$	$\{\}$	$\{4\}$	$\{\}$	$\{\}$	$\{\}$
3	$\{\}$	$\{\}$	$\{5\}$	$\{\}$	$\{\}$	$\{\}$
4	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{5\}$	$\{\}$
5	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$

Einen deterministischen Automaten erhalten wir, indem wir den indeterminischen Automaten simulieren. Ein Zustand des deterministischen Automaten repräsentiert dabei jene Zustände des indeterministischen, in denen sich dieser zu diesem Zeitpunkt befinden kann. Der Startzustand wird mit $\{0\}$ bezeichnet, da sich der indeterministische Automat zu Beginn im Zustand 0 (und nur in diesem) befindet. Von diesem Zustand ausgehend erstellen wir zeilenweise die Tabelle für die Übergangsfunktion des deterministischen Automaten.

$\hat{\delta}$	a	b	c	d	e	f
$\{0\}$	$\{0,1\}$	$\{0,2\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
$\{0,1\}$	$\{0,1\}$	$\{0,2,3\}$	$\{0\}$	$\{0,4\}$	$\{0\}$	$\{0\}$
$\{0,2\}$	$\{0,1\}$	$\{0,2\}$	$\{0,4\}$	$\{0\}$	$\{0\}$	$\{0\}$
$\{0,2,3\}$	$\{0,1\}$	$\{0,2\}$	$\{0,4,5\}$	$\{0\}$	$\{0\}$	$\{0\}$
$\{0,4\}$	$\{0,1\}$	$\{0,2\}$	$\{0\}$	$\{0\}$	$\{0,5\}$	$\{0\}$
$\{0,4,5\}$	$\{0,1\}$	$\{0,2\}$	$\{0\}$	$\{0\}$	$\{0,5\}$	$\{0\}$
$\{0,5\}$	$\{0,1\}$	$\{0,2\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$

Jene Zustände, die einer Situation entsprechen, in der der indeterministische Automat einen Endzustand erreicht hat, sind die Endzustände des deterministischen Automaten; in diesem Beispiel sind das alle Zustände, deren Bezeichnung 5 enthält. Dieser wird somit durch das Tupel $\langle \widehat{Q}, \Sigma, \hat{\delta}, \{0\}, \widehat{F} \rangle$ beschrieben, wobei

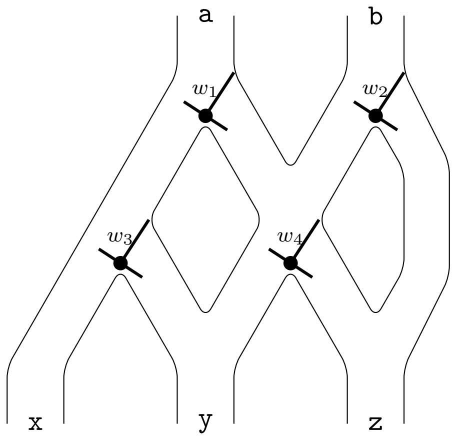
$$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$$

$$\widehat{F} = \{\{0,4,5\}, \{0,5\}\}$$

$$\widehat{Q} = \{\{0\}, \{0,1\}, \{0,2\}, \{0,2,3\}, \{0,4\}\} \cup \widehat{F}$$

Aufgabe 3 (0.4 Punkte)

Max erhält zum Geburtstag ein Knobelspiel, das aus einem Würfel und einer Stahlkugel besteht. Der Würfel besitzt oben zwei Löcher (bezeichnet mit `a` und `b`) und unten drei (bezeichnet mit `x`, `y` und `z`). Wirft man die Kugel bei einem der beiden oberen Löcher hinein, kommt sie bei einem der unteren drei wieder heraus. Die Aufgabe besteht nun darin das Loch vorherzusagen, bei dem die Kugel erscheinen wird. Um die Sache schwieriger zu gestalten, sind die Eingänge nicht fest mit den Ausgängen verbunden, sondern werden durch Weichen umgeleitet, die sich mit jeder vorbeikommenden Kugel verstellen. Nach einiger Zeit verliert Max die Geduld und zerlegt den Würfel. Er findet vier Weichen (w_1 bis w_4) vor, die folgendermaßen angeordnet sind:



Bei der momentanen Weichenstellung werden die Kugeln nach links geleitet, sodass die Kugel vom Eingang `a` zum Ausgang `x` bzw. vom Eingang `b` zum Ausgang `y` rollen würde. Dabei würden die Weichen w_1 und w_3 bzw. w_2 und w_4 umgeschaltet werden. Wirft man etwa die Kugel viermal in den Eingang `a`, kommt sie zuerst bei Ausgang `x`, dann zweimal bei Ausgang `y` und schließlich bei Ausgang `z` zum Vorschein; die Weichen befinden sich danach wieder in der Ausgangsstellung.
Modellieren Sie das Verhalten dieses Spiels mit Hilfe eines Moore- oder Mealy-Automaten. Bei Eingabe eines Wortes über $\{\mathtt{a}, \mathtt{b}\}$ soll der Automat jene Ausgänge liefern, bei denen die Kugel erscheinen würde. Sie können die Übergangs- und Ausgabefunktion graphisch oder als Tabelle darstellen.
Berechnen Sie $\gamma^*(q_0, \mathtt{aabaabb})$, wobei γ die Ausgabefunktion und q_0 den Anfangszustand Ihres Automaten bezeichnet.

Lösung

Der Zustand des Würfels wird durch die Stellung der vier Weichen festgelegt. Wir bezeichnen die Zustände mit $w_1w_2w_3w_4$, wobei $w_i = 0$ bedeutet, dass die Weiche rechts steht (wie in der Skizze eingezeichnet), und $w_i = 1$, dass sich die Weiche in der linken Stellung befindet. Das Spiel wird durch den Mealy-Automaten

$$\langle\{0000, 0001, \ldots, 1111\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}, \delta, \gamma, 0000\rangle$$

beschrieben, wobei die Übergangsfunktion δ und die Ausgabefunktion γ durch folgende Tabelle definiert sind.

	δ		γ	
	a	b	a	b
0000	1010	0101	x	y
0001	1011	0100	x	z
0010	1000	0111	y	y
0011	1001	0110	y	z
0100	1110	0000	x	z
0101	1111	0001	x	z
0110	1100	0010	y	z
0111	1101	0011	y	z
1000	0001	1101	y	y
1001	0000	1100	z	z
1010	0011	1111	y	y
1011	0010	1110	z	z
1100	0101	1000	y	z
1101	0100	1001	z	z
1110	0111	1010	y	z
1111	0110	1011	z	z

Für die Eingabe aabaabb erhalten wir $\gamma^*(0000, \texttt{aabaabb}) = \texttt{xyzyyzz}$.

Aufgabe 4 **(0.2 Punkte)**

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

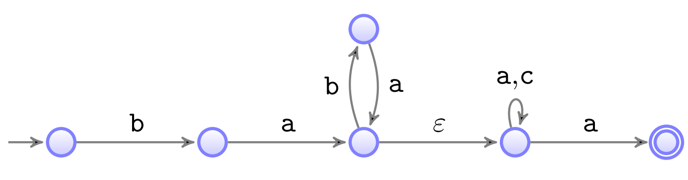
(a) $(\texttt{ba})^+(\texttt{a} + \texttt{c})^*\texttt{a}$

(b) $(\texttt{a} + \texttt{c})^*\texttt{b}(\texttt{ab})^+ + \texttt{c}$

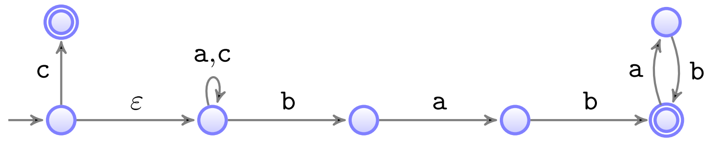
Lösung

Die gesuchten Automaten können mit dem allgemeinen Verfahren konstruiert werden, enthalten dann aber in der Regel viel mehr Zustände und ε-Kanten als notwendig. Die folgenden Automaten wurden „ad-hoc“ konstruiert.

(a)



(b)



Aufgabe 5 **(0.3 Punkte)**

Seien L_1 und L_2 Sprachen. Geben Sie für die folgenden Gleichungen/Aussagen an, ob sie gelten oder nicht. Falls die Gleichung/Aussage gilt, geben Sie eine Begründung an. Gilt die Gleichung/Aussage nicht, so geben Sie ein Gegenbeispiel an.

(a) $L_1 \cdot L_2 = L_2 \cdot L_1$

(b) $L_1^* \cdot \{\varepsilon\} = \{\}$

(c) $\varepsilon \in L_1 \cdot L_2$ genau dann wenn $\varepsilon \in L_1$ und $\varepsilon \in L_2$.

(d) Ist L_1 regulär, so ist jede Sprache $L \subseteq L_1$ auch regulär.

(e) Ist die Menge L_1 endlich, so ist L_1 regulär.

Lösung

(a) Die Behauptung gilt nicht. Seien $L_1 = \{\mathtt{a}\}$ und $L_2 = \{\mathtt{b}\}$. Dann gilt $L_1 \cdot L_2 = \{\mathtt{ab}\}$, jedoch $L_2 \cdot L_1 = \{\mathtt{ba}\}$.

(b) Die Behauptung gilt nicht. Es gilt ja $L \cdot \{\varepsilon\} = L$ für jede Sprache L. Dies ist unmittelbar aus der Definition der Konkatenation von Sprachen einsichtig. Daher ist $L_1^* \cdot \{\varepsilon\} = L_1^*$. Diese Sprache ist jedoch immer verschieden von $\{\}$, da sie mindestens das Leerwort ε enthält.

(c) Die Behauptung gilt. Wenn nun $\varepsilon \in L_1 \cdot L_2$, dann gibt es Wörter $a \in L_1$ und $b \in L_2$, sodass $a \cdot b = \varepsilon$. Die Konkatenation $a \cdot b$ kann jedoch nur dann das Leerwort ergeben, wenn $a = \varepsilon$ und $b = \varepsilon$. Somit gilt $\varepsilon \in L_1$ und $\varepsilon \in L_2$ wie gefordert.

(d) Die Behauptung gilt nicht. So ist z.B. $L_1 = \{a, b\}^*$ regulär, jedoch die Sprache $L := \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ nicht. Offensichtlich gilt jedoch $L \subseteq L_1$, womit die Behauptung widerlegt ist.

(e) Die Behauptung gilt. Ist L_1 endlich, so gilt ja $L_1 = \{w_1, \dots, w_n\}$ für gewisse Wörter $w_1, \dots, w_n$. Man sieht nun leicht ein, dass jede Sprache $\{w_k\}$ $(k = 1, \dots, n)$ regulär ist, da eine solche Sprache aus der Konkatentation von endlich vielen Symbolen entsteht. Damit ist $L_1 = \{w_1\} \cup \cdots \cup \{w_n\}$, also eine endliche Vereinigung von regulären Sprachen. Somit ist auch L_1 regulär.

Aufgabe 6 (0.3 Punkte)

Vereinfachen Sie die folgenden Ausdrücke.

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

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

(c) $(\{\mathtt{b}\}^* \cdot \{\mathtt{b}\}^+) \cdot \{\mathtt{b}\}^*$

(d) $\{\mathtt{ac}, \mathtt{b}, \mathtt{aba}\} \cdot \{\}$

(e) $\{\mathtt{ad}, \mathtt{bc}, \mathtt{ab}\} \cup \{\}^*$

Lösung

(a) $$\begin{aligned} L_1 &= (((\{\mathtt{a}, \mathtt{ab}\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, \mathtt{bb}, \mathtt{cc}\} \cup \{\mathtt{b}\})) \cup \{\mathtt{a}, \mathtt{c}\} \\ &= \{\mathtt{a}, \mathtt{ab}, \varepsilon\} \cdot \{\varepsilon, \mathtt{bb}, \mathtt{cc}, \mathtt{b}\} \cup \{\mathtt{a}, \mathtt{c}\} \\ &= \{\varepsilon, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{ab}, \mathtt{abb}, \mathtt{acc}, \mathtt{abbb}, \mathtt{abcc}, \mathtt{bb}, \mathtt{cc}\} \end{aligned}$$

(b) $L_2 = \{\mathtt{a}\}^* \cdot (\{\mathtt{a}\} \cup \{\varepsilon\}) = \{\mathtt{a}\}^* \cdot \{\mathtt{a}, \varepsilon\} = \{\mathtt{a}\}^* \cdot \{\mathtt{a}\} \cup \{\mathtt{a}\}^* \cdot \{\varepsilon\} = \{\mathtt{a}\}^+ \cup \{\mathtt{a}\}^* = \{\mathtt{a}\}^*$

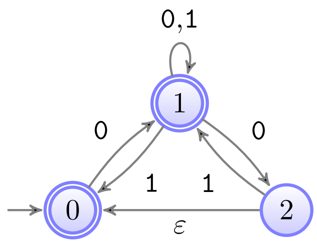
(c) $L_3 = (\{\mathtt{b}\}^* \cdot \{\mathtt{b}\}^+) \cdot \{\mathtt{b}\}^* = \{\mathtt{b}\}^+ \cdot \{\mathtt{b}\}^* = \{\mathtt{b}\}^+$

(d) $L_4 = \{\mathtt{ac}, \mathtt{b}, \mathtt{aba}\} \cdot \{\} = \{\}$

(e) $L_5 = \{\mathtt{ad}, \mathtt{bc}, \mathtt{ab}\} \cup \{\}^* = \{\mathtt{ad}, \mathtt{bc}, \mathtt{ab}\} \cup \{\varepsilon\} = \{\mathtt{ad}, \mathtt{bc}, \mathtt{ab}, \varepsilon\}$

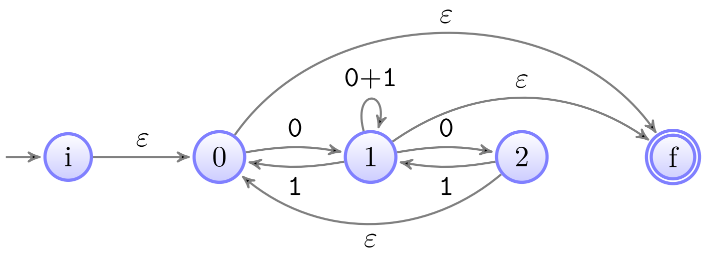
Aufgabe 7 (0.3 Punkte)

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!



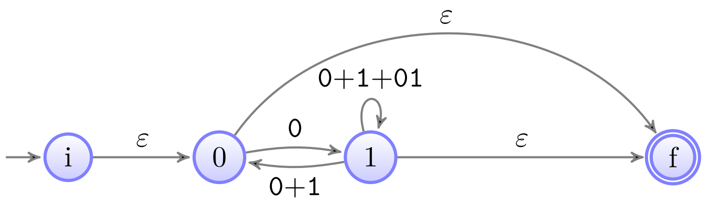
Lösung

Neuer Anfangs- und Endzustand:

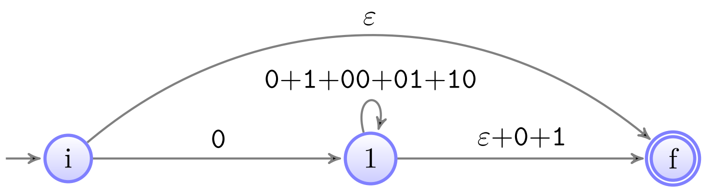


Wir eliminieren die Zustände in der Reihenfolge 2, 0 und 1; andere Reihenfolgen sind ebenfalls möglich.

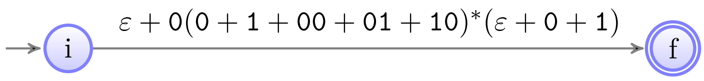
Elimination von Zustand 2:



Elimination von Zustand 0:



Elimination von Zustand 1:



Da $(\mathtt{0}{+}\mathtt{1})^*$ bereits alle möglichen Wörter über $\{\mathtt{0},\mathtt{1}\}$ enthält, gilt $(\mathtt{0}{+}\mathtt{1}{+}\cdots)^* = (\mathtt{0}{+}\mathtt{1})^*$, der reguläre Ausdruck vereinfacht sich damit zu $\varepsilon + \mathtt{0}(\mathtt{0}+\mathtt{1})^*(\varepsilon + \mathtt{0} + \mathtt{1})$. Mit demselben Argument gilt

$$(\mathtt{0}+\mathtt{1})^*(\varepsilon+\mathtt{0}+\mathtt{1}) = (\mathtt{0}+\mathtt{1})^* + (\mathtt{0}+\mathtt{1})^*(\mathtt{0}+\mathtt{1}) = (\mathtt{0}+\mathtt{1})^*$$

Damit erhalten wir als Ergebnis den regulären Ausdruck $\varepsilon + \mathtt{0}(\mathtt{0}+\mathtt{1})^*$. Der Automat akzeptiert somit alle Wörter über $\{\mathtt{0},\mathtt{1}\}$ außer jenen, die mit dem Symbol $\mathtt{1}$ beginnen.

Aufgabe 8 **(0.3 Punkte)**

Zur Identifikation von digitalen Objekten werden eindeutige Bezeichungen vergeben. Dabei beginnt solch ein *Bezeichner* immer mit der Nummer 10 gefolgt von einem Punkt. Danach folgt eine vierstellige Zahl größer oder gleich 1000. Auf diese Zahl folgt das Zeichen / auf welches ein *Suffix* folgt. Ein Suffix ist dabei eine beliebige Folge von Zeichen, Ziffern und Punkten, wobei das erste und letzte Zeichen kein Punkt sein darf und ein Punkt nicht unmittelbar auf einen anderen Punkt folgen darf. Punkte dürfen also nur zwischen Folgen aus Zeichen und Ziffern stehen. Das Suffix darf nicht leer sein. Beispiele für Bezeichner: `10.1234/journal.acm.002045` und `10.1000/proceedings.pods.003bb`

(a) Geben Sie einen regulären Ausdruck für die gültigen Bezeichner in algebraischer Notation an.

(b) Geben Sie einen regulären Ausdruck für die gültigen Bezeichner in `egrep`-Notation an. (Gesucht sind alle Zeilen, die *ausschließlich* einen derartigen Beziechner enthalten.)

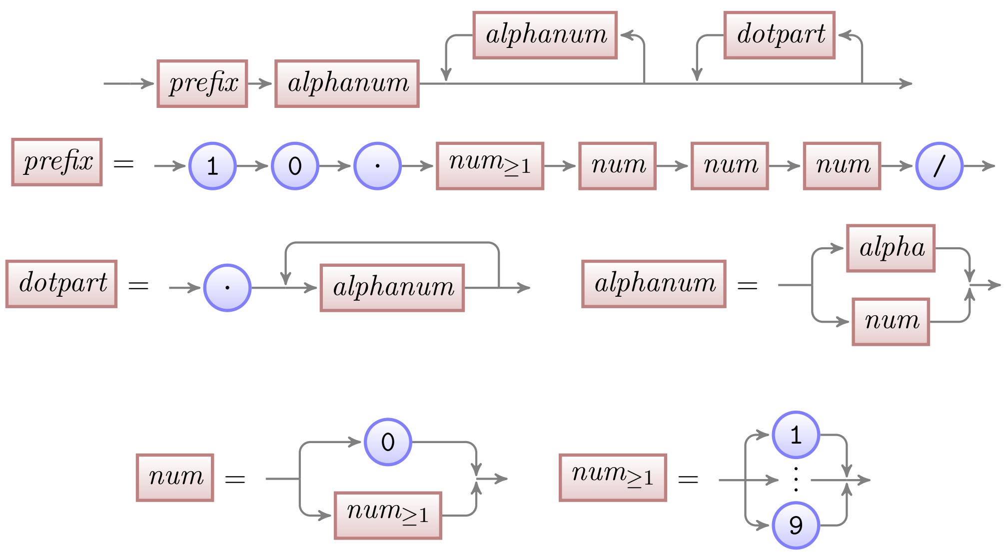
(c) Zeichnen Sie das Syntaxdiagramm, das dem regulären Ausdruck aus Teil a entspricht.

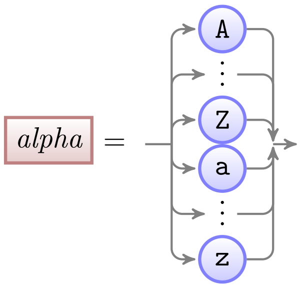
Lösung

(a) $\texttt{10.}\,\mathit{num}_{\geq 1}\mathit{num}^3\texttt{/}(\mathit{alpha} + \mathit{num})^+(\texttt{.}(\mathit{alpha} + \mathit{num})^+)^*$ mit den Abkürzungen
$\mathit{num}_{\geq 1} := \texttt{1} + \texttt{2} + \cdots + \texttt{9}$
$\mathit{num} := \texttt{0} + \mathit{num}_{\geq 1}$
$\mathit{alpha} := \texttt{a} + \cdots + \texttt{z} + \texttt{A} + \cdots + \texttt{Z}$

(b) `^10\.[1-9][0-9][0-9][0-9]/[a-zA-Z0-9]+(\.[a-zA-Z0-9]+)*$`

(c) Syntaxdiagramm zum Ausdruck $(\mathit{alpha} + \mathit{num})^+(\texttt{.}(\mathit{alpha} + \mathit{num})^+)^*$:





Aufgabe 9 (0.4 Punkte)

Gegeben sei die Grammatik $G = \langle N, T, P, A \rangle$, wobei

$$N = \{A, B, C\}$$
$$T = \{\mathtt{a}, \mathtt{d}, \mathtt{e}, \mathtt{g}, \mathtt{i}, \mathtt{n}, \mathtt{l}\}$$
$$\begin{aligned} P = \{ & A \to \mathtt{ling}\, B \mid \mathtt{ling}\ , \\ & C \to \mathtt{ling}\, C \mid \mathtt{e}\, B \mid \mathtt{a}, \\ & B \to \mathtt{ding}\, B \mid \mathtt{e}\, C \mid\ A \mid \varepsilon\ \} \end{aligned}$$

(a) 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 Parallelableitung an. Falls nein, argumentieren Sie, warum nicht.

(1) `lingling`

(2) `lingeling`

(3) `lingelingeding`

(b) Zeigen Sie, dass das Wort `lingdingdingdingdingelingedingelinga` in der Sprache $\mathcal{L}(G)$ liegt.

(c) Sind folgende Aussagen über die Sprache $\mathcal{L}(G)$ korrekt? Wenn ja, warum? Wenn nein, geben Sie ein Gegenbeispiel!

(1) Jedes Wort beginnt mit `ling`.

(2) Man kann beliebig lange Wörter bilden, in denen keine zwei gleichen Buchstaben (= Terminalsymbole) aufeinander folgen.

(3) Nach jedem `ding` muss ein `e` folgen.

(d) Konstruieren Sie einen indeterministischen endlichen Automat, der die Sprache $\mathcal{L}(G)$ akzeptiert.

Lösung

(a) (1) Ja, das Wort liegt in der Sprache $\mathcal{L}(G)$:

$$A \Rightarrow_P \mathtt{ling}\, B \Rightarrow_P \mathtt{ling}\, A \Rightarrow_P \mathtt{ling}\,\mathtt{ling}$$

(2) Die einzige Ableitung, die die Zeichenfolge `lingeling` erzeugt, ist die folgende:

$$A \Rightarrow_P \texttt{ling}\, B \Rightarrow_P \texttt{ling}\,\texttt{e}\, C \Rightarrow_P \texttt{ling}\,\texttt{e}\,\texttt{ling}\, C$$

Da es keine Produktion $C \to \varepsilon$ gibt, ist dieses Wort nicht Teil der Sprache $\mathcal{L}(G)$.

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

$$\begin{aligned} A &\Rightarrow_P \texttt{ling}\, B \Rightarrow_P \texttt{ling}\,\texttt{e}\, C \Rightarrow_P \texttt{ling}\,\texttt{e}\,\texttt{ling}\, C \Rightarrow_P \texttt{ling}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\, B \\ &\Rightarrow_P \texttt{ling}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\, B \Rightarrow_P \texttt{ling}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\,\varepsilon = \texttt{lingelingeding} \end{aligned}$$

(b) Um zu zeigen, dass das Wort in $\mathcal{L}(G)$ liegt, geben wir eine Ableitung aus dem Startsymbol an:

$$\begin{aligned} A &\Rightarrow \texttt{ling}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\, C \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\, C \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\,\texttt{e}\, C \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\, C \\ &\Rightarrow \texttt{ling}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{e}\,\texttt{ding}\,\texttt{e}\,\texttt{ling}\,\texttt{a} \\ &= \texttt{lingdingdingdingdingelingedingelinga} \end{aligned}$$

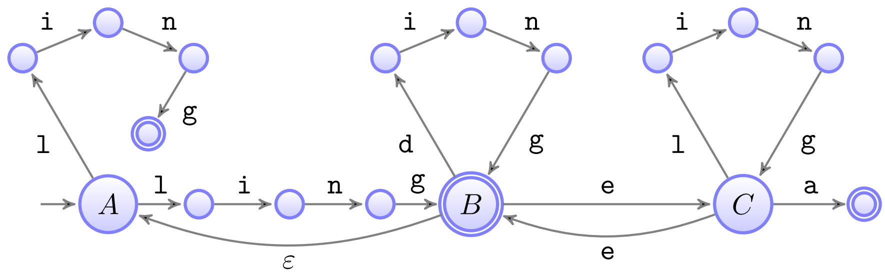
(c) (1) Richtig. Beide Produktionen für das Startsymbol A erzeugen ein Wort, das mit `ling` beginnt.

(2) Richtig. Z.B. lässt sich für jedes $n \geq 1$ das Wort $\texttt{ling}\,(\texttt{ding})^n$ ableiten, das diese Bedingung erfüllt.

$$A \Rightarrow \texttt{ling}\, B \Rightarrow \texttt{lingding}\, B \Rightarrow \cdots \Rightarrow \texttt{ling}\,(\texttt{ding})^n\, B \Rightarrow \texttt{ling}\,(\texttt{ding})^n$$

(3) Falsch. Gegenbeispiele sind die Wörter `lingdingdingdingdingelingeding` und `lingelingeding`, die in der Sprache liegen (siehe oben).

(d) Da das Alphabet laut Angabe aus einzelnen Buchstaben besteht und Übergänge nur für einzelne Symbole definiert sind, müssen wir die Silben mit Hilfe von zusätzlichen Zuständen in einzelne Buchstaben zerlegen.



Aufgabe 10 (0.4 Punkte)

Die Sprache der *Modallogik* erweitert die Aussagenlogik um zusätzliche unäre Konnektive $\Box$ und $\Diamond$. Formeln der Modallogik werden durch Aussagenvariablen und mittels der Konnektive $\{\supset, \wedge, \vee, \equiv, \neg, \Box, \Diamond\}$ gebildet.

Beispiele für Formeln: $(\Box\Box p \wedge \Diamond\neg q)$, $\neg\Box(p \wedge q)$.

(a) Geben Sie eine kontextfreie Grammatik für die Menge dieser Formeln an. Sie können die Aussagenvariablen nach Ihrem Belieben bezeichnen; stellen Sie jedoch sicher, dass unendlich viele von diesen zur Verfügung stehen. Verwenden Sie EBNF-Notation um die Grammatik kompakt zu halten.

(b) Zeigen Sie, dass $(\Box(p \supset q) \supset (\Box p \supset \Box q))$ in der Sprache Ihrer Grammatik liegt.

(c) Eine Formel heiße *positiv*, wenn sie von der Form $(\varphi \supset \psi)$ ist, wobei φ und ψ Formeln sind, welche nur aus Aussagenvariablen sowie den Konnektiven $\Diamond$ und $\wedge$ gebildet sind. Ändern Sie Ihre Grammatik, sodass diese die Menge der positiven Formeln beschreibt.

(d) Ist die Menge aller Formeln der Modallogik regulär? Wenn ja, geben Sie einen regulären Ausdruck für diese; wenn nein, geben Sie eine Begründung an.

Lösung

(a) $\langle V, T, P, \mathit{Formel}\rangle$ mit

$$V = \{\mathit{Formel}, \mathit{Konst}, \mathit{Opformel}, \mathit{Uop}, \mathit{Bop}, \mathit{Variable}, \mathit{Buchstabe}, \mathit{Ziffer}\}$$

$$T = \{\texttt{"}\top\texttt{"}, \texttt{"}\bot\texttt{"}, \texttt{"("}, \texttt{")"}, \texttt{"}\neg\texttt{"}, \texttt{"}\Box\texttt{"}, \texttt{"}\Diamond\texttt{"}, \texttt{"}\supset\texttt{"}, \texttt{"}\wedge\texttt{"}, \texttt{"}\vee\texttt{"}, \texttt{"}\equiv\texttt{"}, \texttt{"p"}, \texttt{"q"}, \texttt{"r"}, \texttt{"0"}, \ldots, \texttt{"9"}\}$$

$$\begin{array}{rlll}
P = \{ & \mathit{Formel} & \to & \mathit{Variable} \mid \mathit{Konst} \mid \mathit{Opformel} \ , \\
& \mathit{Konst} & \to & \texttt{"}\top\texttt{"} \mid \texttt{"}\bot\texttt{"} \ , \\
& \mathit{Opformel} & \to & \mathit{Uop}\ \mathit{Formel} \mid \texttt{"("}\mathit{Formel}\ \mathit{Bop}\ \mathit{Formel}\texttt{")"} \ , \\
& \mathit{Uop} & \to & \texttt{"}\neg\texttt{"} \mid \texttt{"}\Box\texttt{"} \mid \texttt{"}\Diamond\texttt{"} \ , \\
& \mathit{Bop} & \to & \texttt{"}\supset\texttt{"} \mid \texttt{"}\wedge\texttt{"} \mid \texttt{"}\vee\texttt{"} \mid \texttt{"}\equiv\texttt{"} \ , \\
& \mathit{Variable} & \to & \mathit{Buchstabe}\ \{\mathit{Ziffer}\} \ , \\
& \mathit{Buchstabe} & \to & \texttt{"p"} \mid \texttt{"q"} \mid \texttt{"r"} \ , \\
& \mathit{Ziffer} & \to & \texttt{"0"} \mid \cdots \mid \texttt{"9"} \ \}
\end{array}$$

Es gibt auch andere Möglichkeiten gibt, um eine unbeschränkte Menge von Variablennamen zu generieren.

(b) Um zu zeigen, dass die gegebene Formel in der Sprache der Grammatik liegt, geben wir eine Parallelableitung für diese an.

$$
\begin{aligned}
\mathit{Formel} &\Rightarrow_p \mathit{Opformel} \\
&\Rightarrow_p \texttt{"("}\,\mathit{Formel}\ \mathit{Bop}\ \mathit{Formel}\,\texttt{")"} \\
&\Rightarrow_p \texttt{"("}\,\mathit{Opformel}\,\texttt{"}\supset\texttt{"}\ \mathit{Opformel}\,\texttt{")"} \\
&\Rightarrow_p \texttt{"("}\,\mathit{Uop}\ \mathit{Formel}\,\texttt{"}\supset\texttt{ ("}\,\mathit{Formel}\ \mathit{Bop}\ \mathit{Formel}\texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\Box\texttt{"}\ \mathit{Opformel}\,\texttt{"}\supset\texttt{ ("}\,\mathit{Opformel}\,\texttt{"}\supset\texttt{"}\ \mathit{Opformel}\texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\Box\texttt{("}\mathit{Formel}\ \mathit{Bop}\ \mathit{Formel}\texttt{")}\supset\texttt{("}\,\mathit{Uop}\ \mathit{Formel}\,\texttt{"}\supset\texttt{"}\ \mathit{Uop}\ \mathit{Formel}\texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\Box\texttt{("}\,\mathit{Variable}\,\texttt{"}\supset\texttt{"}\ \mathit{Variable}\texttt{")}\supset\texttt{(}\Box\texttt{"}\ \mathit{Variable}\,\texttt{"}\supset\Box\texttt{"}\ \mathit{Variable}\texttt{"))"} \\
&\Rightarrow_p \texttt{"(}\Box\texttt{(p}\supset\texttt{q)}\supset\texttt{(}\Box\texttt{p}\supset\Box\texttt{q))"}
\end{aligned}
$$

(c) $\langle V, T, P, \mathit{Posformel}\rangle$ mit

$$
\begin{aligned}
V &= \{\mathit{Posformel}, \mathit{Formel}, \mathit{Variable}, \mathit{Buchstabe}, \mathit{Ziffer}\} \\
T &= \{\texttt{"("}, \texttt{")"}, \texttt{"}\supset\texttt{"}, \texttt{"}\Diamond\texttt{"}, \texttt{"}\wedge\texttt{"}, \texttt{"p"}, \texttt{"q"}, \texttt{"r"}, \texttt{"0"}, \ldots, \texttt{"9"}\} \\
P &= \{
\begin{array}[t]{lll}
\mathit{Posformel} & \to & \texttt{"("}\mathit{Formel}\,\texttt{"}\supset\texttt{"}\ \mathit{Formel}\,\texttt{")"}\ , \\
\mathit{Formel} & \to & \mathit{Variable} \mid \texttt{"}\Diamond\texttt{"}\ \mathit{Formel} \mid \texttt{"("}\mathit{Formel}\,\texttt{"}\wedge\texttt{"}\ \mathit{Formel}\,\texttt{")"}\ , \\
\mathit{Variable} & \to & \mathit{Buchstabe}\,\{\mathit{Ziffer}\}\ , \\
\mathit{Buchstabe} & \to & \texttt{"p"} \mid \texttt{"q"} \mid \texttt{"r"}\ , \\
\mathit{Ziffer} & \to & \texttt{"0"} \mid \cdots \mid \texttt{"9"}\ \}
\end{array}
\end{aligned}
$$

(d) Diese Sprache ist nicht regulär. Dies lässt sich durch die Verschachtelung der Klammerung einsehen. Es kann eine beliebige Anzahl von öffnenden Klammern auftreten, ehe auch nur eine schließende Klammer vorkommt. Ein endlicher Automat müsste sich die Anzahl der öffnenden Klammern merken. Da diese Zahl nach oben unbeschränkt ist, reicht eine endliche Zahl von Zuständen nicht aus.

Aufgabe 11 (0.3 Punkte)

Seien $P/1$ und $S/1$ einstellige Prädikatensymbole.

(a) Zeigen Sie, dass die Formeln $\forall x\,(P(x) \supset S(x))$ und $\forall x\,(P(x) \wedge S(x))$ nicht äquivalent sind.

(b) Zeigen Sie, dass die Formeln $\exists x\,(P(x) \wedge S(x))$ und $\exists x\,(P(x) \supset S(x))$ nicht äquivalent sind.

(c) Wählen Sie für die Prädikatensymbole P und S Bedeutungen in natürlicher Sprache. Erklären Sie anschaulich anhand der resultierenden Übersetzungen von $\forall x\,(P(x) \supset S(x))$ und $\forall x\,(P(x) \wedge S(x))$, warum diese beiden Formeln nicht äquivalent sind.

Lösung

(a) Wir betrachten eine Interpretation I mit Universum $\mathcal{U} = \{a\}$ und $I(P) = I(S) = \emptyset$. Offenbar erfüllt I die Formel $\forall x\,(P(x) \supset S(x))$, während I die Formel $\forall x\,(P(x) \wedge S(x))$ nicht erfüllt, da a nicht das Prädikat $I(P)$ erfüllt. Es folgt nun unmittelbar, dass die beiden genannten Formeln nicht äquivalent sind.

(b) Wir betrachten die gleiche Interpretation I wie in Teilaufgabe a. Wir stellen fest, dass I die Formel $\exists x\,(P(x) \supset S(x))$ erfüllt. Denn a erfüllt das Prädikat $I(P)$ nicht, somit gilt $\mathsf{val}_{I,\sigma}(P(x) \supset S(x)) = 1$ (wobei $\sigma(x) = a$) *a fortiori*. Jedoch erfüllt I nicht die Formel $\exists x\,(P(x) \wedge S(x))$, da ja kein Objekt von $\mathcal{U}$ das Prädikat $I(P)$ erfüllt.

(c) Die Formel $\forall x\,(P(x) \supset S(x))$ bedeutet „Alle Objekte mit Eigenschaft P haben Eigenschaft S". Die Formel $\forall x\,(P(x) \wedge S(x))$ bedeutet „Alle Objekte haben Eigenschaft P und S". Wählen wir für P und S folgende Bedeutungen:

$$\begin{array}{ll} P(x) \ldots & x \text{ ist ein Engel} \\ S(x) \ldots & x \text{ hat Flügel} \end{array}$$

Dann bedeutet $\forall x\,(P(x) \supset S(x))$, dass alle Engel Flügel haben, während $\forall x\,(P(x) \wedge S(x))$ bedeutet, dass alles ein Engel ist und Flügel hat.

Aufgabe 12 (0.4 Punkte)

Seien *Isst*/2, *Mädchen*/1, *Speise*/1 und *Gesund*/1 Prädikatensymbole sowie *salat* und *steak* Konstantensymbole mit folgender Bedeutung:

$\mathit{Isst}(x, y)$	... x isst y	$\mathit{Gesund}(x)$	... x ist gesund
$\mathit{Mädchen}(x)$	... x ist ein Mädchen	*salat*	... Salat
$\mathit{Speise}(x)$	... x ist eine Speise	*steak*	... Steak

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

(a) Manche Mädchen essen nur dann Steak, wenn sie auch gesunde Speisen essen.

(b) Gesunde Mädchen essen manche Speisen, aber keinen Salat.

Sei weiters folgende Interpretation gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Anna}, \text{Tina}, \text{Mia}, \text{Sophie}, \text{Salat}, \text{Schnitzel}, \text{Spätzle}, \text{Suppe},\\
&\quad\ \text{Steak}, \text{Spaghetti}, \text{Pizza}\}\\
I(\mathit{Mädchen}) &= \{\text{Anna}, \text{Mia}, \text{Sophie}\}\\
I(\mathit{Speise}) &= \{\text{Salat}, \text{Schnitzel}, \text{Suppe}, \text{Steak}\}\\
I(\mathit{Gesund}) &= \{\text{Suppe}, \text{Salat}, \text{Pizza}, \text{Steak}\}\\
I(\mathit{Isst}) &= \{(\text{Anna}, \text{Suppe}), (\text{Anna}, \text{Salat}), (\text{Anna}, \text{Spätzle}),\\
&\quad\ (\text{Mia}, \text{Schnitzel}), (\text{Mia}, \text{Salat}), (\text{Mia}, \text{Suppe}),\\
&\quad\ (\text{Tina}, \text{Pizza}), (\text{Tina}, \text{Salat}), (\text{Tina}, \text{Spaghetti}),\\
&\quad\ (\text{Sophie}, \text{Steak}), (\text{Sophie}, \text{Spaghetti})\}\\
I(\mathit{salat}) &= \text{Salat} \qquad I(\mathit{steak}) = \text{Steak}
\end{aligned}$$

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

(c) $\exists x\,(\mathit{Isst}(x, \mathit{salat}) \supset \mathit{Isst}(x, \mathit{steak}))$

(d) $\forall x\,(\mathit{Isst}(x, \mathit{salat}) \not\equiv \mathit{Isst}(x, \mathit{steak}))$

(e) $\exists x \exists y\,(\mathit{Speise}(y) \land \mathit{Mädchen}(x) \land \mathit{Isst}(x, y))$

(f) $\forall x \exists y\,(\mathit{Gesund}(x) \supset (\mathit{Mädchen}(y) \land \mathit{Isst}(y, x)))$

Lösung

(a) $\exists x(\mathit{Mädchen}(x) \land (\mathit{Isst}(x, \mathit{steak}) \supset \exists y(\mathit{Gesund}(y) \land \mathit{Speise}(y) \land \mathit{Isst}(x, y))))$ oder $\exists x \exists y(\mathit{Mädchen}(x) \land (\mathit{Isst}(x, \mathit{steak}) \supset (\mathit{Gesund}(y) \land \mathit{Speise}(y) \land \mathit{Isst}(x, y))))$

(b) $\forall x((\mathit{Mädchen}(x) \land \mathit{Gesund}(x)) \supset (\exists y(\mathit{Speise}(y) \land \mathit{Isst}(x, y)) \land \neg \mathit{Isst}(x, \mathit{salat})))$ oder $\forall x \exists y((\mathit{Mädchen}(x) \land \mathit{Gesund}(x)) \supset (\mathit{Speise}(y) \land \mathit{Isst}(x, y) \land \neg \mathit{Isst}(x, \mathit{salat})))$

(c) Wahr, da $(\text{Sophie}, \text{Steak}) \in I(\mathit{Isst})$.

(d) Falsch, da sowohl $(\text{Salat}, \text{Salat}) \notin I(\mathit{Isst})$ als auch $(\text{Salat}, \text{Steak}) \notin I(\mathit{Isst})$.

(e) Wahr, da zum Beispiel $\text{Suppe} \in I(\mathit{Speise})$, $\text{Anna} \in I(\mathit{Mädchen})$ und $(\text{Anna}, \text{Suppe}) \in I(\mathit{Isst})$.

(f) Falsch, da $\text{Pizza} \in I(\mathit{Gesund})$, aber keines der Mädchen isst Pizza: $(\text{Anna}, \text{Pizza}) \notin I(\mathit{Isst})$, $(\text{Mia}, \text{Pizza}) \notin I(\mathit{Isst})$ und $(\text{Sophie}, \text{Pizza}) \notin I(\mathit{Isst})$.

Aufgabe 13 (0.4 Punkte)

Es seien A und B prädikatenlogische Formeln. Wir erinnern, dass $A \models B$ genau dann gilt, wenn $A \models_{I,\sigma} B$ für alle I, σ gilt. Für eine Formel F schreiben wir $I \models F$, wenn $\mathsf{val}_{I,\sigma}(F) = 1$ für alle Variablenbelegungen σ. Wir schreiben $A \models^* B$, wenn für alle Interpretationen I gilt: wenn $I \models A$, dann auch $I \models B$.

Wie hängen die Begriffe $A \models B$ und $A \models^* B$ zusammen? Untersuchen Sie, ob aus $A \models B$ auch $A \models^* B$ folgt und umgekehrt. Geben Sie ein Gegenbeispiel an, falls die entsprechende Behauptung nicht gilt und begründen Sie anderenfalls, warum die Behauptung gilt.

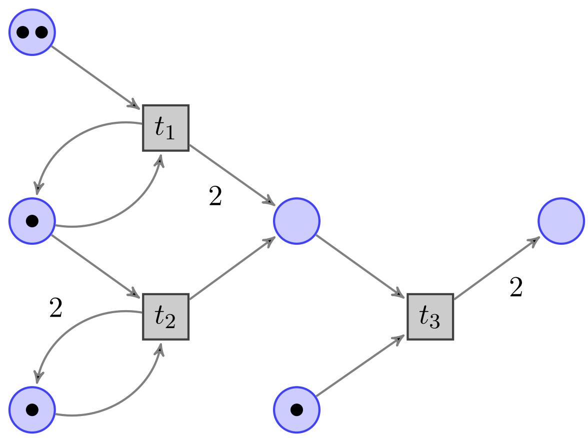
Lösung

Wir stellen fest, dass aus $A \models^* B$ nicht $A \models B$ folgt. Für dies betrachten wir die beiden Formeln $A := P(x)$ und $B := P(y)$, wobei $x \neq y$ zwei voneinander verschiedene Variablensymbole seien. Nun gilt $P(x) \models^* P(y)$. Zum Beweis betrachten wir eine Interpretation I für welche $I \models P(x)$ gilt. Wir müssen zeigen, dass auch $I \models P(y)$ gilt. Wegen $I \models P(x)$ gilt $\mathsf{val}_{I,\sigma}(P(x)) = 1$ für jede Variablenbelegung σ. Somit gilt per Definition auch $I \models P(y)$. Daraus folgt, dass $P(x) \models^* P(y)$ gilt. Es gilt nun aber nicht $P(x) \models P(y)$. Sei I eine Interpretation mit Universum $\mathcal{U} = \{a, b\}$ und $I(P) = \{a\}$. Sei weiters σ eine Variablenbelegung mit $\sigma(x) = a$ und $\sigma(y) = b$. Wir stellen leicht fest, dass $\mathsf{val}_{I,\sigma}(P(x)) = 1$, während $\mathsf{val}_{I,\sigma}(P(y)) = 0$. Daraus folgt, dass $P(x) \models P(y)$ nicht gilt.

Wir behaupten, dass aus $A \models B$ immer auch $A \models^* B$ folgt. Zum Beweis nehmen wir an, dass $A \models B$ gilt und betrachten eine (beliebig gewählte) Interpretation I für die $I \models A$ gilt. Wir müssen zeigen, dass auch $I \models B$ gilt. Da wir I beliebig gewählt haben, folgt dann daraus unmittelbar $A \models^* B$. Nun gilt $I \models B$ genau dann, wenn $\mathsf{val}_{I,\sigma}(B) = 1$ für jede Variablenbelgung σ. Sei σ eine beliebige Variablenbelegung. Wegen $I \models A$ folgt unmittelbar $\mathsf{val}_{I,\sigma}(A) = 1$. Wegen $A \models B$ folgt $A \models_{I,\sigma} B$ und somit $\mathsf{val}_{I,\sigma}(B) = 1$. Da wir σ beliebig gewählt haben, folgt daraus dass $I \models B$, was zu zeigen war.

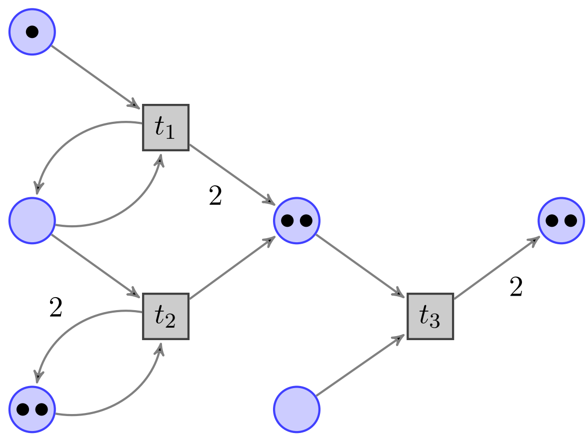
Aufgabe 14 (0.2 Punkte)

Bestimmen Sie eine Folge von Transitionen, die nacheinander feuern können. Die Folge soll jede Transition mindestens einmal enthalten. Nach dem Feuern der letzten Transition soll keine Transition mehr aktiviert sein. Geben Sie die Transitionen dieser Folge sowie die erreichte Endmarkierung an.

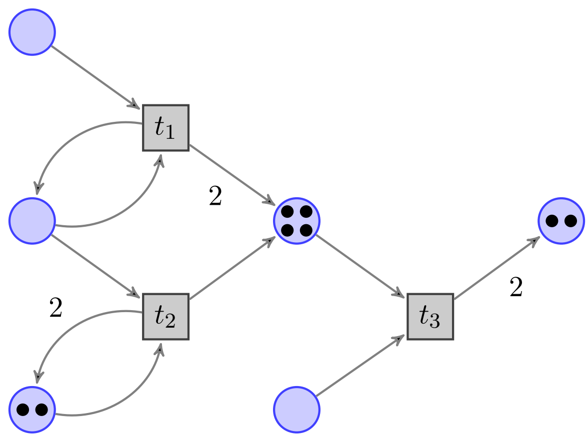


Lösung

Die Transitionsfolgen t_1–t_2–t_3 und t_1–t_3–t_2 liefern:



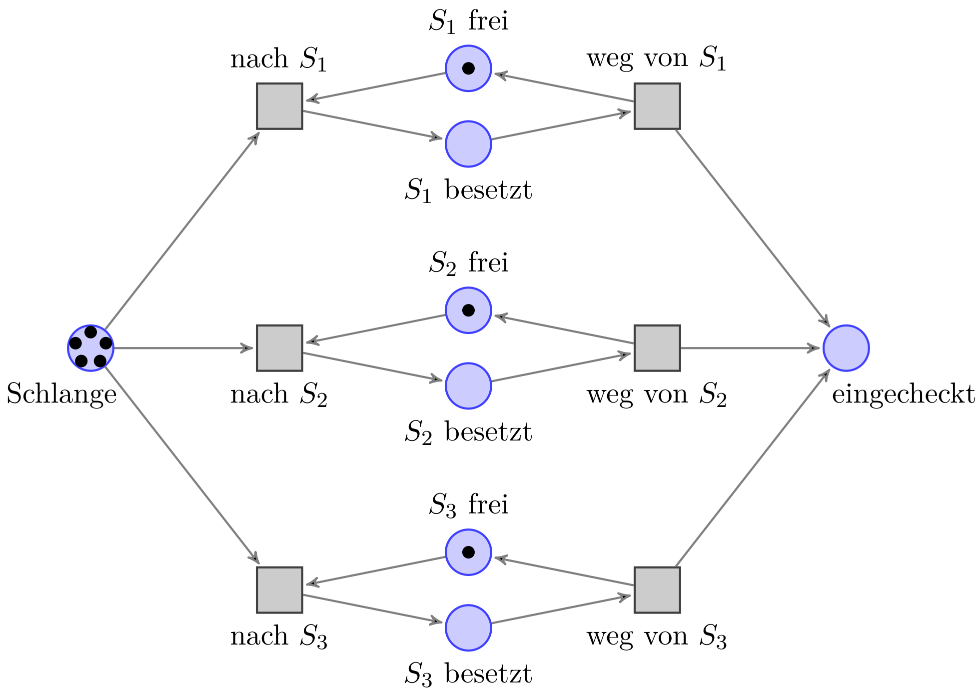
Die Transitionsfolgen t_1–t_1–t_2–t_3, t_1–t_1–t_3–t_2 und t_1–t_3–t_1–t_2 liefern:



Aufgabe 15 (0.4 Punkte)

Am Flughafen gibt es eine Fluglinie mit drei Check-in Schaltern, bei denen die Fluggäste parallel einchecken können. Dazu müssen sie sich in einer gemeinsamen Schlange anstellen. Sobald ein Gast an der Reihe ist, geht er zum nächsten freien Check-in Schalter. Bei jedem Schalter kann immer nur ein Gast bedient werden. Danach sammeln sich die Fluggäste am Gate.
Modellieren Sie dieses System mit Hilfe eines Petri-Netzes. Geben Sie eine geeignete Anfangsmarkierung an. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben.

Lösung



Da die Verteilung auf die drei Schalter von einer einzigen Warteschlange aus erfolgt, befinden sich alle Token, die Kunden repräsentieren, zunächst an einer Stelle, der „Schlange“. Kann eine der Transitionen „nach S_1“, „nach S_2“ bzw. „nach S_3“ schalten, heißt das, dass einer der drei Schalter frei ist und der Kunde zu diesem Schalter gehen kann. Dabei ist nicht definiert, zu welchem Schalter er geht, falls gleichzeitig mehrere Schalter frei sind – also welche Transition schaltet, falls mehrere Tranisitonen aktiviert sind. Wenn der Check-in eines Gastes beendet ist, so wird dieser Schalter für den nächsten Gast frei.