2. Übungsblatt (mit Lösungen)

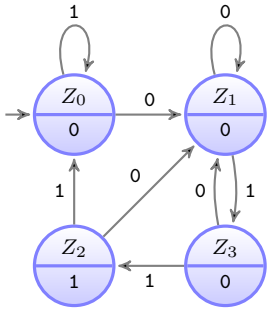
3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

1.Mai 2015 (Korrektur 4.Mai)

Aufgabe 1 (0.3 Punkte)

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



(a) Geben Sie die Ausgaben zu folgenden Eingaben an: 10111, 01001, 01101.

(b) Berechnen Sie schrittweise $\delta^*(Z_0, 10110)$ und $\gamma^*(Z_0, 10110)$.

(c) Beschreibe Sie die Übersetzungsfunktion $[\mathcal{A}]$.

Lösung

(a)

w:	10111	01001	01101
$[\mathcal{A}](w)$:	00010	00000	00100

(b)
$$\begin{aligned}
\delta^*(Z_0, 10110) &= \delta^*(\delta(Z_0, 1), 0110 &&= \delta^*(Z_0, 0110)\\
&= \delta^*(\delta(Z_0, 0), 110) &&= \delta^*(Z_1, 110)\\
&= \delta^*(\delta(Z_1, 1), 10) &&= \delta^*(Z_3, 10)\\
&= \delta^*(\delta(Z_3, 1), 0) &&= \delta^*(Z_2, 0)\\
&= \delta^*(\delta(Z_2, 0), \varepsilon) &&= \delta^*(Z_1, \varepsilon)\\
&= Z_1
\end{aligned}$$

$$
\begin{aligned}
\gamma^*(Z_0, \texttt{10110}) &= \gamma(\delta(Z_0, \texttt{1})) \cdot \gamma^*(\delta(Z_0, \texttt{1}), \texttt{0110}) && = \gamma(Z_0) \cdot \gamma^*(Z_0, \texttt{0110}) \\
&= 0 \cdot \gamma(\delta(Z_0, \texttt{0})) \cdot \gamma^*(\delta(Z_0, \texttt{0}), \texttt{110}) && = 0 \cdot \gamma(Z_1) \cdot \gamma^*(Z_1, \texttt{110}) \\
&= 00 \cdot \gamma(\delta(Z_1, \texttt{1})) \cdot \gamma^*(\delta(Z_1, \texttt{1}), \texttt{10}) && = 00 \cdot \gamma(Z_3) \cdot \gamma^*(Z_3, \texttt{10}) \\
&= 000 \cdot \gamma(\delta(Z_3, \texttt{1})) \cdot \gamma^*(\delta(Z_3, \texttt{1}), \texttt{0}) && = 000 \cdot \gamma(Z_2) \cdot \gamma^*(Z_2, \texttt{0}) \\
&= 0001 \cdot \gamma(\delta(Z_2, \texttt{0})) \cdot \gamma^*(\delta(Z_2, \texttt{0}), \varepsilon) && = 0001 \cdot \gamma(Z_1) \cdot \gamma^*(Z_1, \varepsilon) \\
&= 00010 \cdot \varepsilon = 00010
\end{aligned}
$$

(c) Der Automat erkennt die Folge `011` in der Eingabe: Wird `011` eingelesen, so ist die Ausgabe `1`, sonst `0`.

Aufgabe 2 (0.3 Punkte)

Sei L die Sprache

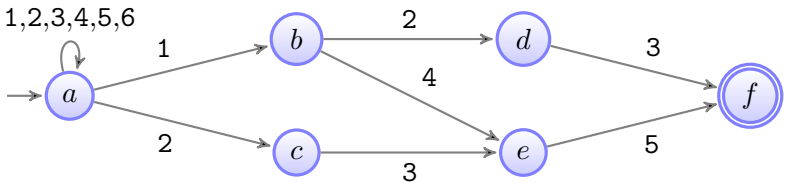
$$\{\, w \in \{\texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}\}^* \mid w \text{ endet mit der Ziffernfolge } \texttt{123}, \texttt{145} \text{ oder } \texttt{235} \,\} \ .$$

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 a indeterministisch: Für das Symbol `1` und für das Symbol `2` 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^*	1	2	3	4	5	6
a	$\{a, b\}$	$\{a, c\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$
b	$\{\}$	$\{d\}$	$\{\}$	$\{e\}$	$\{\}$	$\{\}$
c	$\{\}$	$\{\}$	$\{e\}$	$\{\}$	$\{\}$	$\{\}$
d	$\{\}$	$\{\}$	$\{f\}$	$\{\}$	$\{\}$	$\{\}$
e	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{f\}$	$\{\}$
f	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$

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 $\{a\}$ bezeichnet, da sich der indeterministische Automat zu Beginn im Zustand a (und nur in diesem) befindet. Von diesem Zustand ausgehend erstellen wir zeilenweise die Tabelle für die Übergangsfunktion des deterministischen Automaten.

$\hat{\delta}$	1	2	3	4	5	6
$\{a\}$	$\{a,b\}$	$\{a,c\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{a,b\}$	$\{a,b\}$	$\{a,c,d\}$	$\{a\}$	$\{a,e\}$	$\{a\}$	$\{a\}$
$\{a,c\}$	$\{a,b\}$	$\{a,c\}$	$\{a,e\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{a,c,d\}$	$\{a,b\}$	$\{a,c\}$	$\{a,e,f\}$	$\{a\}$	$\{a\}$	$\{a\}$
$\{a,e\}$	$\{a,b\}$	$\{a,c\}$	$\{a\}$	$\{a\}$	$\{a,f\}$	$\{a\}$
$\{a,e,f\}$	$\{a,b\}$	$\{a,c\}$	$\{a\}$	$\{a\}$	$\{a,f\}$	$\{a\}$
$\{a,f\}$	$\{a,b\}$	$\{a,c\}$	$\{a\}$	$\{a\}$	$\{a\}$	$\{a\}$

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 f enthält. Dieser wird somit durch das Tupel $\langle \widehat{Q}, \Sigma, \hat{\delta}, \{a\}, \widehat{F} \rangle$ beschrieben, wobei

$$\Sigma = \{1,2,3,4,5,6\}$$
$$\widehat{F} = \{\{a,e,f\},\{a,f\}\}$$
$$\widehat{Q} = \{\{a\},\{a,b\},\{a,c\},\{a,c,d\},\{a,e\}\} \cup \widehat{F}$$

Aufgabe 3 (0.4 Punkte)

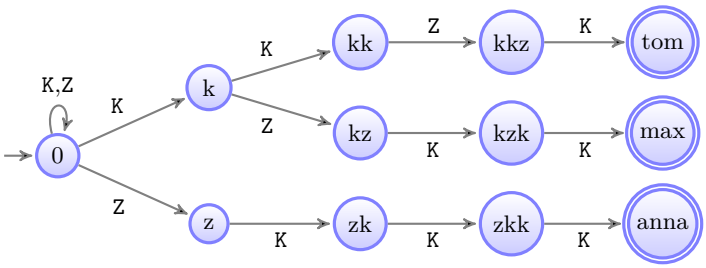
Anna, Max und Tom verwenden folgendes Verfahren um zu entscheiden, wer von ihnen das letzte Stück Torte bekommt. Sie werfen solange eine Münze, bis eines der folgenden Muster auftritt:

- Bei Zahl-Kopf-Kopf-Kopf gewinnt Anna.
- Bei Kopf-Zahl-Kopf-Kopf gewinnt Max.
- Bei Kopf-Kopf-Zahl-Kopf gewinnt Tom.

Beschreiben Sie die möglichen Wurffolgen durch einen deterministischen endlichen Automaten, dessen Endzustände den Gewinner signalisieren. (Geben Sie an, welcher Endzustand welcher Person entspricht.)
Hinweis: Beginnen Sie mit einem indeterministischen Automaten, bei dem Sie nur sicherstellen, dass vor dem Erreichen eines Gewinnzustandes eine der drei Gewinnfolgen gewürfelt wurde, ohne darauf Rücksicht zu nehmen, ob bereits davor Gewinnfolgen aufgetreten sind. Determinisieren Sie diesen Automaten, wobei Sie Endzustände nicht mehr weiterverfolgen müssen. (Wenn jemand gewonnen hat, ist das Spiel ja zu Ende.)

Lösung

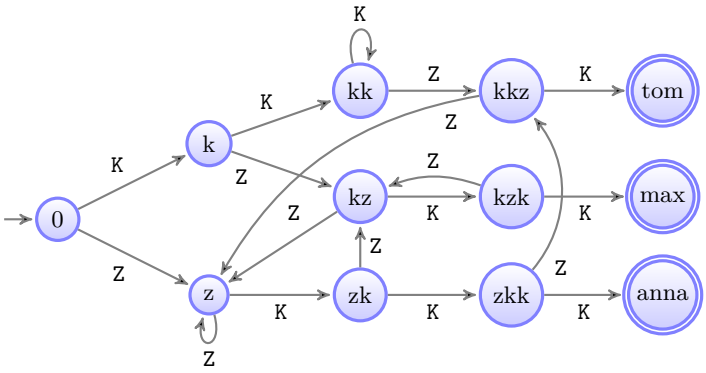
Wir verwenden `K` für Kopf und `Z` für Zahl. Ein indeterministischer Automat, der alle Wurffolgen beschreibt, die mit einer der Gewinnfolgen `ZKKK`, `KZKK` oder `KKZK` enden, sieht folgendermaßen aus.



Im Allgemeinen enthalten die durch diesen Automaten beschriebenen Wurffolgen auch solche mit mehreren Gewinnfolgen, wie etwa `ZKKKZKK`. Wenn wir diesen Automaten determinisieren und dabei abbrechen, sobald ein Endzustand erreicht ist (also einer, der tom, max oder anna enthält), erhalten wir einen deterministischen Automaten für das ursprüngliche Spiel.

	K	Z
$\{0\}$	$\{0, \mathrm{k}\}$	$\{0, \mathrm{z}\}$
$\{0, \mathrm{k}\}$	$\{0, \mathrm{k}, \mathrm{kk}\}$	$\{0, \mathrm{z}, \mathrm{kz}\}$
$\{0, \mathrm{z}\}$	$\{0, \mathrm{k}, \mathrm{zk}\}$	$\{0, \mathrm{z}\}$
$\{0, \mathrm{k}, \mathrm{kk}\}$	$\{0, \mathrm{k}, \mathrm{kk}\}$	$\{0, \mathrm{z}, \mathrm{kz}, \mathrm{kkz}\}$
$\{0, \mathrm{z}, \mathrm{kz}\}$	$\{0, \mathrm{k}, \mathrm{zk}, \mathrm{kzk}\}$	$\{0, \mathrm{z}\}$
$\{0, \mathrm{k}, \mathrm{zk}\}$	$\{0, \mathrm{k}, \mathrm{kk}, \mathrm{zkk}\}$	$\{0, \mathrm{z}, \mathrm{kz}\}$
$\{0, \mathrm{z}, \mathrm{kz}, \mathrm{kkz}\}$	tom	$\{0, \mathrm{z}\}$
$\{0, \mathrm{k}, \mathrm{zk}, \mathrm{kzk}\}$	max	$\{0, \mathrm{z}, \mathrm{kz}\}$
$\{0, \mathrm{k}, \mathrm{kk}, \mathrm{zkk}\}$	anna	$\{0, \mathrm{z}, \mathrm{kz}\}$

In der graphische Darstellung bezeichnen wir die Zustände mit dem letzten Element der Mengen in obiger Tabelle, die die Zustände eindeutig kennzeichnen.



Aufgabe 4 (0.2 Punkte)

Konstruieren Sie endliche Automaten, die dieselbe Sprache beschreiben wie die folgenden regulären Ausdrücke.

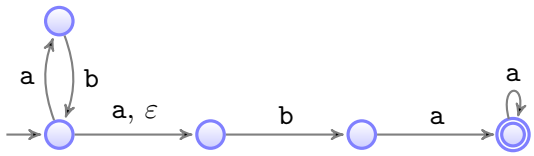
(a) $(\mathtt{ab})^*(\mathtt{b} + \mathtt{ab})\mathtt{a}^+$

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

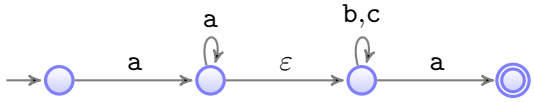
Lösung

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 bereits vereinfacht.

(a)

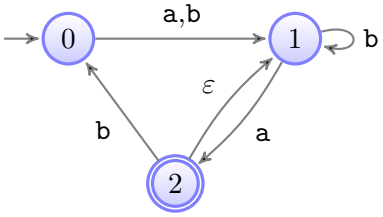


(b)



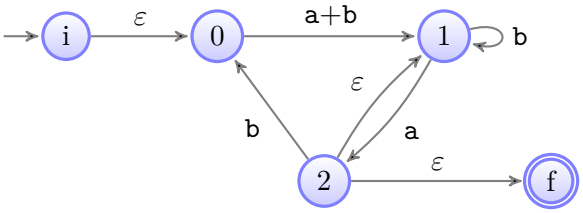
Aufgabe 5 (0.4 Punkte)

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!



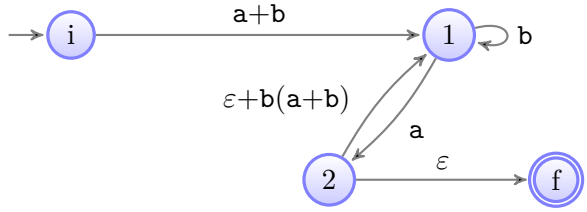
Lösung

Neuer Anfangs- und Endzustand:

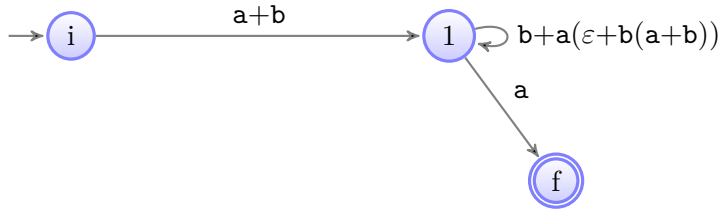


Wir eliminieren die Zustände in der Reihenfolge 0, 2 und 1; andere Reihenfolgen sind ebenfalls möglich.

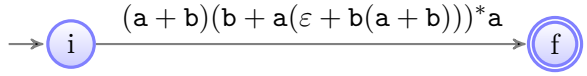
Elimination von Zustand 0:



Elimination von Zustand 2:



Elimination von Zustand 1:



Dieser Ausdruck lässt sich noch vereinfachen:

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

Die Sprache des ursprünglichen Automaten wird also durch den Ausdruck $(\mathtt{a}+\mathtt{b})^+\mathtt{a}$ beschrieben.

Aufgabe 6 (0.3 Punkte)

Das ICAO Dokument 9303[1] der *International Civil Aviation Organization* legt fest, wie maschinenlesbare Reisepässe aussehen müssen. Neben einem Lichtbild und den persönlichen Daten müssen derartige Pässe auch zwei Zeilen in der maschinenlesbaren Schrift OCR-B enthalten (siehe die letzten beiden Zeilen in Abb. 1). Die erste der beiden Zeilen ist folgendermaßen aufgebaut.

- Das erste Zeichen ist immer ein `P` und zeigt an, dass es sich um einen Pass handelt.
- Anschließend folgt optional ein Großbuchstabe, der beliebig von der Behörde festgelegt werden kann. Entfällt dieser zweite Buchstabe, muss stattdessen das Zeichen `<` folgen.

[1] `http://www.icao.int/publications/Documents/9303_p1_v1_cons_en.pdf`

Abbildung 1: Maschinenlesbarer Pass

- Es folgen ein bis drei Großbuchstaben, die das Ausstellerland identifizieren. Werden weniger als drei Buchstaben verwendet, wird der Rest von hinten beginnend mit < aufgefüllt.
- Der Nachname besteht aus beliebig vielen Großbuchstaben sowie dem Apostroph-Zeichen ’ und ist durch zwei < von den Vornamen getrennt.
- Die Vornamen (mindestens einer) bestehen aus beliebig vielen Großbuchstaben und sind voneinander durch ein < getrennt.
- Danach folgt eine beliebige Anzahl von < Zeichen, um die Zeile aufzufüllen.

(a) Geben Sie einen regulären Ausdruck in algebraischer Notation an, der die Menge dieser Zeilen beschreibt.

(b) Geben Sie einen Posix Extended Regular Expression an, der alle Zeilen beschreibt, die ausschließlich eine derartige maschinenlesbare Zeile enthalten.

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

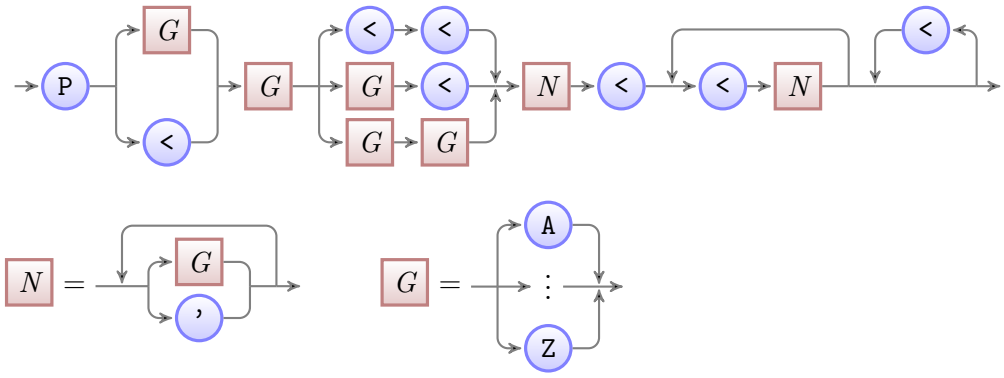
Lösung

Anmerkung: Die Verwendung von Apostrophen innerhalb des Namens entspricht nicht der Passnorm; es handelt sich dabei um eine unbeabsichtigte Änderung im Laufe der Aufgabenerstellung. Darüberhinaus weist die Aufgabe weitere (beabsichtigte) Unterschiede zum Standard auf, um sie zu vereinfachen. Die vorgeschlagene Lösung orientiert sich an der Angabe, nicht am offiziellen Standard.

(a) $\texttt{P}\,(G + \texttt{<})\,G\,(\texttt{<<} + G\texttt{<} + GG)\,N\,\texttt{<}\,(\texttt{<}N)^{+}\texttt{<}^{*}$
wobei $G = \texttt{A} + \cdots + \texttt{Z}$ und $N = (G + \texttt{'})^{+}$ gilt. Der Ausdruck X^{+} ist äquivalent zu XX^{*}.

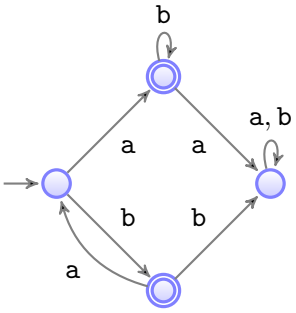
(b) `^P[A-Z<][A-Z](<<|[A-Z]<|[A-Z][A-Z])[A-Z']+<(<[A-Z']+)+<*$`

(c)



Aufgabe 7 (0.3 Punkte)

Sei w^r das Spiegelwort zum Wort w, d.h., w^r ist das Wort w von hinten nach vorne gelesen. Beispielsweise gilt $(\texttt{abac})^r = \texttt{caba}$. Sei weiters L^r die Spiegelsprache zur Sprache L, die dadurch entsteht, dass jedes Wort in L gespiegelt wird: $L^r = \{\, w^r \mid w \in L \,\}$.
Sei $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ ein beliebiger deterministischer Automat und $L = \mathcal{L}(\mathcal{A})$ die von ihm akzeptierte Sprache. Geben Sie ein Verfahren an, um daraus einen Automaten $\mathcal{A}^r$ für die Spiegelsprache zu erhalten; es soll also $\mathcal{L}(\mathcal{A}^r) = L^r$ gelten. Welche Eigenschaft regulärer Sprachen ergibt sich daraus? Lässt sich das Verfahren auch auf nicht-deterministische Automaten anwenden? Wenden Sie Ihr Verfahren auf den folgenden Automaten an und konstruieren Sie einen Automaten für die Spiegelsprache.



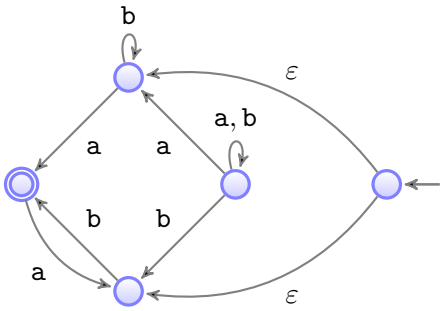
Lösung

Die grundlegende Idee ist, alle Übergänge zu spiegeln und Start- und Endzustände auszutauschen. Falls $\mathcal{A}$ mehrere Endzustände besitzt, gäbe es aber mehrere Startzustände in $\mathcal{A}^r$. Um das zu vermeiden, muss man einen neuen Startzustand einführen, von dem

aus man mit dem Leerwort in die ursprünglichen Endzustände wechseln kann. Im Allgemeinen ist $\mathcal{A}^r$ also aus zwei Gründen indeterministisch: einerseits ε-Kanten vom neuen Startzustand zu den alten Endzuständen, andererseits besitzt ein Zustand q für ein Eingabesymbol nun mehrere Folgezustände, wenn man in $\mathcal{A}$ (deterministsch) von diesen Folgezuständen nach q gelangt.
Formale Darstellung: $\mathcal{A}^r = \langle Q \cup \{i\}, \Sigma, \delta^r, i, \{q_0\}\rangle$, wobei die Übergangsrelation δ^r definiert ist durch

$$\delta^r = \{\, (\delta(q,s), s, q) \mid q \in Q,\ s \in \Sigma \,\} \cup \{\, (i, \varepsilon, q) \mid q \in F \,\}$$

Da die regulären Sprachen genau jene Sprachen sind, die sich durch endliche Automaten darstellen lassen, folgt daraus, dass die regulären Sprachen unter Spiegelung abgeschlossen sind. Das Verfahren kann auf beliebige Automaten angewendet werden, sie müssen nicht deterministisch sein.
Angewendet auf den Beispielautomaten erhalten wir:



Aufgabe 8 (0.4 Punkte)

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

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

(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) `ringring`

(2) `ringering`

(3) `ringeringeding`

(b) Zeigen Sie, dass der erste Satz des Fuchses aus dem Lied *What does the fox say* von Ylvis in der von der Grammatik G spezifizierten Sprache $\mathcal{L}(G)$ liegt. Sollte Ihnen der Satz nicht geläufig sein, hier ist er: „ringdingdingdingdingeringeding“.

(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 `ring`.

(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 $\mathcal{L}(G)$:

$$A \Rightarrow_P \texttt{ring}\, B \Rightarrow_P \texttt{ring}\, A \Rightarrow_P \texttt{ring}\,\texttt{ring}$$

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

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

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 $\mathcal{L}(G)$:

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

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

$$\begin{aligned} A &\Rightarrow \texttt{ring}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\, C \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ring}\, C \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ring}\,\texttt{e}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ring}\,\texttt{e}\,\texttt{ding}\, B \\ &\Rightarrow \texttt{ring}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{ding}\,\texttt{e}\,\texttt{ring}\,\texttt{e}\,\texttt{ding}\,\varepsilon \\ &= \texttt{ringdingdingdingdingeringeding} \end{aligned}$$

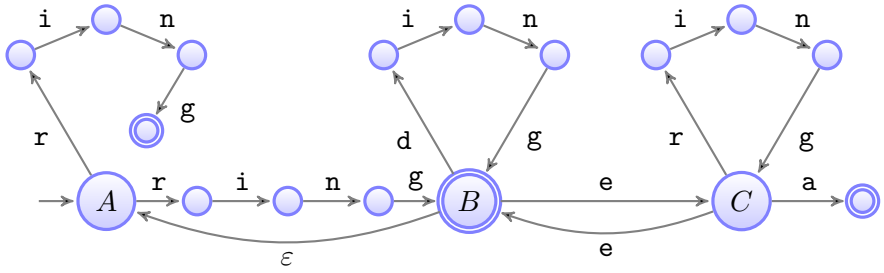
(c) (1) Richtig. Beide Produktionen für das Startsymbol A erzeugen ein Wort, das mit `ring` beginnt.

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

$$A \Rightarrow \texttt{ring}\,B \Rightarrow \texttt{ringring}\,B \Rightarrow \cdots \Rightarrow \texttt{ring}\,(\texttt{ding})^n\,B \Rightarrow \texttt{ring}\,(\texttt{ding})^n$$

(3) Falsch. Gegenbeispiele sind die Wörter `ringdingdingdingdingeringeding` und `ringeringeding`, 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 9 (0.3 Punkte)

Die lineare Optimierung (auch lineare Programmierung genannt) beschäftigt sich mit der Optimierung linearer Zielfunktionen über Mengen, die durch lineare Gleichungen und Ungleichungen beschrieben werden können, das heißt, durch Gleichungen und Ungleichungen folgenden Typs:

$$a_1 \cdot v_1 + a_2 \cdot v_2 + a_3 \cdot v_3 + \cdots = b$$
$$a_1 \cdot v_1 + a_2 \cdot v_2 + a_3 \cdot v_3 + \cdots \leq b$$

wobei a_1, a_2, a_3, ... und b reelle Konstanten und v_1, v_2, v_3 Variablen sind.
Damit solche Optimierungsprobleme durch den Computer verarbeitet werden können, müssen die (Un)Gleichungen in maschinenlesbare Form gebracht werden. Variablen werden dabei durch das Symbol `v` gefolgt von einer oder mehreren Dezimalziffern dargestellt. Die reellen Konstanten werden durch arithmetische Ausdrücke dargestellt, die aus ganzzahligen Numeralen und Konstantensymbolen mittels Addition, Subtraktion, Multiplikation, Division und Klammern gebildet werden können. Konstantensymbole werden durch das Symbol `c` gefolgt von einer oder mehreren Dezimalziffern dargestellt. Die arithmetischen Operationen werden durch die Symbole `+`, `-`, `*`, `/`, `=` bzw. `<=` dargestellt. Die einzelnen Gleichungen und Ungleichungen werden durch einen Strichpunkt (`;`) getrennt.

Beispiel eines linearen Programms:

```
7*v1 + 1*v2 + c3*v3 <= 30;
(3*c100)*v3 + ((1-c1)/2)*v2 = 42;
41*v1 + (1-c1)*v2 <= (c2+1/(10-c3))
```

Beschreiben Sie derartige lineare Programme durch eine kontextfreie Grammatik. Verwenden Sie so weit wie möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten.

Lösung

$\langle V, T, P, \mathit{LinProg}\rangle$ mit

$V = \{\mathit{LinProg}, \mathit{Constraint}, \mathit{Left}, \mathit{Term}, \mathit{Right}, \mathit{CTerm}, \mathit{Op}, \mathit{Num}, \mathit{Const}, \mathit{Var}, \mathit{Digits}, \mathit{Digit}\}$

$T = \{\texttt{"0"}, \ldots, \texttt{"9"}, \texttt{"+"}, \texttt{"-"}, \texttt{"*"}, \texttt{"/"}, \texttt{"<"}, \texttt{"="}, \texttt{";"}, \texttt{"c"}, \texttt{"v"}, \texttt{"("}, \texttt{")"}\}$

$P = \{$

LinProg	$\rightarrow$	*Constraint* $\{$ `";"` *Constraint* $\}$,
Constraint	$\rightarrow$	*Left* (`"="` $\mid$ `"<="`) *Right* ,
Left	$\rightarrow$	*Term* $\{$ `"+"` *Term* $\}$,
Term	$\rightarrow$	*CTerm* `"*"` *Var* ,
Right	$\rightarrow$	*CTerm* ,
CTerm	$\rightarrow$	*Num* $\mid$ *Const* $\mid$ `"("` *CTerm* `")"` $\mid$ *CTerm Op CTerm* ,
Op	$\rightarrow$	`"+"` $\mid$ `"-"` $\mid$ `"*"` $\mid$ `"/"` ,
Num	$\rightarrow$	[`"-"`] *Digits* ,
Const	$\rightarrow$	`"c"` *Digits* ,
Var	$\rightarrow$	`"v"` *Digits* ,
Digits	$\rightarrow$	*Digit* $\{$ *Digit* $\}$,
Digit	$\rightarrow$	`"0"` $\mid \cdots \mid$ `"9"` $\}$

Aufgabe 10 (0.4 Punkte)

In wissenschaftlichen Arbeiten aus dem Bereich der formalen Sprachen ist folgende Definition zu finden:

> Eine *pure Grammatik* G ist ein 3-Tupel $\langle \Sigma, P, S\rangle$, wobei Σ ein endliches Alphabet, $S \subseteq \Sigma^*$ eine endliche Menge von Wörtern über Σ und $P \subseteq \Sigma \times \Sigma^*$ eine endliche Menge von Wortpaaren ist. Die Elemente von P werden Produktionen genannt; statt (x, y) wird auch $x \to y$ geschrieben.
>
> Das Wort uyv ist aus dem Wort uxv in einem Schritt ableitbar, geschrieben $uxv \Rightarrow uyv$, wenn P die Produktion $x \to y$ enthält. Die von G generierte Sprache $\mathcal{L}(G)$ ist definiert als die Menge $\{ w \in \Sigma^* \mid s \stackrel{*}{\Rightarrow} w$ für ein Wort $s \in S \}$, wobei $\stackrel{*}{\Rightarrow}$ den reflexiven und transitiven Abschluss von $\Rightarrow$ bezeichnet.[2]

[2] Das heißt, dass $\stackrel{*}{\Rightarrow}$ die kleinste Relation mit folgenden Eigenschaften ist:

- Aus $u \Rightarrow v$ folgt $u \stackrel{*}{\Rightarrow} v$.
- Es gilt $u \stackrel{*}{\Rightarrow} u$ für alle Wörter $u \in \Sigma^*$.

(a) Geben Sie an, welche der folgenden Tupeln der Form nach pure Grammatiken sind. Begründen Sie Ihre Antwort, wenn das Tupel keine pure Grammatik darstellt.

(1) $\langle \{\mathtt{S}, \mathtt{A}\}, \{\mathtt{S} \to \mathtt{SA}\}, \mathtt{S} \rangle$

(2) $\langle \{\mathtt{S}, \mathtt{A}\}, \{\mathtt{S} \to \mathtt{SA}\}, \{\mathtt{A}\} \rangle$

(3) $\langle \{\mathtt{S}\}, \{\mathtt{S} \to \mathtt{aSb}, \mathtt{S} \to \varepsilon\}, \{\mathtt{S}\} \rangle$, wobei ε für das Leerwort steht.

(4) $\langle \{\mathtt{S}, \mathtt{a}, \mathtt{b}\}, \{\mathtt{S} \to \mathtt{aSb}, \mathtt{S} \to \varepsilon\}, \{\mathtt{aS}, \mathtt{Sb}\} \rangle$, wobei ε für das Leerwort steht.

(b) Sei G die pure Grammatik $\langle \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a} \to \mathtt{ab}\}, \{\mathtt{ab}, \mathtt{ba}\} \rangle$. Zeigen Sie, dass das Wort $\mathtt{babb}$ in der von G generierten Sprache liegt. Wie sehen die Wörter in $\mathcal{L}(G)$ aus? Beschreiben Sie $\mathcal{L}(G)$ mit Hilfe eines regulären Ausdrucks.

(c) Beschreiben Sie eine Methode, um zu einer puren Grammatik G eine kontextfreie Grammatik G' zu erhalten, für die $\mathcal{L}(G') = \mathcal{L}(G)$ gilt.

Lösung

(a) (1) Ist keine pure Grammatik, da die dritte Komponente ein einzelnes Element aus Σ bzw. Σ^* ist, aber eine Menge sein sollte.

(2) Ist eine pure Grammatik.

(3) Ist keine pure Grammatik, da die Produktionen Symbole enthalten, die nicht in der ersten Komponente (Σ) angeführt sind.

(4) Ist eine pure Grammatik.

(b) $\mathtt{ba} \Rightarrow \mathtt{bab} \Rightarrow \mathtt{babb}$

$\mathcal{L}(G) = \{\mathtt{ba}\} \cdot \{\mathtt{b}\}^* \cup \{\mathtt{a}\} \cdot \{\mathtt{b}\}^* \cdot \{\mathtt{b}\} = (\{\mathtt{ab}, \mathtt{ba}\} \cdot \{\mathtt{b}\}^*$ oder
$\mathcal{L}(G) = \mathcal{L}(\mathtt{ba\,b}^* + \mathtt{a\,b}^*\,\mathtt{b}) = \mathcal{L}((\mathtt{ab} + \mathtt{ba})\mathtt{b}^*)$

(c) Die grundsätzliche Idee ist, die Symbole der puren Grammatik zunächst als Nonterminale zu behandeln und diese erst zuletzt in entsprechende Terminale übergehen zu lassen. Zusätzlich ist eine neue Startvariable A erforderlich, die durch die Wörter aus S ersetzt werden kann.

Formal lässt sich diese Transformation folgendermaßen beschreiben. Für jedes Symbol $s \in \Sigma$ führen wir ein Nonterminalsymbol V_s ein. Sei n die Abbildung, die jedem Symbol s das zugehörige Nonterminal V_s zuordnet. Für Wörter definieren $n(s_1 \cdots s_k) = n(s_1) \cdots n(s_k)$. Dann lässt sich die kontextfreie Grammatik G' definieren als $\langle N, \Sigma, P', A \rangle$, wobei

$$N = \{\, V_s \mid s \in \Sigma \,\} \cup \{A\}$$
$$P' = \{\, n(x) \to n(y) \mid x \to y \in P \,\} \cup \{\, V_s \to s \mid s \in \Sigma \,\} \cup \{\, A \to n(x) \mid x \in S \,\}$$

- Aus $u \overset{*}{\Rightarrow} v$ und $v \overset{*}{\Rightarrow} w$ folgt $u \overset{*}{\Rightarrow} w$.

Anschaulich gesprochen steht $\overset{*}{\Rightarrow}$ für die Ableitbarkeit in beliebig vielen Schritten.

Beispielsweise erhält man zur puren Grammatik der vorigen Teilaufgabe die kontextfreie Grammatik $\langle \{V_{\mathtt{a}}, V_{\mathtt{b}}, A\}, \{\mathtt{a}, \mathtt{b}\}, P', A\rangle$, wobei

$$P' = \{V_{\mathtt{a}} \to V_{\mathtt{a}}V_{\mathtt{b}},\ V_{\mathtt{a}} \to \mathtt{a},\ V_{\mathtt{b}} \to \mathtt{b},\ A \to V_{\mathtt{a}}V_{\mathtt{b}} \mid V_{\mathtt{b}}V_{\mathtt{a}}\}\ .$$

Aufgabe 11 (0.4 Punkte)

Zeigen Sie mit Hilfe der algebraischen Gesetze für logische Operatoren und Quantoren, dass die beiden Formeln jeweils äquivalent zueinander sind. Übersetzen Sie jede Formel in natürliche Sprache, wobei die Prädikatensymbole die durch ihren Namen nahegelegten Bedeutungen haben sollen.

(a) $\neg\exists x\,(\mathit{Mensch}(x) \wedge \forall y\,(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$ und
$\forall x\,(\mathit{Mensch}(x) \supset \exists y\,(\mathit{Sprache}(y) \wedge \neg\mathit{Beherrscht}(x,y)))$

(b) $\exists x\,(\mathit{Sprache}(x) \wedge \forall y\,\mathit{Mensch}(y) \wedge \exists z\,\mathit{Beherrscht}(y,x))$ und
$\forall z\,\mathit{Mensch}(z) \wedge \neg\forall z\,(\mathit{Sprache}(z) \supset \neg\mathit{Beherrscht}(y,z))$

Lösung

(a)

$\neg\exists x\,(\mathit{Mensch}(x) \wedge \forall y\,(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$	$\neg\exists xF = \forall x\neg F$
$\forall x\,\neg(\mathit{Mensch}(x) \wedge \forall y\,(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$	$\neg(F \wedge G) = \neg F \vee \neg G$
$\forall x\,(\neg\mathit{Mensch}(x) \vee \neg\forall y\,(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$	$\neg F \vee G = F \supset G$
$\forall x\,(\mathit{Mensch}(x) \supset \neg\forall y\,(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$	$\neg\forall xF = \exists x\neg F$
$\forall x\,(\mathit{Mensch}(x) \supset \exists y\,\neg(\mathit{Sprache}(y) \supset \mathit{Beherrscht}(x,y)))$	$F \supset G = \neg F \vee G$
$\forall x\,(\mathit{Mensch}(x) \supset \exists y\,\neg(\neg\mathit{Sprache}(y) \vee \mathit{Beherrscht}(x,y)))$	$\neg(F \vee G) = \neg F \wedge \neg G$
$\forall x\,(\mathit{Mensch}(x) \supset \exists y\,(\neg\neg\mathit{Sprache}(y) \wedge \neg\mathit{Beherrscht}(x,y)))$	$\neg\neg F = F$
$\forall x\,(\mathit{Mensch}(x) \supset \exists y\,(\mathit{Sprache}(y) \wedge \neg\mathit{Beherrscht}(x,y)))$	

Die erste Formel lässt sich lesen als „Es gibt keinen Menschen, der alle Sprachen beherrscht“ oder „Niemand beherrscht alle Sprachen“, die letzte als „Für jeden Menschen gibt es (mindestens) eine Sprache, die er nicht beherrscht“. Die Zwischenformel in der vierten Zeile entspricht der Formulierung „Für alle Menschen gilt, dass sie nicht alle Sprachen beherrschen“.

(b) Beachten Sie, dass sich der Quantor $\forall y$ nur auf die Formel $\mathit{Mensch}(y)$ bezieht, nicht aber auf die nachfolgende Formel; andernfalls wären Klammern erforderlich.

$\exists x\,(\mathit{Sprache}(x) \wedge \forall y\,\mathit{Mensch}(y) \wedge \exists z\,\mathit{Beherrscht}(y,x))$	$\exists zF = F$ wenn z nicht in F vorkommt
$\exists x\,(\mathit{Sprache}(x) \wedge \forall y\,\mathit{Mensch}(y) \wedge \mathit{Beherrscht}(y,x))$	$F \wedge G = G \wedge F$
$\exists x\,(\forall y\,\mathit{Mensch}(y) \wedge \mathit{Sprache}(x) \wedge \mathit{Beherrscht}(y,x))$	$\exists x(F \wedge G) = F \wedge \exists xG$ wenn x nicht in F vorkommt
$\forall y\,\mathit{Mensch}(y) \wedge \exists x\,(\mathit{Sprache}(x) \wedge \mathit{Beherrscht}(y,x))$	$\forall yF[y] = \forall zF[z]$ wenn z nicht in F vorkommt
$\forall z\,\mathit{Mensch}(z) \wedge \exists x\,(\mathit{Sprache}(x) \wedge \mathit{Beherrscht}(y,x))$	$\exists xF[x] = \exists zF[z]$ wenn z nicht in F vorkommt
$\forall z\,\mathit{Mensch}(z) \wedge \exists z\,(\mathit{Sprache}(z) \wedge \mathit{Beherrscht}(y,z))$	$F = \neg\neg F$
$\forall z\,\mathit{Mensch}(z) \wedge \neg\neg\exists z\,(\mathit{Sprache}(z) \wedge \mathit{Beherrscht}(y,z))$	$\neg\exists xF = \forall x\neg F$
$\forall z\,\mathit{Mensch}(z) \wedge \neg\forall z\,\neg(\mathit{Sprache}(z) \wedge \mathit{Beherrscht}(y,z))$	$\neg(F \wedge G) = \neg F \vee \neg G$
$\forall z\,\mathit{Mensch}(z) \wedge \neg\forall z\,(\neg\mathit{Sprache}(z) \vee \neg\mathit{Beherrscht}(y,z))$	$\neg F \vee G = F \supset G$
$\forall z\,\mathit{Mensch}(z) \wedge \neg\forall z\,(\mathit{Sprache}(z) \supset \neg\mathit{Beherrscht}(y,z))$	

Die erste Formel entspricht der etwas wirren Aussage:

> Es gibt eine Sprache, sodass alles ein Mensch ist und es etwas gibt, sodass y diese Sprache beherrscht.

Auffällig an dieser Formel ist, dass es eine freie Variable y für ein Element des Universiums gibt, das die Sprache x beherrscht. Diese Variable hat nichts mit der gleichnamigen Variable in der Teilformel $\forall y \mathit{Mensch}(y)$ zu tun. Weiters gibt es noch die quantifizierte Variable z, die aber nirgends vorkommt, wodurch der entsprechende Quantor bedeutungslos ist.

Die letzte Formel kann gelesen werden als:

> Alles ist ein Mensch,
> und nicht alle Sprachen werden von y nicht beherrscht.

Die vierte, fünfte und sechste Formel unterscheiden sich nur durch die Benennung der quantifizierten Variablen und sind am einfachsten interpretierbar:

> Alles ist ein Mensch,
> und y beherrscht (mindestens) eine Sprache.

Beachten Sie, dass die freie Variable y nicht umbenannt werden darf, da das die Bedeutung der Formel ändern würde.

Aufgabe 12 (0.6 Punkte)

Seien *Liest*, *Student*, *Dick* und *Buch* Prädikatensymbole und *nibelungen* sowie *comic* Konstantensymbole mit folgender Bedeutung:

$\mathit{Liest}(x, y)$	... x liest y	$\mathit{Buch}(x)$	... x ist ein Buch
$\mathit{Student}(x)$	... x ist ein Student	*nibelungen*	... Nibelungen
$\mathit{Dick}(x)$	... x ist dick	*comic*	... Comic

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

(a) Manche Studenten lesen nur dann die Nibelungen, wenn sie auch dicke Bücher lesen.

(b) Dicke Studenten lesen einige Bücher aber keine Comics.

Sei weiters folgende Interpretation gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Martin}, \text{Nina}, \text{Karl}, \text{Iris}, \text{Taschenbuch}, \text{Comic}, \text{Wörterbuch}, \text{Liederbuch}, \text{Roman}, \\
&\quad \text{Krimi}, \text{Sachbuch}\} \\
I(\mathit{Student}) &= \{\text{Martin}, \text{Nina}, \text{Iris}\} \\
I(\mathit{Buch}) &= \{\text{Taschenbuch}, \text{Comic}, \text{Wörterbuch}, \text{Liederbuch}, \text{Roman}\} \\
I(\mathit{Dick}) &= \{\text{Wörterbuch}, \text{Roman}, \text{Krimi}\} \\
I(\mathit{Liest}) &= \{(\text{Martin}, \text{Comic}), (\text{Martin}, \text{Liederbuch}), \\
&\quad (\text{Nina}, \text{Roman}), (\text{Nina}, \text{Krimi}), \\
&\quad (\text{Iris}, \text{Roman}), (\text{Iris}, \text{Liederbuch}), (\text{Iris}, \text{Krimi}), \\
&\quad (\text{Karl}, \text{Comic}), (\text{Karl}, \text{Krimi})\} \\
I(\mathit{comic}) &= \text{Comic} \\
I(\mathit{krimi}) &= \text{Krimi} \\
I(\mathit{roman}) &= \text{Roman}
\end{aligned}$$

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) $\forall x \, (\mathit{Liest}(x, \mathit{comic}) \not\equiv \mathit{Liest}(x, \mathit{roman}))$

(d) $\exists x \, (\mathit{Liest}(x, \mathit{comic}) \wedge \neg \mathit{Liest}(x, \mathit{krimi}))$

(e) $\forall x \, (\mathit{Student}(x) \supset \exists y \, (\mathit{Dick}(y) \wedge \mathit{Liest}(x, y)))$

(f) $\exists x \, \exists y \, (\mathit{Student}(x) \wedge \mathit{Buch}(y) \wedge \neg \mathit{Liest}(x, y))$

Lösung

(a) $\exists x(\mathit{Student}(x) \wedge (\mathit{Liest}(x, \mathit{nibelungen}) \supset \exists y(\mathit{Buch}(y) \wedge \mathit{Dick}(y) \wedge \mathit{Liest}(x, y))))$

(b) $\forall x((\mathit{Student}(x) \wedge \mathit{Dick}(x)) \supset (\exists y(\mathit{Buch}(y) \wedge \mathit{Liest}(x, y)) \wedge \neg \mathit{Liest}(x, \mathit{comic})))$

(c) Übersetzung: Alles liest einen Comic oder einen Roman (aber nicht beides).
Diese Aussage ist falsch, da $(\text{Comic}, \text{Comic}) \notin I(\mathit{Liest})$ und $(\text{Comic}, \text{Roman}) \notin I(\mathit{Liest})$.

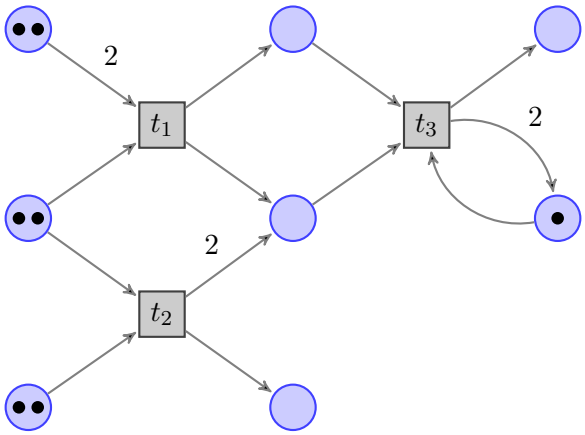
(d) Übersetzung: Jemand (etwas) liest einen Comic aber keinen Krimi.
Diese Aussage ist wahr, da sie auf Martin zutrifft: $(\text{Martin}, \text{Comic}) \in I(\mathit{Liest})$ und $(\text{Martin}, \text{Krimi}) \notin I(\mathit{Liest})$.

(e) Übersetzung: Alle Studenten lesen etwas Dickes.
Diese Aussage ist falsch, da Martin Student ist, aber nur Comics und Liederbücher liest, die beide nicht dick sind.

(f) Übersetzung: Es gibt einen Studenten, der eines der Bücher nicht liest.
Diese Aussage ist wahr, da zum Beispiel $(\text{Martin}, \text{Taschenbuch}) \notin I(\mathit{Liest})$.

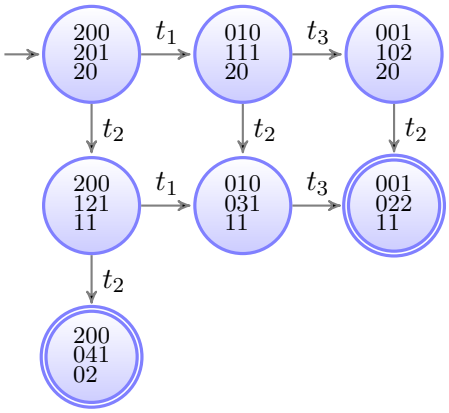
Aufgabe 13 (0.3 Punkte)

Gegeben sei das folgende Petri-Netz mit Anfangsmarkierung. Geben Sie alle möglichen Reihenfolgen an, in denen die Transitionen feuern können. Geben Sie jene erreichbaren Markierungen an, in denen keine Transition aktiviert ist.

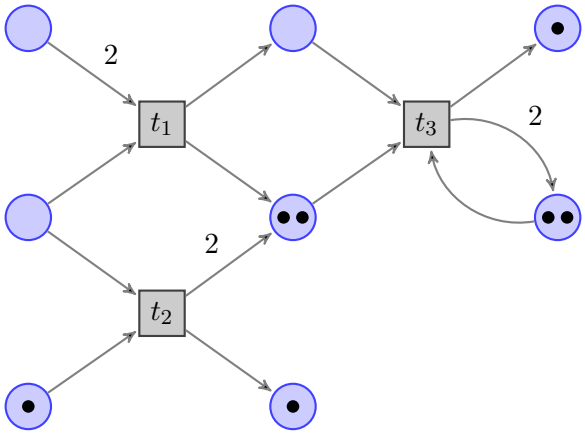


Lösung

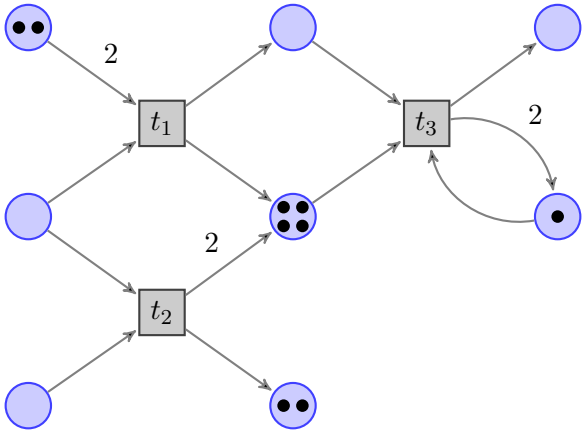
Die Abläufe im Petrinetz lassen sich durch einen (in diesem Fall endlichen) Automaten beschreiben. Die Markierungen bilden die Zustände, die Transitionen sind das Alphabet. Erhält man aus einer Markierung m durch Feuern einer Transition t eine Markierung m', dann gibt es einen Übergang beschriftet mit t vom Zustand für m zu jenem für m'. Wenn wir Deadlocks (d.h. Markierungen, in denen keine Transition aktiviert ist) als Endzustände auffassen, ist die Sprache des Automaten genau die Menge der Transitionsfolgen, die zu einem Deadlock führen.
Wir stellen jede Markierung durch eine Zahlenmatrix dar, in der die Einträge analog den Stellen im Petrinetz angeordnet sind. Jeder Eintrag gibt die Zahl der Marken in der entsprechenden Stelle an. Wir erhalten den folgenden Automaten.



Die Sprache des Automaten besteht aus vier Wörtern. Nach den Transitionsfolgen $t_1t_2t_3$, $t_1t_3t_2$ und $t_2t_1t_3$ sieht das Petrinetz folgendermaßen aus:



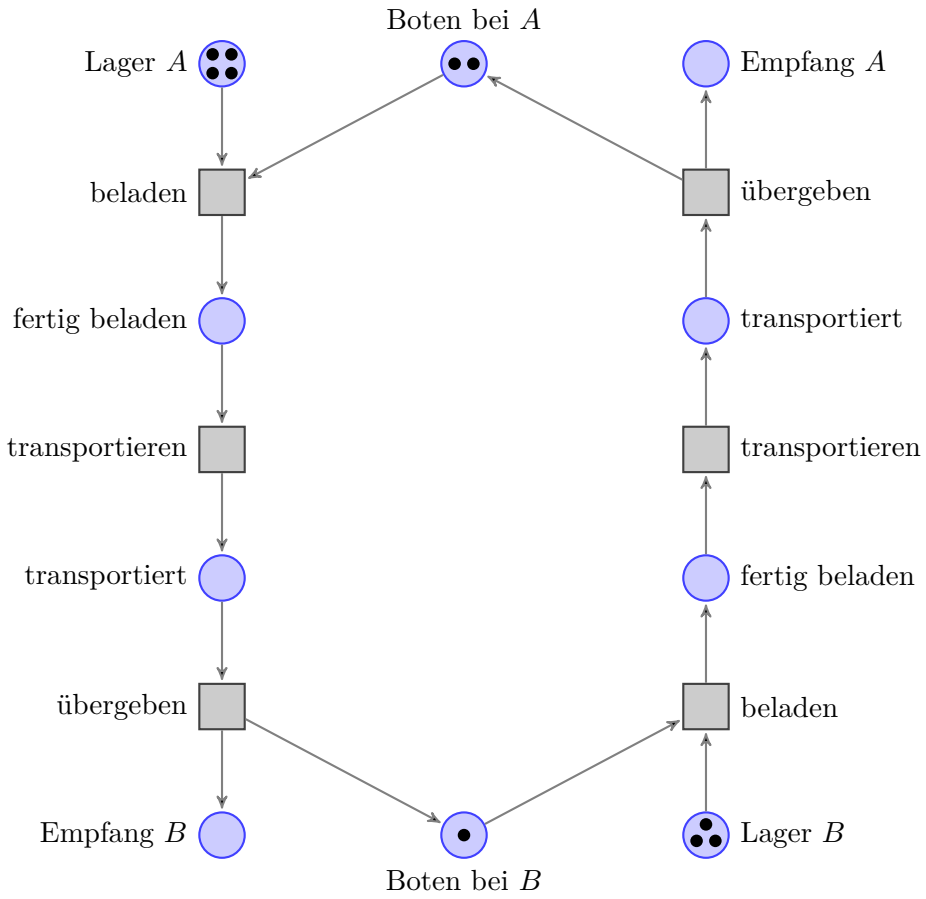
Nach der Transitionsfolge $t_2 t_2$ ergibt sich folgende Situation:



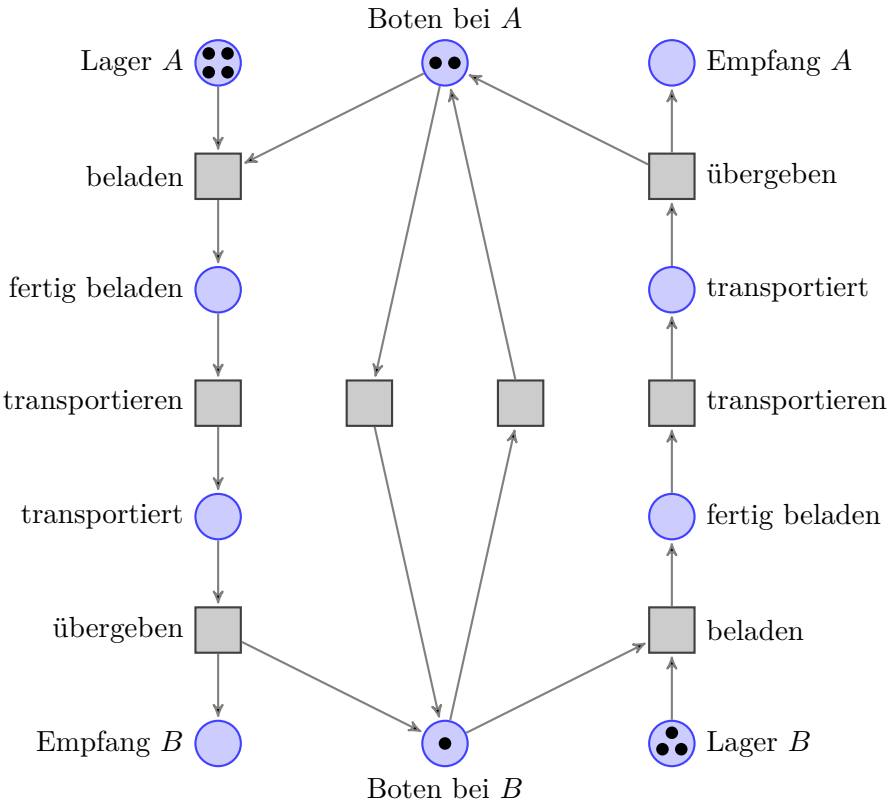
Aufgabe 14 (0.4 Punkte)

Zwei Unternehmen, A und B, besitzen jeweils ein Lager und einen Warenempfang. Fahrradboten transportieren Pakete vom Lager des einen Unternehmens zum Warenempfang des anderen. Ein Transport läuft immer nach demselben Muster ab: Beim Lager des einen Unternehmens wird das Fahrrad mit einem Paket beladen, dann fährt der Bote damit zum anderen Unternehmen, wo er es dem Empfang übergibt. Danach ist der Bote bereit, ein Paket von diesem Unternehmen zurück zum ersten Unternehmen zu transportieren. Jeder Bote bringt also abwechselnd ein Paket vom A nach B und danach ein anderes von B nach A (bzw. umgekehrt, wenn er beim Unternehmen B beginnt).
Modellieren Sie dieses System mit Hilfe eines Petri-Netzes. Wählen Sie die Anfangsmarkierung so, dass zu Beginn 4 Pakete im Lager A und 3 Pakete im Lager B auf Transport warten. Weiters stehen anfänglich zwei Boten beim Unternehmen A und einer beim Unternehmen B für Transporte zur Verfügung.
Dieses System hat den Nachteil, dass Boten unter Umständen untätig bei einem Unternehmen auf Pakete warten, während beim anderen Boten benötigt werden. Wie lässt sich Ihr Petri-Netz erweitern, um dieses Problem zu lösen?

Lösung



Damit die Boten zum anderen Unternehmen wechseln können, auch wenn keine Pakete vorliegen, muss man Leerfahrten zulassen.


Boten bei A
Lager A
Empfang A
beladen
übergeben
fertig beladen
transportiert
transportieren
transportieren
transportiert
fertig beladen
übergeben
beladen
Empfang B
Lager B
Boten bei B