

2. Übungsblatt (mit Lösungen)

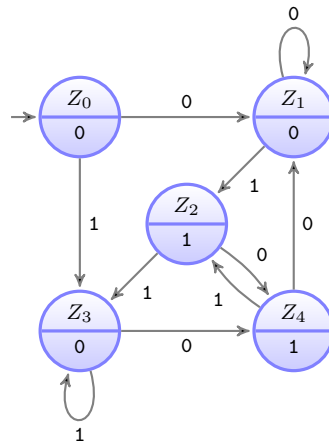
3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

Juni 2014

Aufgabe 1 (0.3 Punkte)

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



- (a) Geben Sie die Ausgaben zu folgenden Eingaben an: 00111, 10101, 01101.
- (b) Berechnen Sie schrittweise $\delta^*(Z_0, 01001)$ und $\gamma^*(Z_0, 01001)$.
- (c) Beschreiben Sie die Übersetzungsfunktion $[\mathcal{A}]$.

Lösung

(a)
$$\begin{array}{l} w: \quad 00111 \quad 10101 \quad 01101 \\ \hline [\mathcal{A}](w): 00100 \quad 01111 \quad 01011 \end{array}$$

(b)
$$\begin{aligned} \delta^*(Z_0, 01001) &= \delta^*(\delta(Z_0, 0), 1001) = \delta^*(Z_1, 1001) \\ &= \delta^*(\delta(Z_1, 1), 001) = \delta^*(Z_2, 001) \\ &= \delta^*(\delta(Z_2, 0), 01) = \delta^*(Z_4, 01) \\ &= \delta^*(\delta(Z_4, 0), 1) = \delta^*(Z_1, 1) \\ &= \delta^*(\delta(Z_1, 1), \varepsilon) = \delta^*(Z_2, \varepsilon) \\ &= Z_2 \end{aligned}$$

$$\begin{aligned}
\gamma^*(Z_0, 01001) &= \gamma(\delta(Z_0, 0)) \cdot \gamma^*(\delta(Z_0, 0), 1001) &= \gamma(Z_1) \cdot \gamma^*(Z_1, 1001) \\
&= 0 \cdot \gamma(\delta(Z_1, 1)) \cdot \gamma^*(\delta(Z_1, 1), 001) &= 0 \cdot \gamma(Z_2) \cdot \gamma^*(Z_2, 001) \\
&= 01 \cdot \gamma(\delta(Z_2, 0)) \cdot \gamma^*(\delta(Z_2, 0), 01) &= 01 \cdot \gamma(Z_4) \cdot \gamma^*(Z_4, 01) \\
&= 011 \cdot \gamma(\delta(Z_4, 0)) \cdot \gamma^*(\delta(Z_4, 0), 1) &= 011 \cdot \gamma(Z_1) \cdot \gamma^*(Z_1, 1) \\
&= 0110 \cdot \gamma(\delta(Z_1, 1)) \cdot \gamma^*(\delta(Z_1, 1), \varepsilon) &= 0110 \cdot \gamma(Z_2) \cdot \gamma^*(Z_2, \varepsilon) \\
&= 01101 \cdot \varepsilon &= 01101
\end{aligned}$$

- (c) $[A]$: Jeder Wechsel in der Eingabe von 0 auf 1 bzw. umgekehrt führt zur Ausgabe einer 1, während bei gleichbleibender Eingabe 0 ausgegeben wird. Der Automat ist also ein Flanken-Detektor.

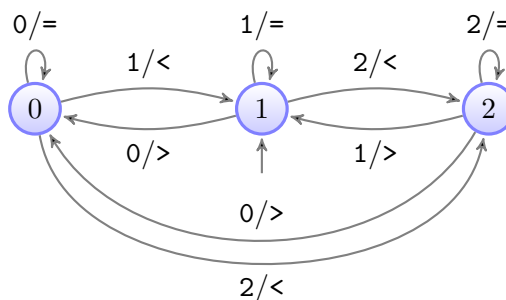
Aufgabe 2 (0.3 Punkte)

Sei compare eine Funktion, die jedem Wort über dem Alphabet $\Sigma = \{0, 1, 2\}$ ein gleichlanges Wort über dem Alphabet $\Gamma = \{<, =, >\}$ zuordnet, das zu jedem Eingabezeichen angibt, ob dessen Zahlenwert gegenüber dem vorherigen angewachsen ($<$), gleich geblieben ($=$) oder kleiner geworden ($>$) ist. Wir nehmen an, dass das erste Zeichen mit 1 verglichen wird und dass das Leerwort auf das Leerwort abgebildet wird. Somit gilt $\text{compare}(\varepsilon) = \varepsilon$, $\text{compare}(0) = >$ und $\text{compare}(1201120) = =<><=<>$. Die Ausgabe lässt sich auch so verstehen, dass eine gültige Aussagenkette entsteht, wenn man der Eingabe einen 1er voranstellt und zwischen je zwei Eingabesymbole ein Ausgabesymbol schreibt; für die Eingabe 1201120 ergibt das die Kette $1 = 1 < 2 > 0 < 1 = 1 < 2 > 0$.

- (a) Geben Sie einen Mealy-Automaten mit der Übersetzungsfunktion compare an.
(b) Geben Sie einen Moore-Automaten mit der Übersetzungsfunktion compare an.
(c) Skizzieren Sie eine allgemeine Methode, mit der man zu einem beliebigen Mealy-Automaten einen äquivalenten Moore-Automaten erhält.

Lösung

- (a) Wir benötigen die Zustände $Q = \{0, 1, 2\}$, um das jeweils zuletzt verarbeitete Eingabesymbol 0, 1 bzw. 2 zu speichern und damit das nächstfolgende Symbol vergleichen zu können. Graphisch lässt sich der Mealy-Automat folgendermaßen darstellen:

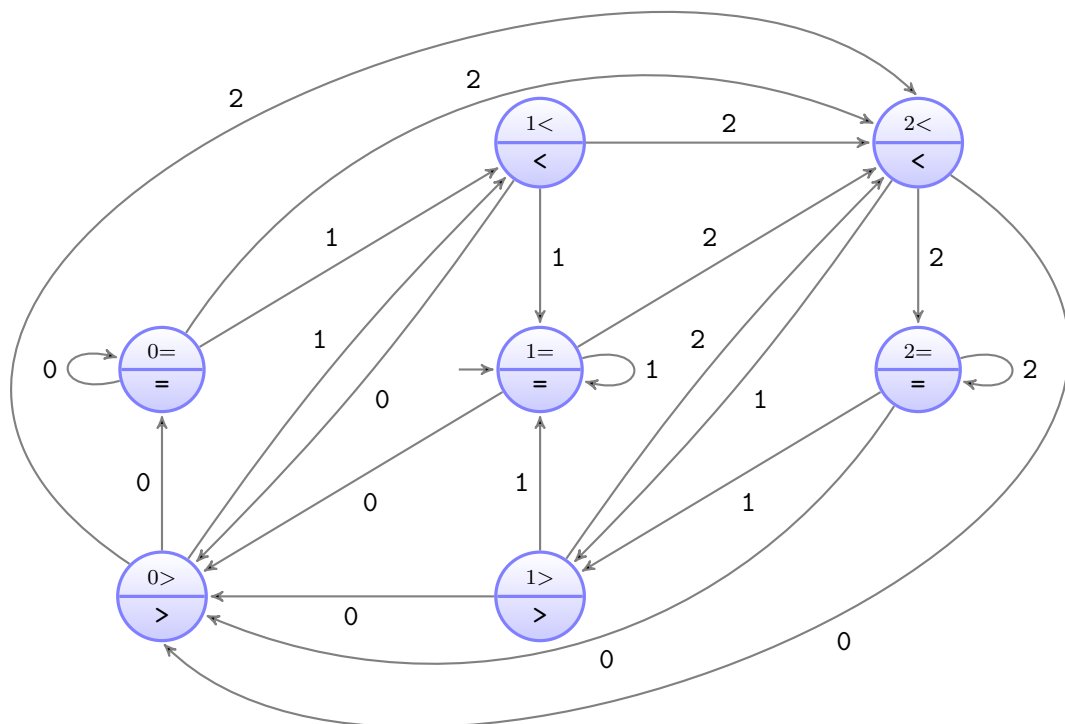


Alternativ lässt sich der Automat durch das Tupel $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, \gamma, 1 \rangle$ beschreiben, wobei die Übergangsfunktion δ und die Ausgabefunktion γ durch folgende Tabellen festgelegt sind:

δ	0	1	2	γ	0	1	2
0	0	1	2	0	=	<	<
1	0	1	2	1	>	=	<
2	0	1	2	2	>	>	=

- (b) An den Spalten der γ -Tabelle des Mealyautomaten lässt sich ablesen, dass beim Übergang in den Zustand 0 die Ausgaben = und > auftreten können, beim Übergang in den Zustand 1 alle drei Ausgabesymbole und beim Übergang in den Zustand 2 die Ausgaben < und =. Bei einem Moore-Automaten dürfte in jeder Spalte nur ein Symbol auftreten. Wir teilen daher den Zustand 0 in die Zustände 0= und 0> auf sowie analog den Zustand 1 in drei und den Zustand 2 in zwei neue Zustände. Von jedem neuen Zustand führt dieselbe Zahl an Übergängen weg wie vom Ursprungszustand, das Ziel der Übergänge hängt aber von der Ausgabe ab.

Graphisch lässt sich der Moore-Automat folgendermaßen darstellen:



Alternativ lässt sich der Automat durch das Tupel $\langle \bar{Q}, \Sigma, \Gamma, \bar{\delta}, \bar{\gamma}, 1= \rangle$ beschreiben, wobei \bar{Q} die Zustandsmenge $\{0=, 0>, 1<, 1=, 1>, 2<, 2=\}$ ist und die Übergangs- bzw. Ausgabefunktion durch folgende Tabellen festgelegt werden:

$\bar{\delta}$	0	1	2	q	$\bar{\gamma}(q)$
0=	0=	1<	2<	0=	=
0>	0=	1<	2<	0>	>
1<	0>	1=	2<	1<	<
1=	0>	1=	2<	1=	=
1>	0>	1=	2<	1>	>
2<	0>	1>	2=	2<	<
2=	0>	1>	2=	2=	=

(c) Die Umwandlung eines Mealy- in einen Moore-Automaten erfolgt in folgenden Schritten:

- Jeder Zustand q wird in n neue Zustände q_{o_1}, \dots, q_{o_n} aufgespaltet, wobei o_1, \dots, 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}, \dots, 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)

Vereinfachen Sie die folgenden Ausdrücke.

(a) $((\{x, xy, zz\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, yy\} \cup \{y\})) \cup \{z\}$

- (b) $\{x, y, xy\} \cdot \{\}$
- (c) $\{x, y, xy\} \cup \{\}$
- (d) $\{x\}^* \cdot (\{x\} \cup \{\varepsilon\})$
- (e) $(\{xy\}^* \cdot \{x\} \cdot \{yx\}^* \cdot \{y\}) \cdot \{x\}^*$

Lösung

- (a) $((\{x, xy, zz\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, yy\} \cup \{y\})) \cup \{z\}$
 $= (\{\varepsilon, x, xy, zz\} \cdot \{\varepsilon, y, yy\}) \cup \{z\}$
 $= \{\varepsilon, x, xy, xyy, xyyy, y, yy, zz, zzy, zzyy\} \cup \{z\}$
 $= \{\varepsilon, x, xy, xyy, xyyy, y, yy, z, zz, zzy, zzyy\}$
- (b) $\{x, y, xy\} \cdot \{\} = \{\}$
- (c) $\{x, y, xy\} \cup \{\} = \{x, y, xy\}$
- (d) $\{x\}^* \cdot (\{x\} \cup \{\varepsilon\}) = \{x\}^+ \cup \{x\}^* = \{x\}^*$
- (e) $(\{xy\}^* \cdot \{x\} \cdot \{yx\}^* \cdot \{y\}) \cdot \{x\}^*$
 $= \{xy\}^* \cdot \{xy\}^* \cdot \{xy\} \cdot \{x\}^*$
 $= \{xy\}^* \cdot \{xy\} \cdot \{x\}^*$
 $= \{xy\}^+ \cdot \{x\}^*$

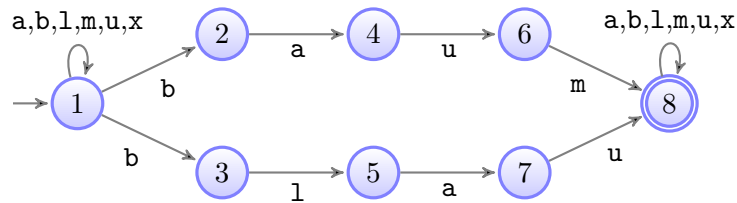
Aufgabe 4 (0.3 Punkte)

Sei Σ das Alphabet $\{a, b, l, m, u, x\}$ und L die Menge aller Wörter über Σ , in denen **baum** oder **blau** als Teilwort vorkommt.

- (a) Geben Sie einen Posix Extended Regular Expression an, der die Sprache L beschreibt.
- (b) Geben Sie einen nichtdeterministischen Automaten an, der die Sprache L akzeptiert. Der Automat soll der Definition der Sprache direkt entsprechen, sodass die Korrektheit der Modellierung unmittelbar einsichtig ist.
- (c) Konstruieren Sie mit Hilfe des in der Vorlesung besprochenen Determinisierungsverfahrens einen zu Ihrem nichtdeterministischen Automaten äquivalenten deterministischen.

Lösung

- (a) $[ablmux]^*(baum|blau)[ablmux]^*$
- (b)



(c) Wir stellen zunächst die Übergangsfunktion des indeterministischen Automaten als Tabelle dar.

δ	a	b	l	m	u	x
1	{1}	{1, 2, 3}	{1}	{1}	{1}	{1}
2	{4}	{}	{}	{}	{}	{}
3	{}	{}	{5}	{}	{}	{}
4	{}	{}	{}	{}	{6}	{}
5	{7}	{}	{}	{}	{}	{}
6	{}	{}	{}	{8}	{}	{}
7	{}	{}	{}	{}	{8}	{}
8	{8}	{8}	{8}	{8}	{8}	{8}

Damit lässt sich nun systematisch die Übergangsfunktion des deterministischen Automaten konstruieren, indem wir die jeweils relevanten Zeilen vereinigen. Wir wählen {1} als Startzustand für den deterministischen Automaten, da 1 der Startzustand des nicht-deterministischen ist.

δ	a	b	l	m	u	x
{1}	{1}	{1, 2, 3}	{1}	{1}	{1}	{1}
{1, 2, 3}	{1, 4}	{1, 2, 3}	{1, 5}	{1}	{1}	{1}
{1, 4}	{1}	{1, 2, 3}	{1}	{1}	{1, 6}	{1}
{1, 5}	{1, 7}	{1, 2, 3}	{1}	{1}	{1}	{1}
{1, 6}	{1}	{1, 2, 3}	{1}	{1, 8}	{1}	{1}
{1, 7}	{1}	{1, 2, 3}	{1}	{1}	{1, 8}	{1}
{1, 8}	{1, 8}	{1, 2, 3, 8}	{1, 8}	{1, 8}	{1, 8}	{1, 8}
{1, 2, 3, 8}	{1, 4, 8}	{1, 2, 3, 8}	{1, 5, 8}	{1, 8}	{1, 8}	{1, 8}
{1, 4, 8}	{1, 8}	{1, 2, 3, 8}	{1, 8}	{1, 8}	{1, 6, 8}	{1, 8}
{1, 5, 8}	{1, 7, 8}	{1, 2, 3, 8}	{1, 8}	{1, 8}	{1, 8}	{1, 8}
{1, 6, 8}	{1, 8}	{1, 2, 3, 8}	{1, 8}	{1, 8}	{1, 8}	{1, 8}
{1, 7, 8}	{1, 8}	{1, 2, 3, 8}	{1, 8}	{1, 8}	{1, 8}	{1, 8}

Die Endzustände sind all jene, die den ursprünglichen Endzustand 8 enthalten, also {1, 8}, {1, 2, 3, 8}, {1, 4, 8}, {1, 5, 8}, {1, 6, 8} und {1, 7, 8}.

In diesem Beispiel haben die Endzustände die Eigenschaft, dass man von ihnen ausgehend mit jedem der sechs möglichen Symbole wieder in einen Endzustand gelangt. Sobald also ein Endzustand erreicht ist, wird ab hier jedes beliebige Wort akzeptiert. Wir können den Automaten also vereinfachen, indem wir die Endzustände zu einem einzigen zusammenfassen, den wir wieder 8 nennen. Abbildung 1 stellt den vereinfachten deterministischen Automaten graphisch dar.

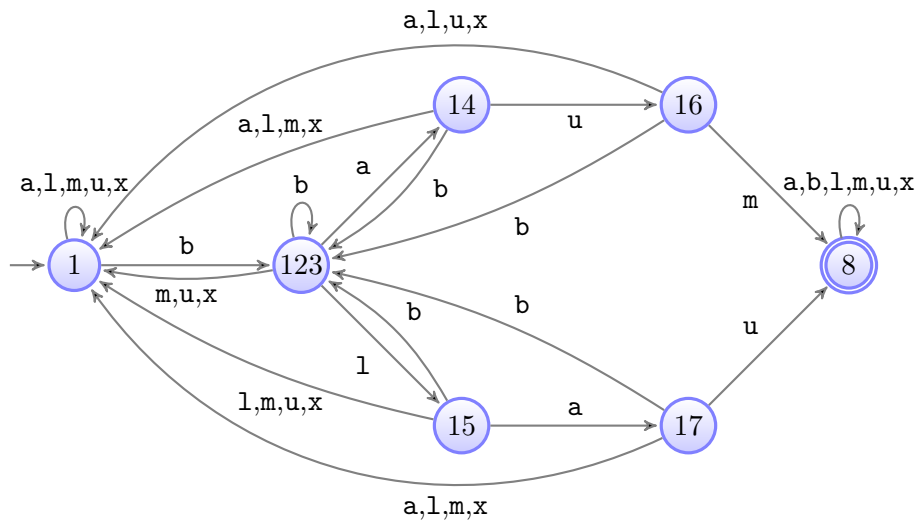


Abbildung 1: Deterministischer Automat für baum und blau (Aufgabe 4)

Aufgabe 5 (0.4 Punkte)

Seien $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, i_1, F_1 \rangle$ und $\mathcal{A}_2 = \langle Q_2, \Sigma, \delta_2, i_2, F_2 \rangle$ zwei beliebige deterministische Automaten über demselben Alphabet Σ . Geben Sie eine Methode an, um daraus einen Automaten \mathcal{A} für den Durchschnitt der beiden zugehörigen Sprachen zu konstruieren, d.h., es soll $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$ gelten.

Hinweis: Überlegen Sie sich die Aufgabenstellung zuerst an Hand einfacher konkreter Automaten und verallgemeinern Sie dann Ihre Beobachtungen. Wählen Sie für den neuen Automaten \mathcal{A} Zustände mit der Bezeichnung q_1q_2 , die signalisieren, dass sich der Automat \mathcal{A}_1 momentan im Zustand q_1 und der Automat \mathcal{A}_2 im Zustand q_2 befindet.

Lösung

Wir konstruieren einen Automaten \mathcal{A} , der die beiden Automaten \mathcal{A}_1 und \mathcal{A}_2 parallel ausführt. Als Zustände für \mathcal{A} verwenden wir Paare (q_1, q_2) , wobei $q_1 \in Q_1$ ein Zustand des ersten Automaten und $q_2 \in Q_2$ ein Zustand des zweiten Automaten ist. Der neue Automat befindet sich bei Eingabe eines Wortes w im Zustand (q_1, q_2) , wenn sich der erste Automat bei diesem Wort im Zustand q_1 und der zweite im Zustand q_2 befinden würde. Der Startzustand (i_1, i_2) entspricht der Situation, in der sich die beiden ursprünglichen Automaten im Startzustand befinden. Ein Übergang mit dem Symbol s von (q_1, q_2) nach (q'_1, q'_2) existiert genau dann, wenn wenn man mit diesem Symbol in \mathcal{A}_1 von q_1 nach $q'_1 = \delta_1(q_1, s)$ und in \mathcal{A}_2 von q_2 nach $q'_2 = \delta_2(q_2, s)$ gelangt. Ein Wort wird von beiden Automaten akzeptiert (und liegt daher im Durchschnitt der Sprachen), wenn der neue Automat einen Zustand (q_1, q_2) erreicht, bei dem beide Zustände Endzustände im jeweiligen Automaten sind, wenn also $q_1 \in F_1$ und $q_2 \in F_2$ gilt. Ein Automat für den

Durchschnitt der Sprachen $\mathcal{L}(\mathcal{A}_1)$ und $\mathcal{L}(\mathcal{A}_2)$ lässt sich somit durch

$$\mathcal{A} = \langle Q_1 \times Q_2, \Sigma, \delta, (i_1, i_2), F_1 \times F_2 \rangle$$

definieren, wobei die Übergangsfunktion festgelegt ist durch

$$\delta((q_1, q_2), s) = (\delta_1(q_1, s), \delta_2(q_2, s)) \quad \text{für alle } (q_1, q_2) \in Q_1 \times Q_2 \text{ und alle } s \in \Sigma.$$

Als Beispiel betrachten wir die Automaten

$$\begin{array}{l} \mathcal{A}_1 = \langle \{1, 2, 3\}, \{\mathbf{a}, \mathbf{b}\}, \delta_1, 1, \{2\} \rangle \\ \mathcal{A}_2 = \langle \{x, y\}, \{\mathbf{a}, \mathbf{b}\}, \delta_2, x, \{y\} \rangle \end{array} \quad \begin{array}{c|cc} \delta_1 & \mathbf{a} & \mathbf{b} \\ \hline 1 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{array} \quad \begin{array}{c|cc} \delta_2 & \mathbf{a} & \mathbf{b} \\ \hline x & y & x \\ y & y & x \end{array}$$

Automat \mathcal{A}_1 akzeptiert alle Wörter über $\Sigma = \{\mathbf{a}, \mathbf{b}\}$, die mit \mathbf{a} beginnen, und Automat \mathcal{A}_2 jene, die mit \mathbf{a} aufhören, d.h., $\mathcal{L}(\mathcal{A}_1) = \{\mathbf{a}\} \cdot \{\mathbf{a}, \mathbf{b}\}^*$ und $\mathcal{L}(\mathcal{A}_2) = \{\mathbf{a}, \mathbf{b}\}^* \cdot \{\mathbf{a}\}$. Das oben beschriebene Verfahren liefert den Automaten

$$\mathcal{A} = \langle \{1x, 1y, 2x, 2y, 3x, 3y\}, \{\mathbf{a}, \mathbf{b}\}, \delta, 1x, \{2y\} \rangle$$

mit der Übergangsfunktion

δ	\mathbf{a}	\mathbf{b}
1x	2y	3x
1y	2y	3x
2x	2y	2x
2y	2y	2x
3x	3y	3x
3y	3y	3x

wobei die Zustandsbezeichnungen (q_1, q_2) auf q_1q_2 verkürzt wurden.¹ Für den Automaten \mathcal{A} gilt $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) = \{\mathbf{a}\} \cdot \{\mathbf{a}, \mathbf{b}\}^* \cdot \{\mathbf{a}\}$, d.h., der Automat akzeptiert alle Wörter, die sowohl mit \mathbf{a} beginnen als auch aufhören. Abbildung 2 stellt die drei Automaten graphisch dar.

Diese Überlegung zeigt, dass der Schnitt von Sprachen, die von Automaten akzeptiert werden, wieder durch einen Automaten beschrieben werden kann. Die Familie der Sprachen, die von endlichen Automaten akzeptiert werden, ist daher abgeschlossen gegenüber Durchschnitt. In der Vorlesung haben wir gezeigt, dass diese Sprachfamilie genau die regulären Sprachen sind. Daher sind auch die regulären Sprachen abgeschlossen gegenüber Durchschnitt: Der Durchschnitt zweier regulärer Sprachen ist wieder regulär.

¹Die konkreten Bezeichnungen der Zustände haben keinen Einfluss auf die akzeptierte Sprache des Automaten. Sie können daher so gewählt werden, dass sie einen Anhaltspunkt für die Rolle bieten, die die Zustände im Automaten spielen.

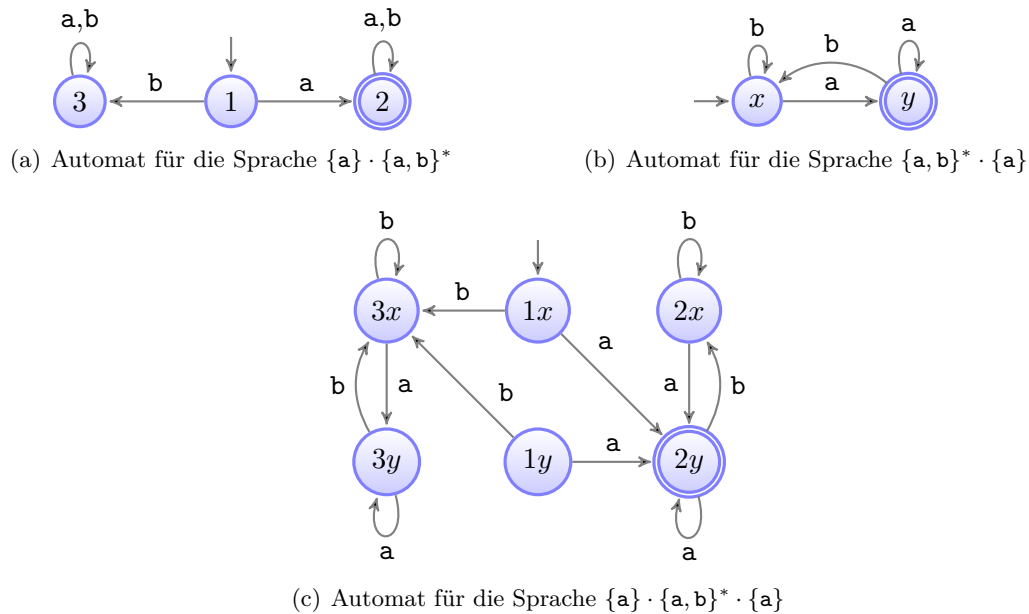
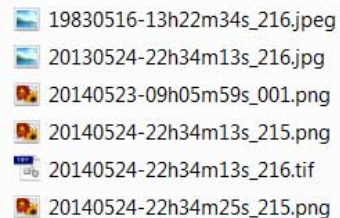


Abbildung 2: Beispiel für die Schnittbildung bei Automaten (Aufgabe 5)

Aufgabe 6 (0.3 Punkte)

Für die Archivierung von Fotos ist eine systematische Benennung hilfreich, etwa nach folgendem Schema.

- Der Fotoname beginnt mit dem Aufnahmedatum im Format $YYYYMMDD$, wobei $YYYY$, MM und DD das Jahr (vierstellig), den Monat (zweistellig) und den Tag (zweistellig) angeben.
- Getrennt durch einen Bindestrich folgt die Uhrzeit in der Form $HHhMMmSSs$, wobei HH , MM und SS die Stunden, Minuten und Sekunden (jeweils zweistellig) der Aufnahme angeben.
- Getrennt durch einen Unterstrich folgt eine Dezimalzahl mit mindestens einer Ziffer.
- Getrennt durch einen Punkt folgt der Dateityp bestehend aus drei oder vier Kleinbuchstaben.



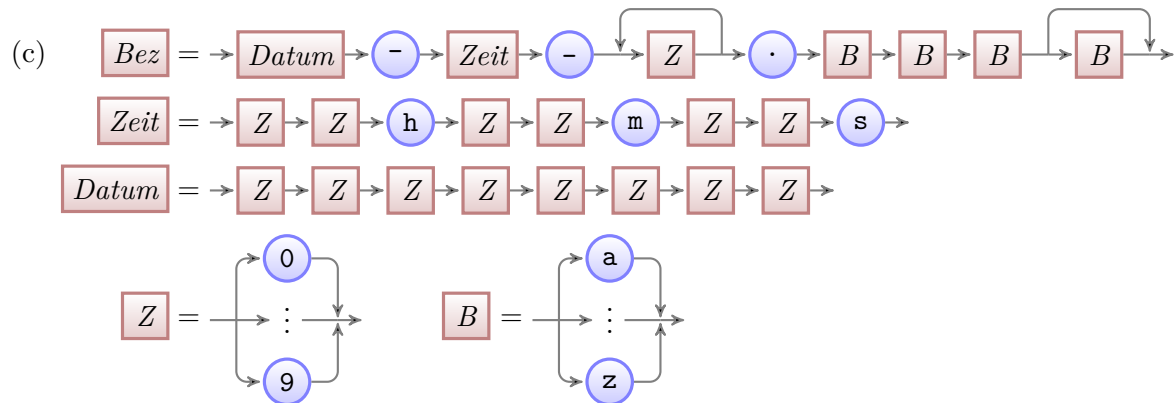
(a) Geben Sie einen regulären Ausdruck in algebraischer Notation an, der die Menge aller derartigen Fotobezeichnungen beschreibt.²

²Bei Datum und Uhrzeit kann von beliebigen Ziffernfolgen ausgegangen werden. Etwa ist auch die Stunde 33 und der Monat 00 zulässig).

- (b) Geben Sie einen POSIX Extended Regular Expression an, der alle Zeilen beschreibt, die ausschließlich eine Foto-Bezeichnung enthalten.
- (c) Stellen Sie Ihren regulären Ausdruck aus Teil a durch ein Syntaxdiagramm dar.

Lösung

- (a) $nnnnnnnnn-nnhnnmnnns_nn^*.lll(\epsilon + l)$ wobei n eine Abkürzung ist für $(0 + \dots + 9)$ und l eine Abkürzung für $(a + \dots + z)$ ist.
- (b) $\wedge[0-9]\{8\}-[0-9]\{2\}h[0-9]\{2\}m[0-9]\{2\}s_ [0-9]+\ . [a-z]\{3,4\}\$$

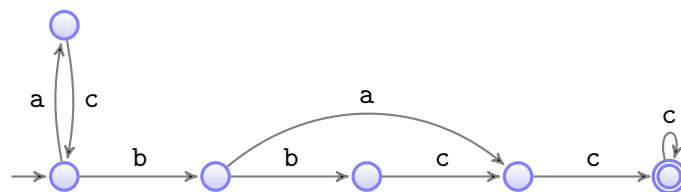


Aufgabe 7 (0.3 Punkte)

Konstruieren Sie einen endlichen Automaten, der dieselbe Sprache wie der reguläre Ausdruck $(ac)^*b(a + bc)cc^*$ (algebraische Notation) beschreibt.

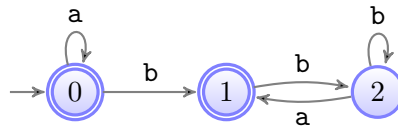
Lösung

Der gesuchte Automat kann mit dem allgemeinen Verfahren konstruiert werden, enthält dann aber in der Regel viel mehr Zustände und ϵ -Kanten als notwendig. Der folgende Automat wurde bereits vereinfacht.



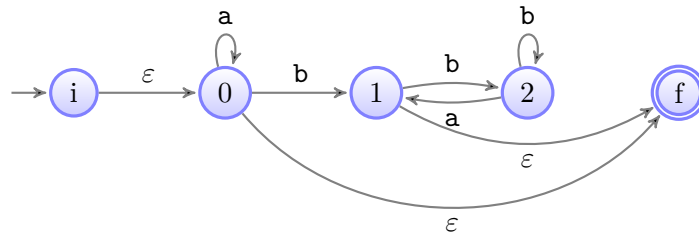
Aufgabe 8 (0.3 Punkte)

Konstruieren Sie zum folgenden endlichen Automaten einen regulären Ausdruck. Orientieren Sie sich am Algorithmus, der in der Vorlesung besprochen wurde; geben Sie den Automaten nach jeder Zustandselimination an.



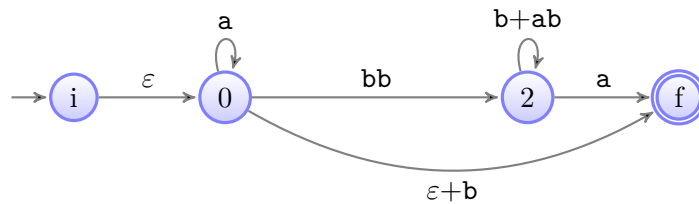
Lösung

Neuer Anfangs- und Endzustand:

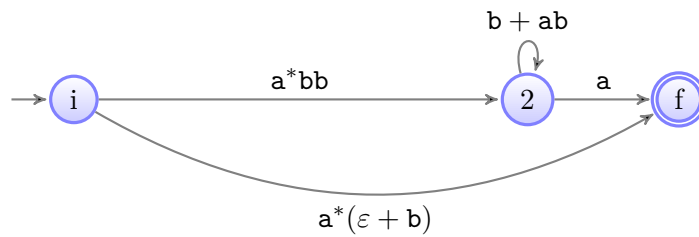


Wir eliminieren die Zustände in der Reihenfolge 1, 0 und 2; andere Reihenfolgen sind ebenfalls möglich.

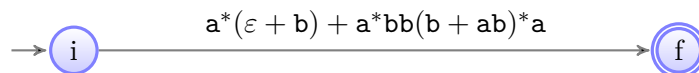
Elimination von Zustand 1:



Elimination von Zustand 0:



Elimination von Zustand 2:



Die Sprache des ursprünglichen Automaten wird somit durch den regulären Ausdruck $a^*(\epsilon + b) + a^*bb(b + ab)^*a = a^*(\epsilon + b + bb(b + ab)^*a)$ beschrieben.

Aufgabe 9 (0.3 Punkte)

Die Sprache der Zaubersprüche werde durch die Grammatik $Z = \langle N, T, P, A \rangle$ beschrieben, wobei N , T und A wie folgt definiert sind.

$$N = \{A, B, C, D\}$$

$$T = \{a, b, d, f, h, i, k, o, p, r, s, u\}$$

$$P = \{A \rightarrow \text{abra } B \mid \text{ho } C \\ B \rightarrow \text{kad } A \mid C \mid \varepsilon \\ C \rightarrow \text{kus } C \mid \text{po } C \mid D \\ D \rightarrow \text{fidibus} \mid \varepsilon \}$$

- (a) Stellen Sie fest, welche der folgenden Wörter zulässige Zaubersprüche sind, indem Sie entweder eine Ableitung in der Grammatik angeben oder argumentieren, warum das Wort nicht abgeleitet werden kann.
- (1) hokus
 - (2) abrakadabrapokusfidibus
 - (3) abrakadhopoku
- (b) Überprüfen Sie für jede der folgenden Aussagen, ob sie auf die Sprache $\mathcal{L}(Z)$ zutrifft. Wenn ja, warum? Wenn nein, geben Sie ein Gegenbeispiel an!
- (1) Vor jedem **kad** muss ein **abra** kommen.
 - (2) Zaubersprüche, die länger als 35 Zeichen sind, enthalten mindestens einmal das Teilwort **kus**.
 - (3) Jeder Zauberspruch endet mit **fidibus**.
- (c) Geben Sie einen endlichen Automaten für $\mathcal{L}(Z)$ an.

Lösung

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

$$A \Rightarrow \text{ho } C \Rightarrow \text{hokus } C \Rightarrow \text{hokus } D \Rightarrow \text{hokus } \varepsilon = \text{hokus}$$

- (2) Ja, das Wort liegt in der Sprache $\mathcal{L}(Z)$:

$$A \Rightarrow \text{abra } B \Rightarrow \text{abra kad } A \Rightarrow \text{abra kad abra } B \Rightarrow \text{abra kad abra } C \\ \Rightarrow \text{abra kad abra po } C \Rightarrow \text{abra kad abra po kus } C \Rightarrow \text{abra kad abra po kus } D \\ \Rightarrow \text{abra kad abra po kus fidibus}$$

- (3) Nein, das Wort liegt nicht in der Sprache $\mathcal{L}(Z)$: Die Grammatik-Produktionen haben die Eigenschaft, dass auf das Symbol **u** immer das Symbol **s** folgen muss, entweder als **kus** in der C -Produktion oder als **fidibus** in der D -Produktion. Da im zu analysierenden Wort dem **u** gar nichts folgt (insbesondere kein **s**), kann dieses Wort nicht durch die Grammatik erzeugt werden.

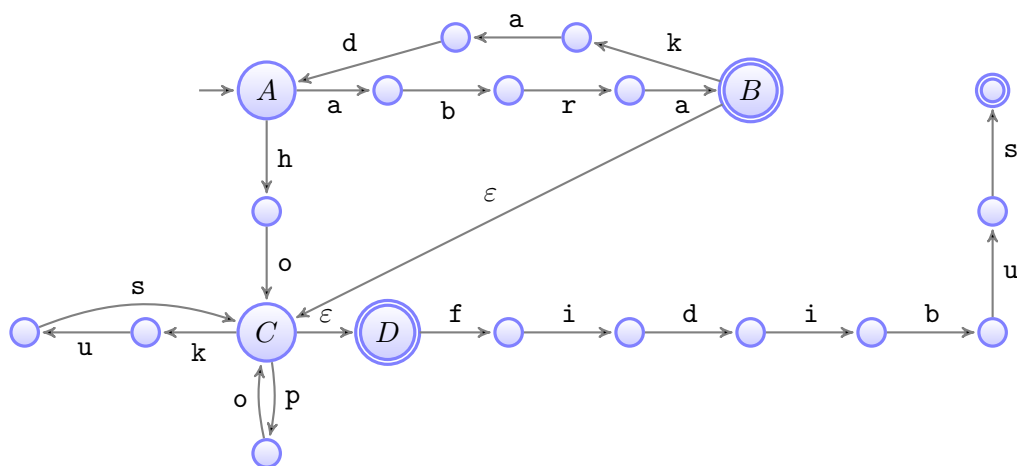
(b) (1) Richtig. Das Teilwort **kad** kann nur durch die Produktion $B \rightarrow \text{kad } A$ entstehen, und das Nonterminal B tritt wiederum nur in der Produktion $A \rightarrow \text{abra } B$ auf. Wer also **abra** sagt, muss auch **kad** sagen.

(2) Falsch. Das Wort $(\text{abrakad})^5 \text{ho}$ besteht aus 37 Zeichen und enthält nicht das Teilwort **kus**, es ist aber in der Grammatik ableitbar:

$$\begin{aligned} A &\Rightarrow \text{abra } B \Rightarrow \text{abrakad } A \Rightarrow \dots \Rightarrow (\text{abrakad})^5 A \\ &\Rightarrow (\text{abrakad})^5 \text{ho } C \Rightarrow (\text{abrakad})^5 \text{ho } D \Rightarrow (\text{abrakad})^5 \text{ho} \end{aligned}$$

(3) Falsch. Als Gegenbeispiel kann das Wort **hokus** dienen, das laut Teilaufgabe a1 in der Sprache $\mathcal{L}(Z)$ liegt, aber nicht mit **fidibus** endet.

(c) 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 (0.3 Punkte)

Einfache Dokumente im Textsatzsystem \LaTeX beginnen mit den Zeilen

```
\documentclass Optionen {Art}
\begin{document}
```

Danach folgt der eigentliche Dokumenteninhalte und die Schlusszeile

```
\end{document}
```

Art ist ein einzelner Name, wobei ein Name eine nicht-leere Folge von Ziffern, Klein- und Großbuchstaben ist. Die *Optionen* können entweder ganz fehlen oder sie sind eine in eckigen Klammern eingeschlossene nicht-leere Folge von Namen, die durch Beistriche getrennt werden.

Der Dokumentinhalt ist eine möglicherweise leere Folge von Texten, Aufzählungen und punktierten Listen in beliebiger Reihenfolge. Ein Text ist eine nicht-leere Folge von Buchstaben, Ziffern, Leerzeichen, Kommas, Punkten und Doppelpunkten. Aufzählungen beginnen mit

```
\begin{enumerate}
```

und enden mit

```
\end{enumerate}
```

Dazwischen liegt eine nicht-leere Folge von Listeneinträgen. Ein Listeneintrag besteht aus dem Kommando `\item` gefolgt von Texten, Aufzählungen und punktierten Listen in beliebiger Reihenfolge. Ist der Listeneintrag leer, besteht er nur aus `\item`. Eine punktierte Liste ist genauso aufgebaut wie eine Aufzählung, außer dass sie mit `\begin{itemize}` beginnt und mit `\end{itemize}` endet.

Ein Beispiel für ein derartiges Dokument ist das folgende.

```
\documentclass[a4paper,12pt]{article}
\begin{document}
Ich bin ein Text, dem eine punktierte Liste folgt:
\begin{itemize}
\item Listeneintrag
\item Aufzählung innerhalb eines Listeneintrags:
  \begin{enumerate}
    \item Schon wieder ein Eintrag.
  \end{enumerate}
\end{itemize}
\end{document}
```

Das Beispieldokument ist von der Art `article` mit den beiden Optionen `a4paper` und `12pt`. Der Dokumenteninhalt besteht aus einem Text und einer punktierten Liste. Diese enthält zwei Einträge, wobei der erste aus einem Text und der zweite aus einem Text und einer Aufzählung besteht. Die Aufzählung enthält nur einen Listeneintrag.

Sei \mathcal{L} die Menge aller solchen einfachen \LaTeX -Dokumente. Beschreiben Sie \mathcal{L} mit Hilfe einer kontextfreien Grammatik. Verwenden Sie EBNF-Notationen, um die Grammatik übersichtlich zu strukturieren.

Lösung

$\langle N, T, P, \text{Dokument} \rangle$, wobei

$N = \{ \text{Dokument}, \text{Optionen}, \text{Inhalt}, \text{Enumerate}, \text{Itemize}, \text{Item}, \text{Name}, \text{NZeichen}, \text{Text}, \text{Zeichen} \}$,

$T = \{ \dots \text{alle Zeichen in den Produktionen zwischen Anführungszeichen} \dots \}$,

$P = \{ \text{Dokument} \rightarrow "\backslash\text{documentclass}" [\text{Optionen}] "\{" \text{Name} "\}"$
 $\quad\quad\quad "\backslash\text{begin}\{\text{document}\}" \text{Inhalt} "\backslash\text{end}\{\text{document}\}"$,
 $\text{Optionen} \rightarrow "[\text{Name} \{ " , " \text{Name} \} "]$,
 $\text{Inhalt} \rightarrow \{ \text{Text} \mid \text{Enumerate} \mid \text{Itemize} \}$,
 $\text{Enumerate} \rightarrow "\backslash\text{begin}\{\text{enumerate}\}" \text{Item} \{ \text{Item} \} "\backslash\text{end}\{\text{enumerate}\}"$,
 $\text{Itemize} \rightarrow "\backslash\text{begin}\{\text{itemize}\}" \text{Item} \{ \text{Item} \} "\backslash\text{end}\{\text{itemize}\}"$,
 $\text{Item} \rightarrow "\backslash\text{item}" \text{Inhalt}$,
 $\text{Name} \rightarrow \text{NZeichen} \{ \text{NZeichen} \}$,
 $\text{NZeichen} \rightarrow "a" \mid \dots \mid "z" \mid "A" \mid \dots \mid "Z" \mid "0" \mid \dots \mid "9"$,
 $\text{Text} \rightarrow \text{Zeichen} \{ \text{Zeichen} \}$,
 $\text{Zeichen} \rightarrow \text{NZeichen} \mid " " \mid " , " \mid " . " \mid " : " \}$.

Aufgabe 11 (0.4 Punkte)

Seien folgende Mengen von Variablen-, Funktions- und Prädikatensymbole gegeben.

$$\mathcal{V} = \{x, y, z\}$$

$$\mathcal{F} = \{ \text{latein}/0, \text{spanisch}/0, \text{mehr}/1 \}$$

$$\mathcal{P} = \{ \text{Schüler}/1, \text{Wahlfach}/1, \text{Wählt}/2 \}$$

Geben Sie eine strukturierte kontextfreie Grammatik für die Sprache der prädikatenlogischen Formeln über diesen Symbolmengen an. Beispiele für Wörter, die in dieser Formelsprache liegen:

$$\forall x (\text{Schüler}(x) \supset \exists y (\text{Wahlfach}(y) \wedge \text{Wählt}(x, y)))$$

$$\forall x (\text{Wählt}(x, \text{spanisch}) \vee \text{Wählt}(x, \text{mehr}(\text{spanisch})) \vee \text{Wählt}(x, \text{mehr}(\text{mehr}(\text{spanisch}))))$$

Lösung

$\langle V, T, P, \text{Formel} \rangle$ mit

$$V = \{ \text{Formel}, \text{Op}, \dots, \text{FS}_0, \text{FS}_1, \text{Var} \},$$

$$T = \{ \top, \perp, \neg, (,), ", \wedge, \dots, \forall, \exists, \text{Schüler}, \dots, \text{mehr}, x, y, z \},$$

$$P = \{ \text{Formel} \rightarrow \text{PS}_1 (" \text{Term} ") \mid \text{PS}_2 (" \text{Term} ", " \text{Term} ") \mid \top \mid \perp \mid \neg " \text{Formel} \mid (" \text{Formel Op Formel} ") \mid \text{Qu Var Formel} ,$$

$$\text{Op} \rightarrow \wedge \mid \uparrow \mid \vee \mid \downarrow \mid \equiv \mid \neq \mid \supset \mid \subset ,$$

$$\text{Qu} \rightarrow \forall \mid \exists ,$$

$$\text{Term} \rightarrow \text{Var} \mid \text{FS}_0 \mid \text{FS}_1 (" \text{Term} ") ,$$

$$\text{PS}_1 \rightarrow \text{Schüler} \mid \text{Wahlfach} ,$$

$$\text{PS}_2 \rightarrow \text{Wählt} ,$$

$$\text{FS}_0 \rightarrow \text{latein} \mid \text{spanisch} ,$$

$$\text{FS}_1 \rightarrow \text{mehr} ,$$

$$\text{Var} \rightarrow x \mid y \mid z .$$

Aufgabe 12 (0.4 Punkte)

Seien $\text{Mag}/2$, $\text{Katze}/1$ und $\text{Mensch}/1$ Prädikatensymbole sowie odie ein Konstantensymbol. Die Bedeutung der Symbole sei festgelegt durch folgende Interpretation I :

$$U = \{ \text{Arlene}, \text{Dr.Liz}, \text{Garfield}, \text{Jon}, \text{Nermal}, \text{Odie}, \text{Pooky} \}$$

$$I(\text{Mensch}) = \{ \text{Jon}, \text{Dr.Liz} \}$$

$$I(\text{Katze}) = \{ \text{Arlene}, \text{Garfield}, \text{Nermal} \}$$

$$I(\text{Mag}) = \{ (\text{Jon}, \text{Odie}), (\text{Jon}, \text{Garfield}), (\text{Jon}, \text{Nermal}), (\text{Dr.Liz}, \text{Nermal}), \\ (\text{Odie}, \text{Jon}), (\text{Odie}, \text{Dr.Liz}), (\text{Garfield}, \text{Arlene}), (\text{Garfield}, \text{Pooky}), \\ (\text{Nermal}, \text{Garfield}), (\text{Arlene}, \text{Garfield}) \}$$

$$I(\text{odie}) = \text{Odie}$$

- (a) Übersetzen Sie die Formel $\forall x (\text{Mensch}(x) \supset \text{Mag}(x, \text{odie}))$ in natürliche Sprache und stellen Sie fest, ob die Aussage auf die gegebene Interpretation zutrifft. Bestimmen Sie den Wert der Formel mittels der Evaluierungsfunktion val_I .
- (b) Übersetzen Sie die Formel $\neg \forall x \exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y)))$ in natürliche Sprache und stellen Sie fest, ob die Aussage auf die gegebene Interpretation zutrifft. Bestimmen Sie den Wert der Formel mittels der Evaluierungsfunktion val_I . Geben Sie weiters möglichst viele äquivalente Formeln an, die sich durch Verschiebung der Negation und der Quantoren ergeben; beachten Sie, dass $A \supset B = \neg A \vee B$ gilt.

Lösung

(a) Übersetzung: „Alle Menschen mögen Odie.“

Diese Aussage ist in der gegebenen Interpretation falsch, da Dr.Liz ein Mensch ist und Odie nicht mag.

$$\begin{aligned}
 & \text{val}_{I,\sigma}(\forall x \text{Mensch}(x) \supset \text{Mag}(x, \text{odie})) = 1 \\
 \iff & \text{val}_{I,\sigma'}(\text{Mensch}(x) \supset \text{Mag}(x, \text{odie})) = 1 \text{ für alle } \sigma' \overset{x}{\sim} \sigma \\
 \iff & \text{Wenn } \text{val}_{I,\sigma'}(\text{Mensch}(x)) = 1 \text{ dann } \text{val}_{I,\sigma'}(\text{Mag}(x, \text{odie})) = 1 \text{ für alle } \sigma' \overset{x}{\sim} \sigma \\
 \iff & \text{Wenn } \text{val}_{I,\sigma'}(x) \in I(\text{Mensch}) \text{ dann } (\text{val}_{I,\sigma'}(x), \text{val}_{I,\sigma'}(\text{odie})) \in I(\text{Mag}) \text{ für alle } \sigma' \overset{x}{\sim} \sigma \\
 \iff & \text{Wenn } \sigma'(x) \in \{\text{Jon}, \text{Dr.Liz}\} \text{ dann } (\sigma'(x), \text{Odie}) \in \{\dots, (\text{Jon}, \text{Odie}), \dots\} \text{ für alle } \sigma' \overset{x}{\sim} \sigma \\
 \iff & \text{Für alle } \sigma'(x) \in \{\text{Jon}, \text{Dr.Liz}\} \text{ gilt } (\sigma'(x), \text{Odie}) \in \{\dots, (\text{Jon}, \text{Odie}), \dots\}
 \end{aligned}$$

Die letzte Aussage ist falsch, da (Dr.Liz, Odie) nicht in $\{\dots, (\text{Jon}, \text{Odie}), \dots\}$ vorkommt. Somit ist die ursprüngliche Formel nicht wahr in der Interpretation I .

(b) Übersetzung: „Nicht alle Katzen mögen eine Katze.“

Diese Aussage ist in der gegebenen Interpretation falsch, da jede der drei Katzen Arlene, Garfield und Nermal eine Katze mag (Garfield mag Arlene, und Arlene und Nermal mögen beide Garfield).

$$\begin{aligned}
 & \text{val}_{I,\sigma}(\neg \forall x \exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y)))) = 1 \\
 \iff & \text{not val}_{I,\sigma}(\forall x \exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y)))) = 1 \\
 \iff & \text{val}_{I,\sigma}(\forall x \exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y)))) = 0 \\
 \iff & \text{val}_{I,\sigma'}(\exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y)))) = 0 \text{ für ein } \sigma' \overset{x}{\sim} \sigma \\
 \iff & \text{val}_{I,\sigma''}(\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y))) = 0 \\
 & \quad \text{für ein } \sigma' \overset{x}{\sim} \sigma \text{ und alle } \sigma'' \overset{y}{\sim} \sigma' \\
 \iff & \text{val}_{I,\sigma''}(\text{Katze}(x)) = 1 \text{ und } \text{val}_{I,\sigma''}(\text{Katze}(y) \wedge \text{Mag}(x, y)) = 0 \\
 & \quad \text{für ein } \sigma' \overset{x}{\sim} \sigma \text{ und alle } \sigma'' \overset{y}{\sim} \sigma' \\
 \iff & \text{val}_{I,\sigma''}(\text{Katze}(x)) = 1 \text{ und } [\text{val}_{I,\sigma''}(\text{Katze}(y)) = 0 \text{ oder } \text{val}_{I,\sigma''}(\text{Mag}(x, y)) = 0] \\
 & \quad \text{für ein } \sigma' \overset{x}{\sim} \sigma \text{ und alle } \sigma'' \overset{y}{\sim} \sigma' \\
 \iff & \sigma''(x) \in I(\text{Katze}) \text{ und } [\sigma''(y) \notin I(\text{Katze}) \text{ oder } (\sigma''(x), \sigma''(y)) \notin I(\text{Mag})] \\
 & \quad \text{für ein } \sigma' \overset{x}{\sim} \sigma \text{ und alle } \sigma'' \overset{y}{\sim} \sigma' \\
 \iff & \text{Es gibt ein } \sigma''(x) \in \mathcal{U}, \text{ sodass für alle } \sigma''(y) \in \mathcal{U} \text{ gilt: } \sigma''(x) \in I(\text{Katze}) \text{ sowie} \\
 & \quad \text{entweder } \sigma''(y) \notin I(\text{Katze}) \text{ oder } (\sigma''(x), \sigma''(y)) \notin I(\text{Mag}). \\
 \iff & \text{Es gibt ein } \sigma''(x) \in I(\text{Katze}), \text{ sodass für alle } \sigma''(y) \in I(\text{Katze}) \text{ das Paar} \\
 & \quad (\sigma''(x), \sigma''(y)) \text{ nicht in } I(\text{Mag}) \text{ liegt.}
 \end{aligned}$$

Diese Aussage ist falsch, da sich zu jedem $\sigma''(x) \in I(\text{Katze})$ – also zu Arlene, Garfield und Nermal – ein $\sigma''(y) \in I(\text{Katze})$ finden lässt, sodass $(\sigma''(x), \sigma''(y))$ in $I(\text{Mag})$ liegt: $I(\text{Mag})$ enthält die Paare (Arlene, Garfield), (Garfield, Arlene), und (Nermal, Garfield).

Beispiele äquivalenter Formeln und Aussagen:

$$\begin{aligned}
 & \neg \forall x \exists y (\text{Katze}(x) \supset (\text{Katze}(y) \wedge \text{Mag}(x, y))) && \text{Angabe} \\
 & \neg \forall x \exists y (\neg \text{Katze}(x) \vee (\text{Katze}(y) \wedge \text{Mag}(x, y)))
 \end{aligned}$$

$\neg\forall x (\neg\text{Katze}(x) \vee \exists y (\text{Katze}(y) \wedge \text{Mag}(x, y)))$
 $\neg\forall x (\text{Katze}(x) \supset \exists y (\text{Katze}(y) \wedge \text{Mag}(x, y)))$ Nicht alle Katzen mögen eine Katze.
 $\exists x \neg(\text{Katze}(x) \supset \exists y (\text{Katze}(y) \wedge \text{Mag}(x, y)))$
 $\exists x \neg(\neg\text{Katze}(x) \vee \exists y (\text{Katze}(y) \wedge \text{Mag}(x, y)))$
 $\exists x (\text{Katze}(x) \wedge \neg\exists y (\text{Katze}(y) \wedge \text{Mag}(x, y)))$ Es gibt Katzen, die keine Katzen mögen.
 $\exists x (\text{Katze}(x) \wedge \forall y \neg(\text{Katze}(y) \wedge \text{Mag}(x, y)))$
 $\exists x (\text{Katze}(x) \wedge \forall y (\neg\text{Katze}(y) \vee \neg\text{Mag}(x, y)))$
 $\exists x (\text{Katze}(x) \wedge \forall y (\text{Katze}(y) \supset \neg\text{Mag}(x, y)))$ Es gibt Katzen, die alle Katzen nicht mögen.

Aufgabe 13 (0.4 Punkte)

Seien *SpieltMit*/2, *Mutter*/1 und *Kind*/1 Prädikatensymbole sowie *anna* und *florian* Konstantensymbole mit folgender Bedeutung:

<i>SpieltMit</i> (<i>x</i> , <i>y</i>) ... <i>x</i> spielt mit <i>y</i>	<i>anna</i> ... Anna
<i>Mutter</i> (<i>x</i>) ... <i>x</i> ist eine Mutter	<i>florian</i> ... Florian
<i>Kind</i> (<i>x</i>) ... <i>x</i> ist ein Kind	

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

- (a) Alle Mütter, die mit Anna spielen, spielen auch mit Florian.
- (b) Manche Kinder spielen mit allen Müttern.

Sei weiters folgende Interpretation *I* gegeben:

$\mathcal{U} = \{\text{Anna, Barbara, Elisabeth, Florian, Kathrin, Nina, Tamara}\}$
 $I(\text{Mutter}) = \{\text{Barbara, Elisabeth}\}$
 $I(\text{Kind}) = \{\text{Anna, Florian, Nina}\}$
 $I(\text{SpieltMit}) = \{(\text{Anna, Florian}), (\text{Barbara, Anna}), (\text{Elisabeth, Anna}), (\text{Elisabeth, Florian}),$
 $(\text{Florian, Nina}), (\text{Florian, Kathrin}), (\text{Florian, Anna}), (\text{Nina, Florian}),$
 $(\text{Tamara, Elisabeth}), (\text{Tamara, Barbara}), (\text{Tamara, Anna})\},$
 $I(\text{anna}) = \text{Anna} \quad I(\text{florian}) = \text{Florian}$

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.

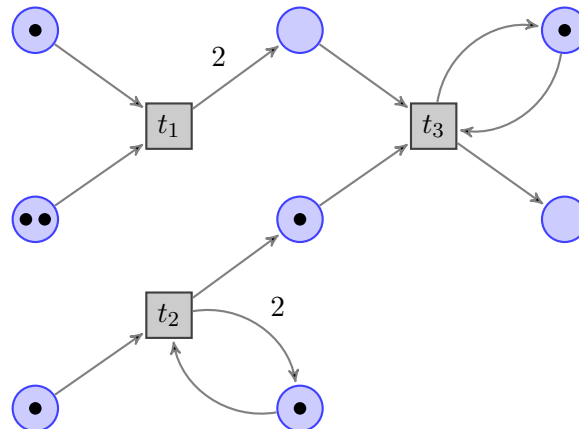
- (c) $\forall x (\text{Mutter}(x) \supset \exists y (\text{Kind}(y) \wedge \text{SpieltMit}(x, y)))$
- (d) $\forall x (\text{Mutter}(x) \wedge \text{SpieltMit}(x, \text{anna}))$
- (e) $\exists x (\text{Kind}(x) \wedge \forall y (\text{Kind}(y) \supset \text{SpieltMit}(x, y)))$
- (f) $\forall x (\text{SpieltMit}(x, \text{florian}) \neq \text{SpieltMit}(x, \text{anna}))$

Lösung

- (a) $\forall x ((Mutter(x) \wedge SpieltMit(x, anna)) \supset SpieltMit(x, florian))$ oder
 $\forall x (Mutter(x) \supset (SpieltMit(x, anna) \supset SpieltMit(x, florian)))$
- (b) $\exists x (Kind(x) \wedge \forall y (Mutter(y) \supset SpieltMit(x, y)))$ oder
 $\exists x \forall y (Kind(x) \wedge (Mutter(y) \supset SpieltMit(x, y)))$
- (c) Übersetzung: Alle Mütter spielen mit (mindestens) einem Kind.
 Diese Aussage ist wahr in I , da die Mutter Barbara mit dem Kind Anna und die Mutter Elisabeth mit dem Kind Florian spielt.
- (d) Übersetzung: Alle sind Mütter und spielen mit Anna.
 Diese Aussage ist falsch, da etwa Anna keine Mutter ist.
- (e) Übersetzung: Es gibt (mindestens) ein Kind, das mit allen Kindern spielt.
 Diese Aussage ist falsch in I , da es die drei Kinder Anna, Florian und Nina gibt und gemäß $I(SpieltMit)$ Anna nicht mit Nina, Florian nicht mit Florian und Nina nicht mit Anna spielt.
- (f) Übersetzung: Alle spielen entweder mit Florian oder Anna (aber nicht mit beiden).
 Diese Aussage ist falsch in I , da etwa Elisabeth sowohl mit Florian als auch mit Anna spielt.

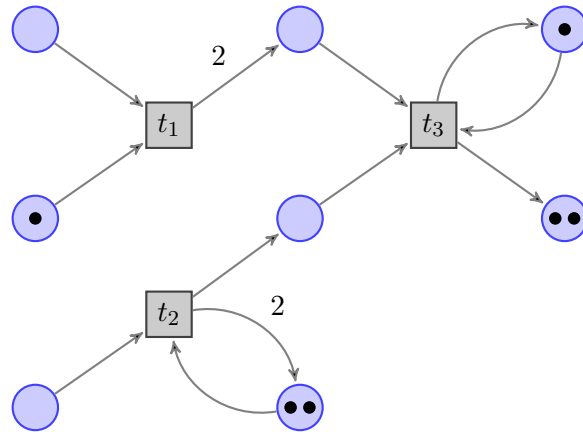
Aufgabe 14 (0.3 Punkte)

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 mehr aktiviert ist.



Lösung

Die möglichen Transitionsfolgen sind $t_1-t_2-t_3-t_3$, $t_1-t_3-t_2-t_3$ und $t_2-t_1-t_3-t_3$. Sie liefern alle dieselbe Endmarkierung:



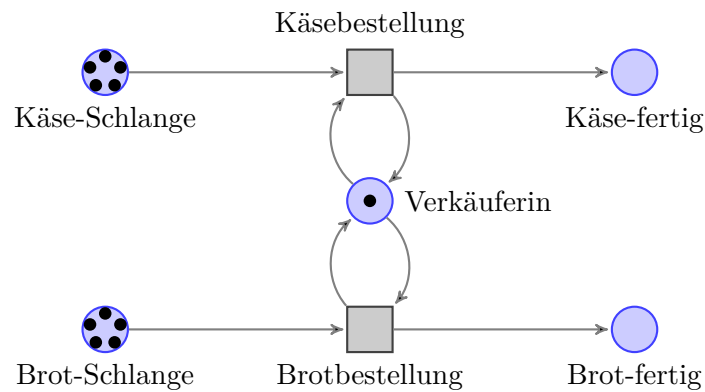
Aufgabe 15 (0.4 Punkte)

In einem Supermarkt gibt es eine Feinkosttheke und eine Brottheke. An beiden Theken stellen sich die Kunden separat an. An einkaufsschwachen Tagen bedient eine Verkäuferin beide Theken, wobei nicht definiert ist, wie viele Kunden sie bei einer Theke bedient, ehe sie zur anderen wechselt.

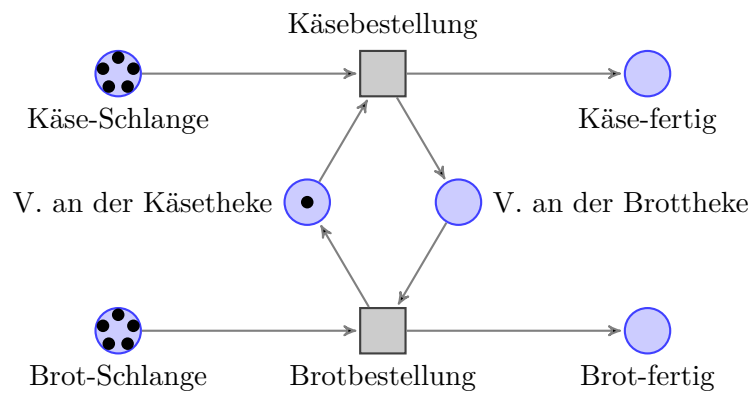
Modellieren Sie dieses System mit Hilfe eines Petri-Netzes. Geben Sie eine geeignete Anfangsmarkierung an. Wie müssen Sie das System modifizieren, wenn Sie sicherstellen wollen, dass abwechselnd ein Kunde von der Feinkosttheke und ein Kunde von der Brottheke bedient wird?

Lösung

Die Markierungen, die Kunden repräsentieren, befinden sich zunächst in der „Käse-schlange“ bzw. in der „Brot-Schlange“. Die Stelle in der Mitte repräsentiert die Verkäuferin. Nun können die Transitionen „Käsebestellung“ und „Brotbestellung“ in beliebiger Reihenfolge feuern, es können also beliebig Kunden an der Käse- bzw. Brottheke bedient werden. Sobald eine der beiden Transitionen gefeuert hat, ist die Verkäuferin wieder verfügbar und die andere Transition kann feuern (oder dieselbe erneut).



Soll sichergestellt werden, dass die Kunden an der Brottheke und an der Käsetheke abwechselnd bedient werden, muss das Netz wie folgt modifiziert werden, sodass die Verkäuferin abwechselnd die beiden Transitionen aktiviert:



Problematisch ist allerdings, wenn in einer Schlange mehr Personen warten als in der anderen. Sind zum Beispiel zwei Personen mehr in der Käse-Schlange, so wird die letzte Person erst dann bedient, wenn zuvor wieder jemand an der Wursttheke bedient wurde.