2. Übungsblatt (mit Lösungen)

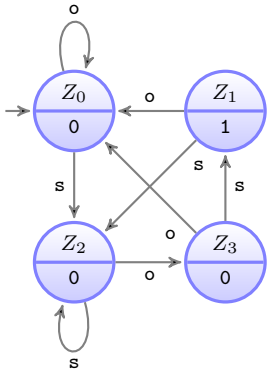
3.0 VU Formale Modellierung

Gernot Salzer, Marion Scholz

22. November 2016

Aufgabe 1 (3 Punkte)

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



(a) Geben Sie die Ausgaben zu folgenden Eingaben an: `sssos`, `ososo`, `oosoo`.

(b) Berechnen Sie schrittweise $\delta^*(Z_0, \texttt{ssoss})$ und $\gamma^*(Z_0, \texttt{ssoss})$.

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

Lösung

(a)

w:	`sssos`	`ososo`	`oosoo`
$[\![\mathcal{A}]\!](w)$:	`00001`	`00010`	`00000`

(b)
$$\begin{aligned}
\delta^*(Z_0, \texttt{ssoss}) &= \delta^*(\delta(Z_0, \texttt{s}), \texttt{soss}) &&= \delta^*(Z_2, \texttt{soss}) \\
&= \delta^*(\delta(Z_2, \texttt{s}), \texttt{oss}) &&= \delta^*(Z_2, \texttt{oss}) \\
&= \delta^*(\delta(Z_2, \texttt{o}), \texttt{ss}) &&= \delta^*(Z_3, \texttt{ss}) \\
&= \delta^*(\delta(Z_3, \texttt{s}), \texttt{s}) &&= \delta^*(Z_1, \texttt{s}) \\
&= \delta^*(\delta(Z_1, \texttt{s}), \varepsilon) &&= \delta^*(Z_2, \varepsilon) \\
&= Z_2
\end{aligned}$$

$$
\begin{aligned}
\gamma^*(Z_0, \texttt{ssoss}) &= \gamma(\delta(Z_0, \texttt{s})) \cdot \gamma^*(\delta(Z_0, \texttt{s}), \texttt{soss}) && = \gamma(Z_2) \cdot \gamma^*(Z_2, \texttt{soss}) \\
&= 0 \cdot \gamma(\delta(Z_2, \texttt{s})) \cdot \gamma^*(\delta(Z_2, \texttt{s}), \texttt{oss}) && = 0 \cdot \gamma(Z_2) \cdot \gamma^*(Z_2, \texttt{oss}) \\
&= 00 \cdot \gamma(\delta(Z_2, \texttt{o})) \cdot \gamma^*(\delta(Z_2, \texttt{o}), \texttt{ss}) && = 00 \cdot \gamma(Z_3) \cdot \gamma^*(Z_3, \texttt{ss}) \\
&= 000 \cdot \gamma(\delta(Z_3, \texttt{s})) \cdot \gamma^*(\delta(Z_3, \texttt{s}), \texttt{s}) && = 000 \cdot \gamma(Z_1) \cdot \gamma^*(Z_1, \texttt{s}) \\
&= 0001 \cdot \gamma(\delta(Z_1, \texttt{s})) \cdot \gamma^*(\delta(Z_1, \texttt{s}), \varepsilon) && = 0001 \cdot \gamma(Z_2) \cdot \gamma^*(Z_2, \varepsilon) \\
&= 00010 \cdot \varepsilon = 00010
\end{aligned}
$$

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

Aufgabe 2 (3 Punkte)

Sei L die Sprache

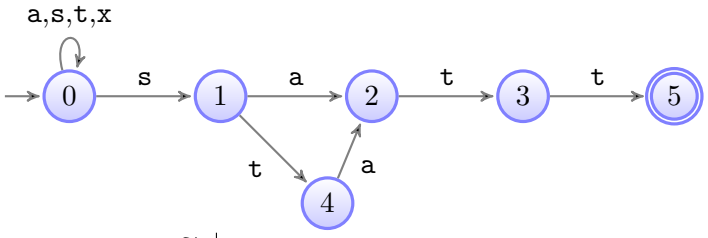
$$\{\, w \in \{\texttt{a}, \texttt{s}, \texttt{t}, \texttt{x}\}^* \mid w \text{ endet mit der Buchstabenfolge } \texttt{satt} \text{ oder } \texttt{statt} \,\} \ .$$

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 `s` gibt es zwei mögliche Folgezustände. Zusätzlich zur graphischen Darstellung geben wir die Übergangsfunktion auch als Tabelle an, da diese bei der Determinisierung hilft.



δ^*	a	s	t	x
0	{0}	{0,1}	{0}	{0}
1	{2}	{}	{4}	{}
2	{}	{}	{3}	{}
3	{}	{}	{5}	{}
4	{2}	{}	{}	{}
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.

δ^*	a	s	t	x
$\{0\}$	$\{0\}$	$\{0,1\}$	$\{0\}$	$\{0\}$
$\{0,1\}$	$\{0,2\}$	$\{0,1\}$	$\{0,4\}$	$\{0\}$
$\{0,2\}$	$\{0\}$	$\{0,1\}$	$\{0,3\}$	$\{0\}$
$\{0,4\}$	$\{0,2\}$	$\{0,1\}$	$\{0\}$	$\{0\}$
$\{0,3\}$	$\{0\}$	$\{0,1\}$	$\{0,5\}$	$\{0\}$
$\{0,5\}$	$\{0\}$	$\{0,1\}$	$\{0\}$	$\{0\}$

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

$$\Sigma = \{\mathtt{a}, \mathtt{s}, \mathtt{t}, \mathtt{x}\}$$
$$\widehat{F} = \{\{0,5\}\}$$
$$\widehat{Q} = \{\{0\}, \{0,1\}, \{0,2\}, \{0,3\}, \{0,4\}\} \cup \widehat{F}$$

Aufgabe 3 (4 Punkte)

Die *B-Sprache*, auch *Bebe-Sprache* oder *Bebe-Sprabachebe* genannt, ist eine bei Kindern beliebte Geheimsprache, bei der die normale Sprache nach folgenden Regeln verfremdet wird.

- Vokale werden verdoppelt und es wird der Buchstabe „b“ eingeschoben. Zum Beispiel ergibt sich aus dem Wort `hallo` das Geheimwort `haballobo`.
  Vokale im Deutschen: a, e, i, o, u, y; die Umlaute ä, ö und ü lassen wir der Einfachheit halber weg.
- Zwielauten (Diphtongen) wird „ab“ vorangestellt. So wird aus `bauer` das Geheimwort `babaueber`.
  Zwielaute im Deutschen: ai, au, ay, ei, eu, ey und ui; die Variante äu lassen wir der Einfachheit halber weg.
- Das lange I, „ie“, zählt als Vokal, dem „ib“ vorangestellt wird. Aus `dieb` ergibt sich somit `dibieb`.

Beispiele für Sätze in der Bebe-Sprache (␣ markiert das Leerzeichen):

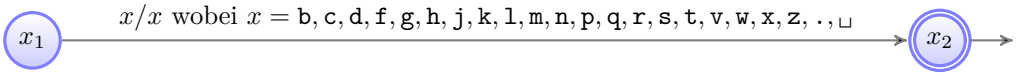
`dabas␣ibist␣dibie␣bebe␣sprabachebe.`
`abaus␣dibieb␣ebergibibt␣sibich␣sobomibit␣dibibibieb.`

Geben Sie einen Transducer an, der eine beliebige Zeichenfolge über dem Alphabet $\{\mathtt{a}, \ldots, \mathtt{z}, \mathtt{␣}, \mathtt{.}\}$ entsprechend dieser Regeln umwandelt.

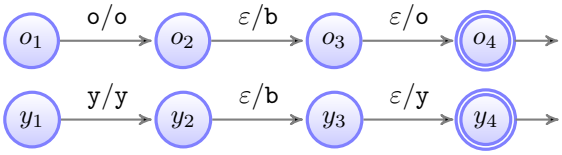
Lösung

Wir konstruieren zunächst Teil-Transducer für einzelne Konsonanten, Vokale und Zwielaute, die wir danach geeignet kombinieren. Jene Zustände, bei denen die Ausgabe eine korrekte Übersetzung der Eingabe darstellt, markieren wir als Endzustände.

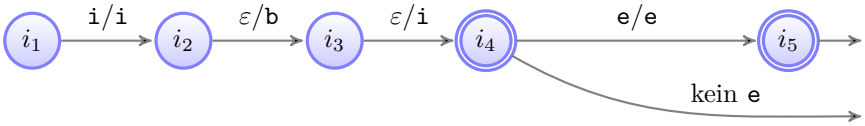
Konsonanten oder Sonderzeichen können direkt ausgegeben werden.



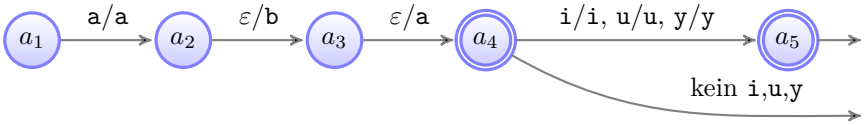
Die Vokale `o` und `y` können unabhängig von nachfolgenden Zeichen übersetzt werden.



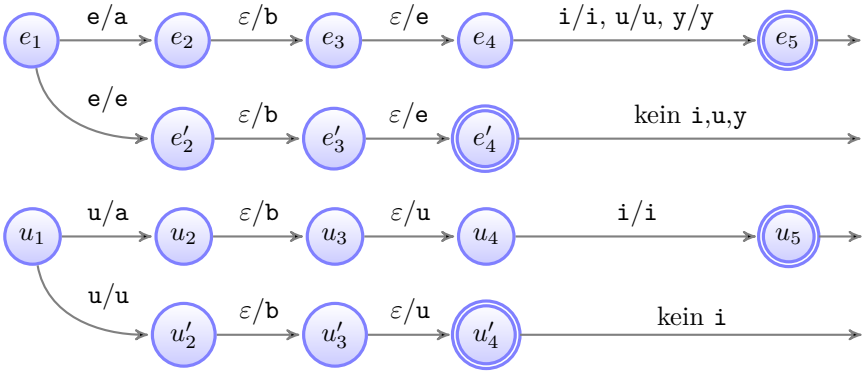
Beim Zeichen `i` müssen wir zwei Fälle unterscheiden, je nachdem, ob ein `e` folgt oder nicht.



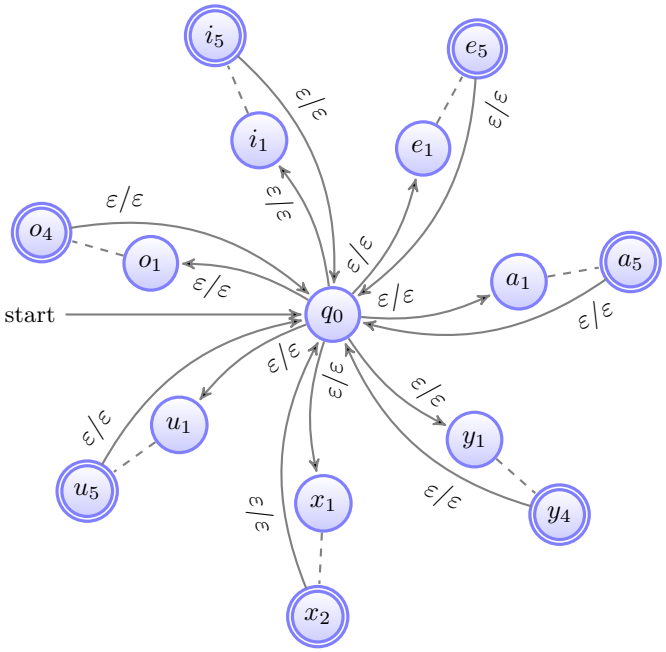
Beim Zeichen `a` müssen wir die Fälle Vokal und Zwielaut unterscheiden.



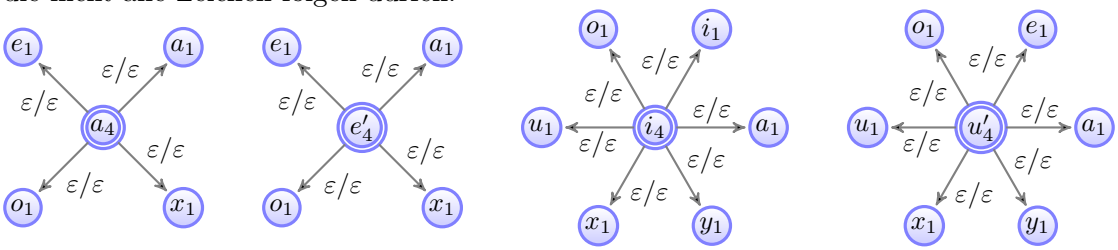
Bei den Zeichen `e` und `u` müssen wir die Fälle Vokal und Zwielaut unterscheiden, wobei sich bereits das erste Zeichen der Ausgabe unterscheidet.



Wir verbinden den Startzustand mit den Anfangsknoten der sieben Teilautomaten. Weiters darf in jedem der Zustände a_5, e_5, i_5, o_4, u_5, x_2 und y_4 ein beliebiges Zeichen folgen, daher sehen wir einen ε-Übergang zurück in den Anfangszustand vor.



Zuletzt müssen wir noch die Folgezustände der Zustände a_4, e'_4, i_4 und u'_4 festlegen, auf die nicht alle Zeichen folgen dürfen.



Aufgabe 4 (3 Punkte)

Vereinfachen Sie die folgenden Ausdücke.

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

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

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

Lösung

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

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

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

Aufgabe 5 (2 Punkte)

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

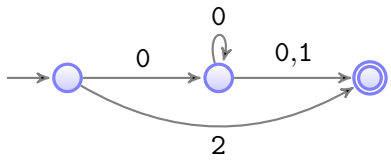
(a) $(0^+(1 + 0)) + 2$

(b) $(12)^*(1 + 2)2$

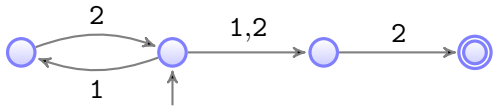
Lösung

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

(a)

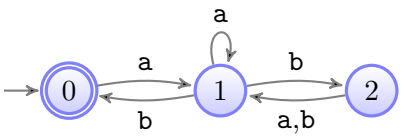


(b)



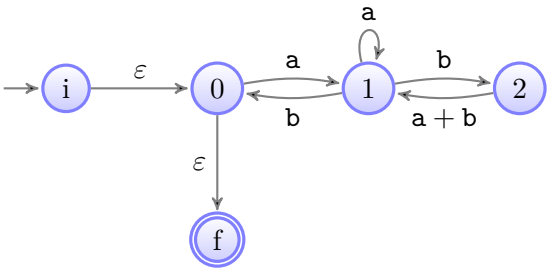
Aufgabe 6 (4 Punkte)

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



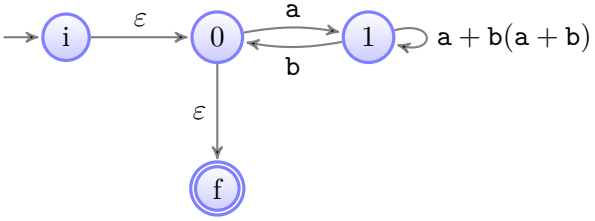
Lösung

Neuer Anfangs- und Endzustand:

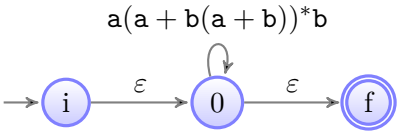


Wir eliminieren die Zustände in der Reihenfolge 2, 1 und 0; die anderen Reihenfolgen sind ebenfalls möglich.

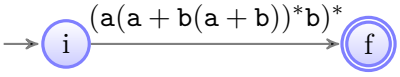
Elimination von Zustand 2:



Elimination von Zustand 1:



Elimination von Zustand 0:



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

Aufgabe 7 (3 Punkte)

Wenn Sie Ihre Lösung als Pdf-Datei in Tuwel hochladen, benennt Tuwel Ihre Datei wie folgt: Zunächst kommt der Nachname, dann ein Leerzeichen, dann eine beliebige Anzahl von Vornamen, ebenfalls durch Leerzeichen getrennt. Nach dem Namen folgt ein Unterstrich, dann eine achtstellige Dezimalzahl zur eindeutigen Identifikation der Abgabe, ein weiterer Unterstrich und die Bezeichnung „file“. Nach einem weiteren Unterstrich folgt der gesamte ursprüngliche Dateiname bestehend aus einer beliebig langen Folge von Buchstaben und Ziffern. Abschließend folgt immer die Dateiendung `.pdf`.

Beispiele derartiger Dateinamen (␣ steht für ein Leerzeichen):

`Mair␣Martin_38502362_file_meineTolleAbgabe12.pdf`
`Koller␣Barbara␣Maria_72304572_file_KollerBarbaraUE2complete.pdf`

Beschreiben Sie den Aufbau solcher Dateinamen mit den folgenden Methoden. Treffen Sie sinnvolle Annahmen, wenn Ihnen Informationen fehlen.

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

(b) Geben Sie einen regulären Ausdruck in Posix-Notation an, der alle Zeilen beschreibt, die *ausschließlich* einen derartigen Dateinamen enthalten.

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

Lösung

(a) Wir schreiben Symbole des Alphabets unter Anführungszeichen, um sie von algebraischen Symbolen (Metanotation) zu unterscheiden.

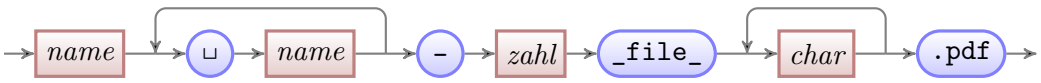
$$\mathit{name}\ \texttt{"␣"}\ \mathit{name}\ (\texttt{"␣"}\ \mathit{name})^*\ \texttt{"\_"}\ \mathit{zahl}\ \texttt{"\_file\_"}\ \mathit{char}\ \mathit{char}^*\ \texttt{".pdf"}$$

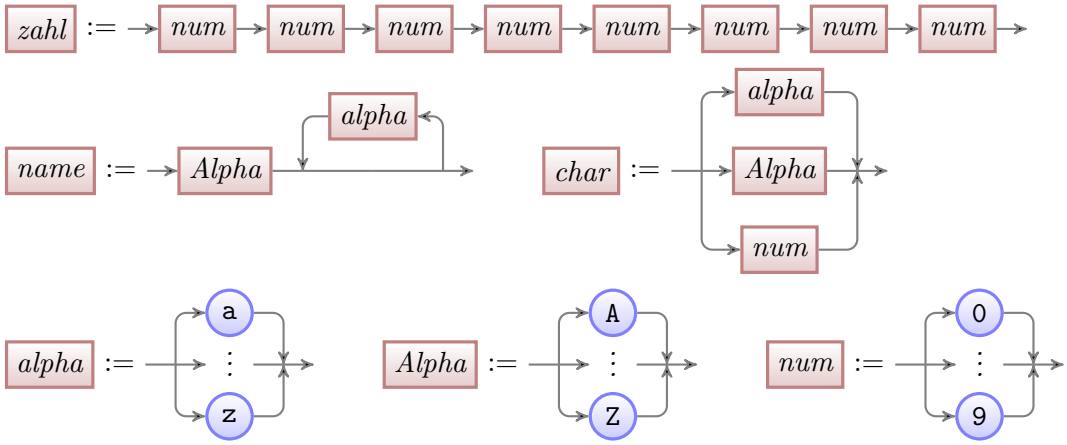
mit den Abkürzungen

$$\begin{aligned}
\mathit{alpha} &:= (\texttt{"a"} + \cdots + \texttt{"z"}) \\
\mathit{Alpha} &:= (\texttt{"A"} + \cdots + \texttt{"Z"}) \\
\mathit{num} &:= (\texttt{"0"} + \cdots + \texttt{"9"}) \\
\mathit{char} &:= (\mathit{alpha} + \mathit{Alpha} + \mathit{num}) \\
\mathit{zahl} &:= \mathit{num}\ \mathit{num}\ \mathit{num}\ \mathit{num}\ \mathit{num}\ \mathit{num}\ \mathit{num}\ \mathit{num} \\
\mathit{name} &:= \mathit{Alpha}\ \mathit{alpha}^*
\end{aligned}$$

(b) `^[A-Z][a-z]*␣[A-Z][a-z]*(␣[A-Z][a-z]*)*_[0-9]{8}_file_[A-Za-z0-9]+\.pdf$`

(c) Syntaxdiagramm:





Aufgabe 8 (4 Punkte)

Beim Spielen mit Spielzeugautos hört man oft Laute wie zum Beispiel „brrmtata“ oder „mööpmööp“ als Untermalung. Die Grammatik $G = \langle V, T, P, A \rangle$ beschreibt solche Autogeräusche, wobei

$$
\begin{aligned}
V &= \{A, B, C, D, E, R\} \\
T &= \{\texttt{a}, \texttt{b}, \texttt{m}, \texttt{ö}, \texttt{p}, \texttt{r}, \texttt{t}\} \\
P &= \{\, A \to B\, D \ , \\
&\qquad B \to \texttt{"br"}\, R\, \texttt{"m"}\, C \mid \texttt{"brm"} \mid \varepsilon \ , \\
&\qquad R \to \texttt{"r"}\, R \mid \varepsilon \ , \\
&\qquad C \to \texttt{"ta"} \mid \texttt{"tata"}\, B \ , \\
&\qquad D \to \texttt{"möpmö"}\, E\, \texttt{"p"} \mid \texttt{"trö"}\, E\, \texttt{"t"} \ , \\
&\qquad E \to \texttt{"ö"}\, E \mid \varepsilon\}
\end{aligned}
$$

Überprüfen Sie für die nachfolgenden Wörter, ob sie in der von der Grammatik G spezifizierten Sprache $\mathcal{L}(G)$ liegen. Falls ja, geben Sie eine Ableitung an. Falls nein, argumentieren Sie, warum nicht, und ändern Sie das Wort möglichst geringfügig ab, sodass es in der Sprache liegt.

(a) `brrmtatabrmtamöpmööööp`

(b) `brmtata`

(c) `brmtatabrrmmöpmöp`

Überlegen Sie weiters:

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

Lösung

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

$$
\begin{aligned}
A &\Rightarrow B\ D \\
&\Rightarrow \texttt{"br"}\ R\ \texttt{"m"}\ C\ D \\
&\Rightarrow \texttt{"brr"}\ R\ \texttt{"m"}\ C\ D \\
&\Rightarrow \texttt{"brr"}\ \varepsilon\ \texttt{"m"}\ C\ D \\
&\Rightarrow \texttt{"brrm"}\ C\ D \\
&\Rightarrow \texttt{"brrmtata"}\ B\ D \\
&\Rightarrow \texttt{"brrmtatabrm"}\ C\ D \\
&\Rightarrow \texttt{"brrmtatabrmta"}\ D \\
&\Rightarrow \texttt{"brrmtatabrmtamöpmö"}\ E\ \texttt{"p"} \\
&\Rightarrow \texttt{"brrmtatabrmtamöpmöö"}\ E\ \texttt{"p"} \\
&\Rightarrow \texttt{"brrmtatabrmtamöpmööö"}\ E\ \texttt{"p"} \\
&\Rightarrow \texttt{"brrmtatabrmtamöpmööö"}\ \varepsilon\ \texttt{"p"} \\
&\Rightarrow \texttt{"brrmtatabrmtamöpmöööp"}
\end{aligned}
$$

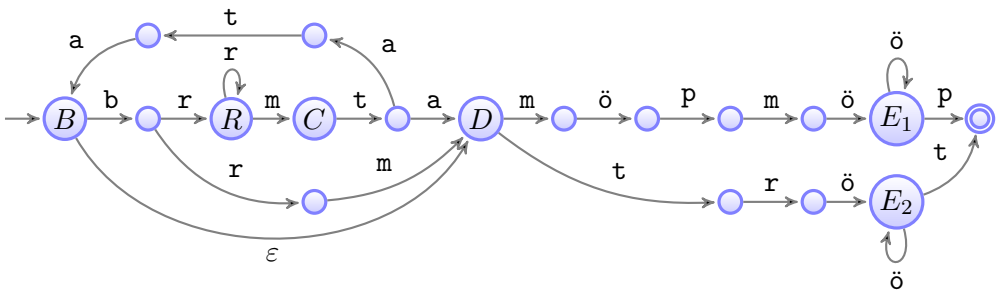
(b) Das Wort liegt nicht in der Sprache $\mathcal{L}(G)$. Aufgrund der ersten Produktion kommt am Ende immer das Nonterminal D welches dafür sorgt, dass am Ende jedes Wortes `möp` oder `tröt` (jeweils mit einer beliebigen Anzahl an ös) kommt.

In der Sprache liegt beispielsweise `brmtata` `tröt`.

(c) Das Wort liegt nicht in der Sprache $\mathcal{L}(G)$, da nach einem `brrm` (mit einem oder mehreren `r`s) immer die Silbe `ta` kommen muss: Diese Zeichenfolge `brrm` kann nur durch die erste Produktion von B zustande kommen. In diesem Fall muss aber auf `m` eine Zeichenkette folgen, die aus C erzeugt werden kann. Beide Produktionen für C liefern aber Zeichenketten, die mit `ta` beginnen.

In der Sprache liegt beispielsweise `brmtatabrrm` `ta` `möpmöp`.

(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 (3 Punkte)

DATALOG-Programme besitzen folgenden Aufbau.

- Ein *Programm* ist eine möglicherweise leere Folge von Klauseln. Eine *Klausel* ist entweder ein Faktum oder eine Regel.
- Ein *Faktum* besteht aus einer Atomformel gefolgt von einem Punkt.
- Eine *Regel* besteht aus einer Atomformel, gefolgt von den Zeichen `:-` sowie einer nicht-leeren Liste von Atomformeln, die durch Kommas (`,`) getrennt werden. Regeln enden ebenfalls mit einem Punkt.
- Eine *Atomformel* ist ein Name, dem optional eine in runden Klammern eingeschlossene Argumentliste folgen kann.
- Eine *Argumentliste* ist eine nicht-leere Folge von Namen und Variablen in beliebiger Reihenfolge, die voneinander durch Kommas getrennt werden.
- Ein *Name* ist eine nicht-leere Folge von Buchstaben und Ziffern, die mit einem Kleinbuchstaben beginnt.
- Eine *Variable* ist eine nicht-leere Folge von Buchstaben und Ziffern, die mit einem Großbuchstaben beginnt.

Das folgende Beispiel besteht aus zwei Fakten und drei Regeln; `adam`, `seth`, `istKindVon` usw. sind Namen, `X` und `Y` sind Variablen.

```
istKindVon(seth,adam).
istKindVon(enosh,seth).
istNachfahreVon(X,Y) :- istKindVon(X,Y).
istNachfahreVon(X,Z) :- istKindVon(X,Y), istNachfahreVon(Y,Z).
istMensch(X) :- istNachfahreVon(X,adam).
```

Beschreiben Sie die zulässigen DATALOG-Programme mittels einer kontextfreien Grammatik. Verwenden Sie so weit als möglich EBNF-Notationen, um die Grammatik übersichtlich zu halten und rekursive Regeln zu vermeiden.

Lösung

$\langle N, T, P, \mathit{Programm}\rangle$, wobei

$$
\begin{aligned}
N &= \{\mathit{Programm}, \mathit{Klausel}, \mathit{Faktum}, \mathit{Regel}, \mathit{Atom}, \dots\},\\
T &= \{\cdots \textit{alle Zeichen in den Produktionen zwischen Anführungszeichen} \cdots\},\\
P &= \{\ \mathit{Programm} \to \{\mathit{Klausel}\},\\
&\qquad \mathit{Klausel} \to \mathit{Faktum} \mid \mathit{Regel},\\
&\qquad \mathit{Faktum} \to \mathit{Atom}\texttt{"."},\\
&\qquad \mathit{Regel} \to \mathit{Atom}\texttt{":-"}\mathit{Atom}\{\texttt{","}\mathit{Atom}\}\texttt{"."},\\
&\qquad \mathit{Atom} \to \mathit{Name}[\texttt{"("}\mathit{Argliste}\texttt{")"}],\\
&\qquad \mathit{Argliste} \to \mathit{Arg}\{\texttt{","}\mathit{Arg}\},\\
&\qquad \mathit{Arg} \to \mathit{Name} \mid \mathit{Var},\\
&\qquad \mathit{Name} \to \mathit{KB}\{\mathit{Alphanum}\},\\
&\qquad \mathit{Var} \to \mathit{GB}\{\mathit{Alphanum}\},\\
&\qquad \mathit{Alphanum} \to \mathit{KB} \mid \mathit{GB} \mid \mathit{Num},\\
&\qquad \mathit{KB} \to \texttt{"a"} \mid \cdots \mid \texttt{"z"},\\
&\qquad \mathit{GB} \to \texttt{"A"} \mid \cdots \mid \texttt{"Z"},\\
&\qquad \mathit{Num} \to \texttt{"0"} \mid \cdots \mid \texttt{"9"}\ \}.
\end{aligned}
$$

Aufgabe 10 (3 Punkte)

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

$$
\begin{aligned}
\mathcal{V} &= \{x, y, z\}\\
\mathcal{F} &= \{\mathit{catwoman}/0,\ \mathit{riddler}/0\}\\
\mathcal{P} &= \{\mathit{Superheld}/1,\ \mathit{Schurke}/1,\ \mathit{Verrückt}/1,\ \mathit{Jagt}/2\}
\end{aligned}
$$

Geben Sie eine strukturierte kontextfreie Grammatik für die Sprache der prädikatenlogischen Formeln über diesen Symbolmengen an. Beispiele für Wörter, die in dieser Formelsprache liegen:
$\forall x\,(\mathit{Schurke}(x) \supset \exists y\,(\mathit{Superheld}(y) \wedge \mathit{Jagt}(y, x)))$
$\exists x\,(\neg \mathit{Jagt}(x, x) \wedge \mathit{Jagt}(x, \mathit{catwoman}))$

Lösung

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

$V = \{\mathit{Formel}, \mathit{Op}, \ldots, \mathit{FS}_0, \mathit{Var}\}$,

$T = \{\texttt{"}\top\texttt{"}, \texttt{"}\bot\texttt{"}, \texttt{"}\neg\texttt{"}, \texttt{"(","")","},\texttt{"},\texttt{"}\wedge\texttt{"}, \ldots, \texttt{"}\forall\texttt{"}, \texttt{"}\exists\texttt{"}, \texttt{"}\mathit{Superheld}\texttt{"}, \ldots, \texttt{"}\mathit{riddler}\texttt{"}, \texttt{"x"}, \texttt{"y"}, \texttt{"z"}\}$,

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

$\mathit{Op} \rightarrow \texttt{"}\wedge\texttt{"} \mid \texttt{"}\uparrow\texttt{"} \mid \texttt{"}\vee\texttt{"} \mid \texttt{"}\downarrow\texttt{"} \mid \texttt{"}\equiv\texttt{"} \mid \texttt{"}\not\equiv\texttt{"} \mid \texttt{"}\supset\texttt{"} \mid \texttt{"}\subset\texttt{"}$,

$\mathit{Qu} \rightarrow \texttt{"}\forall\texttt{"} \mid \texttt{"}\exists\texttt{"}$,

$\mathit{Term} \rightarrow \mathit{Var} \mid \mathit{FS}_0$,

$\mathit{PS}_1 \rightarrow \texttt{"}\mathit{Superheld}\texttt{"} \mid \texttt{"}\mathit{Schurke}\texttt{"} \mid \texttt{"}\mathit{Verrückt}\texttt{"}$,

$\mathit{PS}_2 \rightarrow \texttt{"}\mathit{Jagt}\texttt{"}$,

$\mathit{FS}_0 \rightarrow \texttt{"}\mathit{catwoman}\texttt{"} \mid \texttt{"}\mathit{riddler}\texttt{"}$,

$\mathit{Var} \rightarrow \texttt{"x"} \mid \texttt{"y"} \mid \texttt{"z"}\}$.

Aufgabe 11 (6 Punkte)

Seien *Jagt*/2, *Superheld*/1, *Verrückt*/1 und *Schurke*/1 Prädikatensymbole sowie *catwoman* und *riddler* Konstantensymbole mit folgender Bedeutung:

$\mathit{Jagt}(x, y)$	... x jagt y	*catwoman*	... Catwoman
$\mathit{Superheld}(x)$	... x ist ein Superheld	*riddler*	... Riddler
$\mathit{Verrückt}(x)$	... x ist verrückt		
$\mathit{Schurke}(x)$	... x ist ein Schurke		

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

(a) Alle Superhelden jagen verrückte Schurken.

(b) Manche Verrückte jagen Catwoman aber nicht Riddler.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Aquaman}, \text{Riddler}, \text{Superman}, \text{Batman}, \text{Catwoman}, \text{WonderWoman}, \\
&\quad \text{Joker}, \text{Firebug}, \text{Pinguin}, \text{Cluemaster}\} \\
I(\mathit{Superheld}) &= \{\text{Superman}, \text{Batman}, \text{Catwoman}, \text{WonderWoman}\} \\
I(\mathit{Verrückt}) &= \{\text{Joker}, \text{Riddler}, \text{Firebug}\} \\
I(\mathit{Schurke}) &= \{\text{Joker}, \text{Firebug}, \text{Pinguin}, \text{Catwoman}, \text{Cluemaster}\} \\
I(\mathit{Jagt}) &= \{(\text{Superman}, \text{Riddler}), (\text{Superman}, \text{Joker}), \\
&\quad (\text{Batman}, \text{Riddler}), (\text{Batman}, \text{Pinguin}), (\text{Batman}, \text{Cluemaster}), \\
&\quad (\text{Catwoman}, \text{Riddler}), \\
&\quad (\text{WonderWoman}, \text{Riddler}), (\text{WonderWoman}, \text{Joker}), \\
&\quad (\text{Aquaman}, \text{Riddler}), (\text{Aquaman}, \text{Firebug}), (\text{Aquaman}, \text{Catwoman})\}, \\
I(\mathit{catwoman}) &= \text{Catwoman} \qquad I(\mathit{riddler}) = \text{Riddler}
\end{aligned}$$

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) $\forall x\,(\mathit{Superheld}(x) \supset \exists y\,(\mathit{Verrückt}(y) \wedge \mathit{Jagt}(x, y)))$

(d) $\forall x\,\mathit{Jagt}(x, \mathit{riddler})$

(e) $\forall x\,(\mathit{Schurke}(x) \supset \exists y\,(\mathit{Superheld}(y) \wedge \mathit{Jagt}(y, x)))$

(f) $\exists x\,(\neg \mathit{Jagt}(x, x) \wedge \mathit{Jagt}(x, \mathit{catwoman}))$

Lösung

(a) $\forall x(\mathit{Superheld}(x) \supset \exists y(\mathit{Schurke}(y) \wedge \mathit{Verrückt}(y) \wedge \mathit{Jagt}(x, y)))$

(b) $\exists x(\mathit{Verrückt}(x) \wedge \mathit{Jagt}(x, \mathit{catwoman}) \wedge \neg \mathit{Jagt}(x, \mathit{riddler}))$

(c) Übersetzung: Alle Superhelden jagen etwas Verrücktes.
Diese Aussage ist wahr in I, da Superman, Batman und Catwoman Riddler jagen und WonderWoman Joker jagt.

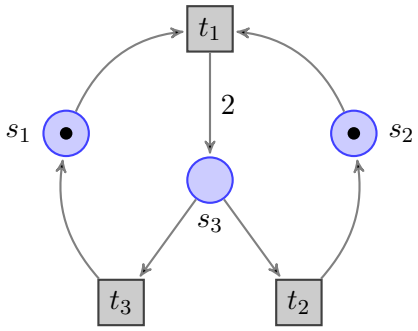
(d) Übersetzung: Alles jagt Riddler.
Diese Aussage ist falsch, da etwa Riddler nicht Riddler jagt.

(e) Übersetzung: Alle Schurken werden von einem Superhelden gejagt.
Diese Aussage ist falsch in I, da Firebug von keinem Superhelden gejagt wird (Aquaman ist kein Superheld!).

(f) Übersetzung: Es gibt etwas, das sich nicht selber aber Catwoman jagt.
Diese Aussage ist wahr in I, da Aquaman Catwoman jagt, aber nicht sich selber ($(\text{Aquaman}, \text{Aquaman}) \notin I(\mathit{Jagt})$).

Aufgabe 12 (4 Punkte)

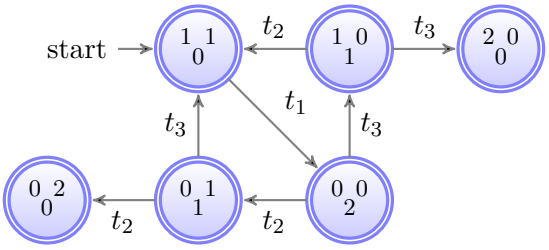
Gegeben sei das folgende Petri-Netz mit Anfangsmarkierung.



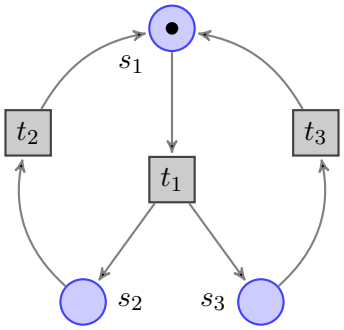
Fassen Sie die Bezeichnungen der Transitionen als Alphabet auf und die Markierungen (also die jeweiligen Belegungen der Stellen mit Marken) als Zustände. Beschreiben Sie die möglichen Reihenfolgen, in der die Transitionen feuern und die Markierungen auftreten können, mit Hilfe eines endlichen Automaten. Der Automat soll also Wörter wie $t_1t_2t_3$ akzeptieren, weil die Transitionen in dieser Reihenfolge feuern können, nicht aber t_3t_2.
Ist es immer möglich, das Verhalten eines Petrinetzes auf diese Weise durch einen endlichen Automaten zu beschreiben? Wenn ja, geben Sie Argumente dafür. Wenn nein, geben Sie als Gegenbeispiel ein Petri-Netz an, bei dem das nicht geht.

Lösung

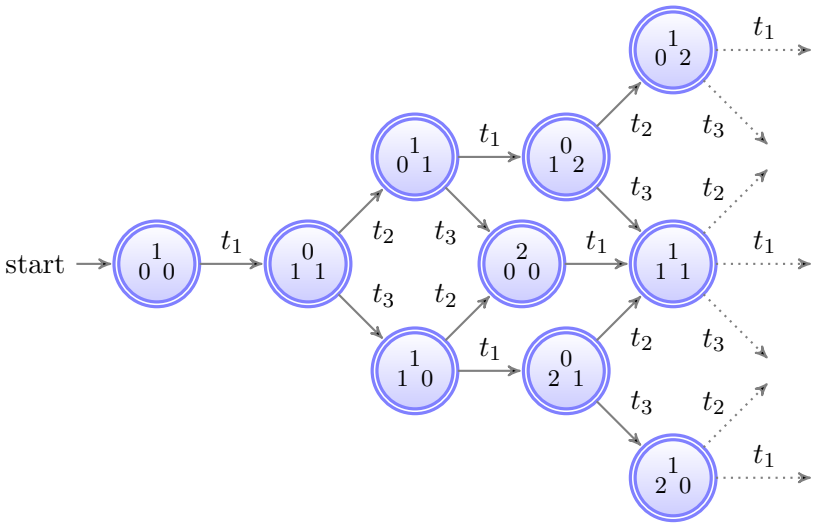
Die Abläufe im Petrinetz lassen sich durch einen Automaten mit (in diesem Fall) endlich vielen Zuständen beschreiben. Die Markierungen bilden die Zustände, die Transitionen das Alphabet. Erhält man aus einer Markierung m durch Feuern einer Transition t eine Markierung m', dann gibt es einen Übergang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



Ist es immer möglich, das Verhalten eines Petrinetzes durch einen endlichen Automaten zu beschreiben? Nein, da die Anzahl der Marken pro Stelle beim Abarbeiten eines Petri-Netze im Allgemeinen nicht beschränkt ist und dadurch unendlich viele verschiedene Markierungen (und damit Zustände) auftreten können. Beispiel für ein solches Petrinetz:



Mit jedem Feuern der Transition t_1 steigt die Anzahl der Marken im Netz. Wenn wir die Markierungen wie oben durch drei Zahlen ${}_{n_2}{}^{n_1}{}_{n_3}$ darstellen, wobei n_i die Anzahl der Marken in der Stelle s_i angibt, erhalten wir folgenden Anfang eines unendlichen Automaten.



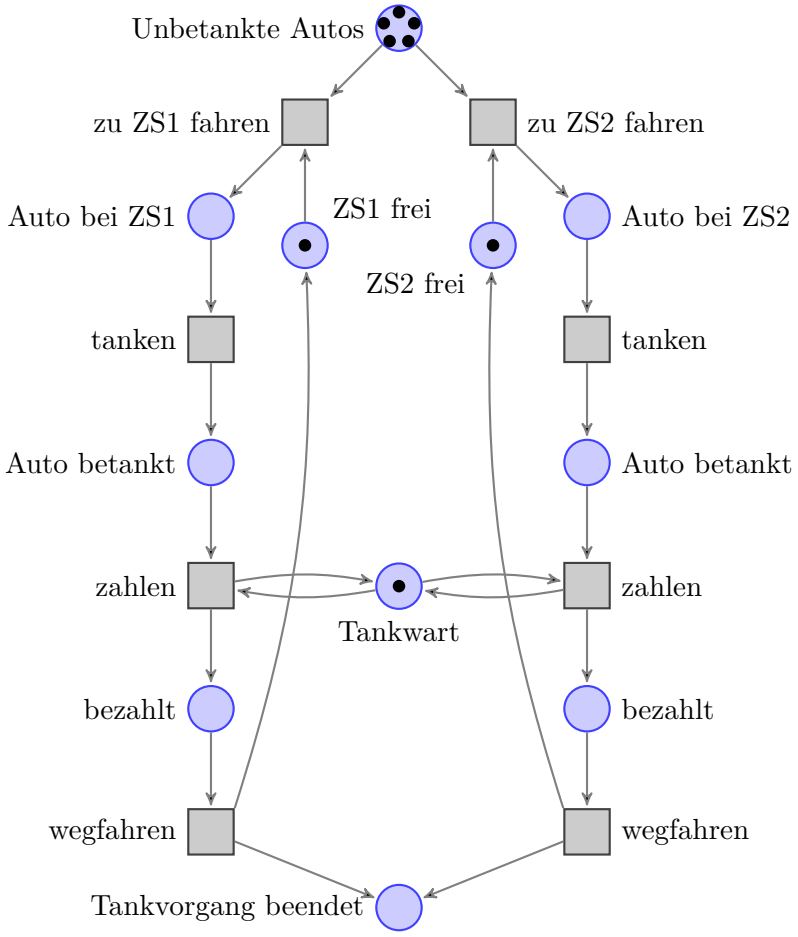
Genau genommen ist damit noch nicht gezeigt, dass nicht doch ein endlicher Automat ausreicht: Es könnten ja verschiedene Zustände zueinander äquivalent sein, d.h., die Menge der von dort aus akzeptierten Wörter könnte identisch sein. In diesem Fall könnte man diese Zustände miteinander verschmelzen, wodurch sich die Zustände auf eine endliche Anzahl reduzieren könnten. Das trifft hier allerdings nicht zu: Wie oft z.B. t_2 hintereinander feuern kann, hängt davon ab, wie oft davor t_1 gefeuert hat. Diese Anzahl ist zu jedem Zeitpunkt beschränkt, kann aber beliebig groß werden. Daher sind die verschiedenen Zustände nicht äquivalent zueinander, eine endliche Anzahl reicht nicht. (Ein formaler Beweis ist das immer noch nicht; dafür müsste man etwa das *Pumping Lemma* anwenden, das aber erst in einer späteren Lehrveranstaltung besprochen werden wird.)

Aufgabe 13 (4 Punkte)

Der Tankvorgang an einer Selbstbedienungstankstelle besteht aus folgenden Schritten. Der Kunde fährt zu einer freien Zapfsäule und betankt das Auto. Anschließend begleicht

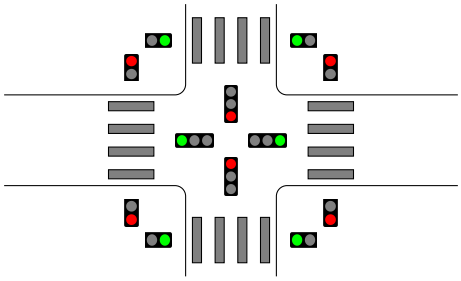
er beim Tankwart die Rechnung. Nach dem Zahlen fährt der Kunde von der Zapfsäule weg und gibt sie damit für den nächsten Kunden frei.
Modellieren Sie eine Tankstelle mit zwei Zapfsäulen mit Hilfe eines Petri-Netzes. Berücksichtigen Sie alle Schritte des beschriebenen Ablaufs. Beachten Sie, dass an jeder Zapfsäule zu jedem Zeitpunkt höchstens ein Auto betankt werden kann und dass der Tankwart immer nur einen Bezahlvorgang durchführen kann. Nehmen Sie für die Anfangsmarkierung an, dass fünf Autos die Tankstelle nützen wollen. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben.

Lösung



Aufgabe 14 (4 Punkte)

Eine kreuzförmige Straßenkreuzung ist mit Fahrzeugampeln (vier Phasen) und Fußgängerampeln (zwei Phasen) für alle Richtungen ausgestattet. Es gibt keine speziellen Ampeln für Abbieger.

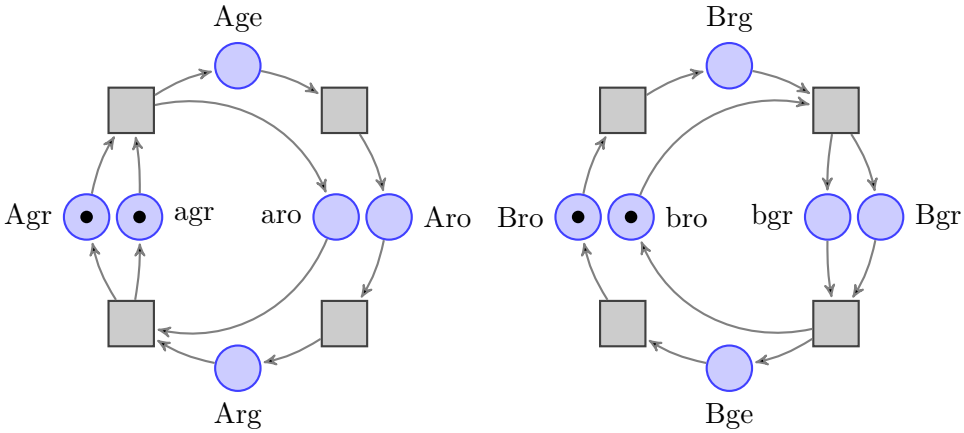


Modellieren Sie die Schaltfolgen der Ampeln mit Hilfe eines Petri-Netzes. Geben Sie eine geeignete Anfangsmarkierung an. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben.

Hinweis: Ampeln, die sich identisch verhalten, können als eine einzige Ampel betrachtet werden.

Lösung

Fahrzeugampeln können vier Phasen durchlaufen, die wir gr (grün), ge (gelb), ro (rot) und rg (rot-gelb) nennen. Analog durchlaufen Fußgängerampeln die Phasen ro und gr. Wir müssen in unserer Modellierung vier Ampeln unterscheiden: zwei Fahrzeugampeln (A und B), die gegengleich geschaltet sind, und zwei Fußgängerampeln (a und b), die jeweils parallel zu den jeweiligen Fahrzeugampeln geschaltet sind. Gegengleich soll hier bedeuten, dass die eine Fahrzeugampel rot ist, während die andere die Nicht-Rot-Phasen durchläuft. Parallel soll bedeuten, dass die Fußgängerampel nur grün ist, wenn die zugehörige Fahrzeugampel grün ist, und rot sonst. In unserer Modellierung werden wir keine überlappenden Rot-Phasen gegengleicher Ampeln vorsehen, und keine unterschiedlichen Grünphasen paralleler Ampeln.
Im Petri-Netz sehen wir eine Stelle für jede Ampel und jede Phase vor. Somit benötigen wir 12 Stellen, die wir Agr, Age, Aro, Arg, agr, aro, Bgr, Bge, Bro, Brg, bgr und bro nennen.
Wenn wir die beiden Ampelpaare A/a und B/b zunächst noch nicht synchronisieren, erhalten wir die folgenden beiden Petri-Netze.



Nun müssen wir noch dafür sorgen, dass immer eines der beiden Ampelpaare rot ist, indem wir die Transitionen zusammenziehen.