3. Übungsblatt (mit Lösungen)

3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

6. Juni 2018

Aufgabe 1 (5 Punkte)

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

$$
\begin{aligned}
N &= \{A, B, C, D\} \\
T &= \{\texttt{aha}, \texttt{dadada}, \texttt{di}, \texttt{ding}\} \\
P &= \{\, A \to \texttt{"aha␣aha␣aha␣"}\, A \mid \texttt{"aha"} \mid B \ , \\
&\qquad B \to \texttt{"dadada␣"}\, C \ , \\
&\qquad C \to \texttt{"di␣"}\, D \ , \\
&\qquad D \to \texttt{"ding␣"}\, D \mid \texttt{"ding"} \mid A\}
\end{aligned}
$$

Das Zeichen ␣ steht für das Leerzeichen.
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) `aha aha aha dadada di ding aha`

(b) `dadada di aha`

(c) `dadada di ding ding dadada`

Überlegen Sie weiters:

(d) Wie lautet das kürzeste Wort der Sprache?

(e) 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 $\mathcal{L}(G)$.

$$\begin{aligned} A &\Rightarrow \texttt{"aha aha aha"}\, A \\ &\Rightarrow \texttt{"aha aha aha"}\, B \\ &\Rightarrow \texttt{"aha aha aha dadada"}\, C \\ &\Rightarrow \texttt{"aha aha aha dadada di"}\, D \\ &\Rightarrow \texttt{"aha aha aha dadada di ding"}\, D \\ &\Rightarrow \texttt{"aha aha aha dadada di ding"}\, A \\ &\Rightarrow \texttt{"aha aha aha dadada di ding aha"} \end{aligned}$$

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

$$\begin{aligned} A &\Rightarrow B \\ &\Rightarrow \texttt{"dadada"}\, C \\ &\Rightarrow \texttt{"dadada di"}\, D \\ &\Rightarrow \texttt{"dadada di"}\, A \\ &\Rightarrow \texttt{"dadada di aha"} \end{aligned}$$

(c) Das Wort liegt nicht in der Sprache $\mathcal{L}(G)$. Zur Begründung reicht es nicht, eine Ableitung anzugeben, die nicht zum gewünschten Resultat führt, sondern wir müssen argumentieren, warum *keine* Ableitung zu diesem Wort führen kann. Wir stellen zunächst fest, dass die Zeichenkette `dadada` nur durch die Produktion

$$B \to \texttt{"dadada␣"}\, C$$

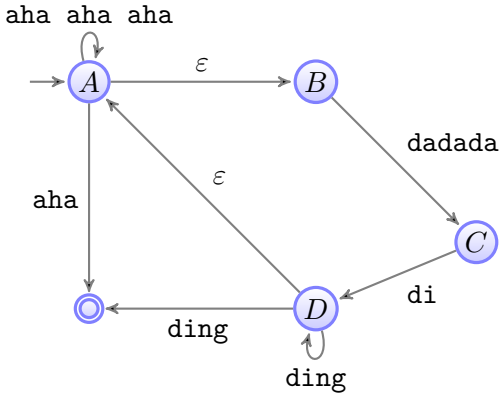
in ein Wort gelangen kann. Daher muss auf `dadada` immer ein Wort folgen, das aus dem Nonterminal C erzeugt werden kann. Die einzige Produktion für C ist aber

$$C \to \texttt{"di␣"}\, D \ .$$

Somit muss in jedem ableitbaren Wort auf `dadada␣` immer die Symbolfolge `di␣` folgen. Offenbar ist das am Ende des zu untersuchenden Wortes nicht der Fall, also ist es in der Grammatik nicht ableitbar.

Das nächstkürzere ableitbare Wort ist `dadada di ding ding`, das nächst längere `dadada di ding ding dadada di ding`.

(d) `aha`



(e)

(Diese Aufgabe wurde von Unterrichtsmaterialien des Gymnasiums Odenthal inspiriert.[1])

Aufgabe 2 (4 Punkte)

Gegeben seien Zahlen mit dem folgenden Aufbau:

- Jede Zahl hat optional ein positives oder negatives Vorzeichen.
- Falls es einen Nachkommateil gibt, trennt ein Komma die Vorkommastellen von den Nachkommastellen.
- Der Vorkommateil und der Nachkommateil (falls er existiert) bestehen aus mindestens einer Ziffer.

(a) Spezifizieren Sie die Sprache $\mathcal{D}$ solcher Dezimalzahlen mit Hilfe einer kontextfreien Grammatik. Verwenden Sie EBNF-Notationen, um die Grammatik übersichtlich zu strukturieren.

(b) Mathematisch sind Zahlen wie zum Beispiel $-05{,}73$ oder $04{,}00$ zwar in Ordnung, oft werden die führenden Nullen aber weggelassen. Sei $\mathcal{D}^*$ die Sprache der Dezimalzahlen ohne führende Nullen, also etwa ohne die beiden angeführten Zahlen. Geben Sie eine kontextfreie Grammatik für $\mathcal{D}^*$ an. Verwenden Sie EBNF-Notationen, um die Grammatik übersichtlich zu strukturieren.

Anmerkung: Zahlen wie zum Beispiel 0,5 oder 0,500 sollen weiterhin akzeptiert werden.

[1] http://projekte.gymnasium-odenthal.de/informatik/anzeige.php?site=&type=pdf&dateiname=2003-2004-3%20Sprachen-AVL-Baeume-Prolog.pdf&verlauf=Informatik/Jahrgangsstufe%20Q/001%20Klausuren/Q2/2003-2004-3%20Sprachen-AVL-Baeume-Prolog.pdf

Lösung

(a) Die Sprache $\mathcal{D}$ wird durch die Grammatik $\langle N, T, P, \mathit{Zahl}\rangle$ beschrieben, wobei

$$
\begin{aligned}
N &= \{\mathit{Zahl}, \mathit{Ziffern}, \mathit{Ziffer}, \mathit{Vorzeichen}\},\\
T &= \{\cdots \textit{alle Zeichen in den Produktionen zwischen Anführungszeichen} \cdots\},\\
P &= \{\quad \mathit{Zahl} \to [\mathit{Vorzeichen}]\ \mathit{Ziffern}\ [\texttt{","}\ \mathit{Ziffern}]\ ,\\
&\qquad \mathit{Ziffern} \to \mathit{Ziffer}\ \{\mathit{Ziffer}\}\ ,\\
&\qquad \mathit{Ziffer} \to \texttt{"0"} \mid \cdots \mid \texttt{"9"}\ ,\\
&\qquad \mathit{Vorzeichen} \to \texttt{"+"} \mid \texttt{"-"}\quad \}
\end{aligned}
$$

(b) Die Sprache $\mathcal{D}^*$ wird durch die Grammatik $\langle N, T, P, \mathit{Zahl}\rangle$ beschrieben, wobei

$$
\begin{aligned}
N &= \{\mathit{Zahl}, \mathit{Vorkomma}, \mathit{Nachkomma}, \mathit{Ziffern}, \mathit{Ziffer}, \mathit{PosZiffer}, \mathit{Vorzeichen}\},\\
T &= \{\cdots \textit{alle Zeichen in den Produktionen zwischen Anführungszeichen} \cdots\},\\
P &= \{\quad \mathit{Zahl} \to [\mathit{Vorzeichen}]\ \mathit{Vorkomma}\ [\mathit{Nachkomma}]\ ,\\
&\qquad \mathit{Vorkomma} \to \mathit{Ziffer} \mid \mathit{PosZiffer}\ \mathit{Ziffern}\ ,\\
&\qquad \mathit{Nachkomma} \to \texttt{","}\ \mathit{Ziffern}\ ,\\
&\qquad \mathit{Ziffern} \to \mathit{Ziffer}\ \{\mathit{Ziffer}\}\ ,\\
&\qquad \mathit{Ziffer} \to \texttt{"0"} \mid \mathit{PosZiffer}\ ,\\
&\qquad \mathit{PosZiffer} \to \texttt{"1"} \mid \cdots \mid \texttt{"9"}\ ,\\
&\qquad \mathit{Vorzeichen} \to \texttt{"+"} \mid \texttt{"-"}\quad \}
\end{aligned}
$$

Aufgabe 3 (4 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 wie 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 4 (5 Punkte)

Wählen Sie geeignete Prädikaten- und Konstantensymbole und übersetzen Sie die folgenden Sätze in prädikatenlogische Formeln.

(a) Tobias hat Hunger.

(b) Es gibt nichts zu Essen.

(c) Tobias mag keine Falafel aber er mag Kebab.

(d) Alle mögen Kebab.

(e) Jedes Kind, das Hunger hat, mag Kebab.

Lösung

Seien $Mag/2$, $Hunger/1$, $Essen/1$ und $Kind/1$ Prädikatensymbole sowie *tobias*, *kebab* und *falafel* Konstantensymbole mit folgender Bedeutung:

$Mag(x, y)$	... x mag y	*tobias*	... Tobias
$Kind(x)$	... x ist ein Kind	*falafel*	... Falafel
$Hunger(x)$	... x hat Hunger	*kebab*	... Kebab
$Essen(x)$	... x ist Essen		

(a) $Hunger(tobias)$

(b) $\neg\exists x Essen(x)$

(c) $\neg Mag(tobias, falafel) \wedge Mag(tobias, kebab)$

(d) $\forall x Mag(x, kebab)$

(e) $\forall x((Kind(x) \wedge Hunger(x)) \supset Mag(x, kebab))$

Aufgabe 5 (6 Punkte)

Seien $Feiert/2$, $Fest/1$, $Kind/1$ und $Lustig/1$ Prädikatensymbole sowie *weihnachten* und *hannukah* Konstantensymbole mit folgender Bedeutung:

$Fest(x)$	... x ist ein Fest	$Feiert(x, y)$	... x feiert y
$Kind(x)$	... x ist ein Kind	*weihnachten*	... Weihnachten
$Lustig(x)$	... x ist lustig	*hannukah*	... Hannukah

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

(a) Kein Kind feiert sowohl Weihnachten als auch Hannukah.

(b) Es gibt lustige Feste, die von allen Kindern gefeiert werden.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Mia}, \text{Max}, \text{Anna}, \text{Tim}, \text{Fasching}, \text{Ostern}, \\
&\quad \text{Laternenfest}, \text{Weihnachten}, \text{Hannukah}, \text{Advent}\} \\
I(Fest) &= \{\text{Fasching}, \text{Ostern}, \text{Laternenfest}, \text{Weihnachten}\} \\
I(Kind) &= \{\text{Mia}, \text{Max}, \text{Anna}\} \\
I(Lustig) &= \{\text{Fasching}, \text{Laternenfest}, \text{Advent}\} \\
I(Feiert) &= \{(\text{Mia}, \text{Fasching}), (\text{Mia}, \text{Ostern}), (\text{Mia}, \text{Weihnachten}), \\
&\quad (\text{Max}, \text{Laternenfest}), (\text{Max}, \text{Weihnachten}), (\text{Max}, \text{Ostern}), \\
&\quad (\text{Tim}, \text{Anna}), (\text{Tim}, \text{Hannukah}), \\
&\quad (\text{Anna}, \text{Fasching}), (\text{Anna}, \text{Weihnachten})\} \\
I(weihnachten) &= \text{Weihnachten} \\
I(hannukah) &= \text{Hannukah}
\end{aligned}$$

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

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

(d) $\exists x\,(\mathit{Kind}(x) \wedge \exists y\,(\mathit{Fest}(y) \wedge \mathit{Lustig}(y) \wedge \mathit{Feiert}(x,y)))$

(e) $\forall x\,(\mathit{Feiert}(x,\mathit{weihnachten}) \not\equiv \mathit{Feiert}(x,\mathit{hannukah}))$

(f) $\forall x\,(\mathit{Fest}(x) \supset \exists y\,(\mathit{Kind}(y) \wedge \mathit{Lustig}(y) \wedge \mathit{Feiert}(x,y)))$

Lösung

(a) $\neg\exists x\,(\mathit{Kind}(x) \wedge \mathit{Feiert}(x,\mathit{weihnachten}) \wedge \mathit{Feiert}(x,\mathit{hannukah}))$ oder
$\forall x\,(\mathit{Kind}(x) \supset (\mathit{Feiert}(x,\mathit{weihnachten}) \uparrow \mathit{Feiert}(x,\mathit{hannukah})))$ oder
$\forall x\,(\mathit{Kind}(x) \supset (\neg\mathit{Feiert}(x,\mathit{weihnachten}) \vee \neg\mathit{Feiert}(x,\mathit{hannukah})))$

(b) $\exists x\,((\mathit{Fest}(x) \wedge \mathit{Lustig}(x) \wedge \forall y\,(\mathit{Kind}(y) \supset \mathit{Feiert}(y,x)))$ oder
$\exists x\forall y\,((\mathit{Fest}(x) \wedge \mathit{Lustig}(x) \wedge \mathit{Kind}(y)) \supset \mathit{Feiert}(y,x))$

(c) Übersetzung: Alle Kinder feiern ein Fest.
Wahr: Die einzigen Kinder sind Mia, Max und Anna, und Mia feiert Fasching, Max feiert das Laternfest und Anna feiert Fasching.

(d) Übersetzung: Es gibt ein Kind, das ein lustiges Fest feiert.
Wahr, da Mia ein Kind und Fasching sowohl lustig als auch ein Fest ist und $(\text{Mia}, \text{Fasching}) \in I(\mathit{Feiert})$.

(e) Übersetzung: Alles feiert entweder Weihnachten oder Hannukah.
Falsch, da $(\text{Weihnachten}, \text{Weihnachten}) \notin I(\mathit{Feiert})$ und $(\text{Weihnachten}, \text{Hannukah}) \notin I(\mathit{Feiert})$.

(f) Übersetzung: Jedes Fest feiert mindestens ein lustiges Kind.
Falsch. Es gibt kein Fest, das ein Kind feiert.

Aufgabe 6 (3 Punkte)

Zeigen Sie, dass die Formeln

$$\neg\exists x(P(x) \wedge \forall y(Q(y) \supset R(x,y))) \quad \text{und} \quad \forall y\exists x(\neg P(y) \vee (Q(x) \wedge \neg R(y,x)))$$

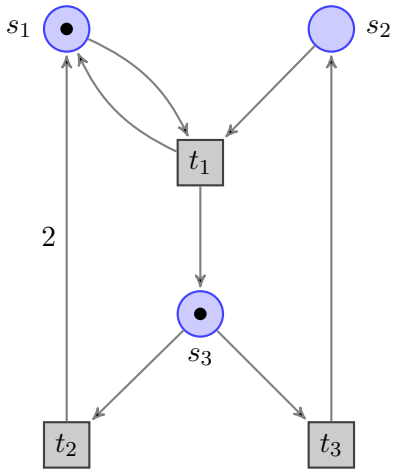
logisch äquivalent sind.

Lösung

$$
\begin{array}{ll}
\neg\exists x(P(x) \wedge \forall y(Q(y) \supset R(x,y))) & \\
= \neg\exists x(\forall y P(x) \wedge \forall y(Q(y) \supset R(x,y))) & \forall x F = F \text{ wenn } x \text{ nicht in } F \text{ vorkommt} \\
= \neg\exists x \forall y(P(x) \wedge (Q(y) \supset R(x,y))) & \forall x(F \wedge G) = \forall x F \wedge \forall x G \\
= \forall x \neg\forall y(P(x) \wedge (Q(y) \supset R(x,y))) & \neg\exists x F = \forall x \neg F \\
= \forall x \exists y \neg(P(x) \wedge (Q(y) \supset R(x,y))) & \neg\forall x F = \exists x \neg F \\
= \forall x \exists y \neg(P(x) \wedge (\neg Q(y) \vee R(x,y))) & F \supset G = \neg F \vee G \\
= \forall x \exists y(\neg P(x) \vee (Q(y) \wedge \neg R(x,y))) & \neg(F \vee G) = \neg F \wedge \neg G,\ \neg(F \wedge G) = \neg F \vee \neg G \text{ (de Morgan)} \\
= \forall z \exists y(\neg P(z) \vee (Q(y) \wedge \neg R(z,y))) & \forall x F[x] = \forall y F[y] \text{ falls } y \text{ nicht in } F[x] \text{ vorkommt} \\
= \forall z \exists x(\neg P(z) \vee (Q(x) \wedge \neg R(z,x))) & \exists x F[x] = \exists y F[y] \text{ falls } y \text{ nicht in } F[x] \text{ vorkommt} \\
= \forall y \exists x(\neg P(y) \vee (Q(x) \wedge \neg R(y,x))) & \forall x F[x] = \forall y F[y] \text{ falls } y \text{ nicht in } F[x] \text{ vorkommt}
\end{array}
$$

Aufgabe 7 (4 Punkte)

Gegeben sei das folgende Petri-Netz mit Anfangsmarkierung.

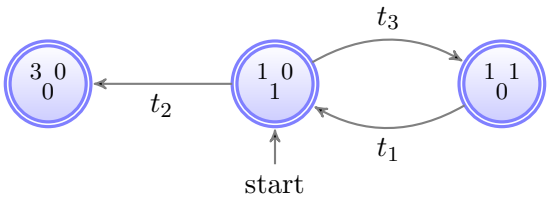


Fassen Sie die Bezeichnungen der Transitionen als Alphabet auf und die Markierungen (also die jeweiligen Belegungen der Stellen mit Marken) als Zustände. Beschreiben Sie die möglichen Reihenfolgen, in der die Transitionen feuern und die Markierungen auftreten können, mit Hilfe eines endlichen Automaten. Der Automat soll also Wörter wie $t_3t_1t_2$ akzeptieren, weil die Transitionen in dieser Reihenfolge feuern können, nicht aber t_2t_2.

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 vier Zahlen ${}^{n_1}{}_{n_3}{}^{n_2}$ dar, wobei n_i die Anzahl der Marken in der Stelle s_i angibt. Die endlichen Reihenfolgen, in denen die Transitionen feuern können, entsprechen der Sprache des folgenden Automaten.

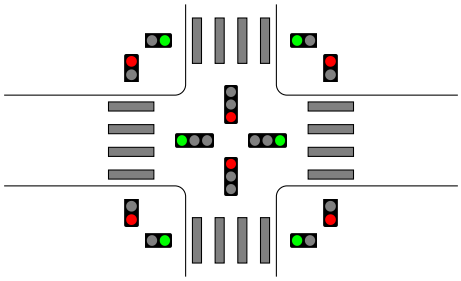


Diese Transitionsfolgen lassen sich auch durch den folgenden regulären Ausdruck in algebraischer Notation beschreiben:

$$(t_3t_1)^* + (t_3t_1)^*t_3 + (t_3t_1)^*t_2$$

Aufgabe 8 (4 Punkte)

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



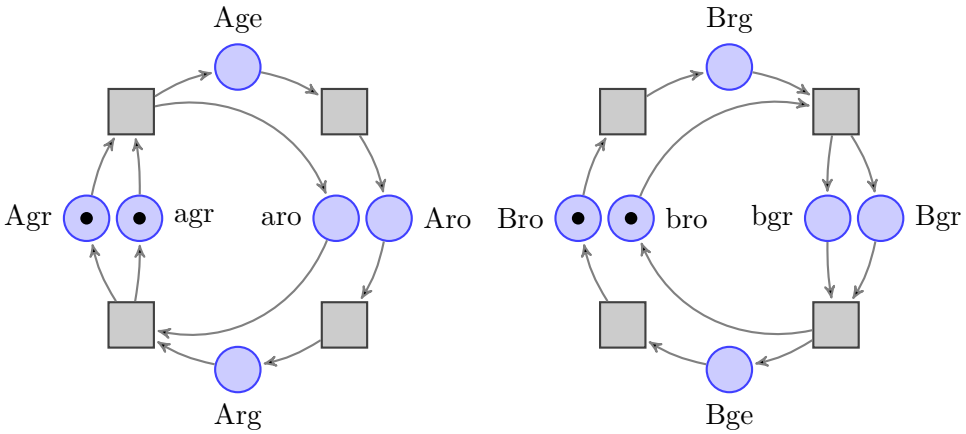
Modellieren Sie die Schaltfolgen der Ampeln mit Hilfe eines Petri-Netzes. Geben Sie eine geeignete Anfangsmarkierung an. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben.

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

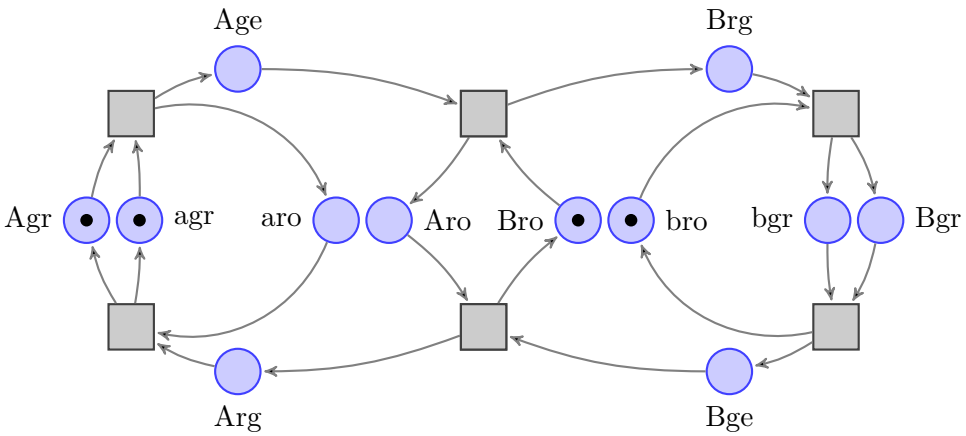
Lösung

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 eine der beiden Ampelpaare rot ist, indem wir die Transitionen zusammenziehen.



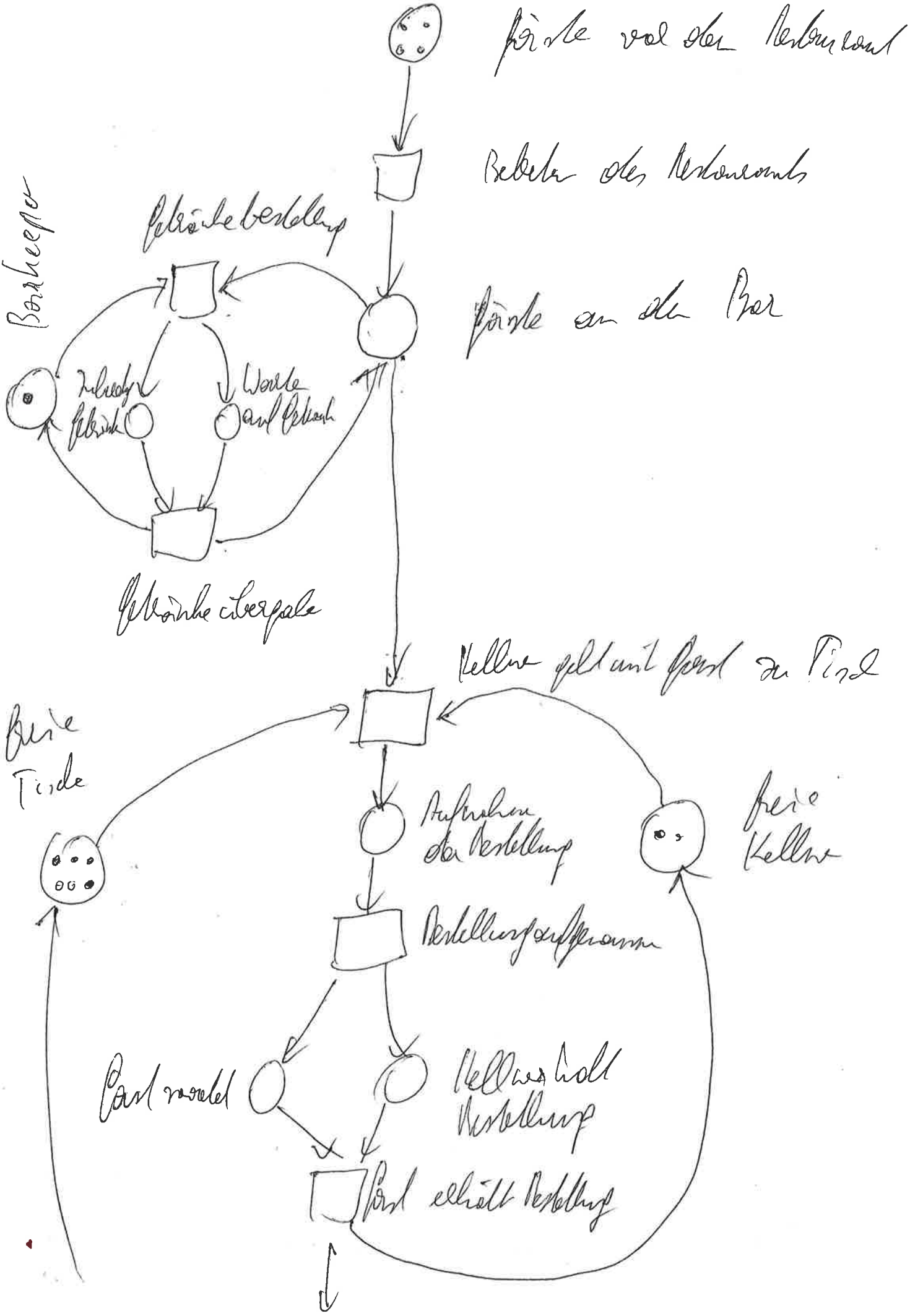
Aufgabe 9 (4 Punkte)

Wenn Gäste das Restaurant *Chez Pierre* betreten, stellen sie sich zunächst zur Bar, um auf einen Tisch zu warten. Während des Wartens können sie beim Barkeeper Getränke bestellen. Der Barkeeper bereitet jedes Getränk einzeln zu und übergibt es dann dem

Gast. Hat ein Gast ein Getränk bestellt, muss er auf dieses an der Bar warten und kann währenddessen nicht zum Tisch gehen.
Ist ein Tisch frei, holt der Kellner einen Gast bei der Bar ab und bringt ihn zum Tisch. Dort wartet er gleich auf die Bestellung und holt diese aus der Küche. Erst danach kann er sich um einen weiteren Gast kümmern.
Hat ein Gast fertig gegessen, zahlt er bei einem freien Kellner und wird von diesem zur Tür gebracht. Anschließend räumt der Kellner den Tisch ab, um ihn für den nächsten Gast vorzubereiten. Anschließend kann er sich wieder um einen Gast kümmern.
Modellieren Sie dieses System mit Hilfe eines Petri-Netzes. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben. Nehmen Sie an, dass es 6 Tische gibt, die zu Beginn alle frei sind. An jedem Tisch kann nur ein Gast Platz nehmen. Im Restaurant arbeiten ein Barkeeper und zwei Kellner. Zu Beginn befinden sich 4 Gäste vor dem Restaurant, die es betreten wollen. Geben Sie den Stellen und Transitionen geeignete Bezeichnungen, die ihre Rolle beschreiben.

Lösung

Die Lösung ist nicht eindeutig und hängt von den Details ab, die man modelliert. Eine mögliche Lösung ist die folgende.


Barkeeper
Getränkebestellung
Getränke übergabe
Kellner geht zum Gast zu Tisch
Freie Tische
Aufnahme der Bestellung
freie Kellner
Bestellung aufgenommen
Gast wartet
Kellner holt Bestellung
Gast erhält Bestellung