2. Übungsblatt (mit Lösungen)

3.0 VU Formale Modellierung SS2022

Gernot Salzer, Marion Scholz

3. Juni 2022

Aufgabe 1 (4 Punkte)

Sei L die Sprache

$$\{\, w \in \{\texttt{E}, \texttt{F}, \texttt{H}, \texttt{R}\}^* \mid w \text{ enthält das Teilwort } \texttt{FEE} \text{ oder } \texttt{REH} \,\} \ .$$

(a) Geben Sie einen Posix Extended Regular Expression an, der die Sprache L beschreibt.

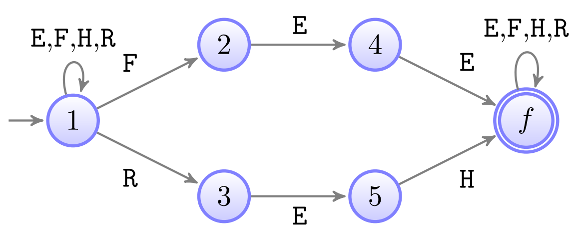
(b) Geben Sie einen nichtdeterministischen Automaten an, der die Sprache L akzeptiert. Der Automat soll der Definition der Sprache direkt entsprechen, sodass die Korrektheit der Modellierung unmittelbar einsichtig ist.

(c) Konstruieren Sie mit Hilfe des in der Vorlesung besprochenen Determinisierungsverfahrens zu Ihrem nichtdeterministischen Automaten einen äquivalenten deterministischen.

Lösung

(a) `[EFHR]*(FEE|REH)[EFHR]*` oder `^[EFHR]*(FEE|REH)[EFHR]*$` (falls die Zeichenkette die gesamte Zeile einnehmen soll)

(b) Ein indeterministischer Automat, der diese Sprache darstellt, ist der folgende:



(c) Wir stellen die Übergangsfunktion als Tabelle dar, da diese besser als Ausgangsbasis für die Determinisierung geeignet ist. (Genauer: Wir bestimmen das Ergebnis der erweiterten Übergangsfunktion δ^* für jeden Zustand und jedes Eingabesymbol. Wenn es ε-Übergänge gibt, müssen auch längere Pfade betrachtet werden.)

δ^*	E	F	H	R
1	$\{1\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
2	$\{4\}$	$\{\}$	$\{\}$	$\{\}$
3	$\{5\}$	$\{\}$	$\{\}$	$\{\}$
4	$\{f\}$	$\{\}$	$\{\}$	$\{\}$
5	$\{\}$	$\{\}$	$\{f\}$	$\{\}$
f	$\{f\}$	$\{f\}$	$\{f\}$	$\{f\}$

Einen deterministischen Automaten erhalten wir, indem wir den indeterminischen Automaten simulieren. Ein Zustand des deterministischen Automaten repräsentiert dabei jene Zustände des indeterministischen, in denen sich dieser zu diesem Zeitpunkt befinden kann. Der Startzustand wird mit $\{1\}$ bezeichnet, da sich der indeterministische Automat zu Beginn im Zustand 1 (und nur in diesem) befindet. Von diesem Zustand ausgehend erstellen wir zeilenweise die Tabelle für die Übergangsfunktion des deterministischen Automaten.

$\hat{\delta}$	E	F	H	R
$\{1\}$	$\{1\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,2\}$	$\{1,4\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,3\}$	$\{1,5\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,4\}$	$\{1,f\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,5\}$	$\{1\}$	$\{1,2\}$	$\{1,f\}$	$\{1,3\}$
$\{1,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$
$\{1,2,f\}$	$\{1,4,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$
$\{1,3,f\}$	$\{1,5,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$
$\{1,4,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$
$\{1,5,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$

Jene Zustände, die einer Situation entsprechen, in der der indeterministische Automat einen Endzustand erreicht hat, sind die Endzustände des deterministischen Automaten; in diesem Beispiel sind das alle Zustände, deren Bezeichnung f enthält. Dieser wird somit durch das Tupel $\langle \widehat{Q}, \Sigma, \hat{\delta}, \{1\}, \widehat{F} \rangle$ beschrieben, wobei

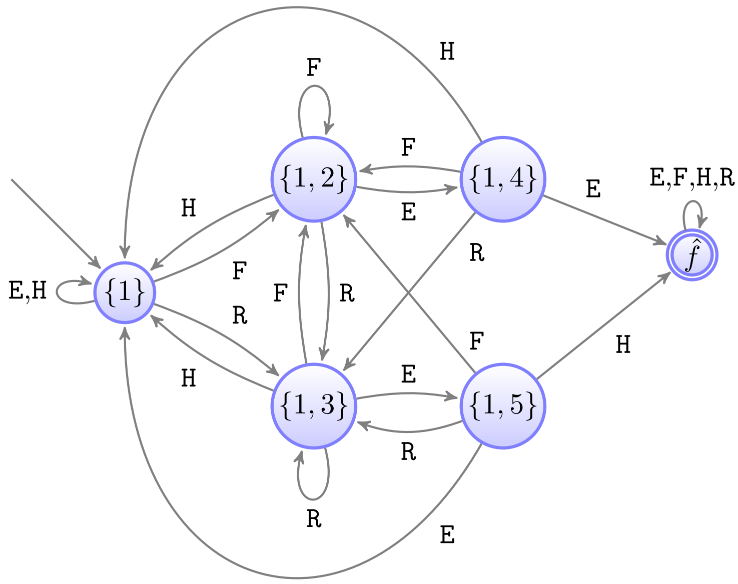
$$\Sigma = \{\mathtt{E}, \mathtt{F}, \mathtt{H}, \mathtt{R}\}$$
$$\widehat{F} = \{\{1,f\}, \{1,2,f\}, \{1,3,f\}, \{1,4,f\}, \{1,5,f\}\}$$
$$\widehat{Q} = \{\{1\}, \{1,2\}, \{1,3\}, \{1,4\}, \{1,5\}\} \cup \widehat{F}$$

Auch ohne Kenntnis des Minimierungsverfahrens für deterministische Automaten lässt sich erkennen, dass die Endzustände zu einem einzigen zusammengefasst werden können: Einmal in einem solchen angekommen, gelangt man mit jedem weiteren

Symbol wieder in einen Endzustand. Die so vereinfachte Übergangsfunktion sieht folgendermaßen aus:

$\hat{\delta}$	E	F	H	R
$\{1\}$	$\{1\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,2\}$	$\{1,4\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,3\}$	$\{1,5\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,4\}$	$\hat{f}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$
$\{1,5\}$	$\{1\}$	$\{1,2\}$	$\hat{f}$	$\{1,3\}$
$\hat{f}$	$\hat{f}$	$\hat{f}$	$\hat{f}$	$\hat{f}$

Graphische Darstellung dieses vereinfachten deterministischen Automaten:



Aufgabe 2 (2 Punkte)

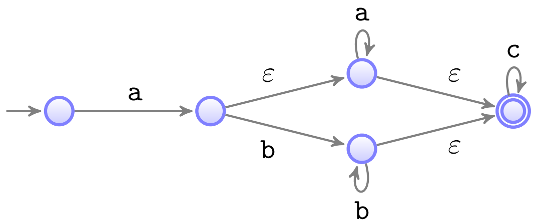
Geben Sie endliche Automaten an, die dieselben Sprachen beschreiben wie die folgenden regulären Ausdrücke in algebraischer Notation.

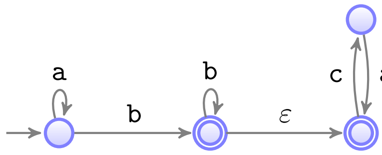
(a) $\mathtt{a}(\mathtt{a}^* + \mathtt{bb}^*)\mathtt{c}^*$

(b) $\mathtt{a}^*\mathtt{b}^*\mathtt{b}(\mathtt{ca})^*$

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. Die verbleibenden ε-Übergänge sind schwerer zu eliminieren, da sie sicher stellen, dass die Teile des Automaten in der richtigen Reihenfolge durchlaufen werden.

(a) 

(b) 

Aufgabe 3 (5 Punkte)

Vereinfachen Sie die folgenden Ausdrücke.

(a) $\{02, 1, 010\} \cdot \{\}$

(b) $\{04, 12, 01\} \cup \{\}^*$

(c) $\{0\}^* \cdot (\{0\} \cup \{\varepsilon\})$

(d) $(\{1\}^* \cdot \{1\}^+) \cdot \{1\}^*$

(e) $((\{0, 01\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, 11, 22\} \cup \{1\})) \cup \{0, 2\}$

Lösung

(a) $\{02, 1, 010\} \cdot \{\} = \{\}$

(b) $\{04, 12, 01\} \cup \{\}^* = \{04, 12, 01\} \cup \{\varepsilon\} = \{04, 12, 01, \varepsilon\}$

(c) $\{0\}^* \cdot (\{0\} \cup \{\varepsilon\}) = \{0\}^* \cdot \{0\} \cup \{0\}^* \cdot \{\varepsilon\} = \{0\}^+ \cup \{0\}^* = \{0\}^*$

(d) $(\{1\}^* \cdot \{1\}^+) \cdot \{1\}^* = \{1\}^+ \cdot \{1\}^* = \{1\}^+$

(e) $((\{0, 01\} \cup \{\varepsilon\}) \cdot (\{\varepsilon, 11, 22\} \cup \{1\})) \cup \{0, 2\}$
$= \{0, 01, \varepsilon\} \cdot \{\varepsilon, 11, 22, 1\} \cup \{0, 2\}$
$= \{\varepsilon, 0, 1, 2, 01, 11, 22, 011, 022, 0111, 0122\}$

Aufgabe 4 (3 Punkte)

Angenommen es sind zwei deterministische endliche Automaten $\mathcal{A}$ und $\mathcal{B}$ gegeben. Geben Sie ein allgemeines Verfahren an, um einen endlichen Automaten zu konstruieren, der genau jene Wörter akzeptiert, die von $\mathcal{A}$ aber nicht von $\mathcal{B}$ akzeptiert werden. Welche Eigenschaft regulärer Sprachen lässt sich daraus ablesen?

Lösung

Seien $\mathcal{A} = \langle Q_1, \Sigma, \delta_1, i_1, F_1\rangle$ und $\mathcal{B} = \langle Q_2, \Sigma, \delta_2, i_2, F_2\rangle$ zwei beliebige deterministische Automaten über dem Alphabet Σ. Wir konstruieren einen Automaten $\mathcal{C}$, der die beiden Automaten $\mathcal{A}$ und $\mathcal{B}$ gleichzeitig ausführt. Als Zustände für $\mathcal{C}$ 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 man mit diesem Symbol in $\mathcal{A}$ von q_1 nach $q_1' = \delta_1(q_1, s)$ und in $\mathcal{B}$ von q_2 nach $q_2' = \delta_2(q_2, s)$ gelangt.
Uns interessieren nun alle Wörter, die von $\mathcal{A}$ aber nicht von $\mathcal{B}$ akzeptiert werden. Ein derartiges Wort liegt genau dann vor, wenn der neue Automat damit einen Zustand (q_1, q_2) erreicht, bei dem q_1 ein Endzustand des Automaten $\mathcal{A}$ aber q_2 kein Endzustand des Automaten $\mathcal{B}$ ist. Der gesucht Automat ist somit gegeben durch

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

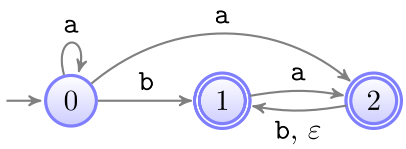
wobei die Übergangsfunktion für alle $q_1 \in Q_1$, $q_2 \in Q_2$ und $s \in \Sigma$ festgelegt ist durch

$$\delta((q_1, q_2), s) = (\delta_1(q_1, s), \delta_2(q_2, s))\ .$$

Die Menge der Wörter, die von $\mathcal{A}$ aber nicht von $\mathcal{B}$ akzeptiert werden, ist nichts anderes als die Mengendifferenz $\mathcal{L}(\mathcal{A}) \setminus \mathcal{L}(\mathcal{B})$. Unsere Konstruktion zeigt, dass die Differenz 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 Mengendifferenz. In der Vorlesung haben wir gezeigt, dass diese Sprachfamilie genau den regulären Sprachen entspricht. Daher sind auch die regulären Sprachen abgeschlossen gegenüber Differenzbildung: Die Differenz zweier regulärer Sprachen ist wieder regulär.

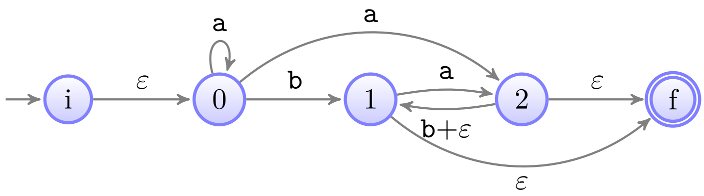
Aufgabe 5 **(3 Punkte)**

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



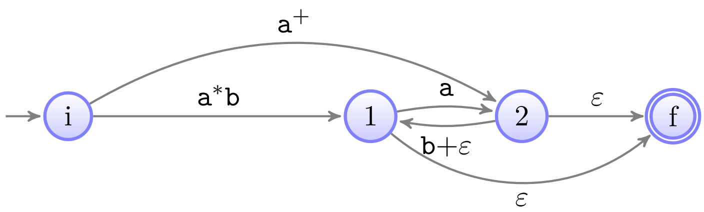
Lösung

Neuer Anfangs- und Endzustand:

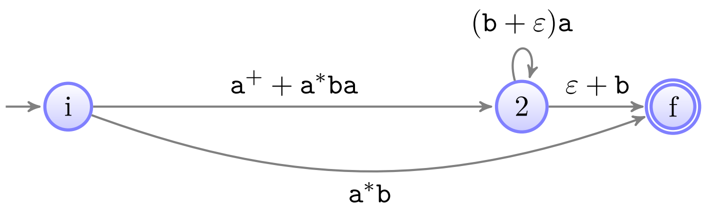


Bei der Elimination von Zuständen bevorzugen wir Zustände mit wenigen Übergängen bzw. solche ohne Schleifen, um die Anzahl der neu entstehenden Ausdrücke niedrig und die Ausdrücke selber so einfach wie möglich zu halten. Wir entscheiden uns für die Reihenfolge 0, 1 und 2. Jede andere Reihenfolge ist ebenfalls möglich und liefert einen äquivalenten Ausdruck.

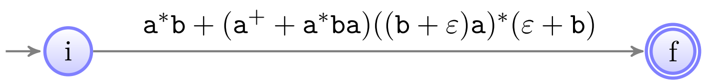
Elimination von Zustand 0:



Elimination von Zustand 1:



Elimination von Zustand 2:



Die Sprache des ursprünglichen Automaten wird also durch den Ausdruck

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

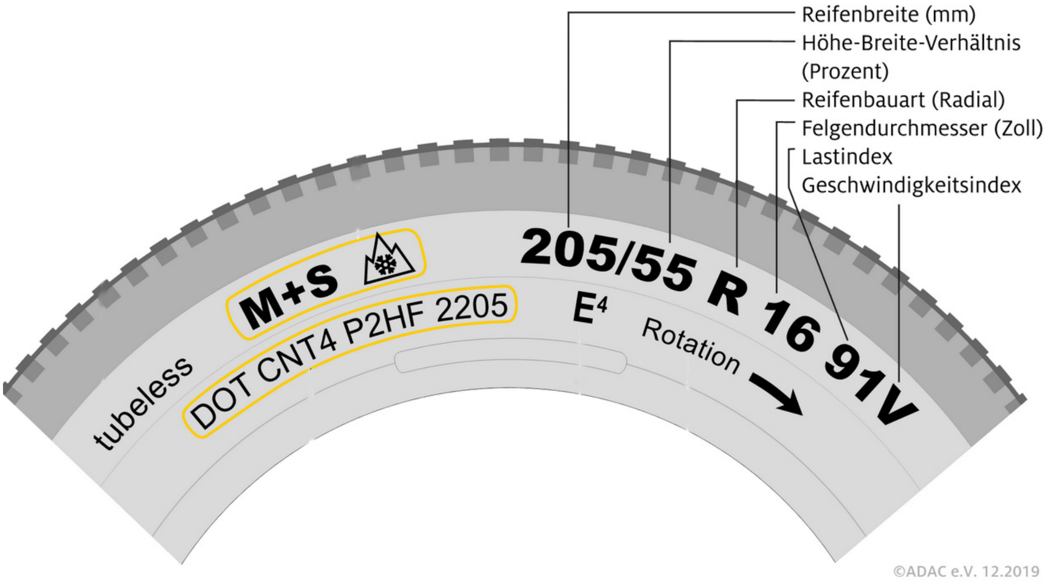
beschrieben.

Aufgabe 6 **(3 Punkte)**

Die EU-Norm ECE-R-30 legt fest, wie PKW-Reifen beschriftet sein müssen, damit ihre Dimensionen und Eigenschaften erkennbar sind. Auf der Reifenflanke müssen unter anderem folgende Werte angegeben sein, wobei die einzelnen Elemente jeweils durch ein Leerzeichen getrennt sind:

- Reifenbreite in mm (dreistellig), gefolgt von einem Schrägstrich und einer zweistelligen Zahl, die das Verhältnis von Reifenhöhe zu Reifenbreite in Prozent angibt;
- Reifenbauart: R (Radial) oder D (Diagonalreifen); anschließend folgt der Buchstabe F, falls es sich um einen Notlaufreifen handelt.
- Felgendurchmesser in Zoll: ein Wert zwischen 10 und 20
- Lastindex (eine zwei- oder dreistellige Zahl), gefolgt vom Geschwindigkeitsindex (ein oder zwei Großbuchstaben).

Die nachfolgende Abbildung zeigt eine solche Reifenbeschriftung. Der für diese Aufgabe relevante Bereich ist der rechte Teil der oberen Zeile.



[www.adac.de/rund-ums-fahrzeug/ausstattung-technik-zubehoer/reifen/reifenkauf/reifenkennzeichnung]

Beschreiben Sie den Aufbau derartiger Reifenbeschriftungen 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* eine derartige Reifenbeschriftung enthalten.

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

Lösung

(a) Eine derartige Reifenbeschriftung kann durch den Ausdruck

$$\mathit{Breite}\,\texttt{"/"}\,\mathit{HBV}\,\texttt{"\textvisiblespace"}\,\mathit{Bauart}\,\texttt{"\textvisiblespace"}\,\mathit{DM}\,\texttt{"\textvisiblespace"}\,\mathit{LI}\;\mathit{SI}$$

beschrieben werden, wobei wir folgende Abkürzungen verwenden:

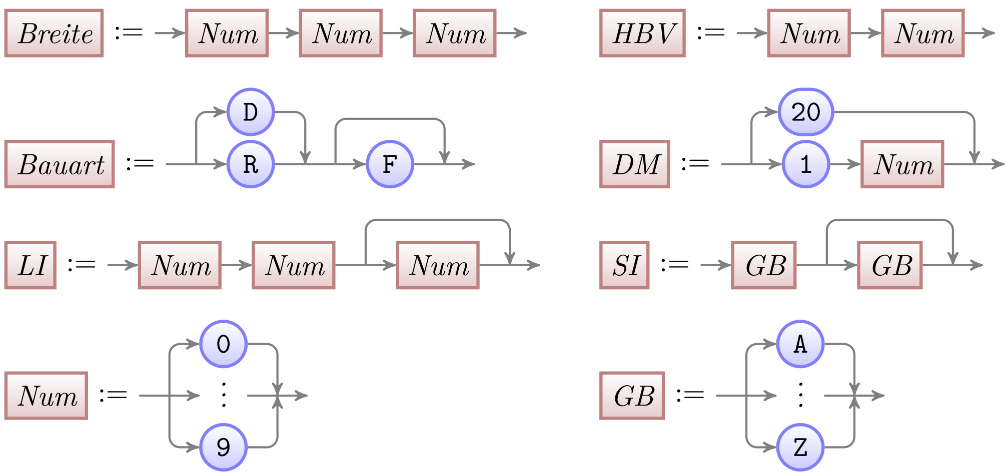
$$\begin{aligned}
\mathit{Breite} &:= \mathit{Num}\;\mathit{Num}\;\mathit{Num}\\
\mathit{HBV} &:= \mathit{Num}\;\mathit{Num}\\
\mathit{Bauart} &:= (\texttt{"D"} + \texttt{"R"})(\texttt{"F"} + \varepsilon)\\
\mathit{DM} &:= \texttt{"1"}\mathit{Num} + \texttt{"20"}\\
\mathit{LI} &:= \mathit{Num}\;\mathit{Num}(\mathit{Num} + \varepsilon)\\
\mathit{SI} &:= \mathit{GB}\,(\mathit{GB} + \varepsilon)\\
\mathit{Num} &:= \texttt{"0"} + \cdots + \texttt{"9"}\\
\mathit{GB} &:= \texttt{"A"} + \cdots + \texttt{"Z"}
\end{aligned}$$

Dabei setzen wir Symbole des Alphabets unter Anführungszeichen, um sie von algebraischen Symbolen (Metanotation) zu unterscheiden.

(b) `^[0-9]{3}/[0-9]{2}␣[DR]F?␣(1[0-9]|20)␣[0-9]{2,3}[A-Z]{1,2}$`

(c) Syntaxdiagramm:

Die Unterdiagramme sind folgendermaßen definiert.



Aufgabe 7 (7 Punkte)

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

$$\begin{aligned} N &= \{A, B, C\} \\ T &= \{\mathtt{a}, \mathtt{i}, \mathtt{l}, \mathtt{r}, \mathtt{t}\} \\ P &= \{\, A \to \mathtt{tra}\, B \mid \mathtt{tra}\ , \\ &\qquad B \to \mathtt{la}\, B \mid \mathtt{l}\, C \mid\ A \mid \varepsilon\ , \\ &\qquad C \to \mathtt{tra}\, C \mid \mathtt{l}\, B \mid \mathtt{i}\ \} \end{aligned}$$

(a) Überprüfen Sie für die nachfolgenden Wörter, ob sie in der von der Grammatik G spezifizierten Sprache $\mathcal{L}(G)$ liegen. Falls ja, geben Sie eine Parallelableitung an. Falls nein, argumentieren Sie, warum nicht.

(1) `traltralla`

(2) `traltra`

(3) `tralalalalaltralla`

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

(1) Jedes Wort beginnt mit `tra`.

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

(3) Nach jedem `la` muss ein `l` folgen.

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

$$\begin{aligned} A &\Rightarrow \mathtt{tra}\, B \qquad \Rightarrow \mathtt{tra\,l}\, C \qquad \Rightarrow \mathtt{tra\,l\,tra}\, C \\ &\Rightarrow \mathtt{tra\,l\,tra\,l}\, B \Rightarrow \mathtt{tra\,l\,tra\,l\,la}\, B \Rightarrow \mathtt{tra\,l\,tra\,l\,la}\,\varepsilon = \mathtt{traltralla} \end{aligned}$$

(2) Die einzige Ableitung, die die Zeichenfolge `traltra` am Wortbeginn erzeugt, ist die folgende:

$$A \Rightarrow \mathtt{tra}\, B \Rightarrow \mathtt{tra\,l}\, C \Rightarrow \mathtt{tra\,l\,tra}\, C$$

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

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

$$\begin{aligned} A &\Rightarrow \texttt{tra}\,B \Rightarrow \texttt{tra}\,\texttt{la}\,B \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,B \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,B \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,B \\ &\Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{l}\,C \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{l}\,\texttt{tra}\,C \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{l}\,\texttt{tra}\,\texttt{l}\,B \\ &\Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{l}\,\texttt{tra}\,\texttt{l}\,\texttt{la}\,B \Rightarrow \texttt{tra}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{la}\,\texttt{l}\,\texttt{tra}\,\texttt{l}\,\texttt{la}\,\varepsilon \\ &= \texttt{tralalalalaltralla} \end{aligned}$$

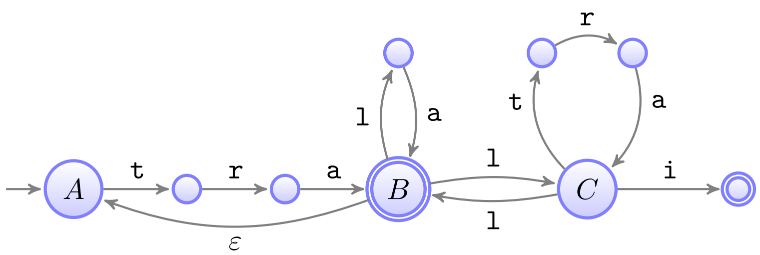
(b) (1) Richtig. Beide Produktionen für das Startsymbol A erzeugen ein Wort, das mit `tra` beginnt.

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

$$A \Rightarrow \texttt{tra}\,B \Rightarrow \texttt{trala}\,B \Rightarrow \cdots \Rightarrow \texttt{tra}\,(\texttt{la})^n\,B \Rightarrow \texttt{tra}\,(\texttt{la})^n$$

(3) Falsch. Ein Gegenbeispiel ist das Wort `traltralla`, das in der Sprache liegt (siehe Teilaufgabe a1 oben), auf die Zeichenfolge `la` am Wortende folgt aber offenbar kein `l`.

(c) Ja, die Sprache $\mathcal{L}(G)$ kann auch durch einen endlichen Automaten beschrieben werden. Da auf der rechten Seite der Produktionen immer nur ein einziges Nonterminal auftritt, das noch dazu ganz am Ende steht, lässt sich die Grammatik systematisch in einen endlichen Automaten umwandeln, indem jedes Nonterminal zu einem Zustand wird. 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 Zeichen zerlegen.



Aufgabe 8 (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 Beispielprogramm 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).
```

(a) Beschreiben Sie die zulässigen Datalog-Programme mittels einer kontextfreien Grammatik. Verwenden Sie Ebnf-Notationen, um die Grammatik übersichtlich zu strukturieren und die Verwendung rekursiver Regeln zu minimieren.

(b) Handelt es sich bei der Menge der Datalog-Programme um eine reguläre Sprache, d.h., lässt sich diese Sprache im Prinzip auch durch einen (komplizierten) regulären Ausdruck spezifizieren? Falls ja, skizzieren Sie den Ausdruck unter Verwendung einer der besprochenen Notationen. Falls nein, begründen Sie Ihre Antwort.

Lösung

(a) $\langle N, T, P, Programm\rangle$, wobei

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

(b) Ja, die Menge der DATALOG-Programme ist eine reguläre Sprache. Um einen regulären Ausdruck dafür zu erhalten, können wir die Produktionen der kontextfreien Grammatik oben als reguläre Ausdrücke auffassen, die durch die Verwendung von Abkürzungen strukturiert wurden.

$$
\begin{aligned}
Programm &:= \{Klausel\}\\
Klausel &:= Faktum \mid Regel\\
Faktum &:= Atom\texttt{"."}\\
Regel &:= Atom\texttt{":-"}Atom\{\texttt{","}Atom\}\texttt{"."}\\
Atom &:= Name[\texttt{"("}Argliste\texttt{")"}]\\
Argliste &:= Arg\{\texttt{","}Arg\}\\
Arg &:= Name \mid Var\\
Name &:= KB\ Zeichenkette,\\
Var &:= GB\ Zeichenkette,\\
Zeichenkette &:= \{KB \mid GB \mid Num\},\\
KB &:= \texttt{"a"} \mid \cdots \mid \texttt{"z"}\\
GB &:= \texttt{"A"} \mid \cdots \mid \texttt{"Z"}\\
Num &:= \texttt{"0"} \mid \cdots \mid \texttt{"9"}
\end{aligned}
$$

Der mit *Programm* abgekürzte reguläre Ausdruck stellt die gesuchte Sprache dar. Wichtig dabei ist zu kontrollieren, dass es sich tatsächlich um einen regulären Ausdruck und nicht um eine zirkuläre Definition handelt. Es lässt sich aber leicht feststellen, dass jeder Ausdruck nur Abkürzungen verwendet, die erst unter dem Ausdruck definiert werden. Dadurch wäre es möglich, auf Kosten der Lesbarkeit alle Abkürzungen zu eliminieren, indem sie durch den zugehörigen regulären Ausdruck ersetzt werden (wobei dieser im Allgemeinen geklammert werden muss, um den Geltungsbereich der Vereinigung zu begrenzen). Das Ergebnis wäre der folgende reguläre Ausdruck in EBNF-Notation.

```
{("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}["("(("a" | ··· |
"z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"} | ("A" | ··· | "Z"){"a" |
··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}){","("a" | ··· | "z"){"a" | ··· | "z" |
"A" | ··· | "Z" | "0" | ··· | "9"} | ("A" | ··· | "Z"){"a" | ··· | "z" | "A" | ··· | "Z" |
"0" | ··· | "9"}}")")"]"." | ("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" |
··· | "9"}["("(("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"} |
("A" | ··· | "Z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}){","("a" | ··· |
"z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"} | ("A" | ··· | "Z"){"a" | ··· |
"z" | "A" | ··· | "Z" | "0" | ··· | "9"}}")")"]":-"("a" | ··· | "z"){"a" | ··· | "z" |
"A" | ··· | "Z" | "0" | ··· | "9"}["("(("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· |
"Z" | "0" | ··· | "9"} | ("A" | ··· | "Z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· |
"9"}){","("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"} | ("A" |
··· | "Z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}}")")"]{","("a" | ··· |
"z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}["("(("a" | ··· | "z"){"a" | ··· |
"z" | "A" | ··· | "Z" | "0" | ··· | "9"} | ("A" | ··· | "Z"){"a" | ··· | "z" | "A" | ··· |
"Z" | "0" | ··· | "9"}){","("a" | ··· | "z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· |
"9"} | ("A" | ··· | "Z"){"a" | ··· | "z" | "A" | ··· | "Z" | "0" | ··· | "9"}}")")"]}"."}
```

Oder kompakter, aber ebenso unlesbar in POSIX-Notation:

```
(([a-z][0-9a-zA-Z]*(\(([a-z][0-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*)(,([a-z][0
-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*))*\))?\.)|([a-z][0-9a-zA-Z]*(\(([a-z][0-
9a-zA-Z]*|[A-Z][0-9a-zA-Z]*)(,([a-z][0-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*))*
\))?:-[a-z][0-9a-zA-Z]*(\(([a-z][0-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*)(,([a-
z][0-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*))*\))?(,[a-z][0-9a-zA-Z]*(\(([a-z][0
-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*)(,([a-z][0-9a-zA-Z]*|[A-Z][0-9a-zA-Z]*))
*\))?)*\.))*
```

Aufgabe 9 (4 Punkte)

In Büchern über formale Sprachen ist folgende Definition zu finden:

> Eine *reguläre Grammatik* wird durch ein 4-Tupel $G = \langle V, T, P, S\rangle$ festgelegt, wobei
>
> - V und T endliche, disjunkte Mengen von Symbolen sind ($V \cap T = \{\}$),
> - S ein Symbol aus V ist ($S \in V$) und
> - $P \subseteq V \times (T \cdot V \cup \{\varepsilon\})$ eine endliche Menge von Paaren ist.

Die Elemente von P werden Produktionen genannt; statt $(x, y) \in P$ wird auch $x \to y$ geschrieben. Die Notation $x \to y_1 \mid \cdots \mid y_n$ ist eine Abkürzung für die Produktionen $x \to y_1, \ldots, x \to y_n$.

Das Wort $u\,y\,v$ ist aus dem Wort $u\,x\,v$ in einem Schritt ableitbar, geschrieben $u\,x\,v \Rightarrow u\,y\,v$, wenn $x \to y$ gilt. Die von G generierte Sprache $\mathcal{L}(G)$ ist definiert als die Menge $\{\, w \in T^* \mid S \overset{*}{\Rightarrow} w \,\}$, wobei $\overset{*}{\Rightarrow}$ den reflexiven und transitiven Abschluss von $\Rightarrow$ bezeichnet.[1]

Geben Sie an, welche der folgenden Tupeln eine reguläre Grammatik gemäß der obigen Definition darstellt. Begründen Sie Ihre Antwort, falls es sich um keine reguläre Grammatik handelt. Entspricht das Tupel der Definition, geben Sie die Sprache an, die durch die Grammatik generiert wird.

(a) $\langle \{\mathtt{X}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{aXb} \mid \varepsilon\}, \mathtt{X} \rangle$

(b) $\langle \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{aX} \mid \mathtt{bY} \mid \varepsilon,\ \mathtt{Y} \to \mathtt{bY}\}, \mathtt{X} \rangle$

(c) $\langle \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a} \to \mathtt{Xa} \mid \mathtt{Yb} \mid \varepsilon,\ \mathtt{b} \to \mathtt{Yb}\}, \mathtt{b} \rangle$

(d) $\langle \{\mathtt{X}, \mathtt{Y}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{X} \to \mathtt{Xa} \mid \mathtt{Yb} \mid \varepsilon,\ \mathtt{Y} \to \mathtt{Yb} \mid \varepsilon\}, \mathtt{X} \rangle$

Lösung

(a) Ist keine reguläre Grammatik, da die rechte Seite der ersten Produktion, `aXb`, weder in der Menge $T \cdot V$ liegt noch das Leerwort ist, da sie mit dem Terminalsymbol `b` endet.

(b) Ist eine reguläre Grammatik, die die Sprache $\{\mathtt{a}\}^*$ generiert.

(c) Ist eine reguläre Grammatik, die die Sprache $\{\}$ generiert.

(d) Ist keine reguläre Grammatik, da die rechte Seite der ersten Produktion, `Xa`, weder in der Menge $T \cdot V$ liegt noch das Leerwort ist, da sie mit dem Nonterminal `X` beginnt.

Aufgabe 10 (4 Punkte)

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

(a) Hunde, die bellen, beißen nicht.

---

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

- Aus $u \Rightarrow v$ folgt $u \overset{*}{\Rightarrow} v$.
- Es gilt $u \overset{*}{\Rightarrow} u$ für alle Wörter $u \in T^*$.
- Aus $u \overset{*}{\Rightarrow} v$ und $v \overset{*}{\Rightarrow} w$ folgt $u \overset{*}{\Rightarrow} w$.

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

(b) Es ist nicht alles Gold, was glänzt.

(c) Kein Mensch ist unsterblich.

(d) Lügen haben kurze Beine.

Lösung

Seien *Hund*/1, *Glänzt*/1, *Gold*/1, *Lüge*/1, *Mensch*/1, *Bellt*/1, *Beißt*/1, *Unsterblich*/1 und *HatKurzeBeine*/1 Prädikatensymbole mit folgender Bedeutung:

$$\begin{array}{rlrl}
\mathit{Hund}(x) \ldots & x \text{ ist ein Hund} & \mathit{Mensch}(x) \ldots & x \text{ ist ein Mensch} \\
\mathit{Bellt}(x) \ldots & x \text{ bellt} & \mathit{Unsterblich}(x) \ldots & x \text{ ist unsterblich} \\
\mathit{Beißt}(x) \ldots & x \text{ beißt} & \mathit{Lüge}(x) \ldots & x \text{ ist eine Lüge} \\
\mathit{Gold}(x) \ldots & x \text{ ist Gold} & \mathit{HatKurzeBeine}(x) \ldots & x \text{ hat kurze Beine} \\
\mathit{Glänzt}(x) \ldots & x \text{ glänzt} & &
\end{array}$$

(a) $\forall x((\mathit{Hund}(x) \land \mathit{Bellt}(x)) \supset \neg \mathit{Beißt}(x))$

(b) $\neg\forall x(\mathit{Glänzt}(x) \supset \mathit{Gold}(x))$

(c) $\neg\exists x(\mathit{Mensch}(x) \land \mathit{Unsterblich}(x))$

(d) $\forall x(\mathit{Lüge}(x) \supset \mathit{HatKurzeBeine}(x))$

Aufgabe 11 (5 Punkte)

Seien *Frisst*/2, *Vogel*/1, *Futter*/1 und *Groß*/1 Prädikatensymbole sowie *samen* und *beeren* Konstantensymbole mit folgender Bedeutung:

$$\begin{array}{rlrl}
\mathit{Vogel}(x) \ldots & x \text{ ist ein Vogel} & \mathit{Frisst}(x, y) \ldots & x \text{ frisst } y \\
\mathit{Futter}(x) \ldots & x \text{ ist Futter} & \mathit{samen} \ldots & \text{Samen} \\
\mathit{Groß}(x) \ldots & x \text{ ist groß} & \mathit{beeren} \ldots & \text{Beeren}
\end{array}$$

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

(a) Manche Vögel fressen Samen, aber keine Beeren.

(b) Kein Futter wird von allen großen Vögeln gefressen.

Sei weiters folgende Interpretation I gegeben:

$$\begin{aligned}
\mathcal{U} &= \{\text{Kakadu}, \text{Kiwi}, \text{Ara}, \text{Specht}, \text{Spatz}, \text{Nüsse}, \text{Körner}, \text{Samen}, \text{Würmer}, \text{Beeren}, \\
&\quad \text{Früchte}\} \\
I(\mathit{Vogel}) &= \{\text{Kakadu}, \text{Kiwi}, \text{Ara}, \text{Specht}\} \\
I(\mathit{Futter}) &= \{\text{Körner}, \text{Samen}, \text{Würmer}, \text{Beeren}, \text{Früchte}\} \\
I(\mathit{Groß}) &= \{\text{Kakadu}, \text{Specht}, \text{Ara}, \text{Nüsse}\} \\
I(\mathit{Frisst}) &= \{(\text{Kakadu}, \text{Beeren}), (\text{Kakadu}, \text{Samen}), \\
&\quad (\text{Kiwi}, \text{Früchte}), (\text{Kiwi}, \text{Beeren}), \\
&\quad (\text{Ara}, \text{Würmer}), (\text{Ara}, \text{Samen}), \\
&\quad (\text{Specht}, \text{Körner}), (\text{Specht}, \text{Samen}), (\text{Specht}, \text{Würmer})\} \\
I(\mathit{beeren}) &= \text{Beeren} \\
I(\mathit{samen}) &= \text{Samen}
\end{aligned}$$

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) $\forall x\, ((\mathit{Groß}(x) \land \mathit{Futter}(x)) \supset \mathit{Frisst}(x, \mathit{samen}))$

(d) $\forall x \exists y\, (\mathit{Vogel}(x) \supset (\mathit{Futter}(y) \land \mathit{Frisst}(x, y))$

(e) $\forall x \exists y\, (\mathit{Futter}(x) \supset (\mathit{Vogel}(y) \land \mathit{Groß}(y) \land \mathit{Frisst}(y, x)))$

Lösung

(a) $\exists x\, (\mathit{Vogel}(x) \land \mathit{Frisst}(x, \mathit{samen}) \land \neg \mathit{Frisst}(x, \mathit{beeren}))$

„Manche" bedeutet eigentlich auch, dass nicht alle betroffen sind. Eine präzisere Lösung wäre daher:

$\exists x\, (\mathit{Vogel}(x) \land \mathit{Frisst}(x, \mathit{samen}) \land \neg \mathit{Frisst}(x, \mathit{beeren}))$
$\land \neg \forall x\, (\mathit{Vogel}(x) \supset (\mathit{Frisst}(x, \mathit{samen}) \land \neg \mathit{Frisst}(x, \mathit{beeren})))$

(b) $\neg \exists x\, (\mathit{Futter}(x) \land \forall y\, ((\mathit{Vogel}(y) \land \mathit{Groß}(y)) \supset \mathit{Frisst}(y, x)))$ oder
$\neg \exists x\, \forall y\, (\mathit{Futter}(x) \land ((\mathit{Vogel}(y) \land \mathit{Groß}(y)) \supset \mathit{Frisst}(y, x)))$ oder
$\forall x\, (\mathit{Futter}(x) \supset \exists y\, (\mathit{Vogel}(y) \land \mathit{Groß}(y) \land \neg \mathit{Frisst}(y, x)))$ oder
$\forall x\, \exists y\, (\mathit{Futter}(x) \supset (\mathit{Vogel}(y) \land \mathit{Groß}(y) \land \neg \mathit{Frisst}(y, x)))$

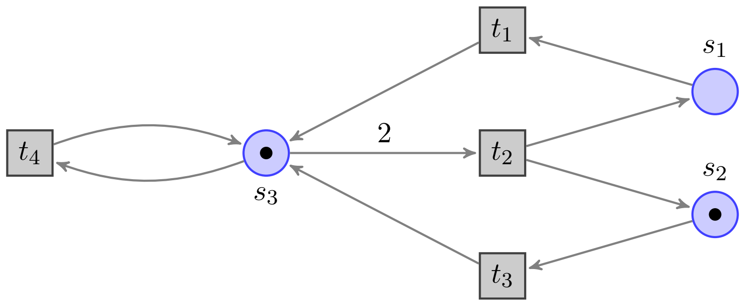
(c) Übersetzung: Jedes große Futter frisst Samen.
Diese Aussage ist wahr in I, da es kein großes Futter gibt, weswegen die Implikation immer wahr liefert.

(d) Übersetzung: Alle Vögel fressen (irgend)ein Futter.
Diese Aussage ist wahr in I. Kakadu, Kiwi, Ara und Specht fressen jeweils etwas aus $I(\mathit{Futter})$.

(e) Übersetzung: Jedes Futter wird von einem großen Vogel gefressen.
Diese Aussage ist falsch in I. Früchte frisst nur der Kiwi, dieser ist zwar ein Vogel, aber er ist nicht groß.

Aufgabe 12 (3 Punkte)

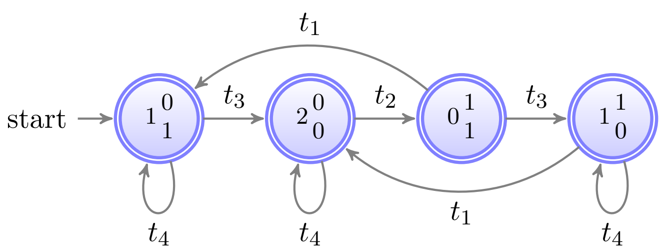
Gegeben sei das folgende Petri-Netz mit Anfangsmarkierung.



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 $t_3t_2t_1$ akzeptieren, weil die Transitionen in dieser Reihenfolge feuern können, nicht aber t_1t_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 drei Zahlen $n_3\,{}^{n_1}_{n_2}$ 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.

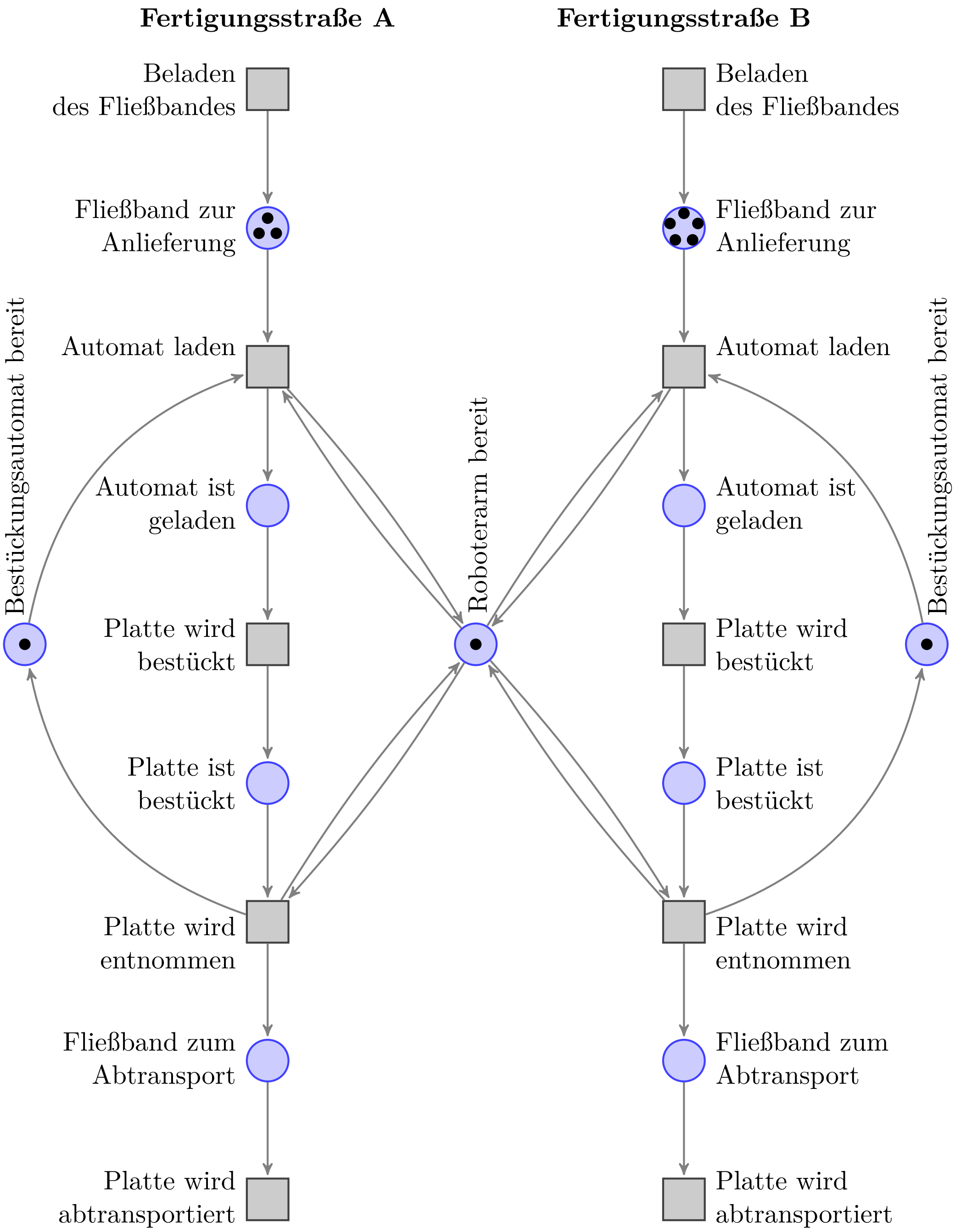


Aufgabe 13 (3 Punkte)

Leiterplatten sind Kunststoffplatten mit leitenden Bahnen (meist aus Kupfer), auf die elektronische Bauteile gelötet werden. Beispielsweise sind Motherboards von Computern solche mit diversen Bauteilen bestückte Leiterplatten.
In einer Leiterplattenfabrik werden zwei unterschiedliche Leiterplatten gefertigt, Typ A und Typ B. Die Leiterplatten vom Typ A werden von einem Fließband herangebracht, in einen Bestückungsautomaten gelegt, von diesem mit den elektronischen Bauteilen bestückt und danach wieder auf ein weiteres Fließband entladen, das die fertige Leiterplatte abtransportiert. Dasselbe passiert mit den Leiterplatten vom Typ B: Auch hier gibt es ein Fließband zum Anliefern der Platten, einen Bestückungsautomaten und ein Fließband zum Abtransport. Die Bestückungsautomaten können zu jedem Zeitpunkt immer nur eine Platte bearbeiten.
Für das Be- und Entladen der Bestückungsautomaten vom bzw. zum jeweiligen Fließband gibt es einen einzelnen Roboterarm, der diese vier Tätigkeiten – Automat für Typ A/Typ B beladen/entladen – durchführt. Beim Bestücken einer Leiterplatte durch den Automaten wird der Roboter nicht benötigt, er kann sich währenddessen einer anderen Tätigkeit widmen.[2]
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 anfänglich 3 Platten des Typs A und 5 Platten des Typs B auf Bestückung warten.

[2] Inspiriert von einer Aufgabe der Vorlesung „Automatisierungsprojekte“ der Hochschule Karlsruhe.

Lösung



Fertigungsstraße A
Fertigungsstraße B
Beladen des Fließbandes
Fließband zur Anlieferung
Automat laden
Automat ist geladen
Platte wird bestückt
Platte ist bestückt
Platte wird entnommen
Fließband zum Abtransport
Platte wird abtransportiert
Bestückungsautomat bereit
Roboterarm bereit
Beladen des Fließbandes
Fließband zur Anlieferung
Automat laden
Automat ist geladen
Platte wird bestückt
Platte ist bestückt
Platte wird entnommen
Fließband zum Abtransport
Platte wird abtransportiert
Bestückungsautomat bereit