2. Übungsblatt (mit Lösungen)

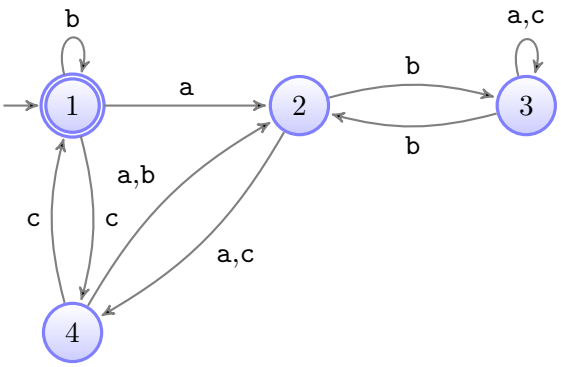
3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

8. Mai 2018

Aufgabe 1 (3 Punkte)

Sei $\mathcal{A}$ der folgende endliche Automat.



(a) Geben Sie alle Wörter an, die aus maximal drei Zeichen bestehen und von $\mathcal{A}$ akzeptiert werden. (Das sollten neun Wörter sein.)

(b) Berechnen Sie schrittweise $\delta^*(1, \texttt{baccb})$.

(c) Spezifizieren Sie $\mathcal{A}$ in tabellarischer Form. Handelt es sich bei $\mathcal{A}$ um einen deterministischen oder indeterministischen Automaten?

Lösung

(a) Die von $\mathcal{A}$ akzeptierten Wörter mit einer Länge von maximal drei Zeichen sind ε, `b`, `bb`, `cc`, `aac`, `acc`, `bbb`, `bcc` und `ccb`.

(b) $\delta^*(1, \texttt{baccb}) = \delta^*(\delta(1, \texttt{b}), \texttt{accb})$
$= \delta^*(1, \texttt{accb})$
$= \delta^*(\delta(1, \texttt{a}), \texttt{ccb})$
$= \delta^*(2, \texttt{ccb})$
$= \delta^*(\delta(2, \texttt{c}), \texttt{cb})$
$= \delta^*(4, \texttt{cb})$
$= \delta^*(\delta(4, \texttt{c}), \texttt{b})$
$= \delta^*(1, \texttt{b})$
$= \delta^*(\delta(1, \texttt{b}), \varepsilon)$
$= \delta^*(1, \varepsilon)$
$= 1$

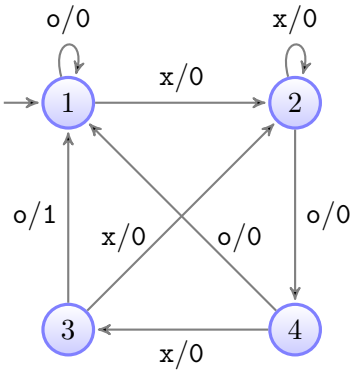
(c) $\mathcal{A} = \langle \{1, 2, 3, 4\}, \{\texttt{a}, \texttt{b}, \texttt{c}\}, \delta, 1, \{1\} \rangle$, wobei die Übergangsfunktion δ durch folgende Tabelle definiert ist:

δ	a	b	c
1	2	1	4
2	4	3	4
3	3	2	3
4	2	2	1

$\mathcal{A}$ ist ein deterministischer Automat, da der momentane Zustand und die nächste Eingabe immer eindeutig den Folgezustand bestimmen. Das äußert sich in der Tabelle dadurch, dass jeder Eintrag genau einen Zustand enthält.

Aufgabe 2 (4 Punkte)

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



(a) Geben Sie die Ausgabe zur Eingabe xxoxoox an.

(b) Berechnen Sie schrittweise $\delta^*(1, \texttt{oxoxox})$ und $\gamma^*(1, \texttt{oxoxox})$.

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

Lösung

(a) $\gamma^*(1, \texttt{xxoxoox}) = \texttt{0000100}$

(b)
$$\begin{aligned}
\delta^*(1, \texttt{oxoxox}) &= \delta^*(\delta(1, \texttt{o}), \texttt{xoxox}) \\
&= \delta^*(1, \texttt{xoxox}) \\
&= \delta^*(\delta(1, \texttt{x}), \texttt{oxox}) \\
&= \delta^*(2, \texttt{oxox}) \\
&= \delta^*(\delta(2, \texttt{o}), \texttt{xox}) \\
&= \delta^*(4, \texttt{xox}) \\
&= \delta^*(\delta(4, \texttt{x}), \texttt{ox}) \\
&= \delta^*(3, \texttt{ox}) \\
&= \delta^*(\delta(3, \texttt{o}), \texttt{x}) \\
&= \delta^*(1, \texttt{x}) \\
&= \delta^*(\delta(1, \texttt{x}), \varepsilon) \\
&= \delta^*(2, \varepsilon) \\
&= 2
\end{aligned}$$

$$\begin{aligned}
\gamma^*(1, \texttt{oxoxox}) &= \gamma(1, \texttt{o}) \cdot \gamma^*(\delta(1, \texttt{o}), \texttt{xoxox}) \\
&= \texttt{0} \cdot \gamma^*(1, \texttt{xoxox}) \\
&= \texttt{0} \cdot \gamma(1, \texttt{x}) \cdot \gamma^*(\delta(1, \texttt{x}), \texttt{oxox}) \\
&= \texttt{0} \cdot \texttt{0} \cdot \gamma^*(2, \texttt{oxox}) \\
&= \texttt{00} \cdot \gamma(2, \texttt{o}) \cdot \gamma^*(\delta(2, \texttt{o}), \texttt{xox}) \\
&= \texttt{00} \cdot \texttt{0} \cdot \gamma^*(4, \texttt{xox}) \\
&= \texttt{000} \cdot \gamma(4, \texttt{x}) \cdot \gamma^*(\delta(4, \texttt{x}), \texttt{ox}) \\
&= \texttt{000} \cdot \texttt{0} \cdot \gamma^*(3, \texttt{ox}) \\
&= \texttt{0000} \cdot \gamma(3, \texttt{o}) \cdot \gamma^*(\delta(3, \texttt{o}), \texttt{x}) \\
&= \texttt{0000} \cdot \texttt{1} \cdot \gamma^*(1, \texttt{x}) \\
&= \texttt{00001} \cdot \gamma(1, \texttt{x}) \cdot \gamma^*(\delta(1, \texttt{x}), \varepsilon) \\
&= \texttt{00001} \cdot \texttt{0} \cdot \gamma^*(2, \varepsilon) \\
&= \texttt{000010} \cdot \varepsilon = \texttt{000010}
\end{aligned}$$

(c) Der Mealy-Automat ist ein `xoxo`-Detektor: Immer wenn in der Eingabe die Symbolfolge `xoxo` auftritt, wird das Symbol `1` ausgegeben, sonst `0`.

Aufgabe 3 (4 Punkte)

Sei L die Sprache

$$\{\, w \in \{\texttt{A}, \texttt{N}, \texttt{S}, \texttt{U}, \texttt{X}\}^* \mid w \text{ enthält das Teilwort } \texttt{USA} \text{ oder } \texttt{NASA} \,\} \ .$$

Geben Sie einen *deterministischen* Automaten für L an. Gehen Sie folgendermaßen vor:

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

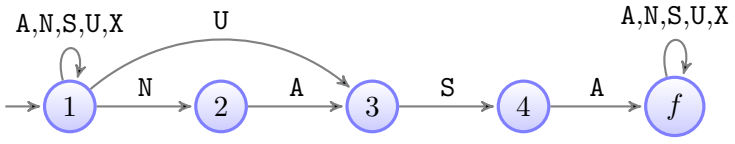
(b) Konstruieren Sie einen indeterministischen Automaten für diese Sprache.

(c) Wandeln Sie den indeterministischen Automaten mit Hilfe des in der Vorlesung besprochenen Verfahrens in einen deterministischen um.

Lösung

(a) `[ANSUX]*(USA|NASA)[ANSUX]*` oder `^[ANSUX]*(USA|NASA)[ANSUX]*$` (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 erweiterte Übergangsfunktion δ^* für jeden Zustand und jedes Eingabesymbol. Wenn es ε-Übergänge gibt, müssen auch längere Pfade betrachtet werden.)

δ^*	A	N	S	U	X
1	$\{1\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$	$\{1\}$
2	$\{3\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
3	$\{\}$	$\{\}$	$\{4\}$	$\{\}$	$\{\}$
4	$\{f\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
f	$\{f\}$	$\{f\}$	$\{f\}$	$\{f\}$	$\{f\}$

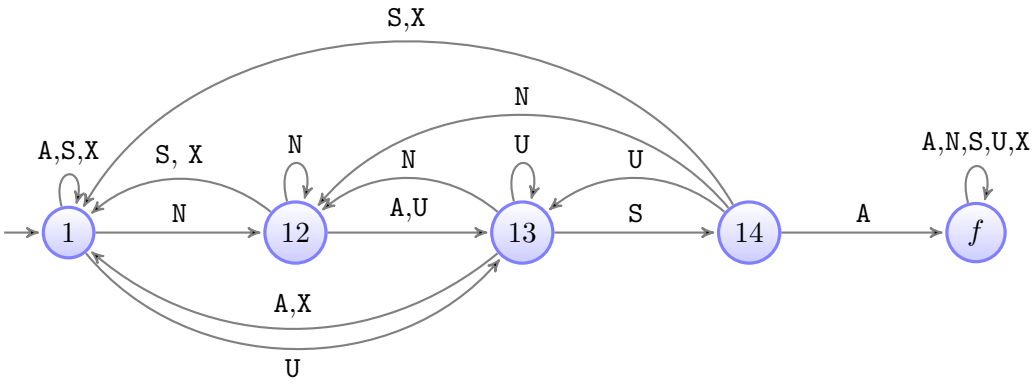
Einen deterministischen Automaten erhalten wir, indem wir den indeterminischen Automaten simulieren. Ein Zustand des deterministischen Automaten repräsentiert dabei jene Zustände des indeterministischen, in denen sich dieser zu diesem Zeitpunkt befinden kann. Der Startzustand wird mit $\{1\}$ bezeichnet, 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.

$\hat{\delta}$	A	N	S	U	X
$\{1\}$	$\{1\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$	$\{1\}$
$\{1,2\}$	$\{1,3\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$	$\{1\}$
$\{1,3\}$	$\{1\}$	$\{1,2\}$	$\{1,4\}$	$\{1,3\}$	$\{1\}$
$\{1,4\}$	$\{1,f\}$	$\{1,2\}$	$\{1\}$	$\{1,3\}$	$\{1\}$
$\{1,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$	$\{1,f\}$
$\{1,2,f\}$	$\{1,3,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$	$\{1,f\}$
$\{1,3,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,4,f\}$	$\{1,3,f\}$	$\{1,f\}$
$\{1,4,f\}$	$\{1,f\}$	$\{1,2,f\}$	$\{1,f\}$	$\{1,3,f\}$	$\{1,f\}$

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

$$\Sigma = \{\mathtt{A}, \mathtt{N}, \mathtt{S}, \mathtt{U}, \mathtt{X}\}$$
$$\widehat{F} = \{\{1,f\}, \{1,2,f\}, \{1,3,f\}, \{1,4,f\}\}$$
$$\widehat{Q} = \{\{1\}, \{1,2\}, \{1,3\}, \{1,4\}\} \cup \widehat{F}$$

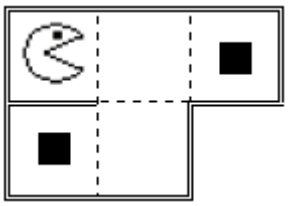
Auch ohne Kenntnis des Minimierungsalgorithmus für deterministische Automaten lässt sich erkennen, dass die Endzustände zu einem einzigen zusammengefasst werden können: Einmal in einem solchen angekommen, gelangt man mit jedem weiteren Symbol wieder in einen Endzustand. Der so vereinfachte Automat lässt sich graphisch wie folgt darstellen.



Aufgabe 4 (3 Punkte)

Bei Pac-Man, einem Videospiel aus den 80ern, muss die Spielfigur Pac-Man Punkte in einem Labyrinth fressen, während sie von Geistern verfolgt wird. Pac-Man und jeder Geist kann pro Zug ein Feld nach oben, unten, links oder rechts bewegt werden, vorausgesetzt, es ist keine Mauer im Weg. Bewegt Pac-Man sich auf ein Feld mit einem Punkt, so frisst er diesen Punkt. Sind alle Punkte gefressen, gilt das Level als gewonnen. Befinden sich Pac-Man und ein Geist auf dem gleichen Feld, so wird Pac-Man vom Geist gefressen und das Spiel ist beendet. Befinden sich der letzte Punkt, Pac-Man und ein Geist auf demselben Feld, hat der Geist Vorrang, d.h., Pac-Man verliert auch in diesem Fall.

Nehmen Sie an, dass das aktuelle Level, das zu bewältigen ist, so aussieht wie rechts skizziert. Zu Beginn befindet sich Pac-Man im linken oberen Feld. Die schwarzen Quadrate symbolisieren zwei Punkte, die Pac-Man fressen muss. Rundherum ist eine Mauer, die auch in das Spielfeld hineinragt (doppelte Linie). Pac-Man gewinnt das Level, wenn er die beiden Punkte gefressen hat.

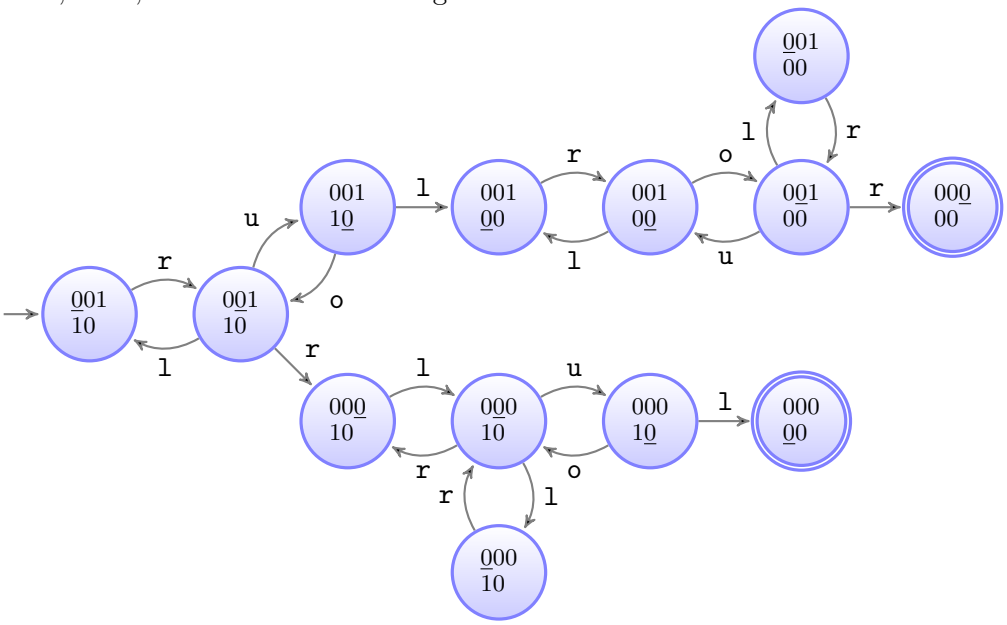


Anmerkung: Es handelt sich um ein Übungslevel. In diesem Level gibt es keinen Geist, Pac-Man kann also nicht verlieren.

- Überlegen Sie, welche Informationen notwendig sind, um den Zustand des Systems zu beschreiben.
- Legen Sie die möglichen Aktionen fest, die zu einem Zustandswechsel führen können.
- Geben Sie einen endlichen Automaten an, der das Systemverhalten vollständig beschreibt.

Lösung

Der Systemzustand ist hinreichend festgelegt, wenn bekannt ist, wo sich Pac-Man gerade befindet und wo sich Punkte befinden. Wir verwenden 0 bzw. 1 um anzugeben, ob sich auf dem jeweiligen Feld ein Punkt befindet und unterstreichen die Stelle, an der sich Pac-Man aufhält. Die Aktionen `r`, `l`, `o` und `u` geben an, ob Pac-Man sich ein Feld nach rechts, links, oben oder unten bewegt.



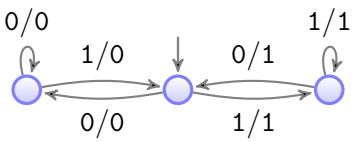
Aufgabe 5 (3 Punkte)

Ein Schwarz-Weiß-Bild sei durch eine Bitfolge bestehend aus `0`ern und `1`ern kodiert. Durch Schmutz oder Rauschen können einzelne Pixel verfälscht sein. Daher soll die Bitfolge geglättet werden, indem Pixel, die sich von ihren beiden Nachbarn unterscheiden, invertiert werden. Beispiele:

Bitfolge:	`11100010011101101000 10`	`00011101100010010111 01`
Geglättete Bitfolge:	`11110000001111111100 00`	`00001111110000000011 11`

Entwerfen Sie einen Mealy-Automaten für eine derartige Glättung von Bitfolgen.

Lösung



Aufgabe 6 (2 Punkte)

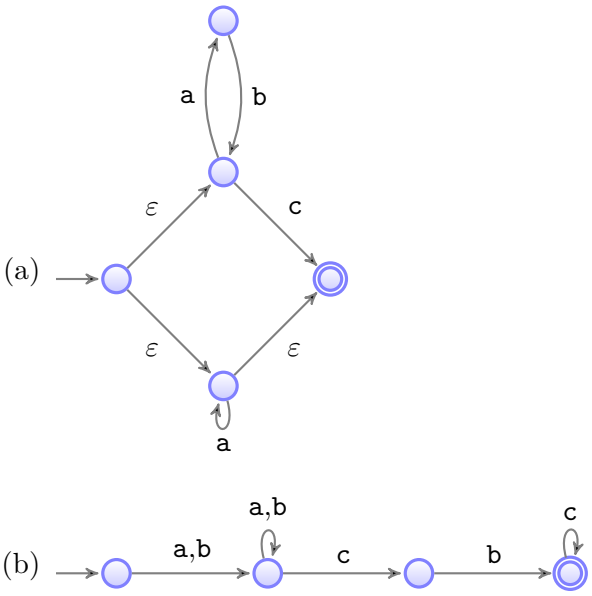
Geben Sie endliche Automaten an, die dieselbe Sprache beschreiben wie die folgenden regulären Ausdrücke in algebraischer Notation.

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

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

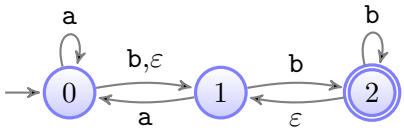
Lösung

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.



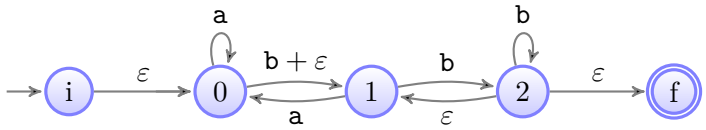
Aufgabe 7 (4 Punkte)

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!



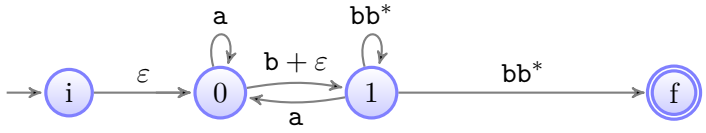
Lösung

Neuer Anfangs- und Endzustand:



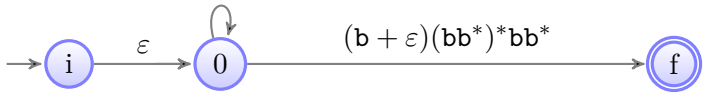
Wir eliminieren die Zustände in der Reihenfolge 2, 1 und 0; die anderen Reihenfolgen sind ebenfalls möglich.

Elimination von Zustand 2:

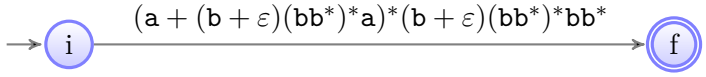


Elimination von Zustand 1:

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



Elimination von Zustand 0:



Die Sprache des ursprünglichen Automaten wird also durch den Ausdruck

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

beschrieben. Mit ein wenig Übung lässt sich der Ausdruck noch vereinfachen. Da $(\mathtt{bb}^*)^* = \mathtt{b}^*$ gilt, erhalten wir

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

Weiters gilt $\mathtt{b}^*\mathtt{bb}^* = \mathtt{bb}^*$:

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

Das Distributivgesetz liefert $(\mathtt{b}+\varepsilon)\mathtt{bb}^* = \mathtt{b}^2\mathtt{b}^* + \mathtt{bb}^*$. Da die Sprache des Ausdrucks $\mathtt{b}^2\mathtt{b}^*$ eine Teilmenge jener von $\mathtt{bb}^*$ ist, können wir den ersten weglassen.

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

Ähnliches gilt für den ersten Teil: $\mathtt{a} + (\mathtt{b} + \varepsilon)\mathtt{b}^*\mathtt{a} = \mathtt{a} + \mathtt{bb}^*\mathtt{a} + \mathtt{b}^*\mathtt{a} = \mathtt{a} + \mathtt{b}^*\mathtt{a} = \mathtt{b}^*\mathtt{a}$. Das Argument ist wieder dasselbe: $\mathtt{b}^*\mathtt{a}$ beschreibt die Menge aller Wörter, die mit einer beliebigen Anzahl von $\mathtt{b}$s beginnen und mit einem $\mathtt{a}$ enden. In dieser Menge ist sowohl $\mathtt{a}$ als auch $\mathtt{bb}^*\mathtt{a}$ enthalten. Wir erhalten:

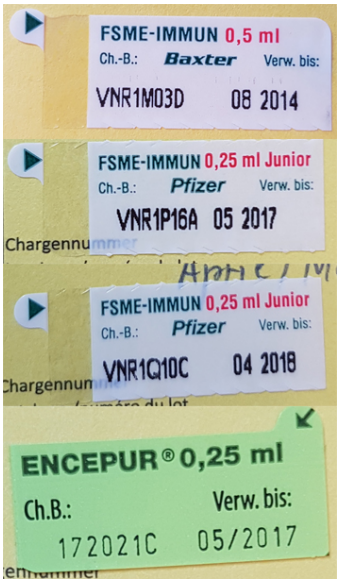
$$(\mathtt{b}^*\mathtt{a})^*\mathtt{bb}^*$$

Dieser Ausdruck beschreibt alle Wörter mit beliebig vielen `a`s, denen jeweils eine beliebige Anzahl von `b`s vorangehen. Am Ende folgen beliebige viele `b`s, mindestens aber eines. Das sind genau jene Wörter über dem Alphabet $\{\mathtt{a}, \mathtt{b}\}$, die mit einem `b` enden, was einfacher beschrieben werden kann durch

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

Aufgabe 8 (3 Punkte)

Bei vielen Impfstoffen gibt es auf der Spritze ein kleines Etikett, das die wichtigsten Informationen zum Impfstoff enthält. Dieses Etikett kann abgelöst und in den Impfpass geklebt werden. Die Abbildung rechts zeigt Beispiele, wie diese Etiketten für einen Impfstoff gegen FSME aussehen. In der ersten Zeile ist zunächst die Bezeichnung des Impfstoffes in Großbuchstaben angeführt, gefolgt von der Impfdosis – 0,25 oder 0,5 ml – bei 0,25 ml eventuell mit dem Zusatz „Junior“. Anschließend folgt optional die Bezeichnung der Erzeugerfirma.[a]
Beschreiben Sie den Aufbau der ersten Zeile solcher Impfstoff-Etiketten mit den folgenden Methoden. Treffen Sie sinnvolle Annahmen, wenn Ihnen Informationen fehlen. Gehen Sie davon aus, dass zwischen den einzelnen Informationen wie in der Abbildung jeweils ein Leerzeichen ist.



[a] die Erzeugerfirma gehört zur ersten Zeile dazu, auch wenn es optisch so aussieht als wäre sie in der nächsten Zeile. Die Zeichenketten „Ch.B.:“ und „Verw.bis“ hingegen gehören zur zweiten Zeile.

(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.

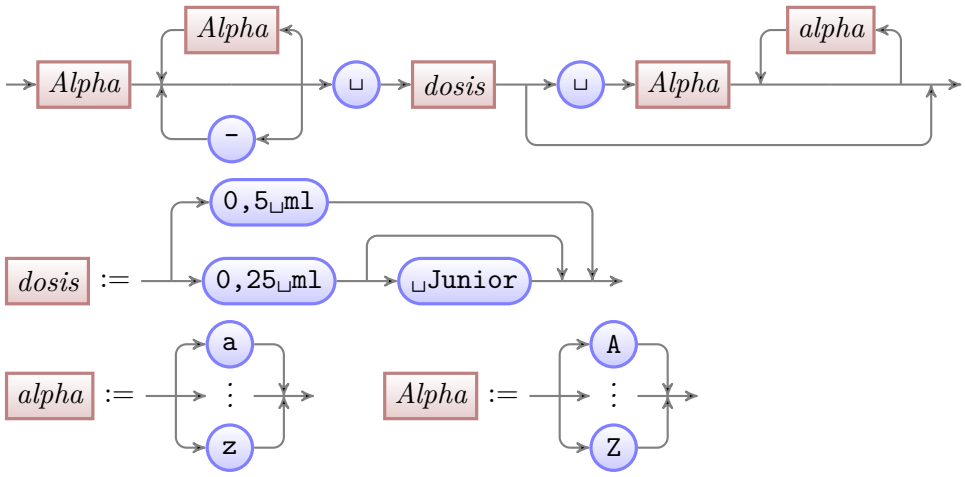
$$\mathit{Alpha}\,(\mathit{Alpha} + \texttt{"-"})^*\ \texttt{"␣"}\ \mathit{dosis}\,((\texttt{"␣"}\ \mathit{Alpha}\ \mathit{alpha}^*) + \varepsilon)$$

mit den Abkürzungen

$$\begin{aligned} \mathit{alpha} &:= \texttt{"a"} + \cdots + \texttt{"z"} \\ \mathit{Alpha} &:= \texttt{"A"} + \cdots + \texttt{"Z"} \\ \mathit{dosis} &:= \texttt{"0,5␣ml"} + \texttt{"0,25␣ml"}(\texttt{"␣Junior"} + \varepsilon) \end{aligned}$$

(b) `^[A-Z][A-Z-]*␣(0,5␣ml|0,25␣ml(␣Junior)?)(␣[A-Z][a-z]*)?$`

(c) Syntaxdiagramm:



Aufgabe 9 (3 Punkte)

Der Lambdakalkül (oder λ-Kalkül) ist eines der Modelle für Berechenbarkeit. Jede mit einem Computer berechenbare Funktion lässt sich durch einen Ausdruck im Lambdakalkül (einen sogenannten λ-Term) darstellen und berechnen.
Es gibt drei Arten von λ-Termen. Zunächst ist bereits jede *Variable* ein solcher. Weiters können mittels λ-*Abstraktion* anonyme (d.h., unbenannte) Funktionen definiert werden, die eine Variable als Parameter besitzen; diese wird später bei der Abarbeitung im zugehörigen λ-Term ersetzt. λ-Abstraktionen werden mit dem Zeichen „λ“ eingeleitet; ein Punkt trennt die Variable vom zugehörigen λ-Term. Eine *Applikation* steht für die Anwendung einer Funktion auf ein Argument und besteht aus zwei aufeinanderfolgenden λ-Termen, einem für die Funktion und einem für das Argument. Getrennt werden die beiden Terme durch ein Leerzeichen. λ-Abstraktionen und Applikationen werden geklammert.

Beispiele zulässiger λ-Terme:

$(\lambda x.x)$	λ-Abstraktion
$(x\ y)$	Applikation, bei der x auf y angewendet wird.
$((x\ y)\ z)$	Applikation, bei der das Ergebnis von $(x\ y)$ auf z angewendet wird.
$(\lambda x.(\lambda x.(x\ y)))$	λ-Term mit zwei λ-Abstraktionen und einer Applikation.

(a) Geben Sie eine induktive Definition für die Menge der λ-Terme an. Nehmen Sie an, dass die Menge V der zulässigen Variablenbezeichnungen bereits existiert.

(b) Zeigen Sie, dass der λ-Term $(\lambda x.(\lambda y.((x\ (\lambda z.y))\ z)))$ in Ihrer induktiv definierten Menge liegt.

Lösung

(a) Die Menge der Lambda-Terme Λ ist die kleinst Menge, für die gilt:

(L1) $V \subseteq \Lambda$, d.h., jede Variable ist ein Lambda-Term.

(L2) $(\lambda x.t) \in \Lambda$, falls $x \in V$ und $t \in \Lambda$, d.h., $(\lambda x.t)$ ist ein Lambda-Term, falls x eine Variable und t ein Lambda-Term ist.

(L3) $(s\ t) \in \Lambda$ falls $s, t \in \Lambda$, d.h., $(s\ t)$ ist ein Lambda-Term, falls s und t welche sind.

Das Zeichen Λ ist der Großbuchstabe zum Kleinbuchstaben λ; die Menge der Lambda-Terme kann auch mit jedem anderen Zeichen oder Namen bezeichnet werden.

(b)

$$\dfrac{\dfrac{x \in V}{x \in \Lambda}\ \text{L1} \quad \dfrac{\dfrac{z \in V \quad \dfrac{y \in V}{y \in \Lambda}\ \text{L1}}{(\lambda z.y) \in \Lambda}\ \text{L2}}{(x\ (\lambda z.y)) \in \Lambda}\ \text{L3} \quad \dfrac{z \in V}{z \in \Lambda}\ \text{L1}}{\dfrac{y \in V \qquad ((x\ (\lambda z.y))\ z) \in \Lambda}{(\lambda y.((x\ (\lambda z.y))\ z)) \in \Lambda}\ \text{L2}}\ \text{L3}}{(\lambda x.(\lambda y.((x\ (\lambda z.y))\ z))) \in \Lambda}\ \text{L2}$$

Diese Notation ist wie folgt zu lesen. Jeder Querstrich steht für einen wenn-dann-Schluss, wobei oberhalb die Annahmen und unterhalb die Konsequenz stehen. Daneben steht die Regel, die diesen Schluss rechtfertigt. Somit steht $\frac{A \quad B}{C}\ ^{r}$ für den Satz „Wegen Regel r folgt aus A und B die Aussage C“ oder „Da A und B gilt, liefert Regel r die Aussage C.“ Ausgeschrieben würde der Ableitungsbaum so beginnen:

- Da y eine Variable ist, ist y auch ein Lambda-Term (Regel L1).
- Da z eine Variable und y ein Lambda-Term ist (siehe vorige Zeile), ist $(\lambda z.y)$ ein Lambda-Term (Regel L2).
- Da x eine Variable ist, ist x auch ein Lambda-Term (Regel L1).
- Da sowohl x aus auch $(\lambda z.y)$ Lambda-Terme sind, ist auch $(x\ (\lambda z.y))$ ein Lambda-Term (Regel L3).
- ...