Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 1

Aufgabe 1

Wir benutzen hilfsweise die folgende Aussage:

Für alle $u, v \in \Sigma^*$ gilt: $(uv)^R = v^R u^R$

a) Zu zeigen: $(w^R)^R = w$ für alle $w \in \Sigma^*$.
   Induktion über die Wortstruktur

   i) Induktionsverankerung:
      $(\varepsilon^R)^R = \varepsilon^R = \varepsilon$

   ii) Induktionsschluß (sei $(w^R)^R = w$):
      $((wa)^R)^R = (a(w^R))^R = (w^R)^R a = wa$

b) Zu zeigen: $(w^R)^n = (w^n)^R$ für alle $w \in \Sigma^*$ und $n \geq 0$.
   Induktion über n

   i) Induktionsverankerung:
      $(w^R)^0 = \varepsilon = (w^0)^R$

   ii) Induktionsschluß (sei $(w^R)^n = (w^n)^R$):
      $(w^R)^{n+1} = w^R(w^R)^n = w^R(w^n)^R = (w^n w)^R = (w^{n+1})^R$

Aufgabe 2

a) Seien $L_1, L_2 \subseteq \Sigma^*$, und sei $L_1 \subseteq L_2$.

$$\begin{array}{ll}
 & w \in L_1^* \\
\curvearrowleft\!\!\curvearrowright & \exists n \in \mathbb{N}:\ w \in L_1^n \\
\curvearrowleft\!\!\curvearrowright & \exists n \in \mathbb{N},\ \exists u_1, \ldots, u_n \in L_1:\ w = u_1 \ldots u_n \\
\curvearrowright & \exists n \in \mathbb{N},\ \exists u_1, \ldots, u_n \in L_2:\ w = u_1 \ldots u_n \\
\curvearrowleft\!\!\curvearrowright & \exists n \in \mathbb{N}:\ w \in L_2^n \\
\curvearrowleft\!\!\curvearrowright & w \in L_2^*
\end{array}$$

   Also gilt $L_1^* \subseteq L_2^*$.

b) Seien $L_1, L_2 \subseteq \Sigma^*$.

$$\begin{array}{ll}
 & L_1 \subseteq L_1 \cup L_2 \quad \text{und } L_2 \subseteq L_1 \cup L_2 \\
\curvearrowright & L_1^* \subseteq (L_1 \cup L_2)^* \text{ und } L_2^* \subseteq (L_1 \cup L_2)^* \\
\curvearrowright & L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*
\end{array}$$

Aufgabe 3

a) $\alpha_1 = (1 \vee 01 \vee 0011)^*(0 \vee \Lambda^*)$

b) $\alpha_2 = (1 \vee 01^*0)^* \vee (0 \vee 10^*10^*1)^*$

c) $\alpha_3 = (01 \vee 10)^*$

Aufgabe 4

Wir geben reguläre Ausdrücke $\beta_1, \beta_2, \beta_3$ an, so daß $\beta_i \sim \alpha_i$ und $sh(\beta_i) < sh(\alpha_i)$, $i = 1, 2, 3$.

a) $\beta_1 = (ab \vee ba)^*$

b) $\beta_2 = a(a \vee b^*c)^* \vee \Lambda^*$

c) $\beta_3 = (abc \vee ab)^*ab \vee \Lambda^*$

Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 2

Aufgabe 1

$\mathfrak{A}^P = \langle \widehat{Q}, \{a,b\}, \widehat{\delta}, \widehat{q_0}, \widehat{F} \rangle \in \mathrm{DFA}(\{a,b\})$ ist durch seine Transitionstafel gegeben wie folgt:

$\widehat{\delta}$	a	b
$\rightarrow\Rightarrow \{q_0, q_1\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\Rightarrow \{q_1, q_2\}$	$\{q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_2\}$	$\emptyset$	$\{q_0, q_1\}$
$\Rightarrow \{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$\emptyset$	$\emptyset$	$\emptyset$

Aufgabe 2

Zu zeigen: $\overline{\widehat{\delta}}(T,w) = \overline{\delta}(T,w)$ für alle $w \in \Sigma^*$ und $T \in \widehat{Q}$.

i) Induktionsverankerung:
$\overline{\widehat{\delta}}(T,\varepsilon) = T = \varepsilon(T) = \overline{\delta}(T,\varepsilon)$

ii) Induktionsschluß (sei $\overline{\widehat{\delta}}(T,w) = \overline{\delta}(T,w)$):

$$
\begin{aligned}
& \overline{\widehat{\delta}}(T,wa) \\
= \; & \widehat{\delta}(\overline{\widehat{\delta}}(T,w),a) \\
= \; & \overline{\delta}(\overline{\widehat{\delta}}(T,w),a) \\
= \; & \varepsilon(\textstyle\bigcup_{q\in\overline{\delta}(\overline{\widehat{\delta}}(T,w),\varepsilon)} \delta(q,a)) \\
= \; & \varepsilon(\textstyle\bigcup_{q\in\overline{\widehat{\delta}}(T,w)} \delta(q,a)) \\
= \; & \varepsilon(\textstyle\bigcup_{q\in\overline{\delta}(T,w)} \delta(q,a)) \\
= \; & \overline{\delta}(T,wa)
\end{aligned}
$$

Aufgabe 3

a) *Idee:* Wir konstruieren zuerst einen Automaten $\mathfrak{A}' \in \mathrm{NFA}(\Sigma)$, so daß $L(\mathfrak{A}') = L(\mathfrak{A}_1)L(\mathfrak{A}_2)$. Sei dann $\mathfrak{A} = \mathfrak{A}'^P \in \mathrm{DFA}(\Sigma)$, und es gilt $L(\mathfrak{A}) = L(\mathfrak{A}_1)L(\mathfrak{A}_2)$. $\mathfrak{A}'$ entsteht nun aus $\mathfrak{A}_1$ und $\mathfrak{A}_2$ durch „Vereinigung" der Zustandsmengen und Transitionen sowie Hinzunahme jeweils einer ε-Transition von den Endzuständen des Automaten $\mathfrak{A}_1$ zum Anfangszustand des Automaten $\mathfrak{A}_2$. q_0^1 wird Anfangszustand, F_2 Endzustandsmenge. Formal sei $\mathfrak{A}' = \langle Q, \Sigma, \delta, q_0, F\rangle$ gegeben wie folgt:

- $Q = Q_1 \cup Q_2$

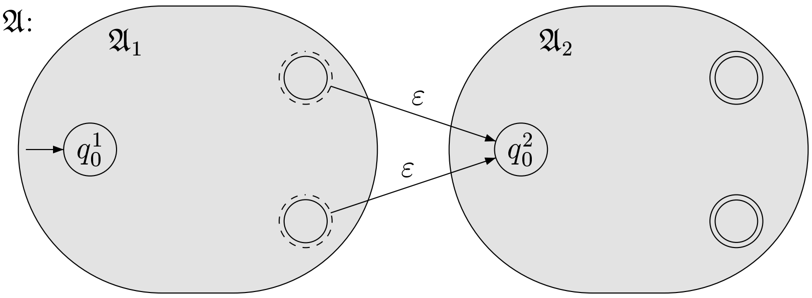
- Für alle $q \in Q$ und $a \in \Sigma$:

  $$\delta(q,a) = \begin{cases} \{\delta_1(q,a)\} & \text{, falls } q \in Q_1 \\ \{\delta_2(q,a)\} & \text{, falls } q \in Q_2 \end{cases}$$

  $$\delta(q,\varepsilon) = \begin{cases} \emptyset & \text{, falls } q \notin F_1 \\ \{q_0^2\} & \text{, falls } q \in F_1 \end{cases}$$

- $q_0 = q_0^1$
- $F = F_2$

$\mathfrak{A}$:



b) *Idee:* Wir übernehmen $\mathfrak{A}_1$ fast unverändert. Nur die Endzustandsmenge besteht nun aus denjenigen Zuständen, von denen ein Pfad zu einem Zustand aus F_1 führt. Sei also $\mathfrak{A} = \langle Q, \Sigma, \delta, q_0, F\rangle$ gegeben durch $Q = Q_1$, $\delta = \delta_1$, $q_0 = q_0^1$ und $F = \{q \in Q \mid \overline{\delta}(q,w) \in F_1,\ w \in \Sigma^*\}$.

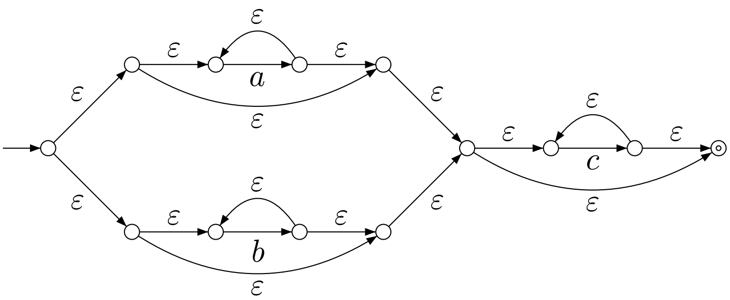
Korrektheitsbeweis:

$$\begin{array}{ll} & w \in L(\mathfrak{A}) \\ \curvearrowleftright & \overline{\delta}(q_0, w) \in F \\ \curvearrowleftright & \exists u \in \Sigma^* :\ \overline{\delta}(\overline{\delta}(q_0,w),u) \in F_1 \\ \curvearrowleftright & \exists u \in \Sigma^* :\ \overline{\delta}(q_0,wu) \in F_1 \\ \curvearrowleftright & \exists v \in L(\mathfrak{A}_1) :\ w \in \mathit{Prf}(v) \\ \curvearrowleftright & w \in \mathit{Pref}(L(\mathfrak{A}_1)) \end{array}$$

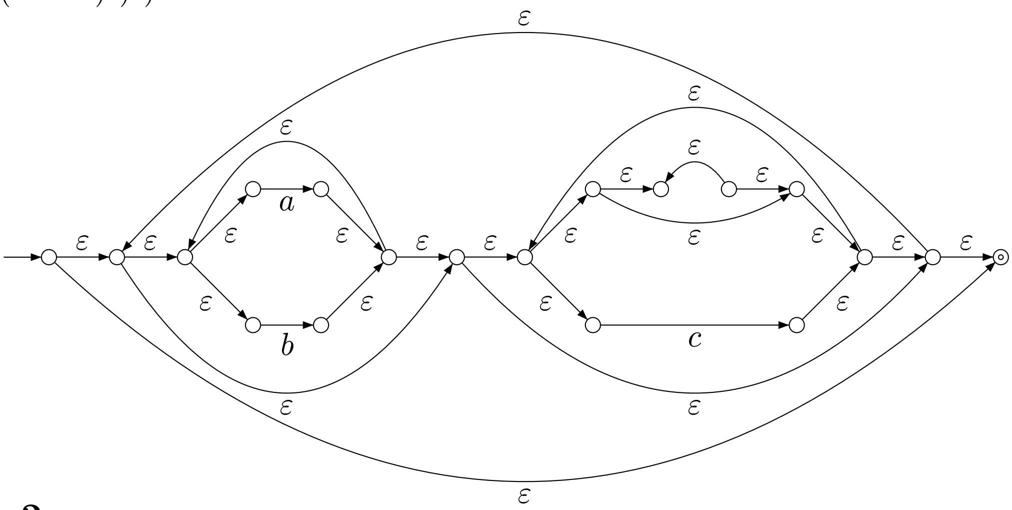
Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 3

Aufgabe 1

$\mathfrak{A}((a^* \vee b^*)c^*)$:



$\mathfrak{A}(((a \vee b)^*(\Lambda^* \vee c)^*)^*)$:



Aufgabe 2

Zu $\mathfrak{A}_1$:

$$\begin{array}{lclclcl}
W^3_{12} & = & W^2_{12} \cup W^2_{13}(W^2_{33})^* W^2_{32} & = & \llbracket a^*b(ba^*b)^* \vee a^*b(ba^*b)^*a(\Lambda^* \vee (ba^*b)^*a)^*ba^*b(ba^*b)^* \rrbracket & & \\
 & & & = & \llbracket a^*b(ba^*b)^*(\Lambda^* \vee a((ba^*b)^*a)^*ba^*b(ba^*b)^*) \rrbracket & & \\
\hline
W^2_{12} & = & W^1_{12} \cup W^1_{12}(W^1_{22})^* W^1_{22} & = & \llbracket a^*b \vee a^*b(\Lambda^* \vee ba^*b)^*(\Lambda^* \vee ba^*b) \rrbracket & = & \llbracket a^*b(ba^*b)^* \rrbracket \\
W^2_{13} & = & W^1_{13} \cup W^1_{12}(W^1_{22})^* W^1_{23} & = & \llbracket \Lambda \vee a^*b(\Lambda^* \vee ba^*b)^*a \rrbracket & = & \llbracket a^*b(ba^*b)^*a \rrbracket \\
W^2_{33} & = & W^1_{33} \cup W^1_{32}(W^1_{22})^* W^1_{23} & = & \llbracket (\Lambda^* \vee a) \vee ba^*b(\Lambda^* \vee ba^*b)^*a \rrbracket & = & \llbracket \Lambda^* \vee (ba^*b)^*a \rrbracket \\
W^2_{32} & = & W^1_{32} \cup W^1_{32}(W^1_{22})^* W^1_{22} & = & \llbracket ba^*b \vee ba^*b(\Lambda^* \vee ba^*b)^*(\Lambda^* \vee ba^*b) \rrbracket & = & \llbracket ba^*b(ba^*b)^* \rrbracket \\
\hline
W^1_{12} & = & W^0_{12} \cup W^0_{11}(W^0_{11})^* W^0_{12} & = & \llbracket b \vee (\Lambda^* \vee a)(\Lambda^* \vee a)^*b \rrbracket & = & \llbracket a^*b \rrbracket \\
W^1_{22} & = & W^0_{22} \cup W^0_{21}(W^0_{11})^* W^0_{12} & = & \llbracket \Lambda^* \vee b(\Lambda^* \vee a)^*b \rrbracket & = & \llbracket \Lambda^* \vee ba^*b \rrbracket \\
W^1_{13} & = & W^0_{13} \cup W^0_{11}(W^0_{11})^* W^0_{13} & = & \llbracket \Lambda \vee (\Lambda^* \vee a)(\Lambda^* \vee a)^*\Lambda \rrbracket & = & \llbracket \Lambda \rrbracket \\
W^1_{23} & = & W^0_{23} \cup W^0_{21}(W^0_{11})^* W^0_{13} & = & \llbracket a \vee b(\Lambda^* \vee a)^*\Lambda \rrbracket & = & \llbracket a \rrbracket \\
W^1_{33} & = & W^0_{33} \cup W^0_{31}(W^0_{11})^* W^0_{13} & = & \llbracket (\Lambda^* \vee a) \vee b(\Lambda^* \vee a)^*\Lambda \rrbracket & = & \llbracket \Lambda^* \vee a \rrbracket \\
W^1_{32} & = & W^0_{32} \cup W^0_{31}(W^0_{11})^* W^0_{12} & = & \llbracket \Lambda \vee b(\Lambda^* \vee a)^*b \rrbracket & = & \llbracket ba^*b \rrbracket \\
\hline
\end{array}$$

$$W^0_{11} = W^0_{33} = \{\varepsilon, a\} = \llbracket \Lambda^* \vee a \rrbracket \qquad W^0_{22} = \{\varepsilon\} = \llbracket \Lambda^* \rrbracket \qquad W^0_{13} = W^0_{32} = \emptyset = \llbracket \Lambda \rrbracket$$

$$W^0_{12} = W^0_{21} = W^0_{31} = \{b\} = \llbracket b \rrbracket \qquad W^0_{23} = \{a\} = \llbracket a \rrbracket$$

Sei also $\alpha_1 = a^*b(ba^*b)^*(\Lambda^* \vee a((ba^*b)^*a)^*ba^*b(ba^*b)^*)$.

Zu $\mathfrak{A}_2$:

$$
\begin{array}{lclcll}
W_{13}^4 & = & W_{13}^3 \cup W_{14}^3 (W_{44}^3)^* W_{43}^3 & = & [\![b \vee bb(\Lambda^* \vee (a \vee b)b)^*(a \vee b)]\!] & = & [\![b(b(a \vee b))^*]\!] \\
\hline
W_{13}^3 & = & W_{13}^2 \cup W_{13}^2 (W_{33}^2)^* W_{33}^2 & = & [\![b \vee b(\Lambda^*)^*\Lambda^*]\!] & = & [\![b]\!] \\
W_{14}^3 & = & W_{14}^2 \cup W_{13}^2 (W_{33}^2)^* W_{34}^2 & = & [\![\Lambda \vee b(\Lambda^*)^* b]\!] & = & [\![bb]\!] \\
W_{44}^3 & = & W_{44}^2 \cup W_{43}^2 (W_{33}^2)^* W_{34}^2 & = & [\![\Lambda^* \vee (a \vee b)(\Lambda^*)^* b]\!] & = & [\![\Lambda^* \vee (a \vee b)b]\!] \\
W_{43}^3 & = & W_{43}^2 \cup W_{43}^2 (W_{33}^2)^* W_{33}^2 & = & [\![(a \vee b) \vee (a \vee b)(\Lambda^*)^*\Lambda^*]\!] & = & [\![a \vee b]\!] \\
\hline
W_{13}^2 & = & W_{13}^1 \cup W_{12}^1 (W_{22}^1)^* W_{23}^1 & = & [\![b \vee a(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![b]\!] \\
W_{33}^2 & = & W_{33}^1 \cup W_{32}^1 (W_{22}^1)^* W_{23}^1 & = & [\![\Lambda^* \vee a(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![\Lambda^*]\!] \\
W_{14}^2 & = & W_{14}^1 \cup W_{12}^1 (W_{22}^1)^* W_{24}^1 & = & [\![\Lambda \vee a(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![\Lambda]\!] \\
W_{34}^2 & = & W_{34}^1 \cup W_{32}^1 (W_{22}^1)^* W_{24}^1 & = & [\![b \vee a(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![b]\!] \\
W_{44}^2 & = & W_{44}^1 \cup W_{42}^1 (W_{22}^1)^* W_{24}^1 & = & [\![\Lambda^* \vee \Lambda(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![\Lambda^*]\!] \\
W_{43}^2 & = & W_{43}^1 \cup W_{42}^1 (W_{22}^1)^* W_{23}^1 & = & [\![(a \vee b) \vee \Lambda(\Lambda^* \vee a \vee b)^*\Lambda]\!] & = & [\![a \vee b]\!] \\
\hline
W_{13}^1 & = & W_{13}^0 \cup W_{11}^0 (W_{11}^0)^* W_{13}^0 & = & [\![b \vee \Lambda^*(\Lambda^*)^* b]\!] & = & [\![b]\!] \\
W_{12}^1 & = & W_{12}^0 \cup W_{11}^0 (W_{11}^0)^* W_{12}^0 & = & [\![a \vee \Lambda^*(\Lambda^*)^* a]\!] & = & [\![a]\!] \\
W_{22}^1 & = & W_{22}^0 \cup W_{21}^0 (W_{11}^0)^* W_{12}^0 & = & [\![(\Lambda^* \vee a \vee b) \vee \Lambda(\Lambda^*)^* a]\!] & = & [\![\Lambda^* \vee a \vee b]\!] \\
W_{23}^1 & = & W_{23}^0 \cup W_{21}^0 (W_{11}^0)^* W_{13}^0 & = & [\![\Lambda \vee \Lambda(\Lambda^*)^* b]\!] & = & [\![\Lambda]\!] \\
W_{33}^1 & = & W_{33}^0 \cup W_{31}^0 (W_{11}^0)^* W_{13}^0 & = & [\![\Lambda^* \vee \Lambda(\Lambda^*)^* b]\!] & = & [\![\Lambda^*]\!] \\
W_{32}^1 & = & W_{32}^0 \cup W_{31}^0 (W_{11}^0)^* W_{12}^0 & = & [\![a \vee \Lambda(\Lambda^*)^* a]\!] & = & [\![a]\!] \\
W_{14}^1 & = & W_{14}^0 \cup W_{11}^0 (W_{11}^0)^* W_{14}^0 & = & [\![\Lambda \vee \Lambda^*(\Lambda^*)^*\Lambda]\!] & = & [\![\Lambda]\!] \\
W_{24}^1 & = & W_{24}^0 \cup W_{21}^0 (W_{11}^0)^* W_{14}^0 & = & [\![\Lambda \vee \Lambda(\Lambda^*)^*\Lambda]\!] & = & [\![\Lambda]\!] \\
W_{34}^1 & = & W_{34}^0 \cup W_{31}^0 (W_{11}^0)^* W_{14}^0 & = & [\![b \vee \Lambda(\Lambda^*)^*\Lambda]\!] & = & [\![b]\!] \\
W_{44}^1 & = & W_{44}^0 \cup W_{41}^0 (W_{11}^0)^* W_{14}^0 & = & [\![\Lambda^* \vee \Lambda(\Lambda^*)^*\Lambda]\!] & = & [\![\Lambda^*]\!] \\
W_{42}^1 & = & W_{42}^0 \cup W_{41}^0 (W_{11}^0)^* W_{12}^0 & = & [\![\Lambda \vee \Lambda(\Lambda^*)^* a]\!] & = & [\![\Lambda]\!] \\
W_{43}^1 & = & W_{43}^0 \cup W_{41}^0 (W_{11}^0)^* W_{13}^0 & = & [\![(a \vee b) \vee \Lambda(\Lambda^*)^* b]\!] & = & [\![a \vee b]\!] \\
\hline
W_{13}^0 & = & \{b\} & = & [\![b]\!] & & \\
W_{11}^0 & = & \{\varepsilon\} & = & [\![\Lambda^*]\!] & & \\
W_{12}^0 & = & \{a\} & = & [\![a]\!] & & \\
W_{22}^0 & = & \{\varepsilon, a, b\} & = & [\![\Lambda^* \vee a \vee b]\!] & & \\
W_{33}^0 & = & \{\varepsilon\} & = & [\![\Lambda^*]\!] & & \\
W_{32}^0 & = & \{a\} & = & [\![a]\!] & & \\
W_{34}^0 & = & \{b\} & = & [\![b]\!] & & \\
W_{44}^0 & = & \{\varepsilon\} & = & [\![\Lambda^*]\!] & & \\
W_{43}^0 & = & \{a, b\} & = & [\![a \vee b]\!] & & \\
\multicolumn{7}{l}{W_{21}^0 = W_{23}^0 = W_{31}^0 = W_{14}^0 = W_{24}^0 = W_{41}^0 = W_{42}^0 = \emptyset = [\![\Lambda]\!]}
\end{array}
$$

Sei also $\alpha_2 = b(b(a \vee b))^*$.

Aufgabe 3

a) Wir zeigen, daß $L_1 = \{a^m ba^n ba^{m+n} \mid m, n \geq 0\}$ nicht regulär ist. Angenommen, L_1 sei regulär. Dann existiert ein Pumping-Index $k \in \mathbb{N}$ mit den Eigenschaften des Pumping-Lemmas. Sei $x = a^k bba^k \in L_1$. Es gibt also eine Zerlegung $x = uvw$, so daß

- $|v| \geq 1$,
- $|uv| \leq k$ und
- $uv^i w \in L_1$ für alle $i \in \mathbb{N}$.

Demnach gilt $uv \in \mathit{Prf}(a^k)$ und $a^{k-|v|}bba^k \in L_1$. Das aber widerspricht der Definition von L_1. Deshalb ist L_1 nicht regulär.

b) Wir zeigen, daß $L_2 = \{w \in \{a, b\}^* \mid w^R = w\}$ nicht regulär ist und nehmen an, L_2 sei regulär. Dann existiert ein Pumping-Index $k \in \mathbb{N}$ mit den Eigenschaften des Pumping-Lemmas. Sei nun $x = a^k ba^k \in L_2$ und uvw eine Zerlegung von x gemäß Pumping-Lemma. Dann gilt $uv \in \mathit{Prf}(a^k)$ und deshalb $a^{k-|v|}ba^k \in L_2$. Da aber $a^{k-|v|}ba^k \neq a^k ba^{k-|v|}$, kann L_2 nicht regulär sein.

c) Wir zeigen, daß $L_3^n = \{w \in \{a, b\}^* \mid (|w| < n \text{ und } |w|_a = |w|_b) \text{ oder } (w = uv,\ |v| = n, \text{ und } |v|_a = |v|_b)\}$ für alle $n \in \mathbb{N}$ regulär ist und konstruieren jeweils einen Automaten $\mathfrak{A}^n \in \mathrm{NFA}(\{a, b\})$, so daß $L(\mathfrak{A}^n) = L_3^n$.

Idee: $\mathfrak{A}^n$ verzweigt spontan (d.h. über ε-Transitionen) in zwei Unterautomaten $\mathfrak{A}^{\geq n}$ und $\mathfrak{A}^{<n}$. $\mathfrak{A}^{\geq n}$ verarbeitet Wörter der Länge $\geq n$, $\mathfrak{A}^{<n}$ Wörter der Länge $< n$. Wir beschränken uns auf die Konstruktion von $\mathfrak{A}^{\geq n}$, da $L(\mathfrak{A}^{<n})$ endlich ist.

Sei nun $f : \{1, \ldots, 2^n\} \to \{w \in \{a, b\}^* \mid |w| = n\}$ eine bijektive Abbildung. Für ein Wort $w = a_1 \ldots a_n$ bezeichne außerdem $w[i] := a_i$ den i-ten Buchstaben von w. Dann sei $\mathfrak{A}^{\geq n} = \langle Q, \{a, b\}, \delta, q_0, F\rangle$ gegeben wie folgt:

- $Q = \{q_0\} \cup \{q_{ij} \mid i \in \{1, \ldots, 2^n\},\ j \in \{0, \ldots, n\}\}$
- $\delta(q_0, a) = \delta(q_0, b) = \{q_0\}$

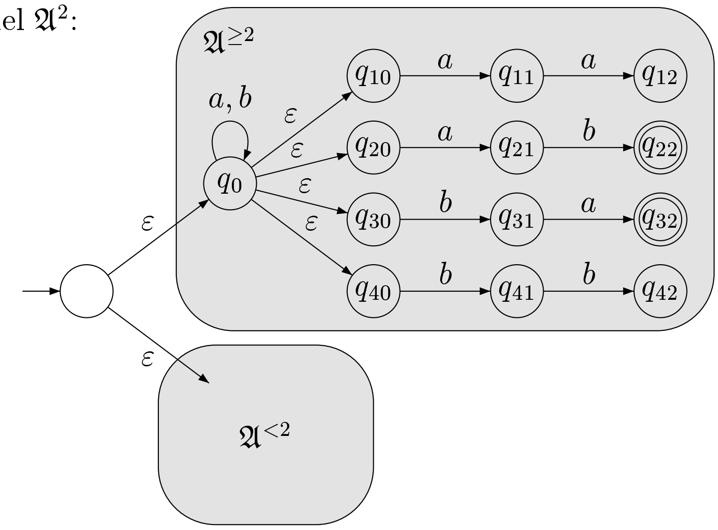
  $\delta(q_0, \varepsilon) = \{q_{i0} \mid i \in \{1, \ldots, 2^n\}\}$

  $$\delta(q_{ij}, c) = \begin{cases} \{q_{i(j+1)}\} & \text{, falls } j < n \text{ und } c = f(i)[j+1] \\ \emptyset & \text{, sonst} \end{cases}$$

  $\delta(q_{ij}, \varepsilon) = \emptyset$ für alle $i \in \{1, \ldots, 2^n\},\ j \in \{0, \ldots, n\}$
- $F = \{q_{in} \mid |f(i)|_a = |f(i)|_b\}$

Beachte, daß für ungerade n gilt: $L(\mathfrak{A}^{\geq n}) = \emptyset$

Beispiel $\mathfrak{A}^2$:



Lösungsvorschlag ***Automatentheorie und Formale Sprachen*** – Blatt 4

Aufgabe 1

Idee: Die Automaten $\mathfrak{A}^\cap$ und $\mathfrak{A}^\cup$ geben ein Eingabewort jeweils an die Automaten $\mathfrak{A}_1$ und $\mathfrak{A}_2$ weiter, die das Wort parallel und synchron verarbeiten. Wir wählen also als Zustandsmenge in beiden Fällen das kartesische Produkt der Zustandsmengen Q_1 und Q_2, wobei auf der ersten Komponente das Verhalten des Automaten $\mathfrak{A}_1$ simuliert wird und auf der zweiten das Verhalten des Automaten $\mathfrak{A}_2$. $\mathfrak{A}^\cap$ und $\mathfrak{A}^\cup$ unterscheiden sich schließlich nur durch ihre Endzustandsmengen.

Sei dann

a) $\mathfrak{A}^\cap = \langle Q, \Sigma, \delta, q_0, F\rangle$ gegeben durch

- $Q = Q_1 \times Q_2$
- $\delta : \begin{cases} (Q_1 \times Q_2) \times \Sigma \to Q_1 \times Q_2 \\ ((q_1, q_2), a) \mapsto (\delta_1(q_1, a), \delta_2(q_2, a)) \end{cases}$
- $q_0 = (q_0^1, q_0^2)$
- $F = F_1 \times F_2$

b) $\mathfrak{A}^\cup = \langle Q, \Sigma, \delta, q_0, F\rangle$ gegeben durch

- $Q = Q_1 \times Q_2$
- $\delta : \begin{cases} (Q_1 \times Q_2) \times \Sigma \to Q_1 \times Q_2 \\ ((q_1, q_2), a) \mapsto (\delta_1(q_1, a), \delta_2(q_2, a)) \end{cases}$
- $q_0 = (q_0^1, q_0^2)$
- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

Aufgabe 2

a) Wir zeigen zunächst durch strukturelle Induktion: $\forall \alpha \in \mathrm{RegE}(\Sigma_1) : \; h([\![\alpha]\!]) = [\![h(\alpha)]\!]$. Sei dabei $h(\alpha)$ der reguläre Ausdruck, den wir erhalten, indem wir in α jeden Buchstaben a durch $h(a)$ ersetzen.

i) Induktionsverankerung:
   - $h([\![\Lambda]\!]) = h(\emptyset) = \emptyset = [\![\Lambda]\!] = [\![h(\Lambda)]\!]$
   - $h([\![a]\!]) = h(\{a\}) = \{h(a)\} = [\![h(a)]\!]$

ii) Induktionsschluß (sei $h([\![\alpha]\!]) = [\![h(\alpha)]\!]$ und $h([\![\beta]\!]) = [\![h(\beta)]\!]$):
   - $h([\![\alpha \vee \beta]\!]) =$ (Semantik regulärer Ausdrücke)
     $h([\![\alpha]\!] \cup [\![\beta]\!]) =$ (Erweiterung von h auf Mengen)
     $h([\![\alpha]\!]) \cup h([\![\beta]\!]) =$ (Induktionsvoraussetzung)
     $[\![h(\alpha)]\!] \cup [\![h(\beta)]\!] =$ (Semantik regulärer Ausdrücke)
     $[\![h(\alpha) \vee h(\beta)]\!] =$ (Definition von $h : \mathrm{RegE}(\Sigma_1) \to \mathrm{RegE}(\Sigma_2)$)
     $[\![h(\alpha \vee \beta)]\!]$

- $h(\llbracket \alpha\beta \rrbracket) =$ (Semantik regulärer Ausdrücke)
  $h(\llbracket \alpha \rrbracket \llbracket \beta \rrbracket) =$ (Homomorphie von h)
  $h(\llbracket \alpha \rrbracket) h(\llbracket \beta \rrbracket) =$ (Induktionsvoraussetzung)
  $\llbracket h(\alpha) \rrbracket \llbracket h(\beta) \rrbracket =$ (Semantik regulärer Ausdrücke)
  $\llbracket h(\alpha) h(\beta) \rrbracket =$ (Definition von $h : \mathrm{RegE}(\Sigma_1) \to \mathrm{RegE}(\Sigma_2)$)
  $\llbracket h(\alpha\beta) \rrbracket$
- $h(\llbracket \alpha^* \rrbracket) =$ (Semantik regulärer Ausdrücke)
  $h(\llbracket \alpha \rrbracket^*) =$ (Erweiterung von h auf Mengen und $h(\llbracket \alpha\beta \rrbracket) = \llbracket h(\alpha\beta) \rrbracket$)
  $h(\llbracket \alpha \rrbracket)^* =$ (Induktionsvoraussetzung)
  $\llbracket h(\alpha) \rrbracket^* =$ (Semantik regulärer Ausdrücke)
  $\llbracket h(\alpha)^* \rrbracket =$ (Definition von $h : \mathrm{RegE}(\Sigma_1) \to \mathrm{RegE}(\Sigma_2)$)
  $\llbracket h(\alpha^*) \rrbracket$

Nun zum Beweis der Aussage a):

$$
\begin{array}{ll}
 & L \in \mathcal{L}(\Sigma_1, \mathrm{DFA}) \\
\curvearrowright & \exists \alpha \in \mathrm{RegE}(\Sigma_1) :\ \llbracket \alpha \rrbracket = L \\
\curvearrowright & \exists \alpha \in \mathrm{RegE}(\Sigma_1) :\ h(\llbracket \alpha \rrbracket) = h(L) \\
\curvearrowright & \exists \alpha \in \mathrm{RegE}(\Sigma_1) :\ \llbracket h(\alpha) \rrbracket = h(L) \\
\curvearrowright & \exists \alpha \in \mathrm{RegE}(\Sigma_2) :\ \llbracket \alpha \rrbracket = h(L) \\
\curvearrowright & h(L) \in \mathcal{L}(\Sigma_2, \mathrm{DFA})
\end{array}
$$

b) Gegenbeispiel:

Seien $\Sigma_1 = \Sigma_2 = \{a, b\}$, h gegeben durch $h(a) = h(b) = \varepsilon$ und $L = \{a^n b^n \mid n \in \mathbb{N}\}$. Es gilt $L \notin \mathrm{RegL}(\Sigma_1)$, aber $h(L) = \{\varepsilon\} \in \mathrm{RegL}(\Sigma_2)$.

Aufgabe 3

Sei L_K die Menge der korrekten Klammerausdrücke über $\{(,)\}$.

a) Wir definieren eine unendliche Folge $(w_n)_{n\in\mathbb{N}}$ von Wörtern über $\{(,)\}$ gemäß $w_n := (^n$ für alle $n \in \mathbb{N}$.

$$
\begin{array}{ll}
 & \forall\ n \neq m :\ )^n \in d_{w_n}(L_K) \text{ und })^n \notin d_{w_m}(L_K) \\
\curvearrowright & \forall\ n \neq m :\ d_{w_n}(L_K) \neq d_{w_m}(L_K) \\
\curvearrowright & |D(L_K)| = \infty \\
\curvearrowright & L_K \text{ nicht regulär}
\end{array}
$$

b) Sei der Homomorphismus $h : (\Sigma \cup \{(,), \Lambda, ^*, \vee\})^* \to \{(,)\}^*$ gegeben durch

$$
h(a) = \begin{cases} a & \text{, falls } a \in \{(,)\} \\ \varepsilon & \text{, sonst} \end{cases}
$$

Angenommen, $\mathrm{RegE}(\Sigma)$ sei regulär. Dann ist auch $h(\mathrm{RegE}(\Sigma)) = L_K$ regulär. Widerspruch!
$\rightsquigarrow$ $\mathrm{RegE}(\Sigma)$ ist nicht regulär.

Aufgabe 4

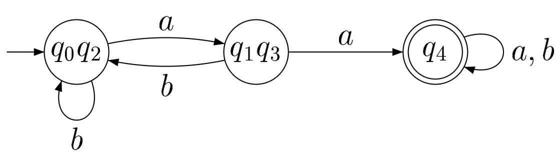
a) $\mathfrak{A} \in \mathrm{DFA}(\{a, b\})$:

δ	a	b
$\rightarrow q_0$	q_1	q_2
q_1	q_4	q_2
q_2	q_3	q_2
q_3	q_4	q_0
$\Rightarrow q_4$	q_4	q_4

Tabelle der Zustandspaare:

q_1	X			
q_2		X		
q_3	X		X	
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Minimaler DFA :



b) q_0q_2 und q_1q_3 sind trennbar über a.

q_0q_2 und q_4 sind trennbar über ε.

q_1q_3 und q_4 sind trennbar über ba.

Lösungsvorschlag ***Automatentheorie und Formale Sprachen*** – Blatt 5

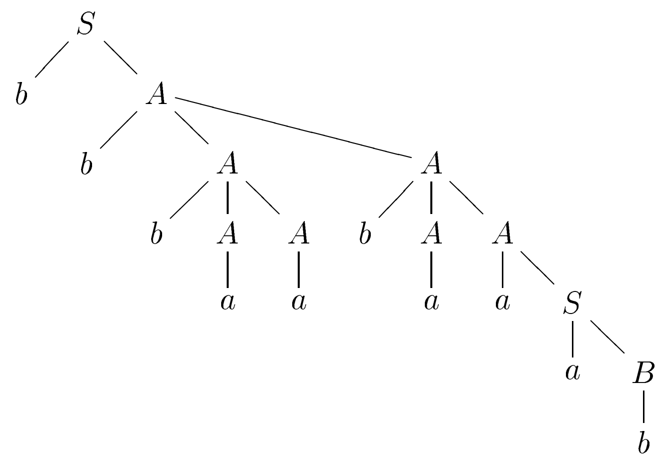
Aufgabe 1

a) Die zum ersten Ableitungsbaum gehörende Linksableitung:
$S \Longrightarrow \overline{bA} \Longrightarrow b\overline{bAA} \Longrightarrow bb\overline{bAA}A \Longrightarrow bbb\overline{a}AA \Longrightarrow bbba\overline{a}A \Longrightarrow bbbaa\overline{bAA} \Longrightarrow bbbaab\overline{a}A \Longrightarrow bbbaaba\overline{aS} \Longrightarrow bbbaabaa\overline{aB} \Longrightarrow bbbaabaaa\overline{b}$.

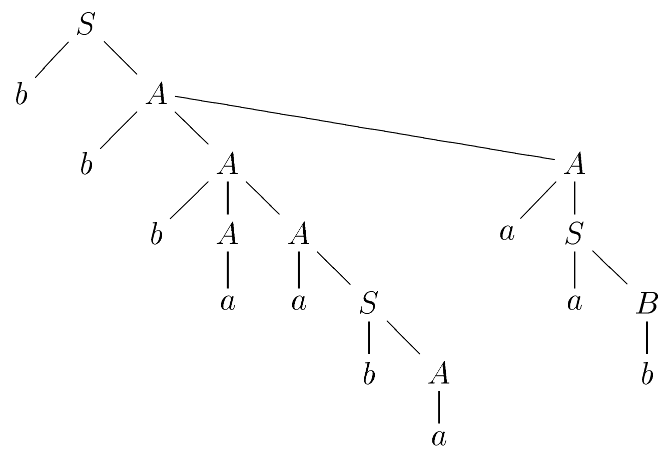
b) Die zum zweiten Ableitungsbaum gehörende Rechtsableitung:
$S \Longrightarrow \overline{bA} \Longrightarrow b\overline{bAA} \Longrightarrow bbA\overline{aS} \Longrightarrow bbAa\overline{aB} \Longrightarrow bbAaa\overline{b} \Longrightarrow bb\overline{bAA}aab \Longrightarrow bbbA\overline{aS}aab \Longrightarrow bbbAa\overline{bA}aab \Longrightarrow bbbAab\overline{a}aab \Longrightarrow bbb\overline{a}abaaab$.

c) Das Wort besitzt drei Ableitungsbäume:

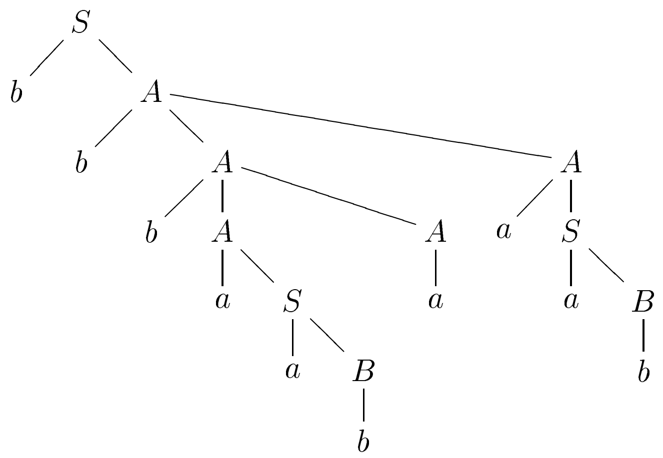
i)



ii)



iii)



d) Die Grammatik G erzeugt die Sprache $L = \{w \in \{a,b\}^* \mid |w|_a = |w|_b,\ w \neq \varepsilon\}$.

Offensichtlich enthalten alle aus S ableitbaren Satzformen α genausoviele a's und A's wie b's und B's. Somit haben alle aus S ableitbaren Wörter genauso viele a's wie b's.

Andererseits läßt sich zu jedem Wort aus L leicht eine Linksableitung in der Grammatik konstruieren (die Grammatik ist fast in Greibach-Normalform!). Mit jedem Ableitungsschritt wird genau ein Buchstabe des Wortes erzeugt.

Ein Beweis erfordert für beide Inklusionen jeweils eine aufwendige simultane dreifache Induktion für die von S, A und B erzeugten Satzformen.

Aufgabe 2

a) $G_1: \quad S \longrightarrow aSa \mid bS \mid \varepsilon$

b) $G_2: \quad S \longrightarrow aSb \mid aSbb \mid aSbbb \mid \varepsilon$

c) $G_2:$
$$\begin{array}{lcl} S & \longrightarrow & AAS'BB \\ S' & \longrightarrow & AS'B \mid \varepsilon \\ A & \longrightarrow & aA \mid b \\ B & \longrightarrow & bB \mid a \end{array}$$

Aufgabe 3

a) Für $G = \langle N, \Sigma, P, S\rangle \in \mathrm{CFG}(\Sigma)$ sei $G^R := \langle N, \Sigma, P^R, S\rangle \in \mathrm{CFG}(\Sigma)$, wobei

$$P^R := \{A \longrightarrow \alpha^R \mid A \longrightarrow \alpha \in P\}.$$

Dann gilt $L(G^R) = L(G)^R$.

Nun zum Beweis der Aussage a):

$$
\begin{array}{ll}
 & L \in \mathrm{CFL}(\Sigma) \\
\curvearrowright & \exists G \in \mathrm{CFG}(\Sigma): \ L(G) = L \\
\curvearrowright & \exists G \in \mathrm{CFG}(\Sigma): \ L(G^R) = L^R \\
\curvearrowright & \exists G \in \mathrm{CFG}(\Sigma): \ L(G) = L^R \\
\curvearrowright & L^R \in \mathrm{CFL}(\Sigma)
\end{array}
$$

b) *Gegenbeispiel:* Für $\Sigma = \{a, b\}$ sei $G = \langle N, \Sigma, P, S \rangle \in \mathrm{CFG}(\Sigma)$ mit $P \subseteq N \times (\Sigma^*(N \cup \{\varepsilon\}) \cup N\Sigma^*)$ gegeben durch

$$
\begin{array}{lcl}
S & \longrightarrow & aB \mid \varepsilon \\
B & \longrightarrow & Sb
\end{array}
$$

Es gilt $L(G) = \{a^n b^n \mid n \in \mathbb{N}\} \notin \mathrm{RegL}(\Sigma)$.

Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 6

Aufgabe 1

a) Seien $\Sigma = \{a, b\}$, $N = \{S, A, B\}$ und $X = \{a, b, S, A, B\}$. Wir zeigen:

$$\begin{aligned} &\forall \alpha \in X^* N X^*, \beta_1, \beta_2 \in X^* : \\ &\qquad S \Longrightarrow_\ell^+ \alpha \Longrightarrow_\ell \beta_1 \text{ und } \alpha \Longrightarrow_\ell \beta_2 \text{ und } \beta_1 \neq \beta_2 \\ &\qquad \curvearrowright \ \nexists w \in \Sigma^* : \beta_1 \Longrightarrow^* w \text{ und } \beta_2 \Longrightarrow^* w \end{aligned}$$

Daraus folgt unmittelbar die Eindeutigkeit von G.

Wir zeigen zunächst leicht (etwa durch Induktion):

$$\begin{aligned} &\forall \alpha \in X^* N X^* : \\ &\qquad S \Longrightarrow_\ell^+ \alpha \\ &\qquad \curvearrowright \ \alpha \in [\![a^* AbB]\!] \text{ oder } \alpha \in [\![a^* b (a \vee b)^* B]\!]. \end{aligned}$$

Sei nun $\alpha = a^n AbB$ für ein $n \in \mathbb{N}$. Von α ausgehend, gibt es genau die beiden Linksableitungsschritte $\alpha \Longrightarrow_\ell a^n aAbB$ und $\alpha \Longrightarrow_\ell a^n bB$. Natürlich existiert kein $w \in \Sigma$, so daß $a^n aAbB \Longrightarrow^* w$ und $a^n bB \Longrightarrow^* w$.

Sei andererseits $\alpha = a^n buB$, $n \in \mathbb{N}$, $u \in \{a, b\}^*$. Nun gibt es gerade die Linksableitungsschritte $\alpha \Longrightarrow_\ell a^n buaB$, $\alpha \Longrightarrow_\ell a^n bubB$ und $\alpha \Longrightarrow_\ell a^n bu$. Wieder existiert kein $w \in \Sigma$, so daß $a^n buaB \Longrightarrow^* w$ und $a^n bubB \Longrightarrow^* w$. Außerdem gilt weder $a^n buaB \Longrightarrow^* a^n bu$ noch $a^n bubB \Longrightarrow^* a^n bu$.

b) Die Grammatik $G' \in \mathrm{CFG}(\{a, b\})$ sei gegeben wie folgt:

$$\begin{aligned} S &\longrightarrow AbB \mid b \\ A &\longrightarrow aA \mid \varepsilon \\ B &\longrightarrow aB \mid bB \mid \varepsilon \end{aligned}$$

Das Wort b besitzt dann die beiden verschiedenen Linksableitungen

$$\begin{aligned} &S \Longrightarrow_{G'} AbB \Longrightarrow_{G'} bB \Longrightarrow_{G'} b \text{ und} \\ &S \Longrightarrow_{G'} b, \end{aligned}$$

G' ist also mehrdeutig. Aber es gilt $L(G') = L(G)$.

Aufgabe 2

Sei $L \in \mathrm{RegL}(\Sigma)$. Dann existiert ein DFA $\mathfrak{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ mit $L(\mathfrak{A}) = L$. Wir konstruieren der Vorlesung entsprechend die rechtslineare Grammatik $G = \langle Q, \Sigma, P, q_0 \rangle$, wobei

$$P = \{q \longrightarrow ap \mid \delta(q, a) = p\} \cup \{q \longrightarrow \varepsilon \mid q \in F\}.$$

Gemäß der Vorlesung ist $L(G) = L(\mathfrak{A})$ (dies wurde allerdings nicht bewiesen).

Behauptung: Zu jedem $w \in L(G)$ gibt es genau eine Ableitung.

Angenommen, zu einem Wort $w \in L(G)$ existieren (mindestens) die beiden Ableitungen

$$\begin{aligned} A_1 &= q_0 \Longrightarrow^* \alpha \Longrightarrow \beta_1 \Longrightarrow^* w \text{ und} \\ A_2 &= q_0 \Longrightarrow^* \alpha \Longrightarrow \beta_2 \Longrightarrow^* w, \end{aligned}$$

wobei die Teilableitungen $q_0 \Longrightarrow^* \alpha$ gleich seien und $\beta_1 \neq \beta_2$.

Es gilt $q_0 \Longrightarrow^n \alpha \Longrightarrow \beta \curvearrowright \alpha \in \Sigma^n Q$ (durch Induktion über n beweisbar). Damit folgt:

$$\begin{aligned} A_1 &= q_0 \Longrightarrow^* vq \Longrightarrow vaq' \Longrightarrow^* vau = w \\ A_2 &= q_0 \Longrightarrow^* vq \Longrightarrow vaq'' \Longrightarrow^* vau = w \end{aligned}$$

Also ist $q \longrightarrow aq' \in P$ und $q \longrightarrow aq'' \in P$. Gemäß Konstruktion von P ist $q' = \delta(q, a) = q''$ und somit $\beta_1 = \beta_2$ im Widerspruch zur Annahme.

Die zu jedem $w \in L(G)$ eindeutig gegebene Ableitung ist natürlich auch die eindeutige Rechts- bzw. Linksableitung.

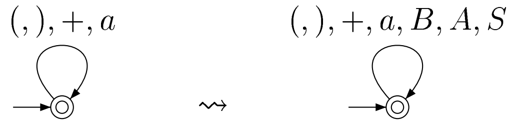
Also ist G eindeutig und damit auch L.

Aufgabe 3

Sei $\Sigma = \{(,), +, a\}$, und sei $G \in \mathrm{CFG}(\Sigma)$ gegeben wie folgt:

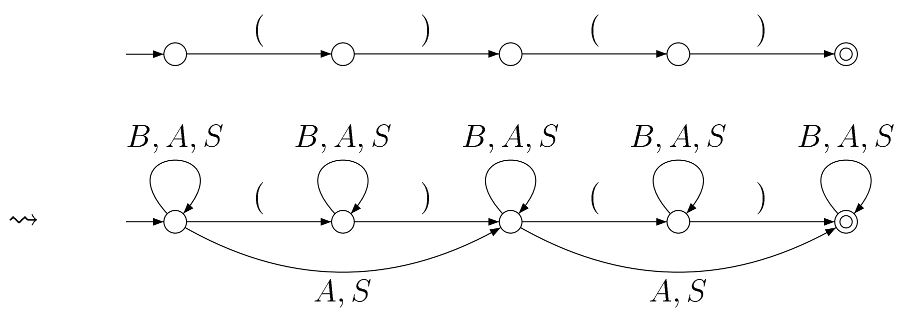
$$\begin{array}{lcl} S & \longrightarrow & S + A \mid A \\ A & \longrightarrow & (S) \mid B \\ B & \longrightarrow & aB \mid \varepsilon \end{array}$$

a) Konstruiere $\mathfrak{A} \in \mathrm{NFA}(\Sigma)$ mit $L(\mathfrak{A}) = Pre^*_G(\Sigma^*)$ wie folgt:



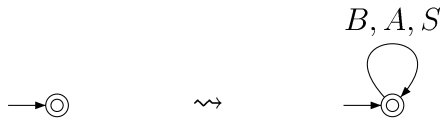
Da offensichtlich $S \in L(\mathfrak{A})$ und damit $S \in Pre^*_G(\Sigma^*)$, gilt $L(G) \neq \emptyset$.

b) Konstruiere $\mathfrak{A} \in \mathrm{NFA}(\Sigma)$ mit $L(\mathfrak{A}) = Pre^*_G(\{()()\})$ wie folgt:



Da $S \notin L(\mathfrak{A})$ und damit $S \notin Pre^*_G(\{()()\})$, gilt $()() \notin L(G)$.

c) Konstruiere $\mathfrak{A} \in \mathrm{NFA}(\Sigma)$ mit $L(\mathfrak{A}) = Pre^*_G(\{\varepsilon\})$ wie folgt:

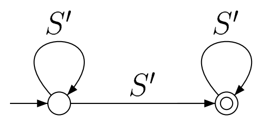


Wir erhalten $N_\varepsilon = \{B, A, S\}$. Sei also G' gegeben wie folgt:

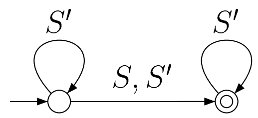
$$\begin{array}{lcl} S' & \longrightarrow & S \mid \varepsilon \\ S & \longrightarrow & S + A \mid +A \mid S+ \mid + \mid A \\ A & \longrightarrow & (S) \mid () \mid B \\ B & \longrightarrow & aB \mid a \end{array}$$

d) Die Vorgängerabschluß-Automaten zu $Pre^*_{G'}(\{S'\})$, $Pre^*_{G'}(\{S\})$, $Pre^*_{G'}(\{A\})$ und $Pre^*_{G'}(\{B\})$ lauten wie folgt:

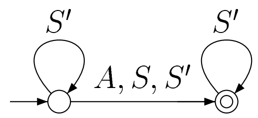
$Pre^*_{G'}(\{S'\})$:



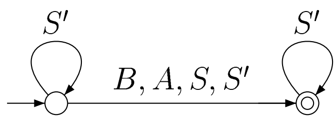
$Pre^*_{G'}(\{S\})$:



$Pre^*_{G'}(\{A\})$:



$Pre^*_{G'}(\{B\})$:



Also gelten $Pre^*_{G'}(\{S'\}) = \{S'\}$, $Pre^*_{G'}(\{S\}) = \{S, S'\}$, $Pre^*_{G'}(\{A\}) = \{A, S, S'\}$ und $Pre^*_{G'}(\{B\}) = \{B, A, S, S'\}$.

Demnach hat G'' die folgende Form:

$$\begin{array}{lcl} S' & \longrightarrow & \varepsilon \mid S + A \mid +A \mid S+ \mid + \mid (S) \mid () \mid aB \mid a \\ S & \longrightarrow & S + A \mid +A \mid S+ \mid + \mid (S) \mid () \mid aB \mid a \\ A & \longrightarrow & (S) \mid () \mid aB \mid a \\ B & \longrightarrow & aB \mid a \end{array}$$

e) G''' in Chomsky-Normalform sei nun nach dem Zwischenschritt

$$\begin{array}{lll}
S' & \longrightarrow & \varepsilon \mid SC_3A \mid C_3A \mid SC_3 \mid + \mid C_1SC_2 \mid C_1C_2 \mid C_4B \mid a \\
S & \longrightarrow & SC_3A \mid C_3A \mid SC_3 \mid + \mid C_1SC_2 \mid C_1C_2 \mid C_4B \mid a \\
A & \longrightarrow & C_1SC_2 \mid C_1C_2 \mid C_4B \mid a \\
B & \longrightarrow & C_4B \mid a \\
C_1 & \longrightarrow (& \\
C_2 & \longrightarrow) & \\
C_3 & \longrightarrow + & \\
C_4 & \longrightarrow a &
\end{array}$$

gegeben durch

$$\begin{array}{lll}
S' & \longrightarrow & \varepsilon \mid SD_1 \mid C_3A \mid SC_3 \mid + \mid C_1D_2 \mid C_1C_2 \mid C_4B \mid a \\
S & \longrightarrow & SD_1 \mid C_3A \mid SC_3 \mid + \mid C_1D_2 \mid C_1C_2 \mid C_4B \mid a \\
A & \longrightarrow & C_1D_2 \mid C_1C_2 \mid C_4B \mid a \\
B & \longrightarrow & C_4B \mid a \\
C_1 & \longrightarrow & (\\
C_2 & \longrightarrow &) \\
C_3 & \longrightarrow & + \\
C_4 & \longrightarrow & a \\
D_1 & \longrightarrow & C_3A \\
D_2 & \longrightarrow & SC_2
\end{array}$$

Aufgabe 4

a) Angenommen, L_1 sei kontextfrei. Dann existiert ein Pumping-Index $k \geq 1$ mit den Eigenschaften des Pumping-Lemmas. Sei $z = a^k b^{2k} a^{2k} b^k \in L_1$. Für alle relevanten Zerlegungen erhalten wir nach Aufpumpen von z gemäß Pumping-Lemma höchstens Wörter folgender Form:

Fall 1: $z' = a^{k+m} b^{2k} a^{2k} b^k \notin L_1$ für $m > 0$.

Fall 2: $z' = a^k b^{2k+m} a^{2k} b^k \notin L_1$ für $m > 0$.

Fall 3: $z' = a^k b^{2k} a^{2k+m} b^k \notin L_1$ für $m > 0$.

Fall 4: $z' = a^k b^{2k} a^{2k} b^{k+m} \notin L_1$ für $m > 0$.

Fall 5: $z' = a^{k+m} b^{2k+n} a^{2k} b^k \notin L_1$ für $m, n > 0$.

Fall 6: $z' = a^k b^{2k+m} a^{2k+n} b^k \notin L_1$ für $m, n > 0$.

Fall 7: $z' = a^k b^{2k} a^{2k+m} b^{k+n} \notin L_1$ für $m, n > 0$.

Fall 8: $z' = a^p (a^m b^n)^i b^{q+k} a^{2k} b^k \notin L_1$ für $m, n > 0,\ i = 2$.

Fall 9: $z' = a^k b^{k+p} (b^m a^n)^i a^{k+q} b^k \notin L_1$ für $m, n > 0,\ i = 2$.

Fall 10: $z' = a^k b^{2k} a^{k+p} (a^m b^n)^i b^q \notin L_1$ für $m, n > 0,\ i = 2$.

In allen Fällen läßt sich leicht begründen, warum z' nicht in L_1 enthalten sein kann im Widerspruch zur Annahme, L_1 sei kontextfrei. Betrachten wir z. B. Fall 1, und nehmen wir an, z' sei in L_1 enthalten, habe also die Form $z' = ww^R w$, $w \in \{a,b\}^*$. Da $|z'| > |z|$, gilt $w = ub^k$ für ein $u \in \{a,b\}^*$ und damit $a^{k+m}b^k \in \mathit{Prf}(w)$. D. h. $|z'| \geq 3(2k+m)$. Widerspruch, da $|z'| = 6k + m < 3(2k+m)$.

Wir schließen also, daß L_1 nicht kontextfrei ist.

b) Wähle $k = 2$, und sei $z \in L_2$ mit $|z| \geq 2$.

Fall 1: $z = a^m b^{n+1} c^{m+n+1}$ $(m, n \in \mathbb{N})$
Wähle $u = a^m b^n$, $v = b$, $w = \varepsilon$, $x = c$, $y = c^{m+n}$.
Dann gelten

- $|vwx| = |bc| = 2 \leq 2$,
- $vx = bc \neq \varepsilon$ und
- $uv^i wx^i y = a^m b^n b^i c^i c^{m+n} = a^m b^{n+i} c^{m+n+i} \in L_2$ für alle $i \in \mathbb{N}$.

Fall 2: $z = a^{m+1} c^{m+1}$ $(m \in \mathbb{N})$
Wähle $u = a^m$, $v = a$, $w = \varepsilon$, $x = c$, $y = c^m$. Dann gelten

- $|vwx| = |ac| = 2 \leq 2$,
- $vx = ac \neq \varepsilon$ und
- $uv^i wx^i y = a^m a^i c^i c^m = a^{m+i} c^{m+i} \in L_2$ für alle $i \in \mathbb{N}$.

c) Nein. Das Pumping-Lemma für kontextfreie (reguläre) Sprachen ist nur für den Nachweis geeignet, daß eine Sprache nicht kontextfrei (regulär) ist.

Lösungsvorschlag ***Automatentheorie und Formale Sprachen*** – Blatt 7

Aufgabe 1

Wir zeigen:

$$\mathcal{L}(\Sigma, \mathrm{PDA}, \mathrm{F}) \overset{\text{a)}}{\subseteq} \mathcal{L}(\Sigma, \mathrm{PDA}, \varepsilon) \overset{\text{b)}}{\subseteq} \mathcal{L}(\Sigma, \mathrm{PDA}, \mathrm{F}, \varepsilon) \overset{\text{c)}}{\subseteq} \mathcal{L}(\Sigma, \mathrm{PDA}, \mathrm{F})$$

a) Sei $L \in \mathcal{L}(\Sigma, \mathrm{PDA}, \mathrm{F})$. Wir zeigen, daß dann auch $L \in \mathcal{L}(\Sigma, \mathrm{PDA}, \varepsilon)$. Nun existiert ein Automat $\mathfrak{A} \in \mathrm{PDA}(\Sigma)$, so daß $L(\mathfrak{A}, \mathrm{F}) = L$. Zu zeigen ist, daß auch ein Automat $\mathfrak{A}' \in \mathrm{PDA}(\Sigma)$ existiert mit $L(\mathfrak{A}', \varepsilon) = L$. Sei also $\mathfrak{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle \in \mathrm{PDA}(\Sigma)$.

Wir konstruieren

$$\mathfrak{A}' = \langle Q', \Sigma, \Gamma', \delta', q_0', Z_0', \emptyset\rangle \in \mathrm{PDA}(\Sigma), \text{ wobei}$$

$$\begin{aligned}
Q' &= Q \uplus \{q_0', q_f'\}\\
\Gamma' &= \Gamma \uplus \{Z_0'\}\\
&\delta' : Q' \times \Sigma_\varepsilon \times \Gamma' \to \mathfrak{P}_f(Q' \times \Gamma'^*)
\end{aligned}$$

$$\delta'(q, X, Z) = \begin{cases} \delta(q, X, Z) & \text{falls } q \in Q,\ X \in \Sigma_\varepsilon,\ Z \in \Gamma \\ \{(q_0, Z_0 Z_0')\} & \text{falls } q = q_0',\ X = \varepsilon,\ Z = Z_0' \\ \{(q_f', Z)\} & \text{falls } q \in F,\ Z \in \Gamma \\ \{(q_f', \varepsilon)\} & \text{falls } q = q_f',\ X = \varepsilon,\ Z \in \Gamma' \\ \emptyset & \text{sonst} \end{cases}$$

kurz:

$$\begin{aligned}
\delta' = \delta\ &\uplus \{(q_0', \varepsilon, Z_0') \longrightarrow (q_0, Z_0 Z_0')\}\\
&\uplus \{(q, \varepsilon, Z) \longrightarrow (q_f', Z) \mid q \in F, Z \in \Gamma\}\\
&\uplus \{(q_f', \varepsilon, Z) \longrightarrow (q_f', \varepsilon) \mid Z \in \Gamma'\}
\end{aligned}$$

Dann gilt $L(\mathfrak{A}', \varepsilon) = L(\mathfrak{A}, \mathrm{F})$.

Beweis:

$$\begin{aligned}
& w \in L(\mathfrak{A}, \mathrm{F})\\
\curvearrowleft\!\curvearrowright\ & \exists q \in F, \alpha \in \Gamma^* : \quad (q_0, w, Z_0) \vdash^*_{\mathfrak{A}} (q, \varepsilon, \alpha)\\
\curvearrowleft\!\curvearrowright\ & \exists q \in F, \alpha \in \Gamma^* : \quad (q_0, w, Z_0) \vdash^*_{\mathfrak{A}'} (q, \varepsilon, \alpha)\\
\curvearrowleft\!\curvearrowright\ & \exists q \in F, \alpha \in \Gamma^* : \quad (q_0, w, Z_0 Z_0') \vdash^*_{\mathfrak{A}'} (q, \varepsilon, \alpha Z_0')\\
\curvearrowleft\!\curvearrowright\ & \exists q \in F, \alpha \in \Gamma^* : \quad (q_0', w, Z_0') \vdash_{\mathfrak{A}'} (q_0, w, Z_0 Z_0') \vdash^*_{\mathfrak{A}'} (q, \varepsilon, \alpha Z_0') \vdash (q_f', \varepsilon, \alpha Z_0') \vdash^* (q_f', \varepsilon, \varepsilon)\\
\curvearrowleft\!\curvearrowright\ & w \in L(\mathfrak{A}', \varepsilon)
\end{aligned}$$

b) Sei $\mathfrak{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle \in \mathrm{PDA}(\Sigma)$. Wir konstruieren

$$\begin{aligned}
\mathfrak{A}' &= \langle Q \uplus \{q_0', q_f'\}, \Sigma, \Gamma \uplus \{Z_0'\}, \delta', q_0', Z_0', F'\rangle \in \mathrm{PDA}(\Sigma), \text{ wobei}\\
\delta' &= \delta \uplus \{(q_0', \varepsilon, Z_0') \longrightarrow (q_0, Z_0 Z_0')\} \uplus \{(q, \varepsilon, Z_0') \longrightarrow (q_f', \varepsilon) \mid q \in Q\}\\
F' &= \{q_f'\}
\end{aligned}$$

Dann gilt $L(\mathfrak{A}', \mathrm{F}, \varepsilon) = L(\mathfrak{A}, \varepsilon)$.

Beweis:

$$
\begin{array}{ll}
 & w \in L(\mathfrak{A}, \varepsilon) \\
\Leftrightarrow & \exists q \in Q: \quad (q_0, w, Z_0) \vdash_{\mathfrak{A}}^{*} (q, \varepsilon, \varepsilon) \\
\Leftrightarrow & \exists q \in Q: \quad (q_0, w, Z_0) \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, \varepsilon) \\
\Leftrightarrow & \exists q \in Q: \quad (q_0, w, Z_0 Z_0') \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, Z_0') \\
\Leftrightarrow & \exists q \in Q: \quad (q_0', w, Z_0') \vdash_{\mathfrak{A}'} (q_0, w, Z_0 Z_0') \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, Z_0') \vdash_{\mathfrak{A}'} (q_f', \varepsilon, \varepsilon) \\
\Leftrightarrow & w \in L(\mathfrak{A}', \mathrm{F}, \varepsilon)
\end{array}
$$

c) Sei $\mathfrak{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle \in \mathrm{PDA}(\Sigma)$. Wir konstruieren

$$
\begin{array}{rcl}
\mathfrak{A}' & = & \langle Q \uplus \{q_0', q_f'\}, \Sigma, \Gamma \uplus \{Z_0'\}, \delta', q_0', Z_0', F' \rangle \in \mathrm{PDA}(\Sigma), \text{ wobei} \\
\delta' & = & \delta \uplus \{(q_0', \varepsilon, Z_0') \longrightarrow (q_0, Z_0 Z_0')\} \uplus \{(q, \varepsilon, Z_0') \longrightarrow (q_f', Z_0') \mid q \in F\} \\
F' & = & \{q_f'\}
\end{array}
$$

Dann gilt $L(\mathfrak{A}', \mathrm{F}) = L(\mathfrak{A}, \mathrm{F}, \varepsilon)$.

Beweis:

$$
\begin{array}{ll}
 & w \in L(\mathfrak{A}, \mathrm{F}, \varepsilon) \\
\Leftrightarrow & \exists q \in F: \quad (q_0, w, Z_0) \vdash_{\mathfrak{A}}^{*} (q, \varepsilon, \varepsilon) \\
\Leftrightarrow & \exists q \in F: \quad (q_0, w, Z_0) \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, \varepsilon) \\
\Leftrightarrow & \exists q \in F: \quad (q_0, w, Z_0 Z_0') \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, Z_0') \\
\Leftrightarrow & \exists q \in F: \quad (q_0', w, Z_0') \vdash_{\mathfrak{A}'} (q_0, w, Z_0 Z_0') \vdash_{\mathfrak{A}'}^{*} (q, \varepsilon, Z_0') \vdash_{\mathfrak{A}'} (q_f', \varepsilon, Z_0') \\
\Leftrightarrow & w \in L(\mathfrak{A}', \mathrm{F})
\end{array}
$$

Aufgabe 2

a) Wir konstruieren

$$
\mathfrak{A}_G = \langle \{q\}, \{a, b, c, d\}, \{a, b, c, d, S, A, B, C\}, \delta, q, S, \emptyset \rangle \in \mathrm{PDA}(\{a, b, c, d\}),
$$

wobei δ wie folgt gegeben ist:

$$
\begin{array}{rcl}
\delta(q, a, a) & = & \{(q, \varepsilon)\} \\
\delta(q, b, b) & = & \{(q, \varepsilon)\} \\
\delta(q, c, c) & = & \{(q, \varepsilon)\} \\
\delta(q, d, d) & = & \{(q, \varepsilon)\} \\
\delta(q, e, X) & = & \emptyset, \text{ falls } e \neq X \\
\delta(q, \varepsilon, S) & = & \{(q, aSd), (q, A), (q, C)\} \\
\delta(q, \varepsilon, A) & = & \{(q, aAc), (q, B)\} \\
\delta(q, \varepsilon, B) & = & \{(q, bBc), (q, \varepsilon)\} \\
\delta(q, \varepsilon, C) & = & \{(q, bCd), (q, B)\} \\
\delta(q, \varepsilon, X) & = & \emptyset, \text{ falls } X \in \{a, b, c, d\}
\end{array}
$$

b)

$$
\begin{array}{rlrlrrr}
 & \text{Ableitung in } G\text{:} & & \multicolumn{4}{l}{\text{Erkennung durch } \mathfrak{A}_G\text{:}} \\
 & & & (q, & abbcdd, & S) \\
 & & \vdash & (q, & abbcdd, & aSd) \\
 & & \vdash & (q, & bbcdd, & Sd) \\
 & S & \vdash & (q, & bbcdd, & Cd) \\
\Longrightarrow & aSd & \vdash & (q, & bbcdd, & bCdd) \\
\Longrightarrow & aCd & \vdash & (q, & bcdd, & Cdd) \\
\Longrightarrow & abCdd & \vdash & (q, & bcdd, & Bdd) \\
\Longrightarrow & abBdd & \vdash & (q, & bcdd, & bBcdd) \\
\Longrightarrow & abbBcdd & \vdash & (q, & cdd, & Bcdd) \\
\Longrightarrow & abbcdd & \vdash & (q, & cdd, & cdd) \\
 & & \vdash & (q, & dd, & dd) \\
 & & \vdash & (q, & d, & d) \\
 & & \vdash & (q, & \varepsilon, & \varepsilon)
\end{array}
$$

Aufgabe 3

Seien $\mathfrak{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle$ ein Kellerautomat mit beschränktem Keller und $n \in \mathbb{N}$ seine maximale Kellergröße.

Demnach gibt es nur

$$\sum_{i=0}^{n} |\Gamma|^i$$

mögliche (erreichbare) Kellerinschriften, also endlich viele. Diese können in der endlichen Kontrolle eines endlichen Automaten codiert werden.

Wir konstruieren die nicht-deterministischen endlichen Automaten

$$\mathfrak{A}'_i = \langle Q', \Sigma, \delta', q'_0, F'_i\rangle,$$

$i \in \{1, 2, 3\}$, wie folgt:

$$
\begin{array}{rcl}
Q' & = & Q \times \Gamma^{\leq n}, \quad \text{wobei } \Gamma^{\leq n} := \bigcup_{i=0}^{n} \Gamma^i \\
\delta'((q, Z\gamma), a) & = & \{(q', Z_m \dots Z_1\gamma) \in Q' \mid (q', Z_m \dots Z_1) \in \delta(q, a, Z)\}, \quad a \in \Sigma_\varepsilon \\
q'_0 & = & (q_0, Z_0) \\
F'_1 & = & \{(q, \gamma) \in Q' \mid q \in F\} \\
F'_2 & = & \{(q, \varepsilon) \in Q'\} \\
F'_3 & = & \{(q, \varepsilon) \in Q' \mid q \in F\}
\end{array}
$$

Nach Konstruktion gilt

$$(q, uv, \gamma) \vdash^*_{\mathfrak{A}} (q', v, \gamma') \curvearrowleft\!\!\curvearrowright (q', \gamma') \in \overline{\delta'}((q, \gamma), u), \;\; \gamma, \gamma' \in \Gamma^{\leq n}.$$

Es folgen die Sprachgleichheiten

$$
\begin{array}{rcl}
L(\mathfrak{A}'_1) & = & L(\mathfrak{A}, \mathrm{F}), \\
L(\mathfrak{A}'_2) & = & L(\mathfrak{A}, \varepsilon) \text{ und} \\
L(\mathfrak{A}'_3) & = & L(\mathfrak{A}, \mathrm{F}, \varepsilon).
\end{array}
$$

$L(\mathfrak{A}, \mathrm{F})$, $L(\mathfrak{A}, \varepsilon)$ und $L(\mathfrak{A}, \mathrm{F}, \varepsilon)$ sind also regulär.

Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 8

Aufgabe 1

Angenommen, es existierte $\mathfrak{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle \in \mathrm{DPDA}(\Sigma)$, so daß $L(\mathfrak{A}, \varepsilon)$ nicht die Präfix-Eigenschaft besitzt. Dann existieren $u, v \in L(\mathfrak{A}, \varepsilon)$, $u \neq v$, mit $u \in Prf(v)$, d. h. $\exists a \in \Sigma,\ w \in \Sigma^* :\ v = uaw$.

Es gilt

$$(q_0, u, Z_0) \vdash_{\mathfrak{A}}^{*} (q, \varepsilon, \varepsilon) \text{ für genau ein } q \in Q$$

und ferner

$$v \in L(\mathfrak{A}, \varepsilon) \curvearrowright (q, aw, \varepsilon) \vdash_{\mathfrak{A}}^{+} (q', \varepsilon, \varepsilon),\ q' \in Q.$$

Widerspruch zur Annahme, da $(q, aw, \varepsilon) \nvdash_{\mathfrak{A}}$.

Aufgabe 2

Sei $\Sigma = \{a, b, c\}$. Angenommen, $L \in \mathcal{L}(\Sigma, \mathrm{DPDA})$.

$$\begin{array}{ll} & L \in \mathcal{L}(\Sigma, \mathrm{DPDA}) \\ \curvearrowright & (\mathcal{L}(\Sigma, \mathrm{DPDA}) \text{ ist unter Komplement abgeschlossen}) \\ & \overline{L} = \{w \in \Sigma^* \mid |w|_a = |w|_b = |w|_c \text{ oder } w \notin [\![a^*b^*c^*]\!]\} \in \mathcal{L}(\Sigma, \mathrm{DPDA}) \\ \curvearrowright & \overline{L} \in \mathcal{L}(\Sigma, \mathrm{PDA}) \\ \curvearrowright & (\mathcal{L}(\Sigma, \mathrm{PDA}) \text{ ist unter Schnitt mit regulären Sprachen abgeschlossen}) \\ & \overline{L} \cap [\![a^*b^*c^*]\!] = \{a^n b^n c^n \mid n \in \mathbb{N}\} \in \mathcal{L}(\Sigma, \mathrm{PDA}) \end{array}$$

Widerspruch zur Annahme, da $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ nicht kontextfrei ist.

Aufgabe 3

a)

$i \setminus j$	1	2	3	4	5
1	$\{B\}$	$\{S, A\}$	$\emptyset$	$\{S, A, C\}$	$\{S, C\}$
2		$\{A, C\}$	$\{B\}$	$\{S, C, A\}$	$\{S, C\}$
3			$\{A, C\}$	$\{B\}$	$\{B\}$
4				$\{A, C\}$	$\{S, C\}$
5					$\{B\}$
	b	a	a	a	b

$$\rightsquigarrow baaab \in L(G)$$

b)

$i \setminus j$	1	2	3	4	5
1	$\{A, C\}$	$\{B\}$	$\{B\}$	$\{S, C, A\}$	$\{S, C\}$
2		$\{A, C\}$	$\{S, C\}$	$\{B\}$	$\{B\}$
3			$\{B\}$	$\{S, A\}$	$\{S, C\}$
4				$\{A, C\}$	$\{S, C\}$
5					$\{B\}$
	a	a	b	a	b

$$\rightsquigarrow aabab \in L(G)$$

Aufgabe 4

$$\begin{array}{lcl}
S & \longrightarrow & V := E \mid \\
 & & \text{if } E \text{ then } S \text{ else } S \mid \\
 & & \text{if } E \text{ then } S\ A \mid \\
 & & \text{begin } S\ B \text{ end} \mid \\
 & & \text{repeat } S\ B \text{ until } E \\
E & \longrightarrow & V \mid \text{true} \mid \text{false} \\
V & \longrightarrow & \text{x} \mid \text{y} \mid \text{z} \\
A & \longrightarrow & L\ A \mid \varepsilon \\
B & \longrightarrow & ;\ S\ B \mid \varepsilon
\end{array}$$

Lösungsvorschlag *Automatentheorie und Formale Sprachen* – Blatt 9

Aufgabe 1

a) G ist vom Typ 2, da $P \subseteq N \times X^*$.
G ist nicht vom Typ 3, da $B \longrightarrow aD \in P$ und $C \longrightarrow Ca \in P$.

b) Sei die Typ-3-Grammatik G' gegeben wie folgt:

$$\begin{array}{lcl} S & \longrightarrow & abB \\ B & \longrightarrow & C \mid aD \\ C & \longrightarrow & aC \mid \varepsilon \\ D & \longrightarrow & bD \mid \varepsilon \end{array}$$

Es gilt $L(G') = L(G)$. Also ist $L(G)$ vom Typ 3.

Aufgabe 2

a) Normierung von G:

1) Sei G' gegeben durch

$$\begin{array}{lcl} S & \longrightarrow & A_aSBC \mid \varepsilon \\ A_aB & \longrightarrow & A_aA_b \\ CB & \longrightarrow & BC \\ A_bB & \longrightarrow & A_bA_b \\ A_bC & \longrightarrow & A_bA_c \\ A_cC & \longrightarrow & A_cA_c \\ A_a & \longrightarrow & a \\ A_b & \longrightarrow & b \\ A_c & \longrightarrow & c \end{array}$$

2) Ausgehend von G' konstruieren wir G'' wie folgt:

$$\begin{array}{lcl} S & \longrightarrow & A_aSBC \mid \varepsilon \\ A_aB & \longrightarrow & A_aA_b \\ CB & \longrightarrow & C'B \\ C'B & \longrightarrow & C'B' \\ C'B' & \longrightarrow & BB' \\ BB' & \longrightarrow & BC \\ A_bB & \longrightarrow & A_bA_b \\ A_bC & \longrightarrow & A_bA_c \\ A_cC & \longrightarrow & A_cA_c \\ A_a & \longrightarrow & a \\ A_b & \longrightarrow & b \\ A_c & \longrightarrow & c \end{array}$$

G'' ist normiert, und es gilt $L(G'') = L(G)$.

b) Seien

- $\Sigma = \{a\}$,
- $\Sigma_a = \{a, b, c\}$,
- $G_1 : \begin{array}{lcl} S' & \longrightarrow & B_a S' \mid \varepsilon \\ B_a & \longrightarrow & a \end{array}$

  eine normierte Grammatik mit $L(G_1) = [\![a^*]\!]$ und
- $\varphi : \mathfrak{P}(\Sigma^*) \to \mathfrak{P}(\Sigma_a^*)$ eine Substitutionsabbildung gemäß $\varphi(a) = L(G'')$.

Wir konstruieren nun $\overline{G}$ mit $L(\overline{G}) = \varphi(L(G_1)) = L(G'')^* = L(G)^*$ wie folgt:

$$\begin{array}{lcl}
\overline{S} & \longrightarrow & \widehat{C} S' \\
\widehat{C} & \longrightarrow & \varepsilon \\
\left.\begin{array}{l} S' \\ S' \\ B_a \end{array}\right. & \begin{array}{c} \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{array} & \left.\begin{array}{l} B_a S' \\ \varepsilon \\ A'_a \end{array}\right\} P_0 \\
\widehat{C} A'_a & \longrightarrow & \widehat{C} S \quad \} P_1 \\
\left.\begin{array}{l} \widehat{C} a \\ \widehat{C} b \\ \widehat{C} c \end{array}\right. & \begin{array}{c} \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{array} & \left.\begin{array}{l} a\widehat{C} \\ b\widehat{C} \\ c\widehat{C} \end{array}\right\} P_2 \\
S & \longrightarrow & \left.\begin{array}{l} A_a S B C \\ \vdots \end{array}\right\} G''
\end{array}$$

Aufgabe 3

a) Sei $G = \langle N, \Sigma, P, S \rangle$ eine Grammatik vom Typ 0. Wir konstruieren $G' = \langle N', \Sigma, P', S' \rangle$ mit

$$L(G') = \{v \mid w \in L(G),\ v \text{ Permutation von } w\}$$

wie folgt:

- $N' = N \cup\!\!\!\cdot\ \{C_1, C_2, C_3, S'\}$
- $P' = P \ \cup\!\!\!\cdot\ \{S' \longrightarrow C_1 C_2 S C_3\}$
  $\cup\!\!\!\cdot\ \{C_2 a \longrightarrow a C_2 \mid a \in \Sigma\}$
  $\cup\!\!\!\cdot\ \{C_1 a \longrightarrow a C_1 \mid a \in \Sigma\}$
  $\cup\!\!\!\cdot\ \{a C_1 \longrightarrow C_1 a \mid a \in \Sigma\}$
  $\cup\!\!\!\cdot\ \{a C_1 b \longrightarrow b C_1 a \mid a, b \in \Sigma\}$
  $\cup\!\!\!\cdot\ \{C_1 C_2 C_3 \longrightarrow \varepsilon\}$

Idee: Das Kontrollsymbol C_2 markiert einen Bereich bestehend aus Terminalsymbolen, in dem sich C_1 beliebig aufhalten und benachbarte Terminalsymbole vertauschen kann. Erhalten wir schließlich in einer Ableitung $C_1 C_2 C_3$ am Ende einer Satzform, so können die Kontrollsymbole gelöscht werden, und übrig bleibt eine Folge von Terminalsymbolen.

b) G' ist in jedem Fall *keine* Typ-1-Grammatik, da die Definition der Typ-1-Grammatiken Regeln etwa der Form $C_1 C_2 C_3 \longrightarrow \varepsilon$ ausschließt.

Lösungsvorschlag *Automatentheorie und Formale Sprachen*
Blatt 10

Aufgabe 1

Idee: Das Eingabewort wird, falls möglich, in zwei Hälften geteilt, woraufhin die linke Hälfte gelöscht wird. Beim Teilen werden die a's der linken Hälfte durch b's und simultan diejenigen der rechten Hälfte durch c's ersetzt. Das Löschen der linken Hälfte erfolgt einfach durch Ersetzen des am weitesten rechts stehenden b's durch $\Box$. Danach werden die c's wieder durch a's ersetzt, und das Teilen beginnt von neuem. Akzeptiert wird die Eingabe schließlich, sobald noch genau ein a übrig geblieben ist.

Sei nun $\mathfrak{A} = \langle \{q_0, q_1, \ldots, q_7, q_f\}, \{a\}, \{a, b, c, \Box\}, q_0, \Box, \{q_f\}, \delta \rangle \in \mathrm{TM}(\{a\})$ gegeben durch:

q_0	a	q_1	b	R	Markiere linken Rand
q_1	$\Box$	q_f	$\Box$	N	Wort wird akzeptiert
q_1	a	q_2	a	N	Ein weiteres a vorhanden
q_2	a	q_2	a	R	Laufe bis zum rechten Rand
q_2	$\Box$	q_3	$\Box$	L	Rechte Seite mit $\Box$ erreicht
q_2	c	q_3	c	L	Rechte Seite mit c erreicht
q_3	a	q_4	c	L	Markiere rechte Seite
q_4	a	q_5	a	N	Noch nicht alles markiert
q_4	b	q_6	$\Box$	R	Alles markiert $\Rightarrow$ lösche linke Seite
q_5	a	q_5	a	L	Laufe bis zum linken Rand
q_5	b	q_0	b	R	Linken Rand gefunden
q_6	c	q_6	a	R	Aus c's werden a's
q_6	$\Box$	q_7	$\Box$	L	Alle c's sind in a's umgewandelt worden
q_7	a	q_7	a	L	Laufe zurück zu erstem a
q_7	$\Box$	q_0	$\Box$	R	Kopf ist wieder auf erstem a

Aufgabe 2

Sei $\mathfrak{A} = \langle Q, \Sigma, \delta, q_0, F \rangle \in \mathrm{DFA}(\Sigma)$, und sei $\mathfrak{A}' = \langle Q \cup \{q_f\}, \Sigma, \Sigma \cup \{\Box\}, q_0, \Box, \{q_f\}, \delta' \rangle \in \mathrm{TM}(\Sigma)$ gegeben durch:

$$\delta'(q,a) = \{(\delta(q,a), a, \mathrm{R})\} \qquad \text{für alle } a \in \Sigma$$

$$\delta'(q,\Box) = \begin{cases} \{(q_f, \Box, \mathrm{R})\} & \text{, falls } q \in F \\ \emptyset & \text{, falls } q \notin F \end{cases}$$

Der neue Endzustand ist notwendig, weil wir zur Erkennung sicherstellen müssen, daß das Eingabewort auf dem Turingband vollständig gelesen wurde. Denn in einem endlichen Automaten durchlaufen wir für eine Eingabe evtl. einen Endzustand, ohne daß das Wort schließlich akzeptiert wird. Die Turingmaschine dagegen akzeptiert, sobald sie sich in einem Endzustand befindet.

Es gilt $L(\mathfrak{A}') = L(\mathfrak{A})$.

Beweis:

$$\begin{array}{rl} & w = a_1 a_2 \ldots a_n \in L(\mathfrak{A}) \\ \curvearrowright & \overline{\delta}(q_0, w) \in F \\ \curvearrowright & \exists q_1, \ldots, q_{n-1} \in Q, q_n \in F: \; \delta(q_0, a_1) = q_1, \delta(q_1, a_2) = q_2, \ldots, \delta(q_{n-1}, a_n) = q_n \\ \curvearrowright & \exists q_1, \ldots, q_{n-1} \in Q, q_n \in F: \; (q_0, \varepsilon, a_1, a_2 \ldots a_n) \vdash_{\mathfrak{A}} (q_1, a_1, a_2, a_3 \ldots a_n) \vdash_{\mathfrak{A}} \ldots \vdash_{\mathfrak{A}} \\ & (q_{n-1}, a_1 \ldots a_{n-1}, a_n, \varepsilon) \vdash_{\mathfrak{A}} (q_n, w, \Box, \varepsilon) \vdash_{\mathfrak{A}} (q_f, w\Box, \Box, \varepsilon) \\ \curvearrowright & w \in L(\mathfrak{A}') \end{array}$$

Aufgabe 3

Sei $\mathfrak{A} = \langle \{q_0, q_f\}, \{\$\}, \{\$, \Box\}, q_0, \Box, \{q_f\}, \delta \rangle \in \mathrm{TM}(\{\$\})$ gegeben durch:

$$\begin{aligned} \delta(q_0, \Box) &= \{(q_0, \Box, \mathrm{L}), (q_0, \Box, \mathrm{R})\} \\ \delta(q_0, \$) &= \{(q_f, \$, \mathrm{N})\} \\ \delta(q_f, \Box) &= \delta(q_f, \$) = \emptyset \end{aligned}$$

Idee: $\mathfrak{A}$ „rät“ nicht-deterministisch, wo sich das nicht-leere Eingabewort befindet.

Lösungsvorschlag *Automatentheorie und Formale Sprachen*
Blatt 11

Aufgabe 1

Idee: $\mathfrak{A}$ arbeitet nach dem Reißverschlußprinzip. Wir markieren auf dem Turingband jeweils durch das Symbol $|$ einen linken und einen rechten Rand und damit einen Bandbereich, den die Turingmaschine schon untersucht hat. Nun werden abwechselnd der rechte und der linke Rand nach außen immer um ein Feld verschoben und der bereits untersuchte Bereich beidseitig vergrößert.

Sei nun $\mathfrak{A} = \langle \{q_0, q_1, \ldots, q_5, q_f\}, \{\$\}, \{\$, \Box, |\}, q_0, \Box, \{q_f\}, \delta \rangle \in 1\text{-}d\mathrm{TM}$ gegeben durch:

q_0	\$	q_f	\$	N	\$ gefunden $\Rightarrow$ Wort wird akzeptiert
q_0	$\Box$	q_1	$\|$	L	Initialisiere rechte Randmarkierung
q_1	\$	q_f	\$	N	\$ gefunden $\Rightarrow$ Wort wird akzeptiert
q_1	$\Box$	q_2	$\|$	R	Initialisiere linke Randmarkierung
q_2	$\Box$	q_2	$\Box$	R	Laufe bis zum rechten Rand
q_2	$\|$	q_3	$\Box$	R	Rechten Rand erreicht
q_3	\$	q_f	\$	N	\$ gefunden $\Rightarrow$ Wort wird akzeptiert
q_3	$\Box$	q_4	$\|$	L	Setze neue rechte Randmarkierung
q_4	$\Box$	q_4	$\Box$	L	Laufe bis zum linken Rand
q_4	$\|$	q_5	$\Box$	L	Linken Rand erreicht
q_5	\$	q_f	\$	N	\$ gefunden $\Rightarrow$ Wort wird akzeptiert
q_5	$\Box$	q_2	$\|$	R	Setze neue linke Randmarkierung

Aufgabe 2

a) Sei $\mathfrak{A} = \langle Q, \Sigma, \Gamma, q_0, \Box, F, \delta \rangle \in \mathrm{TM}(\Sigma)$. Wir definieren eine Äquivalenzrelation $\sim \subseteq \mathit{Conf}(\mathfrak{A})^2$ wie folgt:

$$(q, \alpha, X, \beta) \sim (q', \alpha', X', \beta') \ :\curvearrowleft$$

i) $q = q'$
ii) $X = X'$
iii) $\alpha = \Box^n \alpha'$ oder $\alpha' = \Box^n \alpha$ für ein $n \in \mathbb{N}$
iv) $\beta = \beta' \Box^n$ oder $\beta' = \beta \Box^n$ für ein $n \in \mathbb{N}$

Beispiel: $(q, \Box\Box ab, \$, bc) \sim (q, \Box ab, \$, bc\Box)$

Es gilt offensichtlich:

$$\forall \kappa_1, \kappa_2, \kappa_1' \in \mathit{Conf}(\mathfrak{A}) : \quad \kappa_1 \vdash_{\mathfrak{A}} \kappa_2 \text{ und } \kappa_1 \sim \kappa_1' \curvearrowright \exists \kappa_2' \in \mathit{Conf}(\mathfrak{A}) : \ \kappa_1' \vdash_{\mathfrak{A}} \kappa_2'$$

Sei nun $w \in L(\mathfrak{A})$. Dann gilt:

$$\begin{aligned} & |\{[\kappa]_\sim \mid \kappa \in \mathit{Conf}(\mathfrak{A}),\ |\kappa| \leq s(|w|)\}| \\ = \ & |\{[\kappa]_\sim \mid \kappa \in \mathit{Conf}(\mathfrak{A}),\ |\kappa| = s(|w|)\}| \\ (*) \leq \ & |Q| \cdot s(|w|) \cdot |\Gamma|^{s(|w|)} \end{aligned}$$

Es existiert eine Berechnung $\gamma = (\kappa_0 = \kappa(w) \vdash_{\mathfrak{A}} \kappa_1 \vdash_{\mathfrak{A}} \ldots \vdash_{\mathfrak{A}} \kappa_n)$, $n \in \mathbb{N}$ und κ_n Endkonfiguration, mit $bv(\gamma) \leq s(|w|)$ und o.B.d.A. $\kappa_i \not\sim \kappa_j$, $i \neq j$. Wegen $(*)$ gilt also $n \leq |Q| \cdot s(|w|) \cdot |\Gamma|^{s(|w|)}$.

b) Sei $\mathfrak{A} = \langle Q, \Sigma, \Gamma, q_0, \square, F, \delta \rangle \in \mathrm{TM}(\Sigma)$ linear beschränkt und $p \in \mathbb{N} \backslash \{0\}$, so daß für alle $w \in L(\mathfrak{A}) \backslash \{\varepsilon\}$ gilt $bv_{\mathfrak{A}}(w) \leq p \cdot |w|$. Sei

$$s : \begin{cases} \mathbb{N} \to \mathbb{N} \\ n \mapsto p \cdot n \end{cases} .$$

Für alle Wörter $w \in L(\mathfrak{A}) \backslash \{\varepsilon\}$ existieren also gemäß Aufgabenteil a) eine Endkonfiguration κ und eine natürliche Zahl n, so daß

$$\kappa(w) \vdash_{\mathfrak{A}}^n \kappa \ \text{ und } \ n \leq |Q| \cdot p \cdot |w| \cdot |\Gamma|^{p \cdot |w|}.$$

c) Sei $\mathfrak{A} = \langle Q, \Sigma, \Gamma, q_0, \square, F, \delta \rangle \in \mathrm{TM}(\Sigma)$, und sei $s : \mathbb{N} \to \mathbb{N}$ eine Funktion, so daß für alle Wörter $w \in L(\mathfrak{A})$ gilt $bv_{\mathfrak{A}}(w) \leq s(|w|)$. Falls s berechenbar ist, so ist $L(\mathfrak{A})$ entscheidbar wie folgt:

Für ein $w \in \Sigma^*$ berechne $s(|w|)$ und untersuche die endlich vielen Berechnungen $\gamma = (\kappa(w) \vdash_{\mathfrak{A}} \ldots)$ der Länge $|Q| \cdot s(|w|) \cdot |\Gamma|^{s(|w|)}$ auf Endkonfigurationen.

1. Fall: Endkonfiguration vorhanden $\rightsquigarrow w \in L(\mathfrak{A})$
2. Fall: keine Endkonfiguration vorhanden $\rightsquigarrow w \notin L(\mathfrak{A})$

Lösungsvorschlag ***Automatentheorie und Formale Sprachen***
Aufgaben zur Klausurvorbereitung

Aufgabe 1

Seien $\alpha, \beta \in \mathrm{RegE}(\Sigma)$.

1) Konstruiere $\mathfrak{A}_1, \mathfrak{A}_2 \in \mathrm{NFA}(\Sigma)$, so daß $L(\mathfrak{A}_1) = [\![\alpha]\!]$ und $L(\mathfrak{A}_2) = [\![\beta]\!]$.
(Thompson)

2) Konstruiere $\mathfrak{A}'_1, \mathfrak{A}'_2 \in \mathrm{DFA}(\Sigma)$, so daß $L(\mathfrak{A}'_1) = L(\mathfrak{A}_1)$ und $L(\mathfrak{A}'_2) = L(\mathfrak{A}_2)$.
(Potenzmengenkonstruktion)

3) Konstruiere $\overline{\mathfrak{A}'_1}, \overline{\mathfrak{A}'_2} \in \mathrm{DFA}(\Sigma)$, so daß $L(\overline{\mathfrak{A}'_1}) = \overline{L(\mathfrak{A}'_1)}$ und $L(\overline{\mathfrak{A}'_2}) = \overline{L(\mathfrak{A}'_2)}$.
(Komplementierung der Endzustandsmenge)

4) Konstruiere die Produktautomaten $\mathfrak{A}'_1 \times \overline{\mathfrak{A}'_2}, \overline{\mathfrak{A}'_1} \times \mathfrak{A}'_2 \in \mathrm{DFA}(\Sigma)$, so daß
$L(\mathfrak{A}'_1 \times \overline{\mathfrak{A}'_2}) = L(\mathfrak{A}'_1) \cap L(\overline{\mathfrak{A}'_2})$ und $L(\overline{\mathfrak{A}'_1} \times \mathfrak{A}'_2) = L(\overline{\mathfrak{A}'_1}) \cap L(\mathfrak{A}'_2)$.

5) Führe Lerheitstest durch:

$$L(\mathfrak{A}'_1 \times \overline{\mathfrak{A}'_2}) \stackrel{?}{=} \emptyset$$
$$L(\overline{\mathfrak{A}'_1} \times \mathfrak{A}'_2) \stackrel{?}{=} \emptyset$$

6) $[\![\alpha]\!] = [\![\beta]\!] \curvearrowleftright$ Beide leer!

Korrektheitsbeweis:

$$\begin{array}{rl} & [\![\alpha]\!] = [\![\beta]\!] \\ \curvearrowleftright & L(\mathfrak{A}_1) = L(\mathfrak{A}_2) \\ \curvearrowleftright & L(\mathfrak{A}'_1) = L(\mathfrak{A}'_2) \\ \curvearrowleftright & L(\mathfrak{A}'_1) \subseteq L(\mathfrak{A}'_2) \text{ und } L(\mathfrak{A}'_2) \subseteq L(\mathfrak{A}'_1) \\ \curvearrowleftright & L(\mathfrak{A}'_1) \cap L(\overline{\mathfrak{A}'_2}) = \emptyset \text{ und } L(\overline{\mathfrak{A}'_1}) \cap L(\mathfrak{A}'_2) = \emptyset \\ \curvearrowleftright & L(\mathfrak{A}'_1 \times \overline{\mathfrak{A}'_2}) = \emptyset \text{ und } L(\overline{\mathfrak{A}'_1} \times \mathfrak{A}'_2) = \emptyset \end{array}$$

Aufgabe 2

a) Die Aussage ist richtig. Wähle $L_r = \{a, b\}^*$ und $L_c = \{a^n b^n \mid n \in \mathbb{N}\}$.

b) Die Aussage ist falsch. Angenommen, $\overline{L_c} = L_r$. Dann gilt auch:

$$\begin{array}{rl} & \overline{\overline{L_c}} = \overline{L_r} \\ \curvearrowright & L_c = \overline{L_r} \\ \curvearrowright & L_c \in \mathrm{RegL}(\Sigma) \end{array}$$

Widerspruch!

Aufgabe 3

Idee: Die Automaten werden parallel durchlaufen. Dabei unterscheiden wir zwischen ε-Schritten und a-Schritten auf Seiten des Kellerautomaten.

Sei also $\mathfrak{A}^\cap = \langle Q_1 \times Q_2, \Sigma, \Gamma, \delta, (q_0^1, q_0^2), Z_0, F_1 \times F_2\rangle \in \mathrm{PDA}(\Sigma)$ gegeben durch:

$$\begin{aligned}\delta((q_1, q_2), a, Z) &= \{((q, \delta_2(q_2, a)), \beta) \mid (q, \beta) \in \delta_1(q_1, a, Z)\} \quad \forall a \in \Sigma \\ \delta((q_1, q_2), \varepsilon, Z) &= \{((q, q_2), \beta) \mid (q, \beta) \in \delta_1(q_1, \varepsilon, Z)\}\end{aligned}$$

Aufgabe 4

a) Die Aussage ist falsch, denn:

$$\forall L \subseteq \{a\}^* : \ \{aa\}^* \subseteq L \curvearrowright L\{\varepsilon, a\} = \{a\}^* \in \mathrm{RegL}(\{a\})$$

Aber $\mathcal{L} = \{L \subseteq \{a\}^* \mid \{aa\}^* \subseteq L\}$ ist überabzählbar. Denn es existiert eine Bijektion $f : \mathfrak{P}(\mathbb{N}) \to \mathcal{L}$ gemäß

$$f(N) := \{aa\}^* \cup \{a^{2 \cdot n+1} \mid n \in N\}.$$

Und $\mathfrak{P}(\mathbb{N})$ ist überabzählbar. Da es nur abzählbar viele reguläre Sprachen gibt, gilt schließlich:

$$\exists L \in \mathcal{L} : \ L \notin \mathrm{RegL}(\{a\}) \text{ und } L\{\varepsilon, a\} = \{a\}^*$$

b) Die Aussage ist richtig, denn:

$$\begin{aligned}& L \in \mathrm{CFL}(\Sigma) \\ \curvearrowright\ & \exists G \in \mathrm{CFG}(\Sigma) : \ L(G) = L \\ \curvearrowright\ & \exists G \in \mathrm{CNF}(\Sigma) : \ L(G) = L \\ \curvearrowright\ & \exists G = \langle N, \Sigma, P, S\rangle \in \mathrm{CNF}(\Sigma) : \ L(\langle N, \Sigma, P \backslash \{S \longrightarrow \varepsilon\}, S\rangle) = L \backslash \{\varepsilon\} \\ \curvearrowright\ & \exists G \in \mathrm{CNF}(\Sigma) : \ L(G) = L \backslash \{\varepsilon\} \\ \curvearrowright\ & \exists G \in \mathrm{CFG}(\Sigma) : \ L(G) = L \backslash \{\varepsilon\} \\ \curvearrowright\ & L \backslash \{\varepsilon\} \in \mathrm{CFL}(\Sigma)\end{aligned}$$

c) Die Aussage ist falsch, denn:

Seien $L \in \mathfrak{P}(\Sigma^*) \backslash \mathcal{L}_0(\Sigma)$ und $f : \mathbb{N} \backslash \{0\} \to \Sigma^* \backslash L$ eine surjektive Abbildung. Sei nun

$$L_n := \Sigma^* \backslash \{f(n)\} \in \mathrm{RegL}(\Sigma)$$

für alle $n \in \mathbb{N} \backslash \{0\}$. Dann gilt $\bigcap_{n=1}^{\infty} L_n = L$.