1) *Lösung.* Im Basisfall ist $z = \epsilon$ und somit

$$\hat{\delta}(q, y\epsilon) = \hat{\delta}(q, y) = \hat{\delta}(\hat{\delta}(q, y), \epsilon) .$$

Im Induktionsschritt müssen wir

$$\hat{\delta}(q, yza) = \hat{\delta}(\hat{\delta}(q, y), za)$$

zeigen. Es gilt die Induktionshypothese

$$\hat{\delta}(q, yz) = \hat{\delta}(\hat{\delta}(q, y), z) .$$

Wir erhalten

$$\begin{aligned}
&\text{(Definition 4.15)} && \hat{\delta}(q, yza) = \delta(\hat{\delta}(q, yz), a) \\
&\text{(IH)} && = \delta(\hat{\delta}(\hat{\delta}(q, y), z), a) \\
&\text{(Definition 4.15)} && = \hat{\delta}(\hat{\delta}(q, y), za) .
\end{aligned}$$

□

2) a) Wir definieren eine Grammatik $G = (V, \Sigma, R, S)$ mit

$$\begin{aligned}
V &= \{S, I, I', L, L'\} \\
\Sigma &= \{\mathsf{int}, \mathsf{a}, \dots, \mathsf{z}, =, -, 0, \dots, 9, ;\}
\end{aligned}$$

und den folgenden Regeln R.

$$\begin{aligned}
S &\to \mathsf{int}\ I \\
I &\to \mathsf{a}I' \mid \dots \mid \mathsf{z}I' \\
I' &\to \mathsf{a}I' \mid \dots \mid \mathsf{z}I' \mid = L \\
L &\to \mathsf{0;} \mid -\mathsf{1}L' \mid \dots \mid -\mathsf{9}L' \mid \mathsf{1}L' \mid \dots \mid \mathsf{9}L' \\
L' &\to \mathsf{0}L' \mid \dots \mid \mathsf{9}L' \mid ;
\end{aligned}$$

b) Wir definieren eine Grammatik G', wobei wir einige Teile der Grammatik G wiederverwenden.

$$\begin{aligned}
G' &= (V', \Sigma', R', S) \\
V' &= \{S, I_{\mathsf{i}}, I'_{\mathsf{i}}, I_{\mathsf{f}}, I'_{\mathsf{f}}, I_{\mathsf{s}}, I'_{\mathsf{s}}, L_{\mathsf{i}}, L'_{\mathsf{i}}, L_{\mathsf{f}}, L'_{\mathsf{f}}, L''_{\mathsf{f}}, L'''_{\mathsf{f}}, L_{\mathsf{s}}, L'_{\mathsf{s}}\} \\
\Sigma' &= \{\mathsf{int}, \mathsf{float}, \mathsf{char}*, \mathsf{a}, \dots, \mathsf{z}, =, -, 0, \dots, 9, ., ;, ", \backslash\}
\end{aligned}$$

Ausgehend vom Startsymbol erlauben wir zusätzlich die Ableitung von float und $\mathsf{char}*$.

$$S \to \mathsf{int}\ I_{\mathsf{i}} \mid \mathsf{float}\ I_{\mathsf{f}} \mid \mathsf{char}*\ I_{\mathsf{s}}$$

Die Regeln für die Ableitung des Variablennamen (Identifier), kopieren wir die Regeln von oben. Das Subskript $t \in \{\mathsf{i}, \mathsf{f}, \mathsf{s}\}$ verwenden wir dazu um uns den Typ der Variable zu „merken“.

$$\begin{aligned} I_t &\to \mathsf{a}I'_t \mid \dots \mid \mathsf{z}I'_t \\ I'_t &\to \mathsf{a}I'_t \mid \dots \mid \mathsf{z}I'_t \mid = L_t \end{aligned}$$

Die Regeln für int-Literale können, mit Subskript i versehen, von oben übernommen werden. Für Literale vom Typ float verwenden wir folgende Regeln.

$$\begin{aligned} L_{\mathsf{f}} &\to \mathsf{0}.L''_{\mathsf{f}} \mid -\mathsf{1}L'_{\mathsf{f}} \mid \dots \mid -\mathsf{9}L'_{\mathsf{f}} \mid \mathsf{1}L'_{\mathsf{f}} \mid \dots \mid \mathsf{9}L'_{\mathsf{f}} \\ L'_{\mathsf{f}} &\to \mathsf{0}L'_{\mathsf{f}} \mid \dots \mid \mathsf{9}L'_{\mathsf{f}} \mid .L''_{\mathsf{f}} \\ L''_{\mathsf{f}} &\to \mathsf{0}L'''_{\mathsf{f}} \mid \dots \mid \mathsf{9}L'''_{\mathsf{f}} \\ L'''_{\mathsf{f}} &\to \mathsf{0}L'''_{\mathsf{f}} \mid \dots \mid \mathsf{9}L'''_{\mathsf{f}} \mid ; \end{aligned}$$

Die Regeln für $\mathsf{char}*$-Literale (Strings) lauten wie folgt.

$$\begin{aligned} L_{\mathsf{s}} &= {}"L'_{\mathsf{s}} \\ L'_{\mathsf{s}} &= \mathsf{a}L'_{\mathsf{s}} \mid \dots \mid \mathsf{z}L'_{\mathsf{s}} \mid \mathsf{0}L'_{\mathsf{s}} \mid \dots \mid \mathsf{9}L'_{\mathsf{s}} \mid \backslash"L'_{\mathsf{s}} \mid {}"; \end{aligned}$$

Beachten Sie, dass die von der Grammatik G' erzeugte Sprache nur eine Untermenge der Programmiersprache C möglichen Deklarationen enthält.

c) Nein, es ist nicht möglich allgemeine Rechenausdrücke mit einer rechtslinearen Grammatik auszudrücken. Intuition: In einer regulären Sprache, wie sie von einer rechtslinearen Grammatik erzeugt wird, können wir nur endlich viele Zustände simulieren (oben wird z.B. durch die Variablen I_{i}, I_{f} und I_{s} unterschieden, welchen Typ wir deklarieren). D.h. wir könnten eine eingeschränkte Menge von möglichen Ausdrücken hinzufügen, aber keine allgemeine Form.

3) Vorab formulieren wir einige Hilfssätze für die gegebene Grammatik G.

Sei x ein beliebiges Wort und $j \in \mathbb{N}$, dann existiert

`(H1a)` zu jeder gültigen Ableitung $S \Rightarrow^n xA$ auch eine Ableitung $S \Rightarrow^{n+2j} x(\mathsf{ab})^j A$,

`(H1b)` zu jeder gültigen Ableitung $S \Rightarrow^n xB$ auch eine Ableitung $S \Rightarrow^{n+2j} x(\mathsf{ba})^j B$.

`(H3)` Für ein beliebiges $w \in L_4$ können wir die folgenden vier Fallunterscheidungen treffen. ($w = \mathsf{c} \in L_4$ wird dabei von zwei Fällen erfasst.)

- `(F1)` $w \in \{\mathsf{ab}\}^*\{\mathsf{c}\} = \{(\mathsf{ab})^n\mathsf{c}, \mid n \in \mathbb{N}\}$,
- `(F2)` $w \in \{\mathsf{b}\}\{\mathsf{ab}\}^*\{\mathsf{c}\} = \{\mathsf{b}(\mathsf{ab})^n\mathsf{c}, \mid n \in \mathbb{N}\}$,

(F3) $w \in \{\mathsf{ba}\}^*\{\mathsf{c}\} = \{(\mathsf{ba})^n\mathsf{c}, \mid n \in \mathbb{N}\}$ oder

(F4) $w \in \{\mathsf{a}\}\{\mathsf{ba}\}^*\{\mathsf{c}\} = \{\mathsf{a}(\mathsf{ba})^n\mathsf{c}, \mid n \in \mathbb{N}\}$.

Lösung. Wir beweisen vorab die Hilfsätze.

(H1a) Wir zeigen dies mit vollständiger Induktion:

- Induktionsbasis $j = 0$. $S \Rightarrow xA = x(\mathsf{ab})^0A$.
- Induktionschritt $j + 1$.
  $S \Rightarrow xA \underset{I.H.}{\Rightarrow}^{2j} x(\mathsf{ab})^jA \Rightarrow x(\mathsf{ab})^j\mathsf{a}B \Rightarrow x(\mathsf{ab})^j\mathsf{ab}A = x(\mathsf{ab})^{2(j+1)}A$

Damit ist (H1a) gezeigt.

(H1b) wird wie H1a gezeigt.

(H3) Wir zeigen dies durch Umformung der Definition von L_4:

$$\begin{aligned} \text{Sei } w_1 \in L_4 &= (L_1 \cup L_2)\, L_3 \\ &= \Big(\{\epsilon, \mathsf{b}\}\{\mathsf{ab}\}^* \cup \{\epsilon, \mathsf{a}\}\{\mathsf{ba}\}^*\Big)\{\mathsf{c}\} \\ &= \{\mathsf{ab}\}^*\{\mathsf{c}\} \cup \{\mathsf{b}\}\{\mathsf{ab}\}^*\{\mathsf{c}\} \cup \{\mathsf{ba}\}^*\{\mathsf{c}\} \cup \{\mathsf{a}\}\{\mathsf{ba}\}^*\{\mathsf{c}\} \end{aligned}$$

a) Wir zeigen die Vollständigkeit der Grammatik G.

Wir zeigen nun (F1), also $w = (\mathsf{ab})^n\mathsf{c}$. Wir unterscheiden zwei Fälle:

- $w = (\mathsf{ab})^0\mathsf{c} = \mathsf{c}$. Das Wort ist in einem Schritt ableitbar: $S \Rightarrow \mathsf{c}$.
- $w = (\mathsf{ab})^{n+1}\mathsf{c} = \mathsf{ab}(\mathsf{ab})^n\mathsf{c}$ mit $n \in \mathbb{N}$. Die ersten beiden Schritte der Ableitung sind nun $S \Rightarrow \mathsf{a}B \Rightarrow \mathsf{ab}A$ und mit (H1a) ist bereits gezeigt, dass $S \Rightarrow^* \mathsf{ab}(\mathsf{ab})^nA \Rightarrow \mathsf{ab}(\mathsf{ab})^n\mathsf{c}$ existiert.

Damit ist gezeigt, dass alle Wörter aus (F1) in $L(G)$ vorhanden sind. Auf ähnliche Weise ist leicht zu zeigen, dass dies auch für (F2), (F3), (F4) gilt. Damit ist G eine *vollständige* Grammatik für L_4, d.h. $L_4 \subseteq L(G)$.

b) Wir zeigen die Korrektheit der Grammatik G.

Wir betrachten nun $w \in L(G)$ nach der Länge der Ableitung $S \Rightarrow^\ell w$

- $\ell = 1 : S \Rightarrow \mathsf{c} \in L_4$.
- $\ell = 2k + 2 :$
  - $S \Rightarrow \mathsf{a}B \Rightarrow \mathsf{a}(\mathsf{ba})^kB \Rightarrow \mathsf{a}(\mathsf{ba})^k\mathsf{c} \in L_2L_3$
  - $S \Rightarrow \mathsf{b}A \Rightarrow \mathsf{b}(\mathsf{ab})^kA \Rightarrow \mathsf{b}(\mathsf{ab})^k\mathsf{c} \in L_1L_3$
- $\ell = 2k + 3 :$
  - $S \Rightarrow \mathsf{a}B \Rightarrow \mathsf{ab}A \Rightarrow^{2k} \mathsf{ab}(\mathsf{ab})^kA \Rightarrow (\mathsf{ab})^{k+1}\mathsf{c} \in L_1L_3$
  - $S \Rightarrow \mathsf{b}A \Rightarrow \mathsf{ba}B \Rightarrow^{2k} \mathsf{ba}(\mathsf{ba})^kB \Rightarrow (\mathsf{ba})^{k+1}\mathsf{c} \in L_2L_3$

Dies kann mit vollständiger Induktion für $k \geq 0$ gezeigt werden.

Die Grammatik G erzeugt also in jedem Fall ein Wort der formalen Sprache L_4. Damit ist G eine *korrekte* Grammatik für L_4, d.h. $L(G) \subseteq L_4$.

$\square$