1655 Grundlagen der Theoretischen Informatik
Lösungsvorschläge zur Klausur vom 3. März 2001

Aufgabe 1

(a) Wir beweisen die Behauptung durch Induktion über $i \in \mathbb{N}$.
<u>Induktionsanfang $i = 0$</u>: Sei $n \in \mathbb{N}$. Es ist $n \geq 0 = 2 \cdot 0$. Es gilt

$$\mathrm{ES}^{4\cdot 0}\big(1, (0, n, 0, \dots)\big) = \big(1, (0, n, 0, \dots)\big) = \big(1, (0, n - 2 \cdot 0, 0, \dots)\big).$$

<u>Induktionsschluß $i \to i+1$</u>: Sei $n \in \mathbb{N}$ mit $n \geq 2(i+1)$. Dann gilt insbesondere $n \geq 2i$ und daher nach Induktionsannahme

$$\mathrm{ES}^{4i}\big(1, (0, n, 0, \dots)\big) = \big(1, (0, n - 2i, 0, \dots)\big).$$

Daraus erhalten wir:

$$\begin{aligned}
\mathrm{ES}^{4(i+1)}\big(1, (0, n, 0, \dots)\big) &= \mathrm{ES}^{4}\big(1, (0, n - 2i, 0, \dots)\big) \\
&= \mathrm{ES}^{3}\big(2, (0, n - 2i, 0, \dots)\big) \\
&\qquad \text{(aus } n \geq 2(i+1) \text{ folgt nämlich } n - 2i > 0) \\
&= \mathrm{ES}^{2}\big(3, (0, n - 2i - 1, 0, \dots)\big) \\
&= \mathrm{ES}\big(4, (0, n - 2i - 1, 0, \dots)\big) \\
&\qquad \text{(aus } n \geq 2(i+1) \text{ folgt nämlich } n - 2i - 1 > 0) \\
&= \big(1, (0, n - 2i - 2, 0, \dots)\big) \\
&= \big(1, (0, n - 2(i+1), 0, \dots)\big)
\end{aligned}$$

Damit ist der Induktionsschluß gelungen und die Behauptung bewiesen.
(Anmerkung: Beim Induktionsschluß kann man auch andersherum vorgehen. Sei wieder $n \in \mathbb{N}$ mit $n \geq 2(i+1)$. Man kann erst vier Rechenschritte ausführen und erhält:

$$\mathrm{ES}^{4(i+1)}\big(1, (0, n, 0, \dots)\big) = \mathrm{ES}^{4i}\big(1, (0, n - 2, 0, \dots)\big)$$

(da $n - 1 \geq 2(i+1) - 1 > 0$). Dann wendet man die Induktionsannahme auf $n - 2$ anstelle von n an und erhält wegen $n - 2 \geq 2i$:

$$\begin{aligned}
\mathrm{ES}^{4i}\big(1, (0, n - 2, 0, \dots)\big) &= \big(1, (0, n - 2 - 2i, 0, \dots)\big) \\
&= \big(1, (0, n - 2(i+1), 0, \dots)\big).
\end{aligned}$$

Also gelingt auch auf diese Weise der Induktionsschluß.)

(b) Das Flußdiagramm aus der Aufgabenstellung sei mit F bezeichnet. Wir weisen die Behauptung für alle $n \in \mathbb{N}$ nach. Sei $n \in \mathbb{N}$. Wir unterscheiden die Fälle "n gerade" und "n ungerade".

1. Fall: n ist gerade. Dann gibt es eine Zahl $m \in \mathbb{N}$ mit $n = 2m$. Wir erhalten aus der Behauptung in (a) für $i = m$

$$\mathrm{ES}^{4m}\big(1, (0, n, 0, \dots)\big) = \big(1, (0, 0, 0, \dots)\big),$$

also

$$\mathrm{GS}\big(1, (0, n, 0, \dots)\big) = \mathrm{ES}^{4m+1}\big(1, (0, n, 0, \dots)\big) = \big(6, (0, 0, 0, \dots)\big).$$

Da die Registermaschine einstellig ist, folgt daraus:

$$\begin{aligned} f_M(n) &= \mathrm{AC} \circ f_F \circ \mathrm{EC}^{(1)}(n) \\ &= \mathrm{AC} \circ f_F(0, n, 0, \dots) \\ &= \mathrm{AC}(0, 0, \dots) \\ &= 0 \end{aligned}$$

2. Fall: n ist ungerade. Dann gibt es eine Zahl $m \in \mathbb{N}$ mit $n = 2m + 1$. Wir erhalten aus der Behauptung in (a) für $i = m$

$$\mathrm{ES}^{4m}\big(1, (0, n, 0, \dots)\big) = \big(1, (0, 1, 0, \dots)\big),$$

also

$$\begin{aligned} \mathrm{GS}\big(1, (0, n, 0, \dots)\big) &= \mathrm{ES}^{4m+4}\big(1, (0, 2m+1, 0, \dots)\big) \\ &= \mathrm{ES}^{4}\big(1, (0, 1, 0, \dots)\big) \\ &= \mathrm{ES}^{3}\big(2, (0, 1, 0, \dots)\big) \\ &= \mathrm{ES}^{2}\big(3, (0, 0, 0, \dots)\big) \\ &= \mathrm{ES}^{1}\big(5, (0, 0, 0, \dots)\big) \\ &= \big(6, (1, 0, 0, \dots)\big). \end{aligned}$$

Da die Registermaschine einstellig ist, folgt daraus:

$$\begin{aligned} f_M(n) &= \mathrm{AC} \circ f_F \circ \mathrm{EC}^{(1)}(n) \\ &= \mathrm{AC} \circ f_F(0, n, 0, \dots) \\ &= \mathrm{AC}(1, 0, \dots) \\ &= 1 \end{aligned}$$

Wir haben die Behauptung bewiesen: $f_M(n) = 0$ für gerade n und $f_M(n) = 1$ für ungerade n.

Aufgabe 2

Wir definieren eine Funktion $g : \mathbb{N}^2 \to \mathbb{N}$ durch

$$g(\langle i, j\rangle, \langle k, l\rangle) := \langle 3i + 2l, 3j + 2k\rangle$$

für alle $i, j, k, l \in \mathbb{N}$. Dann ist g berechenbar, und es gilt

$$\begin{aligned}
\nu_{\mathbb{Z}} g(\langle i, j\rangle, \langle k, l\rangle) &= \nu_{\mathbb{Z}}(\langle 3i + 2l, 3j + 2k\rangle) \\
&= 3i + 2l - 3j - 2k \\
&= 3(i - j) - 2(k - l) \\
&= f(i - j, k - l) \\
&= f\left(\nu_{\mathbb{Z}}(\langle i, j\rangle), \nu_{\mathbb{Z}}(\langle k, l\rangle)\right)
\end{aligned}$$

für alle $i, j, k, l \in \mathbb{N}$. Damit ist f eine $(\nu_{\mathbb{Z}}, \nu_{\mathbb{Z}}, \nu_{\mathbb{Z}})$–berechenbare Funktion.

Aufgabe 3

Wir definieren eine Zahlenfunktion $h :\subseteq \mathbb{N}^2 \to \mathbb{N}$ durch

$$h(i, n) := 2 \cdot u_{\varphi}(i, n)$$

für alle $i, n \in \mathbb{N}$. Da nach dem utm–Theorem die Funktion u_{φ} berechenbar ist, ist die Funktion h als Substitution berechenbarer Funktionen selbst berechenbar. Nach dem smn–Theorem gibt es eine total-berechenbare Funktion $r : \mathbb{N} \to \mathbb{N}$, so daß

$$\varphi_{r(i)}(n) = h(i, n)$$

für alle $i, n \in \mathbb{N}$ gilt. Das bedeutet, daß für alle $i, n \in \mathbb{N}$ gilt

$$\begin{aligned}
\varphi_{r(i)}(n) &= h(i, n) \\
&= 2 \cdot u_{\varphi}(i, n) \\
&= 2 \cdot \varphi_i(n) \\
&= H(\varphi_i)(n).
\end{aligned}$$

Also gilt $H(\varphi(i)) = \varphi(r(i))$ für alle $i \in \mathbb{N}$. Das heißt, r ist eine (φ, φ)–Realisierung von H. Also ist H eine (φ, φ)–berechenbare Funktion.

Aufgabe 4

(a) Seien A und B rekursiv. Da das Komplement einer rekursiven Menge wieder rekursiv ist (nach Satz 6.3.1(1)), ist auch $\mathbb{N} \setminus B$ rekursiv. Da der Schnitt zweier rekursiver Mengen wieder rekursiv ist, ist auch $A \cap (\mathbb{N} \setminus B)$ rekursiv. Dies ist gerade die Menge $A \setminus B$. Also ist diese Menge rekursiv.

(b) Sei A rekursiv–aufzählbar und B rekursiv. Wir hatten eben schon gesehen, daß dann auch das Komplement $\mathbb{N} \setminus B$ von B rekursiv ist. Da jede rekursive Menge auch rekursiv–aufzählbar ist (nach Satz 6.2.4), ist $\mathbb{N} \setminus B$ auch rekursiv–aufzählbar. Da der Schnitt zweier rekursiv–aufzählbarer Mengen wieder rekursiv–aufzählbar ist (nach Satz 6.3.1(2)), ist auch die Menge $A \cap (\mathbb{N} \setminus B)$ rekursiv–aufzählbar. Dies ist gerade die Menge $A \setminus B$. Also ist diese Menge rekursiv–aufzählbar.

(c) Wir wählen $A := \mathbb{N}$ und $B := K$, wobei $K = \{i \in \mathbb{N} \mid i \in \mathrm{Def}(\varphi_i)\}$ sei.
Der Vollständigkeit halber geben wir eine kurze Begründung. Die Menge $\mathbb{N}$ ist rekursiv (also nach Satz 6.2.4 auch rekursiv–aufzählbar), da ihre charakteristische Funktion $\mathrm{cf}_{\mathbb{N}}$ einfach die konstante Funktion mit dem Wert 0 ist und diese Funktion bekanntlich berechenbar ist. Die Menge K ist nach Satz 6.2.5 rekursiv–aufzählbar, aber nicht rekursiv. Daher ist nach Satz 6.2.4 ihr Komplement $\mathbb{N} \setminus K$ nicht rekursiv–aufzählbar. Dies ist aber gerade die Menge $A \setminus B$.

Aufgabe 5

(a) Wir beschreiben eine Turingmaschine M mit $f_M(0^n) = 0^{f(n)}$ für alle $n \in \mathbb{N}$ und mit $\widetilde{s}_M \in O(f)$. Bei Eingabe eines Wortes w arbeite die Maschine M wie folgt:

(1) Lies das nächste Zeichen auf dem Eingabeband. Falls es eine 0 ist, gehe nach (2). Sonst schreibe eine 0 auf das Ausgabeband und halte an.

(2) Lies das nächste Zeichen auf dem Eingabeband. Falls es eine 0 ist, gehe nach (1). Sonst halte an.

Der Vollständigkeit halber geben wir auch eine kurze Erläuterung zur **Korrektheit**: wenn die Maschine in den Zustand (1) gelangt, hat sie bis dahin eine gerade Anzahl von Nullen gelesen; wenn sie in den Zustand (2) gelangt, hat sie bis dahin eine ungerade Anzahl von Nullen gelesen. Also gilt tatsächlich:

$$\begin{aligned} f_M(0^n) &= \begin{cases} \varepsilon = 0^0 & \text{falls } n \text{ gerade,} \\ 0 = 0^1 & \text{falls } n \text{ ungerade.} \end{cases} \\ &= 0^{f(n)}. \end{aligned}$$

Zur **Bandbedarfsabschätzung**: Die Maschine M benutzt überhaupt keine Felder auf den Arbeitsbändern. Ihr Bandbedarf ist also bei jeder Eingabe Null, d.h. für alle $n \in \mathbb{N}$ ist $\widetilde{s}_M(n) = 0 \leq f(n)$, also $\widetilde{s}_M \in O(f)$. Damit haben wir gezeigt, daß f bandkonstruierbar ist.

(b) Wir führen einen Widerspruchsbeweis und nehmen an, f sei zeitkonstruierbar. Dann muß es eine Turingmaschine T mit $f_T(0^n) = 0^{f(n)}$ für alle $n \in \mathbb{N}$ und mit $\widetilde{t}_M \in O(f)$ geben. Es muß dann auch eine Konstante $c > 0$ geben mit $\widetilde{t}_M(n) \leq c \cdot f(n) + c \leq 2c$ für alle $n \in \mathbb{N}$. Die Maschine wird also bei jeder Eingabe höchstens $2c$ Schritte lang rechnen. Bei Eingaben der Form 0^m mit $m \geq 2c$ kann die Maschine also höchstens die ersten $2c$ Zeichen der Eingabe lesen. Da diese für alle Eingabewörter der Form 0^m mit $m \geq 2c$ die gleichen sind, muß sie für alle derartigen Eingabewörter die gleiche Ausgabe liefern. Zum Beispiel muß

$$0^{f(2c)} = f_T(0^{2c}) = f_T(0^{2c+1}) = 0^{f(2c+1)}$$

gelten. Dies steht aber im Widerspruch zu $0 = f(2c) \neq f(2c+1) = 1$. Also war unsere Annahme falsch. Wir haben gezeigt, daß die Funktion f nicht zeitkonstruierbar sein kann.

Aufgabe 6

Das Ziel dieser Aufgabe ist, zu zeigen, daß die Sprache ZWEI-RUCKSAECKE **NP**-vollständig ist. Dazu müssen zwei Dinge gezeigt werden: erstens, daß sie in **NP** liegt, und zweitens, daß sie **NP**-hart ist.

(a) Zuerst zeigen wir, daß ZWEI-RUCKSAECKE in **NP** liegt. Dazu beschreiben wir eine Kontrollturingmaschine M mit L_M = ZWEI-RUCKSAECKE, die in polynomieller Zeit arbeitet. Wir beschreiben nur, wie sie bei Eingabewörtern der Form $w = d(N_1)\#d(N_2)\#d(k_1)\#\ldots\#d(k_n)$ mit $n \geq 1$ und $N_1, N_2, k_1, \ldots, k_n \in \mathbb{N}$ arbeitet. Sei $y = y_1y_2\ldots \in \{0,1\}^*$ die Hilfseingabe.

(1) Kopiere $d(N_1)$ auf Band 2 und $d(N_2)$ auf Band 4, und schreibe $d(0)$ auf Band 3 und Band 5.

(2) Lies die nächsten beiden Zeichen $y_{2i-1}y_{2i}$ der Hilfseingabe.

(3) Falls $y_{2i-1}y_{2i} \notin \{0,1\}^2$, brich ab, ohne zu akzeptieren, d.h. gehe zu HALT 2. Sonst fahre fort mit (4).

(4) Falls $y_{2i-1}y_{2i} = 10$, addiere die nächste Dualzahl $d(k_i)$ zum Inhalt von Band 3.
Falls $y_{2i-1}y_{2i} = 01$, addiere die nächste Dualzahl $d(k_i)$ zum Inhalt von Band 5.
Falls $y_{2i-1}y_{2i} = 00$ oder $y_{2i-1}y_{2i} = 11$, überlies die nächste Dualzahl $d(k_i)$.
Falls die Eingabe zu Ende, gehe zu (5), sonst zu (2).

(5) Vergleiche die Inhalte von Band 2 und Band 3. Falls sie gleich sind, gehe zu (6), sonst brich ab, ohne zu akzeptieren, d.h. HALT 2.

(6) Vergleiche die Inhalte von Band 4 und Band 5. Falls sie gleich sind, akzeptiere (HALT 1), sonst brich ab, ohne zu akzeptieren (HALT 2).

Der Vollständigkeit halber machen wir einige Anmerkungen zur **Korrektheit** der Maschine. Die Korrektheit der Maschine, d.h. L_M = ZWEI-RUCKSAECKE, kann man genauso einsehen wie die Korrektheit der Maschine im Kurs in Beispiel 9.2.6, die die Sprache RUCKSACK erkennt. Wenn es nämlich zu einer Eingabe $w = d(N_1)\#d(N_2)\#d(k_1)\#\ldots\#d(k_n)$ zwei disjunkte Teilmengen $M_1, M_2 \subseteq \{1,\ldots,n\}$ mit $\sum_{i\in M_1} k_i = N_1$ und $\sum_{i\in M_2} k_i = N_2$ gibt, so wird die Maschine bei dieser Eingabe w und zum Beispiel bei einer Hilfseingabe $y = y_1y_2\ldots$ mit

$$y_{2i-1}y_{2i} = \begin{cases} 10 & \text{falls } k_i \in M_1, \\ 01 & \text{falls } k_i \in M_2, \\ 00 & \text{sonst, falls } i \leq n, \end{cases}$$

im akzeptierenden Endzustand (HALT 1) anhalten. Wenn es jedoch keine zwei derartigen Teilmengen M_1 und M_2 gibt, wird die Maschine bei jeder Hilfseingabe im verwerfenden Endzustand (HALT 2) anhalten.

Eine **Zeitabschätzung** kann ebenso durchgeführt werden, wie zur Maschine im Kurs in Beispiel 9.2.6. Da das Kopieren und Vergleichen in linearer Zeit geht, da man Dualzahlen in polynomieller Zeit addieren kann, und da die Schleife (2) – (4) höchstens n-mal durchlaufen wird, arbeitet die Maschine insgesamt in polynomieller Zeit.

Wir haben ZWEI-RUCKSAECKE $\in$ **NP** gezeigt.

(b) Wir müssen noch zeigen, daß die Sprache ZWEI-RUCKSAECKE **NP**–hart ist, d.h. daß für alle $L \in$ **NP** gilt: $L \leq_{pol}$ ZWEI-RUCKSAECKE.

Wir betrachten eine Turingmaschine T, die auf das Ausgabeband zuerst das Wort $d(0)\#$ schreibt und dann dahinter das Eingabewort kopiert.
Es ist klar, daß diese Maschine in linearer Zeit arbeitet. Daher ist $f_T \in$ **FP**.
Da die Maschine bei einer Eingabe $x = d(N)\#d(k_1)\#\ldots\#d(k_n)$ die Ausgabe $f_T(x) = d(0)\#d(N)\#d(k_1)\#\ldots\#d(k_n)$ liefert, ist auch klar, daß

$$\forall x \in \Sigma^*.\ (x \in \text{RUCKSACK} \iff f_T(x) \in \text{ZWEI-RUCKSAECKE}).$$

Damit haben wir RUCKSACK $\leq_{pol}$ ZWEI-RUCKSAECKE nachgewiesen. Da die Sprache RUCKSACK bekanntlich **NP**–vollständig — also auch **NP**–hart — ist, gilt $\forall L \in$ **NP**. $L \leq_{pol}$ RUCKSACK. Da die Relation $\leq_{pol}$ nach Satz 9.4.3 transitiv ist, folgt auch $\forall L \in$ **NP**. $L \leq_{pol}$ ZWEI-RUCKSAECKE. Also ist die Sprache ZWEI-RUCKSAECKE ebenfalls **NP**–hart. Das war zu zeigen.

Aufgabe 7

Das Verfahren aus Abschnitt 11.3 liefert:

P	a	b	$W \setminus V$
$\{0\}$	$\{1\}$	$\{0,1\}$	$\{1\},\{0,1\}$
$\{1\}$	$\{2\}$	$\{1\}$	$\{0,1\},\{2\}$
$\{0,1\}$	$\{1,2\}$	$\{0,1\}$	$\{2\},\{1,2\}$
$\{2\}$	$\{0\}$	$\{0\}$	$\{1,2\}$
$\{1,2\}$	$\{0,2\}$	$\{0,1\}$	$\{0,2\}$
$\{0,2\}$	$\{0,1\}$	$\{0,1\}$	

Die Zustandsmenge des vereinfachten Potenzautomaten $B = (\Sigma, Q', q_0', F', \delta')$ ist damit $Q' := \{\{0\},\{0,1\},\{0,2\},\{1\},\{1,2\},\{2\}\}$, der Anfangszustand ist $q_0' := \{0\}$ und die Endzustandsmenge ist $F' := \{\{0,2\},\{1,2\},\{2\}\}$. Die Überführungsrelation ist aus der Tabelle ablesbar.

Aufgabe 8

(a) Wir führen einen Widerspruchsbeweis und nehmen an, L sei kontextfrei. Dann gibt es nach dem Pumping–Lemma für kontextfreie Sprachen (Satz 12.4.3) eine Zahl $p \in \mathbb{N}$, so daß für jedes Wort $z \in L$ mit $lg(z) \geq p$ gilt: es gibt Wörter $u, v, w, x, y \in \Sigma^*$ mit $z = uvwxy$, mit $vx \neq \varepsilon$, mit $lg(vwx) \leq p$ und mit $\forall i \geq 0.\, uv^iwx^iy \in L$. Wir betrachten das Wort $z := \mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Dieses Wort liegt in L. Daher gibt es Wörter $u, v, w, x, y \in \Sigma^*$ mit $z = uvwxy$, mit $vx \neq \varepsilon$, mit $lg(vwx) \leq p$ und mit $\forall i \geq 0.\, uv^iwx^iy \in L$. Wir betrachten jetzt alle möglichen Fälle für die Lage der Teilwörter v und x und zeigen, daß alle Fälle auf einen Widerspruch führen.

1. Fall: v oder x enthält ein b. Dann enthält das Wort uv^2wx^2y mindestens ein b mehr als $uvwxy = \mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$, also mindestens drei b's. Da alle Wörter in L genau zwei b's enthalten, kann uv^2wx^2y nicht in L sein. Widerspruch.

Damit können v und x kein b enthalten. Beide Wörter sind daher Teilwörter von a^p–Blöcken in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Es gilt daher $v = \mathsf{a}^{lg(v)}$ und $x = \mathsf{a}^{lg(x)}$.

2. Fall: v und x sind Teilwörter des ersten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Dann ist $uv^2wx^2y = a^{p+lg(v)+lg(w)}\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Wegen $vx \neq \varepsilon$ ist $lg(v) + lg(w) > 0$, also $p + lg(v) + lg(w) > p$. Daher kann uv^2wx^2y nicht in L sein. Widerspruch.

3. Fall: v und x sind Teilwörter des zweiten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Analog zum 2. Fall.

4. Fall: v und x sind Teilwörter des dritten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Analog zum 2. Fall.

5. Fall: v ist Teilwort des ersten a^p–Blocks und x Teilwort des zweiten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Dann ist $uv^2wx^2y = \mathsf{a}^{p+lg(v)}\mathsf{b}\mathsf{a}^{p+lg(x)}\mathsf{b}\mathsf{a}^p$. Wegen $vx \neq \varepsilon$ ist $lg(v) > 0$ oder $lg(x) > 0$. Da der dritte a–Block aber auch in diesem Wort die Länge p hat, kann dieses Wort nicht in L sein.

6. Fall: v ist Teilwort des zweiten a^p–Blocks und x Teilwort des dritten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Analog zum 5. Fall.

7. Fall: v ist Teilwort des ersten a^p–Blocks und x Teilwort des dritten a^p–Blocks in $\mathsf{a}^p\mathsf{b}\mathsf{a}^p\mathsf{b}\mathsf{a}^p$. Dieser Fall kann nicht eintreten, da dann das Wort w das Wort $\mathsf{b}\mathsf{a}^p\mathsf{b}$ als Teilwort enthalten müßte. Dies ist wegen $lg(vwx) \leq p$ aber nicht möglich.

Damit haben wir unsere Annahme, daß L kontextfrei ist, zum Widerspruch geführt. Also kann L nicht kontextfrei sein.

(b) Für die Grammatik $G := (\{S\}, \Sigma, R, S)$ mit der Regelmenge R:

$$S \longrightarrow \mathsf{a}S\mathsf{a} \mid \mathsf{b}$$

gilt $L(G) = L_2$.

Aufgabe 9

(a) Wir zeigen die Behauptung durch vollständige Induktion über n.

<u>Induktionsanfang $n = 0$:</u> Das einzige Wort in $\{\mathsf{b}, \mathsf{c}\}^0$ ist das leere Wort ε. Es gilt $S \xrightarrow{0} S = \mathsf{a}^0 S\varepsilon$.

<u>Induktionsschluß $n \to n+1$:</u> Sei $w \in \{\mathsf{b}, \mathsf{c}\}^{n+1}$. Dann gibt es ein Wort $v \in \Sigma^n$ und ein Zeichen $z \in \{\mathsf{b}, \mathsf{c}\}$ mit $w = zv$. Nach Induktionsannahme gilt $S \xrightarrow{*} \mathsf{a}^n Sv$. Wir unterscheiden nun zwei Fälle.

1. Fall: $z = \mathsf{b}$. Durch Anwendung der Regel $S \to \mathsf{a}S\mathsf{b}$ erhält man $\mathsf{a}^n Sv \to \mathsf{a}^n \mathsf{a}S\mathsf{b}v$, also $S \xrightarrow{*} \mathsf{a}^{n+1} S\mathsf{b}v = \mathsf{a}^{n+1} Sw$.

2. Fall: $z = \mathsf{c}$. Durch Anwendung der Regel $S \to \mathsf{a}S\mathsf{c}$ erhält man $\mathsf{a}^n Sv \to \mathsf{a}^n \mathsf{a}S\mathsf{c}v$, also $S \xrightarrow{*} \mathsf{a}^{n+1} S\mathsf{c}v = \mathsf{a}^{n+1} Sw$.

Damit gelingt der Induktionsschluß, und die Behauptung ist bewiesen.

(b) Wir sollen zeigen: $L \subseteq L(G)$. Sei $w \in L$. Wir müssen $w \in L(G)$ zeigen, also $S \xrightarrow{*} w$.

Wegen $w \in L$ gibt es ein $v \in \{\mathsf{b}, \mathsf{c}\}^*$ mit $w = \mathsf{a}^{lg(v)} v$. Sei $n := lg(v)$. Nach der in (a) bewiesenen Behauptung gilt $S \xrightarrow{*} \mathsf{a}^n Sv$. Durch Anwendung der Regel $S \to \varepsilon$ erhalten wir $\mathsf{a}^n Sv \to \mathsf{a}^n v$, also $S \xrightarrow{*} \mathsf{a}^n v = w$. Damit ist $w \in L(G)$ gezeigt. Wir haben $L \subseteq L(G)$ bewiesen.

(c) Ja, es gilt auch $L(G) \subseteq L$, also $L(g) = L$. Um $L(g) \subseteq L$ zu beweisen, kann man zum Beispiel erst die folgende Behauptung mit vollständiger Induktion über n beweisen:

$$\forall n \in \mathbb{N}.\ \forall W \in (\Sigma \cup \Pi)^*.$$
$$\Big(S \xrightarrow{*} W \Longrightarrow \big(\exists v \in \{\mathsf{b}, \mathsf{c}\}^*.\ (W = \mathsf{a}^{lg(v)} Sv \ \text{ oder } \ W = \mathsf{a}^{lg(v)} v)\big)\Big).$$

Daraus erhält man dann leicht $L(G) \subseteq L$.