Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 0 / 04.04.2000

Abgabe (nur Aufgabe 2):
Freitag, 07.04.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Aufgabe 1:
Melden Sie sich bis 06.04.2000, 18:00 Uhr unter

http://www-avenhaus.informatik.uni-kl.de/lehre/uni/gdp2000.html

in einer Übungsgruppe an.

Aufgabe 2:
Zeigen Sie mit vollständiger Induktion für $n \in \mathbb{N}$

$$\sum_{i=0}^{n} i^3 = \frac{n^2 \cdot (n+1)^2}{4}$$

Formulieren Sie Ihren Beweis möglichst präzise.

Aufgabe 3:
Seien A, B und C Teilmengen einer Menge E. Zeigen Sie: Aus $A \cup B \subseteq A \cup C$ und $A \cap B \subseteq A \cap C$ folgt $B \subseteq C$. Geben Sie auch ein Kriterium an, wann $B = C$ gilt.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 1 / 07.04.2000

Abgabe: Freitag, 14.04.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Aufgabe 1:

(a) Welche der beiden Funktionen ist total, injektiv, surjektiv?

(1) $f_1 : \mathbb{N} \times \mathbb{N}_+ \to \mathbb{N}, \quad f_1(x, y) = x \text{ div } y$
(div bezeichne die ganzzahlige Division auf $\mathbb{N}$)

(2) $f_2 : \mathbb{N} \to \mathbb{N}, \quad f_2(x) = \text{minimales } y \text{ mit } y^2 = x.$

Beweisen Sie Ihre Behauptungen.

(b) Zeigen Sie: Es gibt eine bijektive Funktion von $\mathbb{N}$ nach $\mathbb{Z}$.

Aufgabe 2:

Sei $P \subseteq \mathbb{N}_+^2$ ein zweistelliges Prädikat mit

$$P(x, y) \quad \text{gdw} \quad y = 1 \quad \vee \quad x = y \quad \vee \quad y \nmid x,$$

wobei $y \nmid x$ das Prädikat „y ist kein Teiler von x" bezeichnet.
Bestimmen Sie zu P die Prädikate

(a) $Q_1(x) \quad \text{gdw} \quad \bigwedge_{y} P(x, y)$

(b) $Q_2(x) \quad \text{gdw} \quad \bigvee_{y} P(x, y)$

(c) $Q_3(x, z) \quad \text{gdw} \quad \bigvee_{y \leq z} P(x, y)$

Beweisen Sie Ihre Behauptungen!

Aufgabe 3:

Gegeben sei folgender Kalkül: $A = (\mathbb{N} \cup \{(,), +, -\})^+$. Die Regeln sind:

$$\frac{u \quad v}{(u+v)} \qquad \frac{u \quad v}{(u-v)}$$

Sei: $P(w)$ gdw in jedem Anfangswort u von w ist die Anzahl der „(" größer gleich der Anzahl der „)".

Zeigen Sie: Aus $\mathbb{N} \vdash w$ folgt $P(w)$.

Aufgabe 4:

Eine Menge A heißt abzählbar unendlich, falls es eine bijektive Funktion $g : \mathbb{N} \to A$ gibt.
Sei nun $\mathcal{F} = \{f : \mathbb{N} \to \mathbb{N} \mid \mathrm{R(f)} \text{ endlich}, f \text{ total}\}$.
Zeigen oder widerlegen Sie: $\mathcal{F}$ ist abzählbar unendlich.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 2 / 14.04.2000

Abgabe: Donnerstag, 20.04.2000, 14:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Gegeben seien die Signatur $sig = (S, \Sigma, \alpha)$ mit $S = \{INT\}$, $\Sigma = \{0, s, plus\}$, $\alpha(0) = \rightarrow INT$, $\alpha(s) = INT \rightarrow INT$, $\alpha(plus) = INT, INT \rightarrow INT$ und die Variablenmenge $V_{INT} = \{x, y, z\}$.

Sei
$$\begin{array}{rcl} t & \equiv & plus(s(s(x)), y) \\ F & \equiv & \forall x.\, (t = 0 \vee \exists y.\, t = s(z)) \end{array}$$

a) Sei die Substitution σ definiert durch $\sigma = \{x/y, y/s(x), z/plus(x, y)\}$.
Bestimmen Sie $\sigma(t)$ und $\sigma(F)$.

b) Sei die sig-Algebra $\mathcal{A} = ((A_s)_{s \in S}, (f^{\mathcal{A}})_{f \in \Sigma}, \emptyset)$ gegeben durch $A_{INT} = \mathbb{N}^*$, $0^{\mathcal{A}} = \varepsilon$, $s^{\mathcal{A}}(l) = l \circ [1]$ für $l \in \mathbb{N}^*$ und $plus^{\mathcal{A}}(l_1, l_2) = l_1 \circ l_2$ für $l_1, l_2 \in \mathbb{N}^*$. Sei weiterhin $val : V_{INT} \rightarrow A_{INT}$ gegeben durch $val(x) = [1, 2, 3]$, $val(y) = [4, 5]$ und $val(z) = [6]$.
Berechnen Sie $val(t)$.

c) Sei $\mathcal{B} = ((B_s)_{s \in S}, (f^{\mathcal{B}})_{f \in \Sigma}, \emptyset)$ eine sig-Algebra mit $B_{INT} = \mathbb{Z}, 0^{\mathcal{B}} = 0 \in \mathbb{Z}, s^{\mathcal{B}}(n) = n+1$ für $n \in \mathbb{Z}$ und $plus^{\mathcal{B}}(n_1, n_2) = n_1 + n_2$ für $n_1, n_2 \in \mathbb{Z}$. Sei weiterhin $val : V_{INT} \rightarrow B_{INT}$ gegeben durch $val(x) = 1$, $val(y) = 2$, $val(z) = 5$. Zeigen Sie, dass $\mathcal{B} \models_{val} F$ gilt.

Aufgabe 2:

Sei $sig = (\{NAT\}, \{0, s, plus\}, \alpha)$ mit $\alpha(0) = \rightarrow NAT$, $\alpha(s) = NAT \rightarrow NAT$, $\alpha(plus) = NAT, NAT \rightarrow NAT$ und $x, y \in V_{NAT}$.
Weiterhin seien folgende Formeln gegeben:

$$\begin{array}{rcl} A & \equiv & \forall x. \forall y. \,(\, plus(x, 0) = x \wedge plus(x, s(y)) = s(plus(x, y)) \,) \\ B & \equiv & \forall x. \forall y. \,(\, plus(x, y) = plus(y, x) \,) \end{array}$$

Geben Sie eine sig-Algebra $\mathcal{A}$ mit $\mathcal{A} \models A$ und $\mathcal{A} \models B$ an. Geben Sie eine sig-Algebra $\mathcal{B}$ an mit $\mathcal{B} \models A$, aber nicht $\mathcal{B} \models B$.

Aufgabe 3:

Sei *sig* eine Signatur. Wir bezeichnen eine Formel A als *allgemeingültig*, wenn für jede *sig*-Algebra $\mathcal{A}$ gilt, dass $\mathcal{A} \models A$.

Sei nun $sig = (S, \Sigma, \alpha)$ mit $real \in S$, $p, q \in \Sigma$, $\alpha(p) = \alpha(q) = real$ und $x, y \in V_{real}$.
Weiterhin sei

$$\begin{array}{lcllcl} A_1 & \equiv & \forall x.(p(x) \wedge q(x)) & A_2 & \equiv & (\forall x.p(x)) \wedge (\forall x.q(x)) \\ B_1 & \equiv & \forall x.(p(x) \vee q(x)) & B_2 & \equiv & (\forall x.p(x)) \vee (\forall x.q(x)) \end{array}$$

Welche der folgenden Formeln sind allgemeingültig?

$$A_1 \to A_2 \qquad A_2 \to A_1 \qquad B_1 \to B_2 \qquad B_2 \to B_1$$

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 3 / 19.04.2000

Abgabe: Freitag, 28.04.2000, 8:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Gegeben sei folgendes Programm α über den natürlichen Zahlen:

$$\begin{array}{ll} \alpha: & X_0 := 0; \\ & Y \ := 0; \\ & \textsf{while}\ \ Y \neq X_1\ \ \textsf{do} \\ & \quad X_0 := X_0 + 1; \\ & \quad Y \ := Y + 3; \\ & \textsf{end}; \end{array}$$

a) Bestimmen Sie für die Anfangszustände z_1 und z_2 mit $z_1(X_1) = 6$ und $z_2(X_1) = 7$ jeweils $[\![\alpha]\!](z_j)$ und $i(\alpha)(z_j)$, $j = 1, 2$.

b) Welche Funktion $g : \mathbb{N} \to \mathbb{N}$ wird von α berechnet? (ohne Beweis)

Aufgabe 2:

In dieser Aufgabe erweitern wir die Programmiersprache für while-Programme um eine do-while-Schleife (analog z. B. zu Java). Die Intention von

$$\textsf{do}\ \beta\ \textsf{while}\ B;$$

ist, den Schleifenrumpf β zunächst einmal und dann solange durchzuführen, bis die Bedingung B nicht mehr erfüllt ist.

Erweitern Sie geeignet die denotationale Semantik $[\![\alpha]\!]$ und die Interpretersemantik $I(\alpha, z)$ (und damit $i(\alpha)$). Zeigen Sie die Äquivalenz von $[\![\textsf{do}\ \beta\ \textsf{while}\ B;]\!]$ und $i(\textsf{do}\ \beta\ \textsf{while}\ B;)$, indem Sie den Beweis aus der Vorlesung voraussetzen und um den entsprechenden Teil für die do-while-Schleife ergänzen.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 4 / 28.04.2000

Abgabe: Freitag, 05.05.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Gegeben sei folgendes Programm α über der Algebra $\mathcal{A}$ der natürlichen Zahlen (vgl. Aufgabe 1, Blatt 3):

$$
\begin{array}{ll}
\alpha : & X_0 := 0; \\
 & Y \;\; := 0; \\
 & \textsf{while } Y \neq X_1 \textsf{ do} \\
 & \quad X_0 := X_0 + 1; \\
 & \quad Y \;\; := Y + 3; \\
 & \textsf{end};
\end{array}
$$

Geben Sie einen Beweis im Hoare'schen Kalkül an für

$$\vdash_{\mathrm{HC}(\mathcal{A})} \{x_1 = a\}\ \alpha\ \{x_0 = a \textsf{ div } 3\}.$$

Aufgabe 2:

Überprüfen Sie die folgenden Regeln auf Korrektheit. Geben Sie im positiven Fall einen Beweis und im negativen Fall ein Gegenbeispiel an.

a)

$$\frac{\{\varphi \wedge B\}\ \alpha\ \{\varphi\}}{\{\varphi\}\ \textsf{while } B \textsf{ do } \alpha \textsf{ end};\ \{\varphi \wedge \neg B\}}$$

b)

$$\frac{\varphi \rightarrow \psi}{\{\varphi\}\ \alpha\ \{\psi\}}$$

Aufgabe 3:

Gegeben seien folgende Programme α und β über den natürlichen Zahlen:

$$
\begin{array}{ll}
\alpha : & x := x + 2 \cdot y;\ y := x + 1; \\
\beta : & \textsf{while } x \neq y \textsf{ do } x := x + 1;\ \textsf{end};
\end{array}
$$

a) Bestimmen Sie (als Formel) $\mathrm{wlp}(\alpha, \varphi)$ für eine beliebige Formel φ.

b) Bestimmen Sie (als Formeln) $\mathrm{wlp}(\beta, x = y)$ und $\mathrm{wlp}(\beta, x \neq y)$.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 5 / 05.05.2000

Abgabe: Freitag, 12.05.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

(a) Seien $f, g : \mathbb{N}^2 \to \mathbb{N}^2$ definiert durch $f(x, y) = (x \cdot y, y^3)$ und $g(x, y) = (y^x, x^2 + 2x + 1)$. Bestimmen Sie $\langle f, g \rangle$ und $f \circ g$ (ohne Beweis).

(b) Seien $h_0 : \mathbb{N} \to \mathbb{N}$, $g : \mathbb{N} \to \mathbb{N}$ und $h : \mathbb{N}^3 \to \mathbb{N}$ definiert durch

$$h_0(x) = \begin{cases} \frac{x}{2} & \text{falls } x \text{ gerade} \\ \uparrow & \text{sonst} \end{cases}$$

$g(x) = x$ und $h(x, y, z) = h_0(x + z)$.

Bestimmen Sie $f = R(g, h)$ (mit Beweis).

(c) Sei $g : \mathbb{N}^2 \to \mathbb{N}$ definiert durch $g(x, y) = |x^2 - y^4|$.
Bestimmen Sie $f = \mu g$ (mit Beweis).

Aufgabe 2:

Schreiben Sie PL-Programme für folgende Funktionen:

(a) $f_1(x) = [x/2]$

(b) $f_2(x) = \begin{cases} x/2 & \text{falls } x \text{ gerade} \\ \uparrow & \text{sonst} \end{cases}$

Die Lösung für (a) soll sprungfrei sein. Sie können Unterprogramme aus der Vorlesung benutzen. Skizzieren Sie die Korrektheit Ihrer Programme!

Aufgabe 3:

Ein Prädikat P heißt sprungfrei PL-entscheidbar, falls χ_P sprungfrei PL-berechenbar ist. Seien P und Q 2-stellige Prädikate, und sei P_1 definiert durch

$$P_1(x, y) \quad \text{gdw} \quad \bigwedge_{z \leq y} P(x, z)$$

Zeigen Sie:
Sind P und Q sprungfrei PL-entscheidbar, so auch die Prädikate $\neg P$, $P \vee Q$ und P_1.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 6 / 12.05.2000

Abgabe: Freitag, 19.05.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Zeigen Sie, dass folgende Funktionen bzw. Prädikate primitiv rekursiv sind:

a) $f(x) = x^2$

b) $g(x) = [\sqrt{x}]$

c) $P(x, y)$ gdw x ist Potenz von y

d) $Q(x, y)$ gdw y ist die Summe aller Teiler von x

Geben Sie bei Teilaufgabe a) eine genaue Herleitung an. Sie dürfen hierbei nicht verwenden, dass die Multiplikation primitiv rekursiv ist; benutzen Sie stattdessen die Addition. Bei den übrigen Teilaufgaben dürfen Sie sämtliche Ergebnisse der Vorlesung verwenden.

Aufgabe 2:

Bestimmen Sie zu der Funktion $g : \mathbb{N}^3 \to \mathbb{N}$ mit $g(x_1, x_2, y) = (x_1 \dot{-} y) \cdot (x_2 \dot{-} y)$ die Funktion $f(x_1, x_2, y) = \mu_{z \leq y}\ [g(x_1, x_2, z) = 0]$.

Aufgabe 3:

Zeigen oder widerlegen Sie:
Sei $f : \mathbb{N} \to \mathbb{N}$ injektiv. Ist f rekursiv, so ist f^{-1} partiell rekursiv.

Zusatzaufgabe: (nicht abzugeben, Besprechung am Montag, 15.05.2000)

a) Sei $f_1(x, y) = \max(x, y)$. Zeigen Sie $f_1 \in \mathcal{P}$ (mit genauer Herleitung).

b) Sei $f_2(x) = \begin{cases} \sqrt{x} & \text{falls } x \text{ Quadratzahl} \\ \uparrow & \text{sonst} \end{cases}$
Zeigen Sie $f_2 \in \mathcal{R}_\text{p}$.

c) Bestimmen Sie zu der Funktion $g : \mathbb{N}^2 \to \mathbb{N}$ mit $g(x, y) = 4x \dot{-} (2y + 1)$ die Funktion $f(x, y) = \mu_{z \leq y}\ [g(x, z) = 0]$.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 7 / 19.05.2000

Abgabe: Freitag, 26.05.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Das folgende PL-Programm P mit Eingabevariable $X1$ und Ausgabevariable $X2$ berechnet die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit $f(x) = \sum_{i=1}^{x} i$:

$$
\left.\begin{array}{l}
\left.\begin{array}{ll} X2 \leftarrow 0; & \}\,P_1 \\ X3 \leftarrow 0; & \}\,P_2 \end{array}\right\} P_3 \\
\left.\begin{array}{l}
\textsf{loop}\ X1; \\
\quad \left.\begin{array}{l} X3 \leftarrow X3+1; \quad \}\,P_4 \\ \left.\begin{array}{l}\textsf{loop}\ X3; \\ \quad X2 \leftarrow X2+1; \quad \}\,P_5 \\ \textsf{end};\end{array}\right\} P_6 \end{array}\right\} P_7 \\
\textsf{end};
\end{array}\right\} P_8
\end{array}\right\} P_9 = P
$$

Geben Sie nach dem Verfahren aus der Vorlesung (Folie 74) primitiv rekursive Funktionen p_{P_i} für $i = 1, \ldots, 9$ und damit ein $\mathcal{P}$-Programm für f an.

Aufgabe 2:

(a) Zeigen Sie die folgenden Abschätzungen für die Ackermann-Funktion:

  (i) $y < A(x, y)$

  (ii) $A(x, y) < A(x, y + 1)$

  (iii) $A(x, y + 1) \leq A(x + 1, y)$

(b) Sei $f : \mathbb{N} \to \mathbb{N}$ definiert durch
$$f(x) = A(x, 0).$$
Gilt $f \in \mathcal{P}$?

Aufgabe 3:

Zeigen oder widerlegen Sie:
Es gibt eine Funktion $f : \mathbb{N} \to \mathbb{N}$ mit

(i) $f \in \mathcal{P}$ und

(ii) für alle $n \in \mathbb{N}$ ist $U_n := \{x | f(x) = n\}$ unendlich.

Zusatzaufgabe: (Besprechung am Montag, 22.05.2000, 17.15 Uhr in Raum 46-215)

(a) Zeigen Sie folgende Gleichungen für die Ackermann-Funktion:

   (i) $A(1, y) = y + 2$

   (ii) $A(2, y) = 2y + 3$

(b) Sei $f : \mathbb{N} \to \mathbb{N}$ definiert durch

$$f(x) = [\sqrt{A(x, x)}].$$

Gilt $f \in \mathcal{P}$?

(c) Sei $f : \mathbb{N} \to \mathbb{N}$ definiert durch

$$f(x) = \mu_y[A(y, x) \dot{-} A(x, y) = 0].$$

Gilt $f \in \mathcal{P}$?

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 8 / 26.05.2000

Abgabe: Freitag, 02.06.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:
Sei $\Sigma = \{1, 2\}$. Schreiben Sie ein RM-Unterprogramm für

$$Z \Leftarrow Z \dot{-} 1.$$

Alle Hilfsvariablen haben beim Start und Stop den Wert ε.

Aufgabe 2:
Sei $\Sigma = \{1, 2\}$. Die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit

$$f(x) = \begin{cases} x/2 & x \text{ gerade} \\ \uparrow & \text{sonst} \end{cases}$$

wird durch folgendes PL-Programm berechnet:

$$\begin{array}{ll} & Y \leftarrow 0;\ V \leftarrow X; \\ W: & \textbf{loop } V; \\ & \quad V \Leftarrow V \dot{-} 1; \\ & \quad \textbf{loop } V; \\ & \quad\quad V \Leftarrow V \dot{-} 1; Y \leftarrow Y + 1; \\ & \quad\quad \textbf{goto } W; \\ & \quad \textbf{end}; \\ L: & \quad \textbf{goto } L; \\ & \textbf{end}; \end{array}$$

(a) Welches RM-Programm erzeugt der Compiler aus der Vorlesung?
Hinweis: Sie dürfen Schritt 4, also die Beseitigung der RM-Unterprogramme, weglassen. Außerdem dürfen Sie für das PL-Unterprogramm $V \dot{-} 1$ das entsprechende RM-Unterprogramm verwenden.

(b) Geben Sie ein effizientes RM-Programm für f an.

Aufgabe 3:
Sei $\Sigma = \{a_1, a_2\}$. Geben Sie eine Turingmaschine (kein Turing-Programm!) für die Berechnung der folgenden Funktion an:

$$f : W \to W \qquad \text{mit} \quad f(x) = a_1^{|x|_{a_1}},$$

wobei $|x|_{a_1}$ die Anzahl der a_1's in x bezeichnen.

Zusatzaufgabe: (Besprechung am Montag, 29.05.2000, 17.15 Uhr in Raum 46-210)

a) Sei $\Sigma = \{a, b\}$ und $f: W \to W$ mit

$$f(x) = \begin{cases} a^n & \text{falls } |x| \text{ gerade, } |x| = 2n \\ \uparrow & \text{falls } |x| \text{ ungerade} \end{cases}$$

Schreiben Sie ein RM-Programm, das f berechnet. Skizzieren Sie die Korrektheit Ihres Programms.

b) Sei $\Sigma = \{a_1, a_2\}$. Geben Sie eine Turingmaschine für die Berechnung der folgenden Funktion an:

$$f : W^3 \to W \qquad \text{mit} \qquad f(x, y, z) = \pi_2^3(x, y, z) = y.$$

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 9 / 02.06.2000

Abgabe: Freitag, 09.06.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:
Zeigen oder widerlegen Sie:
Es gibt eine rekursive universelle Funktion $\psi^{1,1} : W^2 \to W$ für $\mathcal{R}^{1,1}$, d. h.
$\psi^{1,1} \in \mathcal{R}^{2,1}$ und
zu jedem $f \in \mathcal{R}^{1,1}$ gibt es einen Index $p \in W$, so dass
$f(x) = \psi^{1,1}(p, x)$ für alle $x \in W$.

Aufgabe 2:
Sei ψ eine zulässige Algorithmenaufzählung.

a) Zeigen Sie, dass es ein $f \in \mathcal{P}$ gibt mit $\psi^{1,1}_{f(x,y)}(z) = \psi^{1,1}_y(z) \cdot \psi^{1,1}_x(z)$.

b) Zeigen Sie, dass es ein n gibt mit $\psi^{1,1}_n(x) = x^2 + nx + 1$.

Aufgabe 3:
Zeigen Sie folgende Behauptungen:

a) Ist $f \in \mathcal{R}^{1,1}$, $f(x) \geq x$ für alle $x \in \mathbb{N}$ und $A = R(f)$, so ist A rekursiv.

b) Es gibt ein rekursives zweistelliges Prädikat Q, so dass das folgende einstellige Prädikat P nicht rekursiv-aufzählbar ist:
$$P(x) \quad \text{gdw} \quad \bigwedge_y Q(x, y)$$

Zusatzaufgabe: (Besprechung am Montag, 05.06.2000, 17.15 Uhr in Raum 46-210)

a) Sei ψ eine zulässige Algorithmenaufzählung.

   i) Sei $g \in \mathcal{P}$. Zeigen Sie, dass es ein $f \in \mathcal{P}$ gibt mit $\psi^{1,1}_{f(x)}(y) = g(x) \cdot \psi^{1,1}_x(y)$.

   ii) Zeigen Sie, dass es ein n gibt mit $\psi^{1,1}_n(x) = n$.

b) Zeigen Sie, dass es ein rekursives zweistelliges Prädikat Q gibt, so dass das folgende einstellige Prädikat P nicht rekursiv ist:
$$P(x) \quad \text{gdw} \quad \bigvee_y Q(x, y)$$

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 10 / 09.06.2000

Abgabe: Freitag, 23.06.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:
Sei ψ eine zulässige Algorithmenaufzählung und $A = \{p \mid \psi_p^{1,1} \text{ total}\}$.
Zeigen Sie: Weder A noch $\overline{A}$ sind rekursiv-aufzählbar.

Aufgabe 2:
Untersuchen Sie, ob $A \subseteq \mathbb{N}^2$ rekursiv oder rekursiv-aufzählbar ist.
Bestimmen Sie im positiven Fall Entscheidungs- bzw. Aufzählungsfunktionen für A.

(a) $A = \{(x, y) \mid \text{es gibt ein } z \text{ mit } x < z^2 < y\}$

(b) $A = \{(x, y) \mid \psi_x^{1,m}(y) \uparrow\}$

Aufgabe 3:

(a) Sei $G = (N, \Sigma, P, S)$ folgende Grammatik:
$N = \{S, T\}, \ \Sigma = \{a, b, c\}, \ P = \{S \to aTb, \ T \to bSa, \ S \to c\}$
Welche Sprache $L = L(G)$ wird von G erzeugt?
Beweisen Sie Ihre Behauptung.

(b) Sei $\Sigma = \{a, b\}$, ρ die Spiegelungsfunktion (d.h. $\rho(\varepsilon) = \varepsilon$ und $\rho(wx) = x\rho(w)$ für $x \in \Sigma, \ w \in \Sigma^*$) und $L = \{x\rho(x) \mid x \in \Sigma^+\}$.
Geben Sie eine kontextfreie Grammatik G an, die L erzeugt.
Zeigen Sie: $L = L(G)$ für Ihre Grammatik.

Zusatzaufgabe: (Besprechung voraussichtlich am Dienstag, 20.06.2000, 17.15 Uhr; Näheres wird rechtzeitig in der Vorlesung und im WWW bekanntgegeben)

a) Sei $n \in \mathbb{N}$, ψ eine zulässige Algorithmenaufzählung und $A = \{p \mid n \in R(\psi_p)\}$.
Zeigen oder widerlegen Sie:
1) A ist rekursiv
2) A ist rekursiv aufzählbar
3) $\overline{A}$ ist rekursiv aufzählbar

b) Sei $\Sigma = \{a, b\}$ und $L = \{a^n b a^n \mid n > 0\}$.
Geben Sie eine kontextfreie Grammatik G an, die L erzeugt.
Zeigen Sie: $L = L(G)$ für Ihre Grammatik.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 11 / 23.06.2000

Abgabe: Freitag, 30.06.2000, 08:00 Uhr in Geb. 34, 4. Stock (Kasten)

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Sei $G = (N, \Sigma, P, S)$ eine kontextfreie Grammatik. $A \in N$ heißt *überflüssig*, falls es keine Ableitung $S \stackrel{*}{\Longrightarrow} uAv \stackrel{*}{\Longrightarrow} w$ gibt mit $u, v \in V^*$ und $w \in \Sigma^*$.

(a) Geben Sie ein Verfahren an zur Berechnung von $N_0 = \{A \in N \mid A \text{ überflüssig}\}$.

Tip: Berechnen Sie zunächst $N_1 = \{A \in N \mid A \stackrel{*}{\Longrightarrow} w\ ,\ \ w \in \Sigma^*\}$ und
$N_2 = \{A \in N \mid S \stackrel{*}{\Longrightarrow} uAv\ ,\ \ u, v \in V^*\}$.

(b) Es entstehe G' aus G durch Streichen aller überflüssigen Regeln, d.h. aller Regeln, die ein Nonterminal aus N_0 enthalten. Zeigen Sie: $L(G) = L(G')$.

Aufgabe 2:

Moderne Compiler besitzen oft eine Komponente, die sogenannten »toten Code« entfernt. Unter »totem Code« versteht man Programmfragmente, die nie ausgeführt werden. Sie sollen hier die Frage untersuchen, ob »toter Code« immer erkannt werden kann.
Zeigen oder widerlegen Sie die Entscheidbarkeit des folgendes Problems:

Eingabe: Eine Turing-Maschine $M = (\Sigma, Q, q_0, T)$ und ein Zustand $q_i \in Q$

Frage: Wird q_i jemals aktuell bei einer beliebigen Eingabe x ?

Aufgabe 3:

Prüfen Sie, ob $PCP(A, B)$ gilt für

(a) $A:$ 12, 212, 122, 12, 121
$B:$ 121, 12, 222, 221, 211

(b) $A:$ 1, 1211, 112, 1
$B:$ 12, 22, 121, 112

(c) $A:$ 212, 122, 11, 2
$B:$ 21, 112, 21, 212

Zusatzaufgabe: (Besprechung am Montag, 26.06.2000, 17.15 Uhr in Raum 46-210)

(a) Sei $G = (N, \Sigma, P, S)$ eine Grammatik vom Typ 0 in N-Normalform und $G' = (N', \Sigma, P', S')$ eine disjunkte Kopie.
Zeigen Sie: Ist $v = v_1 v_2 v_3$ mit $v_1, v_3 \in (\Sigma \cup N)^*$, $v_2 \in (\Sigma \cup N')^*$ und gilt $v \overset{*}{\Longrightarrow}_{G \cup G'} w$, so gibt es eine Zerlegung $w = w_1 w_2 w_3$ mit $w_1, w_3 \in (\Sigma \cup N)^*$, $w_2 \in (\Sigma \cup N')^*$.

(b) Geben Sie einen Entscheidungsalgorithmus an für das folgende Problem:

Eingabe: Grammatik G von Typ 2

Frage: Ist $L(G) = \emptyset$?

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern
Sommersemester 2000
Blatt 12 / 30.06.2000

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

Zeigen oder widerlegen Sie:

(a) Das PCP über $\Sigma = \{a\}$ ist entscheidbar.

(b) Das PCP über $\Sigma = \{a, b\}$ ist entscheidbar.

Aufgabe 2:

Seien $\Sigma = \{a, b\}$ und L_1, L_2, L_3 definiert durch

$$L_1 = \{(ab)^i \mid i \geq 0\} \qquad L_2 = \{ab^ia \mid i \geq 0\} \qquad L_3 = L_1 \cdot L_2$$

Geben Sie jeweils zu L_1 und L_2 einen Endlichen Automaten und zu L_3 einen Nichtdeterministischen Endlichen Automaten an, der die Sprache erkennt.

Aufgabe 3:

a) Zeigen Sie, dass die Sprache $L = \{x\rho(x) \mid x \in \{a, b\}^+\}$ aus Aufgabe 3b, Blatt 10 nicht vom Typ 3 ist.

b) Bestimmen Sie einen Kellerautomaten, der L akzeptiert.

Übungen zur Vorlesung

Grundlagen der Programmierung

Prof. Dr. J. Avenhaus
Universität Kaiserslautern

Sommersemester 2000

Blatt 13 / 07.07.2000

Informationen zur Vorlesung (auch von außerhalb der Universität zugreifbar):
http://www-avenhaus.informatik.uni-kl.de/lehre/gdp2000.html

Aufgabe 1:

(a) Sei $\Sigma = \{a, b, c, d\}$. Zeigen Sie:
Die Sprache $L_1 = \{a^n b^m c^n d^m \mid n, m \geq 0\}$ ist nicht kontextfrei.

(b) Sei $\Sigma = \{a\}$. Zeigen Sie:
Die Sprache $L_2 = \{a^q \mid q \text{ Quadratzahl }\}$ ist nicht kontextfrei.

Aufgabe 2:

Sei $\Sigma = \{a, b, c\}$ und $G = (N, \Sigma, P, S)$ mit

$$\begin{array}{llllll} P: & S \to S_1 & S_1 \to aS_1 & S_1 \to bT_1c & T_1 \to bT_1c & T_1 \to \varepsilon \\ & S \to S_2 & S_2 \to S_2c & S_2 \to aT_2b & T_2 \to aT_2b & T_2 \to \varepsilon \end{array}$$

(a) Transformieren Sie G in eine äquivalente Grammatik G' in Chomsky-Normalform.

(b) Untersuchen Sie mit Hilfe des Verfahrens von Cocke, Younger und Kasami, für welche Teilwörter w von $a^2b^2c^2$ gilt: $w \in L(G') = L(G)$.