Technische Universität München
Institut für Informatik
PD Dr. Markus Holzer
Dipl.-Inf. Michael Tautschnig

WS 2007/2008
Übungsblatt 6
Mittwoch, 30. November 2007

Übung zur Vorlesung Automaten, Formale Sprachen und Berechenbarkeit

Lösungshinweise

Aufgabe 1 *Akzeptable Programmiersysteme (1)*

Man bilde die Aufzählung $A_0, B_0, A_1, B_1, A_2, B_2, \ldots$; diese ist wiederum ein akzeptables Programmiersystem, da:

- Jede berechenbare Funktion kommt sowohl in $A_0, A_1, \ldots$ als auch in $B_0, B_1, \ldots$ vor; insbesondere kommt jede berechenbare Funktion auch in unserer Aufzählung vor.
- Laut Voraussetzung gibt es eine universelle Funktion $\alpha_u^{(2)}$ für die A_i's und eine universelle Funktion $\beta_u^{(2)}$ für die B_i's; insbesondere hat $\beta_u^{(2)}$ jedoch auch einen Index k in den A_i's, d.h. $\beta_u^{(2)} = \alpha_k^{(2)}$. Es existiert daher eine universelle Funktion $\gamma^{(2)}$ für unsere Aufzählung:

  $$\gamma^{(2)}(x,y) = \begin{cases} \alpha_u(\frac{x}{2}, y) & \text{falls } x \text{ gerade} \\ \alpha_k(\frac{x-1}{2}, y) & \text{falls } x \text{ ungerade.} \end{cases}$$

  Beachte, dass $x/2$ und $(x-1)/2$ berechenbare Funktionen sind, daher ist auch $\gamma^{(2)}$ berechenbar.
- Eine ähnliche Argumentation wie im obigen Fall zeigt, dass auch die s-m-n Eigenschaft für unsere Aufzählung gilt.

Aus diesen drei Eigenschaften folgt, dass das Rekursionstheorem für die Aufzählung $A_0, B_0, A_1, B_1, A_2, B_2, \ldots$ gilt. Wir bezeichnen die Aufzählung dieser Funktionen ab jetzt mit $\gamma_0, \gamma_1, \gamma_2, \ldots$ Definiere

$$f(n) = \begin{cases} n+1 & \text{falls } n \text{ gerade} \\ n-1 & \text{falls } n \text{ ungerade.} \end{cases}$$

Aus dem Rekursionstheorem folgt, dass es einen Index m gibt mit $\gamma_m = \gamma_{f(m)}$. Falls m gerade ist, gilt sowohl $\gamma_m = \alpha_{m/2}$ als auch $\gamma_{f(m)} = \gamma_{m+1} = \beta_{m/2}$. Der Fall m ungerade ist analog, d.h. $\gamma_m = \beta_{(m-1)/2}$ und $\gamma_{f(m)} = \gamma_{m-1} = \alpha_{(m-1)/2}$. Der gesuchte Index existiert daher.

Aufgabe 2 *Akzeptable Programmiersysteme (2)*

(i) Seien n_0 und n_1 die Indices der konstanten Funktionen $f(x) = 0$ und $f(x) = 1$. Das gesuchte akzeptable Programmiersystem ist etwa

$$\alpha_{n_0}, \alpha_0, \alpha_{n_1}, \alpha_{n_0}, \alpha_1, \alpha_{n_1}, \alpha_{n_0}, \alpha_2, \alpha_{n_1}, \ldots$$

(ii) Man betrachte das System

$$\varphi_0, \varphi_1, \varphi_0, \varphi_0, \varphi_1, \varphi_1, \varphi_2, \varphi_0, \varphi_0, \varphi_0, \varphi_1, \varphi_1, \varphi_1, \varphi_2, \varphi_2, \varphi_3, \ldots$$

In dieser Aufzählung kommen beliebig lange Ketten jeder berechenbaren Funktion φ vor.

Für beide Aufzählungen muss man noch nachweisen, dass sie akzeptable Programmiersysteme sind; dies geschieht analog zur vorigen Aufgabe.

Aufgabe 3 *Primitiv Rekursive Funktionen*

(i) $\mathit{plus2}(x) = s(s(x))$

(ii) Wir definieren die Funktion $\mathit{plus}(x, y)$ durch

$$\begin{aligned} \mathit{plus}(k, 0) &= u_1^1(k), \\ \mathit{plus}(k, j+1) &= s(u_3^3(k, j, \mathit{plus}(k, j))). \end{aligned}$$

(iii) Wir definieren die Funktion $\mathit{mul}(x, y)$ durch

$$\begin{aligned} \mathit{mul}(k, 0) &= 0, \\ \mathit{mul}(k, j+1) &= f(k, j, \mathit{mul}(k, j)), \end{aligned}$$

wobei $f(x, y, z) = \mathit{plus}(u_1^3(x, y, z), u_3^3(x, y, z))$.

Aufgabe 4 *Primitive Rekursion und LOOP-Berechenbarkeit*

Dieser Beweis findet sich in jedem klassischen Buch über Berechenbarkeitstheorie (z.B. Engeler/Läuchli, "Berechnungstheorie für Informatiker", Teubner) und etwa auch im Skriptum zur Vorlesung "Berechenbarkeit und Entscheidbarkeit" von Prof. Esparza (WS 1995/96) ab Seite 26, das auf der Homepage des Lehrstuhls zum Download bereitsteht (`http://wind.in.tum.de/lehre/berech/WS9596/`).