Übungsblatt 3

Berechenbarkeitstheorie
Wintersemester 2012/13
Universität Hamburg

Schriftliche Abgabe am Anfang der Übung am 15. November 2012.

Hinweis. Die Vorlesung am 13. November fällt aus.

1. Sei Fml die Menge aller Formeln und $\ulcorner . \urcorner : \mathrm{Fml} \to \mathbb{N}$ eine geeignete Kodierung von Formeln durch natürliche Zahlen (wie in der ersten Woche der Vorlesung). Die Funktion $\mathsf{decode} : \mathbb{N} \to \mathrm{Fml}$ sei die Umkehrfunktion von $\ulcorner . \urcorner$. Zusammen mit einer Kodierung von Folgen erhalten wir eine Kodierung von n-Tupeln von Formeln $\mathsf{G}_n : \mathrm{Fml}^n \to \mathbb{N}$ und ihre Umkehrung $\mathsf{decode}_n : \mathbb{N} \to \mathrm{Fml}^n$.

   Wir nennen eine Menge $R \subseteq \mathrm{Fml}^n \times \mathrm{Fml}$ eine *n-Regel* und wollen sagen, dass eine n-Regel *berechenbar* sei, falls die Menge

   $$\{(k, \ell)\ ;\ (\mathsf{decode}_n(k), \mathsf{decode}(\ell)) \in R\}$$

   berechenbar ist.

   Eine endliche Menge $\mathcal{R} = \{R_0, ..., R_N\}$ heiße *Regelsystem*, falls jedes R_i eine berechenbare k-Regel ist (für ein k). Eine endliche Folge $(\varphi_0, ..., \varphi_n)$ nennen wir einen *$\mathcal{R}$-Regelhaftigkeitszeugen für φ_n* falls für jedes $i \leq n$ eine ℓ-Regel $R_k \in \mathcal{R}$ und $i_0, ..., i_{\ell-1} < i$ existieren, so dass

   $$((\varphi_{i_0}, ..., \varphi_{i_{\ell-1}}), \varphi_i) \in R_k.$$

   Eine Formel φ heiße *$\mathcal{R}$-regelhaftig*, falls es einen $\mathcal{R}$-Regelhaftigkeitszeugen für φ gibt.

   (a) Zeigen Sie, dass für jedes Regelsystem $\mathcal{R}$ die Menge der $\mathcal{R}$-Regelhaftigkeitszeugen berechenbar ist. (Zunächst muß diese Aussage präzise gemacht werden: beachten Sie, dass die Menge der Regelhaftigkeitszeugen keine Teilmenge von $\mathbb{N}^k$ ist!)

   (b) Zeigen Sie, dass für jedes Regelsystem $\mathcal{R}$ die Menge der $\mathcal{R}$-regelhaften Formeln berechenbar aufzählbar (c.e.) ist.

2. Sei $\{x_i\ ;\ i \in \mathbb{N}\}$ eine abzählbare Menge von Variablen. Die Menge aller *LOOP-Programme* ist die kleinste Menge von Zeichenketten, die die folgenden Bedingungen erfüllt:

   - Für natürliche Zahlen i, j und c, ist $x_i := x_j + c$ ein LOOP-Programm;
   - für natürliche Zahlen i, j und c, ist $x_i := x_j - c$ ein LOOP-Programm;
   - falls P und Q LOOP-Programme sind, so auch $P; Q$;
   - für eine natürliche Zahl i und ein LOOP-Programm P ist

     $$\texttt{LOOP}\ x_i\ \texttt{DO}\ P\ \texttt{END}$$

     ein LOOP-Programm.

   Eine Funktion $V : \mathbb{N} \to \mathbb{N}$ heiße *Zuweisung* und wir interpretieren Sie als Zuweisung des Wertes $V(i)$ zur Variablen x_i. Falls P ein LOOP-Programm ist, so definieren wir die Anwendung von P auf eine Zuweisung V:

   - Die Befehlszeile $x_i := x_j + c$ setzt den Wert der Variablen x_i gleich dem Wert $V(j) + c$;
   - die Befehlszeile $x_i := x_j - c$ setzt den Wert der Variablen x_i gleich dem Wert $V(j) \dot{-} c$;
   - die Befehlszeile $\texttt{LOOP}\ x_i\ \texttt{DO}\ P\ \texttt{END}$ führt die Befehle des Programms P insgesamt $V(i)$-mal aus.

   Ein LOOP-Programm definiert für jede Anfangszuweisung also eine Folge von Zuweisungen $(V_k\ ;\ k \in I)$, die wir die *Berechnung* des Programms nennen wollen. Die Zuweisung

   $$C_x(i) := \begin{cases} x & \text{falls } i = 0, \\ 0 & \text{sonst} \end{cases}$$

   werde interpretiert als die Eingabe des Wertes x. Eine Funktion $f : \mathbb{N} \to \mathbb{N}$ heisse *LOOP-berechenbar* falls es ein LOOP-Programm P gibt, welches bei Anfangszuweisung C_x eine endliche Berechnung hat, deren letzte Zuweisung $C_{f(x)}$ ist.

   Beweisen Sie, dass jede primitiv rekursive Funktion LOOP-berechenbar ist.