Sommersemester 2012, Grundbegriffe der mathematischen Logik

Übungsblatt 1

Aufgabe 1 L_k bezeichne die Menge der Funktionen von $f : \mathbb{N}^k \to \mathbb{N}$, so daß$f$ gegeben ist durch $f(x_1, \ldots, x_k) = \sum_{i=1}^k n_i \cdot x_i$ für gewisse $n_1, \ldots, n_k \in \mathbb{N}$. Zeigen Sie:

(a) $\bigcup_{k\in\mathbb{N}} L_k$ ist die kleinste Menge $F \subseteq \bigcup_{k\in\mathbb{N}} \mathbb{N}^{\mathbb{N}^k}$, die die konstante Nullfunktion, die zweistellige Addition $+$ und die Projektionsfunktionen π_i^k enthält und unter Zusammensetzung abgeschlossen ist.

(b) Geben Sie eine ähnliche Beschreibung der kleinsten Menge $F \subseteq \bigcup_{k\in\mathbb{N}} \mathbb{N}^{\mathbb{N}^k}$, die die konstante Nullfunktion, die zweistellige Addition $+$, die zweistellige Multiplikation $\cdot$ und die Projektionsfunktionen π_i^k enthält und unter Zusammensetzung abgeschlossen ist.

Aufgabe 2 Eine Funktion $f : \mathbb{N} \to \mathbb{N}$ heisse *polynomiell beschränkt*, wenn es ein Polynom p gibt, so daß $f(n) \leq p(n)$ für genügend große $n \in \mathbb{N}$. Zeigen Sie: für alle $c, d \in \mathbb{N}$ ist die durch

$$\begin{aligned} h(0) &= d \\ h(x+1) &= c \cdot h(\lfloor x/2 \rfloor) \end{aligned}$$

gegebene Funktion $h : \mathbb{N} \to \mathbb{N}$ polynomiell beschränkt (hier steht $\lfloor \cdot \rfloor$ für "Abrunden").

Aufgabe 3 Seien $f, g, g' : \mathbb{N}^k \to \mathbb{N}$ primitiv rekursiv. Zeigen Sie, daß dann auch die durch

$$h(x_1, \ldots, x_k) = \begin{cases} g(x_1, \ldots, x_k) & \text{falls } f(x_1, \ldots, x_k) > 0 \\ g'(x_1, \ldots, x_k) & \text{sonst} \end{cases}$$

gegebene Funktion $h : \mathbb{N}^k \to \mathbb{N}$ primitiv rekursiv ist.

Hinweis: Lösen Sie zuerst Übungsaufgabe 1.3 im Skript und zeigen dann, daß die durch $sgn(x) = \begin{cases} 0 & \text{falls } x = 0 \\ 1 & \text{falls } x > 0 \end{cases}$ gegebene Funktion primitiv rekursiv ist.

Aufgabe 4 Schreiben Sie ein LOOP-Programm, das die durch $f(x,y) = x^y$ gegebene Funktion berechnet.

Aufgabe 5 Zeigen Sie, daß es zu jedem LOOP-Programm ein LOOP-Programm gibt, das dieselbe Funktion berechnet und in dem weder der `Val`-Befehl noch der `Zero`-Befehl vorkommt.