Berechenbarkeitstheorie

Prof. Dr. Christoph Walther / Nathan Wasser

Technische Universität Darmstadt — Sommersemester 2011

Übung 1

Aufgabe 1.1 intuitive Berechenbarkeit

Sind die folgenden Funktionen im intuitiven Sinne berechenbar? Für welche kann man sogar einen Algorithmus explizit angeben? Dabei seien $g : \mathbb{N} \to \mathbb{N}$ eine totale und berechenbare Funktion und $h : \mathbb{N} \mapsto \mathbb{N}$ eine nicht berechenbare Funktion.

1. $f_1(n) = \begin{cases} 1 & \text{, falls } n \text{ eine Primzahl ist} \\ 0 & \text{, sonst} \end{cases}$

2. $f_2(n) = \begin{cases} 1 & \text{, falls es eine Primzahl } p \geq n \text{ gibt} \\ 0 & \text{, sonst} \end{cases}$

3. $f_3(n) = \begin{cases} 1 & \text{, falls es einen Primzahlzwilling } (p, p+2) \text{ mit } p \geq n \text{ gibt} \\ 0 & \text{, sonst} \end{cases}$

   Hinweis: $(p, p+2)$ heißt *Primzahlzwilling*, falls sowohl p als auch $p+2$ Primzahlen sind. Es ist nicht bekannt, ob es unendlich viele Primzahlzwillinge gibt.

4. $f_4(n) = \begin{cases} 1 & \text{, falls } g(n) = n \\ 0 & \text{, sonst} \end{cases}$

5. $f_5(n) = \begin{cases} 0 & \text{, falls } n = 0 \\ h(n) & \text{, sonst} \end{cases}$

6. $f_6(n) = \begin{cases} 0 & \text{, falls } n = 0 \\ h(f_6(n-1)) & \text{sonst} \end{cases}$

7. $f_7(n) = \begin{cases} n & \text{, falls es ein } m \text{ gibt mit } h(m) \leq m \\ 0 & \text{, sonst} \end{cases}$

Aufgabe 1.2 Programmiersprache $\mathcal{P}$

Schreiben Sie ein $\mathcal{P}$-Programm zur Berechnung von $n!$ für beliebige natürliche Zahlen n. Verwenden Sie die erweiterte Syntax gemäß den Definitionen 2.2 und 2.3.

Aufgabe 1.3 Semantik von $\mathcal{P}$

1. Bestimmen Sie $\mathit{value}(M_P, \mathtt{E})$ für die folgenden Ausdrücke `E` unter einer beliebigen Speicherbelegung M_P in der Form wie in Definition 2.6 angegeben. Hierfür werden Sie Definition 2.7 und die Abkürzungen in Definitionen 2.2 und 2.3 benötigen.

   a) $\mathtt{E} = \neg\mathtt{EXPR}$

   b) $\mathtt{E} = \mathtt{EXPR1} \vee \mathtt{EXPR2}$

   c) $\mathtt{E} = \mathtt{EXPR1} \wedge \mathtt{EXPR2}$

2. Bestimmen Sie $\mathit{eval}(M_P, \mathtt{S})$ für die folgenden Anweisungen `S` unter einer beliebigen Speicherbelegung M_P in der Form wie in Definition 2.7 angegeben. Auch hier benötigen Sie Definition 2.3.

   d) `S` = `if EXPR then STA end_if`

   e) `S` = `repeat STA until EXPR`