Prof. Carsten Lutz/Dr. Stefan Göller/Dr. Thomas Schneider SS 2011

Theoretische Informatik 2

Gewertete Aufgaben, Blatt 4

Abgabe:Bis 6.6.11 ins Postfach Ihrer Tutorin/Ihres Tutors *Besprechung: KW 23*

Hinweis: In diesem Aufgabenblatt dürfen sowohl bei **LOOP**-Programmen als auch bei **WHILE**-Programmen folgende Konstrukte verwendet werden, wobei x_i, x_j und x_k Variablen sind und P ein Programm ist:

- $x_i := x_j + x_k$
- $x_i := x_j \dot{-} x_k$
- $x_i := x_j * x_k$
- **IF** $x_i = x_j$ **THEN** P **ENDIF**

1. (30%=10%+10%+10%) Geben Sie für folgende Funktionen **LOOP**-Programme an (für Aufgabe b) dürfen Sie Aufgabe a) und für Aufgabe c) dürfen Sie Aufgabe b) als Unterprogramm verwenden):

a) $\mathrm{teilt} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit

$$\mathrm{teilt}(x,y) = \begin{cases} 0 & \text{falls } x = 0 \text{ oder } (x > 0 \text{ und } x \text{ teilt } y \text{ nicht}), \\ 1 & \text{falls } x > 0 \text{ und } x \text{ teilt } y. \end{cases}$$

b) $\mathrm{tzahl} : \mathbb{N} \to \mathbb{N}$ mit

$$\mathrm{tzahl}(x) = \sum_{i=0}^{x} \mathrm{teilt}(i,x).$$

c) $\mathrm{prim} : \mathbb{N} \to \mathbb{N}$ mit

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

2. (10%) Die n-te Fibonacci-Zahl F_n ist rekursiv wie folgt definiert:

$$F_n = \begin{cases} 1 & \text{falls } n \in \{0,1\}, \\ F_{n-1} + F_{n-2} & \text{falls } n \geq 2. \end{cases}$$

Geben Sie ein **LOOP**-Programm an, das die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit $f(n) = F_n$ berechnet.

3. (30%=20%+10%) Gegeben sei die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit

$$f(n) = \max\{k \in \mathbb{N} \mid k \leq \log_{10}(n)\}.$$

Hinweis: Die Funktion f ist (nur) bei 0 undefiniert.

a) Geben Sie ein **WHILE**-Programm an, das f berechnet.

b) Notieren Sie f als μ-rekursive Funktion. Dabei dürfen Sie alle in der Vorlesung als primitiv/μ-rekursiv nachgewiesene Funktionen als μ-rekursiv voraussetzen.

4. (30%=10%+10%+10%) Welche der folgenden Aussagen sind wahr? Begründen Sie Ihre Antwort.

a) Ist eine Funktion f nicht **LOOP**-berechenbar, dann ist f auch keine totale berechenbare Funktion.

b) Es gibt μ-rekursive Funktionen, die nicht **LOOP**-berechenbar sind.

c) Es gibt ein $i \in \{1, 2, 3\}$ so dass die Funktion $f : \mathbb{N} \to \mathbb{N}$ mit $f(n) = A(i, n)$ **LOOP**-berechenbar ist. Dabei bezeichnet A die Ackermann-Funktion.