Theoretische Informatik II

Prof. Christoph Kreitz / Nuria Brede
Universität Potsdam, Theoretische Informatik — Sommersemester 2009

Übungsblatt 3 (Version 1) — Abgabetermin: Dienstag, 19.05.09, 16:00 Uhr

---

Zur Vorbereitung ist es sinnvoll, sich mit folgenden Fragen auseinanderzusetzen:

- Aus welchen Grundfunktionen und welchen Operationen auf Funktionen können primitiv-rekursive Funktionen gebildet werden?
- Welche Programmiertechniken für primitiv-rekursive Funktionen gibt es? Welche Beispiele für primitiv-rekursive Funktionen gibt es?
- Was versteht man unter der "Standard-Tupelfunktion" $\langle \cdot, \cdot \rangle$ und ihren Umkehrfunktionen?
- Rechnen Sie die im Skript definierten Beispiele primitiv-rekursiver Funktionen durch!

Kurzquiz 3

(1) Eine Funktion f ist genau dann berechenbar , wenn die charakteristische Funktion $\chi_L(w) = \begin{cases} 1 \text{ falls } w \in L \\ 0 \text{ sonst} \end{cases}$ der Menge L, die f beschreibt, berechenbar ist. **ja** **nein**

(2) Die Funktion, die durch den Ausdruck $Pr[pr_1^1, s \circ pr_3^3]$ beschrieben wird, ist dreistellig. **ja** **nein**

(3) Der Ausdruck $Pr[c_1^0, mul \circ (pr_2^2, s \circ pr_1^2)]$ beschreibt die Funktion $fak(n) = n!$, wenn $mul(m, n) = m * n$. **ja** **nein**

(4) Zu jeder Funktion $f : \mathbb{N}^k \to \mathbb{N}, k \in \mathbb{N}_{>1}$ gibt es eine Funktion $g : \mathbb{N} \to \mathbb{N}$, so dass $f(x_1, \ldots, x_k) = g(\langle x_1, \ldots, x_k \rangle^k)$. **ja** **nein**

(5) Jede berechenbare Funktion ist primitiv-rekursiv. **ja** **nein**

Aufgabe 3.1 *(Umgang mit primitiv-rekursiven Funktionen)*

Die Funktionen sub, fak und mul sind in den Vorlesungsfolien definiert.
Berechnen Sie jeweils schrittweise die Funktionswerte $sub(5, 3)$, $fak(4)$ und $mul(3, 2)$.

Aufgabe 3.2 *(Primitive Rekursion)*

Zeigen Sie für die folgenden Funktionen, daß sie primitiv-rekursiv sind !

(a) $f_> : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit: $f_>(m, n) = \begin{cases} 1 \text{ falls } m > n \\ 0 \text{ falls } m \leq n. \end{cases}$

(b) $f_\geq : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit: $f_\geq(m, n) = \begin{cases} 1 \text{ falls } m \geq n \\ 0 \text{ falls } m < n. \end{cases}$

(c) $f_= : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit: $f_=(m, n) = \begin{cases} 1 \text{ falls } m = n \\ 0 \text{ falls } m \neq n. \end{cases}$

(d) $f_{divides} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit: $f_{divides}(m, n) = \begin{cases} 1 \text{ falls } m \text{ Teiler von } n \\ 0 \text{ falls } m \text{ nicht Teiler von } n. \end{cases}$

(e) $f_{prim} : \mathbb{N} \to \mathbb{N}$ mit: $f_{prim}(n) = \begin{cases} 1 \text{ falls } n \text{ Primzahl} \\ 0 \text{ falls } n \text{ keine Primzahl .} \end{cases}$

Aufgabe 3.3 *(Parametrisierte primitive Rekursion)*

Manche primitiv-rekursiven Funktionen (z.B. die Fibonacci Funktion oder das Rekursionsschema der Türme von Hanoi) lassen sich nicht direkt mit dem Operator Pr ausdrücken. Anstelle dessen verwendet man den parametrisierten Operator Pr_t. Beweisen Sie folgenden Satz:

Wenn f, g und t primitiv-rekursive Funktionen sind, dann ist auch die Funktion $h = Pr_t[f, g]$ primitiv-rekursiv mit

$$h(x, 0) = f(x) \qquad \text{und}$$
$$h(x, n+1) = g(x, n, h(t(x), n))$$

Hinweis: Überlegen Sie sich zunächst die Werte für $h(x, 0), h(x, 1), h(x, 2), \ldots, h(x, n+1)$!

(Haus-)Aufgabe 3.4 *(Primitive Rekursion)*

Zeigen Sie, dass die Funktion $f_{pn}(n) : \mathbb{N} \to \mathbb{N}$ mit „$f_{pn}(n)$ ist die n-te Primzahl“ primitiv-rekursiv ist.

(Haus-)Aufgabe 3.5 *(Rekursive Relationen und ihre Abschlusseigenschaften)*

Eine k-stellige Relation R auf den natürlichen Zahlen ist beschreibbar als Menge von k-Tupeln, für die $R(x_1, \ldots, x_k)$ gilt, $x_j \in \mathbb{N}, 1 \leq j \leq k$. Eine solche Relation heißt primitiv-rekursiv, falls ihre charakteristische Funktion χ_R, primitiv-rekursiv ist, wobei

$$\chi_R(x_1, \ldots, x_k) = \begin{cases} 1 \text{ falls } R(x_1, \ldots, x_k) \text{ gilt} \\ 0 \text{ sonst} \end{cases}$$

Beweisen oder widerlegen Sie:

(a) Angenommen zwei k-stellige Relationen R_1 und R_2 sind primitiv-rekursiv. Dann ist ihre Konjunktion $R_\wedge$ mit $R_\wedge(x_1, \ldots, x_k) \Leftrightarrow R_1(x_1, \ldots, x_k) \wedge R_2(x_1, \ldots, x_k)$ ebenfalls primitiv-rekursiv.

(b) Angenommen eine $(k+1)$-stellige Relation R ist primitiv-rekursiv. Dann ist auch die Relation $R_\exists$ mit $R_\exists(x_1, \ldots, x_k, u) \Leftrightarrow \exists v.((v < u) \wedge R(x_1, \ldots, x_k, v))$ primitiv-rekursiv.

(Haus-)Aufgabe 3.6 *(Schachtelungstiefe rekursiver Funktionen)*

Zeigen Sie, dass es nicht möglich ist, die Additions-Funktion $f_{add}(m, n) = m + n$ aus den Basis-Funktionen Nachfolger, Konstante und Projektion nur durch Komposition und ohne Verwendung von Rekursion zu erhalten.

Hinweis: Die Beweisidee ist, dass wir für jede Funktion f, die sich durch Komposition aus den Basisfunktionen ergibt, eine obere Schranke angeben können, d.h. $f(x_1, \ldots, x_n) < x + a$ wobei x das größte der $x_1, \ldots, x_n$ ist und a eine positive ganze Zahl; für f_{add} kann ein solches a jedoch nicht existieren.

Zur Erinnerung...

In diesem Semester ist die Bearbeitung der Übungen fakultativ. Zur Selbstkontrolle ist es jedoch sinnvoll, die von Ihnen bearbeiteten Übungen korrigieren zu lassen (in diesem Fall gilt der Abgabetermin).

Und ganz wichtig:

Falls Sie Schwierigkeiten haben, sprechen Sie uns an! Auch wenn es etwas Überwindung kostet, Fragen hilft!