

Formale Sprachen und Komplexität, SS 18,
Prof. Dr. Volker Heun

Übungsblatt 9

Abgabe: bis Mo. 02.07.2018 8 Uhr

Formale Sprachen und Komplexität, SS 18
Übungsblatt 9

Abgabe: bis Mo. 02.07.2018 8 Uhr

Nach Bearbeitung dieses Übungsblattes sollten Sie:

	Check
Die Funktion μf zu einer Funktion f bestimmen können.	
Neue μ -rekursive Funktionen gemäß des μ -Schemas aufstellen können.	
Primitiv-rekursive Funktionen angeben können	

Diese Ziele sind wichtige Hinweise für die Klausur!

Aufgabe 9-1 schriftlich bearbeiten
 μ -Schema

- a) Sei $h(x, y) = \text{sub}(y, \text{mult}(x, x))$. Welche Funktion ist $f = \mu h$? (man schreibt für f auch $f(y) = (\mu x)[h(x, y) = 0]$)
- b) Definieren Sie eine Funktion $h'(x, y)$, so dass $f' = \mu h'$ ($f'(y) = (\mu x)[h'(x, y) = 0]$) die partielle Wurzelfunktion auf \mathbb{N} ist:

$$\begin{aligned} \text{sqrt} : \mathbb{N} &\rightarrow \mathbb{N} \\ x &\mapsto \begin{cases} \sqrt{x} & \text{falls } x \text{ Quadratzahl} \\ \text{undefiniert} & \text{andernfalls} \end{cases} \end{aligned}$$

Aufgabe 9-2 schriftlich bearbeiten
 μ -Schema

Geben Sie $f(y) = (\mu x)[h(x, y) = 0]$ jeweils in geschlossener Form an (mit Beweis) für:

- a) $h(x, y) = \text{sub}(x, y)$
- b) $h(x, y) = \text{sub}(y, x)$
- c) $h(x, y) = |y - 2x|$
- d) $h(x, y) = \begin{cases} \text{sub}(y, 2x) & \text{falls } y \geq x \\ \text{undefiniert} & \text{falls } y < x \end{cases}$

Aufgabe 9-3 Ackermann-Funktion

Sei $ack : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ die Ackermann-Funktion mit der Definition

$$\begin{aligned}ack(0, n) &= n + 1 \\ack(k + 1, 0) &= ack(k, 1) \\ack(k + 1, n + 1) &= ack(k, ack(k + 1, n))\end{aligned}$$

- a) Sei $s_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, $s_2(x, y) = y + 1$ (also Nachfolger des zweiten Arguments).
Zeigen Sie, dass für jedes $n \in \mathbb{N}$ gilt: $ack(0, n) = s_2(2, n+3) - 3$.
- b) Zeigen Sie, dass für jedes $n \in \mathbb{N}$ gilt: $ack(1, n) = add(2, n+3) - 3$.
- c) Es gelten völlig analoge Gleichungen für $ack(2, n)$ und $ack(3, n)$ und $ack(4, n)$, wobei lediglich anstelle von add andere zweistellige Funktionen stehen. Geben Sie diese Gleichungen an und beweisen Sie sie.

Die folgenden Funktionen können als primitiv rekursiv vorausgesetzt werden:

Addition $add : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ $(x, y) \mapsto x + y$	Multiplikation $mult : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ $(x, y) \mapsto x \cdot y$
modifizierte Subtraktion $sub : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ $(x, y) \mapsto \begin{cases} 0 & \text{falls } x < y \\ x - y & \text{andernfalls} \end{cases}$	absolute Differenz $absdiff : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ $(x, y) \mapsto x - y $
Signum $sg : \mathbb{N} \rightarrow \mathbb{N}$ $x \mapsto \begin{cases} 0 & \text{falls } x = 0 \\ 1 & \text{andernfalls} \end{cases}$	Kosignum $\overline{sg} : \mathbb{N} \rightarrow \mathbb{N}$ $x \mapsto \begin{cases} 1 & \text{falls } x = 0 \\ 0 & \text{andernfalls} \end{cases}$

Aufgabe 9-4 schriftlich bearbeiten
primitive Rekursion

Weisen Sie formal nach, dass die folgenden Funktionen primitiv rekursiv sind. Geben Sie dazu jeweils eine Definition der Funktion an, die strikt nach den syntaktischen Vorgaben des Kompositions- und/oder Rekursionsschemas für primitive Rekursion aufgebaut ist.

Bei Bedarf können Sie die oben angegebenen Funktionen ohne Beweis verwenden.

a) Die Geradefunktion $evn : \mathbb{N} \rightarrow \mathbb{N}$

$$n \mapsto \begin{cases} 1 & \text{falls } n \text{ gerade} \\ 0 & \text{falls } n \text{ ungerade} \end{cases}$$

b) Die Potenzierfunktion $hoch : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$$(x, n) \mapsto x^n$$

Hinweis: das Rekursionsschema verlangt die Rekursion im ersten Argument. Zeigen Sie zunächst, dass die Funktion mit dem Kompositionsschema auf eine Hilfsfunktion mit vertauschten Argumenten zurückgeführt werden kann: $hoch(x, n) = potenz(n, x)$. Definieren Sie dann diese Hilfsfunktion.

c) Die modifizierte Modulo-Funktion $rest : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$

$$(n, x) \mapsto \begin{cases} n & \text{falls } x = 0 \\ n \bmod x & \text{andernfalls} \end{cases}$$

Es gilt $(n + 1) \bmod x = (n \bmod x) + 1$, falls $(n \bmod x) + 1 \neq x$ ist, andernfalls 0. Aufbauend auf dieser Beobachtung kann $rest$ so definiert werden, dass gilt:

$$\begin{array}{l} rest(0, x) \\ rest(n + 1, x) \end{array} \left[\begin{array}{l} = 0 \\ = mult \left(s(rest(n, x)), sg(absdiff(s(rest(n, x)), x)) \right) \end{array} \right]$$