Vorlesung

Theoretische Informatik II

Bernhard Beckert

Institut für Informatik

UNIVERSITÄT KOBLENZ · LANDAU

Wintersemester 2007/2008

Dank

Diese Vorlesungsmaterialien basieren zum Teil auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Christoph Kreitz (gehalten an der Universität Potsdam)

Ihnen gilt mein herzlicher Dank.

– Bernhard Beckert, Oktober 2007

μ-rekursive Funktionen

Definition 12.1 (μ-rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

- Null 0
- Nachfolger $+1$
- Projektion π_i^k $(1 \leq i \leq k)$

sind μ-rekursiv

Komposition: Die durch Komposition aus μ-rekursiven Funktionen gebildeten Funktionen sind μ-rekursiv

Primitive Rekursion: Die durch primitive Rekursion aus μ-rekursiven Funktionen gebildeten Funktionen sind μ-rekursiv

μ-Operator: Die durch Anwendung des μ-Operators aus μ-rekursiven Funktionen gebildeten Funktionen sind μ-rekursiv

μ-rekursive Funktionen

Notation

$$
\begin{aligned}
F_\mu &= \text{Menge der totalen } \mu\text{-rekursiven Funktionen} \\
F_\mu^{part} &= \text{Menge aller } \mu\text{-rekursiven Funktionen}
\end{aligned}
$$

μ-rekursive Funktionen

Theorem 12.2

$$F_\mu \subseteq \mathbf{WHILE} \quad \textit{und} \quad F_\mu^{part} \subseteq \mathbf{WHILE}^{part}$$

μ-rekursive Funktionen

Beweis (Skizze)

Wir haben schon bewiesen:

$$\wp = \textbf{LOOP} \subset \textbf{WHILE}$$

Bei Verwendung des entsprechenden Beweises bleibt zu zeigen, dass der μ-Operator in WHILE nachgebildet werden kann.

$$f(\boldsymbol{n}) = \mu i(g(\boldsymbol{n}, i) = 0)$$

kann berechnet werden durch:

$$\mathtt{x}_i = 0; \quad \mathtt{while}\ g(\boldsymbol{n}, \mathtt{x}_i) \neq 0\ \mathtt{do}\ \mathtt{x}_i := \mathtt{x}_i + 1\ \mathtt{end}$$

(informelle Notation!)

Ackermann-Funktion

Wilhelm Ackermann $\star$ 1896, † 1962

- Mathematiker und Logiker
- Promovierte bei D. Hilbert in Göttingen
  (Koautor von Hiberts Buch
  Grundzüge der Theoretischen Logik)
- Mathematiklehrer in Lüdenscheid

Aus seinem Nachruf:

> *Oberstudienrat Dr. Ackermann war aber nicht nur ein allseits geschätzter und beliebter Lehrer, sondern auch ein weltbekannter Wissenschaftler . . .*

Ackermann-Funktion

Definition 12.3 (Ackermann-Funktion A)

$$A_0(x) = \begin{cases} 1 & \text{falls } x = 0 \\ 2 & \text{falls } x = 1 \\ x + 2 & \text{sonst} \end{cases}$$

$$\begin{aligned} A_{n+1}(0) &= 1 \\ A_{n+1}(x+1) &= A_n(A_{n+1}(x)) \end{aligned}$$

$$A(x) = A_x(x)$$

Ackermann-Funktion

Wachstumsverhalten

$$A_1(x) = 2 * x$$

$$A_2(x) = 2^x$$

$$A_3(x) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{x \text{ mal}}$$

Ackermann-Funktion

Wachstumsverhalten

$$A_4(3) = 2^{2^{2^{2}}} = 65536$$

$$A_4(4) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{65536 \text{ mal}}$$

$$A_4(5) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{A_4(4) \text{ mal}}$$

Ackermann-Funktion

Theorem 12.4

Die Ackermann-Funktion ist

- *total*
- *μ-rekursiv*
- ***nicht primitiv-rekursiv***

Ackermann-Funktion

Beweis

Die Funktionen A_n sind total, da in jedem Rekursionsschritt eines der Argumente kleiner wird.

A ist TM-berechenbar, weil man den Rekursions-Stack auf dem Band speichern kann.

Wegen $F_\mu =$ **WHILE** (Beweis später)
ist A auch μ-rekursiv

Bei einer primitiv rekursiven Funktion f ist die zur Berechnung von $f(n)$ notwendige Funktions-Schachtelungstiefe für alle n gleich.

A kann mit konstanter Funktions-Schachtelungstiefe nicht berechnet werden. (Detaillierter Beweis sehr aufwendig)

Teil III

Rück- und Ausblick

Aus dem Vorlesungsteil „Registermaschinen“ wissen wir:

- $\mathbf{LOOP} \subsetneq \mathbf{WHILE} = \mathbf{GOTO} \subseteq \mathbf{TM}$
- $\mathbf{WHILE}^{part} = \mathbf{GOTO}^{part} \subseteq \mathbf{TM}^{part}$

In diesem Abschnitt schon gezeigt:

- $\mathbf{LOOP} = \wp$
- $F_\mu \subseteq \mathbf{WHILE}$ und $F_\mu^{part} \subseteq \mathbf{WHILE}$

Noch zu zeigen:

$\mathbf{TM} \subseteq F_\mu$ und $\mathbf{TM}^{part} \subseteq F_\mu^{part}$