

Vorlesung

Theoretische Informatik II

Bernhard Beckert

Institut für Informatik



Wintersemester 2007/2008

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*

Teil III

Rekursive Funktionen

- 1 Einführung
- 2 Primitiv rekursive Funktionen
- 3 $\wp = \text{LOOP}$
- 4 μ -rekursive Funktionen
- 5 $F_\mu = \text{WHILE}$
- 6 Zusammenfassung

Teil III

Rekursive Funktionen

- 1 Einführung
- 2 Primitiv rekursive Funktionen
- 3 $\wp = \text{LOOP}$
- 4 μ -rekursive Funktionen
- 5 $F_\mu = \text{WHILE}$
- 6 Zusammenfassung

Teil III

Rekursive Funktionen

1 Einführung

2 Primitiv rekursive Funktionen

3 \wp = LOOP

4 μ -rekursive Funktionen

5 F_μ = WHILE

6 Zusammenfassung

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- **Einfache (atomare) Funktionen sind berechenbar**
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- **Kombinationen berechenbarer Funktionen sind berechenbar**
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- **Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)**

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- **Welches sind die atomaren Funktionen?**
- Welche Art der Kombination ist zulässig?

Rekursive Funktionen

Motivation

- Funktionen als Berechnungsmodell
- Ohne darunterliegendes Maschinenmodell

Idee

- Einfache (atomare) Funktionen sind berechenbar
- Kombinationen berechenbarer Funktionen sind berechenbar
- Wir betrachten Funktionen $f : \mathbb{N}^k \rightarrow \mathbb{N}$ ($k \geq 0$)

Fragen

- Welches sind die atomaren Funktionen?
- Welche Art der Kombination ist zulässig?

Atomare Funktionen

Folgende Funktionen sind **primitiv rekursiv** und μ -rekursiv:

Konstante Null

$$0 : \mathbb{N}^0 \rightarrow \mathbb{N} \quad \text{mit} \quad 0() = 0$$

Nachfolger (Successor)

$$+1 : \mathbb{N}^1 \rightarrow \mathbb{N} \quad \text{mit} \quad +1(n) = n + 1$$

Projektion (Auswahl)

$$\pi_i^k : \mathbb{N}^k \rightarrow \mathbb{N} \quad \text{mit} \quad \pi_i^k(n_1, \dots, n_i, \dots, n_k) = n_i \quad (1 \leq i \leq k)$$

Atomare Funktionen

Folgende Funktionen sind **primitiv rekursiv** und μ -rekursiv:

Konstante Null

$$0 : \mathbb{N}^0 \rightarrow \mathbb{N} \quad \text{mit} \quad 0() = 0$$

Nachfolger (Successor)

$$+1 : \mathbb{N}^1 \rightarrow \mathbb{N} \quad \text{mit} \quad +1(n) = n + 1$$

Projektion (Auswahl)

$$\pi_i^k : \mathbb{N}^k \rightarrow \mathbb{N} \quad \text{mit} \quad \pi_i^k(n_1, \dots, n_i, \dots, n_k) = n_i \quad (1 \leq i \leq k)$$

Atomare Funktionen

Folgende Funktionen sind **primitiv rekursiv** und **μ -rekursiv**:

Konstante Null

$$0 : \mathbb{N}^0 \rightarrow \mathbb{N} \quad \text{mit} \quad 0() = 0$$

Nachfolger (Successor)

$$+1 : \mathbb{N}^1 \rightarrow \mathbb{N} \quad \text{mit} \quad +1(n) = n + 1$$

Projektion (Auswahl)

$$\pi_i^k : \mathbb{N}^k \rightarrow \mathbb{N} \quad \text{mit} \quad \pi_i^k(n_1, \dots, n_i, \dots, n_k) = n_i \quad (1 \leq i \leq k)$$

Schreibweise bei Argumententupeln

n für n_1, \dots, n_k ($k \geq 0$)

Komposition (simultanes Einsetzen)

Sind

$$g : \mathbb{N}^r \rightarrow \mathbb{N} \quad (r \geq 1)$$

$$h_1 : \mathbb{N}^k \rightarrow \mathbb{N}, \dots, h_r : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

primitiv rekursiv bzw. μ -**rekursiv**,

dann ist auch

$$f : \mathbb{N}^k \rightarrow \mathbb{N} \quad \text{mit} \quad f(\mathbf{n}) = g(h_1(\mathbf{n}), \dots, h_r(\mathbf{n}))$$

primitiv rekursiv bzw. μ -**rekursiv**

Schreibweise ohne Argumente:

$$f = g \circ (h_1, \dots, h_r)$$

Komposition (simultanes Einsetzen)

Sind

$$g : \mathbb{N}^r \rightarrow \mathbb{N} \quad (r \geq 1)$$

$$h_1 : \mathbb{N}^k \rightarrow \mathbb{N}, \dots, h_r : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

primitiv rekursiv bzw. μ -**rekursiv**,

dann ist auch

$$f : \mathbb{N}^k \rightarrow \mathbb{N} \quad \text{mit} \quad f(\mathbf{n}) = g(h_1(\mathbf{n}), \dots, h_r(\mathbf{n}))$$

primitiv rekursiv bzw. μ -**rekursiv**

Schreibweise ohne Argumente:

$$f = g \circ (h_1, \dots, h_r)$$

Rekursive Funktionen

1 Einführung

2 Primitiv rekursive Funktionen

3 \wp = LOOP

4 μ -rekursive Funktionen

5 F_μ = WHILE

6 Zusammenfassung

Teil III

Rekursive Funktionen

1 Einführung

2 Primitiv rekursive Funktionen

3 $\wp = \text{LOOP}$

4 μ -rekursive Funktionen

5 $F_\mu = \text{WHILE}$

6 Zusammenfassung

Primitive Rekursion

Sind

$$g : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

$$h : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$$

primitiv rekursiv,

dann ist auch

$$f : \mathbb{N}^{k+1} \rightarrow \mathbb{N} \quad \text{mit} \quad \begin{aligned} f(\mathbf{n}, 0) &= g(\mathbf{n}) \\ f(\mathbf{n}, m+1) &= h(\mathbf{n}, m, f(\mathbf{n}, m)) \end{aligned}$$

primitiv rekursiv

Schreibweise ohne Argumente:

$$f = \mathcal{PR}[g, h]$$

Primitive Rekursion

Sind

$$g : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

$$h : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$$

primitiv rekursiv,

dann ist auch

$$f : \mathbb{N}^{k+1} \rightarrow \mathbb{N} \quad \text{mit} \quad \begin{aligned} f(\mathbf{n}, 0) &= g(\mathbf{n}) \\ f(\mathbf{n}, m+1) &= h(\mathbf{n}, m, f(\mathbf{n}, m)) \end{aligned}$$

primitiv rekursiv

Schreibweise ohne Argumente:

$$f = \mathcal{PR}[g, h]$$

Primitiv rekursive Funktionen

Definition 10.1 (Primitiv rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

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

sind primitiv rekursiv

Komposition: Die durch Komposition aus primitiv rekursiven Funktionen gebildeten Funktionen sind primitiv rekursiv

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

Notation

\mathcal{P} = Menge der primitiv rekursiven Funktionen

Primitiv rekursive Funktionen

Definition 10.1 (Primitiv rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

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

sind primitiv rekursiv

Komposition: Die durch Komposition aus primitiv rekursiven Funktionen gebildeten Funktionen sind primitiv rekursiv

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

Notation

\mathcal{P} = Menge der primitiv rekursiven Funktionen

Primitiv rekursive Funktionen

Definition 10.1 (Primitiv rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

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

sind primitiv rekursiv

Komposition: Die durch Komposition aus primitiv rekursiven Funktionen gebildeten Funktionen sind primitiv rekursiv

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

Notation

\mathcal{P} = Menge der primitiv rekursiven Funktionen

Primitiv rekursive Funktionen

Definition 10.1 (Primitiv rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

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

sind primitiv rekursiv

Komposition: Die durch Komposition aus primitiv rekursiven Funktionen gebildeten Funktionen sind primitiv rekursiv

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

Notation

\mathcal{R} = Menge der primitiv rekursiven Funktionen

Primitiv rekursive Funktionen

Definition 10.1 (Primitiv rekursive Funktionen)

Atomare Funktionen: Die konstanten Funktionen

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

Komposition: Die durch Komposition aus primitiv rekursiven Funktionen gebildeten Funktionen sind primitiv rekursiv

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

Notation

\mathcal{P} = Menge der primitiv rekursiven Funktionen

Definition arithmetischer Funktionen

$$f(n) = n + c \quad \text{für } c \in \mathbb{N}, c > 0$$

$$f = \underbrace{+1 \circ \dots \circ +1}_{c \text{ mal}}$$

Identität

$$f = \pi_1^1$$

$$f(n, m) = n + m$$

$$f = \mathcal{PR}[\pi_1^1, +1 \circ \pi_3^3]$$

Definition arithmetischer Funktionen

$$f(n) = n + c \quad \text{für } c \in \mathbb{N}, c > 0$$

$$f = \underbrace{+1 \circ \dots \circ +1}_{c \text{ mal}}$$

Identität

$$f = \pi_1^1$$

$$f(n, m) = n + m$$

$$f = \mathcal{PR}[\pi_1^1, +1 \circ \pi_3^3]$$

Definition arithmetischer Funktionen

$$f(n) = n + c \quad \text{für } c \in \mathbb{N}, c > 0$$

$$f = \underbrace{+1 \circ \dots \circ +1}_{c \text{ mal}}$$

Identität

$$f = \pi_1^1$$

$$f(n, m) = n + m$$

$$f = \mathcal{PR}[\pi_1^1, +1 \circ \pi_3^3]$$

Definition arithmetischer Funktionen

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, -1 \circ \pi_3^3]$$

$$f(n, m) = n * m$$

$$f = \mathcal{PR}[0, + \circ (\pi_1^3, \pi_3^3)]$$

Definition arithmetischer Funktionen

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, -1 \circ \pi_3^3]$$

$$f(n, m) = n * m$$

$$f = \mathcal{PR}[0, + \circ (\pi_1^3, \pi_3^3)]$$

Definition arithmetischer Funktionen

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, -1 \circ \pi_3^3]$$

$$f(n, m) = n * m$$

$$f = \mathcal{PR}[0, + \circ (\pi_1^3, \pi_3^3)]$$