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

Darstellung von Aussagenlogik im λ-Kalkül

Wahrheitswerte

$$\mathit{true} =_{\mathrm{def}} \lambda x.\lambda y.\, x$$
$$\mathit{false} =_{\mathrm{def}} \lambda x.\lambda y.\, y$$

if-then-else

$$\mathit{if}\ C\ \mathit{then}\ U\ \mathit{else}\ V =_{\mathrm{def}} \mathbf{???}\, C\ U\ V$$

Darstellung von Aussagenlogik im λ-Kalkül

Mit true, false, if-then-else alles darstellbar

$$\neg x \equiv \mathit{if}\ x\ \mathit{then}\ \mathit{false}\ \mathit{else}\ \mathit{true}$$
$$x \wedge y \equiv \mathit{if}\ x\ \mathit{then}\ y\ \mathit{else}\ \mathit{false}$$
$$x \vee y \equiv \mathit{if}\ x\ \mathit{then}\ \mathit{true}\ \mathit{else}\ y$$

Damit

$$\neg x \equiv x\ \mathit{false}\ \mathit{true}$$
$$x \wedge y \equiv x\ y\ \mathit{false}$$
$$x \vee y \equiv x\ \mathit{true}\ y$$

Darstellung Paaren im λ-Kalkül

Paare

$$\begin{aligned} mkpair(x,y) &=_{\text{def}} \lambda b.\, b\ x\ y \\ fst(p) &=_{\text{def}} p\ true \\ snd(p) &=_{\text{def}} p\ false \end{aligned}$$

Ähnlich für

- Tupel
- Listen
- etc.

Darstellung natürlicher Zahlen im λ-Kalkül

Natürliche Zahlen

$$\begin{aligned} 0 &=_{\text{def}} \lambda f.\lambda x.\, x \\ 1 &=_{\text{def}} \lambda f.\lambda x.\, f\ x \\ 2 &=_{\text{def}} \lambda f.\lambda x.\, f\,(f\ x) \\ 3 &=_{\text{def}} \lambda f.\lambda x.\, f(\,f\,(f\ x)) \\ &\vdots \\ n &=_{\text{def}} \lambda f.\lambda x.\, \underbrace{f(\ldots(f}_{n \text{ mal}}\ x)\ldots) \end{aligned}$$

Darstellung natürlicher Zahlen im λ-Kalkül

Operationen auf natürlichen Zahlen

$$\begin{aligned} succ(n) &=_{\text{def}} \lambda g.\lambda y.\, n\ g\ (g\, y) \\ +(n,m) &=_{\text{def}} n\ succ\ m \\ *(n,m) &=_{\text{def}} n\ (m\ succ)\ 0 \\ iszero(n) &=_{\text{def}} n\ (\lambda b.false)\ true \end{aligned}$$

Rekursion im λ-Kalkül

Der Y-Operator

$$Y =_{\text{def}} \lambda F.\,(\lambda y.\,F(y\ y))\ (\lambda x.\,F(xx))$$

Y ist Fixpunkt-Operator

$$f(Y\,f) = Y\,f$$

für alle f

Rekursion im λ-Kalkül

Beispiel

$$fak\ n =_{\text{def}} (Y\ F)\ n$$

mit

$$F =_{\text{def}} \lambda f.\lambda n.\ if\ iszero(n)\ then\ 1\ else\ n * (f(n-1))$$

Allgemein

Der Y-Operator erlaubt es, beliebige μ-rekursive Funktionen im λ-Kalkül zu definieren.