6. Übung Programmierung

Dominic Deckert

16. Mai 2017

Previously on ...

- Lambda-Kalkül
  - Freie / Gebundene Variablen
  - Reduktion

Letzte Woche

Aufgabe 1

Kombinator: Lambda-Term ohne freie Variablen

Aufgabe 1

Kombinator: Lambda-Term ohne freie Variablen

$$A = (\lambda xyz.y)$$
$$B = (\lambda xy.y\,x)$$
$$C = (\lambda x.x\,x)$$
$$D = CC$$
$$E = (\lambda ft.f\;t\;f)$$

2 a)

$$
\begin{array}{llll}
 & \underbrace{(\lambda fx.f\ f\ x)}_{GV=\{x\}} & \underbrace{(\lambda y.x)}_{FV=\{x\}} & z \\
\Rightarrow_\alpha & (\lambda fa.f\ f\ a) & (\lambda y.x) & z \\
\Rightarrow_\beta & (\lambda a.(\lambda y.x)(\lambda y.x)a) & z & \\
\Rightarrow_\beta & (\lambda y.x)(\lambda y.x)z & & \\
\Rightarrow_\beta & xz & &
\end{array}
$$

gängige Kombinatoren

Kombinator	Struktur	Bedeutung
$< pred >, < succ >$	$< pred > n$	Vorgänger-/ Nachfolger von n
$< add >, < sub >$	$< add > nm$	Summe $n + m$ (Differenz immer ≥ 0)
$< mult >$	$< mult > nm$	Produkt $n \cdot m$
$< ite >$	$< ite > bte$	Bedingung b: Wahr $\rightarrow$ t Falsch $\rightarrow$ e
$< iszero >$	$< iszero > n$	Wahr, wenn $n \stackrel{?}{=} 0$

Fixpunktkombinator

$< Y >= (\lambda h.(\lambda x.h(x\,x))(\lambda x.h(x\,x)))$

Idee: Funktion h kann rekursiv immer wieder angewandt werden

Bsp:

$$
\begin{aligned}
< Y >< F > = (\lambda h.(\lambda x.h(x\,x))(\lambda x.h(x\,x))) < F > \\
\Rightarrow_\beta ((\lambda x. < F > (x\,x)(\lambda x. < F > (x\,x)) =< Y_F > \\
\Rightarrow_\beta < F > ((\lambda x. < F > (xx))(\lambda x. < F > (xx))) =< F >< Y_F >
\end{aligned}
$$

2 b), c)

siehe Tafel

3 a)

$$
\begin{array}{llll}
 & \underbrace{(\lambda fx.f(fx))}_{GV=\{x\}} & \underbrace{(\lambda y.x)}_{FV=\{x\}} & y \\
\Rightarrow_\alpha & (\lambda fv.f(fv)) & (\lambda y.x) & z \\
\Rightarrow_\beta & (\lambda y.(\lambda y.x)((\lambda y.x)v)) & z & \\
\Rightarrow_\beta & (\lambda y.x)((\lambda y.x)z) & & \\
\Rightarrow_\beta & x & &
\end{array}
$$

3 b)

$$
\begin{array}{lll}
(\lambda gmn. & < ite > & (< iszero > n)m \\
 & (< ite > & (< iszero > (< pred > n))(< succ > m) \\
 & & (< add > (g\, m(< pred >< pred > n))(g\, m(< pred >< n >)))))
\end{array}
$$

3 c)

Wie in 2 gilt hier $<Y><F> \Rightarrow^* <Y_F> \Rightarrow_\beta <F><Y_F>$

$$
\begin{aligned}
<Y><F><2><1> &\Rightarrow^* <F><Y_F><2><1> \\
&\Rightarrow^* (<ite> (<iszero><1>) <1> \\
&\quad (<mult><2> (<Y_F><2> (<pred><1>)))) \\
&\Rightarrow^* <mult><2> (<Y_F><2><0>) \\
&\Rightarrow^* <mult><2> (<F><Y_F><2><0>) \\
&\Rightarrow^* <mult><2> ((<ite> (<iszero><0>) <1> \\
&\quad (<mult><2> (<Y_F><2> (<pred><0>))))) \\
&\Rightarrow^* <mult><2><1> \\
&\Rightarrow^* <2>
\end{aligned}
$$