

Programming languages - U6

Jan Dietrich - 10-100-436

1

Given:

$t \equiv \lambda f. \lambda n_1. \lambda n_2. \text{if } (\text{isZero } n_1) \ 0 \ (\text{add } n_2 (f(\text{pred } n_1) n_2))$

and

$((Yt)1)k \rightarrow k$

we can replace the insert the values for n_1, n_2, f

$\text{if } (\text{isZero } 1) \ 0 \ (\text{add } k((Yt)(\text{pred } 1)k))$

with $\text{isZero } 1 = \text{false}$ und $\text{pred } 1 = 0$

$\text{add } k((Yt)0)k$

since we have

$Yt \leftrightarrow t(Yt)$

we can reuse the function definition to achieve

$\text{add } k((\lambda f. \lambda n_1. \lambda n_2. \text{if } (\text{isZero } n_1) \ 0 \ (\text{add } n_2 (f(\text{pred } n_1) n_2)))0)k$

with $\text{isZero } n_1$ with $n_1 = 0$ is true results in

$\text{add } k0$

since we know the implementation of add we can safely say

$\text{add } k \ 0 = k$

2

replace append with a f

$\text{app} \equiv \lambda f. \lambda l_1. \lambda l_2. \text{if } (\text{null } l_1) l_2 (\text{cons } (\text{head } l_1) (f(\text{tail } l_1) l_2))$

To test insert $L_1 = \text{cons } 1 (\text{cons } 2 \text{ nil})$ and $L_2 = \text{cons } 3 \text{ nil}$

$\text{if } (\text{null } (\text{cons } 1 (\text{cons } 2 \text{ nil}))) (\text{cons } 3 \text{ nil}) (\text{cons } (\text{head } (\text{cons } 1 (\text{cons } 2 \text{ nil}))) (f(\text{tail } (\text{cons } 1 (\text{cons } 2 \text{ nil}))) (\text{cons } 3 \text{ nil})))$

since it's pretty unreadable we try to simplify

we can simplify the head and tail parts

if (null (cons 1(cons 2 nil))(cons 3 nil)(cons (cons 1)(f(cons 2 nil)(cons 3 nil))

next step we can simplify cons

if (null (cons 1(cons 2 nil))(cons 3 nil)(cons 1(f(cons 2 nil)(cons 3 nil))

we now that null (cons 1(cons 2 nil)) must be false

cons 1(f(cons 2 nil)(cons 3 nil))

and since $Yapp \leftrightarrow app(Yapp)$

cons 1(app (Yapp)(cons 2 nil)(cons 3 nil))

By inserting the app definition

cons 1($\lambda f.\lambda l_1.\lambda l_2.$ if (null l_1) l_2 (cons (head l_1)(f(tail l_1) l_2))(Yapp)(cons 2 nil)(cons 3 nil))

with null cons 2 nil = false and head (cons 2 nil) = 2 and tail cons 2 nil = nil

cons 1(cons 2((Yapp)nil (cons 3 nil)

again use app definition with $l_1 = \text{nil}$

and since we now have

nil nil which results to true

we can simplify to

cons 1(cons 2(cons 3 nil)

In the example there is written that this is equivalent to [1,2,3]