Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Simon Wimmer

Lambda Calculus
Winter Term 2017/18
Exercise Sheet 2

Exercise 1 (β-reduction)

List all terms t such that:

$$(\lambda x.(\lambda xy.x)zy)((\lambda x.xx)(\lambda x.xx)((\lambda xy.x)y)) \to_\beta^* t$$

Which are normal forms?

Exercise 2 (Lists in λ-calculus)

Specify λ-terms for nil, cons, hd, tl and null, that enocde lists in the λ-calculus. Show that your terms satisfy the following conditions:

$$\begin{array}{llllll} \mathsf{null}\ \mathsf{nil} & \to_\beta^* & \mathsf{true} & \mathsf{hd}\ (\mathsf{cons}\ x\ l) & \to_\beta^* & x \\ \mathsf{null}\ (\mathsf{cons}\ x\ l) & \to_\beta^* & \mathsf{false} & \mathsf{tl}\ (\mathsf{cons}\ x\ l) & \to_\beta^* & l \end{array}$$

Hint: Use pairs.

Homework 3 (Substitution Lemma)

Show that, given $x \neq y$ and $x \notin \mathrm{FV}(u)$:

$$s[t/x][u/y] = s[u/y][t[u/y]/x]$$

Homework 4 (Trees in λ-calculus)

Encode a datatype of binary trees in lambda calculus. Specify the operations tip and node that construct trees, as well as isTip, left, right, and value. Each tip should carry a value, whereas each node should consist of two subtrees.

Show that the following holds:

$$\begin{aligned} \mathsf{isTip}\ (\mathsf{tip}\ a) &\to_\beta^* \mathsf{true} \\ \mathsf{isTip}\ (\mathsf{node}\ x\ y) &\to_\beta^* \mathsf{false} \\ \mathsf{value}\ (\mathsf{tip}\ a) &\to_\beta^* a \\ \mathsf{left}\ (\mathsf{node}\ x\ y) &\to_\beta^* x \\ \mathsf{right}\ (\mathsf{node}\ x\ y) &\to_\beta^* y \end{aligned}$$

Homework 5 (Alternative Encoding of Lists)

In this exercise, we consider an alternative encoding of lists. The list $[x, y, z]$, for instance, will now be encoded as: $\lambda cn.cx(cy(czn))$ (speaking in terms of functional programming, each list now encodes its corresponding *fold*). As in the tutorial, define the functions nil, cons, hd, and null for this encoding and show that they satisfy the same conditions. You do not need to define tl.

Homework 6 (Multiplication)

Define multiplication as a closed λ-term using add but no other helper functions and demonstrate its correctness on an example.