Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Johannes Hölzl and Lars Hupel

Equational Logic
Summer Term 2015
Exercise Sheet 11

Exercise 55 (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:

$$\begin{array}{lll@{\qquad}lll} \mathsf{null}\ \mathsf{nil} & \longrightarrow^* & \mathsf{true} & \mathsf{hd}\ (\mathsf{cons}\ x\ l) & \longrightarrow^* & x \\ \mathsf{null}\ (\mathsf{cons}\ x\ l) & \longrightarrow^* & \mathsf{false} & \mathsf{tl}\ (\mathsf{cons}\ x\ l) & \longrightarrow^* & l \end{array}$$

Hint: Use pairs.

Exercise 56 (Fixed-point Combinator)

Use a fixed-point combinator to compute the length of lists.

Exercise 57 (β-reduction on de Bruijn preserves substitution)

Prove Lemma 1.2.5 for de-Bruijn terms:

$$s \rightarrow_\beta s' \implies s[u/x] \rightarrow_\beta s'[u/x]$$

Homework 58 (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:

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

Homework 59 (β-reduction preserves types)

A type system has the *subject reduction property* if evaluating an expression preserves its type. Prove that the simply typed λ-calculus ($\lambda^{\rightarrow}$) has the subject reduction property:

$$\Gamma \vdash t : \tau \wedge t \longrightarrow_{\beta} t' \implies \Gamma \vdash t' : \tau$$

Hints: Use induction over the inductive definition of $\longrightarrow_{\beta}$ (Def. 1.2.2). State your inductive hypotheses precisely – it may help to introduce a binary predicate $P(t, t')$ to express the property you are proving by induction. Also note that the proof will require *rule inversion*: Given $\Gamma \vdash t : \tau$, the shape of t (variable, application, or λ-abstraction) may determine which typing rule must have been used to derive the typing judgment.

Within your proof, you are free to use the following lemma about substitution:

$$\Gamma \vdash u : \tau_0 \wedge \Gamma[x : \tau_0] \vdash t : \tau \implies \Gamma \vdash t[u/x] : \tau \tag{1}$$