Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Lukas Stevens

Lambda Calculus
Winter Term 2022/23
Exercise Sheet 8

Exercise 1 (Progress Property)

Let t be a closed and well-typed term, i.e. $[] \vdash t : \tau$ for some τ. Show that t is either a value or there is a t' such that $t \to_{cbv} t'$.

Exercise 2 (Normal Form)

Show that every type-correct $\lambda^{\rightarrow}$-term has a β-normal form.

Homework 3 (Typing)

a) Prove:

$$[] \vdash (\lambda x\colon \tau_2 \to \tau_3.\ \lambda y\colon \tau_1 \to \tau_2.\ \lambda z\colon \tau_1.\ \ x\,(y\,z))\colon (\tau_2 \to \tau_3) \to (\tau_1 \to \tau_2) \to \tau_1 \to \tau_3$$

b) Give suitable solutions for $?\tau_1$, $?\tau_2$, $?\tau_3$ and $?\tau_4$ and prove that the term is type-correct given your solution.

$$[] \vdash \lambda x:?\tau_1.\ \lambda y:?\tau_2.\ \lambda z:?\tau_3.\ \ x\ y\ (y\ z)\ :?\tau_4$$

Homework 4 (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 ($\lambda^{\to}$) has the subject reduction property:

$$\Gamma \vdash t\colon \tau \wedge t \to_\beta t' \Longrightarrow \Gamma \vdash t'\colon \tau$$

Hints: Use induction over the inductive definition of $\to_\beta$ (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 $\Gamma \vdash t\colon \tau$, 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:

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

Homework 5 (Implementation of multiset-ordering and reduction)

Implement the multiset ordering and the reduction strategy from the second tutorial exercise in your favorite programming language.