Exercise 1 (Progress Property)

Let \( t \) be a closed and well-typed term, i.e. \( \Gamma \vdash t : \tau \) for some \( \tau \). Show that \( t \) is either a value or there is a \( t' \) such that \( t \rightarrow_{cbv} t' \).

Solution

The proof follows an induction on the derivation of \( \Gamma \vdash t : \tau \). The variable case cannot occur. Abstractions are values, so there is nothing to do here. For the application case, assume \( t = t_1 t_2 \) and \( \Gamma \vdash t_1 : \tau_1 \rightarrow \tau_2 \) and \( \Gamma \vdash t_2 : \tau_1 \). By the induction hypothesis, both, \( t_1 \) and \( t_2 \), can take a step or are a value. If \( t_1 \) can take a step, we can use the left application rule on \( t \). If \( t_1 \) is a value and \( t_2 \) can take a step, then the right application rule can be used. If \( t_1 \) and \( t_2 \) are both values, we know \( t_1 = \lambda x. t'_1 \) for some \( t'_1 \) as \( t_1 \) is of type \( \tau_1 \rightarrow \tau_2 \). Thus we can apply the rule for reducing an abstraction.

Exercise 2 (Normal Form)

Show that every type-correct \( \lambda \rightarrow \)-term has a \( \beta \)-normal form.

Solution

We regard a reduction strategy that is guaranteed to terminate. The strategy is chosen such that it decreases the types of subterms.

Let \(|\tau|\) be the size of a type \( \tau \), i.e. the number of function-arrows occurring in \( \tau \).

\[
|\alpha| = 0 \\
|\alpha \rightarrow \beta| = |\alpha| + |\beta| + 1
\]

With this measure, we can assign a natural number to each \( \beta \)-redex:

\[
|(\lambda x. s) t| = |\tau_1 \rightarrow \tau_2| \quad \text{where} \quad \Gamma \vdash \lambda x. s : \tau_1 \rightarrow \tau_2
\]

Thus, for any term \( t \), we obtain a multiset

\[
M(t) = \{ |(\lambda x. s) t'| \mid \exists x \ s \ t'. \ ((\lambda x. s) t') \text{is a subterm of } t \}
\]

\( M(t) \) is the multiset of the sizes of all \( \beta \)-redexes in \( t \).

We can view multisets as functions into the natural numbers and define an ordering on them:

\[
M <_M N \iff M \neq N \land \forall y. M(y) > N(y) \implies \exists x. y < x \land M(x) < N(x)
\]
It can be proved that the multiset ordering terminates (is well-founded).

If one regards a \( \beta \)-redex of the form \( r = (\lambda x. \ u) \ v \) with \( u \) and \( v \) in \( \beta \)-NF, then we have for the reduct \( r' = u[v/x] : M(r) >_M M(r') \). This is because of \( [] \vdash \lambda x. \ u : \tau_1 \rightarrow \tau_2 \) and \( [] \vdash v : \tau_1 \), the substitution may create new \( \beta \)-redexes \( w \), but for all those \( w \) in \( r' \) we have \( |w| < |r| \):

\[
\begin{align*}
w \text{ is of the form } (v \ v') \\
\text{and thus } |w| = |\tau_1| < |\tau_1 \rightarrow \tau_2| = |r|.
\end{align*}
\]

Thus, if we choose a reduction strategy \( \rightarrow_p \) that reduces an innermost \( \beta \)-redex in \( t \), we have:

\[
t \rightarrow_p t' \Rightarrow M(t) >_M M(t')
\]

We can obtain such a reduction strategy by restricting the first rule of \( \rightarrow_\beta \) to:

\[
\frac{s \in \text{NF} \quad t \in \text{NF}}{\lambda x. s \rightarrow_p s[t/x]}
\]

As the multiset ordering terminates, also the chosen reduction strategy must terminate. If it terminates with \( t' \), then \( t' \) is in \( \beta \)-NF.

\[ \square \]

Homework 3 (Typing)

a) Prove:

\[
[] \vdash (\lambda x : \tau_2 \rightarrow \tau_3. \lambda y : \tau_1 \rightarrow \tau_2. \lambda z : \tau_1. x (y z)) : (\tau_2 \rightarrow \tau_3) \rightarrow (\tau_1 \rightarrow \tau_2) \rightarrow \tau_1 \rightarrow \tau_3
\]

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 (\( \beta \)-reduction preserves types)

A type system has the subject reduction property if evaluating an expression preserves its type. Prove that the simply typed \( \lambda \)-calculus (\( \lambda^\tau \)) has the subject reduction property:

\[
\Gamma \vdash t : \tau \land t \rightarrow_\beta t' \Rightarrow \Gamma \vdash t' : \tau
\]

Hints: Use induction over the inductive definition of \( \rightarrow_\beta \) (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 \( \lambda \)-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:

\[
\Gamma \vdash u : \tau_0 \land \Gamma[x : \tau_0] \vdash t : \tau \Rightarrow \Gamma \vdash t[u/x] : \tau
\]

(1)