Lambda-calculus and programming language semantics

Exam - duration 2:00 - course notes allowed

Exercise 1. Terms and reductions

1. Show that, under some conditions on x, y, t, u and v we have

$$(\lambda x.t)\ ((\lambda y.u)\ v)\ =_\beta\ (\lambda y.((\lambda x.t)\ u))\ v$$

2. Define the size $|t|$, the width $w(t)$ and the height $h(t)$ of a λ-term t by the following equations:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
|x| & = & 1 & w(x) & = & 1 & h(x) & = & 0 \\
|\lambda x.t| & = & 1 + |t| & w(\lambda x.t) & = & w(t) & h(\lambda x.t) & = & 1 + h(t) \\
|t_1\ t_2| & = & 1 + |t_1| + |t_2| & w(t_1\ t_2) & = & w(t_1) + w(t_2) & h(t_1\ t_2) & = & \max(h(t_1), h(t_2))
\end{array}$$

Show that for any term t we have $|t| \geqslant w(t) + h(t)$.

3. Draw a graph showing all the reductions starting from the term

$$t\ \equiv\ (\lambda x.x\ (x\ a))\ (\mathsf{I}\ \mathsf{F})$$

Reminder: $\mathsf{I} \equiv \lambda x.x$ and $\mathsf{F} \equiv \lambda xy.y$.

4. Is there a λ-term t such that $t =_\beta t\ t$? If so, provide one such term.

Exercise 2. Lambda-calculus with let

We consider a λ-calculus extended with a construct $\mathsf{let}\ x = s\ \mathsf{in}\ t$ linking a variable x to a term s in a term t. Full syntax of the extended calculus Λ^+:

$$\begin{array}{rcll}
t & ::= & x & \text{variable} \\
 & | & t\,t & \text{application} \\
 & | & \lambda x.t & \text{abstraction} \\
 & | & \mathsf{let}\ x = t\ \mathsf{in}\ t & \text{local variable}
\end{array}$$

The definition of substitution is extended as well, to take into account the new terms.

$$\begin{array}{rcll}
y\{x \leftarrow s\} & = & \left\{\begin{array}{ll} s & \text{if } x = y \\ y & \text{otherwise} \end{array}\right. & \\
(t\ u)\{x \leftarrow s\} & = & (t\{x \leftarrow s\})\ (u\{x \leftarrow s\}) & \\
(\lambda y.t)\{x \leftarrow s\} & = & \lambda y.t\{x \leftarrow s\} & \text{if } y \neq x \text{ and } y \notin \mathsf{fv}(s) \\
(\mathsf{let}\ y = u\ \mathsf{in}\ t)\{x \leftarrow s\} & = & \mathsf{let}\ y = u\{x \leftarrow s\}\ \mathsf{in}\ t\{x \leftarrow s\} & \text{if } y \neq x \text{ and } y \notin \mathsf{fv}(s)
\end{array}$$

We write Λ for the usual λ-calculus without let. Any term $t \in \Lambda^+$ can be *unfolded* to a term $(\!|t|\!) \in \Lambda$ by expanding its let-definitions as follows.

$$\begin{array}{rcl}
(\!|x|\!) & = & x \\
(\!|t\ u|\!) & = & (\!|t|\!)\ (\!|u|\!) \\
(\!|\lambda x.t|\!) & = & \lambda x.(\!|t|\!) \\
(\!|\mathsf{let}\ x = s\ \mathsf{in}\ t|\!) & = & (\!|t|\!)\{x \leftarrow (\!|s|\!)\}
\end{array}$$

We write $t \longrightarrow t'$ the reduction relation for terms in Λ^+. The relation is defined as follows.

$$\frac{}{(\lambda x.t)\ u \longrightarrow \mathsf{let}\ x = u\ \mathsf{in}\ t} \qquad \frac{}{\mathsf{let}\ x = s\ \mathsf{in}\ t \longrightarrow t\{x \leftarrow s\}} \qquad \frac{t \longrightarrow t'}{t\ u \longrightarrow t'\ u} \qquad \frac{t \longrightarrow t'}{\mathsf{let}\ x = s\ \mathsf{in}\ t \longrightarrow \mathsf{let}\ x = s\ \mathsf{in}\ t'}$$

$$\frac{s \longrightarrow s'}{\mathsf{let}\ x = s\ \mathsf{in}\ t \longrightarrow \mathsf{let}\ x = s'\ \mathsf{in}\ t}$$

Here is an example of a reduction sequence using this system:

$$\begin{array}{rcl}
(\lambda x.(\lambda y.y)x)\ a & \longrightarrow & \mathsf{let}\ x = a\ \mathsf{in}\ (\lambda y.y)x \\
 & \longrightarrow & \mathsf{let}\ x = a\ \mathsf{in}\ \mathsf{let}\ y = x\ \mathsf{in}\ y \\
 & \longrightarrow & \mathsf{let}\ x = a\ \mathsf{in}\ x \\
 & \longrightarrow & a
\end{array}$$

We write $\longrightarrow_\beta$ the usual β-reduction in Λ, and $\longrightarrow_\beta^*$ its reflexive-transitive closure.

Questions.

1. Prove that for any terms $s, t \in \Lambda^+$ and any variable x we have

$$(\!|t|\!)\{x \leftarrow (\!|s|\!)\} \quad = \quad (\!|t\{x \leftarrow s\}|\!)$$

Indication. You may assume the following lemma:

$$\forall x_1, x_2, t_0, t_1, t_2, \quad t_0\{x_1 \leftarrow t_1\}\{x_2 \leftarrow t_2\} \; = \; t_0\{x_2 \leftarrow t_2\}\{x_1 \leftarrow t_1\{x_2 \leftarrow t_2\}\}$$

2. Let $s, t \in \Lambda$ and x be a variable. We claim that:
   (a) if $t \longrightarrow_\beta t'$ then $t\{x \leftarrow s\} \longrightarrow^*_\beta t'\{x \leftarrow s\}$, and
   (b) if $s \longrightarrow_\beta s'$ then $t\{x \leftarrow s\} \longrightarrow^*_\beta t\{x \leftarrow s'\}$.

   Can you provide more precise information on the possible numbers of steps in these two reductions $t\{x \leftarrow s\} \longrightarrow^*_\beta$ ...? *Illustrate your answer with examples.*

3. Prove that for any two terms $t, t' \in \Lambda^+$, if $t \longrightarrow t'$ then $(\!|t|\!) \longrightarrow^*_\beta (\!|t'|\!)$.
   Indication. You may assume the claims of the previous question.

4. Consider a term $t \in \Lambda^+$ and a reduction of its unfolding $(\!|t|\!) \longrightarrow_\beta u$. Show that in some cases we can find a $t' \in \Lambda^+$ such that $t \longrightarrow t'$ and $(\!|t'|\!) = u$, but not always.

Exercise 3. CPS transformation and type preservation

Consider the usual definitions for terms (t) and simple types (T) of the λ-calculus, with β-reduction, where o is some base type.

$$\begin{array}{rcl} t & ::= & x \mid \lambda x.t \mid t\ t \\ T & ::= & o \mid T \longrightarrow T \end{array}$$

$$(\lambda x.t_1)\ t_2 \quad \longrightarrow_\beta \quad t_1\{x := t_2\}$$

Typing judgements $\Gamma \vdash t : T$ are derived using the following inference rules.

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T} \qquad \frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x.t : T_1 \longrightarrow T_2} \qquad \frac{\Gamma \vdash t_1 : T_2 \longrightarrow T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash t_1\ t_2 : T_1}$$

Questions.

1. Which of the following terms are well-typed? Provide a type derivation or explain the problem.
   (a) $(\lambda xy.x)$
   (b) $(\lambda xyz.(x\ z)\ (y\ z))$
   (c) $(\lambda fx.(x\ f)\ x)$
   (d) $(\lambda f.f\ (\lambda x.x))$

2. For any term t, define its *CPS transform* $[\![t]\!]$ by the following equations:

$$\begin{array}{rcl} [\![x]\!] & = & (\lambda\kappa.\kappa\ x) \\ [\![\lambda x.t]\!] & = & (\lambda\kappa.\kappa\ (\lambda x.[\![t]\!])) \\ [\![t_1\ t_2]\!] & = & (\lambda\kappa.[\![t_1]\!]\ (\lambda v_1.[\![t_2]\!]\ (\lambda v_2.v_1\ v_2\ \kappa))) \end{array}$$

   We want to apply the transformation $[\![.]\!]$ to the term d from the previous question.
   (a) Compute $[\![d]\!]$.
   (b) Reduce every β-redex in $[\![d]\!]$ and give its normal form.

3. We also define a translation on types:

$$[\![T]\!] \quad = \quad ([T] \longrightarrow o) \longrightarrow o$$

$$\begin{array}{rcl} [o] & = & o \\ [T_1 \longrightarrow T_2] & = & [T_1] \longrightarrow [\![T_2]\!] \end{array}$$

   and extend this translation to environments in the following way:

$$[\![x_1 : T_1, \ldots, x_n : T_n]\!] = x_1 : [T_1], \ldots, x_n : [T_n]$$

   Prove that if the judgment $\Gamma \vdash t : T$ is valid, then the judgment $[\![\Gamma]\!] \vdash [\![t]\!] : [\![T]\!]$ is also valid.