Computability Assignment
Year 2012/13 - Number 6

Please keep this file anonymous: do not write your name inside this file.

More information about assignments at http://disi.unitn.it/~zunino/teaching/computability/assignments

Please do not submit a file containing only the answers; edit this file, instead, filling the answer sections.

1 Question

Write a λ-term M implementing the following specification:

$$M \ulcorner n \urcorner = \ulcorner \lambda x_0 \ldots x_n.\ x_n x_{n-1} \cdots x_0 \urcorner$$

(Note: The notation $\ulcorner n \urcorner$ above stands for the numeral n, while $\ulcorner N \urcorner$ stands for $\ulcorner \# N \urcorner$ – inside LyX it's hard to tell them apart, but will appear correctly in the PDFs)

1.1 Answer

First we need to define two recursive support functions that can construct the solution of the exercise. The first called L will buil the part with the lambdas by incrementing a value and when it is equal to our number the second function will be called. The other function is A and build the second part of variables. Finally M just call L passing the initial value of the index that is zero.

$M = \lambda n.L \ulcorner 0 \urcorner$
$L = \Theta(\lambda li.Eq\, n\, i\, (Lam\, i\, (A\, n))\, (Lam\, i\, (l\, (Succ\, i))))$
$A = \Theta(\lambda aj.isZero\, j\, (Var \ulcorner 0 \urcorner)\, (App\, (Var\, j)\, (a\, (Pred\, j))))$

(RZ: the last line is building $x_n(x_{n-1}(\cdots x_0))$ instead of $((x_n x_{n-1}) \cdots x_0))$

2 Question

Write a λ-term M which, when given as input $\ulcorner N \urcorner$, evaluates to $\ulcorner O \urcorner$, where O is obtained from N by replacing every syntactic occurrence of Ω with $\mathbf{I}$.

To the purpose of this exercise, assume $\Omega = (\lambda x_0.x_0x_0)(\lambda x_0.x_0x_0)$ and $\mathbf{I} = \lambda x_0.x_0$.

For example, here are some expected outputs:

$$
\begin{aligned}
&M \ulcorner \lambda x_5.\Omega \urcorner =_{\beta\eta} \ulcorner \lambda x_5.\mathbf{I} \urcorner \\
&M \ulcorner \lambda x_3.\mathbf{K}\Omega \urcorner =_{\beta\eta} \ulcorner \lambda x_3.\mathbf{KI} \urcorner \\
&M \ulcorner \lambda x_1.x_1\Omega(\lambda x_7.x_1\Omega) \urcorner =_{\beta\eta} \ulcorner \lambda x_1.x_1\mathbf{I}(\lambda x_7.x_1\mathbf{I}) \urcorner \\
&M \ulcorner (\lambda x_0.x_0x_0)(\lambda x_0.x_0x_0) \urcorner =_{\beta\eta} \ulcorner \mathbf{I} \urcorner \\
&M \ulcorner (\lambda x_1.x_1x_1)(\lambda x_1.x_1x_1) \urcorner =_{\beta\eta} \ulcorner (\lambda x_1.x_1x_1)(\lambda x_1.x_1x_1) \urcorner
\end{aligned}
$$

Hint: use **Sd**, etc. as approprate.

2.1 Answer

From lesson we know that if two terms have the same value of # function, they are identical. By this consideration, we can do equality checks between parts of the term with the Ω and with the support of the Sd function. First we need to comput the value of Ω that is 449. Now we define a function M that recursively analyse the term: if it is equal to Ω then it returns the requested value, otherwise it proceed in the analysis. If we arrive in a situation in which we can have the Ω we simply delegate the check to the M function (for example in the A and L functions).

$$
\begin{aligned}
&M = \Theta(\lambda rm.Eq \ulcorner\ulcorner 449 \urcorner\urcorner \, m \ulcorner I \urcorner \, (Sd \, m \, V \, A \, L) \\
&V = \lambda i.Var \, i \\
&A = \lambda ab.App \, (r \, a) \, (r \, b) \\
&L = \lambda il.Lambda \, i \, (r \, l)
\end{aligned}
$$