Computer Science 131, Fall 2002
Assignment 11: λ-Calculus

Out: Wednesday, April 11

Due: Wednesday, April 18

1 Predecessor (30%)

For more practice working with λ-calculus, in this problem you will prove that the definition of predecessor that you saw in class is correct.

1. Prove that $\ulcorner 0 \urcorner\, M_1\, M_2 \longleftrightarrow^*_\beta M_2$ and that $\ulcorner n{+}1 \urcorner\, M_1\, M_2 \longleftrightarrow^*_\beta M_1(\ulcorner n \urcorner\, M_1\, M_2)$.

2. In class the predecessor function was defined as

$$\begin{array}{rcl} \mathbf{pred}' & = & \lambda m.m\ \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle\ (\lambda p.\langle \mathbf{snd}\ p, \mathbf{succ}(\mathbf{snd}\ p)\rangle) \\ \mathbf{pred} & = & \lambda n.\mathbf{fst}\ (\mathbf{pred}'\ n) \end{array}$$

   Prove that $\forall m \geq 0.\ \mathbf{pred}' \ulcorner m{+}1 \urcorner \longleftrightarrow^*_\beta \langle \ulcorner m \urcorner, \ulcorner m{+}1 \urcorner \rangle$.

3. Prove that $\forall m \geq 0.\ \mathbf{pred} \ulcorner m+1 \urcorner \longleftrightarrow^*_\beta \ulcorner m \urcorner$.

2 Lambda Calculus Encodings (50%)

For this problem you will devise an encoding for lists within the untyped λ-calculus. Recall that a list is either empty or it has a head (first element) and a tail (a list containing the rest of the elements, possibly empty).

1. Find a lambda term **nil** to represent the empty list and a lambda term **cons** such that $\mathbf{cons}\, M\ N$ represents the non-empty list with head M and with tail N. Then a finite list $[M_1, \ldots, M_n]$ would be represented as $\mathbf{cons}\, M_1\, (\mathbf{cons}\, M_2\, (\cdots (\mathbf{cons}\, M_n\, \mathbf{nil}) \cdots))$. Explain how to define the following functions for your encoding:

   - The function **isnil** that returns **tt** if given **nil** and returns **ff** if given a non-empty list.
   - The function **hd** that returns the head of a non-empty list.
   - The function **tl** that returns the tail of a non-empty list.

Verify that

$$\begin{aligned}
&\textbf{isnil nil} \longleftrightarrow^*_\beta \textbf{tt}\\
&\textbf{isnil}\ (\textbf{cons}\, M_1\ M_2) \longleftrightarrow^*_\beta \textbf{ff}\\
&\textbf{hd}\ (\textbf{cons}\, M_1\ M_2) \longleftrightarrow^*_\beta M_1\\
&\textbf{tl}\ (\textbf{cons}\, M_1\ M_2) \longleftrightarrow^*_\beta M_2
\end{aligned}$$

2. Find a lambda term **length** such that $\textbf{length}\, M \longleftrightarrow^*_\beta \ulcorner n \urcorner$ when M represents a finite list with n elements.

3. Find a lambda term **zeros** that represents an infinite list whose elements are all $\ulcorner 0 \urcorner$. Prove that it satisfies $\textbf{hd}(\textbf{tl}^{(n)}\, \textbf{zeros}) \longleftrightarrow^*_\beta \ulcorner 0 \urcorner$ for every $n \geq 0$.

3 Fixed Points (20%)

In class you saw the Y combinator for finding fixed points, which satisfies $YM \longleftrightarrow^*_\beta M(YM)$ for any term M. There are actually many terms other than Y which compute fixed points.

1. The Turing fixed-point combinator Θ is defined to be

$$(\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

   Prove that for any term M we have not just $\Theta M \longleftrightarrow^*_\beta M(\Theta M)$, but $\Theta M \rightarrow^*_\beta M(\Theta M)$.

2. For curried functions with many arguments, it is common to write terms leaving out all but the first lambda, for example abbreviating $\lambda x.\lambda y.\lambda z.\lambda w.M$ as $\lambda xyzw.M$. Define

$$\begin{aligned}
T &:= \lambda abcdefghijklmnopqstuvwxyzr.r(thisisafixedpointcombinator)\\
U &:= TTTTTTTTTTTTTTTTTTTTTTTTTT
\end{aligned}$$

   Show that U satisfies $UM \longleftrightarrow^*_\beta M(UM)$ for every term M.