Computer Science 131, Spring 2001
Assignment 9: λ-Calculus

Out: Wednesday, April 4

Due: Wednesday, April 11, 11:00am

This assignment involves no programming. Your written answers must be given directly to the professor (or put under his office door, 1253 Olin). Typewritten solutions (e.g., using LaTeX or Framemaker) are strongly preferred. You may submit handwritten work only if it is clearly legible; *the graders have been instructed to mark as wrong any part they have trouble reading.* If you submit your solution late, be sure to mark it with the date and time that it was handed in.

1 Fixed Points (10%)

In class you saw the Y combinator for finding fixed points, which satisfies $YM \longleftrightarrow_\beta^* M(YM)$ for any term M. There are in fact many terms other than Y which also compute fixed points. For example, the *Turing fixed-point combinator* is defined by

$$\Theta \;:=\; (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

Prove that for any term M we have $\Theta M \rightarrow_\beta^* M(\Theta M)$.

2 Predecessor (30%)

Prove that the definition of predecessor that you saw in class is correct:

1. Show 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

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

   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 $\forall m \geq 0.\ \mathbf{pred}\ \ulcorner m+1 \urcorner \longleftrightarrow_\beta^* \ulcorner m \urcorner$.

3 Encodings (40%)

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

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

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

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

4 λ-Definability [20%]

The following problems are key steps in proving that λ-calculus can define any function on natural numbers that is computable by a Turing machine.

1. **Primitive recursion:** Let two λ-terms M_1 and M_2 be given. In terms of these, find a λ-term F such that $F\ \ulcorner 0 \urcorner \longleftrightarrow_\beta^* M_1$, and $F\ \ulcorner n \urcorner \longleftrightarrow_\beta^* (M_2\ (F\ \ulcorner n{-}1 \urcorner)\ \ulcorner n \urcorner)$ for any $n \geq 1$.

   (You should be able to do this in general without knowing what M_1 and M_2 are. As a special case, if $M_1 = \ulcorner 1 \urcorner$ and $M_2 = \mathbf{mult}$ then the resulting expression F would compute factorial.)

2. **Minimization:** Find a term **mu** in the λ-calculus with the following property:

   > If $M\ \ulcorner n \urcorner \longleftrightarrow_\beta^* \ulcorner 0 \urcorner$ for at least one Church numeral $\ulcorner n \urcorner$, then $\mathbf{mu}\, M$ is convertible to the *least* Church numeral $\ulcorner m \urcorner$ such that $M\ \ulcorner m \urcorner \longleftrightarrow_\beta^* \ulcorner 0 \urcorner$.