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

April 17, 2001

1 Fixed Points (10%)

Put $D := (\lambda x.\lambda y.y(xxy))$, so that $\Theta = DD$. Then $\Theta M = (DD)M = ((\lambda x.\lambda y.y(xxy))D)(M) \rightarrow_\beta (\lambda.y.y(DDy))(M) \rightarrow_\beta M(DDM) = M(\Theta M)$.

2 Predecessor (30%)

1. (10%)

   - $\ulcorner 0 \urcorner\ M_1\, M_2$
     $= (\lambda f.\lambda b.b)\ M_1\, M_2$
     $\rightarrow_\beta (\lambda b.b) M_2$
     $\rightarrow_\beta M_2$
   - $\ulcorner n+1 \urcorner\ M_1\, M_2$
     $= (\lambda f.\lambda b.f^{(n+1)}(b))\ M_1\, M_2$
     $\rightarrow_\beta (\lambda b.M_1^{(n+1)} b) M_2$
     $\rightarrow_\beta M_1^{(n+1)}(M_2)$
     $= M_1(M_1^{(n)}(M_2))$
     $\leftarrow_\beta M_1((\lambda b.(M_1)^{(n)} b) M_2)$
     $\leftarrow_\beta M_1((\lambda f.\lambda b.f^{(n)}(b))\ M_1\, M_2) = M_1(\ulcorner n \urcorner\ M_1\, M_2)$

2. (20%)

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

   To show: $\forall m \geq 0.\ \mathbf{pred}'\ \ulcorner m+1 \urcorner \longleftrightarrow_\beta^* \langle \ulcorner m \urcorner, \ulcorner m+1 \urcorner \rangle$.

   By induction.

   - Case: $m = 0$. Then
     $\mathbf{pred}'\ \ulcorner 1 \urcorner$
     $= (\lambda m.m\ \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle\ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p) \rangle))(\ulcorner 1 \urcorner)$

$\to_\beta \ulcorner 1 \urcorner \ \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle))$
$\to_\beta{}^2 \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle))\langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle$
$\to_\beta{}^* \ \langle \ulcorner 0 \urcorner, \ulcorner 1 \urcorner \rangle$.

- Inductive step: Assume $\mathbf{pred}'\ \ulcorner m+1 \urcorner \longleftrightarrow_\beta^* \langle \ulcorner m \urcorner, \ulcorner m+1 \urcorner \rangle$; we need to show that $\mathbf{pred}'\ \ulcorner m+2 \urcorner \longleftrightarrow_\beta^* \langle \ulcorner m+1 \urcorner, \ulcorner m+2 \urcorner \rangle$.
  Then
  $\mathbf{pred}'\ \ulcorner m+2 \urcorner \to_\beta \ulcorner m+2 \urcorner \ \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle)$.
  By the previous part, this
  $\to_\beta{}^* \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle)(\ulcorner m+1 \urcorner \ \langle \ulcorner 0 \urcorner, \ulcorner 0 \urcorner \rangle \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle))$
  $\leftarrow_\beta \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle)(\mathbf{pred}'\ \ulcorner m+1 \urcorner)$. By the inductive hypothesis, this $\longleftrightarrow_\beta \ (\lambda p.\langle \mathbf{snd}\, p, \mathbf{succ}(\mathbf{snd}\, p)\rangle)\langle \ulcorner m \urcorner, \ulcorner m+1 \urcorner \rangle \to_\beta{}^* \langle \ulcorner m+1 \urcorner, \ulcorner m+2 \urcorner \rangle$.

3. $\mathbf{pred}\ \ulcorner m+1 \urcorner \to_\beta \mathbf{fst}\,(\mathbf{pred}'\ \ulcorner m+1 \urcorner) \longleftrightarrow_\beta{}^* \mathbf{fst}\,\langle \ulcorner m \urcorner, \ulcorner m+1 \urcorner \rangle \longleftrightarrow_\beta{}^* \ulcorner m \urcorner$.

3 Lambda Calculus 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. (20%)

   There are many, many possible encodings; here's a fairly direct encoding using pairing and booleans as defined in class. A list a pair whose first element is a boolean saying whether it is nil or not; if not, the second element is a pair containing the head and tail of the list.

   $$\begin{aligned}
   &\mathbf{nil} := \langle tt, M_0\rangle \\
   &\mathbf{cons} := \lambda h.\lambda t.\langle ff, \langle h, t\rangle\rangle \\
   &\mathbf{isnil} := \lambda l.(\mathbf{fst}\, l) \\
   &\mathbf{hd} := \lambda l.(\mathbf{fst}\,(\mathbf{snd}\, l)) \\
   &\mathbf{tl} := \lambda l.(\mathbf{snd}\,(\mathbf{snd}\, l))
   \end{aligned}$$

   where M_0 is arbitrary.

   Here's another; the idea is that lists are "things that one can do `listcase`-like operation on; if M_1 is a term representing a list, then $M_1\ M_2\ (\lambda x.\lambda y.M_3)$ will act like `listcase` M_1 `of nil =>` M_2 `| x::y =>` M_3.

   $$\begin{aligned}
   &\mathbf{nil} := (\lambda n.\lambda c.n) \\
   &\mathbf{cons} := \lambda h.\lambda t.(\lambda n.\lambda c.c\ h\ t) \\
   &\mathbf{isnil} := \lambda l.l\ tt\ (\lambda h.\lambda t.ff) \\
   &\mathbf{hd} := \lambda l.l\ M_0\ (\lambda h.\lambda t.h) \\
   &\mathbf{tl} := \lambda l.l\ M_0\ (\lambda h.\lambda t.t)
   \end{aligned}$$

   where M_0 is arbitrary. Then

   - $\mathbf{isnil}\ \mathbf{nil} \to_\beta{}^* (\lambda n.\lambda c.n)\ tt\ (\lambda h.\lambda t.ff) \to_\beta{}^2 tt$.

- $\mathbf{isnil}\,(\mathbf{cons}\,M_1\;M_2) \to_\beta{}^* \mathbf{isnil}(\lambda n.\lambda c.c\,M_1\;M_2) \to_\beta (\lambda n.\lambda c.c\,M_1\;M_2)\;tt\;(\lambda h.\lambda t.ff) \to_\beta{}^2 (\lambda h.\lambda t.ff)\;M_1\;M_2 \to_\beta{}^2 ff$.
- $\mathbf{hd}\,(\mathbf{cons}\,M_1\;M_2) \to_\beta{}^* \mathbf{hd}(\lambda n.\lambda c.c\,M_1\;M_2) \to_\beta (\lambda n.\lambda c.c\,M_1\;M_2)M_0(\lambda h.\lambda t.h) \to_\beta{}^2 (\lambda h.\lambda t.h)\;M_1\;M_2 \to_\beta{}^2 M_1$.
- $\mathbf{tl}\;(\mathbf{cons}\,M_1\;M_2) \to_\beta{}^* \mathbf{tl}(\lambda n.\lambda c.c\;M_1\;M_2) \to_\beta (\lambda n.\lambda c.c\;M_1\;M_2)M_0(\lambda h.\lambda t.t) \to_\beta{}^2 (\lambda h.\lambda t.t)\;M_1\;M_2 \to_\beta{}^2 M_2$.

2. (10%)

   Let $\mathbf{length} := Y(\lambda f.\lambda l.(\mathbf{isnil}\;l)\ulcorner 0\urcorner(\mathbf{succ}(f(\mathbf{tl}\;l))))$.

3. (10%)

   $\mathbf{zeros} := Y(\lambda l.\,\mathbf{cons}\;\ulcorner 0\urcorner\;l)$.

   First, note that $\mathbf{zeros} \longleftrightarrow_\beta ((\lambda l.\,\mathbf{cons}\;\ulcorner 0\urcorner\;l)\mathbf{zeros}) \longleftrightarrow_\beta \mathbf{cons}\;\ulcorner 0\urcorner\;\mathbf{zeros}$. Thus $\mathbf{hd}\;\mathbf{zeros} \longleftrightarrow_\beta \mathbf{hd}(\mathbf{cons}\;\ulcorner 0\urcorner\;\mathbf{zeros}) \longleftrightarrow_\beta \ulcorner 0\urcorner$ and $\mathbf{tl}\;\mathbf{zeros} \longleftrightarrow_\beta \mathbf{tl}(\mathbf{cons}\;\ulcorner 0\urcorner\;\mathbf{zeros}) \longleftrightarrow_\beta \mathbf{zeros}$. Hence $\mathbf{hd}(\mathbf{tl}^{(n)}\,\mathbf{zeros}) \longleftrightarrow_\beta^* \ulcorner 0\urcorner$ for every $n \geq 0$.

   The definition $Y(\mathbf{cons}\;\ulcorner 0\urcorner)$ is even simpler, while more convoluted definitions are possible as well.

4 λ-Definability (20%)

1. (10%) One possibility is $F := Y(\lambda f.\lambda n.(\mathbf{iszero}\,n)\;M_1\;(M_2\;(f\;(\mathbf{pred}\,n))\;n))$.

2. (10%) We want a lambda term equivalent to the SML code

```
fun mu m =
  let
     fun loop n =
       if (m(n) = 0) then
          n
       else
          loop (n+1)
  in
     loop 0
  end
```

   One possibility is $\mathbf{mu} := \lambda m.(Y(\lambda f.\lambda n.(\mathbf{iszero}\;(m\;n)\;n\;(f\;(\mathbf{succ}\,n)))))\ulcorner 0\urcorner$.