Theory of Computation Homework 5: Solutions

December 5, 2004

This assignment is due on Tuesday, November 23, 2004.

1. In lecture I presented five schemas for defining primitive recursive functions. They are as follows:

   (a) [Zero] There is a constant function zero of every arity.

   $$Z^k(x_1, \ldots, x_k) = 0$$

   (b) [Successor] There is a successor function of arity 1.

   $$S(x) = x + 1$$

   (c) [Projection] There are projection functions for every argument position of every arity.

   $$P_i^k(x_1, \ldots, x_k) = x_i \qquad \text{where } k > 0,\ i \leq k$$

   (d) [Composition (also called substitution)] The composition of the function f of arity k with functions $g_1, \ldots g_k$, each of arity l, defines a $f \circ_k^l [g_1 \ldots g_k]$ of arity l satisfying:

   $$f \circ_k^l [g_1 \ldots g_k](x_1, \ldots, x_l) = f(g_1(x_1, \ldots, x_l), \ldots, g_k(x_1, \ldots, x_l))$$

   (e) [Primitive Recursion] The arity k function defined by primitive recursion from a function g of arity $k-1$ and a function h of arity $k+1$ is indicated $\mathrm{PR}^k[g, h]$. It satisfies:

   $$\begin{array}{lcl} \mathrm{PR}^k[g,h](0, x_2, \ldots, x_k) & = & g(x_2, \ldots, x_k) \\ \mathrm{PR}^k[g,h](x+1, x_2, \ldots, x_k) & = & h(x, \mathrm{PR}^k[g,h](x, x_2, \ldots, x_k), x_2, \ldots, x_k) \end{array}$$

   In lecture we showed how to define addition by primitive recursion:

   $$\mathrm{PR}^2[P_1^1, S \circ_1^3 [P_2^3]]$$

   Using primitive recursion define:

(a) Multiplication

Answer: Using + to represent the addition function given above:

$$\mathrm{PR}^2[Z^1, +\circ_2^3 [P_2^3, P_3^3]]$$

(b) Bounded quantification (see M&Y)

Answer: There are two functions to be defined:

i. $\mathrm{BEQ}[\mathcal{P}] = \exists x < y.\mathcal{P}(x)$

ii. $\mathrm{BUQ}[\mathcal{P}] = \forall x < y.\mathcal{P}(x)$

The functions are defined via primitive recursion on y, so the order of arguments is y first, x second. It helps to define if-then-else and "boolean" functions (here I use 0 for false and 1 for true) first.

$$\begin{aligned}
\mathrm{ITE} &= \mathrm{PR}^3[P_2^2, P_3^4] \\
\mathrm{OR} &= \mathrm{ITE} \circ_3^2 [P_1^2, S \circ Z^2, P_2^2] \\
\mathrm{AND} &= \mathrm{ITE} \circ_3^2 [P_1^2, P_2^2, Z^2] \\
\mathrm{BEQ}[\mathcal{P}] &= \mathrm{PR}^2[Z^1, \mathrm{OR} \circ_2^3 [\mathcal{P} \circ_1^3 P_3^3, P_2^3]] \\
\mathrm{BUQ}[\mathcal{P}] &= \mathrm{PR}^2[S \circ_1^1 Z^1, \mathrm{AND} \circ_2^3 [\mathcal{P} \circ_1^3 P_3^3, P_2^3]]
\end{aligned}$$

2. In the λ-calculus values are encoded by the control structures that analyze them. Booleans, thus, are represented by the equivalent of if-then-else:

$$\mathit{true} = \lambda t.\lambda f.t \qquad \mathit{false} = \lambda t.\lambda f.f$$

(a) Using this representation define:

i. and

Answer: $\lambda a.\lambda b.a\ (b\ \mathit{true}\ \mathit{false})\ \mathit{false} = \lambda a.\lambda b.a\ b\ \mathit{false}$

ii. or

Answer: $\lambda a.\lambda b.a\ \mathit{true}\ (b\ \mathit{true}\ \mathit{false}) = \lambda a.\lambda b.a\ \mathit{true}\ b$

iii. not

Answer: $\lambda a.a\ \mathit{false}\ \mathit{true}$

(b) Over Church numerals the key is to use them as iterators. Recall the definition of the successor function given in class:

$$\lambda n.\lambda s.\lambda z.s(n\ s\ z)$$

Use this to define addition (we did this in class too) and multiplication. See how the number n is represented by a loop that applies its first argument n times to its second argument. In the successor function above the first argument (s) gets applied one more time. That is the essence of the successor function.

Answer:

$$\mathit{plus} = \lambda x.\lambda y.x\ \mathit{succ}\ y$$

$$\mathit{times} = \lambda x.\lambda y.x\ (\lambda z.\mathit{plus}\ y\ z)\ 0$$

(c) Pairing and projection operators can be defined in the same manner. The function below can construct pairs.

$$mkpair = \lambda a.\lambda b.\lambda c.c\ a\ b$$

These pairs are analyzed by providing the correct projection function. The first projection function is:

$$\pi_1 = \lambda p.p(\lambda x.\lambda y.x)$$

Define the second projection function (π_2).

Answer:

$$\pi_2 = \lambda p.p(\lambda x.\lambda y.y)$$

(d) Define a function that maps 0 to the representation of the pair $(0, 0)$, and maps every other natural number n to $(n, n-1)$. Use this function to define the predecessor function.

Answer:

$$aux = \lambda n.n(\lambda p.mkpair(s(\pi_1 p))(\pi_1 p))(mkpair\ 0\ 0)$$

$$pred = \lambda n.\pi_2(aux\ n)$$

(e) Define the monus function, which returns the difference of two numbers if the difference is non-negative. If the difference is negative monus should return zero. [Hint: use the predecessor function.]

Answer:

$$monus = \lambda a.\lambda b.b\ pred\ a$$

(f) Define an integer equality function.

Answer:

$$isZero = \lambda n.n(\lambda p.false)\ true$$

$$eq = \lambda a.\lambda b.and\ (isZero\ (monus\ a\ b))(isZero\ (monus\ b\ a))$$

(g) Argue convincingly that the factorial function given in class is definable in the lambda calculus:

$$FACT = \lambda fact.\lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * fact(n-1)$$

Answer: It has been demonstrated that natural numbers, multiplication, subtraction, booleans/if-then-else, and equality over the natural numbers can be defined in the lambda calculus, so all the pieces needed to define $FACT$ are available.

(h) Illustrate a fragment of the computation of $(Y\,FACT)(\lambda s.\lambda z.s(sz))$. (Recall that $Y = \lambda f(\lambda x.f(xx))(\lambda x.f(xx))$)

Answer:

$$\begin{array}{rcl}
(Y\ FACT)\ 2 & = & ((\lambda f.(\lambda x.f(x\ x))(\lambda x.f(x\ x)))FACT)\ 2 \\
& \Rightarrow_\beta & (\lambda x.FACT(x\ x))(\lambda x.FACT(x\ x))\ 2 \\
& \Rightarrow_\beta & FACT((\lambda x.FACT(x\ x))(\lambda x.FACT(x\ x)))\ 2 \\
& = & FACT\ (Y\ FACT)\ 2 \\
& = & (\lambda fact.\lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * fact(n-1))(Y\ FACT)\ 2 \\
& \Rightarrow_\beta & (\lambda n.\text{if } n = 0 \text{ then } 1 \text{ else } n * (Y\ FACT)(n-1))\ 2 \\
& \Rightarrow_\beta & \text{if } 2 = 0 \text{ then } 1 \text{ else } n * (Y\ FACT)(2-1) \\
& = & \ldots
\end{array}$$

3. (a) Show that all primitive recurisve functions are definable in the lambda calculus by giving lambda terms for every schema. You may assume any of the results of the previous problem, even if you didn't solve it. Illustrate your construction by showing the translation of the addition function given in the first exercise.

Answer: Shorthand: x_{i-j} means $x_i \ldots x_j$, and λx_{i-j} means $\lambda x_i \ldots \lambda x_j$.

$$\begin{array}{rcl}
Z^k & = & \lambda x_{1-k}.\lambda s.\lambda z.z \\
S & = & \lambda n.\lambda s.\lambda z.s(n\ s\ z) \\
P_i^k & = & \lambda x_{1-k}.x_i \\
f \circ_k^l g_{1-k} & = & \lambda x_{1-l}.f\ (g_1\ x_{1-l}) \ldots (g_k\ x_{1-l}) \\
\mathrm{PR}^2[g,h] & = & \lambda x_1.\lambda x_2.\pi_2(doPR\ g\ h\ x_1\ x_2) \\
doPR & = & \lambda g.\lambda h.\lambda x_1.x_1(\lambda p.mkpair(S(\pi_1 p))(h(\pi_1 p)(\pi_2 p)x_2))(mkpair\ 0\ (g\ x_2))
\end{array}$$

OR

$$\mathrm{PR}^k[g,h] = (Y(\lambda f.\lambda x_{1-k}.(isZero\ x_1)\ (g\ x_{2-k})(h(pred\ x_1)\ (f(pred\ x_1)\ x_{2-k})\ x_{2-k})))$$

Addition was defined above as $\mathrm{PR}^2[P_1^1, S \circ_1^3 [P_2^3]]$. The projections are translated as follows:

$$\begin{array}{rcl}
P_1^1 & = & \lambda x_1.x_1 \\
P_2^3 & = & \lambda x_1.\lambda x_2.\lambda x_3.x_2
\end{array}$$

The composition is therefore:

$$\begin{array}{rcl}
S \circ_1^3 P_2^3 & = & \lambda x_1.\lambda x_2.\lambda x_3.(\lambda n.\lambda s.\lambda z.s(n\ s\ z))((\lambda x_1.\lambda x_2.\lambda x_3.x_2)\ x_1\ x_2\ x_3) \\
& \Rightarrow_\beta & \lambda x_1.\lambda x_2.\lambda x_3.(\lambda n.\lambda s.\lambda z.s(n\ s\ z))\ x_2 \\
& \Rightarrow_\beta & \lambda x_1.\lambda x_2.\lambda x_3.\lambda s.\lambda z.s(x_2\ s\ z)
\end{array}$$

Putting it all together (using the Y formulation of PR):

$$\begin{array}{rcl} \mathit{plus} & = & Y(\lambda f.\lambda x_1.\lambda x_2.(\mathit{isZero}\ x_1)\ x_2 \\ & & ((\lambda y_1.\lambda y_2.\lambda y_3.\lambda s.\lambda z.s(y_2\ s\ z))\ (\mathit{pred}\ x_1)\ (f(\mathit{pred}\ x_1)\ x_2)\ x_2) \\ & \Rightarrow_\beta & Y(\lambda f.\lambda x_1.\lambda x_2.(\mathit{isZero}\ x_1)\ x_2\ (s\ ((f(\mathit{pred}\ x_1)\ x_2)\ s\ z))) \\ & = & Y(\lambda f.\lambda x_1.\lambda x_2.(\mathit{isZero}\ x_1)\ x_2\ (s\ (f(\mathit{pred}\ x_1)\ x_2))) \end{array}$$

(b) Recall the minimization schema:
The function of arity k defined by minimization of a function f of arity $k+1$, written μf, satisfies:

$$\mu f(x_1,\ldots,x_k) = \text{the least } x \text{ such that } f(x,x_1,\ldots,x_k) \neq 0 \text{ and for all } y < x,\ f(y,x_1,\ldots,x_k) \text{ is defined and equal to } 0$$

Show that functions defined by minimization can be defined by the lambda calculus.
Answer:

$$\begin{array}{rcl} \mathit{min}\ f & = & \mathit{min}'\ f\ 0 \\ \mathit{min}'f & = & \lambda x_{0-k}.\text{if } f\ x_0 = 0 \text{ then } \mathit{min}'\ f\ (x_0+1)\ x_{1-k} \text{ else } x_0 \\ \mathit{min}' & = & Y\ \mathit{MIN} \\ \mathit{MIN} & = & \lambda \mathit{min}'.\lambda f.\lambda x_{0-k}.\text{if } f\ x_0 = 0 \text{ then } \mathit{min}'\ f\ (x_0+1)\ x_{1-k} \text{ else } x_0 \end{array}$$