Theory of Computation Assignment 8

November 23, 2005

This assignment is due on Thursday, December 1, 2005.

1 Lambda Calculus

1.1 Exercises

1. Pairing and projection operators can be defined in a similar manner to Church numerals and Booleans. 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 functions satisfy the laws:

$$\begin{aligned} \pi_1(mkpair\ a\ b) &= a \\ \pi_2(mkpair\ a\ b) &= b \end{aligned}$$

The first projection function is:

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

Define the second projection function (π_2).

2. 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.

3. 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.]

4. Define an integer equality function.

5. Argue convincingly that the factorial function below is definable in the lambda calculus:

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

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

2 Representability of recursive functions in lambda calculus

In this section you will demonstrate that all primitive and partial recursive functions are definable in the lambda calculus.

When answering these questions note that in many cases the function definition schema has parameters that vary with the arity. It is acceptable to give a family of lambda terms to implement the schema. For example the constant function zero of arity k is given by the family of terms:

$$\lambda x_1. \ldots .\lambda x_k.\lambda s.\lambda z.z$$

2.1 Exercies

1. 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.

   My implementation of the primitive recursion schema uses some of the same techniques as the implementation of predecessor. If it helps, pick a fixed arity of PR to implement.

2. Illustrate your construction by showing the translation of the addition function given in the first exercise.

3. 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.

   My implementation of the minimization schema makes essential use of the fixed point combinator used in the factorial example.