

CM20167: More programming in the λ -calculus

November 23rd 2007

Consider the following alternative encoding of lists in the λ -calculus: a list $(M_1 M_2 \dots M_k)$ is encoded as the term

$$\lambda c. \lambda n. c M_1 (c M_2 (c M_3 \dots (c M_k n))).$$

1. Verify that the function

$$\lambda e. \lambda l. \lambda c. \lambda n. c e (l c n)$$

encodes cons and that the function

$$\lambda l. l (K(KF)) T$$

encodes the test for null; here K is the function $\lambda x. \lambda y. x$ and T and F are appropriate encodings of the truth values true and false.

Hint First work out what it would mean for these to be the correct encodings, i.e. decide what properties such encodings should have; then check that they do indeed have those properties.

2. Write λ -terms which encode car and cdr appropriately. Again, you should first work out what it would mean to encode these functions, and also verify that your solutions work as expected.
3. Without using a Y combinator, define λ -calculus terms which encode the functions
 - (a) **length**, which calculates the length of a list (you may assume that we have appropriate encodings of numbers and the usual operators of arithmetic, and use them in your solution however you like)
 - (b) **sum**, which adds together the elements of a list (assuming they are all numbers)
 - (c) **append**, which takes two lists and returns the list you get by joining them together i.e. concatenating them

- (d) `map`, which takes a function f and a list l and returns the list obtained by applying f to each element of l
- (e) `foldr`, as defined in lectures a few weeks ago.

For added fun, try to encode `reverse`, which reverses a list.

Hint The first couple can be solved by supplying the encoding of a list with appropriate functions for the c and n arguments. If you are comfortable with the `foldr` function in Lisp, you might prefer to attempt question (3e) first.