Harvard School of Engineering and Applied Sciences — CS 152: Programming Languages

Lambda calculus encodings and Recursion; Definitional translations (Lectures 8–9) Section and Practice Problems

Week 5: Tue Feb 20-Fri 23, 2018

---

1 Lambda calculus encodings

(a) Evaluate $\mathit{AND}\ \mathit{FALSE}\ \mathit{TRUE}$ under CBV semantics.

(b) Evaluate $\mathit{IF}\ \mathit{FALSE}\ \Omega\ \lambda x.\, x$ under CBN semantics. What happens when you evaluated it under CBV semantics?

(c) Evaluate $\mathit{ADD}\ \overline{2}\ \overline{1}$ under CBV semantics. (Make sure you know what the Church encoding of 1 and 2 are, and check that the answer is equal to the Church encoding of 3.)

(d) In class we made use of a combinator ISZERO, which takes a Church encoding of a natural number n, and evaluates to TRUE if n is zero, and FALSE if n is not zero. (We don't care what ISZERO does if it is applied to a lambda term that is not a Church encoding of a natural number.)

Define ISZERO.

2 Recursion

Assume we have an applied lambda calculus with integers, booleans, conditionals, etc. Consider the following higher-order function H.

$$H \triangleq \lambda f.\, \lambda n.\, \mathsf{if}\ n = 1\ \mathsf{then\ true\ else\ if}\ n = 0\ \mathsf{then\ false\ else\ not}\ (f\ (n-1))$$

(a) Suppose that g is the fixed point of H. What does g compute?

(b) Compute $Y\ H$ under CBN semantics. What has happened to the function call $f\ (n-1)$?

(c) Compute $(Y\ H)\ 2$ under CBN semantics.

(d) Use the "recursion removal trick" to write another function that behaves the same as the fixed point of H.

3 Evaluation context

Consider the lambda calculus with let expressions and pairs (§1.3 of Lecture 9), and a semantics defined using evaluation contexts. For each of the following expressions, show one step of evaluation. Be clear about what the evaluation context is.

(a) $(\lambda x.\, x)\ (\lambda y.\, y)\ (\lambda z.\, z)$

(b) $\mathsf{let}\ x = 5\ \mathsf{in}\ (\lambda y.\, y + x)\ 9$

(c) $(4, ((\lambda x.\, x)\ 8, 9))$

(d) $\mathsf{let}\ x = \#1\ ((\lambda y.\, y)\ (3, 4))\ \mathsf{in}\ x + 2$

4 Definitional translations

Consider an applied lambda calculus with booleans, conjunction, a few constant natural numbers, and addition, whose syntax is defined as follows.

$$e ::= x \mid \lambda x.\, e \mid e_1\ e_2 \mid \mathsf{true} \mid \mathsf{false} \mid e_1 \ \mathsf{and}\ e_2 \mid 0 \mid 1 \mid 2 \mid e_1 + e_2 \mid \mathsf{if}\ e_1\ \mathsf{then}\ e_2\ \mathsf{else}\ e_3$$

Give a translation to the pure lambda calculus. Use the encodings of booleans and natural numbers that we considered in class.