1 Lambda calculus encodings

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

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

(c) Evaluate \( \text{ADD } 2 \ T \) 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 \( \text{ISZERO} \), which takes a Church encoding of a natural number \( n \), and evaluates to \( \text{TRUE} \) if \( n \) is zero, and \( \text{FALSE} \) if \( n \) is not zero. (We don’t care what \( \text{ISZERO} \) does if it is applied to a lambda term that is not a Church encoding of a natural number.) Define \( \text{ISZERO} \).

2 Recursion

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

\[
H \triangleq \lambda f. \lambda n. \text{if } n = 1 \text{ then } \text{true} \text{ else if } n = 0 \text{ then } \text{false} \text{ else } f (n - 1)
\]

(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) \( \text{let } x = 5 \ \text{in } \lambda y. y + x \) \( 9 \)

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

(d) \( \text{let } x = \#1 ((\lambda y. y) \ (3) \ 4) \ \text{in } x + 2 \)
4 Definitional translations

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

\[ e ::= x | \lambda x. e | e_1 e_2 | \text{true} | \text{false} | e_1 \text{ and } e_2 | 0 | 1 | 2 | e_1 + e_2 | \text{if } e_1 \text{ then } e_2 \text{ else } e_3 \]

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