

# Advanced Programming Languages (COP 4930/CIS 6930) [Spring 2015]

## Assignment II

**Due Date:** Monday 2/2/15 at 5pm (please turn in solutions in hardcopy at the beginning of class).

### Assignment Description

Do the following by yourself (please don't discuss solutions until after the due date).

1. Let  $L$  be an untyped lambda calculus having functions, applications, variables, and natural numbers, all defined as usual.

- a) Define a function  $FV$  that returns the set of free (i.e., used but undeclared) variables in a given  $L$ -expression.
- b) Define a function  $BV$  that returns the set of bound (i.e., declared) variables in a given  $L$ -expression.
- c) Define a function  $V$  that returns the set of (used or declared) variables in a given  $L$ -expression.
- d) Prove the following theorem or provide a counterexample.

**Theorem.** For all  $L$ -expressions  $e$ :  $|FV(e)| + |BV(e)| = |V(e)|$

(NB: You don't need to define deductive systems for addition or set-size operators; please just use our normal rules and understanding of these judgments.)

2. Suppose a call-by-value language  $X$  has recursive types  $(\mu.t)$ , with the relevant constructs defined as in class—but with dynamic semantics defined using evaluation contexts. Show all the parts of  $X$ 's type-safety proof that are specific to recursive types (or roll/unroll expressions). You'll need to provide proofs for several cases of various lemmata.