Advanced Programming Languages	**Problem Set 4**
CS 6110 Spring 2015	**Due: Fri. April 24, 2015**

Problem Set 4

Exercises

1. Recall our discussion of the *fixed point induction* rule of Edinburgh LCF in Lecture 24 (March 20). This topic is covered in *Winskel Chapter 10* section 10.2. He proves the version of fixed point induction studied by Scott, Proposition 10.4. It requires that the predicate P be an *inclusive subset* of the domain D. The domain corresponds to LCF types. The LCF notion of admissible proposition is a generalization.

   (a) Consider $\overline{\mathbb{N}}$ as a complete partial order (CPO) with $\bot \sqsubseteq n$ for n any natural number. Read Winskel's account of inclusive predicates on page 167. Show that $x \sqsubseteq y$ and $x = y$ on $\overline{\mathbb{N}} \times \overline{\mathbb{N}}$ are inclusive.

   (b) Find a predicate on $\overline{\mathbb{N}} \to \overline{\mathbb{N}}$ that is *not inclusive* and for which fixed point induction fails. (This will be *inadmissible* in the sense of Edinburgh LCF.)

   (c) Find the one sentence on page 172 of Winskel that hints at the failure of LCF to become the logic of modern proof assistants.

2. Show how to enumerate the shortest primitive recursive programs using a standard notion of size. Consider only one argument functions for simplicity.

3. Finish the proof of Rice's Theorem from *Computational foundations of basic recursive function theory* (Theorem 3.10 page 100, case $f(\bot) = 1$).

4. Consider the recursive function $\mathbb{N} \to \overline{\mathbb{N}}$,

   $f(x) = \textbf{if } x = 1 \textbf{ then } 0 \textbf{ else if } even(x) \textbf{ then } f(x/2) \textbf{ else } f(3x+1)$

   To show that f has type $\mathbb{N} \to \mathbb{N}$ corresponds to solving the Collatz conjecture.

   (a) Read about this conjecture and sketch a proof of this claim.

   (b) Discuss how this "function" could be studied in a logic that only allows total functions such as generalizations of Goodstein's recursive arithmetic or Coq.

5. (a) Consider another way in which the Hoare function rule might be "fixed". Consider how partial types are used and whether partial propositions might make sense.

   (b) Consider the following definition of a recursive function, clearly in type $\mathbb{Z} \to \overline{\mathbb{Z}}$.

   $f(x) = \textbf{ if } x > 100 \textbf{ then } x - 10 \textbf{ else } f(f(x+11)) \textbf{ fi}$ show that f is equal to the "91-function", i.e. let $f_{91}(x) = \textbf{ if } x > 100 \textbf{ then } x - 10 \textbf{ else } 91 \textbf{ fi}$ and prove that $f(x) = f_{91}(x)$ for all $x \in \mathbb{Z}$.

   We can also use the type $\mathbb{N}$ and $\overline{\mathbb{N}}$ if we use $x \dot{-} 10$, e.g. for $x \leq 10$ the value is 0.