Problem Set 3 Due Friday Mar 8
CS6116 students should solve Problems 2 and 5

Problem 1
Revisiting Euclid's Theorem in Prim Rec Arith.

Euclid's Theorem symbolically:

$$\forall a, b{:}\mathbb{N}.\exists\, d{:}\mathbb{N}.div(d,a)\ \&\ div(d,b)\ \&\ \forall z{:}\mathbb{N}.(div(z,a)\ \&\ div(z,b) \Rightarrow z \leq d)$$

If we let $div(x,y) = E^y_k(k * x) = y$, then using Goodstein's notation, Euclid's Theorem is

$$A^n_a(A^a_b(E^a_d(div(d,a)\ \&\ div(d,b))))\ \&\ A^n_z(A^n_a(A^a_b(div(z,a)\ \&\ div(z,b)) \Rightarrow (z \dot{-} d = 0)))$$

(a) The key step in proving this theorem is witnessing E^a_d. A function $gcd(a,b)$ computing the greatest common divisor is a witness for the existential quantifier. Write a primitive recursive function to compute *gcd*.

(b) Write a while program to compute *gcd* and add assertions that document the code and provide a basis for proving that the program is correct.

(c) Prove that the while program is correct either using the structural operational semantics or the Binary Relation Semantics — both in CS6110 2010 Lecture 17 posted with Monday's lecture.

(d) Sketch a Hoare Logic asserted program *total correctness* proof.

Problem 2
Define the integer square root of a non-negative number n to be the least non-negative number r such that $r^2 \leq n < (r+1)^2$. Call this relation $Root(r,n)$.

(a) Write an IMP program to compute r.

(b) Prove that your program is *totally correct*, either using asserted programs with a termination predicate (see notes on PL/CV) or using structured operational semantics or Relational Semantics (on-line notes for Lect 16) or positional semantics.

Problem 3
Write a careful proof using Relational Semantics, that the Hoare while rule is valid for partial correctness.

Problem 4
Pick an imperative programming language that you know well (e.g. Java, Python, C, C^{++}, etc.) and explain at least four caveats that need to be stated in order to make the Hoare assignment rule valid.

Problem 5

(a) Define a function $\lambda(x.big(x))$, related to the functions f_n in Def 3.2 page 3 of Meyer and Ritchie's paper, which is not Loop computable, thus not primitive recursive.

(b) Sketch a proof that $\lambda(x.big(x))$ is not Loop computable.

(c) For **extra credit**, write the continuation passing form of this function.

Problem 6 - More Extra Credit
Write your intuitive explanation of why continuation passing computations are tail recursive.