Computational models – Home Assignment 1

Fall 2008 - 10/11/08

Guildelines: When presenting a function or an algorithm you can use scheme/java/c/baby syntax, or any pseudo version of those so long as the explanation is clear, precise and unambiguous.

1. Prove without using a reduction that the function $MinProg:N \rightarrow N$ is not computable. Reminder: $MinProg(n) = \min\{|p|: [p()] \geq n, p \text{ is a program with no input in a programming language } L\}$

2. Show a reduction and prove correctness:

   a. From *bb* to *halt*.
      What can be proved using this reduction?
   b. From *MinProg* to *bb*.
      What can be proved using this reduction?

   Reminder: $bb(n) = \max\{k : k \in N, \exists p, p \in L_0, |p| \leq n, [p()]=k\}$, halt(p) – given a string *p* decide whether *p* is a program in *L* that halts.

3. Are the following predicates computable? Prove correctness.
   a. Given a program *P* with no input, does *P* return '2008'?
   b. Given two programs P_1 and P_2 with no input, do P_1 and P_2 have the same output?

4. Programs are executed in discrete steps. One operation is done in each step.
   a. Prove that the function $MaxSteps:N \rightarrow N$ is not computable, where $MaxSteps(n) = \max\{k: k \in N, \exists p, p \in L, |p| \leq n, p \text{ executes in } k \text{ steps}\}$
   b. Assume *MaxSteps* is computable. Show an algorithm for proving that Fermat's last theorem is correct ($\forall n \geq 2, a,b,c > 0\ a^n+b^n \neq c^n$).

5. A man with a wolf, a goat and a cabbage are on the west bank of the river. They need to cross to the east bank. There is a boat large enough to carry only two of them at a time and the man is the only one who can row. If the wolf and the goat are left alone at one bank, the wolf will eat the goat. If the goat and the cabbage are left alone at one bank, the goat will eat the cabbage. Write down a suitable State Transition System diagram and show a computation that proves there is a solution to the problem.

6. Recall the counter machine that multiplies by two presented in lecture 1. Describe this counter machine as a State Transition System.

7.

a. Are the following predicates computable?
    i. Given a finite STS (an STS with a finite number of states), does it have a finite accepting (terminating) computation?
    ii. Given a finite STS and a number n, does it have an accepting (terminating) computation whose length is n?
b. Construct a finite STS such that the set of lengths of its accepting (terminating) computations is $\{2^n \bullet 3^{bb(n)} \mid n \in N\}$ or prove such an STS cannot be constructed.