Exam in Models of Computation TDA 183

Date: Dec 14, 2009, 14.00 - 17.00

Permitted aids: English-Swedish or English-other language dictionary.

Teacher: Bengt Nordström, phone 070 600 1546, Computer Science, University of Gothenburg and Chalmers

All solutions must be explained! It is not enough to just give a program without an explanation of why it works. The examination of the course consists of three parts: homework assignments count up to 40 points, weekly exercises count up to 20 points and this written exam count up to 140 points (20 points for each problem). You have to have 100 points in total in order to pass the course.

Solutions to the exam will be available from the homepage of the course.

1. Answer the following questions (with a proof)!
   (a) Is the set **Fin(N)** of all finite subsets of **N** enumerable?
   (b) Is the set **Pow(N)** of all subsets of **N** enumerable?
2. Give an example of a fixpoint combinator in lambda-calculus and show how this is used to solve an equation like:

   ```
   fact n = if (iszero n) one (mult n (fact (pred n)))
   ```

   Here we assume that `if`, `iszero`, `one`, `mult` and `pred` are terms in lambda-calculus.
3. The bounded minimization operator is defined informally by:

   ```
   bmin(g)[m, j1, ..., jn] = k
   ```

   if `k` is the least natural number less than `m` such that

   ```
   g[k, j1, ..., jn] = 0
   ```

   If no such number exists, then the value is `m`.

   Show how to add this to **PRF** (the language of primitive recursive functions) by explaining what clauses have to be added to the definition of the abstract syntax and operational semantics of **PRF**. There is no need to write the clauses for the operators of **PRF**.

4. The **bmin** operator does not add any expressive power to **PRF**, unlike the operator **min** which expresses unbounded minimization. So we could say that **PRF = PRF + bmin**, while **RF = PRF + min**. Here **RF** is the computation model of recursive functions. Give an example of a program `g` in **RF** which has a root which cannot be computed by `min`. So you have to find a program `g` with the two properties:

   ```
   g[l, j1, ..., jn] = 0   for some natural number l
   ```

   and

   ```
   min(g)[j1, ..., jn] is not defined
   ```

5. Give an example of a term in lambda-calculus which - depending on the evaluation strategy - sometimes terminates, sometimes does not terminate. Explain why!

6. Describe Turing's model of computation (Turing machine)! Describe also what it means for a function `f : N -> N` to be computable by a Turing machine.

7. Explain how the mathematical object 1 (the natural number) is represented in lambda-calculus, in the language **X** (using concrete syntax) and in a Turing machine.

Good Luck!

Bengt