Exam in Models of Computation TDA 183

Date: April, 6, 2010, 14.00 - 18.00

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

2. What does it mean that a function `f : N -> N` is computable in lambda-calculus?

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

4. Show that the union **A** U **B** of two enumerable sets **A** and **B** is enumerable.

5. Assume that the sets **A_i** are enumerable (i : Nat). Show that the infinite union **A_1** U **A_2** U ... U **A_n** U ... is also enumerable. (There is no need to give an explicit enumeration function, it is enough to describe it in such a detail that it is possible to see that it exists.)

6. Show how to extend the language **PRF** of primitive recursive functions to a language **RF** (recursive functions) so that it can express all computable functions!

7. Define what it means for a program in **RF** to be a self-evaluator. In order to do this you have to explain how to represent **RF**-programs in **RF**. There is no need to do this in full detail, just outline the construction.

Good Luck!

Bengt