From Implicit Complexity to Quantitative Resource Analysis
July 24, 2015 — Written Assignment
Due on August 15, 2015 by mail at `ugo.dallago@unibo.it`

Some remarks:

- If possible, typeset your manuscript in LaTeX.
- Try to solve all the exercises (even if doing all of them correctly is not necessary to pass the exam).
- Questions about the assignment need to be addressed to `ugo.dallago@unibo.it`. If received before August 10, they will be handled in at most 48 hours.

Esercise 1. Prove that the set of functions which can be represented in the λ-calculus (following the definition given in the slides) is the set of all partial recursive functions.

Esercise 2. Consider a natural generalisation BC_Σ of the function algebra BC defined in the first part of the course. In BC_Σ, the domain and codomain of the functions both are the set of free terms from the signature Σ. For the sake of simplicity, consider a signature $\{\mathtt{bin}, \mathtt{nil}\}$ with only two symbols, the first one having arity 2 and the second one having arity 0. Terms from this signature are just unlabelled, binary trees.

- Formally define BC_Σ.
- Prove that, as in the case of words, that for every function f in BC_Σ, there is a polynomial p_f such that $|f(\vec{x})| \leq p(|\vec{x}|)$, or give a counterexample.
- What does the above tell us about the expressive power of the obtained algebra?

Esercise 3. Prove that any functional program which admits an additive polynomial interpretation computes a polytime function.

Esercise 4. Play with the Tyrolean Complexity Tool (see `http://cl-informatik.uibk.ac.at/software/tct/`). Find the simplest example of a polytime program that cannot be recognised so by the tool.