A higher order perspective on computational analysis

Dag Normann
Department of mathematics, The university of Oslo

Abstract
This talk will be based on joint work with Sam Sanders (since 2015), and will focus on the computability-theoretical aspects of our project.

In traditional computational analysis the objects under investigation will be coded in ways that in principle make them accessible to a digital computer, normally as sets of integers or infinite sequences of integers. The reasons for doing it this way are self evident, and will not be questioned.

However, the objects coded will be set-theoretically more complex, like sets of reals or functions defined on some Euclidian space. Then the existence of codes are mathematical (mostly easy, as such) theorems, but the ``construction" of a code for a particular object is often via a definition involving quantifiers over the continuum. We consider the question of how complex the construction of a code from an object is, to be of foundational interest, and we will use computability models involving higher order objects, mostly variations over Kleene's model from 1959, to answer such questions.

Existence of codes are just special examples of proofs of existence in general, like the Heine-Borel theorem and the Jordan decomposition theorem. It is also of foundational interest to analyse how complex it is to find the objects proved to exsist, and it is then important to have a framework where such problems can find exact answers.

In this talk we will use higher order representations of familiar objects from mathematical analysis, as suggested by Kohlenbach in his higher order reverse mathematics, and we will discuss the complexity of basic constructions in the framework of higher order computability theory. We will start with a discusssion of the Heine-Borel theorem as formulated by Borel in a paper from1895, and move on to other well known theorems.

We will give a brief description of Kleene's model from 1959 and our customized modiication of it, but will also mention other possible computability models.

We will isolate some ``new" functionals of type 3 that we name *structure functionals* and that can be used to calibrate the complexities of well known mathematical constructions, like e.g. Jordan decomposition. All these are genuinly type 3 in the sense that they can not be computed from functionals of type 2, and they are of a nature that had not been studied. Using examples, we will discusss the distinction between countably based functionals and those that are not, and we will see that some arguments, like writing a set A of reals, known to be finite, as $A = \{a_1 , \dots , a_n\}$ actually involves using a partial, computationally powerful, functional.

Towards the end we will discuss an open problem meant to answer the problem:

How hard is it to find the code of an open set of reals?