Theory and Practice of Higher-type Computation (Tutorial)

Martín Escardó

School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

In higher-type computation, established by Kleene and Kreisel in the late 1950's (independently), one works with the data types obtained from the discrete natural numbers by closing under finite products and function spaces. For the theory of higher-type programming languages, it is natural to work with a corresponding hierarchy, or type structure, of domains, identified by Ershov and Scott in the late 1960's (again independently). The Kleene–Kreisel and Ershov–Scott hierarchies account for total and partial computation respectively.

In this tutorial I'll explain the theory and practice of higher-type computation and programming languages, and develop old and new applications.

From a theoretical point of view, I'll present Kleene–Kreisel spaces and Ershov–Scott domains, and relate the two. Moreover, I'll discuss common generalizations, chiefly QCB spaces and equilogical spaces, which admit further useful closure properties, and their relationship to TTE (Schröder, Simpson. Scott, Bauer, Weihrauch and many others). I'll also present a natural higher-type model of computation/programming language, namely PCF (Platek, Scott, Plotkin).

From a practical point of view, I'll introduce a fragment of the language Haskell as a faithful implementation of PCF. Moreover, I'll develop and run several examples (and prove theorems about them), pertaining to (i) exhaustive search of infinite sets in finite time (in particular Ulrich Berger's algorithm and generalizations), and (ii) computation with real numbers (in particular Alex Simpson's integration algorithm and generalizations).

Andrej Bauer, Peter Hertling, Ker-I Ko (Eds.)
6th Int'l Conf. on Computability and Complexity in Analysis, 2009, pp. 21
http://drops.dagstuhl.de/opus/volltexte/2009/2254