Realizability and Higher-type Computability

John Longley

TUTORIAL

It is well known what is meant by a computable function on the natural numbers, but it is far from obvious what should be meant by a "computable" operation of higher type —say, of type $((N \to N) \to N) \to N$. Indeed, it is not even clear *a priori* whether there are any really sensible notions of higher type computability. Nevertheless, the study of computability at higher types turns out to be mathematically fruitful, and yields a small handful of different notions of computability that appear to be mathematically natural and robust. Besides its intrinsic conceptual interest, the subject has connections with:

- Constructive logic and metamathematics
- Programming language semantics and logic
- Complexity theory
- Real number computability
- Computability in physics

Since the 1950s, the subject has evolved in a somewhat chaotic fashion, different strands being scattered widely across the literature in recursion theory, constructive logic and computer science. However, it now seems possible, with hindsight, to give a coherent account of much of this material within a uniform framework.

In the first lecture I will give a brief historical survey of the subject and the contributions of Kleene, Kreisel, Platek, Scott and many others. I will then describe some recent approaches to imposing a sense of coherence on this landscape. I will explore, within a simple abstract framework, the various known notions of higher type computable *functional* and how they are related. I will then show how ideas from realizability offer a framework for discussing even more general (e.g. non-extensional) notions of "computable operation".

Throughout the course I will emphasize the close interplay between computability and realizability, and the potential applications to computer science.

Address: Division of Informatics, University of Edinburgh, JCMB, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK
E-mail: jrl@dcs.ed.ac.uk