

# Computing Over Generalisations of the Reals

Lorenzo Galeotti

In classical computability theory computations are thought of as *finite* and *discrete* processes carried out by idealised machines on a *finite* amount of data. Although these assumptions are quite natural, since the beginning of the research in this area, researchers have been developing theories in which these assumptions are weakened.

Particularly interesting are those notions of computability in which the finiteness of the process and of the data are relaxed. Prominent in this area are models of computability over the real line. These machines do indeed work on (sometimes representations of) real numbers and are in some cases allowed to run for infinitely many stages.

An extreme weakening of the finiteness restrictions on computability led to the notion of *transfinite computability*. The idea is that of taking classical models of computability and extend them to the transfinite. Particularly important in this context are the works of Hamkins and Lewis on *Infinite Time Turing Machines* (ITTMs) and of Koepke on *Ordinal Turing Machines* (OTMs).

It is natural to ask whether classical notions of computability over the reals can be generalised to the transfinite. The first problem that has to be solved in order to answer this question is that of finding generalisations of the reals which are suitable for computability. The main focus of this talk will be on the development of theories of computability over generalisations of the real line.

We will begin with a brief introduction to two classical models of computability over the reals. Then, we will present a generalisation of the reals suitable to do real analysis and computability. After this, we will focus on the use of transfinite models of computability in the context of generalisations of the reals. We will present two different approaches to the problem of defining a notion of computability over generalisations of the reals, one which makes use of OTMs and one which uses a generalisation of Blum-Shub-Smale Machines.