Behavioral Computation Theory 2005

Yuri Gurevich

Microsoft Research, One Microsoft Way
Redmond, WA 98052, USA

Classical computation theory was a major advance in understanding computability. It is based on the Church-Turing thesis and is known also as theory of algorithms, and recursive functions theory. Structural complexity theory refined classical computation theory with the analysis of necessary resources. It is based on the Turing computation model.

There are also theories of special purpose algorithms: sorting algorithms, linear programming, etc. But this is a separate story. Here we are interested in algorithms without any purpose restriction.

Classical computation theory, even together with modern structural complexity theory, suffers of certain limitations. It is largely a theory of computable functions, and there is in general more to an algorithm than the function it computes. Besides the classical theory is about computations from strings to strings which is too restrictive for e.g. database queries or software specifications.

Behavioral computation theory aims to ameliorate classical computation theory. It is based on characterization of algorithms by means of axioms and by means of abstract state machines. In the talk we survey the current state of behavioral computation theory.