

Calibrating the Complexity of Mathematical Proofs and Constructions

Richard Shore
Cornell University

Thursday, March 6, 2008
007 Kemeny Hall, 4:00 pm
(Tea 3:30 pm 300 Kemeny Hall)

Abstract

We will discuss two related measures of complexity for mathematical theorems and constructions. One asks what proof techniques (or formally axioms) are needed to prove specific theorems. The other asks (for existence proofs) how complicated (in the sense of computability) are the objects that are asserted to exist.

For this talk we will consider some illustrative examples from Combinatorics. In particular, we will consider several theorems of matching theory such as those of Frobenius, (M. and P.) Hall and König. While in the finite case these theorems seem both different and yet somehow the same, an analysis of the countable case in terms of computability or provability clearly distinguishes among them and assigns precise levels to their complexity.

At the most complicated level that we will consider lies the König Duality Theorem: Every bipartite graph has a matching such that one can choose a vertex from each edge of the matching so as to produce a cover, i.e. a set with an element from every edge. This theorem cannot be proven using algorithmic methods even when combined with compactness (König's lemma for binary trees) or full König's lemma. We will show that it requires highly nonelementary methods as typified by constructions by transfinite recursion, choice principles and, for some versions, even more.

If time permits, we may also mention the calibration of some results of Ramsey theory that lie at the other (low) end of our classification scheme: Ramsey's theorem for n -tuples for different n and some consequences such as the theorems of Dilworth and Erdős-Szekeres. (Every infinite partial order has an infinite chain or antichain and every infinite linear order has an infinite ascending or descending sequence.)

We will not use, or even consider, any formal systems and no knowledge of logic is presupposed. We will work instead with an intuitive notion of what it means for a function to be computable, i.e. there is a computer program that calculates it given time and space enough and no mechanical failures. We will also explain the relevant combinatorial notions.

This talk should be accessible to graduate students.

1202147706