Randomized Methods in Computational Complexity

Prof. Dr. Nitin Saxena

Sommersemester 2009
From Monday, 20th April 2009.
Lectures: Mon 11-13 & Fri 14-16. Exercises: Fri 16-18
At: Seminar room N327, Informatik V, Römerstraße 164

Background:
No prerequisite except a clear mathematical thinking. But students who are also aware of the basics of computation - Turing machines, complexity classes, P, NP, probabilistic methods - will find the course especially interesting.

Outline:
In this course we will study how randomness helps in designing algorithms and how randomness can be removed from algorithms.

We will start by formalizing computation in terms of algorithms and circuits. We will see an example of randomized algorithms, *identity testing*, and prove that eliminating randomness would require proving hardness results. We then prove a hardness result for the problem of *parity* using randomized methods. We construct certain graphs called *expanders* that are useful in reducing randomness in algorithms. These lead to a surprising logarithmic-space algorithm for checking connectivity in graphs. We show that if there is hardness in nature then randomness cannot exist! This we prove by developing *pseudo-random generators*. We show how to extract randomness from a weakly random source by using *extractors*. Finally, we show how to probabilistically check proofs (*PCP*) and prove the hardness of approximating some NP-hard problems.