Introduction to computational complexity (L16)

Professor W. T. Gowers

Computational complexity is the study of the resources needed to perform computational tasks. Two resources of obvious importance are the time needed, which corresponds to the number of steps an algorithm needs to take, and the amount of storage needed. A less obvious one is the "amount of randomness" needed by a randomized algorithm. Over the last 50 years, the subject has developed in many important directions, so in one course it is not possible to cover more than a small fraction of it, even at an introductory level. Thus, while this course has the same title as a course given last year, its content will be substantially different: last year the focus of the course was on completeness results and on lower bounds for circuit complexity, while this year there will be a strong emphasis on the role played in the theory by randomness and pseudorandomness.

The following list of topics is a rough guide to what will be covered (but some topics listed here may not end up in the course and others not listed may).

- Some of the basic complexity classes, including P, NP, PSPACE, AC, NC, P/poly, RP, and BPP, and relationships between them.
- Arithmetic complexity.
- The power of randomness.
- Derandomization.
- Pseudorandom generators.
- Interactive proof systems.

Prerequisites

Familiarity with the basics of (discrete) probability and graph theory will be assumed. Otherwise, there are few prerequisites.

Literature

1. Sanjeev Arora and Boaz Barak, *Computational complexity: a modern approach*, CUP, 2009. Also available in draft form: https://users.cs.duke.edu/~reif/courses/complectures/books/AB/ABbook.pdf
2. Oded Goldreich, *Computational complexity: a conceptual perspective* CUP, 2008. Also available online in draft form: https://www.wisdom.weizmann.ac.il/~oded/cc-drafts.html

Additional support

Three examples sheets will be provided and three associated examples classes will be given. There will be a one-hour revision class in the Easter Term.