MAT 598/494: Theory of NP-Completeness, Fall 2017

Instructor

Hal Kierstead.

Class

Day & Time: T & Th 3:00pm;
Place: ECG 237

Course Description

Need: There seems to be a sharp divide between students and even researchers who have proved an NP-completeness result (at least as an exercise) and those who have only a vague idea of what the concept even means. Those in the first category tend to treat the process as trivial, which hopelessly discourages those in the second. In my experience at ASU most undergraduates in math and computer science as well as most graduate students in mathematics fall into the second category. The course aims to be a friendly introduction for the second category.

Audience: This combined undergraduate and graduate course will have combined lectures, but lower standards/expectations for the undergraduates. For example, graduate students might be required to know a whole proof, while undergraduates might only be expected to answer questions about steps in the proof, and graduate students would be responsible for harder homework problems. The MAT 598 should be useful for graduate students in discrete and computational mathematics who find themselves in the second category discussed above. Moreover, computational complexity, and in particular NP-completeness, is a fundamental part of modern mathematics, and so should interest a broad range of students in theoretical mathematics. I intend to make the course accessible to a wide range of graduate students. Among undergraduates, MAT 494 should be interesting and accessible to students from the honors college, computer science, and mathematics.

Goals: Students should be able to consider a typical computational problem, translate it into a decision problem if relevant, and then attempt to decide whether it is polynomial or NP-complete (or something else); in routine cases they should expect success.

Topics:

- 1. Friendly introduction to problems, algorithms, complexity, Church's thesis, polynomial time, intractable problems, NP problems and NP-Completeness
- 2. Technical definitions of *decision problems, languages, encoding schemes, deterministic Turing machines, the class P, nondeterministic computation, the class NP, polynomial transformations and NP-completeness*
- 3. Cook's Theorem and its proof
- 4. Proving NP-completeness results:
    a) Some basic NP-complete problems
    b) Special techniques for proving NP-completeness: restriction, local replacement, gadgets and component design
    c) many exercises

- 5. Using NP-completeness to analyze problems:
  - a) Subproblems
  - b) Number problems and strong NP-completeness
- 6. NP-Hardness
- 7. Coping with NP-complete problems
  - a) Performance guarantees for approximation algorithms
  - b) Applying NP-completeness to approximation algorithms
  - c) Behavior “in practice”