CS 2429 – Winter 2018
Location: BA 4010
Time: 3-5

Instructor: Toniann Pitassi
Office: 2305A Sandford Fleming Building
email: toni@cs.toronto.edu
Office hours: by appointment

Course Web Site: http://www.cs.toronto.edu/ toni/Courses/ProofComp2017/CS2429.html
Refer to this site periodically for important announcements and other information. All handouts will be available on the site in postscript form.

Course Materials: There is no book for this course. Each lecture will have supplemental reading material such as a paper or lecture notes, available on the website.

Course Description

This is a topics course which will cover a new and growing body of work in algorithms and complexity theory that focuses on algebraic and semi-algebraic proof systems. Sherali Adams (SA) is the basic proof system that underlies linear programming, and the Sum-of-Squares (SOS) proof system underlies semidefinite programming. Similarly their algebraic counterparts (which manipulate low degree polynomial equalities rather than inequalities) are the Nullstellensatz proof system, and the Polynomial Calculus (PC), which are closely connected to the Grobner basis algorithm for studying systems of polynomial equations.

We will cover the importance of the proof systems, both from the point of view of leading to a flurry of new algorithms, and conversely, as a tool for proving limitations of large families of algorithms for solving/approximating optimization problems. The style of the course will vary depending on the material. For the lower bounds, I will mostly lecture. For the algorithms/upper bounds, we will follow Sam Hopkin's multipart blog, and then read a couple of other papers, with someone leading the discussion.

New Algorithms via SOS. A proof system is automatizable if it is possible to find short proofs whenever they exist. We will show that all of the above algebraic and semi-algebraic proof systems are automatizable in the sense that proofs of degree d can be found in time $n^{O(d)}$. We will then explore how the important property of automatizability of these systems has led to a recent flurry of new algorithms for some well-studied problems in optimization and machine learning. We will start with the famous GW approximation algorithm for max-cut using SOS. We will then discuss SOS based algorithms for ML problems (such as dictionary learning, tensor decomposition, clustering, and community detection.)

Limitations via Proof Systems. On the flip side, lower bounds and integrality gaps for these proof systems rules out a large class of linear and semidefinite programming approaches for solving and approximating many problems. We will start by proving lower bounds for these systems and explain their connection to extended formulations in Linear and Semi-Definite Programming. We will cover several deep connections between such lower bounds and various models of computation and communication. These connections, obtained via "lifting theorems", allow one to pass from lower bounds for the basic systems mentioned above to significantly stronger versions of them. In particular, we will cover recent lower bounds for extended formulations for approximating CSP problems, via a lifting theorem from extended formulations to SA.

A more detailed list of topics (not necessarily covered in this order) is as follows.

(1.) Introduction to Proof Systems and Algorithmic Applications. Basic concepts and definitions, motivation, connections to complexity theory and algorithm design. Overview, Motivation and Highlights. Important concepts: automatizability p-simulation.

(2.) Algebraic and Semi-algebraic proof systems: Resolution, Nullstellensatz, Polynomial Calculus (PC), Sherali Adams (SA), Sum-of-Squares (SOS). Automatizability of these proof systems Connection between Sherali Adams and linear programming, and SOS and semi-definite programming. Duality–proofs versus pseudo-distributions.

(3.) Algorithms: Using SA and SOS to obtain new algorithms. We will follow Sam Hopkin's recent multipart blog to describe new algorithms for problems in machine learning via sum-of-squares, including: dictionary learning, tensor decomposition and learning mixtures of Gaussians.

(4.) Lower bounds for PC and SOS (includes as special cases lower bounds for Resolution and SA).

(5.) Extension Complexity and Linear Programming (LP), Semi-Definite Extension Complexity and Semi-Definite Programming (SDP). (We will formulate as proof systems and compare to other proof systems)

(7.) Lifting Theorems: Lower Bounds for stronger models of computation (Extension Complexity, as well as other models) from SA/SOS lower bounds.

(8.) On Size Automatizability for SA and SOS.

(9.) Time permitting: Cutting Planes lower bounds for random CNF, the IPS proof system and connections to algebraic circuit complexity.

Grading and Assignments

Grading will be based on a couple of assignments which will be handed out during the semester – probably 2. You will have at least one week to turn in each assignment. The work you submit must be your own or with at most one other person. Class attendence is manditory and you are encouraged to ask many questions in class. I may also ask you to either prepare scribe notes for a lecture, or to read a paper and present it. For the latter, you are welcome to work in pairs.