SoCal Theory Day 2014

Friday, October 17, 2014
University of California, San Diego
CSE Building (EBU3B), Room 1202

Program:

- 08:30 - 08:50 Coffee
- 08:50 - 09:00 Opening remarks
- 09:00 - 10:00 Advances in Obfuscation, Amit Sahai (UCLA)
- 10:00 - 11:00 Questions/Discussions/Break
- 11:00 - 12:00 Spectral graph algorithms for partitioning problems, Luca Trevisan (Stanford)
- 12:00 - 14:00 Lunch
- 14:00 - 15:00 Approaches to bounding the exponent of matrix multiplication, Chris Umans (Caltech)
- 15:00 - 16:00 Questions/Discussions/Break
- 16:00 - 17:00 Algorithms for circuits and circuits for algorithms: connecting the tractable and intractable, Ryan Williams (Stanford)

Abstracts

Advances in Obfuscation
Amit Sahai, University of California, Los Angeles

The goal of general-purpose program obfuscation is to make an arbitrary computer program unintelligible while preserving its functionality. Obfuscation allows us to achieve a powerful capability: software that can keep a secret. This talk will cover recent advances in obfuscation research, that have allowed for the first time constructions of general-purpose obfuscation mechanisms with security based on plausible intractability assumptions.

Spectral graph algorithms for partitioning problems
Luca Trevisan, Stanford University

Spectral graph theory studies applications of linear algebra to graph theory and to the design and analysis of graph algorithms. "Spectral" graph algorithms are algorithms that exploit properties of the eigenvalues and eigenvectors of matrices associated with a graph, such as the Laplacian matrix.

The Cheeger inequality is a classical result in spectral graph theory which states that the second Laplacian eigenvalue of a graph is small if and only if the graph has a sparse cut. The proof of the Cheeger inequality also gives a worst-case analysis of the "sweep" spectral partitioning algorithm of Fiedler as an approximation algorithm for the sparsest cut problem.

We discuss three generalizations of this result:

(i) the k-th Laplacian eigenvalue is small if and only if the vertices can be partitioned into k subsets, each defining a sparse cut

(ii) if the k-th Laplacian eigenvalue is large, then Fiedler's sweep algorithm performs better than the worst-case bounds implied by Cheeger's inequality

(iii) if the k-th Laplacian eigenvalue is small and the (k+1)-st is large, then the vertices can be partitioned into k subsets such that each subset defines a sparse cut and each subset induces an expanding subgraph.

Approaches to bounding the exponent of matrix multiplication
Chris Umans, California Institute of Technology

We begin by describing the ideas behind the state-of-the-art bounds on ω, the exponent of matrix multiplication. We then present the "group-theoretic" approach of Cohn and Umans as an alternative to these methods, and we generalize this approach from group algebras to general algebras. We identify adjacency algebras of coherent configurations as a promising family of algebras in the generalized framework. We prove a closure property involving symmetric powers of adjacency algebras, which enables us to prove nontrivial bounds on ω using commutative coherent configurations, and suggests that commutative coherent configurations may be sufficient to prove $\omega = 2$.

Along the way, we introduce a relaxation of the notion of tensor rank, called s-rank, and show that upper bounds on the s-rank of the matrix multiplication tensor imply upper bounds on the ordinary rank. In particular, if the "s-rank exponent of matrix multiplication" equals 2, then the (ordinary) exponent of matrix multiplication, ω, equals 2. Finally, we will mention connections between several conjectures implying $\omega = 2$, and variants of the classical sunflower conjecture of Erdos and Rado.

No special background is assumed. Based on joint works with Noga Alon, Henry Cohn, Bobby Kleinberg, Amir Shpilka, and Balazs Szegedy.

Algorithms for circuits and circuits for algorithms: connecting the tractable and intractable
Ryan Williams, Stanford University

This talk will survey two basic topics in algorithms and complexity theory, and interesting connections that have been developed between them:

"Algorithms for circuits" refers to the design of algorithms which can analyze logical circuits or Boolean functions as input, checking a simple property about the complexity of the underlying function. For instance, an algorithm determining if a given logical circuit C has an input that makes C output true would solve the NP-complete Circuit-SAT problem. Such an algorithm is unlikely to run in polynomial time, but could possibly be more efficient than exhaustively trying all possible inputs to the circuit.

"Circuits for algorithms" refers to the modeling of complex uniform algorithms with "simple" Boolean circuit families, or proving that such modeling is impossible. For example, can every exponential-time algorithm be simulated using Boolean circuit families of only polynomial size? It is widely conjectured that the answer is no, but the present mathematical tools available are still too crude to resolve this kind of separation problem.