Mathematisches Forschungsinstitut Oberwolfach

Report No. 26/2005

Complexity Theory

Organised by
Joachim von zur Gathen (Bonn)
Oded Goldreich (Rehovot)
Claus-Peter Schnorr (Frankfurt)
Madhu Sudan (Cambridge)

June 5th – June 11th, 2005

Abstract. Computational Complexity Theory is the mathematical study of resources like time, space, or randomness that are required to solve computational problems. The current workshop was focused on recent developments, and the interplay between randomness and computation played a central role in many of them.

Mathematics Subject Classification (2000): 68-06, 68Q01, 68Q10, 68Q15, 68Q17, 68Q25.

Introduction by the Organisers

The workshop *Complexity Theory* was organised by Joachim von zur Gathen (Bonn), Oded Goldreich (Rehovot), Claus-Peter Schnorr (Frankfurt), and Madhu Sudan (Cambridge). The workshop was held on June 5th–11th 2005, and attended by approximately 50 participants spanning a wide range of interests within the field of Computational Complexity. Sixteen talks were presented in the mornings, attended by all participants. In addition, extensive interaction took place in smaller groups. Specifically, several more specialized sessions were held in the afternoons and were typically attended by 5-20 participants, and numerous imformal meetings of 2-5 participants took place at various times (including at night).

The Oberwolfach Meeting on Complexity Theory is marked by a long tradition and a continuous transformation. Originally starting with a focus on Algebraic and Boolean Complexity, the meeting has continuously evolved to cover a wide variety of areas, most of which were not even in existence at the time of the first meeting (in 1972). The format of the meetings has also been drastically changed in the recent meetings so that the focus is on interactions in small specialized sessions, maintaining unity via general plenary sessions. While inviting many of the most prominent researchers in the field, the organizers try to identify and invite a fair number of promising young researchers. The current meeting marks the retirement from the organizing team of the last and youngest member of the founding team (Claus Schnorr).

Computational Complexity (a.k.a Complexity Theory) is a central field of Computer Science with a remarkable list of celebrated achievements as well as vibrant research activity. The field is concerned with the study of the *intrinsic complexity* of computational tasks, and this study tends to *aim at generality*: it focuses on natural computational resources, and considers the effect of limiting these resources on the class of problems that can be solved. Computational complexity is related to and has substantial interaction with other areas of mathematics such as number theory, algebra, combinatorics, coding theory, and optimization.

The workshop has focused on several sub-areas of complexity theory and its nature may be best examplified by a brief survey of some of the meeting's highlights.

The complexity of Undirected Connectivity. For more than two decades, undirected connectivity was one of the most appealing examples of the computational power of randomness. Whereas every graph (e.g., a planar graph representing a maze) can be efficiently traversed by a deterministic algorithm, the classical deterministic algorithms required an extensive use of (extra) memory (i.e., linear in the size of the graph). On the other hand, it was known that, with high probability, a random walk (of polynomial length) visits all vertices in the corresponding connected component. Thus, the randomized algorithm requires a minimal amount of auxiliary memory (i.e., logarithmic in the size of the graph). Even after more than a decade of focused attension at the issue, a significant gap remained between the space complexity of randomized and deterministic polynomial-time algorithms for this natural and ubiquitous problem. After deterministic polynomial-time primality testing was discovered in 2003, undirected connectivity became the most famous example where randomized computations seemed more powerful than deterministic ones.

In the workshop, Omer Reingold presented his recent breakthrough result asserting that any graph can be traversed by a deterministic polynomial-time algorithm that only uses a logarithmic amount of auxiliary memory. His algorithm is based on a novel approach that departs from previous attempts, where the latter tried to derandomize the random-walk algorithm. Instead, Reingold's algorithm traverses a virtual graph, which (being an "expander") is easy to traverse (in deterministic logarithmic-space), and maps the virtual traversal of the virtual graph to a real traversal of the actual input graph. The virtual graph is constructed in (logarithmically many) iterations, where in each iteration the graph becomes easier to traverse.

A new proof of the PCP Theorem. The PCP Theorem is one of the most influential and impressive results of complexity theory. Proven in the early 1990's, the theorem asserts that membership in any NP-set can be verified, with constant error probability (say 1%), by a verifier that probes a polynomially long (redundant) proof at only a constant number of randomly selected bit locations. The PCP Theorem led to a breakthrough in the study of the complexity of combinatorial approximation problems. Its original proof is very complex and involves the composition of two highly non-trivial proof systems, each minimizing a different parameter of the asserted PCP system (i.e., the proof length and the number of probed bits).

In the workshop, Irit Dinur presented an alternative approach to the proof of the PCP Theorem. Her recent breakthrough approach leads to a simpler proof of the PCP Theorem as well as to resolving an important open problem regarding PCP systems (namely, constructing a PCP system having proofs of almost-linear rather than polynomial length). Dinur's approach is based on gradually improving the performance of PCP-like systems, starting with a trivial system and performing (logarithmically) many amplification steps. In each step, the PCP-like system is composed with itself in a way that almost preserves all parameters while drastically improving one particular parameter.

Extracting randomness. Extracting almost-perfect randomness from weak sources of (imperfect) randomness is crucial for the actual use of randomized procedures. The latter are analyzed assuming they are given access to a perfect random source, while in reality one typically has access only to sources of weak randomness (e.g., having constant entropy rate). Indeed, the problem has attracted a lot of attention in the last couple of decades. In the 1990's and early 2000's, the focus was on single-source extractors that utilize a very short auxiliary random seed. After more than a decade of impressive progress, culminating in an almost optimal construction, the focus has shifted back to "seedless' extraction from a few independent weak sources. In the workshop, Avi Wigderson surveyed the progress made on the latter problem in the last couple of years, and the techniques used towards this end. His presentation was followed by a specialized session devoted to this subject.

Cryptography. Modern Cryptography is intimately related to Complexity Theory. A new aspect of this relationship was manifested in a talk by Yuval Ishai, which described a recent work by himself, Eyal Kushilevitz and their graduate student Benny Applebaum. They showed that, for many central cryptographic primitives, secure implementations that have moderate complexity (which exists under standard complexity assumptions) can be transformed into secure implementations that have very low (and in fact minimal) complexity (i.e., each output bit in these implementations can be computed in constant time). Additional works in the area of Cryptography were presented and discussed in a specialized session devoted to this area.

Holographic Reductions. Standard (many-to-one) reductions between computational problems utilize gadgets that enforce a correspondence between global solutions and a sequence of partial local solutions (within the gadgets). In the workshop Les Valiant presented a novel type of reductions, called holographic, in which individual global solutions are not a combination of corresponding local solutions, but rather the set of global solutions is a combination of the sets of local solutions. He presented holographic reductions between counting problems, noting that the corresponding gadgets cannot be implemented in the standard (non-holographic) manner. These reductions (to a problem that is solvable in polynomial-time) yield polynomial-time algorithms for problems that were not known to be efficiently solvable.

The complexity of Matrix Multiplication. Improved algorithms for matrix multiplication were the focus of extensive research in the 1970's and 1980's, culminating in a $n^{2.38}$-time algorithm for multiplying two n-by-n matrices. Much of the progress on this question has occurred at the various Oberwolfach meetings on Complexity Theory. In the workshop, Chris Umans presented a novel approach to the design of such algorithms. So far, this approach has not yielded an improved algorithm, however it yields significantly a simpler proof of the fact that matrix multiplication can be performed in $n^{2.41}$ steps. This is remarkable in light of the formidable complexity of previous proofs in the area. Additional works in the area of Algebraic Complexity were presented and discussed in a specialized session devoted to this area.

Additional topics that were discussed in the workshop include a geometric approach to combinatorial optimization problems (see Sanjeev Arora's extended abstract), the pursuit of even stronger PCP systems (see extended abstracts by Eli Ben-Sasson and Oded Regev), computational problems regarding integer lattices (see specialized session devoted to the topic), the complexity of approximation problems (see Julia Chuzhoy's extended abstract), computational problems in coding theory (see Eyal Kushilevitz's extended abstract), the relation between worst-case and average-case complexity (see extended abstracts by Adi Akavia and Amnon Ta-Shma), and Quantum Computing (see extended abstracts by Scott Aaronson and Ran Raz).

This report contains extended abstracts of the sixteen plenary talks as well as summaries of the specialized sessions, which were written by the organizers of these sessions. In addition, the report includes three extended abstracts of talks given in the specialized sessions (by Peter Buergisser, Ran Raz, and Ronen Shaltiel).

Workshop: Complexity Theory

Table of Contents

Abstracts

Undirected ST-Connectivity in Log-Space

Omer Reingold

We present a *deterministic*, log-space algorithm that solves st-connectivity in undirected graphs. The previous bound on the space complexity of undirected st-connectivity was $\log^{4/3}(\cdot)$ obtained by Armoni, Ta-Shma, Wigderson and Zhou [4]. As undirected st-connectivity is complete for the class of problems solvable by symmetric, non-deterministic, log-space computations (the class SL), this algorithm implies that SL = L (where L is the class of problems solvable by deterministic log-space computations). Independent of our work (and using different techniques), Trifonov [19] has presented an $O(\log n \log\log n)$-space, deterministic algorithm for undirected st-connectivity.

Our algorithm also implies a way to construct in log-space a *fixed* sequence of directions that guides a deterministic walk through all of the vertices of any connected graph. Specifically, we give log-space constructible universal-traversal sequences for graphs with restricted labelling and log-space constructible universal-exploration sequences for general graphs.

1. Introduction

We resolve the space complexity of undirected st-connectivity (denoted USTCON), up to a constant factor, by presenting a log-space (polynomial-time) algorithm for solving it. Given as input an undirected graph G and two vertices s and t, the USTCON problem is to decide whether or not the two vertices are connected by a path in G (our algorithm will also solve the corresponding search problem, of finding a path from s to t if such a path exists). This fundamental combinatorial problem has received a lot of attention in the last few decades and was studied in a large variety of computational models. It is a basic building block for more complex graph algorithms and is complete for the class SL of problems solvable by symmetric, non-deterministic, log-space computations (see [3] for a recent study of SL and quite a few of its complete problems).

The time complexity of USTCON is well understood as basic search algorithms, particularly breadth-first search (BFS) and depth-first search (DFS), are capable of solving USTCON in linear time. In fact, these algorithms apply to the more complex problem of st-connectivity in directed graphs (denoted STCON), which is complete for NL (non-deterministic log-space computations). Unfortunately, the space required to run these algorithms is linear as well. A much more space efficient algorithm is Savitch's [18], which solves STCON in space $\log^2(\cdot)$ (and super-polynomial time).

Major progress in understanding the space complexity of USTCON was made by Aleliunas, Karp, Lipton, Lovász, and Rackoff [2], who gave a *randomized* log-space algorithm for the problem. Specifically, they showed that a random walk (a path that selects a uniform edge at each step) starting from an arbitrary vertex of any connected undirected graph will visit all the vertices of the graph in polynomial number of steps. Therefore, the algorithm can perform a random walk starting from s and verify that it reaches t within the specified polynomial number of steps. Essentially all that the algorithm needs to remember is the name of the current vertex and a counter for the number of steps already taken. With this result we get the following view of space complexity classes: $\mathrm{L} \subseteq \mathrm{SL} \subseteq \mathrm{RL} \subseteq \mathrm{NL} \subseteq \mathrm{L}^2$ (where RL is the class of problems that can be decided by randomized log-space algorithms with one-sided error and L^c is the class of problems that can be decided deterministically in space $\log^c(\cdot)$).

The existence of a randomized log-space algorithm for USTCON puts this problem in the context of derandomization. Can this randomized algorithm be derandomized without substantial increase in space? Furthermore, the study of the space complexity of USTCON has gained additional motivation as an important test case for understanding the tradeoff between two central resources of computations, namely between memory space and randomness. Particularly, a natural goal on the way to proving $\mathrm{RL} = \mathrm{L}$ is to prove that USTCON $\in \mathrm{L}$, as USTCON is undoubtedly one of the most interesting problems in RL.

Following [2], most of the progress on the space complexity of USTCON indeed relied on the tools of derandomization. In particular, this line of work greatly benefited from the development of pseudorandom generators that fool space-bounded algorithms [1, 5, 10, 7] and it progressed concurrently with the study of the L vs. RL problem. Another very influential notion, introduced by Stephen Cook in the late 70's, is that of a universal-traversal sequence. Loosely, this is a fixed sequence of directions that guides a *deterministic* walk through all of the vertices of any connected graph of the appropriate size (see further discussion below).

While Nisan's space-bounded generator [10], did not directly imply a more space efficient USTCON algorithm it did imply quasi-polynomially-long, universal-traversal sequences, constructible in space $\log^2(\cdot)$. These were extremely instrumental in the work of Nisan, Szemeredi and Wigderson [11] who showed that USTCON $\in \mathrm{L}^{3/2}$ – The first improvement over Savitch's algorithm in terms of space (limited of course to the case of undirected graphs). Using different methods, but still heavily relying on [10], Saks and Zhou [17] showed that *every* RL *problem* is also in $\mathrm{L}^{3/2}$ (their result in fact generalizes to randomized algorithms with two-sided error). Relying on the techniques of both [11] and [17], Armoni, et. al. [4] showed that USTCON $\in \mathrm{L}^{4/3}$. Their USTCON algorithm was the most space-efficient one previous to this work. We note that the most space-efficient *polynomial-time* algorithm for USTCON previously known was Nisan's [10], which still required space $\log^2(\cdot)$. Independent of our work (and using different techniques), Trifonov [19] has presented an $O(\log n \log\log n)$-space, deterministic algorithm for USTCON.

2. Main Idea at a Glance

In retrospect, the essence of our algorithm is very natural: If you want to solve a connectivity problem on your input graph, first *improve its connectivity.*

In other words, transform your input graph (or rather, each one of its connected components), into a expander. We will also insist on the final graph being constant degree. Once the connected component of s is a constant-degree expander, then it is trivial to decide if s and t are connected: Since expander graphs have logarithmic diameter, it is enough to enumerate all logarithmically long paths starting with s and to see if one of these paths visits t. Since the degree is constant, the number of such paths is polynomial and they can easily be enumerated in log space. Our transformation of an arbitrary graph into an expander rely on techniques developed by Reingold, Vadhan and Wigderson [16] in the context of combinatorial constructions of constant degree expanders.

References

[1] Miklós Ajtai, János Komlós, and E. Szemerédi. Deterministic simulation in LOGSPACE. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, New York City, 25–27 May 1987.

[2] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, 29–31 October 1979. IEEE.

[3] Carme Alvarez and Raymond Greenlaw. A compendium of problems complete for symmetric logarithmic space. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(039), 1996.

[4] Roy Armoni, Amnon Ta-Shma, Avi Wigderson, and Shiyu Zhou. An $o(log(n)^{4/3})$ space algorithm for (s,t) connectivity in undirected graphs. *Journal of the ACM*, 47(2):294–311, 2000.

[5] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, pages 204–232, 15–17 May 1989.

[6] Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *RANDOM*, pages 209–223, 2002.

[7] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

[8] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

[9] Michal Koucky. Universal traversal sequences with backtracking. In *IEEE Conference on Computational Complexity*, pages 21–27, 2001.

[10] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[11] Noam Nisan, Endre Szemeredi, and Avi Wigderson. Undirected connectivity in $o(log^{1.5}n)$ space. In *Proceedings of the 30th FOCS*, pages 24–29, Research Triangle Park, North Carolina, 30 October–1 November 1989. IEEE.

[12] Noam Nisan and Amnon Ta-Shma. Symmetric logspace is closed under complement. *Chicago J. Theor. Comput. Sci.*, 1995.

[13] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[14] Ran Raz and Omer Reingold. On recycling the randomness of the states in space bounded computation. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, Atlanta, GA, May 1999.

[15] Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks in biregular graphs and the RL vs. L problem. *Electronic Colloquium on Computational Complexity* Technical Report TR05-022, 2005.

[16] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1), January 2001. Extended abstract in Proc. of *FOCS '00*.

[17] Michael Saks and Shiyu Zhou. $\mathrm{bp_h space}(S) \subseteq \mathrm{dspace}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999. 36th IEEE Symposium on the Foundations of Computer Science (Milwaukee, WI, 1995).

[18] J. Savitch. Relationship between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[19] Vladimir Trifonov. An o(log n log log n) space algorithm for undirected s,t-connectivity. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC 2005)*, 2005.

On Khot's Unique Games Conjecture

Oded Regev

In 2002, Khot [15] presented a conjecture known as the unique games conjecture. We survey recent progress including applications of this conjecture and attempts to prove (or disprove) it.

We first describe some of the known NP-hardness results. Many of the known results are tight. For example,

- MAX3SAT: a random assignment satisfies 0.875 of the clauses. [14] has shown a tight hardness of $0.875 + \varepsilon$ for any constant $\varepsilon > 0$.
- E3LIN2: a random assignment satisfies 0.5. [14] has shown a tight hardness of $0.5 + \varepsilon$ for any constant $\varepsilon > 0$.
- MaxClique: [13] has shown hardness of $n^{1-\varepsilon}$ for any constant $\varepsilon > 0$. This is essentially tight (a trivial algorithm gives n).
- SetCover: Hardness result of $\ln n$ [8] matching the greedy algorithm.

On the other hand, there are many problems for which the known NP-hardness results are very far from the best known algorithms. For example,

- VertexCover: A simple algorithm gives an approximation of 2. The best NP-hardness result is 1.36 [7].
- Coloring 3-colorable graphs: The best algorithm colors in $n^{3/14}$ colors [4]. The best known hardness result shows that it is NP-hard to color with 5 colors [17, 10].
- SparsestCut: Best algorithm approximates within $(\log n)^{0.5}$ [2]. No known NP-hardness results are known.
- MaxCut: Best algorithm approximates to within 0.878 [11]. Best known NP-hardness result is 0.941 [14].

For all these problems, the unique games conjecture implies a stronger, and often tight, hardness result:

- VertexCover: Unique-game-hardness of 1.999. [18].[1]
- Coloring 3-colorable graphs: Unique-game-hardness for any constant [6].
- SparsestCut: Unique-game-hardness within any constant (and beyond) [5].
- MaxCut: Unique-game-hardness to within 0.878 [16].

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[2] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proc. 36th ACM Symp. on Theory of Computing*, pages 222–231, 2004.

[3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[4] Avrim Blum and David Karger. An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs. *Inform. Process. Lett.*, 61(1):49–53, 1997.

[1] We thank Scott Aaronson for suggesting the name 'unique-game-hardness' during the Oberwolfach talk.

[5] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating sparsest cut and multicut. In *Proc. of 20th IEEE Annual Conference on Computational Complexity (CCC)*, 2005.

[6] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring, 2005. Submitted.

[7] Irit Dinur and Muli Safra. On the importance of being biased. *Annals of Mathematics*, 2004. To appear. Conference version appeared in STOC 2002.

[8] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[9] Uriel Feige and Daniel Reichman. On systems of linear equations with two variables per equation. In *APPROX*, 2004.

[10] Venkatesan Guruswami and Sanjeev Khanna. On the hardness of 4-coloring a 3-colorable graph. In *15th Annual IEEE Conference on Computational Complexity (Florence, 2000)*, pages 188–197. IEEE Computer Soc., Los Alamitos, CA, 2000.

[11] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *JACM*, 42:1115–1145, 1995.

[12] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[13] Johan Håstad. Clique is hard to approximate within n to the power $1-\varepsilon$. *Acta Mathematica*, 182(1):105–142, 1999.

[14] Johan Håstad. Some optimal inapproximability results. *Journal of ACM*, 48:798–859, 2001.

[15] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775. ACM Press, 2002.

[16] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? In *Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS 2004), ROME, Italy*, pages 146–154. IEEE Computer society, 2004.

[17] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000.

[18] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proc. of 18th IEEE Annual Conference on Computational Complexity (CCC)*, pages 379–386, 2003.

[19] Subhash Khot and Nisheeth Vishnoi. On embeddability of negative type metrics into l_1, 2005. Manuscript.

[20] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. In preperation, 2005.

[21] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.

Extracting Randomness from Few Independent Sources

Avi Wigderson

This abstract surveys recent progress on the problem of extracting almost perfect randomness from a few independent sources of defected randomness. We refer to defected sources that have a constant min-entropy rate, where a distribution X over binary strings of length n has min-entropy k if every string has probability at most 2^{-k} in X. We seek to use sources that have constant min-entropy rate (i.e., min-entropy $\Omega(n)$) in order to obtain an almost perfect virtual source of bits, by using a suitable randomness extractor. A randomness extractor for t independent sources of min-entropy k with error ϵ is a function $\mathtt{ext} : (\{0,1\}^n)^t \to \{0,1\}^m$ such that for every t independent sources, $X_1, ..., X_t$, if each X_i has min-entropy at least k then the distribution $\texttt{ext}(X_1, ..., X_t)$ is ϵ-close to the uniform distribution over m-bit strings. Our goal is to obtain such *explicit* constructions; that is, the function `ext` needs to be polynomial-time computable.

The motivation to this problem is evident given the prevalent role of randomness computer science especially in the design of algorithms, distributed systems, and cryptography. The justification for the use of randomness in computation is that randomness seems to exist in nature, and thus it is possible to sample natural phenomena (such as tossing coins) in order to make random choices in computation. However, there is a discrepancy between the type of random input that we expect when designing randomized algorithms and protocols, and the type of random data that can be found in nature. While randomized algorithms and protocols expect a stream of independent uniformly distributed random bits, this is too much to hope for from samples of natural phenomena. Indeed, the aforementioned min-entropy sources are intended to provide a general and flexible model of the type of samples one may hope to obtain in reality.

Unfortunately, randomness extraction (as defined above) is impossible from a *single* source (i.e., $t = 1$), even if the source has min-entropy $n-1$. Previous works have dealt with this problem in two ways: The first way is to add a short truly random *seed* as a secondary input to the extractor (see survey article [10]). In algorithmic applications, the random seed may be replaced by a deterministic scanning of all possibilities, applying the extractor (on the single source sample) with each possible seed, running the algorithm using each resulting string, and using the median value of the algorithm's output. This strategy is typically impossible in distributed and cryptographic applications, and thus a different approach is called for. The second approach is to use no seed, but make further assumptions on the structure of the weak sources (in addition to the minimal assumption of it containing sufficient min-entropy). Indeed, allowing few *independent* sources may be viewed as a special case of the second approach, and this motivates the construction of multiple-source extractors. Needless to say, we wish to use a small number of sources. Specifically, we want the number of sources to be a constant (independent of the sample length, n), and preferably use only two sources.

We note that the construction of a 2-source extractor is a generalization of a bipartite Ramsey graph. A bipartite graph with N vertices on each side is called k-Ramsey if, for every choice of 2^k vertices on each side, the induced subgraph contains some edges and misses some other edges. Indeed a 2-source extractor for sources of length n and min-entropy k, yields a bipartite k-Ramsey graph with 2^n vertices on each side.

It is easy to show that 2-source extractors exists for min-entropy $O(1)+\log_2(n/\epsilon^2)$, but explicit constructions were previously known only in case the min-entropies of both independent sources sum-up to more than n (cf. [5], following [11]). Explicit t-extractors for *min-entropy rate below half* were not know for any constant t. Here we report of recent results that break this barrier; that is, we discuss explicit constrictions of $O(1)$-source extractors for *any constant min-entropy rate*. We mention few of these results:

- *Multiple-source extraction for any entropy rate* [1]. For every $\delta > 0$, there exists an explicit poly$(1/\delta)$-source extractor for sources of min-entropy rate δ. The extractor's output (i.e., m) has length n and its error (i.e., ϵ) is exponentially small.
- *Three-source extraction for any entropy rate* [2]. For every $\delta > 0$, there exists an explicit 3-source extractor for sources of min-entropy rate δ. The extractor's output has length slightly greater than any constant and its error is slightly smaller than any positive constant.
- *Two-source extraction for entropy rate 0.499* [3]. For some constant $\delta < 1/2$, there exists an explicit 2-source extractor for sources of min-entropy rate δ.
- *Extraction in an asymmetric setting* [8]. Many results that hold for extraction using a single weak source (even with logarithmic min-entropy) and a perfectly random short seed, extend to the case that the seed has min-entropy rate ρ, for any constant $\rho > 1/2$.

These works build on results from additive number theory, which are briefly reviewed next.

Let A be a subset of some field $\mathcal{F}$ (or even a ring), and define $A + A \stackrel{\text{def}}{=} \{a + b : a, b \in A\}$ and $A \cdot A \stackrel{\text{def}}{=} \{a \cdot b : a, b \in A\}$. Note that $|A| \leq |A + A| \leq |A|^2$ (and similarly $|A| \leq |A \cdot A| \leq |A|^2$). An example for a set A where $A + A$ is small (of size about $2|A|$) is an *arithmetic progression*. An example for a set A where $A \cdot A$ is small is a *geometric progression*. The Erdős-Szemerédi Theorem asserts that for every finite set of integers A either $A + A$ or $A \cdot A$ is of size at least $|A|^{1+\epsilon_0}$, for some universal constant ϵ_0. In some sense, one can view this theorem as saying that a set of integers can't be simultaneously close to both an arithmetic progression and a geometric progression.

A natural question is whether this theorem also holds in *finite* fields. A first observation is that this theorem is *false* in a field $\mathcal{F}$ that contains a non-trivial subfield $\mathcal{F}'$. This because if we let $A = \mathcal{F}'$ then $A + A = A \cdot A = A$. However, Bourgain, Katz and Tao [4] proved that a variant of the Erdős-Szemerédi Theorem does hold in a finite field with no non-trivial subfields. In particular it holds in the fields GF(p) and GF(2^p) for every prime p. That is, they proved a corresponding lower-bound holds provided that A is neither too small nor too big (i.e., $|A| \in (|\mathcal{F}|^{\delta}, |\mathcal{F}|^{1-\delta})$, for some universal constant $\delta > 0$). Konyagin [7] gave a stronger result for *prime* fields, and showed that, as long as $|A| < |\mathcal{F}|^{0.99}$, the lower-bound holds (even if $|A|$ is very small).

The foregoing suggests that the function $f_3(x, y, z) = x \cdot y + z$ may be a good 3-source extractor. For starters, for X, Y and Z that are uniformly and independently distributed on A, Konyagin's result implies that $f_3(X, Y, Z)$ has either a very large support or a significantly larger support than X. Thus, starting with $3^{\log 1/\delta}$ copies of X, which has min-entropy rate δ, and combining these copies via a ternary-tree construction using f_3, we obtain a random variable with support size $|\mathcal{F}|^{0.99}$. Two extensions are required in order to obtain the desired extractor. Firstly, we need to deal with different sources rather than with identical sources (or copies of the same random variable). More importantly, we need to obtain bounds on the min-entropy of the resulting distribution, and not merely on the size of its support.

We note that a straightforward statistical analogs of the foregoing set size results do *not* hold. For example, consider random variables X and Y that are uniformly distributed on A and G respectively, where A (resp., G) is an arithmetic (resp., geometric) progression of size 2^k. Then, X and Y have each min-entropy k, but both $X+Y$ and $X \cdot Y$ assign $1/4$ of their probability weight to A and G, respectively, and so their min-entropy is at most $k+2$. Fortunately, it can be shown that for any independent sources X, Y and Z of min-entropy $k < 0.9 \log |\mathcal{F}|$, the distribution $f_3(X, Y, Z)$ has min-entropy $(1+\epsilon) \cdot k$, where $\epsilon > 0$ is a universal constant. In fact, this is the main technical result of [1], and its proof utilizes the result of Konyagin [7] along with some other additive number-theoretic results of Rusza [9] and Gowers [6].

Using the aforementioned result, we observe that the recursive tree construction using f_3 allows us to obtain a random variable over $\mathcal{F}$ having min-entropy $0.9 \log |\mathcal{F}|$, using $\mathrm{poly}(1/\delta)$ independent sources of min-entropy rate δ. To obtain an extractor, we repeat the construction twice, using different sources, and combine the results using an explicit 2-source extractor for high min-entropy (cf. [5, 12]).[1]

References

[1] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting Randomness from Few Independent Sources. In *Proc. 45th FOCS*, 2004.

[2] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating Independence: New Constructions of Condensers, Ramsey Graphs, Dispersers, and Extractors. In *Proc. 37th STOC*, 2005.

[3] J. Bourgain. More on the Sum-Product Phenomenon in Prime Fields and its Applications. Unpublished manuscript, 2005.

[4] J. Bourgain, N. Katz, and T. Tao. A sum-product estimate in finite fields, and applications. Arxiv technical report, `http://arxiv.org/abs/math.CO/0301343`, 2003. To appear in GAFA.

[5] B. Chor and O. Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM J. Comput.*, Vol. 17, 1988. Preliminary version in FOCS'85.

[6] W. T. Gowers. A new proof of Szemerédi's theorem for arithmetic progressions of length four. *Geom. Funct. Anal.*, 8(3):529–551, 1998.

[7] S. Konyagin. A sum-product estimate in fields of prime order. Arxiv technical report, `http://arxiv.org/abs/math.NT/0304217`, 2003.

[8] R. Raz. Extractors with Weak Random Seeds. In *Proc. 37th STOC*, 2005.

[9] I. Z. Ruzsa. Sums of finite sets. In *Number theory (New York, 1991–1995)*, pages 281–293. Springer, New York, 1996.

[10] R. Shaltiel. Recent developments in extractors. *Bulletin of the European Association for Theoretical Computer Science*, 2002.

[11] U. Vazirani. Strong Communication Complexity or Generating Quasi-Random Sequences from Two Communicating Semi-Random Sources. *Combinatorica*, Vol. 7, 1987. Preliminary version in STOC'85.

[12] U. Vazirani. Efficiency Considerations in Using Semi-random Sources. *Proc. STOC*, 1987.

[13] D. Zuckerman. General weak random sources. In *Proc. 31st FOCS*, pages 534–543, 1990.

[1] The $\mathrm{poly}(1/\delta)$-source extractor was proposed before by Zuckerman [13], but his analysis relies on an unproven number theoretic conjecture. A 2-source extractor for any constant min-entropy rate, follows from a seemingly weaker number theoretic conjecture (cf. [5, Cor. 11]).

The PCP Theorem via gap amplification

Irit Dinur

Background. The PCP Theorem characterizes the class NP as the set of languages for which membership can be proven with a robust, or '**P**robabilistically **C**heckable', **P**roof. That is, a verifier can verify correctness of such a proof, by tossing $O(\log n)$ random coins and reading only a *constant* number of proof symbols. Equivalently formulated, the PCP theorem asserts the existence of a polynomial-time reduction from SAT to gap-CSP (gap constraint satisfaction) where each constraint is (say) over two variables. This means that every satisfiable formula is transformed into a system of constraints that is totally satisfiable, and every unsatisfiable formula is transformed into a constraint system that only $1-\alpha$ fraction of which can be satisfied, for some $\alpha > 0$. This interpretation of the PCP theorem was discovered by [12, 1], and together with the proof of the PCP Theorem by [2, 1], brought about a revolution of the field of inapproximability. The proof of the theorem followed an exciting sequence of developments in interactive proofs, [15, 3, 7, 14, 19, 23, 4, 5, 12] to list just a few. The proof techniques were mainly algebraic including low-degree extension, low-degree test, parallelization through curves, a sum-check protocol, and the Hadamard and quadratic functions encodings.

Our approach. In this work we take a different approach for proving the PCP Theorem, which is perhaps natural in the context of inapproximability. For a given system of constraints $\mathcal{C}$, we consider the *satisfiability gap* of the system, denoted $\overline{\text{SAT}}(\mathcal{C})$, which is the smallest fraction of constraints that every assignment must leave unsatisfied. The outline of our proof is as follows. We start with a constraint system $\mathcal{C}$, for which it is NP-hard to decide if $\mathcal{C}$ is satisfiable or not. Namely, it is NP-hard to distinguish between the cases (i) $\overline{\text{SAT}}(\mathcal{C}) = 0$ and (ii) $\overline{\text{SAT}}(\mathcal{C}) \geq 1/n$. Such a statement is immediate from the NP-completeness of, say, 3SAT. Now repeatedly apply an amplification step to $\mathcal{C}$, doubling the satisfiability gap at each iteration (but so that if it was zero it remains zero). We will elaborate on this step further below. The final outcome $\mathcal{C}'$ is a constraint system for which in the first case still $\overline{\text{SAT}}(\mathcal{C}') = 0$, and in the second case $\overline{\text{SAT}}(\mathcal{C}') \geq \alpha$ for some $\alpha > 0$. The amplification step will only incur a linear blowup in the size of $\mathcal{C}$ so it is possible to apply it $\log n$ times, with the size of the final output still polynomially related to the size of the original input. This gives a reduction from 3SAT to gap-3SAT, thus proving the PCP Theorem.

Let us describe the amplification step in some more details. Our inductive step consists of three operations on constraint systems: (1) Preprocessing, (2) Graph powering, and (3) Alphabet reduction.

The most important step is the middle (graph powering) step which is the one that doubles the satisfiability gap. In order to describe this step let us focus on systems of constraints over two variables. Such systems can naturally be described as *constraint graphs*, whose vertices are variables that take values from some finite alphabet Σ, and whose edges are associated with constraints. So each edge carries a list of pairs of Σ-values that are 'allowed' for the endpoints of that edge. We note in passing that it is clearly NP-hard to decide if a given constraint graph is completely satisfiable or not, e.g., by reduction from 3-colorability (the alphabet Σ is the set of three colors, and the edges carry inequality constraints).

In order to amplify the gap of a constraint graph we simply raise it to the power t, for some $t = O(1)$. The *graph powering* operation is defined as follows: The new underlying graph is the t-th power of the original graph (with the same vertex-set, and an edge for each length-t path, and we allow parallel edges). Each vertex v will hold a value over a larger alphabet, that is a vector of d^t values from Σ. This vector is interpreted as v's "opinion" about the values of all of its neighbors at distance $\leq t$, including itself. The constraint over two adjacent vertices u, v in the new graph will be satisfied iff the values and opinions of u and v are consistent with an assignment that satisfies all of the constraints induced by u, v and their neighborhoods.

Our main lemma asserts that the satisfiability gap of G^t is at least that of G multiplied by a factor of roughly $\sqrt{t}$. This is true as long as the initial underlying graph is sufficiently "well-structured". By this we mean that the graph is d-regular for a constant d, has self-loops, and is an expander. All of these properties are easily obtained in the preprocessing stage.

The main advantage of this operation is that it *does not increase* the number of variables in each constraint (which stays 2 throughout). Moreover, when applied to d-regular graphs for $d = O(1)$, it only incurs a *linear* blowup in the graph size (the number of edges is multiplied by d^{t-1}), and an affordable increase in the alphabet size (which goes from Σ to Σ^{d^t}). Combined with an operation that reduces the alphabet back to Σ, we get an inductive step that can be repeated $\log n$ times until a constant gap is attained.

Gap amplification lemma. Let us give a high-level description of why the gap of G^t is larger than that of G. The intuitive reason is that each vertex in G^t has access to more information, seeing a vector of d^t values instead of just one. Also, it is compared with vertices "further away", so there is a higher chance to detect the inconsistency inherent in the graph G (which is measured by the satisfiability gap).

The idea of the proof is to fix some "best" assignment $A : V \to \Sigma^{d^t}$, which falsifies the smallest fraction of constraints in G^t. We then extract from it an assignment $a : V \to \Sigma$, according to popular opinion (under A).

We then relate the fraction of G-constraints that violate a to the fraction of G^t-constraints that violate A. Recall that G^t had a constraint for every length-t path, so we are counting how many bad paths there are, given that there is a certain fraction of bad edges. Already it should seem reasonable that if the density of bad edges is α, then the probability that a length-t path in a graph *that is an expander* would see a bad edge is on the order of $t\alpha$. The proof is more subtle than that because having a path pass through a bad edge, does not yet mean that the constraint on that path is falsified under A. However, we prove that a constant fraction of the paths that pass through a fixed bad edge in their middle portion (i.e., the edge is the i-th step in the path, for $t/2-\sqrt{t} \le i \le t/2+\sqrt{t}$) reject under A. Here we exploit the connection of A to the popular-vote assignment a.

The full inductive step. The inductive step can be illustrated as

$$G_{i+1} = (prep(G_i))^t \circ \mathcal{P}$$

where $prep(G)$ denotes a relatively simple transformation of any constraint-graph G into a constant degree regular expander graph with similar satisfiability gap. The operation $G \circ \mathcal{P}$ denotes composition with a constant-size "PCP" algorithm $\mathcal{P}$, which is an algorithm that inputs a constraint over a large alphabet, and outputs a system of constraints over a small alphabet. We run $\mathcal{P}$ on each constraint in our constraint graph, and take the union of the outputs to be the new constraint system $G \circ \mathcal{P}$. It is not hard to show that this yields alphabet reduction, without harming the satisfiability gap. The point is that since in our setting the input to $\mathcal{P}$ always has *constant size*, $\mathcal{P}$ is allowed to be extremely inefficient. This relaxation makes $\mathcal{P}$ not too difficult to construct, and one can choose their favorite implementation, be it Long-code based or Hadamard-code based. In fact, $\mathcal{P}$ can be found by exhaustive search, provided we have proven its existence in an independent fashion.

Short PCPs and Locally Testable Codes. Constructing extremely short Probabilistically Checkable Proofs and Locally-Testable Codes (LTCs) has been the focus of several works [5, 20, 18, 17, 10, 6, 9]. The shortest PCPs/LTCs are due to [6] and [9], each best in a different parameter setting. We show how to use the gap-amplification lemma to prove that $SAT \in PCP_{\frac{1}{2},1}[\log_2(n \cdot \text{poly}\log n),\, O(1)]$. This construction uses the PCP of [9] as starting point.

Final Remarks. This work follows [16, 11] in the attempt to find an alternative proof for the PCP Theorem that is combinatorial and/or simpler.

The construction described herein is inspired by Reingold's breakthrough proof for $SL = L$ [22]. Reingold shows how one iteration of powering / zigzagging, increases the spectral gap of any graph; so after $\log n$ iterations the initial graph becomes an expander. In our proof, the same form of amplification occurs for the satisfiability gap of a constraint graph. The steady increase of the satisfiability gap is inherently different from the original proof of the PCP Theorem. There, a constant satisfiability gap (using our terminology) is generated by one powerful transformation, and then a host of additional transformations are incorporated into the final result to take care of other parameters.

It is interesting to contrast our amplification and the amplification that occurs in Raz's parallel repetition theorem [21]. In some weak sense, our amplification can be viewed as a derandomized parallel repetition, but there are several differences between the two approaches. Parallel repetition takes a constraint system of size n to a new one whose size is n^t. Our amplification step takes a system of size n into a system of size $n \cdot const(t)$. Indeed, this is the largest blowup we can tolerate if we want to repeat the amplification step $\log n$ times.

Applying parallel repetition to a constraint system that has a constant satisfiability gap, can result in a new system whose gap is $1-\epsilon$ for arbitrarily small $\epsilon > 0$. We remark that having such a gap of nearly 1 has proved extremely useful in inapproximability reductions. In our proof, once the satisfiability gap reached some constant, it does not continue to grow to reach $1-\epsilon$. In fact, very recently Bogdanov [8] gave an example of a constraint graph with a constant satisfiability gap, for which graph powering does not amplify the gap beyond 1/2. This limitation is in agreement with the fact that, generally speaking, derandomization of the parallel repetition theorem is impossible [13].

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[3] L. Babai. Trading group theory for randomness. In *Proc. 17th ACM Symp. on Theory of Computing*, pages 421–429, 1985.

[4] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[5] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 21–31, 1991.

[6] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proc. 36th ACM Symp. on Theory of Computing*, 2004.

[7] M. Ben-or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi prover interactive proofs: How to remove intractability assumptions. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 113–121, 1988.

[8] Anrej Bogdanov. Gap amplification fails below 1/2. Comment on ECCC TR05-046, can be found at `http://eccc.uni-trier.de/eccc-reports/2005/TR05-046/commt01.pdf`, 2005.

[9] Eli Ben-Sasson and Madhu Sudan. Robust PCPs of proximity, shorter PCPs and applications to coding. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005.

[10] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proc. 35th ACM Symp. on Theory of Computing*, pages 612–621, 2003.

[11] Irit Dinur and Omer Reingold. Assignment testers: Towards combinatorial proofs of the PCP theorem. In *Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS)*, 2004.

[12] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. *Journal of the ACM*, 43(2):268–292, 1996.

[13] Uri Feige and Joe Kilian. Impossibility results for recycling random bits in two-prover proof systems. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 457–468, 1995.

[14] L. Fortnow, J. Rompel, and M. Sipser. On the power of multi-prover interactive protocols. In *Proceedings of the 3rd Conference on Structure in Complexity Theory*, pages 156–161, 1988.

[15] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal on Computing*, 18:186–208, 1989.

[16] Goldreich and Safra. A combinatorial consistency lemma with application to proving the PCP theorem. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science.* LNCS, 1997.

[17] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. In *Proc. 43rd IEEE Symp. on Foundations of Computer Science*, pages 13–22, 2002.

[18] Prahladh Harsha and Madhu Sudan. Small PCPs with low query complexity. In *STACS*, pages 327–338, 2001.
[19] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, October 1992.
[20] A. Polishchuk and D. Spielman. Nearly linear size holographic proofs. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 194–203, 1994.
[21] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.
[22] Omer Reingold. Undirected st-connectivity in log-space. In *Proc. 37th ACM Symp. on Theory of Computing*, 2005.
[23] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, October 1992. Prelim. version in 1990 FOCS, pages 11–15.

Geometry and expansion: A survey of recent results

Sanjeev Arora

Graph expansion occurs as a unifying concept across several areas of theoretical computer science, including theory of communication networks, theory of error-correcting codes, theory of approximation algorithms, and theory of computational pseudo-randomness. This brief survey concerns new, geometric ways of looking at expansion that have engendered new breakthroughs in approximation algorithms, geometric embeddings of metric spaces, and probabilistically checkable proofs.

In approximation algorithms the breakthrough is new $O(\sqrt{\log n})$-approximation algorithms for a host of NP-hard optimization problems, starting with the discovery of such an algorithm for SPARSEST CUT in [3]. These new algorithms rely on a new analysis of a family of semidefinite programs.

In geometric embeddings new results include an almost-tight embedding of ℓ_1-spaces into ℓ_2 with distortion $O(\sqrt{\log n}\log\log n)$. There have also been a spate of results ruling out certain types of embeddings, most notably a paper of Khot and Vishnoi which rules out $O(1)$-distortion embedding of ℓ_2^2 into ℓ_1.

Constructions of PCPs in recent years have relied upon theorems in Fourier Analysis which are also geometric in nature, and this has also become clearer thanks to the results on embeddings.

Yet another connection between geometry and expansion is that the above results rely upon a geometric analog of the study of expansion, namely, *isoperimetric problems*. The simplest is the classical result that every closed set in $\Re^2$ whose area is A has perimeter at least $2\sqrt{\pi A}$, the perimeter of the circle of area A. One can in fact prove the stronger statement that if this set has perimeter "close to" $2\sqrt{\pi A}$, then it "looks like" a circle of area A. The latter type of theorems we be referred to as *Strong Isoperimetric Theorems*. Isoperimetric theorems about the n-dimensional sphere and the boolean hypercube play an important role in the above results.

References

[1] S. Arora, E. Hazan, and S. Kale. $O(\sqrt{\log n})$ approximation to Sparsest Cut in $\tilde{O}(n^2)$ time. *IEEE Foundations of Computer Science*.

[2] Sanjeev Arora, James Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. *ACM STOC 2005.*
[3] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings, and graph partitioning. In *ACM STOC 2004*, pages 222–231.

On Lattices, Learning with Errors, Random Linear Codes, and Cryptography

Oded Regev

Our main result is a reduction from worst-case lattice problems such as SVP and SIVP to a certain learning problem. This learning problem is a natural extension of the 'learning from parity with error' problem to higher moduli. It can also be viewed as the problem of decoding from a random linear code. This, we believe, gives a strong indication that these problems are hard. Our reduction, however, is quantum. Hence, an efficient solution to the learning problem implies a *quantum* algorithm for SVP and SIVP. A main open question is whether this reduction can be made classical.

Using the main result, we obtain a public-key cryptosystem whose hardness is based on the worst-case quantum hardness of SVP and SIVP. Previous lattice-based public-key cryptosystems such as the one by Ajtai and Dwork were only based on unique-SVP, a special case of SVP. The new cryptosystem is much more efficient than previous cryptosystems: the public key is of size $\tilde{O}(n^2)$ and encrypting a message increases its size by $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively). In fact, under the assumption that all parties share a random bit string of length $\tilde{O}(n^2)$, the size of the public key can be reduced to $\tilde{O}(n)$.

Main theorem. Let n be some integer and let $\varepsilon \geq 0$ be some real. Consider the 'learning from parity with error' problem, defined as follows: find $\mathbf{s} \in \mathbb{Z}_2^n$ given a list of 'equations with errors'

$$\begin{gathered}\langle \mathbf{s}, \mathbf{a}_1 \rangle \approx_\varepsilon b_1 \pmod 2 \\ \langle \mathbf{s}, \mathbf{a}_2 \rangle \approx_\varepsilon b_2 \pmod 2 \\ \vdots\end{gathered}$$

where the $\mathbf{a}_i$'s are chosen independently from the uniform distribution on $\mathbb{Z}_2^n$ and $\langle \mathbf{s}, \mathbf{a}_i \rangle = \sum_j s_j (a_i)_j$ is the inner product modulo 2 of $\mathbf{s}$ and $\mathbf{a}_i$. The input to the problem consists of the pairs $(\mathbf{a}_i, b_i)$ and the output is a guess for $\mathbf{s}$. By the $\approx_\varepsilon$ symbol we mean that each equation is independently chosen to be correct with probability $1 - \varepsilon$ and incorrect with probability ε. Notice that the case $\varepsilon = 0$ can be solved efficiently by, say, Gaussian elimination. This requires $O(n)$ equations and poly(n) time.

The problem seems to become significantly harder when we take any positive $\varepsilon > 0$. For example, let us consider again the Gaussian elimination process and assume we are interested in recovering only the first bit of $\mathbf{s}$. Using Gaussian elimination, we can find a set S of $O(n)$ equations such that $\sum_S \mathbf{a}_i$ is $(1, 0, \ldots, 0)$. Summing the corresponding values b_i gives us a guess for the first bit of $\mathbf{s}$. However, a standard calculation shows that this guess is correct with probability $\frac{1}{2} + 2^{-\Theta(n)}$. Hence, in order to obtain the first bit with good confidence, we have to repeat the whole procedure $2^{\Theta(n)}$ times. This yields an algorithm that uses $2^{O(n)}$ equations and $2^{O(n)}$ time. In fact, it can be shown that given only $O(n)$ equations, the $\mathbf{s}' \in \mathbb{Z}_2^n$ that maximizes the number of satisfied equations is with high probability $\mathbf{s}$. This yields a simple maximum likelihood algorithm that requires only $O(n)$ equations and runs in time $2^{O(n)}$.

Blum, Kalai, and Wasserman [8] provided the first subexponential algorithm for this problem. Their algorithm requires only $2^{O(n/\log n)}$ equations/time and is currently the best known algorithm for the problem. It is based on a clever idea that allows to find a small set S of equations (say, $O(\sqrt{n})$) among $2^{O(n/\log n)}$ equations, such that $\sum_S \mathbf{a}_i$ is, say, $(1, 0, \ldots, 0)$. This gives us a guess for the first bit of $\mathbf{s}$ that is correct with probability $\frac{1}{2} + 2^{-\Theta(\sqrt{n})}$. We can obtain the correct value with high probability by repeating the whole procedure only $2^{O(\sqrt{n})}$ times. Their algorithm was later shown to have other important applications, such as the first $2^{O(n)}$-time algorithm for solving the shortest vector problem in a lattice [11, 5].

An important open question is to explain the apparent difficulty in finding efficient algorithms for this learning problem. Our main theorem explains this difficulty for a natural extension of this problem to higher moduli, defined next.

Let $p = p(n) \le \mathrm{poly}(n)$ be some prime integer and consider a list of 'equations with error'

$$\langle \mathbf{s}, \mathbf{a}_1 \rangle \approx_\chi b_1 \pmod{p}$$
$$\langle \mathbf{s}, \mathbf{a}_2 \rangle \approx_\chi b_2 \pmod{p}$$
$$\vdots$$

where this time $\mathbf{s} \in \mathbb{Z}_p^n$, $\mathbf{a}_i$ are chosen independently and uniformly from $\mathbb{Z}_p^n$, and $b_i \in \mathbb{Z}_p$. The error in the equations is now specified by a probability distribution $\chi : \mathbb{Z}_p \to \mathbb{R}^+$ on $\mathbb{Z}_p$. Namely, for each equation i, $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$ where each $e_i \in \mathbb{Z}_p$ is chosen independently according to χ. We denote the problem of recovering $\mathbf{s}$ from such equations by $\mathsf{LWE}_{p,\chi}$ (learning with error). For example, the learning from parity problem with error ε is the special case where $p = 2$, $\chi(0) = 1 - \varepsilon$, and $\chi(1) = \varepsilon$. Under a reasonable assumption on χ (namely, that $\chi(0) > 1/p + 1/\mathrm{poly}(n)$), the maximum likelihood algorithm described above solves $\mathsf{LWE}_{p,\chi}$ for $p \le \mathrm{poly}(n)$ using $\mathrm{poly}(n)$ equations and $2^{O(n \log n)}$ time. Under a similar assumption, an algorithm resembling the one by Blum et al. [8] requires only $2^{O(n)}$ equations/time. This is the best known algorithm for the LWE problem.

Our main theorem shows that for certain choices of p and χ, a solution to $\mathsf{LWE}_{p,\chi}$ implies a quantum solution to worst-case lattice problems.

Theorem 1 (Informal)**.** *Let n, p be integers and $\alpha \in (0, 1)$ be some real such that $\alpha p > 2\sqrt{n}$. If there exists a polynomial time algorithm that solves $\mathsf{LWE}_{p,\bar{\Psi}_\alpha}$ then* *there exists a quantum algorithm that approximates the shortest vector problem (*SVP*) and the shortest independent vectors problem (*SIVP*) to within $\tilde{O}(n/\alpha)$ in the worst case.*

We define $\bar{\Psi}_\alpha$ as a distribution on $\mathbb{Z}_p$ that has the shape of a discrete Gaussian centered around 0 with standard deviation αp. Also, the probability of 0 (i.e., no error) is roughly $1/(\alpha p)$. A possible setting for the parameters is $p = O(n^2)$ and $\alpha = 1/(\sqrt{n}\log n)$ (in fact, these are the parameters that we use in our cryptographic application).

The SVP and SIVP are two of the main computational problems on lattices. The best known polynomial time algorithms yield only mildly subexponential approximation factors. It is conjectured that there is no classical polynomial time algorithm that approximates them to within any polynomial factor. Lattice-based constructions of one-way functions, such as the one by Ajtai [2], are based on this conjecture.

One might guess that the same conjecture holds in the quantum world, i.e., there is no quantum polynomial time algorithm that approximates SVP (or SIVP) to within any polynomial factor. Thus one can interpret the main theorem as saying that based on this conjecture, the LWE problem is hard. The only evidence supporting this conjecture is that there are no quantum algorithms for lattice problems that are known to outperform classical algorithms, even though this is probably one of the most important open questions in the field of quantum computing. We do not know, however, if this conjecture is true.

In fact, one could also interpret our main theorem as a way to disprove this conjecture: if one finds an efficient algorithm for LWE, then one also obtains a quantum algorithm for approximating worst-case lattice problems. Such a result would be of tremendous importance on its own. Finally, we would like to stress that it is possible that our result can be made classical. This would make all our results stronger and the above discussion unnecessary.

The LWE problem can be equivalently presented as the problem of decoding random linear codes. More specifically, let $m = \mathrm{poly}(n)$ be arbitrary and let $\mathbf{s} \in \mathbb{Z}_p^n$ be some vector. Then, consider the following problem: given a random matrix $Q \in \mathbb{Z}_p^{m\times n}$ and the vector $\mathbf{t} = Q\mathbf{s} + \mathbf{e} \in \mathbb{Z}_p^m$ where each coordinate of the error vector $\mathbf{e} \in \mathbb{Z}_p^m$ is chosen independently from $\bar{\Psi}_\alpha$, recover $\mathbf{s}$. The Hamming weight of $\mathbf{e}$ is roughly $m(1 - 1/(\alpha p))$ (since a value chosen from $\bar{\Psi}_\alpha$ is 0 with probability roughly $1/(\alpha p)$). Hence, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}$ is roughly $m(1-1/(\alpha p))$. Moreover, it can be seen that for large enough m, for any other word $\mathbf{s}'$, the Hamming distance of $\mathbf{t}$ from $Q\mathbf{s}'$ is roughly $m(1 - 1/p)$. Hence, we obtain that approximating the nearest codeword problem to within factors smaller than $(1 - 1/p)/(1 - 1/(\alpha p))$ on random codes is as hard as quantumly approximating worst-case lattice problems. This gives a partial answer to the important open question of understanding the hardness of decoding from random linear codes.

References

[1] D. Aharonov and O. Regev. Lattice problems in NP intersect coNP. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 362–371, 2004.
[2] M. Ajtai. Generating hard instances of lattice problems. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, 1996.
[3] M. Ajtai. Representing hard lattices with $O(n \log n)$ bits. In *Proc. 37th Annual ACM Symp. on Theory of Computing (STOC)*, 2005.
[4] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th Annual ACM Symp. on Theory of Computing (STOC)*, pages 284–293, 1997.
[5] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 601–610, 2001.
[6] M. Alekhnovich. More on average case vs approximation complexity. In *Proc. 44th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 298–307, 2003.
[7] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in cryptology—CRYPTO '93 (Santa Barbara, CA, 1993)*, volume 773 of *Lecture Notes in Comput. Sci.*, pages 278–291. Springer, Berlin, 1994.
[8] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.
[9] J.-Y. Cai and A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *Proc. 38th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 468–477, 1997.
[10] U. Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th Annual ACM Symp. on Theory of Computing (STOC)*, pages 534–543, 2002.
[11] R. Kumar and D. Sivakumar. On polynomial approximation to the shortest lattice vector length. In *Proc. 12th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 126–127, 2001.
[12] D. Micciancio. Improved cryptographic hash functions with worst-case/average-case connection. In *Proc. 34th Annual ACM Symp. on Theory of Computing (STOC)*, pages 609–618, 2002.
[13] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 2004. Accepted for publication. Available from author's web page at URL http://www.cse.ucsd.edu/users/daniele.
[14] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, 2004.
[15] O. Regev. New lattice based cryptographic constructions. In *Proc. 35th Annual ACM Symp. on Theory of Computing (STOC)*, pages 407–416, 2003.

Cryptography in NC^0

Yuval Ishai
(joint work with Benny Applebaum and Eyal Kushilevitz)

The efficiency of cryptographic primitives is of both theoretical and practical interest. In this work, we consider the question of minimizing the *parallel time-complexity* of basic cryptographic primitives such as one-way functions (OWFs) and pseudorandom generators (PRGs) [2, 12]. Taking this question to an extreme, it is natural to ask if there are instances of these primitives that can be computed in *constant* parallel time. Specifically, the following fundamental question was posed in several previous works (e.g., [5, 4, 3, 8, 9]):

> Are there one-way functions, or even pseudorandom generators, in NC^0?

Recall that NC^0 is the class of functions that can be computed by (a uniform family of) constant-depth circuits with bounded fan-in. In an NC^0 function each bit of the output depends on a constant number of input bits. We refer to this constant as the *output locality* of the function and denote by NC^0_c the class of NC^0 functions with locality c.

The above question is qualitatively interesting, since one might be tempted to conjecture that cryptographic hardness requires some output bits to depend on many input bits. Indeed, this view is advocated by Cryan and Miltersen [3], whereas Goldreich [4] takes an opposite view and suggests a concrete candidate for OWF in NC^0. However, despite previous efforts, there has been no convincing theoretical evidence supporting either a positive or a negative resolution of this question.

Our Results. As indicated above, the possibility of implementing most cryptographic primitives in NC^0 was left wide open. We present a positive answer to this basic question, showing that surprisingly many cryptographic tasks can be performed in constant parallel time.

Since the existence of cryptographic primitives implies that $\mathrm{P} \neq \mathrm{NP}$, we cannot expect unconditional results and have to rely on some unproven assumptions.[1] However, we avoid relying on *specific* intractability assumptions. Instead, we assume the existence of cryptographic primitives in a relatively "high" complexity class and transform them to the seemingly degenerate complexity class NC^0 without substantial loss of their cryptographic strength. These transformations are inherently non-black-box, thus providing further evidence for the usefulness of non-black-box techniques in cryptography.

We now give a more detailed account of our results.

A GENERAL COMPILER. Our main result is that any OWF (resp., PRG) in a relatively high complexity class, containing uniform NC^1 and even $\oplus\mathrm{L}/poly$, can be efficiently "compiled" into a corresponding OWF (resp., sublinear-stretch PRG) in NC^0_4. (The class $\oplus\mathrm{L}/poly$ contains the classes $\mathrm{L}/poly$ and NC^1 and is contained in NC^2. In a non-uniform setting it also contains the class $\mathrm{NL}/poly$ [11].) The existence of OWF and PRG in this class is a mild assumption, implied in particular by most number-theoretic or algebraic intractability assumptions commonly used in cryptography. Hence, the existence of OWF and sublinear-stretch PRG in NC^0 follows from a variety of standard assumptions and is not affected by the potential weakness of a particular algebraic structure. A similar compiler can also be obtained for other cryptographic primitives including one-way permutations, encryption, signatures, commitment, and collision-resistant hashing.

[1] This is not the case for non-cryptographic PRGs such as ϵ-biased generators, for which we do obtain unconditional results.

It is important to note that the PRG produced by our compiler will generally have a sublinear additive stretch even if the original PRG has a large stretch. However, one cannot do much better when insisting on an NC^0_4 PRG, as there is no PRG with superlinear stretch in NC^0_4 [9].

OWF WITH OPTIMAL LOCALITY. The above results leave a small gap between the possibility of cryptography in NC^0_4 and the known impossibility of implementing even OWF in NC^0_2. We partially close this gap by providing positive evidence for the existence of OWF in NC^0_3. In particular, we construct such OWF based on the intractability of decoding a random linear code.

NON-CRYPTOGRAPHIC GENERATORS. Our techniques can also be applied to obtain unconditional constructions of non-cryptographic PRGs. In particular, building on an ϵ-biased generator in NC^0_5 constructed by Mossel et al. [9], we obtain a linear-stretch ϵ-biased generator in NC^0_3. This generator has optimal locality, answering an open question posed in [9]. It is also essentially optimal with respect to stretch, since locality 3 does not allow for a superlinear stretch [3]. Our techniques apply also to other types of non-cryptographic PRGs such as generators for space-bounded computation [1, 10], yielding such generators (with sublinear stretch) in NC^0_3.

Techniques. Our key observation is that instead of computing a given "cryptographic" function $f(x)$, it might suffice to compute a function $\hat{f}(x,r)$ having the following relation to f:

1. For every fixed input x and a uniformly random choice of r, the output distribution $\hat{f}(x,r)$ forms a "randomized encoding" of $f(x)$, from which $f(x)$ can be decoded. That is, if $f(x) \neq f(x')$ then the random variables $\hat{f}(x,r)$ and $\hat{f}(x',r')$, induced by a uniform choice of r,r', should have disjoint supports.
2. The distribution of this randomized encoding depends only on the encoded value $f(x)$ and does not further depend on x. That is, if $f(x) = f(x')$ then the random variables $\hat{f}(x,r)$ and $\hat{f}(x',r')$ should be identically distributed. Furthermore, we require that the randomized encoding of an output value y be efficiently samplable given y. Intuitively, this means that the output distribution of $\hat{f}$ on input x reveals no information about x except what follows from $f(x)$.

Each of these requirements alone can be satisfied by a trivial function $\hat{f}$ (e.g., $\hat{f}(x,r) = x$ and $\hat{f}(x,r) = 0$, respectively). However, the combination of the two requirements can be viewed as a non-trivial natural relaxation of the usual notion of computing. In a sense, the function $\hat{f}$ defines an "information-theoretically equivalent" representation of f. In the following, we refer to $\hat{f}$ as a *randomized encoding* of f.

For this approach to be useful in our context, two conditions should be met. First, we show that a randomized encoding $\hat{f}$ can be *securely* used as a substitute for f. For instance, if f is a OWF then so is $\hat{f}$. Second, we show that this relaxation is sufficiently *liberal*, in the sense that it allows to efficiently encode relatively complex functions f by functions $\hat{f}$ in NC^0. Our main constructions of randomized encodings in NC^0 build on the machinery of *randomizing polynomials* from [6, 7], where it was shown that any function f in $\oplus\mathrm{L}/poly$ can be efficiently encoded by a function $\hat{f}$ whose *algebraic degree* is 3. The notion of randomizing polynomials was originally motivated by questions in the seemingly unrelated domain of information-theoretic secure multiparty computation.

References

[1] L. Babai, N. Nisan, and M. Szegedy, *Multiparty protocols and logspace-hard pseudorandom sequences*, Proc. 21st STOC, 1–11, 1989.

[2] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput. **13**:850–864, 1984.

[3] M. Cryan and P. B. Miltersen, *On pseudorandom generators in* NC^0, Proc. 26th MFCS, 2001.

[4] O. Goldreich, *Candidate one-way functions based on expander graphs*, Electronic Colloquium on Computational Complexity (ECCC) **7**(090), 2000.

[5] J. Håstad, *One-way permutations in* NC^0, Information Processing Letters **26**:153–155, 1987.

[6] Y. Ishai and E. Kushilevitz, *Randomizing polynomials: A new representation with applications to round-efficient secure computation*, Proc. 41st FOCS, 294–304, 2000.

[7] Y. Ishai and E. Kushilevitz, *Perfect constant-round secure computation via perfect randomizing polynomials*, Proc. 29th ICALP, 244–256, 2002.

[8] M. Krause and S. Lucks, *On the minimal hardware complexity of pseudorandom function generators (extended abstract)*, Proc. 18th STACS, 419–430, 2001.

[9] E. Mossel, A. Shpilka, and L. Trevisan, *On ϵ-biased generators in* NC^0, Proc. 44th FOCS, 136–145, 2003.

[10] N. Nisan, *Pseudorandom generators for space-bounded computation*, Combinatorica, **12**(4):449–461, 1992.

[11] A. Wigderson, $\mathrm{NL}/poly \subseteq \oplus\mathrm{L}/poly$, Proc. 9th Structure in Complexity Theory Conference, 59–62, 1994.

[12] A. C. Yao, *Theory and application of trapdoor functions*, Proc. 23rd FOCS, 80–91, 1982.

If NP languages are hard on the worst-case then it is easy to find their hard instances

Amnon Ta-Shma

(joint work with Dan Gutfreund, Ronen Shaltiel)

It is traditional in computational complexity to measure *worst-case* complexities, and say that an algorithm is feasible if it can be solved in worst-case polynomial time (i.e., in P or BPP). A general belief is that all NP-complete languages do not have feasible algorithms that are correct on every input. Thus under a worst-case measure of complexity, these problems are hard. However, this does not mean that *in practice* NP-complete problems are hard. It is possible that for a given problem, its hard instances are "rare", and in fact it is solvable efficiently on all instances that actually appear in practice.

Trying to capture the notion of "real-life" instances, we look at input distributions that can be efficiently generated. Often, we don't have precise knowledge of the distribution of the inputs, and even worse, this distribution may change in the future. A reasonable guarantee is that the inputs are drawn from some samplable distribution. We say that $\mathcal{D}$ is *samplable* if there exists some probabilistic polynomial-time machine that generates the distribution. We would like to design an algorithm that is guaranteed to succeed with good probability whenever the inputs are sampled from some samplable distribution. This gives rise to the following definition, due to Kabanets [8].

Definition 1. *(Pseudo classes) Let $\mathcal{C}$ be a class of algorithms and L a language. We say that $L \in Pseudo_{p(n)}\ \mathcal{C}$ if there exists an algorithm $B \in \mathcal{C}$ such that for every samplable distributions $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ we have that for large enough n, $\Pr_{x\in\mathcal{D}_n}[B(x) = L(x)] \geq p(n)$.*

When $\mathcal{C}$ is a class of probabilistic algorithms, there are subtleties in this definition. In this abstract we ignore these subtleties and we refer the reader to the paper [5].

Our main result is a worst-case to average-case reduction for PseudoBPP.

Theorem 2.

1. $NP \neq P \ \Rightarrow\ NP \not\subseteq Pseudo_{5/6}\ P$
2. $NP \neq RP \ \Rightarrow\ NP \not\subseteq Pseudo_{97/100}\ BPP$

This worst-case to average-case reduction in the algorithmic setting, stands in contrast to the failure in proving such a reduction in the cryptographic setting (for the class Avg BPP[4, 3, 9]). To the best of our knowledge, it is the first worst-case to average-case reduction for NP-complete languages under a natural notion of average-case complexity. Stated in words, Theorem 2 says that if NP is hard on the worst case then for any efficient algorithm trying to solve some NP complete language it is possible to efficiently sample instances on which the algorithm errs.

Overview of the technique. We now give a high level overview of the proof of Theorem 2. We assume that NP $\neq$ P, our goal is to show that for any deterministic algorithm BSAT there is a samplable distribution which generates hard instances for BSAT. The main step in the proof is a lemma that shows that there is a deterministic procedure R that when given as input the description of BSAT and an input n outputs at most three formulas, and for infinitely many n, BSAT errs on at least one of the formulas. In other words, the procedure R *finds* instances such that one of them is hard for BSAT.

We know that BSAT does not solve SAT, fix some length n on which BSAT makes an error. The basic idea is to consider the following statement denoted ϕ_n: *"there exists an instance x of length n such that $BSAT(x) \neq SAT(x)$".* Note that this statement is a true statement. If this statement was an NP statement then we could reduce it into an instance of SAT and feed it to BSAT. If BSAT answers 'no' then ϕ_n is an instance on which BSAT errs. If BSAT answers 'yes' then in some sense BSAT "admits" that it makes an error on inputs of length n. We can hope to use BSAT to *find* a witness x to ϕ_n and such a witness x is a formula on which BSAT errs.

Note however, that at the moment it is not necessarily the case that deciding ϕ_n is in NP. This is because it could be the case that BSAT errs only on unsatisfiable formulas. (Say for example that BSAT always answers 'yes'.) Verifying that ϕ_n holds seems to require verifying that a given formula x is unsatisfiable. We overcome this difficulty by replacing BSAT with an algorithm SSAT that has the following properties:

- When SSAT answers 'yes' then it also outputs a satisfying assignment, and in particular it never errs when it answers 'yes'.
- If SSAT answers 'no' then BSAT answers 'no'.
- If BSAT answers 'yes' on input x then either SSAT answers 'yes' (and finds a satisfying assignment) or else SSAT outputs three formulas such that BSAT errs on at least one of them.

It is easy to construct such an algorithm SSAT by using the standard self-reducibility property of SAT. More precisely, on input x, the algorithm SSAT attempts to use BSAT to find a satisfying assignment. In every step it holds a formula x that BSAT answers 'yes' on. It then substitutes one variable of x to both "zero" and "one" and feeds these formulas to BSAT. If BSAT answers 'yes' on one of them, then the search continues on this formula. Otherwise, at least one of the answers of BSAT on x and the two derived formulas is clearly incorrect. Finally, SSAT accepts if it finds a satisfying assignment. It is easy to verify that SSAT has the properties listed above.

To find a hard instance we change ϕ_n to be the following statement: *"there exists an instance x of length n such that $SAT(x) = 1$ yet $SSAT(x) \neq$ 'yes'"*. Note that now deciding ϕ_n is in NP and therefore we can reduce it to a formula. To find hard instances we run SSAT(ϕ_n). There are three possibilities.

- SSAT finds three instances such that on one of them BSAT errs.
- SSAT answers 'no', but in this case BSAT answers 'no' and ϕ_n is a formula on which BSAT errs.
- SSAT answers 'yes' and finds a satisfying assignment x.

It is important to stress that we're not yet done in the third case. While we know that SSAT errs on x, it's not necessarily the case that BSAT errs on x. In the third case, we run SSAT on x. This time we know that the third possibility cannot occur (because we are guaranteed that SSAT does not answer 'yes' on x) and therefore we will be able to find a hard instance.

Extending the argument to the case where BSAT is randomized. is done as follows. We say that a randomized algorithm conforms with confidence level 2/3 if for every input x, either the algorithm accepts x with probability 2/3 or it rejects x with probability 2/3. When given such an algorithm BSAT we can easily use amplification and get an algorithm $\overline{\text{BSAT}}$ that conforms with confidence level $1 - 2^{-2n}$. As in Adelman's argument [1], for almost all choices of random strings u, $\overline{\text{BSAT}}(\cdot, u)$'s answer "captures" whether BSAT accepts or rejects x. Thus, we can do the same argument as above replacing BSAT with $\overline{\text{BSAT}}(\cdot, u)$ for a uniformly chosen u. We will find hard instances for $\overline{\text{BSAT}}(\cdot, u)$, and with high probability (over the choice of u) one of the instances will be a formula on which BSAT errs with noticeable probability.

In general, we cannot assume that BSAT conforms to some confidence level. (For example, BSAT is allowed to flip a coin on some instances). For the general case, we use amplified versions of BSAT and SSAT, together with a more cumbersome case-analysis to implement the idea of the deterministic case.

References

[1] L. Adelman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.
[2] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 379–386, 1990.
[3] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 308–317, 2003.
[4] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993.
[5] D. Gutfreund, R. Shaltiel, and A. Ta-Shma. If NP languages are hard on the worst-case then it is easy to find their hard instances. *Proceedings of the Twentieth Annual IEEE Conference on Computational Complexity*, ??–??, 2005.
[6] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Annual Conference on Structure in Complexity Theory*, pages 134–147, 1995.
[7] R. Impagliazzo and A. Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 734–743, 1998.
[8] V. Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63 (2):236–252, 2001.
[9] E. Viola. Hardness vs. randomness within alternating time. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity*, pages 53–62, 2003.

3-Server Information-Theoretic Private-Information Retrieval

Eyal Kushilevitz
(joint work with A. Beimel, Y. Ishai and J.F. Raymond)

We survey the state-of-the-art in information-theoretic Private Information Retrieval (PIR) protocols. In such protocols there are k servers $S_1, \dots, S_k$, each holding an identical copy of an n-bit string x (sometimes referred to as the "database") and a user U that holds an index $i \in [n]$. The goal of such a protocol is for the user to learn x_i while keeping i secret from each of the servers.[1] There is a trivial solution for the problem: let one of the servers, e.g. S_1, send the entire string x to the user. While this indeed solves the problem, the communication

[1]Various natural extensions and generalizations of this problem are discussed in the literature but are ignored in this survey. Examples of such extensions include the problem of *computational* PIR (where privacy is obtained by using cryptographic assumption and under the assumption that the server(s) are limited to efficient computations) [5, 12, 7], symmetric PIR (where there is an additional requirement that the user learns no information on x other than the value of x_i) [8], PIR against coalitions of t servers [6, 9, 13], etc.

complexity of this protocol (i.e., n bits of communication) is too large. In contrast, without the privacy requirement $\log n + 1$ bits suffice for the user to learn x_i. The main goal of PIR research is to get the communication complexity lower (this alone can be shown to require $k > 1$ servers).

The study of PIR protocols was initiated by Chor et al [6] and since then attracted a significant amount of attention. Specifically, the following results are known: (1) If the number of servers, k, can be a function of the database size, n, then k =polylog(n) servers suffice for obtaining polylog(n) communication [6, 2]. (2) A protocol for $k = 2$ servers with communication complexity $O(n^{1/3})$ and for general k with communication complexity $O(n^{1/k})$ [6]. (3) k-server protocol with communication complexity $O(n^{1/2k-1})$ [1, 10, 9, 3, 13].[2] (4) k-server protocol with communication complexity $O(n^{c \log\log k/(k \log k)})$ [4]. Some of these results also have implications for the problem of constructing Locally Decodable Codes (LDCs) [11]. Specifically, the best known LDCs are constructed via (binary answer) PIR protocols.

This survey concentrates on the case of $k = 3$ servers.[3] We present three protocols:

- A protocol of complexity $O(n^{1/2})$. This protocol (as well as the following protocols) uses as basic ingredients *arithmetization* and *replication secret-sharing*. This is a binary-answer PIR protocol (of the type needed to construct LDCs) and is still the best known protocol of this type for $k = 3$.
- A protocol of complexity $O(n^{1/5})$. This protocol balances the communication between the user and servers by making a simple observation about the structure of the (low degree) polynomials that are coming out of the arithmetization and the replication secret-sharing scheme.
- A protocol of complexity $O(n^{4/21})$. This protocol, on top of the above ingredients, uses recursion in the context of PIR, which is the main idea in [4].

References

[1] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proc. of the 24th International Colloquium on Automata, Languages and Programming*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407. Springer-Verlag, 1997.

[2] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In C. Choffrut and T. Lengauer, editors, *STACS '90, 7th Symp. on Theoretical Aspects of Computer Science*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, 1990.

[3] A. Beimel and Y. Ishai. Information-theoretic private information retrieval: A unified construction. In P. G. Spirakis and J. van Leeuven, editors, *Proc. of the 28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 912–926. Springer, 2001.

[2] Each of the papers in this sequence of works achieves some improvements over the previous ones in various aspects; e.g., in the dependency of the complexity in k.

[3] This number of servers is the smallest k where all the currently known techniques already affect the complexity.

[4] A. Beimel, Y. Ishai, E. Kushilevitz, and J. F. Raymond. Breaking the $O(n^{\frac{1}{2k-1}})$ barrier for information-theoretic private information retrieval. In *Proc. of the 43rd IEEE Symp. on Foundations of Computer Science*, pages 261–270, 2002.
[5] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of the 29th ACM Symp. on the Theory of Computing*, pages 304–313, 1997.
[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proc. of the 36th IEEE Symp. on Foundations of Computer Science*, pages 41–51, 1995. Journal version: *J. of the ACM*, 45:965–981, 1998.
[7] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer-Verlag, 1999.
[8] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proc. of the 30th ACM Symp. on the Theory of Computing*, pages 151–160, 1998. Journal version: *J. of Computer and System Sciences*, 60(3):592–629, 2000.
[9] Y. Ishai and E. Kushilevitz. Improved upper bounds on information theoretic private information retrieval. In *Proc. of the 31st ACM Symp. on the Theory of Computing*, pages 79 – 88, 1999.
[10] T. Itoh. Efficient private information retrieval. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E82-A(1):11–20, 1999.
[11] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. of the 32nd ACM Symp. on the Theory of Computing*, pages 80–86, 2000.
[12] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proc. of the 38th IEEE Symp. on Foundations of Computer Science*, pages 364–373, 1997.
[13] D. Woodruff and S. Yekhanin. A geometric approach to information-theoretic private information retrieval. In *Proc. of the 20th IEEE Conf. on Computational Complexity*, 2005.

Holographic Algorithms

Leslie Valiant

Using the notion of polynomial time reduction computer scientists have discovered an astonishingly rich web of interrelationships among the myriad natural computational problems that arise in diverse applications. These relationships have been used both to give evidence of intractability, such as that of NP-completeness [1, 2] or #P-completeness [3], as well as some surprising new algorithms [4].

In this talk we discuss a new notion of reduction [5], which we call a holographic reduction, that is more general than the traditional one in the following sense. Instead of locally mapping solutions one-to-one it maps them many-to-many but preserves the sum of the solutions.

One application is to finding new polynomial time algorithms where none was known before. We shall give such algorithms for several counting problems related to planar graphs. These include a restricted case of the problem of counting the number matchings in planar graphs, the unrestricted case being known to be #P-complete [6]. They also include counting the number of satisfying assignments of a planar formula consisting of not-all-equal-of-3 gates, and counting the number of ways the edges of a cubic planar graph can be directed so that there are no sources or sinks. Also included is the problem of deciding the minimal number of nodes that need to be removed from a degree three planar graph to make it bipartite, and the problem of counting the parity of approximate solutions of planar linear equations of even length over GF[2].

A more radical proposal is that of revisiting the currently accepted conjectures of computer science, such as that P $\neq$ NP, and seeing whether holographic reductions offer any insights towards either positive or negative resolutions. The talk reviews complexity theory in this light. We show that there exist infinite families of polynomials with integer coefficients such that the existence of a solution over the complex numbers for any one member would imply the existence of fixed size algebraic gadgets for certain natural fixed size combinatorial constraints, the existence of which in turn would imply that there are polynomial time algorithms for #P. This relationship may be viewed both as an approach to finding surprising new algorithms, and also as a restricted model of computation for which lower bound proofs might be sought.

References

[1] S. A. Cook 1971. The complexity of theorem proving procedures. *Proc. 3rd ACM Symp. on Theory of Computing*: 151–158.
[2] R. M. Karp 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations* (R. E. Miller and J. W. Thatcher, eds.), Plenum Press, New York, pp. 85–103
[3] L. G. Valiant 1979b. The complexity of enumeration and reliability problems, *SIAM J. on Comput.* 8, 3: 410–421.
[4] V. Strassen 1969. Gaussian elimination is not optimal, *Numer. Math.* 14(3): 354–356.
[5] L.G. Valiant 2004. Holographic algorithms, *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Press, 306-315.
[6] M.R. Jerrum 1987. Two-dimensional monomer-dimer systems are computationally intractable, *J. Statistical Physics* 48, 1/2: 121–134. (Also 1990, 59,3/4: 1087–1088.)

Are Quantum States Exponentially Long Vectors?

Scott Aaronson

I'm grateful to Oded Goldreich for inviting me to the 2005 Oberwolfach Complexity Theory meeting. In this extended abstract, which is based on a talk that I gave there, I demonstrate that gratitude by explaining why Goldreich's views about quantum computing are wrong.

Why should anyone care? Because in my opinion, Goldreich, along with Leonid Levin [6] and other "extreme" quantum computing skeptics, deserves credit for focusing attention on the key issues, the ones that ought to motivate quantum computing research in the first place. Personally, I have never lain awake at night yearning for the factors of a 1024-bit RSA modulus, let alone the class group of a number field. The real reason to study quantum computing is not to learn other people's secrets, but to unravel the ultimate Secret of Secrets: *is our universe a polynomial or an exponential place?*

Last year Goldreich [5] came down firmly on the "polynomial" side, in a short essay expressing his belief that quantum computing is impossible not only in practice but also in principle:

> As far as I am concern[ed], the QC model consists of exponentially-long vectors (possible configurations) and some "uniform" (or "simple") operations (computation steps) on such vectors ... The key point is that the associated complexity measure postulates that each such operation can be effected at unit cost (or unit time). My main concern is with this postulate. My own intuition is that the cost of such an operation or of maintaining such vectors should be linearly related to the amount of "non-degeneracy" of these vectors, where the "non-degeneracy" may vary from a constant to linear in the length of the vector (depending on the vector). Needless to say, I am not suggesting a concrete definition of "non-degeneracy," I am merely conjecturing that such exists and that it capture[s] the inherent cost of the computation.

My response consists of two theorem-encrusted prongs:[1] first, that you'd have trouble explaining even current experiments, if you didn't think that quantum states really *were* exponentially long vectors; and second, that for most complexity-theoretic purposes, the exponentiality of quantum states is not that much "worse" than the exponentiality of classical probability distributions, which of course nobody complains about. Due to the length limitation, in this abstract I'll discuss only the first prong, which is based on my paper "Multilinear Formulas and Skepticism of Quantum Computing" [1], and not the second prong, which is based on my paper "Limitations of Quantum Advice and One-Way Communication" [2].

Prong 1: Quantum States *Are* Exponential

For me, the main weakness in the arguments of quantum computing skeptics has always been their failure to suggest an answer to the following question: *what criterion separates the quantum states we're sure we can prepare, from the states that arise in Shor's factoring algorithm?* I call such a criterion a "Sure/Shor separator." To be clear, I'm not asking for a red line partitioning Hilbert space into two regions, "accessible" and "inaccessible." But a skeptic could at least propose a complexity measure for quantum states, and then declare that a state of n qubits is "efficiently accessible" only if its complexity is upper-bounded by a small polynomial in n.

In his essay [5], Goldreich agrees that such a Sure/Shor separator would be desirable, but avers that it's not his job to propose one. Motivated by the "hands-off" approach of Goldreich and other skeptics, in [1] I tried to carry out the skeptics' research program for them, by proposing and analyzing possible Sure/Shor separators. The goal was to illustrate what a scientific argument against quantum computing might look like.

[1] Sanjeev Arora asked why I don't have *three* prongs, thereby forming a ψ-shaped pitchfork.

For starters, such an argument would take care to assert the impossibility only of *future* experiments, not experiments that have already been done. So for example, it would not dismiss exponentially-small amplitudes as physically meaningless, since one can easily produce such amplitudes by polarizing n photons each at $45°$. Nor would it appeal to the "absurd" number of particles that a quantum computer would need to maintain coherently—since, to give one example, the Zeilinger group's C_{60} experiment [3] has already demonstrated "Schrödinger cat states," of the form $\frac{|0\rangle^{\otimes n}+|1\rangle^{\otimes n}}{\sqrt{2}}$, for n large enough to be interesting for quantum computation.

Of course, the real problem is that, once we accept $|\psi\rangle$ and $|\varphi\rangle$ into our set of possible states, consistency almost *forces* us to accept $\alpha|\psi\rangle + \beta|\varphi\rangle$ and $|\psi\rangle \otimes |\varphi\rangle$ as well. So is there any defensible place to draw a line? This conundrum is what led me to investigate "tree states": the class of n-qubit pure states that are expressible by polynomial-size *trees* of linear combinations and tensor products. As an example, the state $\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) \otimes \cdots \otimes \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)$ is a tree state; and indeed, so is any state that can be written succinctly in the Dirac notation, using only the symbols $|0\rangle, |1\rangle, +, \otimes, (,)$ together with constants (no $\sum$'s allowed). In evaluating tree states as a possible Sure/Shor separator, we need to address two questions: first, should all quantum states that arise in present-day experiments be seen as tree states? And second, would a quantum computer allow the creation of non-tree states?

My results imply a positive answer to the second question: not only could a quantum computer efficiently generate non-tree states, but such states arise naturally in several quantum algorithms.[2] In particular, let C be a random linear code over $\mathbb{GF}_2$. Then with overwhelming probability, a uniform superposition over the codewords of C cannot be represented by any tree of size $n^{\varepsilon \log n}$, for some fixed $\varepsilon > 0$. Indeed, $n^{\Omega(\log n)}$ symbols would be needed even to *approximate* such a state well in L_2-distance, and even if we replaced the random linear code by a certain explicit code (obtained by concatenating the Reed-Solomon and Hadamard codes). I also showed an $n^{\Omega(\log n)}$ lower bound for the states arising in Shor's algorithm, modulo a number-theoretic conjecture: basically, that the multiples of a large prime number, when written in binary, constitute a decent erasure code. All of these results rely on a spectacular recent advance in classical theoretical computer science: Raz's superpolynomial lower bounds on multilinear formula size [7] (which were proven about a month before I needed them for my application!). Incidentally, in all of the cases discussed above, I conjecture that the actual tree sizes are exponential in n; currently, though, Raz's method can only prove lower bounds of the form $n^{\Omega(\log n)}$.[3]

[2] On the other hand, I do not know whether a quantum computer restricted to tree states always has an efficient classical simulation. All I can show is that such a computer would be simulable in $\Sigma_3^p \cap \Pi_3^p$, the third level of the polynomial-time hierarchy.

[3] I did manage to prove an exponential lower bound, provided we restrict ourselves to linear combinations $\alpha|\psi\rangle + \beta|\varphi\rangle$ that are "manifestly orthogonal"—which means that for all computational basis states $|x\rangle$. either $\langle\psi|x\rangle = 0$ or $\langle\varphi|x\rangle = 0$.

Perhaps more relevant to physics, I also conjecture that 2-D and 3-D "cluster states" (informally, 2-D and 3-D lattices of qubits with pairwise nearest-neighbor interactions) have exponential tree size.[4] If true, this conjecture suggests that states with enormous tree sizes might have already been observed in condensed-matter experiments—for example, those of Ghosh et al. [4] on long-range entanglement in magnetic salts. In my personal fantasy land, once the evidence characterizing the ground states of these condensed-matter systems became undeniable, the skeptics would hit back with a *new* Sure/Shor separator. Then the experimentalists would try to refute *that* separator, and so on. As a result, what started out as a philosophical debate would gradually evolve into a scientific one—on which progress not only can be made, but is.

References

[1] S. Aaronson. Multilinear formulas and skepticism of quantum computing. In *Proc. ACM STOC*, pages 118–127, 2004. quant-ph/0311039.

[2] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005. quant-ph/0402095.

[3] M. Arndt, O. Nairz, J. Vos-Andreae, C. Keller, G. van der Zouw, and A. Zeilinger. Wave-particle duality of C_{60} molecules. *Nature*, 401:680–682, 1999.

[4] S. Ghosh, T. F. Rosenbaum, G. Aeppli, and S. N. Coppersmith. Entangled quantum state of magnetic dipoles. *Nature*, 425:48–51, 2003. cond-mat/0402456.

[5] O. Goldreich. On quantum computing. www.wisdom.weizmann.ac.il/~oded/on-qc.html, 2004.

[6] L. A. Levin. Polynomial time and extravagant models, in The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003. cs.CR/0012023.

[7] R. Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Proc. ACM STOC*, pages 633–641, 2004. ECCC TR03-067.

Short PCPs

Eli Ben-Sasson

(joint work with Oded Goldreich, Prahladh Harsha, Madhu Sudan, Salil Vadhan)

1. Efficient Verification of Proofs

Probabilistically Checkable Proof (PCP) systems [10, 2, 1] (also known as Holographic Proofs [3]) are proof systems that allow *efficient* probabilistic verification of proofs. Formally, a PCP system is given by a verifier, called a PCP verifier, that probabilistically queries a purported proof of a claimed theorem and accepts valid proofs of true theorems with probability one, while accepting any claimed proof of false assertions with low probability, say at most 1/2. In early works on this subject [3, 10, 2, 1], the notion of *efficiency* took on two different meanings.

[4] By contrast, I can show that 1-D cluster states have tree size $O\left(n^4\right)$.

- In the work of Babai et al. [3], which refer to inputs in error-correcting form, *efficient* verification meant the *running time* of the verifier is small (poly-logarithmic) and the *length* of the "Holographic" proof is not much larger than that of its classical analog (in [3], a classical proof of length n is converted to a Holographic proof of length $n^{1+\epsilon}$, for arbitrarily small $\epsilon > 0$). However, the *query complexity* of the verifier (i.e. the number of bits it reads from the proof) was only bounded by its running time.
- The work of Feige et al. [10] showed that obtaining a verifier with small *query complexity* yields hardness of approximation results. The PCP Theorem [2, 1] indeed showed the existence of verifiers with *constant* query complexity for any language in NP. Such query efficient proofs translate to strong non-approximability results for many combinatorial optimization problems (cf. [5, 4, 12, 11, 14]). However, in [10, 2, 1] and subsequent works, the running time of the PCP-verifier as well as the length of the "probabilistically checkable" proof were allowed to be arbitrary polynomials.

In this talk we describe recent research that shows one can obtain efficient verification under both interpretations. In other words, one gets PCP verifiers (say, for the NP-complete language 3SAT) that run in *poly-logarithmic* time and make a *constant number of queries* to a proof of *sub-polynomial* length. Notice we improve upon [3] in terms of proof-length, while matching the efficiency of [3] in running time and the efficiency of [1] in query complexity.

2. Results

We described the following two results from [8] and [7] respectively.

1. Constructions of probabilistically checkable proofs (PCPs) of length $n \cdot \mathrm{poly}(\log n)$ (to prove satisfiability of circuits of size n) that can verified by querying $\mathrm{poly}(\log n)$ bits of the proof. (Notice this result does not claim poly-logarithmic running time for the verifier). We also give constructions of locally testable codes (LTCs) with similar parameters.

   We pointed out that Dinur [9] recently showed (among other things) that the query complexity can be reduced to a *constant* while retaining the proof length at $n \cdot \mathrm{poly}(\log n)$. This result is obtained by applying her novel proof of the PCP Theorem (also presented in this workshop) to our result.
2. Every language in NP has a probabilistically checkable proof of proximity (i.e., proofs asserting that an instance is "close" to a member of the language), where the verifier's running time is poly-logarithmic in the input size and the length of the probabilistically checkable proof is only poly-logarithmically larger that the length of the classical proof. (Such a verifier can only query poly-logarithmically many bits of the input instance and the proof. Thus it needs oracle access to the input as well as the proof, and cannot guarantee that the input is in the language — only that it is close to some string in the language.) The time complexity of the verifier and the size of the proof were the original emphases in the definition of holographic

proofs, due to Babai et al. (STOC '91), and our work is the first to return to these emphases since their work.

3. Techniques

We focused on (sketching) the proof of the first result mentioned above. Previous constructions of short PCPs (from [3] to [6]) relied extensively on properties of *low* degree *multi*-variate polynomials. In contrast, our constructions rely on new problems and techniques revolving around the properties of codes based on *high* degree polynomials in *one* variable (also known as Reed-Solomon codes). We show how to convert the problem of verifying the satisfaction of a circuit by a given assignment to the task of verifying that a given function is close to being a Reed-Solomon codeword, i.e., a univariate polynomial of specified degree. This reduction is simpler than the corresponding steps in previous reductions, and gives a new alternative to using the popular "sum-check protocol". We then give a new PCP for the special task of proving that a function is close to being a Reed-Solomon codeword. This step of the construction is by a self-contained recursion, and the only ingredient needed in the analysis is the bi-variate low-degree test of Polischuk and Spielman [13].

Note that our constructions yield LTCs first, which are then converted to PCPs. In contrast, most recent constructions go in the opposite (and less natural) direction of getting LTCs from PCPs.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, *Proof verification and the hardness of approximation problems*, Journal of the ACM **45**, 3 (May 1998), 501–555.

[2] S. Arora, S. Safra, *Probabilistic checking of proofs: A new characterization of NP*, Journal of the ACM **45**, 1 (Jan. 1998), 70–122.

[3] L. Babai, L. Fortnow, L. A. Levin, M. Szegedy, *Checking computations in polylogarithmic time.* Proc. 23rd ACM Symp. on Theory of Computing (1991), 21–31.

[4] M. Bellare, O. Goldreich, M. Sudan, *Free bits, PCPs, and nonapproximability—towards tight results.* SIAM Journal of Computing **27**, (1998), 804–915.

[5] M. Bellare, S. Goldwasser, C. Lund, A. Russell *Efficient probabilistically checkable proofs and applications to approximation,* Proc. 25th ACM Symp. on Theory of Computing (1993), 294–304.

[6] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, S. Vadhan. *Robust PCPs of Proximity, Shorter PCPs and Applications to Coding.* Proc. 36th ACM Symp. on Theory of Computing, (2004), 1–10.

[7] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, S. Vadhan. *Short PCPs verifiable in polylogarithmic time*, Proc. 20th IEEE Conference on Computational Complexity, (2005), 120–134.

[8] E. Ben-Sasson, M. Sudan. *Simple PCPs with Poly-log Rate and Query Complexity*, Proc. 37th ACM Symposium on Theory of Computing (2005), 266–275.

[9] I. Dinur, *The PCP theorem by gap amplification.* Preliminary version at `http://eccc.uni-trier.de/eccc-reports/2005/TR05-046/index.html`

[10] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy. *Interactive proofs and the hardness of approximating cliques.* Journal of the ACM **43**, 2 (Mar. 1996), 268–292.

[11] V. Guruswami, D. Lewin, M. Sudan, L. Trevisan, *A tight characterization of NP with 3-query PCPs.* Proc. 39th IEEE Symp. on Foundations of Comp. Science (1998), 18–27.

[12] J. Håstad, *Some optimal inapproximability results.* Journal of the ACM **48**, (2001), 798–859.
[13] A. Polishchuk, D. A. Spielman, *Nearly-linear size holographic proofs.* Proc. 26th ACM Symp. on Theory of Computing (1994), 194–203.
[14] A. Samorodnitsky, L. Trevisan, *A PCP characterization of NP with optimal amortized query complexity.* Proc. 32nd ACM Symp. on Theory of Computing (2000), 191–199.

A Group-Theoretic Approach to Fast Matrix Multiplication

Chris Umans
(joint work with Henry Cohn, Robert Kleinberg, Balázs Szegedy)

The *exponent of matrix multiplication* is the smallest real number ω such that for all $\epsilon > 0$, $O(n^{\omega+\epsilon})$ arithmetic operations suffice to multiply two $n \times n$ matrices. The standard algorithm for matrix multiplication shows that $\omega \leq 3$. Strassen's remarkable result [5] shows that $\omega \leq 2.81\ldots$, and a sequence of further works culminating in the work of Coppersmith and Winograd [4] have improved this upper bound to $\omega \leq 2.376\ldots$ (see [1] for a full history). Most researchers believe that in fact $\omega = 2$, but there have been no further improvements in the known upper bounds for the past fifteen years.

In this talk we describe ongoing work on a new "group-theoretic" approach to matrix multiplication, recently proposed in [2]. The basic idea is to reduce matrix multiplication to group algebra multiplication with respect to a suitable non-abelian group. In the first part of the talk we describe this reduction together with a property of groups that is sufficient to admit such a reduction. We sketch a proof that an infinite family of groups admits such a reduction with parameters that are *necessary* (but not yet sufficient) to achieve $\omega = 2$. In the second part of the talk we describe further demands on the *representation theory* of the groups used in the reduction in order for the overall approach to yield non-trivial bounds on ω. We end by describing a specific group that proves $\omega < 2.908\ldots$ in this framework, and we speculate that generalizing this example may provide a route to proving $\omega = 2$.

Recall that by employing Strassen's framework for recursive matrix multiplication, any method for multiplying $k \times k$ matrices A and B that operates by (1) forming linear combinations of the entries of A and linear combinations of the entries of B, (2) multiplying m pairs of these sums, and (3) expressing the entries of the result matrix $C = AB$ as linear combinations of the m products, immediately yields $\omega \leq \log_k m$ (in Strassen's original algorithm $k = 2$ and $m = 7$).

Our method follows exactly this strategy. To describe it we need to recall the definition of the *group algebra* $\mathbb{C}[G]$; this is the set of all formal linear combinations of elements of group G, with addition and multiplication defined naturally on these formal sums (using the group multiplication law to multiply group elements). We often think of elements of $\mathbb{C}[G]$ as vectors of length G. The *Discrete Fourier Transform* (DFT) is a linear transform that turns group algebra multiplication into block-diagonal matrix multiplication, where the sizes of the blocks are the character degrees of G. Formally the DFT realizes the isomorphism

$$\mathbb{C}[G] \simeq \mathbb{C}^{d_1 \times d_1} \cdots \mathbb{C}^{d_\ell \times d_\ell},$$

where the d_i are the character degrees of G (and then, necessarily, $\sum_i d_i^2 = |G|$).

Multiplication of $k \times k$ matrices "embeds" into $\mathbb{C}[G]$ multiplication if there exist three subgroups $H_1, H_2, H_3 \subseteq G$ that satisfy the *triple product property*: for all $h_1 \in H_1, h_2 \in H_2, h_3 \in H_3$ we have

$$h_1 h_2 h_3 = 1 \Leftrightarrow h_1 = h_2 = h_3 = 1.$$

It can be verified that if we index the rows and columns of matrix A by elements of H_1 and H_2, and define $\bar{A} = \sum_{h_1 \in H_1, h_2 \in H_2} A_{h_1,h_2}(h_1 h_2^{-1})$; and if we index the rows and columns of matrix B by elements of H_2 and H_3, and define $\bar{B} = \sum_{h_2 \in H_2, h_3 \in H_3} A_{h_2,h_3}(h_2 h_3^{-1})$; then the coefficient on $(h_1 h_3^{-1})$ in the product $\bar{A}\bar{B}$ is exactly the (h_1, h_3) entry in the result matrix $C = AB$. Moreover, if we "multiply in the Fourier domain," i.e., we compute the DFT of $\bar{A}$, the DFT of $\bar{B}$, then perform the block-diagonal matrix multiplication, and finally compute the inverse DFT of the result, then we conform to the framework required for recursive matrix multiplication. The number m of multiplications required is the number of multiplications required for the block-diagonal matrix multiplication, which is roughly $\sum d_i^\omega$. Altogether we obtain the following theorem:

Theorem 1 (Cohn and Umans [2])**.** *Suppose that subgroups $H_1, H_2, H_3 \subseteq G$, each of size k, satisfy the triple product property, and let $d_1, d_2, \ldots, d_\ell$ be the character degrees of G. Then $k^\omega \leq \sum_i d_i^\omega$.*

Notice that $\sum_i d_i^2 = |G|$ is a lower bound on the right hand side of the above inequality, and thus to have a hope of proving $\omega = 2$, we need a family of groups G of size $k^{2+o(1)}$, each containing three subgroups of size k satisfying the triple product property. The following theorem shows that this is in fact possible:

Theorem 2 (Cohn and Umans [2])**.** *Let G_n be the symmetric group acting on $n(n+1)/2$ points arranged in a triangular array with sidelength n. Let H_1, H_2, H_3 be the three subgroups of G_n that preserve (set-wise) the rows of points parallel to each of the three sides, respectively. Then H_1, H_2, H_3 satisfy the triple product property in G_n, and $|H_1| = |H_2| = |H_3| = |G_n|^{1/2-o(1)}$.*

Unfortunately when plugged into Theorem 1 this family of groups does not even yield $\omega < 3$, because the character degrees d_i are too large. The challenge thus becomes to find a family of groups together with subgroups satisfying the triple product property *and* for which the character degrees are small enough for Theorem 1 to yield nontrivial bounds on ω.

How small is "small enough"? One corollary of Theorem 1 is that if for some family of groups admitting $k \times k$ matrix multiplication with $|G| = k^{2+o(1)}$, we have the maximum character degree $d_{\max} \leq |G|^\gamma$ for some constant $\gamma < 1/2$, then $\omega = 2$. Since *a priori* $d_{\max} < |G|^{1/2}$ for any group G, this seems like it may be within reach. A second corollary of Theorem 1 is that this method proves $\omega < 3$

iff we can find a group G admitting $k \times k$ matrix multiplication via H_1, H_2, H_3, and for which

$$|H_1||H_2||H_3| > \sum_i d_i^3.$$

In fact we can even relax the requirement that H_1, H_2, H_3 are subgroups, and instead allow *subsets* of G. If $Q(H_i)$ denotes the set of (right-) quotients of pairs of elements from H_i, then the condition in the triple product property becomes $q_1q_2q_3 = 1 \Leftrightarrow q_1 = q_2 = q_3$, where $q_i \in Q(H_i)$.

In [3] we construct a group and three subsets that "beat the sum of the cubes" of the character degrees, and thus prove a non-trivial bound on ω. The construction is as follows. Let A be any abelian group of size m, and consider the semidirect product of $C_2 = \{1, z\}$ (the cyclic group of order 2) with A^6, where z acts by interchanging the first three and last three coordinates; i.e., if $(a, b, c, d, e, f) \in A^6$, then $z(a, b, c, d, e, f)z = (d, e, f, a, b, c)$. The three subsets we consider are:

$$\begin{aligned} F &= \{(a, 0, 0, 0, a', 0)z^i : a, a' \in A, a \neq 0, i \in \{0, 1\}\} \\ G &= \{(0, b, 0, 0, 0, b')z^j : b, b' \in A, b \neq 0, j \in \{0, 1\}\} \\ H &= \{(0, 0, c, c', 0, 0)z^k : c, c' \in A, c \neq 0, k \in \{0, 1\}\} \end{aligned}$$

In [3] we prove that these subsets satisfy the triple product property. The size of each of the subsets is $2m(m-1)$, so $|F||G||H| = 8m^3(m-1)^3$. It is easy to see that the containing group has $d_{\max} = 2$, and thus

$$\sum_i d_i^3 \leq d_{\max} \sum_i d_i^2 = d_{\max}(2m^6) = 4m^6,$$

which for sufficiently large m is exceeded by $|F||G||H|$. As argued above such a construction proves nontrivial bounds on ω and in fact taking $m = 17$ yields $\omega < 2.908\ldots$.

This group can also be described as the *wreath product* of the symmetric group of order 2 with the abelian group A^3. By generalizing this construction in different ways, we can prove better bounds ($\omega < 2.48\ldots$ and $\omega < 2.41\ldots$) using wreath products of the full symmetric group with abelian groups. The difficult part in such constructions seems to be apportioning the abelian part among the three subsets in a way that ensures that the triple product property holds. In [3], we make two conjectures regarding this apportionment that would improve our existing constructions to the point that they would yield $\omega = 2$.

References

[1] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1997.

[2] H. Cohn and C. Umans. A Group-theoretic Approach to Fast Matrix Multiplication. Proceedings of the 44th Annual Symposium on Foundations of Computer Science, 11–14 October 2003, Cambridge, MA, IEEE Computer Society, pp. 438–449, `arXiv:math.GR/0307321`.

[3] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans. Group-theoretic Algorithms for Matrix Multiplication. To appear in FOCS 2005.

[4] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Computation*, 9:251–280, 1990.
[5] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

On Basing One-Way Functions on NP-Hardness

Adi Akavia
(joint work with Oded Goldreich, Shafi Goldwasser, Dana Moshkovitz)

One-way functions are functions that are easy to compute but hard to invert, where the hardness condition refers to the average-case complexity of the inverting task. The existence of one-way functions is the cornerstone of modern cryptography: almost all cryptographic primitives imply the existence of one-way functions, and most of them can be constructed based either on the existence of one-way functions or on related (but seemingly stronger) versions of this assumption.

As noted above, the hardness condition of one-way functions is an average-case complexity condition. Clearly, this average-case hardness condition implies a worst-case hardness condition; that is, the existence of one-way functions implies that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$. A puzzling question of fundamental nature is whether or not the necessary worst-case condition is a sufficient one; that is, can one base the existence of one-way functions on the assumption that $\mathcal{NP}$ is not contained in $\mathcal{BPP}$.

More than two decades ago, Brassard [2] observed that the inverting task associated with a one-way *permutation* (or, more generally, a 1-1 one-way function) cannot be $\mathcal{NP}$-hard, unless $\mathcal{NP} = \text{co}\mathcal{NP}$. The question was further addressed (indirectly), in the works of Feigenbaum and Fortnow [3] and Bogdanov and Trevisan [1], which focused on the study of worst-case to average-case reductions among decision problems.

Our Main Results. In this work we re-visit the aforementioned question, but do so explicitly. We study possible reductions from a worst-case decision problem to the task of average-case inverting a polynomial-time computable function (i.e., reductions that are supposed to establish that the latter function is one-way based on a worst-case assumption regarding the decision problem). Specifically, we consider (randomized) reductions of $\mathcal{NP}$ to the task of average-case inverting a polynomial-time computable function f, and capitalize on the additional "computational structure" of the search problem associated with the inverting task. This allows us to strengthen previously known negative results, and obtain the following two main results:

1. If given y one can efficiently compute $|f^{-1}(y)|$ then the existence of a (randomized) reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq \text{co}\mathcal{AM}$.

   The result extends to functions for which the preimage size is efficiently verifiable via an AM protocol. For example, this includes regular functions with efficiently recognizable range. Recall that $\mathcal{AM}$ is the class of sets having two-round interactive proof systems, and that it is widely believed that co$\mathcal{NP}$ is not contained in $\mathcal{AM}$ (equiv., $\mathcal{NP}$ is not contained in co$\mathcal{AM}$). Thus, it follows that such reductions cannot exist (unless $\mathcal{NP} \subseteq$ co$\mathcal{AM}$).

We stress that this result holds for any reduction, including *adaptive* ones. We note that the previously known negative results regarding worst-case to average-case reductions were essentially confined to *non-adaptive* reductions (cf. [3, 1], where [3] also handles restricted levels of adaptivity).

2. For any (polynomial-time computable) function f, the existence of a (randomized) *non-adaptive* reduction of $\mathcal{NP}$ to the task of average-case inverting f implies that $\mathcal{NP} \subseteq$ co$\mathcal{AM}$.

   This result improves over the previous negative results of [3, 1] that placed $\mathcal{NP}$ in non-uniform co$\mathcal{AM}$ (instead of in *uniform* co$\mathcal{AM}$).

These negative results can be interpreted in several ways. The straightforward view is that they narrow down the means by which one can base one-way functions on $\mathcal{NP}$-hardness. Namely, under the assumption that $\mathcal{NP}$ is not contained in co$\mathcal{AM}$, these results show that (1) *non-adaptive* randomized reductions are not suitable for basing one-way functions on $\mathcal{NP}$-hardness, and (2) that one-way functions based on $\mathcal{NP}$-hardness can not have efficient algorithms for computing (or, more generally, verifying) the preimage size. Another interpretation is that these negative results are an indication that (worst-case) complexity assumptions regarding $\mathcal{NP}$ as a whole (i.e., $\mathcal{NP} \not\subseteq \mathcal{BPP}$) are not sufficient to base one-way functions on. But this does not rule out the possibility of basing one-way functions on the worst-case hardness of a subclass of $\mathcal{NP}$ (e.g., the conjecture that $\mathcal{NP} \cap$ co$\mathcal{NP} \not\subseteq \mathcal{BPP}$). Yet another interpretation is that these negative results suggest that we should turn to "non black-box" reductions for basing one-way functions on $\mathcal{NP}$-hardness.

Relation to Feigenbaum-Fortnow and Bogdanov-Trevisan. Our work builds on the previous works of Feigenbaum and Fortnow [3] and Bogdanov and Trevisan [1], while capitalizing on the additional "computational structure" of the search problem associated with the task of inverting polynomial-time computable functions. We believe that our results illustrate the gain of directly studying the context of one-way functions rather than inferring results for it from a the general study of worst-case to average-case reductions.

Although a main motivation of [1] is the question of basing one-way functions on worst-case $\mathcal{NP}$-hardness, its focus (like that of [3]) is on *decision problems*. Using known reductions between search and decision problems, Bogdanov and Trevisan [1] also derive implications on the (im)possibility of basing one-way functions on $\mathcal{NP}$-hardness. In particular, they conclude that if there exists an $\mathcal{NP}$-complete set for which deciding any instance is *non-adaptively* reducible to *inverting a one-way function* (or, more generally, to a search problem with respect to a sampleable distribution), then co$\mathcal{NP} \subseteq \mathcal{AM}_{\text{poly}}$.

The works [1, 3] fall short of a general impossibility result in two ways. First, they only consider *non-adaptive* reductions, whereas Ajtai's celebrated worst-case to average-case reductions of lattice problems are adaptive. Second, [1, 3] reach conclusions involving a *non-uniform* complexity class (i.e., $\mathcal{AM}_{\rm poly}$). Non-uniformity seems an artifact of their techniques, and one may hope to conclude that co$\mathcal{NP} \subseteq \mathcal{AM}$ rather than co$\mathcal{NP} \subseteq \mathcal{AM}_{\rm poly}$. (One consequence of the uniform conclusion is that it implies that the polynomial time hierarchy collapses to the second level, whereas the non-uniform conclusion only implies a collapse to the third level.)

The Benefits of Direct Study of One-Way Functions. Working directly with one-way functions allows us to remove both the aforementioned shortcomings. That is, we get rid of the non-uniformity altogether, and obtain a meaningful negative result for the case of general (adaptive) reductions. Specifically, working directly with one-way functions allows us to consider natural special cases of potential one-way functions, which we treat for general (i.e., possibly adaptive) reductions. One special case of potential one-way functions is that of *regular* one-way functions. Loosely speaking, in such a function f, each image of f has a number of preimages that is (easily) determined by the length of the image. We prove that any reduction (which may be *fully adaptive*) of $\mathcal{NP}$ to inverting a regular polynomial-time computable function that has an efficiently recognizable range (possibly via an AM-protocol) implies co$\mathcal{NP} \subseteq \mathcal{AM}$. More generally, this holds for any function f for which there is an AM-protocol for determining the number of inverses $|f^{-1}(y)|$ of each given y. We call such functions size-verifiable, and note that they contain all functions for which (given y) one can efficiently compute $|f^{-1}(y)|$.

As stated above, we believe that the study of the possibility of basing one-way functions on worst-case $\mathcal{NP}$-hardness is the most important motivation for the study of worst-case to average-case reductions for $\mathcal{NP}$. In such a case, one should consider the possible gain from studying the former question directly, rather than as a special case of a more general study. We believe that the results presented in this work indicate such gains. Firstly, working directly in the context of one-way function enabled us to get rid of the non-uniformity in all our results (by replacing non-uniform advice that provide needed statistics with AM-protocols designed to provide these statistics). Secondly, the context of one-way function enabled us to consider meaningful types of one-way functions and to establish even stronger results for them. We hope that this framework may lead to resolving the general question of the possibility of basing *any* one-way function on worst-case $\mathcal{NP}$-hardness via *any* reduction. In light of the results of this paper, we are tempted to conjecture an impossibility result (pending, as usual, on co$\mathcal{NP} \not\subseteq \mathcal{AM}$).

References

[1] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for NP problems. In *Proc. 44th IEEE Symposium on Foundations of Computer Science*, pages 308–317, 2003.

[2] G. Brassard. Relativized Cryptography. In *20th IEEE Symposium on Foundations of Computer Science*, pages 383–391, 1979.
[3] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993. Extended Abstract appeared in Proc. of IEEE Structures'91.

Hardness of Undirected Routing Problems

Julia Chuzhoy

In this talk we present several recent hardness of approximation results for undirected routing problems, with a focus on the Edge Disjoint Paths and related problems.

In general, the input to a routing problem consists of a graph G (directed or undirected), and a number of source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$ that need to be connected. In the Edge Disjoint Paths problem (EDP), the objective is to connect as many pairs as possible via edge-disjoint paths.

The best approximation algorithm for EDP in directed graphs has a ratio of $O(\min(n^{2/3}, \sqrt{m}))$ [12, 5, 13] where n and m denote the number of vertices and edges respectively in the input graph. This upper bound is matched by an $\Omega(m^{1/2-\epsilon})$-hardness due to Guruswami *et al.* [10]. Therefore, the directed version of the problem is quite well-understood. However, this is not the case with undirected graphs, for which the problem is still widely open. The best current upper bound is $O(\sqrt{n})$ [6], while on the negative side, until recently, only APX-hardness has been known.

A related routing problem is Congestion Minimization. The input to this problem is exactly the same as in the EDP problem, namely graph G and a collection of source-sink pairs. The goal is to connect **each** source to its sink by a single path, such that the edge congestion is minimized, where edge congestion is the maximum number of paths sharing an edge. For this problem, Raghavan and Thompson's randomized rounding technique [14] gives an $O(\frac{\log n}{\log \log n})$-approximation algorithm for both directed and undirected versions. When the input graph is directed, an $\Omega(\log \log n)$-hardness was proved by Chuzhoy and Naor [8]. However, until recently no non-trivial lower bounds were known for the undirected version.

The last few years have seen a significant progress in understanding the hardness of undirected routing problems. In particular, Andrews [1] introduced a new approach for proving hardness of undirected routing problems, and showed $\Omega(\log^{1/2-\epsilon})$-hardness of the Buy-at-Bulk problem. Following [1], Andrews and Zhang [3] proved $\Omega(\log \log^{1-\epsilon} n)$ hardness of undirected Congestion Minimization. Andrews and Zhang [2] also showed that undirected EDP is $\Omega(\log^{1/3-\epsilon} n)$-hard to approximate. This result was recently improved to $\Omega(\log^{1/2-\epsilon})$-hardness by Chuzhoy and Khanna [7].

We demonstrate this new approach on the hardness of Edge Disjoint Paths problem. Consider the following reduction from the Maximum Independent Set problem (MIS) to EDP: for each vertex in the MIS instance, we create a source-sink pair and a *canonical path* that connects this pair. The canonical paths are defined in such a way that whenever there is an edge between vertices u and v in the MIS instance, the two corresponding canonical paths share an edge. It is easy to see that if the solution to the resulting EDP instance consists of canonical paths only, then it can be translated into a solution of the MIS instance of the same cost. The opposite is also true: any solution to the MIS instance naturally defines a solution to the EDP instance. The problem is that in general, solutions of the EDP instance do not necessarily follow the canonical paths, and if such a solution has many non-canonical paths, then it cannot be translated into a large cardinality independent set. The main idea is to convert the above EDP instance into a random graph with "almost" high girth. Roughly speaking, in order to create the random instance, we make many copies of each vertex from the original EDP instance. Each edge in the original EDP instance is then replaced by a random matching between the copies of its endpoints. In the new instance, we can bound the number of non-canonical paths in any solution as follows: the number of long non-canonical paths is restricted due to the graph capacity. As for the short non-canonical paths, each such path forms a small cycle with some canonical path. The number of such small cycles can be bounded due to the random structure of the graph.

An interesting variation of the EDP and the Congestion Minimization problems is EDP with congestion. In this problem, we are given an input graph G, a collection of source-sink pairs $(s_1,t_1),\ldots,(s_k,t_k)$, and an integer c. The goal is to route maximum number of $s-t$ pairs, while the congestion on any edge is at most c. We are interested in a bi-criteria setting here, where the optimal solution uses edge disjoint paths only, while the algorithm is allowed congestion up to c. Since the Edge Disjoint Paths problem seems to be hard to approximate, it is interesting whether a better approximation can be found for its natural relaxation, namely EDP with congestion. Recently, an $\Omega(\log^{\frac{1-\epsilon}{c+2}})$-hardness was proved independently by [7, 4, 11]. We present the construction of [7] in this talk.

Finally, we study the multicommodity flow relaxation of the Edge Disjoint Paths problem. It is known that the linear program has integrality gap of $\Omega\sqrt{\log n}$ [9]. However, even for $c = 2$, so far no superconstant lower bounds on the integrality gap of the multicommodity flow relaxation has been known. The hardness results of [7, 4, 11] naturally show that the integrality gap is at least $\Omega(\log^{\frac{1-\epsilon}{c+2}})$. However, the constructions are unnecessarily complex. In this talk we describe a direct simple construction of $\Omega\left(\left(\frac{\log n}{\log\log^2 n}\right)^{1/(c+1)}/c^2\right)$ integrality gap due to [7].

References

[1] M. Andrews, *Hardness of Buy-at-bulk Network Design*, Proc. of FOCS, 2004.

[2] M. Andrews and L. Zhang, *Hardness of the Undirected Edge-Disjoint Paths Problem*, Proc. of STOC, 2005.

[3] M. Andrews and L. Zhang, *Hardness of the undirected congestion minimization problem*, Proc. of STOC. 2005.

[4] M. Andrews and L. Zhang, *Hardness of Edge-Disjoint Paths with Congestion*, manuscript, 2005.

[5] C. Chekuri and S. Khanna, *Edge Disjoint Paths Revisited*, Proc. of SODA, 2003.

[6] C. Chekuri, S. Khanna and F. B. Shepherd, personal communication.

[7] J. Chuzhoy and S. Khanna, *New Hardness Results for Undirected Edge Disjoint Paths*, manuscript, 2005.

[8] J. Chuzhoy and J. Naor, *New inapproximability results for congestion minimization and machine scheduling*, Proc. of STOC, 2004.

[9] N. Garg, V. Vazirani, M. Yannakakis. *Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees*, Algorithmica, **18(1)** (1997), 3-20. Preliminary version appeared in Proc. of ICALP, (1993).

[10] V. Guruswami, S. Khanna, R. Rajaraman, F. B. Shepherd, and M. Yannakakis, *Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems*, To appear in JCSS. Preliminary version appeared in Proc. of STOC, 1999.

[11] V. Guruswami, K. Talwar, personal communication.

[12] J. M. Kleinberg, *Approximation algorithms for disjoint paths problems*, PhD thesis, MIT, Cambridge, MA, May 1996.

[13] K. Varadarajan and G. Venkataraman, *Graph Decomposition and a Greedy Algorithm for Edge-disjoint Paths*, Proc. of SODA, 2004.

[14] P. Raghavan and C. D. Thompson. *Randomized rounding: A technique for provably good algorithms and algorithmic proofs*, Combinatorica, **7** (1987), 365–374.

Quantum Information and the PCP Theorem

Ran Raz

We present the recent paper: "Quantum Information and the PCP Theorem" [1].

Probabilistic Checkable Proofs with an additional quantum witness. Our main result is that the membership $x \in SAT$ (for x of length n) can be proved by a combination of the following two witnesses:

1. A logarithmic-size quantum witness
2. A polynomial-size classical witness consisting of blocks of length $polylog(n)$ bits each, s.t. only one of these blocks is read by the verifier

In other words, after seeing the logarithmic-size quantum witness the verifier only needs to read **one** of the blocks of the classical witness in order to verify the membership $x \in SAT$.

Interactive proofs with quantum advice. We also study the power of interactive proofs with quantum advice and we show that the class $QIP/qpoly$ contains **all** languages. That is, for any language L (even non-recursive), the membership $x \in L$ (for x of length n) can be proved by a polynomial-size quantum interactive proof, where the verifier is a polynomial-size quantum circuit with working space initiated with some quantum state $|\Psi_{L,n}\rangle$ (depending only on L and n).

Moreover, the interactive proof that we give is of only one round, and the messages communicated are classical.

Interactive proofs with randomized advice. The quantum advice in the last result can be replaced by a randomized advice. The last result can hence be presented as a classical result.

Our protocol shows that the class $IP/rpoly$ contains all languages. That is, for any language L, the membership $x \in L$ (for x of length n) can be proved by a polynomial-size classical interactive proof, where the verifier is a polynomial-size circuit with working space initiated with a random string chosen from some distribution $D_{L,n}$ (depending only on L and n). Moreover, the interactive proof that we give is of only one round.

It is important to note that the classical result only holds if the setting is such that the prover (of the interactive proof) cannot see the advice that was given to the verifier. In other words, the result holds only if the class $IP/rpoly$ is defined with an advice that is kept as a secret from the prover.

Representation of classical bits by a quantum or random string. Both of the above results are based on a new representation of an exponential number of classical bits by a short quantum or random string.

We show how to encode 2^n (classical) bits $a_1, ..., a_{2^n}$ by a single quantum state $|\Psi\rangle$ of size $O(n)$ qubits, such that: for any constant k and any $i_1, ..., i_k \in \{1, ..., 2^n\}$, the values of the bits $a_{i_1}, ..., a_{i_k}$ can be retrieved from $|\Psi\rangle$ by a one-round Arthur-Merlin interactive protocol of size polynomial in n. This shows how to go around Holevo-Nayak's Theorem, using Arthur-Merlin proofs.

As before, the quantum advice in the last result can be replaced by a randomized advice. The last result can hence be presented as a classical result.

Our protocol hence shows how to encode 2^n (classical) bits $a_1, ..., a_{2^n}$ by a single random string ρ of size $O(n)$, such that: for any constant k and any $i_1, ..., i_k \in \{1, ..., 2^n\}$, the values of the bits $a_{i_1}, ..., a_{i_k}$ can be retrieved from ρ by a one-round Arthur-Merlin interactive protocol of size polynomial in n.

As before, the classical result only holds if the setting is such that the string ρ is kept as a secret from the prover.

A quantum low degree test. Our main result also relies on a new machinery of *quantum low-degree-test* that may be interesting in its own right. Technically, this is the hardest part of the paper.

References

[1] R. Raz. *Quantum Information and the PCP Theorem*, Mansucript (2005)

The Computational Complexity of the Euler characteristic and the Hilbert Polynomial

Peter Bürgisser
(joint work with Felipe Cucker and Martin Lotz)

The talk presented results from [10, 11, 12].

Motivation. The *Euler characteristic* $\chi(V)$ of a topological space V is one of the most basic invariants in algebraic topology and occurs in many branches of geometry. Remarkably, it can be characterized in various different ways. For instance, for spaces V admitting a finite triangulation, it is the alternating sum of the number of i-simplices of the triangulation. The *Hilbert polynomial* is an important discrete object attached to a complex projective variety $V \subseteq \mathbb{P}^n$. Among other things, it encodes the dimension, the degree and the arithmetic genus of V.

Previous Work. S. Basu [3] gave the first single exponential time algorithm for the computation of the Euler characteristic of a semialgebraic set. Algorithms for computing Hilbert polynomials were described in [17, 6, 5]. These algorithms are based on the computation of Gröbner bases, which leads to bad upper complexity estimates. Currently, no upper bound better than exponential space is known for the computation of the Hilbert polynomial of a projective variety.

In [9] a systematic study of the inherent complexity of computing algebraic or topological invariants of (semi)algebraic sets was initiated, with the goal of characterizing the complexity of various such problems by completeness results in a suitable hierarchy of complexity classes. Versions of L. Valiant's counting complexity class #P [18], tailored to the Blum-Shub-Smale model of computation [8], turned out to be relevant for this purpose. Over the reals, such a counting class was first introduced by K. Meer [16].

For instance, the problem $\#\mathrm{HN}_{\mathbb{C}}$ of counting the number of complex common zeros of a finite set of multivariate polynomials is complete for the counting class $\#\mathrm{P}_{\mathbb{C}}$ over $\mathbb{C}$. One of the results of [9] states that the computation of the modified Euler characteristic of a semialgebraic set is polynomial time equivalent to the problem of counting the number of real common zeros of a multivariate polynomial.

Our results. We show that the problem $\mathrm{EULER}_{\mathbb{C}}$ of computing the topological Euler characteristic of a complex algebraic variety is polynomial time equivalent to the problem $\#\mathrm{HN}_{\mathbb{C}}$. Moreover, we prove that the problem $\textsc{Hilbert}_{\mathrm{sm}}$ of computing the Hilbert polynomial of a smooth equidimensional complex projective variety can be reduced in polynomial time to the problem $\#\mathrm{HN}_{\mathbb{C}}$. We can prove analogous statements in the Turing model of computation. Finally, we show that the more general problem of computing the Hilbert polynomial of a homogeneous ideal is polynomial space hard. This implies polynomial space lower bounds for both the problems of computing the rank and the Euler characteristic of cohomology groups of coherent sheaves on projective space as well as for the problem of computing the corresponding Euler characteristic, thus improving the #P-lower bound in E. Bach [2].

Proof Ideas. The class $\#\mathrm{P}_{\mathbb{C}}$ captures the complexity of counting the number of complex solutions to systems of polynomial equations. It is therefore not surprising that some of the ideas and tools of intersection theory, enumerative geometry, and Schubert calculus are salient for our purposes.

A first ingredient of our proofs is a complexity framework for analyzing general position arguments (generic parsimonious reductions). Efficient algorithms for quantifier elimination elimination over $\mathbb{R}$ are essential in this context, see [4].

The reduction from $\mathrm{EULER}_{\mathbb{C}}$ to $\mathrm{HN}_{\mathbb{C}}$ crucially depends on a recent result due to P. Aluffi [1]. This result characterizes the Euler characteristic of a (possibly singular) projective hypersurface $Z(f)$ in terms of the multidegrees of the projective gradient map of f. Our reduction from $\textsc{Hilbert}_{\mathrm{sm}}$ to $\#\mathrm{HN}_{\mathbb{C}}$ consists of the following three steps:

1. We interpret the value $p_V(d)$ of the Hilbert polynomial of $V \subseteq \mathbb{P}^n$ on $d \in \mathbb{Z}$ as the Euler characteristic $\chi(\mathcal{O}_V(d))$ of the twisted sheaf $\mathcal{O}_V(d)$.
2. The Hirzebruch-Riemann-Roch Theorem [13] gives an explicit combinatorial description of $\chi(\mathcal{O}_V(d))$ in terms of certain determinants $\Delta_\lambda(c)$ (related to Schur polynomials) in the Chern classes c_i of the tangent bundle of V.
3. The homology class corresponding to the cohomology class $\Delta_\lambda(c)$ can be realized up to sign by a degeneracy locus, which is defined as the pullback of a Schubert variety under the Gauss map [14].

We call the geometric degree of such a degeneracy locus a projective character. The above observation allows to express the coefficients of the Hilbert polynomial as rational linear combinations of projective characters. We now use the fact that the computation of the geometric degree of varieties is essentially possible in the complexity class $\#\mathrm{P}_{\mathbb{C}}$, and that the class $\#\mathrm{P}_{\mathbb{C}}$ is closed under exponential summation.

References

[1] P. Aluffi, *Computing characteristic classes of projective schemes*, J. Symbolic Comput. **35(1)** (2003), 3–19.

[2] E. Bach, *Sheaf cohomology is #P-hard*, J. Symbolic Comput. **27(4)** (1999), 429–433.

[3] S. Basu, *On bounding the Betti numbers and computing the Euler characteristic of semi-algebraic sets*, Discrete Comput. Geom. **22** (1999), 1–18.

[4] S. Basu, R. Pollack and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics **10** (2003), Springer Verlag.

[5] D. Bayer and M. Stillman, *Computation of Hilbert functions*, J. Symb. Comp. **14** (1992), 31–50.

[6] A.M. Bigatti, M. Caboara, and L. Robbiano, *On the computation of Hilbert-Poincaré series*, Appl. Algebra Engrg. Comm. Comput. **2(1)** (1991), 21–33.

[7] D. Bayer and D. Mumford, *What can be computed in algebraic geometry?*, In *Computational algebraic geometry and commutative algebra (Cortona, 1991)*, Sympos. Math., XXXIV, pages 1–48. Cambridge Univ. Press, Cambridge, 1993.

[8] L. Blum and M. Shub and S. Smale, *On a theory of computation and complexity over the real numbers*, Bulletin of the AMS **21** (1989), 1–46.

[9] P. Bürgisser and F. Cucker, *Counting complexity classes for numeric computations II: Algebraic and semialgebraic sets*, Proc. 36th Ann. ACM STOC (2004), 475–485. Full version at `http://www.arxiv.org/abs/cs/cs.CC/0312007`.

[10] P. Bürgisser, F. Cucker, M. Lotz, *The Complexity to Compute the Euler Characteristic of Complex Varieties*, Comptes rendus de l'Académie des sciences Paris, Ser. I **339** (2004), 371–376.

[11] P. Bürgisser, F. Cucker, M. Lotz, *Counting Complexity Classes for Numeric Computations III: Complex Projective Sets*, Foundations of Computational Mathematics, to appear.

[12] P. Bürgisser, M. Lotz, *The complexity of computing the Hilbert polynomial of smooth equidimensional complex projective varieties*, `http://www.arxiv.org/abs/cs/cs.CC/0502044`.

[13] F. Hirzebruch, *New Topological Methods in Algebraic Geometry*, Die Grundlehren der Mathematischen Wissenschaften **131** (1966), Springer Verlag.

[14] G. Kempf and D. Laksov, *The determinantal formula of Schubert calculus*, Acta Math. **132** (1974), 153–162.

[15] E.W. Mayr, *Some Complexity Results for Polynomial Ideals*, J. Compl. **13** (1997), 303–325.

[16] K. Meer, *Counting problems over the reals*, Theoretical Computer Science **242** (2000), 41–58.

[17] F. Mora and H.M. Möller, *The computation of the Hilbert function*, Proc. EUROCAL, Lecture Notes in Computer Science **162** (1983), Springer, 157–167.

[18] L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science **8** (1979), 189–201.

Simulating Independence: New Constructions of Condensers, Ramsey Graphs, Dispersers, and Extractors

Ronen Shaltiel
(joint work with Boaz Barak, Guy Kindler, Benny Sudakov, Avi Wigderson)

A distribution X over binary strings of length n has min-entropy k if every string has probability at most 2^{-k} in X. We say that X is a δ-source if its rate k/n is at least δ. In this work we continue a long line of research concerned with "extracting randomness from high entropy distributions" (see survey article [8]).

Main results. We give the following new explicit constructions (namely, poly(n)-time computable functions) of *deterministic* extractors, dispersers and related objects. All work for any fixed rate $\delta > 0$. No previous explicit construction was known for either of these, for any $\delta < 1/2$. The first two constitute major progress to very long-standing open problems.

1. **Bipartite Ramsey graph** $Ramsey : (\{0,1\}^n)2 \to \{0,1\}$, such that for any two independent δ-sources X_1, X_2 we have $Ramsey(X_1, X_2) = \{0,1\}$.
   A corollary is a new explicit construction of bipartite Ramsey graphs. That is, a 2-coloring of the edges of the complete N by N bipartite graph with $N = 2^n$, such that no induced N^δ by N^δ subgraph is monochromatic. This improves a previous construction by [5] which achieves $\delta = 1/2 - 1/\sqrt{n}$.
2. **Three source extraction** $Ext : (\{0,1\}^n)3 \to \{0,1\}$, such that for any three independent δ-sources X_1, X_2, X_3 we have that $Ext(X_1, X_2, X_3)$ is ($o(1)$-close to being) an unbiased random bit.
   This result improves previous results by [1] that requires $O(1/\delta 2)$ sources (although that result achieves smaller error). The aforementioned construction is used as a componenet in our constructions together with incomparable results by [2, 9, 3] that for $\delta > 1/2$ require only two sources.
3. **Constant seed condenser**[1] $Con : \{0,1\}^n \to (\{0,1\}^m)^c$, such that for any δ-source X, one of the c output distributions $Con(X)_i$, is a 0.9-source over $\{0,1\}^m$. Here c is a constant depending only on δ.

In the rest of this abstract, we provide an overview of our techniques.

A constant seed condenser. Our basic condenser $\mathtt{bcon}$ will take strings of length n with $n = 3p$ for some prime p. For every $x \in \{0,1\}^n$ let $x = x_1x_2x_3$ its natural partition to three length p blocks. Define $\mathtt{bcon} : \{0,1\}^{3p} \to (\{0,1\}^p)4$ by $\mathtt{bcon}(x) = x_1, x_2, x_3, x_1 \cdot x_2 + x_3$ (with arithmetic in $GF(2^p)$).

We prove that if X is a δ-source with $\delta < 0.9$, then at least one of the output blocks is a $(\delta + \Omega(\delta 2))$-source. Iterating it a constant number of times on a δ-source allows us to increase to rate (of some output block) above 0.9 and achieve the aforementioned result.

The proof heavily relies on the main lemma of [1], who proved $x_1 \cdot x_2 + x_3$ is condensed *assuming* that the x_i's are *independent*. We certainly *cannot* assume

[1] This result was also independently obtained by Ran Raz.

that in our case, as X is a general source. Still, we use that lemma to show that if none of these first 3 blocks is more condensed than the input source, then they are "independent enough" for using that main lemma.

A 2-source constant-seed/"somewhere" extractor. Our two main deterministic constructions in this paper are a 3-source extractor and a bipartite Ramsey. For both, an essential building block, is a constant seed 2-source extractor `s_ext` (short for "somewhere extractor") for constant entropy rate, which we describe next.

What we prove is that for every $\delta > 0$ there are integers c, d and a poly(n)-time computable function $\mathtt{s\_ext} : (\{0,1\}^n)2 \to (\{0,1\}^{n/c})^d$, such that for every two δ-sources X_1, X_2 there is at least one output block $\mathtt{s\_ext}(X_1, X_2)_i$ which is (exponentially close to) uniform.

Constructing the somewhere extractor `s_ext` is simple, given the condenser `con` of the previous subsection. To compute $\mathtt{s\_ext}(X_1, X_2)$, compute the output blocks of $\mathtt{con}(X_1)$ and $\mathtt{con}(X_2)$. By definition, some output block of each has rate $> .9$. We don't know which, but we can try all pairs! For each pair we apply a 2-source extractor which expects its sources to have entropy rate $> 1/2$. (Recall that there are such explicit constructions). We obtain a constant number of linear length blocks, one of which is very close to uniform. Formally, if d is the number of output block of `con`, then `s_ext` will produce $d2$ blocks, with $\mathtt{s\_ext}(X_1, X_2)_{i,j} =$ 2-src-ext$(\mathtt{con}(X_1)_i, \mathtt{con}(X_2)_j)$.

A 4-source extractor. In the paper we construct a 3-source extractor. For the purpose of explaining some of the ideas in the construction it is easier to show a 4-source extractor. Recall that our 2-source *somewhere* extractor `s_ext` produces a constant number (say) d of linear length output blocks, one of which is random. First we note that producing shorter output blocks maintains this property as a prefix of a random string is random.

Let us indeed output only a constant b bits in every block (satisfying $b \geq \log(db)$). Concatenating all output blocks of this $\mathtt{s\_ext}(X_1, X_2)$ gives us a distribution (say Z_1) on db bits with min-entropy $\geq b$. If we have 4 sources, we can get another independent such distribution Z_2 from $\mathtt{s\_ext}(X_3, X_4)$. But note that these are two independent distributions on a constant number of bits with sufficient min-entropy for (existential) 2-source extraction. Now apply an optimal (non-constructive) 2-source extractor on Z_1, Z_2 to get a uniform bit; as db is only a constant, such an extractor exists by the probabilistic method and can be found in constant time by brute-force search! We denote it by `opt`. To sum up, our 4-source extractor is

$$\mathtt{4ext}((X_1, X_2); (X_3, X_4)) = \mathtt{opt}(\mathtt{s\_ext}(X_1, X_2), \mathtt{s\_ext}(X_3, X_4))$$

The Construction of bipartite Ramsey graphs. This construction uses the components above but is significantly more complicated. In the next paragraph we try to highlight some (but not all) of the ideas that are used.

We first observe that our 4-source extractor can extract randomness even when given two independent block-wise sources.[2] We then use methods from [7] to show that for any source X there exists a way to partition it into two contingent blocks (X_1, X_2) that form a block wise source. Thus, given two independent sources we can hope to partition each one of them into a block-wise source and apply our 4-source extractor. An obvious problem is that we do not know where to split a given source X. The main construction of the paper gives a technique that given two samples x and y from two independent sources X and Y with rate $\geq \delta$ finds a way to partition the two strings so that the resulting distributions are independent block-wise source (at least on a non-negligible fraction of the original probability space). Applying this method gives a construction of a bipartite Ramsey graph.

References

[1] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting Randomness from Few Independent Sources. In *Proc. 45th FOCS*. IEEE, 2004.
[2] B. Chor and O. Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. In *Proc. 26th FOCS*, pages 429–442. IEEE, 1985.
[3] Y. Dodis, A. Elbaz, R. Oliveira, and R. Raz. Improved Randomness Extraction from Two Independent Sources. In *Proc. of 8th RANDOM*, 2004.
[4] P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.
[5] P. Pudlák and V. Rödl. Pseudorandom sets and explicit constructions of Ramsey graphs, 2004. Submitted for publication.
[6] R. Raz. Extractors with Weak Random Seeds, 2005. To appear.
[7] M. Saks, A. Srinivasan, and S. Zhou. Explicit OR-dispersers with polylogarithmic degree. *Journal of the ACM*, 45(1):123–154, January 1998.
[8] R. Shaltiel. Recent developments in extractors. *Bulletin of the European Association for Theoretical Computer Science*, 2002.
[9] U. Vazirani. Strong Communication Complexity or Generating Quasi-Random Sequences from Two Communicating Semi-Random Sources. *Combinatorica*, 7, 1987. Preliminary version in STOC' 85.

Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size

Ran Raz

An arithmetic formula is multi-linear if the polynomial computed by each of its sub-formulas is multi-linear. We prove that any multi-linear arithmetic formula for the permanent or the determinant of an $n \times n$ matrix is of size super-polynomial in n. Previously, super-polynomial lower bounds were not known (for any explicit function) even for the special case of multi-linear formulas of constant depth.

The talk presented lower bounds and methods from [3, 4].

[2] A block-wise source is a distribution $X = (X_1, X_2)$ where X_1 has "large" entropy and X_2 has large entropy *conditioned* on any fixing of X_1.

Introduction. Arithmetic formulas for computing the permanent and the determinant of a matrix have been studied since the 19th century. Are there polynomial size formulas for these functions ? Although the permanent and the determinant are among the most extensively studied computational problems, polynomial size formulas for these functions are not known. The smallest known formula for the permanent of an $n \times n$ matrix is of size $O(n^2 2^n)$. The smallest known formula for the determinant of an $n \times n$ matrix is of size $n^{O(\log n)}$. An outstanding open problem in complexity theory is to prove that polynomial size formulas for these functions do not exist. Note, however, that super-polynomial lower bounds for the size of arithmetic formulas are not known for any explicit function and that questions of this type are considered to be among the most challenging open problems in theoretical computer science.

We prove super-polynomial lower bounds for the subclass of *multi-linear formulas*. An arithmetic formula is *multi-linear* if the polynomial computed by each of its sub-formulas is multi-linear (as a formal polynomial), that is, in each of its monomials the power of every input variable is at most one.

Multi-Linear Formulas. Let F be a field, and let $\{x_1, ..., x_m\}$ be a set of input variables. An *arithmetic formula* is a binary tree whose edges are directed towards the root. Every leaf of the tree is labelled with either an input variable or a field element. Every other node of the tree is labelled with either $+$ or $\times$ (in the first case the node is a *plus gate* and in the second case a *product gate*).

An arithmetic formula computes a polynomial in the ring $\mathrm{F}[x_1, ..., x_m]$ in the following way. A leaf just computes the input variable or field element that labels it. A plus gate computes the sum of the two polynomials computed by its sons. A product gate computes the product of the two polynomials computed by its sons. The *output* of the formula is the polynomial computed by the root. The *size* of the formula is defined to be the number of nodes in the tree.

A polynomial in the ring $\mathrm{F}[x_1, ..., x_m]$ is *multi-linear* if in each of its monomials the power of every input variable is at most one. An arithmetic formula is *multi-linear* if the polynomial computed by each gate of the formula is multi-linear.

Motivation. Multi-linear formulas are restricted, as they do not allow the intermediate use of higher powers of variables in order to finally compute a certain multi-linear function. Note, however, that for many multi-linear functions, formulas that are not multi-linear are very counter-intuitive, as they require a "magical" cancellation of all high powers of variables. For many multi-linear functions, it seems "obvious" that the smallest formulas should be multi-linear.

Multi-linear polynomials are very powerful and are extensively used in theoretical computer science. Hence, the class of multi-linear formulas seems to be quite strong and it is very interesting to study its computational power.

Note also that both the permanent and the determinant are multi-linear functions in the input variables and that many of the well known formulas for these functions are multi-linear formulas. In particular, the smallest known arithmetic formula for the permanent is multi-linear. (For the determinant, the smallest known formulas are not multi-linear. Sub-exponential size multi-linear formulas for the determinant are not known.)

Finally, we note that several classes of formulas that were studied in the past are subclasses of multi-linear formulas. One example is *monotone arithmetic formulas.* It is easy to see that a monotone arithmetic formula for a multi-linear function is always multi-linear.

Our Results. We prove that over any field, any multi-linear arithmetic formula for the permanent or the determinant of an $n \times n$ matrix is of size $n^{\Omega(\log n)}$. An obvious corollary of our result is that over any field, any multi-linear arithmetic circuit for the permanent or the determinant of an $n \times n$ matrix is of depth $\Omega(\log^2 n)$. Our method is quite general and can be applied for many other functions.

Previous Work. Multi-linear arithmetic formulas were formally defined in [2]. Previous to our result, lower bounds for the size of multi-linear formulas were not known even for formulas of constant depth. Exponential lower bounds for a variant of constant depth multi-linear formulas were obtained in [2]. Lower bounds for several other restricted subclasses of multi-linear formulas were obtained in [1, 2, 5].

Methods. The starting point for our proof is the partial derivatives method of Nisan and Wigderson [1, 2]. It was suggested in [2] that for certain restricted subclasses of arithmetic formulas (and circuits), the dimension of the space spanned by all partial derivatives of the output is quite small. The method was used in [1, 2, 5] to obtain lower bounds for several subclasses of formulas and circuits. Note, however, that for multi-linear formulas the dimension of the space spanned by all partial derivatives may be very large, even if the formula is of linear size. In particular, that dimension may be much larger than the dimension of the space spanned by all partial derivatives of the permanent or the determinant. Nevertheless, the set of partial derivatives still plays a crucial roll in our proof.

To handle sets of partial derivatives, we make use of the *partial derivatives matrix.* The partial derivatives matrix was first used for proving lower bounds by Nisan [1], and was later on used in several other works.

In our proof, we also use rank arguments as well as random restrictions. Both these methods were used for proving lower bounds in numerous of works. However, we use them here in a completely different way. For example, random restrictions were used in many works in order to eliminate gates. Here, we use random restrictions in order to make gates unbalanced without eliminating even a single gate.

References

[1] N. Nisan. *Lower Bounds for Non-Commutative Computation*, STOC (1991), 410–418

[2] N. Nisan, A. Wigderson. *Lower Bounds on Arithmetic Circuits Via Partial Derivatives*, Computational Complexity **6**(3) (1996), 217–234.

[3] R. Raz. *Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size*, STOC (2004), 633–641

[4] R. Raz. *Multilinear-$NC_1 \neq$ Multilinear-NC_2*, FOCS (2004), 344–351

[5] R. Raz, A. Shpilka. *Deterministic Polynomial Identity Testing in Non Commutative Models*, Conference on Computational Complexity (2004), 215–222

Specialized Session on Cryptography

Shafi Goldwasser and Moni Naor (Session Chairs)

Cryptography and complexity have been fertilizing each other for the last three decades. Therefore, there were two sessions concentrating on the connection between complexity and cryptography, an afternoon and an evening one. The talks and the abstracts are listed below.

1. Yael Kalai Tauman: On the Impossibility of Obfuscation with Auxiliary Inputs

Barak et. al. [1] formulated the notion of obfuscation, and showed that there exist (contrived) classes of functions that cannot be obfuscated. In contrast, Canetti [7] and Wee [19] showed how to obfuscate point functions, under various complexity assumptions. Thus, it would seem possible that most programs of interest can be obfuscated even though in principle general purpose obfuscators do not exist.

We show that this is unlikely to be the case. In particular, we consider the notion of "obfuscation w.r.t. auxiliary inputs," which corresponds to the setting where the adversary, which is given the obfuscated circuit, may have some a priori information. This is essentially the case of interest in any usage of obfuscation we can imagine. We prove that there exist many natural classes of functions that cannot be obfuscated w.r.t. auxiliary inputs, both when the auxiliary input is dependent on the function being obfuscated and even when the auxiliary input is independent of the function being obfuscated.

2. Boaz Barak: Concurrent Composition Using Super-Polynomial Simulation [2]

We consider the problem of constructing a secure protocol for any multi-party functionality, which remains secure when executed concurrently with multiple copies of itself and other protocols, *without* any assumptions on existence of trusted parties, honest majority or synchronicity of the network. Recently it was shown by Lindell [13] that such a protocol is *impossible* to obtain under the standard definition of security, namely, polynomial-time simulation by an ideal adversary.

We construct a protocol for this problem which is secure in this setting, under a relaxed definition security, namely, *quasi-polynomial-time* simulation by an ideal adversary. Quasi-polynomial-time simulation seems to suffice for the *canonical application* of multi-party secure computation; that is obtaining protocols for any task whose privacy, integrity and input independence cannot broken by efficient adversaries under reasonable cryptographic assumptions. We emphasize that the security of our protocol does not rely on setup conditions such as the existence of a common reference string, nor does it require an existence of honest majority of parties.

Our construction is the first such protocol under reasonably standard cryptographic assumptions (i.e., existence of a hash function collection that is collision resistent with respect to circuits of subexponential size, and existence of trapdoor permutations which are secure with respect to circuits of quasi-polynomial size).

The main new technique introduced is "protocol condensing". That is, taking a protocol that has strong security properties but requires *super-polynomial* communication and computation, and then transforming it into a protocol with *polynomial* communication and computation that still inherits the strong security properties of the original protocol. Our main result is obtained by combining this technique with previous results of Pass [17] and Canetti et al [8].

3. Guy Rothblum: The Complexity of Online Memory Checking[16]

We consider the problem of storing a large file on a remote and unreliable server. To verify that the file has not been corrupted, a user could store a small private (randomized)"fingerprint" on his own computer. This is the setting for the well-studied authentication problem, and the size of the required private fingerprint is well understood. We study the problem of sub-linear authentication: suppose the user would like to encode and store the file in a way that allows him to verify that it has not been corrupted, but without reading the entire file. If the user only wants to read t bits of the file, how large does the size s of the private fingerprint need to be? We define this problem formally, and show a tight lower bound on the relationship between s and t when the adversary is not computationally bounded, namely: $s \times t = \Omega(n)$, where n is the file size. The problem of sublinear authentication is an easier case of the online memory checking problem, introduced by Blum et al. [6] in 1991, and hence the same (tight) lower bound applies also to this problem.

It was shown by [6] that when the adversary is computationally bounded, under the assumption that one-way functions exist, it is possible to construct much better online memory checkers and sub-linear authentication schemes. It was not previously known, however, whether one-way functions are *required* for the implementation of efficient online checkers. The study of which computational assumptions are necessary for implementing cryptographic tasks was initiated by Impagliazzo and Luby [12]. We continue this study and show that the existence of one-way functions is also a *necessary* condition for implementing efficient online memory checker: even slightly breaking the lower bound in a computational setting implies the existence of one-way functions.

To show lower bounds we reduce the problems of online memory checking and sublinear authentication to a communication complexity problem. We show these cryptographic primitives are related to the *simultaneous messages (SM)* communication model, introduced by Yao [18]. Newman and Szegedy [15] showed tight bounds for the SM complexity of the equality function, and their result was generalized by Babai and Kimmel [4]. To prove a lower bound for sublinear authentication, we generalize Yao's SM model, introducing a Consecutive Messages model of communication complexity. We then extend Babai and Kimmel's result to the new model. We also show that breaking the lower bound in a computational setting implies the existence of one-way functions. Another essential ingredient of our results is an algorithm for learning adaptively changing distributions (ACDs), see Naor and Rothblum [14]. We use this learning algorithm to show that an adversary can "learn" the distribution of addresses that the sublinear authenticator will read in its next run.

4. Sergey Yekhanin: A Geometric Approach to Information-Theoretic Private Information Retrieval [20]

A t-private information retrieval (PIR) scheme allows a user to retrieve the i'th bit of an n-bit string x replicated among k servers, while any coalition of up to t servers learns no information about i. We present a new geometric approach to PIR, and obtain (1) A t-private k-server protocol with smaller communication complexity, (2) A 2-server protocol with $O(n^{1/3})$ communication, polynomial preprocessing, and online work $O(n/\log^r n)$ for any constant r, improving the previously known bound of $O(n/\log^2 n)$, (3) Smaller communication for instance hiding, PIR with a polylogarithmic number of servers, robust PIR, and PIR with fixed answer sizes. Finally, our techniques are of independent interest, and may serve as a tool for obtaining better upper bounds. As an example of the model's power we give a new geometric proof of the best known upper bound for 1-private k-server PIR protocols of [3] for $k < 26$.

5. Moni Naor: Using Complexity Lower Bounds for Fighting Spam - Pebbling and Proofs of Work [11]

In 1992 Dwork and Naor proposed that e-mail messages be accompanied by easy-to-check *proofs of computational effort* in order to discourage junk e-mail, now known as spam [10],and suggested specific CPU-bound functions for this purpose. Noting that memory access speeds vary across machines much less than do CPU speeds, Abadi, et al. [5] initiated a fascinating new direction: replacing CPU-intensive functions with *memory-bound* functions, an approach that treats senders more equitably. Memory-bound functions were further explored by by Dwork, Goldberg, and Naor [9], who designed a class of functions based on pointer chasing in a very large shared random table T. We may think of T as part of the definition of their functions. Using hash functions modelled as truly random functions (i.e. 'random oracles'), they proved lower bounds on the amortized number of memory accesses that an adversary must expend per proof of effort.

The drawbacks to the use of a large random table in the definition of the function is that it makes distributing the software for proof-of-effort harder to distributed and to modify. We answer an open question of [9] by designing a compact representation for the table. The paradox, compressing an incompressible table, is resolved by embedding a time/space tradeoff into the process for constructing the table from its representation. Roughly speaking, our approach is to generate T using a memory-bound *process*. Sources for such processes are time/space tradeoffs, such as those offered by *graph pebbling*, and *sorting*. We exploit known dramatic time/space tradeoffs for pebbling in constructing a theoretical solution, with provable complexity bounds; the solution uses a hash function, modelled by a random oracle in the proof.

References

[1] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, Ke Yang, *On the (Im)possibility of Obfuscating Programs.* CRYPTO 2001, Lecture Notes in Computer Science, Springer, pages 1–18.

[2] B. Barak and A. Sahai. *How to Play Almost Any Mental Game Over the Net - Concurrent Composition Using Super-Polynomial Simulation*, 2005. To appear in FOCS' 05.

[3] A. Beimel, Y. Ishai, E. Kushelevitz, and J. F. Raymond, *Breaking the $O(n^{1/(2k-1)})$ barrier for information theoretic private information retrieval*, In Proc. of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS), pp. 261-270, 2002.

[4] László Babai and Peter G. Kimmel. *Randomized Simultaneous Messages: Solution of a Problem of Yao in Communication Complexity*, IEEE Conference on Computational Complexity 1997 239-246.

[5] M. Abadi, M. Burrows, M. Manasse, and T. Wobber. Moderately hard, memory-bound functions. In *Proc. 10th NDSS*, 2003.

[6] M. Blum, W. S. Evans, P. Gemmell, S. Kannan and M. Naor, *Checking the Correctness of Memories*, *Algorithmica* 12(2/3): 225-244 (1994)

[7] R. Canetti, *Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information*, CRYPTO 1997, pages 455–469.

[8] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally Composable Two-party Computation. In *Proc.* 34*th STOC*, pages 494–503. ACM, 2002.

[9] C. Dwork, A. V. Goldberg, and M. Naor. *On memory-bound functions for fighting spam*, Advances in Cryptology – CRYPTO'03, Lecture Notes in Computer Science, Springer, pages 426–444, 2003.

[10] C. Dwork and M. Naor. Pricing via processing, or, combatting junk mail. In *Advances in Cryptology – CRYPTO'92*, pages 139–147, 1993.

[11] C. Dwork, M. Naor and H. Wee, *Pebbling and Proofs of Work*, to appear, *Advances in Cryptology – CRYPTO'05.*

[12] Russell Impagliazzo and Michael Luby. *One-way Functions are Essential for Complexity Based Cryptography*, *FOCS 1989* 230-235

[13] Y. Lindell. *General Composition and Universal Composability in Secure Multi-Party Computation*, *Proc.* 44*th FOCS*. IEEE, 2003.

[14] Moni Naor and Guy N. Rothblum. "Simulating Secret Knowledge: Learning Adaptively Changing Distributions". *Manuscript* (2005). Available from authors' webpages.

[15] Ilan Newman and Mario Szegedy. "Public vs. Private Coin Flips in One Round Communication Games". *STOC 1996* 561-570

[16] M. Naor and G. N. Rothblum, *The Complexity of Online Memory Checking*, to appear in FOCS' 05.

[17] R. Pass, *Bounded-concurrent secure multi-party computation with a dishonest majority*, *Proc.* 36*th STOC*, pages 232–241. ACM, 2004.

[18] Andrew Chi-Chih Yao. "Some Complexity Questions Related to Distributive Computing". *STOC 1979* 209-213

[19] H. Wee, *On Obfuscating Point Functions*, *Proc.* 37*th STOC*, 2005.

[20] D. Woodruff, S. Yekhanin. *A geometric approach to information theoretic private information retrieval,* In Proc. of the 20th IEEE Conference on Computational Complexity, pp. 275–284, 2005.

Specialized Session on Complexity of Lattice Problems

Oded Regev (Session Chair)

This specialized session consisted of four talks. The first talk [1], presented by Henrik Koy of Frankfurt University, focused on a new method for lattice basis reduction. Unlike more traditional method for block basis reduction (such as that of Schnorr), Koy's basis reduction uses the dual basis throughout the reduction algorithm. This method yields improved running time, on the order of $n^3k^{k/2}$ when blocks of size k are used.

In the second talk [2], Claus P. Schnorr of Frankfurt University presented several approaches to lattice basis reduction based on the birthday method. These methods yield greatly improved running times. However, some of them have the drawback that their space requirement is very large, essentially the same as the time requirement. This forms the main bottleneck is applying these method practically, and it is an interesting open question to reduce the space requirement.

In the third talk [3], Oded Regev of Tel Aviv University presented reductions among lattice problems of different norms. The result is based on the method of random embedding. It shows a gap-preserving reduction from the l_2 norm of lattice problems to the corresponding problem in the l_p norm for any $1 \leq p \leq \infty$. This implies that it is enough to prove NP-hardness in the l_2 norm as this automatically implies NP-hardness in all other norms.

In the fourth talk [4], Johannes Blömer of University of Paderborn, described an improved reduction between the two main lattice problems: the shortest vector problem (SVP) and the closest vector problem (CVP). More precisely, the reduction is from approximating CVP to within $n \cdot f(n)$ to approximating SVP to within $f(n)$ where $f(n) \geq 1$ is any function of n. This improves on earlier work of Kannan. However, unlike Kannan's reduction, Blömer's reduction only solves the optimization version of CVP (where the goal is to find the distance of the target vector from the lattice) as opposed to the search version of CVP. The reason for this has to do with the use of non-constructive transference theorems in the reduction.

References

[1] H. Koy. Primal-dual segment reduction of lattice bases. Manuscript.
[2] C.P. Schnorr. General Birthday Reduction Reconsidered. Manuscript.
[3] O. Regev and R. Rosen. Lattices, Norms, and Embeddings. Manuscript.
[4] J. Blömer. Reductions between two lattice problems. Manuscript.

Specialized Session on Algebraic Complexity

Peter Bürgisser (Session Chair)

The talks in this session on algebraic complexity were dealing with a variety of topics centering around the evaluation and factorization of polynomials, problems in (semi)algebraic geometry, and derandomization in the context of arithmetic circuits.

Although the permanent and the determinant are among the most extensively studied computational problems, polynomial size formulas for these functions are not known. The talk by Ran Raz presented an exciting super-polynomial lower bound on the size of multilinear formulas for the permanent and determinant from [18, 19]. Previous lower bounds results for restricted subclasses of multilinear formulas were obtained in [16, 17, 20]. The lower bound proof is based on rank arguments and a novel use of random restrictions, which are used to unbalance gates (instead of eliminating them as usual).

Joos Heintz discussed some new aspects of effective elimination theory (joint work with Bart Kuijpers). He presented a model in the spirit of Constraint Database Theory [13], which allows the descriptive specification of the most fundamental tasks of effective elimination theory in algebraic and semialgebraic geometry. This requires a suitable extension and refinement of the traditional database model. In particular, polynomial equation solving is modeled by so called "sample point queries". By means of a suitable genericity condition the notion of "geometric query" is introduced [10]. This notion allows a fairly realistic repesentation of traditional elimination tasks and in particular the *descriptive* specification of elimination polynomials (different from their more traditional *operative* specifications). In this model, it is possible to prove the *intrinsic exponential time* character of geometric elimination procedures, under the restriction that they are parsimonious with respect to branchings. As a byproduct one obtains that the branching-free interpolation of polynomials of given arithmetic circuit complexity requires exponential time (compare with [8]).

Peter Bürgisser studied the computational complexity of two of the most fundamental invariants of complex algebraic varieties: the Euler characteristic and the Hilbert polynomial. He presented results from [3, 4, 5, 6] (joint work with Felipe Cucker and Martin Lotz). A version $\#\mathrm{P}_{\mathbb{C}}$ of Valiant's counting complexity class #P [21], tailored to the Blum-Shub-Smale model of computation [7] over $\mathbb{C}$, is defined and studied. (Over $\mathbb{R}$, such a counting class was first introduced by Meer [15].) The problem $\#\mathrm{HN}_{\mathbb{C}}$ of counting the number of complex common zeros of a finite set of multivariate polynomials turns out to be complete for $\#\mathrm{P}_{\mathbb{C}}$. The first main result states that the problem $\mathrm{EULER}_{\mathbb{C}}$ of computing the topological Euler characteristic of a complex algebraic variety is polynomial time equivalent to the problem $\#\mathrm{HN}_{\mathbb{C}}$. The second main result establishes a polynomial time reduction from the problem $\textsc{Hilbert}_{\mathrm{sm}}$ of computing the Hilbert polynomial of a smooth equidimensional complex projective variety to $\#\mathrm{HN}_{\mathbb{C}}$. Analogous statements are shown for the Turing model of computation.

The reduction from $\mathrm{EULER}_{\mathbb{C}}$ to $\mathrm{HN}_{\mathbb{C}}$ crucially depends on a recent result due to Aluffi [1]. This result characterizes the Euler characteristic of a (possibly singular) projective hypersurface $Z(f)$ in terms of the multidegrees of the projective gradient map of f. The reduction from $\textsc{Hilbert}_{\mathrm{sm}}$ to $\#\mathrm{HN}_{\mathbb{C}}$ is based on ideas and tools of intersection theory, enumerative geometry, and Schubert calculus. In particular, the Hirzebruch-Riemann-Roch Theorem [11] is used.

Erich Kaltofen presented a new result about factoring sparse polynomials (joint work with Pascal Koiran). H.W. Lenstra Jr. [14] had found a polynomial time algorithm for finding the small degree factors of a sparse univariate rational polynomial The new algorithm by Kaltofen and Koiran [12] allows to compute the rational linear factors of sparse bivariate rational polynomials with rational coefficients in deterministic polynomial time. The essence of the proof is a "gap theorem" based on the Bogomolov property of cyclotomic extensions [2], which separates the Weil height for non-roots of unity by a constant from 1.

The talk by Zeev Dvir was motivated by the fundamental *Polynomial Identity Testing* (PIT) problem: given a circuit computing a multivariate polynomial, determine whether the polynomial is identically zero. It is well known that this task can be solved in polynomial time by *randomized* algorithms. Two results for depth-3 circuits with a bounded top fan-in were shown: a deterministic algorithm that runs in quasipolynomial time, and a randomized algorithm that runs in polynomial time and uses only polylogarithmic number of random bits (joint work with Amir Shpilka). The proof is based on a relation to *Locally Decodable Codes*. Those are codes that allow the recovery of each message bit from a constant number of entries of the codeword. Along the way, known results on locally decodable codes were improved, cf. [9].

References

[1] P. Aluffi, *Computing characteristic classes of projective schemes*, J. Symbolic Comput. **35(1)** (2003), 3–19.

[2] F. Amoroso and R. Dvornicich, *A lower bound for the height in Abelian extensions*, J. Number Theory **80** (2000), 260–272.

[3] P. Bürgisser and F. Cucker, *Counting complexity classes for numeric computations II: Algebraic and semialgebraic sets*, Proc. 36th Ann. ACM STOC (2004), 475–485. Full version at `http://www.arxiv.org/abs/cs/cs.CC/0312007`.

[4] P. Bürgisser, F. Cucker, M. Lotz, *The Complexity to Compute the Euler Characteristic of Complex Varieties*, Comptes rendus de l'Académie des sciences Paris, Ser. I **339** (2004), 371–376.

[5] P. Bürgisser, F. Cucker, M. Lotz, *Counting Complexity Classes for Numeric Computations III: Complex Projective Sets*, Foundations of Computational Mathematics, to appear.

[6] P. Bürgisser, M. Lotz, *The complexity of computing the Hilbert polynomial of smooth equidimensional complex projective varieties*, `http://www.arxiv.org/abs/cs/cs.CC/0502044`.

[7] L. Blum and M. Shub and S. Smale, *On a theory of computation and complexity over the real numbers*, Bulletin of the AMS **21** (1989), 1–46.

[8] D. Castro, M. Giusti, J. Heintz, G. Matera, L. M. Pardo, *The hardness of polynomial equation solving*, Foundations of Computational Mathematics **3** (2003), 1-74.

[9] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan, *Lower bounds for linear locally decodable codes and private information retrieval*, Electronic Colloquium on Computational Complexity (ECCC) **080** (2001).

[10] J. Heintz, B. Kuijpers, *Constraint data bases, data structures and efficient query elimination*, in Proceedings of the 1st International Symposium Applications of Constraint Databases (CDB'04), B. Kuijpers, P. Revesz, eds., Springer Lecture Notes in Computer Science **3074** (2004), 1-24.
[11] F. Hirzebruch, *New Topological Methods in Algebraic Geometry*, Die Grundlehren der Mathematischen Wissenschaften **131** (1966), Springer Verlag.
[12] Erich Kaltofen and Pascal Koiran, *On the complexity of factoring bivariate supersparse (lacunary) polynomials*, in Proc. Internat. Symp. Symbolic Algebraic Comput. 2005, ACM Press, New York, to appear.
[13] G. M. Kuper, J. Paredans, L. Libkin, *Constraint databases*, Springer Verlag 2000.
[14] H. W. Lenstra, Jr, *Finding small degree factors of lacunary polynomials*, in *Number Theory in Progress*, volume 1 Diophantine Problems and Polynomials, Kálmán Győry, Henryk Iwaniec, and Jerzy Urbanowicz, eds., Stefan Banach Internat. Center, Walter de Gruyter Berlin, New York (1999), 267–276.
[15] K. Meer, *Counting problems over the reals*, Theoretical Computer Science **242** (2000), 41–58.
[16] N. Nisan, *Lower Bounds for Non-Commutative Computation*, STOC (1991), 410–418.
[17] N. Nisan, A. Wigderson, *Lower Bounds on Arithmetic Circuits Via Partial Derivatives*, Computational Complexity **6(3)** (1996), 217–234.
[18] R. Raz, *Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size*, STOC (2004), 633–641.
[19] R. Raz, *Multilinear-$NC_1 \neq$ Multilinear-NC_2*, FOCS (2004), 344–351.
[20] R. Raz, A. Shpilka. *Deterministic Polynomial Identity Testing in Non Commutative Models*, Conference on Computational Complexity (2004), 215–222.
[21] L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science **8** (1979), 189–201.

Specialized Session on Randomness Extractors

Boaz Barak (Session Chair)

Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$ (e.g., the family of distributions over affine subspaces of a certain dimension; the family of products of independent distributions of a certain entropy). A *randomness extractor* w.r.t. $\mathcal{X}$ is a deterministic function $E : \{0,1\}^n \to \{0,1\}^m$ such that for every random variable $X \in \mathcal{X}$, $E(X)$ is close to the uniform distribution. In recent years, constructing explicit, efficiently computable randomness extractors for interesting families of distributions has been an important research direction in theoretical Computer Science, with many important connections and applications. In this session several new results in this direction were reported. In addition, a talk about the related notion of randomness *dispersers* was also presented in a different session of the same workshop by Ronen Shaltiel. The following talks were given in this session:

Extracting Randomness Using Few Independent Sources — Boaz Barak. In this work we give the first deterministic extractors from a constant number of weak sources whose entropy rate is less than 1/2. Specifically, for every $\delta > 0$ we give an explicit construction for extracting randomness from a constant (depending polynomially on $1/\delta$) number of distributions over $\{0,1\}^n$, each having min-entropy δn. These extractors output n bits, which are 2^{-n} close to uniform.

This construction uses several results from additive number theory, and in particular a recent one by Bourgain, Katz and Tao [3] and of Konyagin [4].

Joint work with Russell Impagliazzo and Avi Wigderson. An extended abstract of this work appeared in the FOCS' 2004 conference [1].

Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number — David Zuckerman. A randomness extractor is an algorithm which extracts randomness from a low-quality random source, using some additional truly random bits. We construct new extractors which require only $\log n + O(1)$ additional random bits for sources with constant entropy rate. We further construct dispersers, which are similar to one-sided extractors, which use an arbitrarily small constant times $\log n$ additional random bits for sources with constant entropy rate.

We use our dispersers to derandomize the results of Hastad [6] and Feige-Kilian [5] and show that approximating Max Clique and Chromatic Number to within $n^{1-\epsilon}$ are NP-complete, for any $\epsilon > 0$. We also derandomize the results of Khot [7] and show that there is a $\gamma > 0$ such that no quasi-polynomial time algorithm approximates the clique number or chromatic number to within $n/2^{(\log n)^{1-\gamma}}$, unless $\tilde{\mathrm{NP}} = \tilde{\mathrm{P}}$.

Our constructions rely on recent results in additive number theory and extractors by Bourgain-Katz-Tao [3], Barak-Impagliazzo-Wigderson [1], Barak-Kindler-Shaltiel-Sudakov-Wigderson [2], and Raz [8]. We also simplify and slightly strengthen key lemmas in the second and third of these papers.

Deterministic Extractors for Affine Sources over Large Fields — Ariel Gabizon. An (n,k)-*affine source* over a finite field $\mathbb{F}$ is a random variable $X = (X_1, ..., X_n) \in \mathbb{F}^n$, which is uniformly distributed over an (unknown) k-dimensional affine subspace of $\mathbb{F}^n$. For the case of sufficiently large fields, we improve over [2] and show how to (deterministically) extract practically all the randomness from affine sources, for any field of size larger than n^c (where c is a large enough constant). Our main results are as follows:

1. **(For arbitrary k):** For any n, k and any $\mathbb{F}$ of size larger than n^{20}, we give an explicit construction for a function $D : \mathbb{F}^n \to \mathbb{F}^{k-1}$, such that for any (n,k)-affine source X over $\mathbb{F}$, the distribution of $D(X)$ is ϵ-close to uniform, where ϵ is polynomially small in $|\mathbb{F}|$.
2. **(For $k = 1$):** For any n and any $\mathbb{F}$ of size larger than n^c, we give an explicit construction for a function $D : \mathbb{F}^n \to \{0,1\}^{(1-\delta)\log_2|\mathbb{F}|}$, such that for any $(n,1)$-affine source X over $\mathbb{F}$, the distribution of $D(X)$ is ϵ-close to uniform, where ϵ is polynomially small in $|\mathbb{F}|$. Here, $\delta > 0$ is an arbitrary small constant, and c is a constant depending on δ.

Joint work with Ran Raz.

References

[1] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting randomness from few independent sources. In *Proc. 45th FOCS* , 2004.

[2] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.
[3] J. Bourgain, N. Katz, and T. Tao. A sum-product estimate in finite fields, and applications. *Geometric and Functional Analysis*, 14:27–57, 2004.
[4] S. Konyagin. A sum-product estimate in fields of prime order. Arxiv technical report, `http://arxiv.org/abs/math.NT/0304217`, 2003.
[5] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57:187–199, 1998.
[6] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
[7] S. Khot. Improved inapproximability results for MaxClique, Chromatic Number and Approximate Graph Coloring. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 600–609, 2001.
[8] R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

Specialized Session on Pseudorandomness

Ronen Shaltiel (Session Chair)

This summary covers two talks which were given in the informal "pseudorandomness session". The first given by Ronen Shaltiel is based on the paper [1] and the second given by Avi Wigderson is based on the paper [2].

Pseudorandomness for approximate counting and sampling - Ronen Shaltiel. We study computational procedures that use both randomness and nondeterminism. Examples are Arthur-Merlin games and approximate counting and sampling of NP-witnesses. The goal of this paper is to derandomize such procedures under the weakest possible assumptions.

Our main technical contribution allows one to "boost" a given hardness assumption. One special case is a proof that

$$\mathrm{EXP} \not\subseteq \mathrm{NP/poly} \Rightarrow \mathrm{EXP} \not\subseteq \mathrm{P}_{||}^{\mathrm{NP}}/\mathrm{poly}.$$

In words, if there is a problem in EXP that cannot be computed by poly-size nondeterministic circuits then there is one which cannot be computed by poly-size circuits which make non-adaptive NP oracle queries. This in particular shows that the various assumptions used over the last few years by several authors to derandomize Arthur-Merlin games (i.e., show AM = NP) are in fact all *equivalent*. In addition to simplifying the framework of AM derandomization, we show that this "unified assumption" suffices to derandomize several other probabilistic procedures.

For these results we define two new primitives that we regard as the natural pseudorandom objects associated with *approximate counting* and *sampling* of NP-witnesses. We use the "boosting" theorem (as well as some hashing techniques) to construct these primitives using an assumption that is no stronger than that used to derandomize Arthur-Merlin games. As a consequence, under this assumption, there are *deterministic* polynomial time algorithms that use *non-adaptive* NP-queries and perform the following tasks:

- approximate counting of NP-witnesses: given a Boolean circuit A, output r such that $(1-\epsilon)|A^{-1}(1)| \leq r \leq |A^{-1}(1)|$.
- pseudorandom sampling of NP-witnesses: given a Boolean circuit A, produce a polynomial-size sample space that is computationally indistinguishable from the uniform distribution over $A^{-1}(1)$.

We also present applications. For example, we observe that Cai's proof that $S_2^p \subseteq \mathrm{ZPP}^{\mathrm{NP}}$ and the learning algorithm of Bshouty et al. can be seen as a reduction to sampling that is not probabilistic. As a consequence they can be derandomized under the assumption stated above, which is weaker than the assumption that was previously known to suffice.

Joint work with Chris Umans.

A Randomness-Efficient Sampler for Matrix-valued Functions and Applications - Avi Wigderson. In this paper we give a randomness efficient sampler for matrix-valued functions. Specifically, we show that the random walk on an expander approximates the recent Chernoff-like bound for matrix-valued functions of Ahlswede and Winter, in a manner which depends optimally on the spectral gap. The proof uses perturbation theory, and is a generalization of Gillman's and Lezaud's analysis of the Ajtai-Komlos-Szemeredi sampler for real-valued functions.

Derandomizing our sampler gives a few applications, yielding deterministic polynomial time algorithms for problems in which derandomizing independent sampling gives only quasipolynomial time deterministic algorithms. The first (which was our original motivation) is to a polynomial-time derandomization of the Alon-Roichman theorem: given a group of size n, find O(log n) elements which generate it as an expander. This implies a second application - efficiently constructing a randomness-optimal homomorphism tester, significantly improving the previous result of Shpilka and Wigderson. The third is to a "non-commutative" hypergraph covering problem - a natural extension of the set-cover problem which arises in quantum information theory, in which we efficiently attain the integrality gap when the fractional semi-definite relaxation cost is constant.

Joint work with David Xiao.

References

[1] R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. In *Proc. of 20th IEEE conference on computational complexity*, 2005.
[2] A. Wigderson and D. Xiao. A Randomness-Efficient Sampler for Matrix-valued Functions and Applications. To appear in *Proc. of 46th IEEE conference on foundations of computer science*, 2005.

Specialized Session on the Complexity of Low Distortion Embeddings

Muli Safre (Session chair)

Embeddings of one metric space into another have been investigated for many years, as an active area of Banach spaces, (see Johnson-Lindenstrauss lemma, and Bourgain's upper-bound). One considers a mapping of one metric space M to a metric space M' with the smallest distortion, namely when all distances between any pair of mapped points are within some factor of the distances between the original preimage points. Embeddings recently were shown to be a prolific algorithmic methodology [6, 1]

The shortest path between points in an undirected graph is a metric. One can consider embedding such a given metric in another, simpler metric, and applying a known algorithmic on that simpler metric, so as to altogether solve the problem at hand. The simplest metric possible for such purposes is the L1 metric. A low distortion embedding into L1 amounts to an embedding into the binary hypercube, hence translates shortest distance into Hamming distance.

There are many other potential ways by which to apply embedding techniques to efficiently solve computational problems. In fact, Semi Definite Programming [4] can be thought of as a related technique, where one maps a graph into the Euclidean sphere. Recent results regarding computing the expansion of a graph, or more generally the sparsest cut [1] in a graph, namely, where a set of demand is imposed and the cut need to satisfy as many of those as possible.

It is therefore worthwhile to consider whether one metric embeds into another with as low as possible distoryion. And if indeed that is the case what is the computational complexity of such embeddings. Such results were shown recently by Khot and Vishnoy and by Khot and Naor (in as of yet unpublished papers) for the L1 metric. An exciting aspect of these results is their use of Analysis of Boolean Functions [5, 3, 2]. Specifically, in order to prove such non embedability results one should apply one of the theorems proved regarding influences of low-degree functions, namely Boolean function whose Fourier transform weight is concentrated of small characters. It also seems to be the case that in order to improve such results one have to rely on and hopefully prove some open conjecture regarding the distribution of the Fourier weight of Boolean functions with rather high degree.

References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning.

[2] J. Bourgain. On the distribution of the fourier spectrum of boolean functions. to appear in Israel J. of Math., 2001.

[3] Ehud Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–35, 1998.

[4] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[5] J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In IEEE, editor, *29th annual Symposium on Foundations of Computer Science, October 24–26, 1988, White Plains , New York*, pages 68–80. IEEE Computer Society Press, 1988.

[6] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

Reporter: Zeev Dvir