Mathematisches Forschungsinstitut Oberwolfach

Report No. 52/2009

DOI: 10.4171/OWR/2009/52

Complexity Theory

Organised by
Peter Bürgisser (Paderborn)
Joachim von zur Gathen (Bonn)
Oded Goldreich (Rehovot)
Madhu Sudan (Cambridge, MA)

November 15th – November 21st, 2009

Abstract. Computational Complexity Theory is the mathematical study of the intrinsic power and limitations of computational resources like time, space, or randomness. The current workshop focused on recent developments in various sub-areas including arithmetic complexity, Boolean complexity, communication complexity, cryptography, probabilistic proof systems, pseudorandomness, and quantum computation. Many of the developements are related to diverse mathematical fields such as algebraic geometry, combinatorial number theory, probability theory, quantum mechanics, representation theory, and the theory of error-correcting codes.

Mathematics Subject Classification (2000): 68-06, 68Q01, 68Q10, 68Q15, 68Q17, 68Q25, 94B05, 94B35.

Introduction by the Organisers

The workshop *Complexity Theory* was organized by Peter Bürgisser (Universität Paderborn), Joachim von zur Gathen (B-IT, Bonn), Oded Goldreich (Weizmann Institute), and Madhu Sudan (MIT). The workshop was held on November 15th–21st 2009, and attended by approximately 50 participants spanning a wide range of interests within the field of Computational Complexity. The plenary program, attended by all participants, featured few long lectures as well as short (5-minute) reports by almost all participants. In addition, intensive interaction took place in smaller groups.

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). 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.

Computational complexity (a.k.a. complexity theory) is a central field of computer science with a remarkable list of celebrated achievements as well as a 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 algebra, analysis, combinatorics, geometry, number theory, optimization, probability theory, and quantum computation.

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

Efficient Simulation of Quantum Mechanics. The power of the standard model of quantum computation (QC), demonstrated by Shor's celebrated quantum algorithm for integer factorization, presses the fundamental question of whether this standard model is feasibly realizable. In the meeting, Scott Aaronson presented a different evidence to the difficulty of (classically) simulating a quantum mechanical (QM) system. His fundamental result exhibits a specific distribution that arises in QM and is easily generated by a QC, and he provides strong evidence that no (classical) probabilistic polynomial-time algorithm can generate it. Namely, such an algorithm would imply that any counting problem can be efficiently solved using an oracle to $\mathcal{NP}$ (i.e., $\mathcal{P}^{\#\mathcal{P}}$ would equal $\mathcal{BPP}^{\mathcal{NP}}$). This holds even if the classical algorithm only approximates the said distribution of the QM system. Furthermore, the QM system is a very simple and special one; it consists of a system of identical, non-interacting bosonic particles. This contrasts with the efficient simulation of a system of fermions shown by Valiant.

Kakeya Sets and Extractors. In 1999 Wolff posed the finite field analogue to the Kakeya problem, conjecturing that for every $K \subseteq \mathbb{F}_q^n$ that contains a line in every direction it holds that $K = \Omega(|\mathbb{F}|)$, where the constant hidden in the Omega-notation may depend on n. This analogue was observed to be related to the design of randomness extractors, hence the complexity theoretic interest in it. In the meeting, Zeev Dvir surveyed a recent series of works that settle this conjecture and obtained almost the optimal constant in the Omega-notation. Furthermore, he showed that the proof techniques are indeed applicable to the analysis of constructions of randomness extractors, yielding improvements in some of the parameters of such constructions. Recall that a (k,ϵ)-randomness extractor is a function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ such that for every random variable X of min-entropy at least k, when s is selected uniformly in $\{0,1\}^t$ it holds that $E(X,s)$ is ϵ-close to the uniform distribution over $\{0,1\}^m$.

Locally Decodable Codes of Sub-exponential Length. An error-correcting code is called locally decodable if, given a corrupted codeword, any bit in the original message can be correctly reconstructed (with high probability) based on a constant number of probes. Locally Decodable Codes (LDC) are closely related to (multi-server) Private Information Retrieval (PIR) schemes, which are of interest to cryptography. In the meeting, Klim Efremenko presented a 3-query LDC of sub-exponential length, thus improving on a breakthrough result of Yekhanin (which was presented in the 2007 meeting). Furthermore, in contrast to Yekhanin's construction, the current construction scheme has the pleasing feature of benefiting from more queries (i.e., it yields shorter lengths when more queries are allowed). The analogue of the main result for PIR yields a three-server scheme for n-bit long databases with communication $\exp(\widetilde{O}(\sqrt{\log n}))$, improving over Yekhanin's bound of $n^{1/30000000}$.

Constructing Low-Error 2-Query PCPs. Probabilistically Checkable Proofs (PCPs) are proofs that offer a trade-off between the number of locations inspected at random in the alleged proof and the statistical confidence in its validity. In the meeting, Irit Dinur presented a methodology for constructing (relatively short) PCPs in which verification is performed by two queries such that the error probability is inversely related to the length of the answers. The core of her new methodology is a new composition theorem that refers to "decodable PCPs" (a notion implicit in prior work). The resulting construction matches the parameters of the construction of Moshkovitz and Raz, but the current construction is significantly simpler. In contrast to prior results, her new constructions yield inapproximability results (for many natural optimization problems such as Max-Clique) in which approaching the "threshold of approximability" does not cause a deterioration in the complexity of the reduction.

Parallel Repetition of Interactive Protocols. It has been known for more than a decade that parallel repetition may fail to reduce the error in computationally-sound proof (a.k.a. argument) systems. In the meeting, Iftach Haitner presented a methodology for (slightly) modifying an interactive protocol such that parallel repetition does reduce the (observable) error in the *resulting* protocol. The modification amounts to having the verifier abort at random with probability $1/4$ (i.e., after each round, the verifier aborts with probability $1/4r$, where r denotes the number of rounds). In case of abort, the verifier always accepts, which means that this modification increases the probability of error. The benefit of this modification is that the probability of cheating in the parallel execution is not sensitive to whether the verifier aborts in any typical individual copy, which establishes sufficient independence between the copies.

On the Best Possible Approximation of CSPs. Constraint Satisfaction Problems (CSPs) are specified by a finite set of finite predicates (e.g., 3-SAT is specified by the set of predicates on at most three variables that may be written as disjunctions of the corresponding literals). In the meeting, Prasad Raghavendra presented an approximation threshold result for any CSP, assuming the Unique

Game Conjecture (UGC). Specifically, for every CSP and every $\epsilon > 0$, there exists a polynomial-time algorithm that gets within a factor ϵ of the threshold beyond which approximation becomes UGC-hard. Furthermore, this seemingly optimal approximation threshold factor can be efficiently approximated.

New Notions of Computational Entropy. Omer Reingold and Salil Vadhan presented two complementary notions of "computational entropy" (a.k.a. pseudoentropy, akin to pseudorandomness which refers to distributions that are computationally indistinguishable from the uniform distribution on n-bit strings). The first notion, called next bit (or block) pseudoentropy measures the computational unpredictability of the next bit (given the previous bits). In contrast to the extreme case (of full unpredictability), in general next bit pseudoentropy does not yield the standard notion of pseudoentropy. Nevertheless, the new notion is instrumental for deriving an improved construction of pseudorandom generators based on any one-way function. The second notion, called inaccessible entropy, refers to the infeasibility of generating a next block that is as random as expected by an unbounded observer. For example, if some party sends a (statistically hiding) commitment to a random value, then when asked to reveal the value it can provide at most one possible value, whereas from the (unbounded) observer's point of view any value is possible. Indeed, the notion of inaccessible entropy is is related to statistically hiding commitment schemes, and is actually pivotal to their construction.

The Average-Case Complexity of k-Clique. For any constant k, constant-depth (unbounded fan-in) circuits of size $O(n^k)$ can distinguish n-vertex graphs having a clique of size k from graphs lacking such a clique. Ben Rossman's presentation addressed the average case complexity of this problem, where the input distribution corresponds to the standard random graph model with arbitrary edge density (which may be thought of as studying the problem at the threshold edge density, where the problem is not trivial). Interestingly, relatively tight lower and upper bounds, asserting that the size is $n^{(k/4)+\Theta(1)}$, can be obtained. The same holds when considering monotone circuits (of arbitrary depth).

Poly-Logarithmic Independence Fools AC^0 Circuits. Two decades ago, it was conjectured that poly-sized constant-depth circuits (of unbounded fan-in) cannot distinguish between any two poly-logarithmically independent distributions, and hence any such (poly-log independent) distribution is pseudorandom with respect to AC^0 circuits. Mark Braverman presented a proof of this conjecture. The proof combines two known approximation methods that yield different and incomparable approximations of AC^0 circuits by low degree polynomials.

Fast Polynomial Factorization and Modular Composition. Chris Umans presented an improved randomized algorithms for factoring univariate polynomials over a finite field. The source of the improvement is a new algorithm for modular composition of univariate polynomials that operates in nearly linear time. In the case of very big finite fields, the algorithm uses a sequence of reductions that first reduce to a multivariate problem, then lift to the integers, next reduces modulo small primes, and finally applies a FFT. Most previous methods used only operations in the original field. As an interesting feature, the new method shows that (at the current state of knowledge) Boolean computations beat arithmetic ones for this algebraic problem.

Informal sessions. Besides the plenary formal program, intense interaction between the participants took place in smaller groups, as witnessed by the following list of afternoon or evening sessions.

- Structure problems and results on non-abelian groups (Wigderson)
- Polynomial identity testing (Shpilka)
- Compressing interactive communication (Rao)
- Security in steganography (Reischuk)
- Tutorial on group representations and matrix multiplication (Umans)
- Semantic communication (Sudan)
- On Smale's 17th problem (Cucker)
- Informal session of "going down hill" (Reingold)
- Semantic communication (Sudan)
- Open session on probabilistic proof systems, IP and PCP (Or Meir)
- Codes (Guruswami, Saraf, Kabanets, Kopparty, Impagliazzo)
- The Generalized Linial-Nisan Conjecture and BQP vs. PH (Aaronson)
- Sum of Squares (Koiran)
- Real polynomials for Boolean functions (Lovett, Beame)
- More on non-abelian groups (Wigderson)

The rest of this report. This report contains extended abstracts of the 15 long presentations as well as abstracts of 9 short cummunications.

Workshop: Complexity Theory

Table of Contents

Plenary Talks

Short Communications

Abstracts

Plenary Talks

Efficient simulation of quantum mechanics collapses the polynomial hierarchy

Scott Aaronson
(joint work with Alex Arkhipov)

We give a new type of evidence that quantum mechanics is hard to simulate classically—evidence that is more complexity-theoretic than (say) Shor's factoring algorithm. Specifically we show the following:

Theorem 1. *Suppose there exists a* BPP *machine* M *that, given any quantum circuit* Q*, approximately samples (with constant error in variation distance) from the probability distribution over* Q*'s possible output strings. Then* $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$*, so in particular the polynomial hierarchy collapses. Indeed, even if* M *is a* $\mathsf{BPP}^{\mathsf{PH}}$ *machine, we still get that* $\mathsf{P}^{\#\mathsf{P}} = \mathsf{PH}$ *and the polynomial hierarchy collapses.*

The proof uses a quantum algorithm that simulates a system of n identical, non-interacting bosonic particles. We exploit an old observation: that the amplitude for n non-interacting bosons to evolve to a particular state is given by the squared permanent of an $n \times n$ matrix, a $\#\mathsf{P}$-complete function. Therefore, one might hope that the ability to classically sample the bosons' final distribution would put $\#\mathsf{P}$ in the polynomial hierarchy. However, pushing this implication through requires some further ideas from complexity theory, including the random self-reducibility of the permanent, noisy interpolation of polynomials, and approximate counting in $\mathsf{BPP}^{\mathsf{NP}}$. We also need to upper-bound the probability that two or more bosons will occupy the same state (unlike fermions, bosons are not subject to the Pauli exclusion principle, and can "pile on top of each other").

Our result can be strengthened in two ways. First, even if every distribution samplable in BQP can be approximately sampled in $\mathsf{BPP}^{\mathsf{PH}}$, we still get that $\mathsf{P}^{\#\mathsf{P}} = \mathsf{PH}$ and PH collapses. This provides a new sort of evidence that quantum computers have capabilities outside the entire polynomial hierarchy.

Second, we can prove a version of our result for relational problems (problems where the output is an n-bit string, and there could be many valid outputs), rather than sampling problems. What remains unknown is a result for decision problems (e.g., "if $\mathsf{P} = \mathsf{BQP}$ then PH collapses").

1. Introduction

Yes, we are literally saying that we can separate quantum from classical computing using only standard complexity assumptions like

$\mathsf{P}^{\#\mathsf{P}} \neq \mathsf{PH}$, rather than cryptographic assumptions like the classical hardness of factoring.

Let us explain how this can be so. As a first observation, if we removed the word "approximately" from Theorem 1, the theorem would be straightforward to prove. For in 2005, Aaronson [1] showed that $\mathsf{PostBQP} = \mathsf{PP}$. Here $\mathsf{PostBQP}$ means BQP augmented with the ability to *postselect* on exponentially-unlikely measurement outcomes, while PP is the class of problems of the form, "given a Boolean formula φ, does φ have at least K satisfying assignments?" This already showed that *exact* simulation of quantum computers was closely related to counting problems—and that a classical algorithm to perform such simulation would have bizarre complexity consequences. But there are easier ways to get the same conclusion. For example, suppose we prepare the state $\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} f(x)|x\rangle$, for some efficiently-computable Boolean function $f : \{0,1\}^n \rightarrow \{-1,1\}$, then apply Hadamard gates to all n qubits, and finally measure in the standard basis. Then the probability of observing $|0\rangle^{\otimes n}$ is $p := \frac{1}{2^{2n}} \left(\sum_{x\in\{0,1\}^n} f(x) \right)^2$. So if we could only calculate p, we could solve a #P-complete problem!

Indeed, suppose we just had a BPP algorithm M that *sampled* from the same probability distribution as the quantum algorithm above. Then we could use *approximate counting* to estimate the fraction of random strings r such that $M(r)$ outputs $|0\rangle^{\otimes n}$. Now, even *estimating* p to within a constant multiplicative factor turns out to be #P-complete, basically because the function f can take both positive and negative values. Therefore, the sampling algorithm M would let us solve a #P-complete problem in the class $\mathsf{BPP}^{\mathsf{NP}}$ (which is known to contain approximate counting). By Toda's Theorem [3] that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$, this would imply that $\mathsf{P}^{\#\mathsf{P}} = \mathsf{BPP}^{\mathsf{NP}}$, and hence that the polynomial hierarchy would collapse.

2. Bringing In Bosons

These simple observations already reveal a striking connection between simulating quantum mechanics and solving #P-complete problems. However they might be dismissed as cheating—since *all the work of solving the* #P*-complete problem is shoehorned into the calculation of a single, exponentially-small amplitude.* If our simulation happened to be wrong about that one amplitude, then it might no longer compute anything #P-complete.

This immediately raises a question: can we design a quantum algorithm Q with the property that, even if we could only sample *approximately* from Q's output distribution on a classical computer, we would still be reducing a counting problem (which, presumably, is not in PH) to an *approximate* counting problem (which is)? This is the question that we answer affirmatively in this work.

The quantum algorithm Q that we use is extremely interesting in its own right. We will consider a system of n identical, non-interacting bosons, each of which can be in m modes $|1\rangle, \ldots, |m\rangle$ for some $m = \operatorname{poly}(n)$. The basis states of this system can be represented by lists $S = (s_1, \ldots, s_m)$ of *occupation numbers*: that is, nonnegative integers such that $s_1 + \cdots + s_m = n$. In the initial state $|\Psi\rangle$, modes 1 to n are occupied with a single boson each, while modes $n+1$ to m are unoccupied. The algorithm is as follows: first, choose a random $m \times m$ unitary matrix U, and let V be the $\binom{m+n-1}{n} \times \binom{m+n-1}{n}$ unitary matrix that corresponds to applying U to each of n identical bosons governed by the exchange interaction. Apply V to $|\Psi\rangle$, measure the resulting state $V|\Psi\rangle$ in the standard basis, and output the list S of occupation numbers that are observed.

Let A be the $m \times n$ matrix that consists of the first n columns of U. Also, given a list S of occupation numbers, let $A[S]$ be an $n \times n$ matrix in which the i^{th} row of A occurs s_i times. Then standard quantum mechanics tells us that each basis state $|S\rangle$ is observed with probability $|\mathrm{Per}(A[S])|^2 / s_1! \cdots s_m!$.

This is the central fact we exploit. For we know from complexity theory that, not only is the permanent of an $n \times n$ matrix a #P-complete function (as shown by Valiant [4]), it is a particularly nice #P-complete function—one with remarkable properties such as *random self-reducibility* (that is, the ability to solve any given instance by solving random instances instead). As it turns out, the permanent function has exactly the properties we need to prove our result.

We regret that we lack the space to sketch the proof of Theorem 1. Instead, let us provide a "sneak preview," by simply stating the main sub-claims in the proof. First, the quantum algorithm Q can be simulated efficiently on a "standard" quantum computer, with qubits subject to local gates. Second, provided the number of modes m is sufficiently large (say, at least $2n^2$), basis states S with multiple bosons "bunched together in the same mode" (that is, $s_i > 1$ for some i) contribute only negligibly to the total probability mass.[1] Third, the permanent function is random self-reducible not merely over large finite fields, but also over the complex numbers $\mathbf{C}$ with Gaussian norm. (This is the hardest part technically: linear interpolation no longer works, and we must instead do the interpolation using random trigonometric polynomials.) Fourth, given an $n \times n$ matrix Y of independent Gaussians with mean 0 and variance $1/m$, one can effectively "smuggle" Y into an $m \times n$ bosonic transition matrix A, in such a way that *the ability to sample a submatrix* $A[S]$ *of* A *with probability approximately* $|\mathrm{Per}(A[S])|^2 / s_1! \cdots s_m!$ *in* BPP, *implies the ability to approximate* $|\mathrm{Per}(Y)|^2$ *in* $\mathsf{BPP}^{\mathsf{NP}}$. Fifth, approximating $|\mathrm{Per}(Y)|^2$ is #P-complete—so that we obtain the promised collapse of $\mathsf{P}^{\#\mathsf{P}}$ with $\mathsf{BPP}^{\mathsf{NP}}$.

References

[1] S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Roy. Soc. London*, A461(2063):3473–3482, 2005. quant-ph/0412187.

[1] Of course *fermions* are subject to the Pauli exclusion principle, which prevents two or more particles from ever occupying the same mode. But that only comes about because fermionic amplitudes involve the determinant (which is computable in P), rather than the permanent (which is #P-complete). Indeed, by exploiting the fact that the determinant is in P, Terhal and DiVincenzo [2] (building on work of Valiant [5]) were able to give an efficient classical algorithm to simulate non-interacting fermions. Our result shows that, if their algorithm could be extended to bosons, then the polynomial hierarchy would collapse.

[2] B. M. Terhal and D. P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A*, 65(032325), 2002. quant-ph/0108010.

[3] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[4] L. G. Valiant. The complexity of computing the permanent. *Theoretical Comput. Sci.*, 8(2):189–201, 1979.

[5] L. G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM J. Comput.*, 31(4):1229–1254, 2002. Earlier version in STOC'2001.

Poly-logarithmic independence fools AC^0 circuits

Mark Braverman

Overview

We prove that poly-sized AC^0 circuits cannot distinguish a poly-logarithmically independent distribution from the uniform one. This settles the 1990 conjecture by Linial and Nisan [LN90]. The only prior progress on the problem was by Bazzi [Baz07], who showed that $O(\log^2 n)$-independent distributions fool poly-size DNF formulas. Razborov [Raz08] has later given a much simpler proof for Bazzi's theorem.

1. Summary

1.1. **The problem.** The main problem we consider is on the power of r-independence to fool AC^0 circuits. For a distribution μ on the finite support $\{0,1\}^n$, we denote by $\mathbf{E}_\mu[F]$ the expected value of F on inputs drawn according to μ. For an event X, we denote by $\mathbf{P}_\mu[X]$ its probability under μ. When the distribution under consideration is the uniform distribution on $\{0,1\}^n$, we suppress the subscript and let $\mathbf{E}[F]$ denote the expected value of F, and $\mathbf{P}[X]$ the probability of X. A distribution μ is said to *ε-fool* a function F if

$$|\mathbf{E}_\mu[F] - \mathbf{E}[F]| < \varepsilon.$$

The distribution μ on $\{0,1\}^n$ is r-independent if every restriction of μ to r coordinates is uniform on $\{0,1\}^r$. AC^0 circuits are circuits with AND, OR and NOT gates, where the fan-in of the gates is unbounded. The depth of a circuit C is the maximum number of AND/OR gates between an input of C and its output. The problem we study is

Main Problem. How large does $r = r(m, d, \varepsilon)$ have to be in order for every r-independent distribution μ on $\{0,1\}^n$ to ε-fool every function F that is computed by a depth-d AC^0 circuit of size $\le m$?

Prior to our work, Bazzi [Baz07, Baz09], in a proof that was later simplified by Razborov [Raz08], showed that a poly-logarithmic r is sufficient for $d = 2$ (i.e. when the F's are DNF or CNF formulas):

Theorem 1. [Baz07, Raz08] *$r(m, 2, \varepsilon)$-independence ε-fools depth-2 circuits, where*

$$r(m, 2, \varepsilon) = O\left(\log^2 \frac{m}{\varepsilon}\right).$$

Our main result is that for any constant d, $r(m, d, \varepsilon)$ is poly-logarithmic in m/ε. This gives a huge class of distributions that look random to AC^0 circuits. For example, as in [Baz09], it implies that linear codes with poly-logarithmic seed length can be PRGs for AC^0.

1.2. **Main results.** We prove the following:

Main Theorem. *Let $s \geq \log m$ be any parameter. Let F be a boolean function computed by a circuit of depth d and size m. Let μ be an r-independent distribution where*

$$r \geq r(s, d) = 3 \cdot 60^{d+3} \cdot (\log m)^{(d+1)(d+3)} \cdot s^{d(d+3)},$$

then

$$|\mathbf{E}_{\mu}[F] - \mathbf{E}[F]| < \varepsilon(s, d),$$

where $\varepsilon(s, d) = 0.82^s \cdot (10m)$.

In particular, by taking $s = 5 \log \frac{10m}{\varepsilon}$, we get the following:

Corrolary 2. *$r(m, d, \varepsilon)$-independence ε-fools depth-d AC^0 circuits of size m, where*

$$r(m, d, \varepsilon) = 3 \cdot 60^{d+3} \cdot (\log m)^{(d+1)(d+3)} \cdot \left(5 \log \frac{10m}{\varepsilon}\right)^{d(d+3)} = \left(\log \frac{m}{\varepsilon}\right)^{O(d^2)}.$$

Note that by choosing $\varepsilon = 2^{-n^{\delta}}$ for a small $\delta = \delta(d)$, one sees that polynomial independence fools AC^0 circuits up to an exponentially small error. The results carry meaning for super-constant d's up to $d = \tilde{O}(\sqrt{\log m})$.

The original conjecture by [LN90] was that for constant ε, $r(m, d, \varepsilon) = O((\log m)^{d-1})$. Thus our results leave a gap between $O(d)$ and $O(d^2)$ in the exponent. We believe that the conjecture is true with $O(d)$.

1.3. **Techniques and proof outline.** As in [Baz09], our strategy is to approximate F with low degree polynomials over $\mathbb{R}$. The reason being that degree-r polynomials are completely fooled by r-independence.

Proposition 3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a degree-r polynomial, and let μ be an r-independent distribution. Then f is completely fooled by μ:*

$$\mathbf{E}_{\mu}[f] = \mathbf{E}[f].$$

Proposition 3 is true by linearity of expectation, since every term of f is a product of $\leq r$ variables, whose distribution is uniform under μ.

In our construction we combine two types of approximations of AC^0 circuits by low degree polynomials over $\mathbb{R}$. The first one is combinatorial in the spirit of [Raz87, Smo87, BRS91, Tar93] (for a comprehensive survey on polynomials in circuit complexity see e.g. [Bei93]). These approximating polynomials agree with F on all but a small fraction of inputs. Thus for such a polynomial f, $\mathbf{P}[f = F]$ is very close to 1. While essentially using the same construction as [BRS91, Tar93], utilizing tools from [VV85], we repeat the construction from scratch, since we want to reason about details of the construction. We believe that any construction in this spirit would fit in our proof.

The second approximation is based on Fourier analysis and uses [LMN93] where it is shown that any AC^0 function G can be approximated by a low degree polynomial g so that the ℓ_2 norm $\|G - g\|_2^2$ is small. There is no guarantee, however, that g agrees with G on any input (most likely, it doesn't).

We use an approximation f of F of the first type as the starting point of our construction. Thus $\mathbf{P}[f \neq F]$ is very small. If we knew that $\|F - f\|_2^2$ is small we would be done by a simple argument similar to one that appeared in [Baz09]. Unfortunately, there are no guarantees, that f is close to F *on average*, since f may deviate wildly on points where $f \neq F$ (in fact, it is likely untrue that $\|F - f\|_2^2$ is small).

Our key insight is that in the construction of f, the indicator function $\mathcal{E}$ of where f fails to agree with F is an AC^0 function itself. Thus $\mathcal{E} = 1$ whenever $f \neq F$, and $\mathbf{P}[\mathcal{E} = 1]$ is very small (since $f = F$ most of the time). We then use a low-degree approximation $\tilde{\mathcal{E}}$ of $\mathcal{E}$ of the second type so that $\|\tilde{\mathcal{E}} - \mathcal{E}\|_2^2$ is very small. We then take $f' = f \cdot (1 - \tilde{\mathcal{E}})$. The idea is that $1 - \tilde{\mathcal{E}} \approx 1 - \mathcal{E}$ will kill the values of f where it misbehaves (and thus $\mathcal{E} = 1$), while leaving other values (where $\mathcal{E} = 0$) almost unchanged. Note that the values where $f = 0$ remain completely unchanged, and thus f' is a semi-exact approximation of F. We then show that $\|F - f'\|_2^2$ is small. We choose f' to "almost agree" with F against both the uniform distribution and the distribution μ, a property we use to finish the proof.

It should be noted that while an inductive proof on the depth d of F is a natural approach to the problem, a non-inductive construction appears to yield much better parameters for the theorem.

References

[Baz07] L. M. J. Bazzi, *Polylogarithmic independence can fool DNF formulas*, Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), IEEE Computer Society Washington, DC, USA, 2007, pp. 63–73.

[Baz09] ———, *Polylogarithmic independence can fool DNF formulas*, SIAM Journal on Computing (SICOMP) (2009), to appear.

[Bei93] R. Beigel, *The polynomial method in circuit complexity*, Proceedings of the 8th IEEE Structure in Complexity Theory Conference, 1993, pp. 82–95.

[BRS91] R. Beigel, N. Reingold, and D. Spielman, *The perceptron strikes back*, Proceedings of the Sixth Annual Structure in Complexity Theory Conference, 1991, pp. 286–291.

[Bra09] M. Braverman, *Poly-logarithmic independence fools AC^0 circuits*, Proceedings of Complexity'09, 2009.

[LMN93] N. Linial, Y. Mansour, and N. Nisan, *Constant depth circuits, Fourier transform, and learnability*, Journal of the ACM (JACM) **40** (1993), no. 3, 607–620.

[LN90] N. Linial and N. Nisan, *Approximate inclusion-exclusion*, Combinatorica **10** (1990), no. 4, 349–365.

[Raz87] A. A. Razborov, *Lower bounds on the size of bounded-depth networks over a complete basis with logical addition*, Math. Notes Acad. Sci. USSR **41** (1987), no. 4, 333–338.

[Raz08] ———, *A simple proof of Bazzi's theorem*, Electronic Colloquium on Computational Complexity. Report TR08-081, 2008.

[Smo87] R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, Proceedings of the nineteenth annual ACM Symposium on Theory of Computing (STOC'87), ACM New York, NY, USA, 1987, pp. 77–82.

[Tar93] J. Tarui, *Probabilistic polynomials, AC^0 functions and the polynomial-time hierarchy*, Theoretical computer science **113** (1993), no. 1, 167–183.

[VV85] L. G. Valiant and V. V. Vazirani, *NP is as easy as detecting unique solutions*, Proceedings of the seventeenth annual ACM Symposium on Theory of Computing (STOC'85), ACM New York, NY, USA, 1985, pp. 458–463.

Overview of Geometric Complexity Theory

Peter Bürgisser

(joint work with M. Christandl, C. Ikenmeyer, J.M. Landsberg, L. Manivel, J. Weyman)

In a seminal work, Valiant [11] proposed in 1979 an algebraic analogue of the P versus NP question in terms of the permanent versus determinant question (VP versus VNP). We outline Mulmuley and Sohoni's GCT program [7, 8] towards resolving this question, based on the overview [3], and also discuss some new results [1]. The GCT program attempts to resolve the VP versus VNP question by reformulating it as specific orbit closure problems, that are then analyzed by means of tools of geometric invariant theory and representation theory.

1. Obstructions to orbit closure problems

The group GL_{n^2} acts on the space of homogeneous complex polynomials of degree n in the variables x_{ij} by substitution. Consider $\det_n = \det[x_{ij}]_{i,j\le n}$ and $\mathrm{per}_m = \mathrm{per}[x_{ij}]_{i,j\le m}$, where $m < n$ and $z := x_{1m+1}$. The following main conjecture is a variant of VP $\neq$ VNP:

Conjecture 1 ([7])**.** *$z^{n-m}\mathrm{per}_m \in \overline{\mathrm{GL}_{n^2}\cdot \det_n}$ is impossible for $n = m^{O(1)}$.*

If $z^{n-m}\mathrm{per}_m \in \overline{\mathrm{GL}_{n^2}\cdot \det_n}$, then we have a surjective GL_{n^2}-module morphism

$$\mathbb{C}[\overline{\mathrm{GL}_{n^2}\cdot \det_n}] \to \mathbb{C}[\overline{\mathrm{GL}_{n^2}\cdot z^{n-m}\mathrm{per}_m}].$$

Hence, by Schur's lemma, any Weyl module $V_\lambda^*(\mathrm{GL}_{n^2})$ occuring as a submodule on the right-hand side must also occur on the left-hand side.

By a *representation theoretic obstruction* we understand a partition λ such that $V_\lambda^*(\mathrm{GL}_{n^2})$ occurs on the right-hand side, but not on the left-hand side. In [8] some evidence is given that, if Conjecture 1 is true, then it can in fact be shown by exhibiting obstructions. In order to actually prove this, one would need to settle the *separation conjecture* [8, Conj. 12.4] on the "separating power" of Kronecker coefficients.

2. Stability

The following result is a consequence of the Hilbert-Mumford-Kempf criterion for stability.

Theorem 1 ([7]). *The* SL_{n^2}*-orbit of* $\det_n$ *is closed. Similarly for* per_n.

This implies that when focusing on SL_{n^2}-modules and ignoring multiplicities, orbit closures can be replaced by orbits.

Corollary 1. $V_\lambda^*(\mathrm{SL}_{n^2})$ *occurs in* $\mathbb{C}[\overline{\mathrm{GL}_{n^2}\cdot \det_n}]$ *if and only if it occurs in* $\mathbb{C}[\mathrm{SL}_{n^2}\cdot \det_n]$. *Similarly for* per_n.

3. Symmetries of determinant and permanent

A result by Frobenius states that the stabilizer of $\det_n$ is the subgroup of GL_{n^2} consisting of the following maps:

$$X \mapsto AXB \quad \text{or } X \mapsto AX^TB, \quad A,B \in \mathrm{GL}_n,\ \det(AB)=1.$$

The stabilizer of per_n is generated by the above maps where A, B are both diagonal with $\det(AB) = 1$ or both permutation matrices [6]. Moreover, it can be shown that $\det_n$ and per_n are characterized by their stabilizers.

4. Kronecker coefficients

The *Kronecker coefficient* $g_{\lambda\mu\nu}$ can be defined as the multiplicity of the irreducible $\mathrm{GL}_n \times \mathrm{GL}_n$-module $V_\mu(\mathrm{GL}_n) \otimes V_\nu(\mathrm{GL}_n)$ in $V_\lambda(\mathrm{GL}_{n^2})$ via the morphism $\mathrm{GL}_n \times \mathrm{GL}_n \to \mathrm{GL}(\mathbb{C}^n \otimes \mathbb{C}^n) \simeq \mathrm{GL}_{n^2},\ (A,B) \mapsto A \otimes B$. Describing Kronecker coefficients is a classical problem: but little is known.

Conjecture 2 ([10]). *Deciding whether* $g_{\lambda\mu\nu} \neq 0$ *can be done in polynomial time.*

For a special case of Kronecker coefficients, the so-called Littlewood-Richardson coefficients, this is known. It follows [9] from their saturation property [5] and the fact that linear programming can be solved in polynomial time. A combinatorial polynomial time algorithm for deciding positivity of Littlewood-Richardson coefficients, based on optimizing flows in networks, was developed in [2].

5. How the coordinate ring of an orbit splits

Let G be a reductive group (e.g. GL_N or SL_N or products thereof) and H be a closed subgroup of G. Note that G/H is isomorphic to the G-orbit of a point with stabilizer H. The algebraic Peter Weyl Theorem implies that

$$\mathbb{C}[G/H] \simeq \bigoplus_\lambda V_\lambda^*(G)^{\oplus \dim V_\lambda(G)^H}, \tag{1}$$

where $V_\lambda(G)^H$ denotes the space of H-invariants in $V_\lambda(G)$. Hence $V_\lambda^*(G)$ occurs in $\mathbb{C}[G/H]$ iff $V_\lambda(G)$ contains a nonzero H-invariant.

6. Modules in the orbit closure of the determinant

The stabilizer of $\det_n$ consists essentially of the maps $A \otimes B$ with $A, B \in \mathrm{GL}_n$ such that $\det(AB) = 1$. Combining this insight with Corollary 1, Equation (1), and using the fact that $V_\mu(\mathrm{GL}_n)$ is 1-dimensional iff μ has the rectangular shape $\square := (n, \ldots, n)$, one can derive the following.

Theorem 2. *$V_\lambda^*(\mathrm{SL}_{n^2})$ does not occur in $\mathbb{C}[\overline{\mathrm{GL}_{n^2} \cdot \det_n}]$ iff $g_{\lambda\square\square} = 0$.*

Hence obstructions λ must satisfy $g_{\lambda\square\square} = 0$. We call such λ "candidates for obstructions".

7. Asymptotic study of Kronecker coefficients

We normalize a partition $\lambda \vdash_n |\lambda|$ to obtain the probability distribution $\overline{\lambda} := \frac{1}{|\lambda|}\lambda$. Let $u = (\frac{1}{n}, \ldots, \frac{1}{n})$ denote the uniform distribution on $[n]$. It is a well-known fact that

$$\mathrm{Kron}(n_1, n_2, n_3) := \left\{(\overline{\lambda_1}, \overline{\lambda_2}, \overline{\lambda_3}) : \exists D \ s.t. \ \lambda_i \vdash_{n_i} D, \ g_{\lambda_1\lambda_2\lambda_3} \neq 0\right\}$$

is a rational polytope. It can be interpreted as a *moment polytope*, a concept of symplectic geometry and geometric invariant theory.

Theorem 3 ([1])**.** *We have $(r, u, u) \in \mathrm{Kron}(n^2, n, n)$ for any nonincreasing probability distribution $r \in \mathbb{Q}^{n^2}$.*

This is bad news: it shows that candidates for obstructions are rare. One can rephrase Theorem 3 by saying that the moment polytopes of SL_{n^2}-modules in the the orbit closure of the determinants do not provide any information. It can be shown that the same holds true for the permanents. Hence moment polytopes seem a too rough description for the separation goals of geometric complexity theory.

The proof of Theorem 3 relies on a recently discovered relation of $\mathrm{Kron}(n_1, n_2, n_3)$ to the *quantum marginal problem*, a problem of quantum information theory [4].

8. Modules in the orbit closure of the permanent

Finding obstructions requires to exhibit irreducible modules in the coordinate ring of the orbit closure of $z^{n-m}\mathrm{per}_m$. In the case $n = m$ we have:

Theorem 4. *$V_\lambda^*(\mathrm{SL}_{n^2})$ occurs in $\mathbb{C}[\overline{\mathrm{GL}_{n^2} \cdot \mathrm{per}_n}]$ iff $|\lambda| = \delta n$ for some δ and there exist $\mu, \nu \vdash_n |\lambda|$ such that $g_{\lambda\mu\nu} \neq 0$ and $V_\mu(\mathrm{SL}_{n^2})$ and $V_\mu(\mathrm{SL}_{n^2})$ both occur in the plethysm $\mathrm{Sym}^n(\mathrm{Sym}^\delta \mathbb{C}^n)$.*

In the case $n > m$ the following holds (which also follows from [12]).

Theorem 5 ([8])**.** *$\mathbb{C}[\overline{\mathrm{GL}_{n^2} \cdot z^{n-m}\mathrm{per}_m}]$ has the same "types" of irreducible representations as $\mathbb{C}[\overline{\mathrm{GL}_{m^2+1} \cdot z^{n-m}\mathrm{per}_m}]$.*

One of the biggest difficulty in finding obstructions (besides Kronecker coefficients and plethysms) is that we currently only have weak information on how to relate the irreducible modules in $\mathbb{C}[\overline{\mathrm{GL}_{m^2+1} \cdot z^{n-m}\mathrm{per}_m}]$ with those in $\mathbb{C}[\overline{\mathrm{GL}_{m^2} \cdot \mathrm{per}_m}]$.

References

[1] P. Bürgisser, M. Christandl, and C. Ikenmeyer, *Nonvanishing of Kronecker coefficients for rectangular shapes*, (2009) arXiv 0910.4512.

[2] P. Bürgisser and C. Ikenmeyer, *A max-flow algorithm for positivity of Littlewood-Richardson coefficients*, FPSAC 2009, Hagenberg, Austria, DMTCS proc. AK (2009), pp. 267-278.

[3] P. Bürgisser, J.M. Landsberg, L. Manivel, and J. Weyman, *An overview of mathematical issues arising in the Geometric complexity theory approach to VP v.s. VNP*, (2009), arXiv 0907.2850.

[4] A. Klyachko, *Quantum marginal problem and representations of the symmetric group*, 5(2004), quant-ph/0409113.

[5] A. Knutson and T. Tao, *The honeycomb model of GL(n,C) tensor products. I. Proof of the saturation conjecture*, J. Amer. Math. Soc. **12(4)** (1999), 10551090.

[6] M. Marvin and F.C. May, *The permanent function*, Canad. J. Math. **14** (1962), 177–189.

[7] K. Mulumuley and M. Sohoni, *Geometric Complexity Theory I: An Approach to the P vs. NP and related problems*, SIAM J. Comput. **31** (2001), 496–526.

[8] K. Mulumuley and M. Sohoni, *Geometric Complexity Theory II: towards explicit obstructions for embeddings among class varieties*, SIAM J. Comput. **38** (2008), 1175–1206.

[9] K. Mulmuley and M. Sohoni, *Geometric complexity theory III: On deciding positivity of Littlewood-Richardson coefficients*, cs.ArXive preprint (2005), cs.CC/0501076.

[10] K. Mulmuley,*Geometric complexity theory VI: The flip via saturated and positive integer programming in representation theory and algebraic geometry*, Technical Report TR-2007-04 (2007), Computer Science Department, The University of Chicago.

[11] L.G. Valiant, *Completeness classes in algebra*, Proc. 11th ACM STOC (1979), 249–261.

[12] J. Weyman, *Cohomology of vector bundles and syzygies*, Cambridge Tracts in Mathematics 149 (2003), Cambridge University Press.

Composition of low-error 2-query PCPs using decodable PCPs

Irit Dinur
(joint work with Prahladh Harsha)

Probabilistically checkable proofs (PCPs) provide a proof format that enables verification with only a constant number of queries into the proof. The celebrated PCP Theorem [AS98, ALM$^+$98] states that every language in NP has a verifier that can always be convinced of a correct statement and will reject with some probability $1 - \delta$ a proof of a false statement. Most importantly, this can be done while using only a logarithmic number of random coins, and reading only $q = O(1)$ proof bits. Naturally, (and motivated by the fruitful connection to inapproximability due to [FGL$^+$96]), much attention has been given to obtaining PCPs with "good" parameters, such as $q = 2$, smallest possible soundness error δ, and smallest possible alphabet size $|\Sigma|$. These are the parameters of focus in this work.

Proof composition is an essential ingredient of all known constructions of PCPs. Composition of PCPs with high soundness error (greater than 1/2) is by now well understood using the notion of *PCPs of proximity* [BGH$^+$06] (called *assignment testers* in [DR06]) (see also [Sze99]). These allow for modular composition, in the high soundness error regime which in turn led to alternate proofs of the PCP Theorem and constructions of shorter PCPs [BGH$^+$06, Din07, BS08]. However, these composition theorems are inapplicable when constructing PCPs with low-soundness error (arbitrarily small soundness error or even any constant less than 1/2).

Our first contribution is a definition of an object which we call a *decodable PCP*, which allows for clean and modular composition in the low error regime.

Decodable PCPs (dPCPs). Consider a probabilistically checkable proof for the language CircuitSat (the language of all satisfiable circuits). The natural NP proof for CircuitSat is simply a satisfying assignment. An intuitive way to construct a PCP for CircuitSat is to *encode* the assignment in a way that enables probabilistic checking. This intuition guides all known constructions, although it is not stipulated in the definition.

In this work, we make the intuitive notion of proof encoding explicit by introducing the notion of a *decodable PCP (dPCP)*. A dPCP for CircuitSat is an encoding of the satisfying assignment that can be both verified and decoded locally in a probabilistic manner. In this setting, the verifier is supposed to both verify that the dPCP is encoding a *satisfying* assignment, as well as to decode a symbol in that assignment. More precisely, we define a *PCP decoder* for CircuitSat to be a probabilistic algorithm that is given an input circuit C, oracle access to a dPCP π, and, in addition, an index i. Based on C, i and the randomness r it computes a window I and a *function* f (rather than a predicate). This function is supposed to evaluate to the i-th symbol of a satisfying assignment for C; or to reject.

- The PCP decoder is *complete* if for every y such that $C(y) = 1$ there is a dPCP π such that $\Pr_{i,I,f}[f(\pi_I) = y_i] = 1$.
- The PCP decoder has *soundness error* δ and list size L if for any (purported) dPCP π there is a list of $\leq \mathsf{L}$ valid proofs such that the probability (over the index i and (I, f)) that $f(\pi_I)$ is inconsistent with the list but does not reject is at most δ.

The list of valid proofs can be viewed as a "list decoding" of the dPCP π. Since we are interested in the low soundness error regime, list-decoding is unavoidable.

Composition. There is a natural and modular way to compose a PCP verifier V with a PCP decoder D. The composed PCP verifier V' begins by simulating V on a probabilistically checkable proof Π. It determines a set of queries into Π (a local window I), and a local predicate f. Instead of directly querying Π and testing if $f(\Pi_I) = 1$, V' relies on the inner PCP decoder D to perform this action. For this task, the inner PCP decoder D is supplied with a dedicated proof that is supposedly an encoding of the relevant local view Π_I. The main issue is consistency: the composed verifier V' must ensure that the dedicated proofs supposedly encoding the various local views are consistent with the same Π (i.e. they should be encodings of local views coming from a single valid PCP for V). This is achieved easily with PCP decoders: the composed verifier V' asks D to decode a random value from the encoded local view, and compares it to the appropriate symbol in Π.

Two Query Composition. Our main contribution is a composition theorem that does not incur an extra query. The extra query above comes from the need to check that all the inner PCP decoders decode to the same symbol. This check was performed by comparing the decoded symbol to the symbol in the outer PCP Π. Instead, we verify consistency by invoking *all* the inner PCP decoders that involve this symbol *in parallel*, and then checking that they all decode to the same symbol. This avoids the necessity to query the outer PCP Π for this symbol and saves us the extra query.

Hardness of Label Cover, and the [MR08] **result.** In a recent breakthrough, Moshkovitz and Raz [MR08] constructed almost linear-sized low-error 2-query PCPs for every language in NP. Their result strengthens a large number of inapproximability results through a standard reduction from an intermediate problem called label cover.

The main technical component of their construction is a novel composition of certain specific PCPs. We give a modular and simpler proof of their result by repeatedly applying the new composition theorem to known PCP components.

References

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. (Preliminary Version in *33rd FOCS*, 1992).

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, January 1998. (Preliminary Version in *33rd FOCS*, 1992).

[BGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. *SIAM J. Computing*, 36(4):889–974, 2006. (Preliminary Version in *36th STOC*, 2004).

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Computing*, 38(2):551–607, 2008. (Preliminary Version in *37th STOC*, 2005).

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. (Preliminary Version in *38th STOC*, 2006).

[DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP Theorem. *SIAM J. Computing*, 36:975–1024, 2006. (Preliminary Version in *45th FOCS*, 2004).

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, March 1996. (Preliminary version in *32nd FOCS*, 1991).

[MR08] Dana Moshkovitz and Ran Raz. Two query PCP with sub-constant error. In *Proc.* 49*th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 314–323. IEEE, 2008.

[Sze99] Mario Szegedy. Many-valued logics and holographic proofs. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Proc. 26th International Colloquium of Automata, Languages and Programming (ICALP)*, volume 1644 of *LNCS*, pages 676–686. Springer, 1999.

Kakeya sets and Extractors

Zeev Dvir
(joint work with S. Kopparty, S. Saraf, M. Sudan and A. Wigderson)

Let $\mathbb{F}$ denote a finite field of size q. A set $K \subset \mathbb{F}^n$ is called a *Kakeya* (the term *Besicovitch set* is also used in the literature) if it contains a line in every direction. More formally, if for every (direction) $b \in \mathbb{F}^n$ there exists a point $a \in \mathbb{F}^n$ such that the set $\{a + t \cdot b \,|\, t \in \mathbb{F}\}$ is contained in K. In a survey paper, Wolff [Wol99] made a conjecture about the size of such sets.

Conjecture 1 (The finite field Kakeya conjecture [Wol99])**.** *Let $K \subset \mathbb{F}^n$ be a Kakeya set, then*

$$|K| \geq C_n \cdot q^n,$$

where C_n is a constant depending only on n.

This conjecture originates from the famous Euclidean Kakeya conjecture which deals with bounding the dimension of sets in $\mathbb{R}^n$ containing a unit line segment in every direction. This natural question on the geometry of finite fields was posed by Wolff as a 'stripped down' version of its Euclidean sibling on which new ideas could be tested without having to deal with the technical difficulties of Euclidean geometry.

Until recently, progress on the finite field Kakeya problem and on the Euclidean problem went hand-in-hand. The best bounds for both problems were obtained using a technique of Bourgain [Bou99] (later improved in [KT02]) which uses tools from additive combinatorics. These techniques (which are still the most effective for the Euclidean problem) give a lower bound of $\approx q^{\frac{4}{7}n}$ on the size of K [Rog01, MT04]. We note that a bound of the form $|K| \geq q^{n/2}$ can be easily obtained by observing that the difference set $K - K$ is equal to the whole space. Recently, the finite field Kakeya conjecture was proved [Dvi08]. The proof is based on the observation that a polynomial of degree less than q that vanishes on a Kakeya set must vanish identically.

Theorem 1 ([Dvi08])**.** *Let $K \subset \mathbb{F}^n$ be a Kakeya set, then*

$$|K| \geq \frac{1}{n!} \cdot q^n.$$

The finite field Kakeya problem originated independently in the quest for constructing functions with 'special' properties used in theoretical computer science. These functions, called randomness extractors (or just extractors for short), play an important role in the proofs of many results on a large number of topics including de-randomization (the relation between deterministic and randomized algorithms), error correcting codes, cryptography and many others.

Roughly speaking, an extractor is a function that 'extracts' randomness from arbitrary random distributions, with the help of a short random seed. It is known that a random function will be, with overwhelming probability, an extractor with the best possible parameters. The challenge is therefore, not to show that good extractors exists, but rather to give explicit (efficiently computable) constructions, matching the parameters of a random construction.

Mergers are similar to extractors in the sense that they are functions that extract randomness from weak distributions. However, unlike extractors, they relax two of the conditions on the input and output distributions. The first relaxation is a structural condition on the input X. Instead of being an arbitrary distribution (with high min entropy), X is now divided into s blocks $X_1, \ldots, X_s$, each of length n bits, and we are guaranteed that one of these blocks is uniform (the dependencies between the blocks can be arbitrary). This type of source is referred to in the literature as a 'somewhere-random source'. The second relaxation is that, instead of requiring the output, another n-bit string, to be close to uniform, we only require it to have very high min-entropy (say, at least $\frac{9}{10}n$). As is the case with extractors, mergers have to rely on an additional short random seed.

Stated more formally, a merger is a function

$$M : (\{0,1\}^n)^s \times \{0,1\}^d \mapsto \{0,1\}^n$$

such that if $X = (X_1, \ldots, X_s)$ is a random variable on $(\{0,1\}^n)^s$ for which one of the X_i's is uniform, then $M(X, U_d)$ has (up to some small statistical error) min entropy at least $\frac{9}{10}n$ (the choice of constant $\frac{9}{10}$ is arbitrary). It was shown in [TS96, NTS99] that explicit constructions of good mergers (for a large number of blocks) imply good constructions of extractors and so the task of building good mergers became one of equal interest to that of building extractors.

In [DW08] a new merger was constructed that makes use of the fact that the proof technique used in [Dvi08] to bound the size of Kakeya sets can be applied just as efficiently to control intersections of low degree *curves* over finite fields. roughly speaking, the merger passes a low degree curve through the s points $X_1, \ldots, X_s \in \mathbb{F}^r$ and outputs a random point on this curve. It was proved in [DW08] that this construction gives a good merger and that this merger can be used (in conjunction with other results) to give good extractors. A better analysis of the merger was given in [DKSS09].

Theorem 2 ([DKSS09])**.** *The output of the merger described above is ϵ-close (in statistical distance) to having min entropy at least $(1-\delta)\cdot n$, whenever*

$$q \geq \left(\frac{2 \cdot s}{\epsilon}\right)^{\frac{1}{\delta}}.$$

As a result of the merger analysis of Theorem 2, a new extractor construction was given in [DKSS09] with parameters that were not obtainable using previous methods.

The techniques developed in [DKSS09], which include the use of high degree polynomials (compared to relatively low degree in previous works) give, in particular, the strongest known bound on the size of Kakeya sets which is within a factor of 2 of the known upper bounds.

Theorem 3 ([DKSS09]). *Let $K \subset \mathbb{F}^n$ be a Kakeya set. Then*

$$|K| \geq \frac{1}{2^n} \cdot q^n.$$

References

[Bou99] J. Bourgain. On the dimension of Kakeya sets and related maximal inequalities. *Geom. Funct. Anal.*, 9(2):256–282, 1999.

[DKSS09] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. In *FOCS 09 (to appear)*, 2009.

[Dvi08] Z. Dvir. On the size of Kakeya sets in finite fields. *J. AMS (to appear)*, 2008.

[DW08] Z. Dvir and A. Wigderson. Kakeya sets, new mergers and old extractors. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 625–633, Washington, DC, USA, 2008. IEEE Computer Society.

[KT02] N. Katz and T. Tao. New bounds for Kakeya problems. *Journal d'Analyse de Jerusalem*, 87:231–263, 2002.

[MT04] G. Mockenhaupt and T. Tao. Restriction and Kakeya phenomena for finite fields. *Duke Math. J.*, 121:35–74, 2004.

[NTS99] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58, 1999.

[Rog01] K.M Rogers. The finite field Kakeya problem. *Amer. Math. Monthly 108*, (8):756–759, 2001.

[TS96] A. Ta-Shma. *Refining Randomness.* PhD thesis, The Hebrew Univerity, Jerusalem, Israel, 1996.

[Wol99] T. Wolff. Recent work connected with the Kakeya problem. *Prospects in mathematics (Princeton, NJ, 1996)*, pages 129–162, 1999.

Locally decodable codes of subexponantial length

Klim Efremenko

1. Locally decodable codes

Locally decodable codes (LDCs) are codes that allow to retrieve any symbol of the original message by reading only a constant number of symbols from the codeword. Formally a code C is said to be locally decodable with parameters (q, δ, ε) if it is possible to recover any bit x_i of message x by making at most q queries to $C(x)$. Such that if up to a δ fraction of $C(x)$ is corrupted then the decoding algorithm will return the correct answer with probability at least $1 - \varepsilon$.

Locally decodable codes have many applications in cryptography and complexity theory, see surveys in [Tre04] and [Gas04]. The first formal definition of locally decodable codes was given by Katz and Trevisan in [KT00]. The Hadamard code is the most famous 2-query locally decodable code of length 2^n. For a two-query LDC tight lower bounds of $2^{\theta(n)}$ were given for linear codes in [GKST02] and [KdW03] proved tight lower bounds for two queries for arbitrary codes. For an arbitrary number of queries Katz and Trevisan [KT00] established super-linear lower bounds of $\Omega(n^{q/(q-1)})$ for LDCs with q queries. This lower bound was later improved in [KdW03] to $\Omega\left((\frac{n}{\log n})^{1+1/(\lceil q/2\rceil-1)}\right)$ and in [Woo07] to $\Omega\left(\frac{n^{1+1/(\lceil q/2\rceil-1)}}{\log n}\right)$.

For many years it was conjectured that LDCs should have an exponential dependence on n for any constant number of queries, until Yekhanin's recent breakthrough [Yek08]. Yekhanin obtained 3-query LDCs with sub-exponential length of $\exp(\exp(O(\frac{\log n}{\log\log n})))$ under a highly believable conjecture that there are infinitely many Mersenne primes. Using the known Mersenne primes, Yekhanin also obtained unconditional results which significantly improved the previous results on LDCs(i.e. length of $\exp(n^{10^{-7}})$). In [KY08] Kedlaya and Yekhanin proved that infinitely many Mersenne numbers with large prime factors are essential for Yekhanin's construction. Due to the best of our knowlage Yekhanin's construction can not generalized for higher number of queries.

Our Results. In this paper we give an unconditional construction of 3-query LDC with sub-exponential codeword length. The length that we achieve for 3 queries is:

$$\exp\exp(O(\sqrt{\log n\log\log n})).$$

We also give a 2^r-query LDC with a codeword length $\exp\exp(O(\sqrt[r]{\log n(\log\log n)^{r-1}}))$.

Our construction is a kind of a generalization and simplification of [Yek08]. We extend Yekhanin's construction to work not only with primes but also with composite numbers. Raghavendra in [Rag07] gives a nice presentation of Yekhanin's construction using homomorphisms, and we will follow this approach. The main ingredient in our construction is the Grolmusz construction [Gro00] of super-polynomial size set-systems with restricted intersections over composite numbers.

Private Information Retrieval schemes: The notion of locally decodabale codes is closely related to the notion of private information retrieval(PIR) schemes. PIR schemes with k servers is a protocol which allows for a user to access a database distributed between k servers without yielding any information on the identity of the accessed place to any individual server (we assume that there is no communication between servers). The main parameter of interest in PIR schemes is the total communication complexity between the user and the servers. PIR schemes were first introduced by [CGKS95]. After that there were many works written on this topic, see [CGKS95, Amb97, Man98, Ito99, BIK05, GKST06, KdW03, RY07, WdW05, Yek08]. The best upper bound for 2-server PIR is $O(n^{1/3})$ due to [CGKS95]. The best upper bound of 3 and more server PIR schemes is $\exp\left(O\left(\frac{\log n}{(\log\log n)^{1-\varepsilon}}\right)\right)$ due to [Yek08] which is based on the construction of LDCs.

Let us define formally perfect PIR schemes:

Definition 4. A one-round perfect *private information retrieval* scheme is a randomized algorithm $\mathcal{U}$ (for the user), and k deterministic algorithms $\mathcal{S}_1, \dots \mathcal{S}_k$ (for the servers), s.t.

(1) (a) On input $i \in [n]$ the user $\mathcal{U}$ produces k random queries $q_1 \dots q_k$ and sends them to respective servers.
(b) Each server based on his query q_j and database $\mathcal{D}$ produces a response $r_j = \mathcal{S}_i(\mathcal{D}, q_j)$ and sends it back to the user.
(c) The user based on $i, r_1, \dots, r_k$ and his randomness calculates $\mathcal{D}[i]$.

(2) The distribution of each query q_j is independent of the input i.

The communication complexity of this protocol is a total number of bits exchanged between user and servers.

It is well known that LDCs with perfectly smooth decoder imply PIR schemes. In particular, as in [Yek08], our LDC yields a PIR schemes with communication complexity $\exp(O(\sqrt{\log n \log\log n}))$ for 3-servers and $\exp(O(\sqrt[r]{\log n (\log\log n)^{r-1}}))$ for 2^r-servers.

1.1. **Future work.** In this paper we give a general construction of LDCs from any S-matching set and S-decoding polynomial. Any improvement in size of a set-system with restricted intersections will immediately yield improvement in the rate of LDCs. We hope that this paper will give a motivation for future work on set-systems with restricted intersections. We also believe that it is possible to choose an S-decoding polynomial with less monomials.

References

[Amb97] Andris Ambainis. Upper bound on communication complexity of private information retrieval. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407. Springer, 1997.

[BIK05] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. Comput. Syst. Sci.*, 71(2):213–247, 2005.

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *FOCS*, pages 41–50, 1995.

[Gas04] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

[GKST02] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *IEEE Conference on Computational Complexity*, pages 175–183, 2002.

[GKST06] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[Gro00] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20(1):71–86, 2000.

[Ito99] Toshiya Itoh. Efficient private information retrieval. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E82-A No.1 pp.11-20*, 1999.

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *STOC*, pages 106–115. ACM, 2003.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

[KY08] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of mersenne numbers. In *IEEE Conference on Computational Complexity*, pages 175–186. IEEE Computer Society, 2008.

[Man98] Eran Mann. Private access to distributed information. In *Master's thesis, Technion - Israel Institute of Technology*, 1998.

[Rag07] Prasad Raghavendra. A note on yekhanin's locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 2007.

[RY07] Alexander A. Razborov and Sergey Yekhanin. An omega($^{1/3}$) lower bound for bilinear group based private information retrieval. *Theory of Computing*, 3(1):221–238, 2007.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, (043), 2004.

[WdW05] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *ICALP*, pages 1424–1436, 2005.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 2007.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008.

A Parallel Repetition Theorem for Any Interactive Argument

Iftach Haitner

In an interactive proof, a prover P is trying to convince the verifier V in the validity of some statement. Typically, P has some advantage over V, such as additional computational resources or some extra information (e.g., an NP witness that validates the claim). The two basic properties we would like such protocols to have are *completeness* and *soundness*. The completeness means that P convinces V to accept *valid* statements, and the soundness means that no cheating prover (of a certain class) can convince V to accept *invalid* statements. More generally, (P, V) has completeness β if for any valid statement x, V accepts in $(P, V)(x)$ with probability at least β (where P typically gets an advice $w(x)$ as an additional input). Where V has soundness $1 - \epsilon$ with respect to a given class of algorithms, if no malicious P^* from this class can convince V to accept an invalid statement with probability greater than ϵ. The bound ϵ is typically called the *soundness error* of the protocol.

The basic distinction one may make about the soundness of a given protocol, is whether it holds unconditionally (i.e., even an all-powerful prover cannot break the soundness) or that it only holds against computationally bounded (uniform, or non-uniform) provers. Protocols with unconditional soundness are called *interactive proofs*, whereas protocols with the weaker type of soundness are called *interactive arguments*. In this work we focus on computationally bounded provers. In particular, we consider polynomial-time provers.

A common paradigm for constructing protocols with low soundness error, is to start by constructing a protocol with noticeable soundness error, and then manipulate the original protocol in a certain way that decreases its soundness error while keeping its completeness high. The most natural such manipulation that comes to mind, is to use repetition. Namely, to repeat the protocol many times (with independent randomness), where the verifier accepts only if the verifiers (of the original protocol) accept in all executions. The above repetition can be done in essentially two different ways: sequentially (known as *sequential repetition*), where the $(i+1)$ execution of the protocol is only started after the i'th execution is finished, or in parallel (known as *parallel repetition*), where all the executions are done simultaneously.

Sequential repetition is known to reduce the soundness error at an exponential rate in most computational models (cf., [3]). Unfortunately, sequential repetition has the undesired effect of increasing the round complexity. Parallel repetition on the other hand, does preserve the round complexity, and for the case of interactive proofs, it also reduces the soundness error at an exponential rate [4]. Unfortunately, as shown by Bellare, Impagliazzo and Naor [1], in the case of interactive arguments parallel repetition might not reduce the soundness error at all.

Let us be more precise about the latter statement. Parallel repetition does reduce the soundness error in the case of 3-message protocol ([1, 2, 6]) and in the case of public-coin verifiers ([7, 5]). On the negative side, for any $k \in N$ [1] presented an 8-message protocol with soundness error $\frac{1}{2}$, whose k-parallel repetition soundness remains $\frac{1}{2}$. Recently, Pietrzak and Wikström [8] gave an example of a single protocol for which the above phenomena holds for all polynomial k simultaneously. Moreover, both results extend to 4-message protocols, assuming a rather natural limitation about the soundness proof.

1. Our Result

We present a simple method for transforming any efficient interactive argument whose soundness error is bounded away from one, into an efficient interactive argument with the same number of rounds and negligible soundness error. Given an m-round interactive protocol (P, V), we define the *random-termination* variant of V, denoted by $\widetilde{V}$, as follows: through the interaction with P algorithm $\widetilde{V}$ acts exactly as V does, but with the following additional step: at the end of each round, $\widetilde{V}$ tosses an $(1 - 1/4m, 1/4m)$ biased coin (i.e., 1 is tossed with probability $1/4m$). If the outcome of the coin is 1, then $\widetilde{V}$ accepts the interaction and halts. Otherwise, $\widetilde{V}$ proceeds as V does (where in particular, at the end of the protocol, if reached, $\widetilde{V}$ accepts iff V does). Note that the completeness of $(P, \widetilde{V})$ is at least as high as the completeness of (P, V), where the soundness of $\widetilde{V}$ is at least $(1 - \frac{1}{4m})^m \cdot \alpha \geq \frac{3}{4} \cdot \alpha$, given that the soundness of V is at least α.

In the following we refer to $(P, \widetilde{V})$ as the random-termination variant of (P, V). Our main contribution is stated in the following theorem.

Theorem 1 (informal). *Parallel repetition of the random-termination variant of* any *interactive argument, reduces the soundness error at a weak exponential rate.*

We note that our result holds with respect to any interactive protocol that can be cast as an interactive argument. For instance, our result yields a round-preserving binding amplification for computationally binding commitment schemes.

Our result also extends to the more general threshold case, where the prover in the k-fold repetition is only required to make $t < k$ of the verifiers accept.

2. Our Technique

Let (P, V) be an interactive argument with soundness error ϵ and let $(P^{(k)}, V^{(k)})$ be its k'th parallel repetition. We show that if (P, V) is a random-termination variant of some protocol, then any efficient strategy ${P^{(k)}}^*$ that breaks the soundness of $(P^{(k)}, V^{(k)})$ with "too high" probability ϵ_k, implies an *efficient* algorithm P^* that breaks the soundness of (P, V) with probability higher than ϵ. As a warm up, we start by presenting such strategy for the parallel repetition of public-coin protocols (with no random-termination), and then explain how to adapt this strategy the random-termination case.

Public-coin protocols. In the following we loosely follow the approach presented by [5]. In order to interact with V, algorithm P^* emulates a random execution of $({P^{(k)}}^*, V^{(k)})$, where the "real" V plays the role of the i^*'th V, for i^* that is chosen at random from $[k]$, and P^* emulates the execution of the other $(k-1)$ verifiers and of ${P^{(k)}}^*$. In the j'th round, P^* acts as follows: upon receiving the j'th message from V, it samples at random a value $M_j = (M_{j,1}, \ldots, M_{j,k})$ for the j'th messages of emulated verifiers, and evaluates their "quality" α_{M_j} — the probability that ${P^{(k)}}^*$ makes $V^{(k)}$ accept conditioned on the current transcript and on M_j. In order to do so, P^* samples many random continuations of the protocol, and measures the fraction of accepting ones (i.e., where all the verifiers accept). If the estimated value of α_{M_j} is higher than some threshold β_j (e.g., $\beta_j = (1 - \frac{j}{4m}) \cdot \epsilon_k$, where we recall that ϵ_k is the success probability of ${P^{(k)}}^*$), then P^* sends $M^j_{i^*}$ back to the real V. In addition, P^* sets the state of the emulated verifiers and ${P^{(k)}}^*$ according to M_j. P^* keeps sampling random values for M_j until a good value is found, or until n/ϵ_k unsuccessful attempts, where in the latter case it aborts. We note that V accepts whenever P^* does not abort.

The proof that P^* breaks the soundness of (P^*, V) with high probability, goes by showing that conditioned on P^* not aborting in the j'th round, the probability that P^* abort in the $j+1$ round is small. For proving the above, it suffices to show that ${P^{(k)}}^*$'s conditional success probability after getting the $j+1$ message from the real verifier, is not much smaller than α_{M_j}. While in the worst case the latter probability might be arbitrarily small (and in particular, much smaller than α_{M_j}), using a result of Raz [9] one can show that for most values of i^*, this conditional probability is with high probability close to α_{M_j}.

Random-termination protocols. When one tries to adopt the above strategy for non public-coin protocols, he should first decide what the values of M_j and α_{M_j} stand for in this case. The first (and the more natural) option, is to choose M_j at random from the j'th *messages* of the emulated verifier that are consistent with the current *transcript*, and let α_{M_j} be the probability that ${P^{(k)}}^*$ makes $V^{(k)}$ accept conditioned on M_j and on the current transcript. The very same argument we used above for the public-coin case, yields that P^* makes V accepts with high probability also in this settings. The problem is, however, that the above strategy is not necessarily efficient. (Indeed, the task of sampling M_j and of estimating α_{M_j} using the above strategy, are essentially the task of finding a random preimage of an arbitrary function).[1]

The way we adopt the public-coin strategy for the non public-coin case is different. We assume without loss of generality that the random (private) coins that V is using in each round are chosen uniformly at random from $\{0,1\}^t$ (for some value of t that might depend on the round). In each round, P^* chooses M_j uniformly random from $\{0,1\}^{t\cdot(k-1)}$, and estimates the value of α_{M_j} defined as the probability that ${P^{(k)}}^*$ makes $V^{(k)}$ accept, conditioned on the random coins flipped by all the verifiers (emulated and real) till now, and that the random coins of the emulated verifiers in the j'th round are set to M_j. Upon finding a good value for M_j (i.e., the estimation of α_{M_j} is at least β_j), P^* fixes the random coins of the emulated verifiers in the j'th round to M_j, and sends the message that ${P^{(k)}}^*$ sends to the i^* verifier in the j'th round to V (given this fixing). As in the case of former approach, it follows that P^* makes V accepts with high probability.

On a first look, the above approach does not look very promising, as in general no strategy (even not an unbounded one) can evaluate α_{M_j}.[2] Interestingly, we show that a close variant of the above strategy can be implemented efficiently for any random-termination verifier.

Let V be a random-termination verifier and assume without loss of generality that it chooses all but its decision bits (the bits uses for deciding whether or not to terminate the executions) before the interaction starts. In order to approximate the value of α_{M_j}, P^* samples the future random coins of all the verifiers conditioned that the real verifier's decision bit in the end of the j'th round is one (i.e., it decides to halt in the end of the j'th round). Sampling in this case is very easy, since the real verifier sends no further messages, and the future random coins for the emulated verifiers (under any conditioning) are simply uniform random strings. The obvious problem with the above approach is that by adding this additional conditioning we might reduce the success probability of P^*. We prove that the latter does not happen for most choices of i^*, by proving the following stronger statement: for a given $i^* \in [k]$, consider the distribution that a random execution of $(P^*, \widetilde{V})$ described above induces on the value of $(M_1 \ldots, M_m)$ with respect to to this choice of i^* (hereafter, the "real" distribution). For such i^*, we also consider the "ideal" version of the above distribution. In this version, P^* has access to the

[1] We mention that the proofs of all interactive argument protocols for which parallel repetition is known to reduce soundness, follow (implicitly or explicitly) the above strategy. Indeed, such proofs were only given for protocols for which the above sampling strategy can be carried efficiently: public-coin protocol [5], with extensions to protocols in which the last message of the verifier (which contains its decision bit) is not necessarily efficiently samplable: 3-message protocols [1] and "extendable and simulatable" verifiers [5].

[2] The random coins that the real verifier chooses in the j'th round, might only affect the transcript on a later round. Therefore, the transcript of the protocol in the j'th round might not contain the required information for estimating α_{M_j} (recall that the value of α_{M_j} is determined by the random coined that were already flipped by the verifiers, and not by the transcript).

random coins of the i^* verifier, and uses them for approximating the values of α_{M_j} well. Our main technical contribution is showing that for most values of $i^* \in [k]$ (i.e., for $(1 - (\frac{m}{k})^{\Omega(1)})$ fraction of them), the above distributions are statistically close.

Bounding the distance between the ideal and real distributions. Let $k \geq m \cdot n^2$. For concreteness, we consider the distribution of M_1 induced by the first round of the protocol, given an arbitrary fixing of the real verifier random coins. We say that $i^* \in [k]$ has *global effect*, if by conditioning that the i^*'th verifier halts at the end of the first round, we significantly change the probability that P^* finds a good value for M_1 in a single first round iteration. We say that i^* has *local effect* on some value of M_1, if by conditioning on the i^* verifier halting at the end of the first round, we significantly change the value of α_{M_1} (recall that α_{M_1} was defined as the success probability ${P^{(k)}}^*$, conditioned that the emulated verifiers random coins in the first round are set to M_1).

We first show that the fraction of local effect indices is small for every value of M_1. Assume that the number of local effect indices on some value of M_1 is larger than $m \cdot n$. Further, assume for simplicity that by conditioning on half of these indices, we reduce the value of α_{M_1} significantly. In this case, at least one of these local high effect verifiers halts in almost every random continuation of the protocol (recall that any of the verifiers halts with probability $1/4m$). This means that the value of α_{M_1} should have been smaller than what we assume it is. A similar proof also show that the number of global effect indices is small.

In the following we assume for simplicity that every index has local effect only on a small portion of the possible values for M_1, and let i^* be an index with no global effect. It is easy to verify that the following holds in a random first round iteration of P^* with such choice of i^*: the probability that P^* picks a good value for M_1 (P^* estimates that $\alpha_{M_1} > \beta_1$) and i^* *does not have* local large effect on, is much larger than the probability that P^* picks a good value for M_1 that i^* *has* local large effect on. It follows that the probability that such choice of i^* induces on most value of M_1, is close to the probability in which each M_1 is drawn with probability $\frac{\alpha_{M_1}}{Ex_{M_1}[\alpha_{M_1}]}$. Namely, the distribution induced by i^* is close to the real distribution.

References

[1] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, 1997.

[2] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *Theory of Cryptography, Second Theory of Cryptography Conference (TCC).*

[3] Ivan B. Damgård and Birgit Pfitzmann. Sequential iteration arguments and an efficient zero-knowledge argument for NP. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 1998.

[4] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Springer-Verlag, 1999.

[5] Johan Håstad, Rafael Pass, Krzysztof Pietrzak, and Douglas Wikström. An efficient parallel repetition theorem. Unpublished manuscript, 2008.

[6] Russell Impagliazzo, Ragesh Jaiswal, and Ragesh Kabanets. Approximately list-decoding direct product codes and uniform hardness amplification. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science (FOCS)*, 2006.

[7] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. An efficient parallel repetition theorem for arthur-merlin games. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, 2007.

[8] Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. In *Theory of Cryptography, Fourth Theory of Cryptography Conference (TCC)*, 2007.

[9] Ran Raz. A parallel repetition theorem. *Journal of the ACM*, 27(3):763–803, 1998. Preliminary version in *STOC'95*.

Direct-Product Decoding and Testing

Valentine Kabanets

(joint work with Russell Impagliazzo, Ragesh Jaiswal, and Avi Wigderson)

Applications of complexity theory such as cryptography and derandomization require reliably hard problems that cannot be solved by any algorithm with a non-trivial advantage over random guessing. Direct-product theorems are a primary tool in hardness amplification, allowing one to convert problems that are somewhat hard into problems that are more reliably hard. In a direct-product theorem, we start with a function f such that any feasible algorithm has a non-negligible chance of failing to compute $f(x)$ given a random x. We then show that no feasible algorithm can, given multiple instances of the problem $x_1, \dots, x_k$, compute all of the values $f(x_i)$, with even a small probability of success. (Usually, the x_i's are chosen independently, but there are also derandomized direct-product theorems where the x_i's are chosen pseudo-randomly.) Many strong direct product theorems are known for non-uniform models, such as Boolean circuits [Yao82, Lev87, GNW95, Imp95, IW97, STV01]. Unfortunately, in general, direct-product theorems fail in completely uniform models such as probabilistic computation.

For further discussion, it will be more convenient to view direct product theorems in the language of error-correcting codes. Impagliazzo [Imp02] and Trevisan [Tre05] pointed out that proofs of direct product theorems correspond to (approximate) local error-correction of sparse codes. Using this view, we think of a function f as being encoded by $Code(f) = f^k$, its values on all k-tuples. That is, the message is the truth table of the function f, and the encoding of f is the truth table of the direct-product function f^k.[1] Given a highly corrupted encoding C' of some function f, we would like to recover f. We want *local* decoding in the sense that the decoding algorithm, given oracle access to C', should produce an efficient circuit for f (which may also use oracle access to C'). Having efficient local decoding of the direct-product code immediately translates into the hardness amplification properties of the direct-product construction. Intuitively, if the

[1] Note that if f is a Boolean function, then the message is a string over the binary alphabet $\{0,1\}$, whereas its encoding is a string over the larger alphabet $\{0,1\}^k$.

decoder can recover a small circuit computing f well on average (thereby contradicting the assumed average-case hardness of f) from a small circuit C' that has only ϵ agreement with f^k, then f^k must be hard to compute by small circuits on all but less than ϵ fraction of inputs.

A completely *uniform* decoding algorithm for the direct-product encoding $Code(f)$ is an algorithm that constructs a *single* circuit C computing f well on average, when given as input some circuit C' that agrees with f^k on a small, say ϵ, fraction of all k-tuples. When ϵ is sufficiently close to 1, e.g., if $\epsilon \geq 0.9$, then such uniform decoding is possible (and easy to analyze). However, if $\epsilon \leq 1/2$, it is easy to see that C' does not uniquely determine f. Indeed, consider $t = 1/\epsilon$ different (e.g., randomly chosen) functions $f_1, \ldots, f_t$. Partition the set of all k-tuples into t sets (of measure ϵ each). Define C' so that C' agrees with f_i^k on the k-tuples in the ith set of the partition. Then this C' has agreement ϵ with each of the $t = 1/\epsilon$ functions $f_1, \ldots, f_t$.

The example given above shows that the direct-product code $Code(f) = f^k$ is not uniquely decodable when the fraction of corrupted symbols in the codeword is at least $1/2$. In order to tolerate high corruption rates (which is the interesting case for hardness amplification), we need to allow list-decoding: Given C', a corrupted version of f^k, a decoding algorithm may output a list of circuits such that one of them computes f well on average. The list size is an important parameter that we would like to minimize. The example above shows the list size lower bound $1/\epsilon$ (for list-decoding from C' that has ϵ agreement with the function f^k which we wish to decode).

Most previously known proofs of the direct-product theorem are highly non-uniform in the sense that they yield decoding algorithms with the list size *exponential* in $1/\epsilon$. In contrast, more uniform proofs of the direct-product theorem should yield list-decoding algorithms with the list size at most polynomial in $1/\epsilon$. [IJK06] gave the first such proof of the direct-product theorem achieving the list size $poly(1/\epsilon)$; however, the proof was quite complex and fell short of the information-theoretic bounds in many respects.

We give a new uniform direct-product theorem that has the following features: (1) **Optimality:** The parameters achieved by our list decoding algorithm are information theoretically optimal (to within constant factors). In particular, the list size is $O(1/\epsilon)$, which matches the list-size lower bound given in the example above (up to a constant factor). (2) **Derandomization:** We get the first derandomized direct-product theorems in the uniform setting. A direct application of the above intersection codes to subspaces yields amplification with input size $O(n)$, instead of the trivial bound of $O(kn)$ when using all subsets.

Our second result concerns testing if a given oracle function is a direct-product of some function, i.e., testing if the oracle is a DP codeword. Goldreich and Safra [GS00] pioneered local *testing* of the DP code and its PCP application. A recent result by Dinur and Goldenberg [DG08] enabled for the first time testing proximity to this important code in the "list-decoding" regime. In particular, they give a 2-query test which works for *polynomially small* success probability $1/k^{\alpha}$, and show that no such test works below success probability $1/k$.

Our main result is a 3-query test which works for *exponentially small* success probability $\exp(-k^{\alpha})$. Our techniques (based on recent simplified decoding algorithms for the same code [IJKW08], discussed above) also allow us to considerably simplify the analysis of the 2-query test of [DG08]. We then show how to *derandomize* their test, achieving a code of polynomial rate, independent of k, and success probability $1/k^{\alpha}$.

Finally we show the applicability of the new tests to PCPs. Starting with a 2-query PCP over an alphabet Σ and with soundness error $1-\delta$, Rao [Rao08] (building on Raz's (k-fold) parallel repetition theorem [Raz98] and Holenstein's proof [Hol07]) obtains a new 2-query PCP over the alphabet Σ^k with soundness error $\exp(-\delta^2 k)$. Our techniques yield a 2-query PCP with soundness error $\exp(-\delta\sqrt{k})$. Our PCP construction turns out to be essentially the same as the miss-match proof system defined and analyzed by Feige and Kilian [FK00], but with simpler analysis and exponentially better soundness error.

References

[DG08] I. Dinur and E. Goldenberg. Locally testing direct products in the low error range. In *Proceedings of the Forty-Ninth Annual IEEE Symposium on Foundations of Computer Science*, pages 613–622, 2008.

[FK00] U. Feige and J. Kilian. Two-prover protocols - low error at affordable rates. *SIAM Journal on Computing*, 30(1):324–346, 2000.

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR-Lemma. *Electronic Colloquium on Computational Complexity*, TR95-050, 1995.

[GS00] O. Goldreich and S. Safra. A combinatorial consistency lemma with application to proving the PCP theorem. *SIAM Journal on Computing*, 29(4):1132–1154, 2000.

[Hol07] T. Holenstein. Parallel repetition: Simplifications and the no-signaling case. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 411–419, 2007.

[IJK06] R. Impagliazzo, R. Jaiswal, and V. Kabanets. Approximately list-decoding direct product codes and uniform hardness amplification. In *Proceedings of the Forty-Seventh Annual IEEE Symposium on Foundations of Computer Science*, pages 187–196, 2006.

[IJKW08] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson. Uniform direct-product theorems: Simplified, optimized, and derandomized. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 579–588, 2008. (full version available as *ECCC TR08-079*).

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the Thirty-Sixth Annual IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[Imp02] R. Impagliazzo. Hardness as randomness: A survey of universal derandomization. *Proceedings of the ICM*, 3:659–672, 2002. (available online at arxiv.org/abs/cs.CC/0304040).

[IW97] R. Impagliazzo and A. Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[Lev87] L.A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[Rao08] A. Rao. Parallel repetition in projection games and a concentration bound. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pages 1–10, 2008.

[Raz98] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[STV01] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[Tre05] L. Trevisan. On uniform amplification of hardness in NP. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pages 31–38, 2005.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

Complexity of Constraint Satisfaction Problems: Exact and Approximate

Prasad Raghavendra

A Constraint Satisfaction Problem (CSP) Λ is specified by a family of predicates over a finite domain $[q] = \{1, 2, \ldots, q\}$. Every instance of the CSP Λ consists of a set of variables $\mathcal{V}$, along with a set of constraints $\mathcal{P}$ on them. Each constraint in $\mathcal{P}$ consists of a predicate from the family Λ applied to a subset of variables. Well known examples of CSPs include MaxCut, 2-SAT and 3-SAT.

1. Perfect Satisfiability

Problem 1 (Exact-Λ)**.** Given an instance $\Im$ of the Λ-CSP, determine whether there is an assignment satisfying all the constraints in $\mathcal{P}$.

As it turns out, for most CSPs Λ, the Exact-Λ is **NP**-hard. In fact, the following well known theorem lists all boolean CSPs that are tractable.

Theorem 1 (Schraefer's Theorem)**.** *A boolean CSP Λ is **NP**-hard unless it is one of the following:* XOR *(linear equations over $\mathbb{F}_2$),* 2-SAT*,* Horn SAT*,* Dual Horn SAT *or a trivial CSP for which $\vec{0}$ or $\vec{1}$ is always a satisfying assignment.*

A natural question to ask is what makes CSPs **NP**-hard or easy. Indeed, there is apparently an elegant characterization of easy CSPs.

Consider the XOR problem. Fix an instance $\Im$ of XOR over n variables. Given three solutions $X^{(1)}, X^{(2)}, X^{(3)} \in \{0,1\}^n$ to $\Im$, one can create a new solution $Y \in \{0,1\}^n$ as follows:

$$Y_i = XOR(X_i^{(1)}, X_i^{(2)}, X_i^{(3)}) \qquad \forall i \in [n]\,.$$

It is easy to check that Y is also a feasible solution to the instance $\Im$. Thus the $XOR : \{0,1\}^3 \rightarrow \{0,1\}$ yields a way to combine three solutions in to a new solution for the same instance. A function of this form is known as a *polymorphism.* Formally, a polymorphism of a CSP Λ is defined as follows:

Definition 1.1 (Polymorphisms)**.** A function $F : [q]^R \to [q]$ is said to be *a polymorphism* for the CSP Λ, if for every instance $\Im$ of Λ, and R solutions $X^{(1)}, X^{(2)}, \ldots, X^{(R)} \in [q]^n$ that satisfy all constraints in $\Im$, the vector $Y \in [q]^n$ defined below is also a feasible solution.

$$Y_i = F(X_i^{(1)}, X_i^{(2)}, X_i^{(3)}, \ldots, X_i^{(R)}) \qquad \forall i \in [n] .$$

Note that the dictator functions $F(x) = x^{(i)}$ are polymorphisms for every CSP Λ. These will be referred to as *projections* or trivial polymorphisms.

The boolean CSPs that are described in Schraefer's theorem all have non-trivial polymorphisms. Specifically, 2-SAT has the Majority functions, Horn-SAT has OR functions, and Dual Horn-SAT has the AND functions as polymorphisms.

More generally, it is conjectured by Bulatov et al.[1] that the existence of non-trivial polymorphisms characterizes CSPs that are tractable. A rough statement of their conjecture is as follows:

Conjecture 3. *Let Λ be a "core". Exact-Λ is polynomial time tractable if there exists $F \in \mathrm{Poly}(\Lambda)$ that are "not juntas". Otherwise Exact-Λ is **NP**-hard.*

The condition that Λ is a *core* enforces that there are no homomorphisms from Λ in to itself. The above conjecture is true for CSPs over domain sizes 2 and 3. Furthermore, this conjecture seems supported by a wealth of evidence. For instance, there are results that assume a CSP has polymorphisms of a specific kind, and then obtain polytime algorithms for them.

2. Approximating CSPs

We now turn to the problem of approximating CSPs, specifically the Max-Λ problem defined below.

Problem 2 (Max-Λ)**.** Given an instance $\Im$ of the Λ-CSP, find an assignment that satisfies the maximum number (equivalently fraction) of constraints.

Generalizing the notion of polymorphisms from Exact-Λ to Max-Λ we define a function F to be a α-approximate polymorphism if it is given R solutions of value at least c, then the output has value at least $\alpha \cdot c$. The formal definition of approximate polymorphisms is presented below.

Definition 2.1. A Distributional Function F outputs a probability distribution over $[q]$ on input from $[q]^R$. Formally, it is a map $F : [q]^R \to \blacktriangle_q$ where $\blacktriangle_q$ is the set of probability distributions over $[q]$.

Alternatively, $\blacktriangle_q$ is the q-dimensional simplex in $\mathbb{R}^q$. A distributional function is given by $F = (F_1, F_2, \ldots, F_q)$ where $F_i : [q]^R \to \mathbb{R}$, $F_i(x) \geq 0$ and $\sum_i F_i(x) = 1$.

Definition 2.2. A distribution over distributional functions (DDF) $\mathcal{F}$ is a probability distribution over distributional functions F, $F : [q]^R \to \blacktriangle_q$

Let $\mathrm{val}_{\Im}(X)$ denote the objective value of an assignment X on instance $\Im$.

Definition 2.3 (α-approximate polymorphism)**.** A DDF $\mathcal{F}$ is an α-approximate polymorphism if the following holds:

For every instance $\Im$, and R assignments $X^{(1)}, X^{(2)}, \ldots, X^{(R)} \in [q]^n$, if $\mathrm{val}_{\Im}(X^{(i)}) \geq c$ for all i then,

$$\mathop{\mathbb{E}}_{F \in \mathcal{F}}[\mathrm{val}_{\Im}(F(X^{(1)}, \ldots, X^{(R)}))] \geq \alpha \cdot c$$

Here, by $F(X^{(1)}, \ldots, X^{(R)})$ we mean a random assignment $Y \in [q]^n$ generated as follows:

- For each $i \in [n]$, apply F on the i^{th} bits of $X^{(1)}, X^{(2)}, \ldots, X^{(R)}$ to get a distribution D_i on $\blacktriangle_q$.
- Set Y_i by sampling from D_i independent of everything else.

$\mathrm{val}_{\Im}(F(X^{(1)}, \ldots, X^{(R)}))$ is the expected value of the assignment Y obtained above.

Similarly define (c, α)-approximate polymorphisms by fixing the value of c in the above definition.

For every CSP, it is easy to see that the dictator functions are 1-approximate polymorphisms. As in the case of Exact-Λ, we will be interested in non-trivial polymorphisms. Specifically, we make the following definition:

Definition 2.4. A DDF $\mathcal{F}$ is τ-pseudorandom if for every $F \in \mathcal{F}$, and every distribution μ on $[q]$, all the influences of F under distribution μ^R are less than τ.

In analogy to Exact-Λ, it is natural to conjecture that the existence of α-approximate and non-dictator polymorphisms capture the approximability of a CSP. Indeed, the well-known Unique Games Conjecture of Khot [3] is equivalent to this natural conjecture. Formally define,

- $\alpha_\Lambda \overset{\text{def}}{=}$ largest α, such that there exists a τ-pseudorandom α-approximate DDF for every $\tau > 0$.
- $\alpha_{\Lambda,c} \overset{\text{def}}{=}$ largest α, such that there exists a τ-pseudorandom (c, α)-approximate DDF for every $\tau > 0$.

We show the following theorem that settles the approximability of every CSP under **UGC**:

Theorem 2. [4] *Unique Games Conjecture $\Longrightarrow$ For every Λ, α_Λ is the approximation threshold. Furthermore, Unique Games Conjecture $\Longleftrightarrow$ that for every Λ and $c > 0$, $\alpha_{\Lambda,c}$ is the correct approximation threshold for instances of value c on Λ.*

There are two aspects to the above theorem. First, the following lemma follows directly from the techniques of Khot et al.[2] for producing UG hardness reductions.

Lemma 2.1 (Hardness Part)**.** Unique Games Conjecture $\Longrightarrow$ For every Λ, α_Λ it is **NP**-hard to approximate better than α_Λ. Furthermore, Unique Games Conjecture $\Longleftrightarrow$ For every Λ and $c > 0$, on instances of value c, it is **NP**-hard to approximate better than $\alpha_{\Lambda,c}$.

Lemma 2.2 (Algorithmic Part)**.** If a CSP Λ has α-approximate polymorphisms then there is an $\alpha - \varepsilon$-approximation algorithm for Λ. (same extends to $\alpha_{\Lambda,c}$)

The above lemma is the core of the soundness analysis in [4], stated in a purely algorithmic way. The algorithm is based on a simple semidefinite programming relaxation (see [4]).

References

[1] Andrei. Bulatov, Peter Jeavons, Andrei Krokhin *Classifying the complexity of constraints using finite algebras*, SIAM J. Comput. **34**(3) (2005), 720–742.

[2] Subhash Khot, Guy Kindler, Elchanan Mossel, Ryan O'Donnell *Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?*, SIAM J. Comput., **37**(1) (2007), 319–357.

[3] Subhash Khot *On the power of unique 2-prover 1-round games*, ACM STOC (2002) 767–775.

[4] Prasad Raghavendra *Optimal Algorithms and Inapproximability Results for Every CSP?*, ACM STOC (2008), 245–254.

Efficiency Improvements in Constructions of Pseudorandom Generators from Any One-way Function

Omer Reingold

(joint work with Iftach Haitner, and Salil Vadhan)

Abstract: We give a new construction of pseudorandom generators from any one-way function. The construction achieves better parameters and is simpler than that given in the seminal work of Håstad, Impagliazzo, Levin and Luby [SICOMP '99]. The key to our construction is a new notion of *next-block pseudoentropy*, which is inspired by the notion of "inaccessible entropy" recently introduced in [Haitner, Reingold, Vadhan and Wee, STOC '09]. An additional advantage over all previous constructions is that our pseudorandom generators are highly parallelizable and invoke the one-way function in a non-adaptive manner. Using [Applebaum, Ishai and Kushilevitz, SICOMP '06], this implies the existence of pseudorandom generators in NC^0 based on the existence of one-way functions in NC^1.

1. Introduction

The result of Håstad, Impagliazzo, Levin and Luby [6] that one-way functions imply pseudorandom generators is one of the centerpieces of the foundations of cryptography and the theory of pseudorandomness.

From the perspective of cryptography, it shows that a very powerful and useful cryptographic primitive (namely, pseudorandom generators) can be constructed from the minimal assumption for complexity-based cryptography (namely, one-way functions). With this starting point, numerous other cryptographic primitives can also be constructed from one-way functions, such as private-key cryptography [1, 8], bit-commitment schemes [9], zero-knowledge proofs for NP [2], and identification schemes [3].

From the perspective of pseudorandomness, it provides strong evidence that pseudorandom bits can be generated very efficiently, with smaller computational resources than the "distinguishers" to whom the bits should look random. Such kinds of pseudorandom generators are needed, for example, for hardness results in learning [11] and the natural proofs barrier for circuit lower bounds [10]. Moreover, the paper of Håstad, Impagliazzo, Levin and Luby introduced concepts and techniques that now permeate the theory of pseudorandomness, such as pseudoentropy and the Leftover Hash Lemma.

A drawback of the construction of Håstad, Impagliazzo, Levin and Luby, however, is that it is quite complicated. While it utilizes many elegant ideas and notions, the final construction combines these in a rather ad hoc and indirect fashion due to various technical issues. In addition to being less satisfactory from an aesthetic and pedagogical perspective, the complexity of the construction also has a significant impact on its efficiency. Indeed, it is too inefficient to be implemented even for very modest settings of parameters.

In the last few years, progress has been made on simplifying the construction of Håstad, Impagliazzo, Levin and Luby [7] and improving its efficiency [4]. These constructions, however, still retain the overall structure of the Håstad, Impagliazzo, Levin and Luby construction, and thus retain some of the complex and ad hoc elements.

In this paper, we present a significantly more direct and efficient construction of pseudorandom generators from one-way functions. The key to our construction is a new notion of *next-block pseudoentropy*, which is inspired by the recently introduced notion of "inaccessible entropy" [5].

References

[1] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[2] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991. Preliminary version in *FOCS'86*.

[3] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[4] I. Haitner, D. Harnik, and O. Reingold. On the power of the randomized iterate. In *Advances in Cryptology – CRYPTO 2006*, 2006.

[5] I. Haitner, O. Reingold, S. Vadhan, and H. Wee. Inaccessible entropy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pages 611–620, 31 May–2 June 2009.

[6] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Preliminary versions in *STOC'89* and *STOC'90*.

[7] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, 2006.

[8] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[9] M. Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. Preliminary version in *CRYPTO'89*.

[10] A. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, Aug. 1997.

[11] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

The k-Clique Problem on Random Graphs & New Conjectures on $\mathbf{AC^0}$

Benjamin Rossman

We discuss recent work on the average-case complexity of the k-Clique problem. One result is a lower bound, for every constant k, of $\omega(n^{k/4})$ on the size of constant-depth (AC^0) circuits which approximate k-Clique on Erdős-Rényi random graph $G(n,p)$ where $p(n) \in \Theta(n^{-2/(k-1)})$ is any threshold function for the existence of k-cliques [2]. We also obtain a lower bound of $\omega(n^{k/4})$ on the size of *monotone* circuits which approximate k-Clique on both $G(n,p)$ and $G(n,p+p^{1+\epsilon})$ (for some small constant $\epsilon > 0$) [3].

These lower bounds are essentially tight. By a result of Amano [1], there exist constant-depth circuits of size $n^{k/4+O(1)}$ which approximate k-Clique on $G(n,p)$ for all functions $p(n)$. We give a monotone version of these circuits in [3].

In this talk we also present two new conjectures concerning AC^0. The first concerns a notion which we call *average maximal sensitivity.* For boolean function $f_1,\dots,f_m : \{0,1\}^n \longrightarrow \{0,1\}$, let

$$\mathbf{ams}(f_1,\dots,f_m) \triangleq \mathrm{E}_{x\in\{0,1\}^n}\left[\max_{j\in\{1,\dots,m\}} \mathbf{sens}(f_j,x)\right]$$

where $\mathbf{sens}(f,x) \triangleq \#\{i \in \{1,\dots,n\} \mid f(x) \neq f(x \text{ with the } i^{\text{th}} \text{ bit flipped})\}$. Using Håstad's Switching Lemma, it can be shown that if functions $f_1,\dots,f_m$ are computed by an AC^0 circuit of depth d with $m = \mathrm{poly}(n)$ output nodes, then $\mathbf{ams}(f_1,\dots,f_m) = O((\log n)^d)$. We conjecture that there is a *direct inductive proof* of this fact. Precisely, suppose $f_1,\dots,f_n$ satisfy $\mathbf{ams}(f_1,\dots,f_n) = O((\log n)^d)$ and let $g_1,\dots,g_n$ be boolean functions where each g_i is the product (i.e. AND) of a subset of f_j's. Must it hold that $\mathbf{ams}(g_1,\dots,g_n) = O((\log n)^{d+1})$? (We can ask the same question replacing both $O((\log n)^d)$ and $O((\log n)^{d+1})$ with $n^{o(1)}$. Even this modified conjecture would give a brand-new proof that Parity $\notin \mathrm{AC}^0$.)

Our second conjecture is that there is no *balanced* AC^0 *graph property*, i.e., no AC^0 property of graphs that is invariant under permutations of vertices and holds for between an ϵ and $1-\epsilon$ fraction of n-vertex graphs for some $\epsilon > 0$. This conjecture implies a zero-one law for successor-invariant first-order logic. Resolving this question seems to call for new techniques on AC^0 (beyond the Switching Lemma and LMN Theorem). We observe:

- there is no balanced AC^0 symmetric boolean function,
- there is a balanced AC^0 boolean function which is invariant under a transitive group action (the Tribes function of Ben-Or and Linial),
- there is a balanced graph property whose Fourier coefficients satisfy the conclusion of the LMN Theorem ("there exists a k-clique of the expected maximum size ($= (2+o(1))\log n$)").

References

[1] K. Amano, *k-Subgraph isomorphism on* AC^0, CCC (2009), pages 9-18.

[2] B. Rossman, *On the constant-depth complexity of k-clique*, STOC (2008), pages 721-730.

[3] B. Rossman, *The monotone complexity of k-clique on random graphs*, manuscript.

Recent Results on Polynomial Identity Testing

Amir Shpilka

Polynomial Identity Testing (PIT) is a fundamental problem in algebraic complexity: We are given a circuit computing a multivariate polynomial, over some field $\mathbb{F}$, and we have to determine whether it is identically zero or not. Note that we want the polynomial to be identically zero and not just to be equal to the zero function so, for example, $x^2 - x$ is the zero function over $\mathbb{F}_2$ but not the zero polynomial. The importance of this problem follows from its many applications: Algorithms for primality testing [2, 3], for deciding if a graph contains a perfect matching [17, 18, 7] and more, are based on reductions to the PIT problem (see the introduction of [16] for more applications).

There are two well studied scenarios in which the PIT problem is considered. The first is the so called *black-box* model in which the circuit is given as a black-box and we can access it only by querying its value on inputs of our choice. It is clear that every such algorithm must produce a test set for the circuit. Namely, a set of points such that if the circuit vanishes on all the points in the set then the circuit computes the zero polynomial. Another well studied scenario is the non black-box case in which the circuit is given to us as input. In particular, we have access to the polynomials that are being computed at various gates of the circuit. Clearly this is an 'easier' version of the problem, yet PIT is extremely difficult in this model as well.

Determining the complexity of PIT is one of the greatest challenges of theoretical computer science. It is one of a few problems for which we have coRP algorithms but no sub-exponential time deterministic algorithms. Indeed, many clever randomized algorithms are known for the general PIT question [23, 27, 9, 8, 16, 2] whereas sub-exponential time deterministic algorithms are known only for very restricted models. One explanation for this state of affairs is the strong relation between PIT and lower bounds for arithmetic circuits. Although seemingly very different, the problem of derandomizing PIT (i.e., that of giving efficient deterministic algorithms for the problem) is closely related to the problem of proving super polynomial lower bounds for arithmetic circuits. In [12] Kabanets and Impagliazzo showed that efficient deterministic algorithms for PIT imply that NEXP does not have polynomial size arithmetic circuits. Specifically, if PIT can be solved deterministically in polynomial time, even in the non black-box model, then either the Permanent cannot be computed by polynomial size arithmetic circuits or NEXP $\not\subseteq$ P/poly. That is, we get a super polynomial lower bound either for NEXP or for the Permanent. In [12] it was also shown that from super-polynomial lower bounds for arithmetic circuits one can design a deterministic quasi-polynomial time algorithm for PIT. In [11] we obtained analogous results for bounded depth circuits. In [1], Agrawal observed that polynomial time derandomization of PIT, in the black-box model, implies exponential lower bounds for arithmetic circuits. These results show the strong connection between PIT and lower bounds and indicate how difficult and important this problem is.

Because of the strong connection to proving lower bounds it is not surprising that the PIT problem becomes very interesting already for bounded depth circuits. Specifically, [4] proved that polynomial time derandomization of PIT for depth 4 circuits already implies exponential lower bounds for *general* arithmetic circuits. In combination with the results of [12] this gives a quasi-polynomial time derandomization of PIT for general arithmetic circuits. Hence, the problem of derandomizing PIT for depth 4 circuits is as difficult (and as important) as the problem for general arithmetic circuits.

Currently, deterministic subexponential PIT algorithms are known for non-commutative arithmetic formulae [19], for depth 3 circuits with a small top fan-in [10, 15, 5, 13, 22, 25, 14], for sums of read-once formulas [24, 25] and for multilinear depth 4 circuits with bounded top fan-in (as well as several very restricted versions of depth 4 circuits [5, 21, 25]). It is not known whether derandomizing PIT for depth 4 multilinear circuit implies a derandomization of PIT for general multilinear circuits. However, such a derandomization does imply an exponential lower bound for general multilinear circuits, thus improving the slightly super linear bound of [20].

Another line of research concerning PIT is better understanding its relation to other computational problems. In [26] a relation between PIT and multivariate polynomial factorization was found. Specifically, [26] showed that one can derandomize PIT if and only if one can derandomize the problem of computing variable disjoint factors of a multilinear given polynomial (that relation holds both in the black-box and non black-box models). In [6] a relation between deterministic PIT as well as circuit lower bounds and the isolation lemma was found.

In this talk we shall survey most of the recent results on PIT that were mentioned above and will give a list of what we think are the most accessible and important open problems.

References

[1] M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th FSTTCS*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105, 2005.

[2] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. *JACM*, 50(4):429–443, 2003.

[3] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[4] M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual FOCS*, pages 67–75, 2008.

[5] V. Arvind and P. Mukhopadhyay. The monomial ideal membership problem and polynomial identity testing. In *Proceedings of the 18th ISAAC*, pages 800–811, 2007.

[6] V. Arvind and P. Mukhopadhyay. Derandomizing the isolation lemma and lower bounds for circuit size. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, RANDOM 2008*, pages 276–289, 2008.

[7] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM J. on Computing*, 24(5):1036–1050, 1995.

[8] Z. Chen and M. Kao. Reducing randomness via irrational numbers. *SIAM J. on Computing*, 29(4):1247–1256, 2000.

[9] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978.

[10] Z. Dvir and A. Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. *SIAM J. on Computing*, 36(5):1404–1434, 2006.

[11] Z. Dvir, A. Shpilka, and A. Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. on Computing*, 39(4):1279–1293, 2009.

[12] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[13] Z. S. Karnin and A. Shpilka. Deterministic black box polynomial identity testing of depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 23rd Annual CCC*, pages 280–291, 2008.

[14] N. Kayal and S. Saraf. Blackbox polynomial identity testing for depth 3 circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, (32), 2009.

[15] N. Kayal and N. Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.

[16] D. Lewin and S. Vadhan. Checking polynomial identities over any field: Towards a derandomization? In *Proceedings of the 30th Annual STOC*, pages 428–437, 1998.

[17] L. Lovasz. On determinants, matchings, and random algorithms. In L. Budach, editor, *Fundamentals of Computing Theory*. Akademia-Verlag, 1979.

[18] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[19] R. Raz and A. Shpilka. Deterministic polynomial identity testing in non commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[20] R. Raz, A. Shpilka, and A. Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. on Computing*, 38(4):1624–1647, 2008.

[21] N. Saxena. Diagonal circuit identity testing and lower bounds. In *ICALP (1)*, pages 60–71, 2008.

[22] N. Saxena and C. Seshadhri. An almost optimal rank bound for depth-3 identities. In *Proceedings of the 24th annual CCC*, 2009.

[23] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *JACM*, 27(4):701–717, 1980.

[24] A. Shpilka and I. Volkovich. Read-once polynomial identity testing. In *Proceedings of the 40th Annual STOC*, pages 507–516, 2008.

[25] A. Shpilka and I. Volkovich. Improved polynomial identity testing for read-once formulas. In *APPROX-RANDOM*, pages 700–713, 2009.

[26] A. Shpilka and I. Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. Submitted, 2009.

[27] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation*, pages 216–226. 1979.

Fast Polynomial Factorization and Modular Composition

Chris Umans
(joint work with Kiran Kedlaya)

One of the most prominent success stories of algorithmic algebra is the invention of polynomial time algorithms for factoring polynomials [9, 5, 8]. In this work we focus on the most basic version of this problem, factoring univariate polynomials over a finite field. Classical polynomial-time algorithms for this problem are due to Berlekamp [1], and Cantor and Zassenhaus [3]. With additional innovations [10], these algorithms can be made to run in (nearly) quadratic time in n, the degree of the polynomial to be factored. Subquadratic-time algorithms were devised by Kaltofen and Shoup [6].

The bottleneck in all of these "fast" algorithms turns out to be to computing the polynomial $X^{q^i} \bmod A(X)$, where $A \in \mathbb{F}_q[X]$ is the polynomial to be factored, and i ranges between 1 and n. It is not hard to see that this can be accomplished with $O(\log q^i) = O(i \log q)$ modular polynomial multiplications (each costing $\widetilde{O}(n)$ arithmetic operations), by repeated squaring. Since i may be as large as n, this strategy takes quadratic time in the worst case. A different algorithm employs *modular composition* of degree n polynomials, and can break the quadratic barrier. This algorithm works as follows: first, compute $X^q \bmod A(X)$ using repeated squaring, then *compose* X^q with itself (modulo $A(X)$) to obtain X^{q^2}, compose that polynomial with itself (modulo $A(X)$) to obtain X^{q^4}, and so on. The overall cost is $O(\log q)$ modular polynomial multiplications followed by $O(\log i)$ modular polynomial compositions. One can hope to compute each modular composition in $\widetilde{O}(n)$ arithmetic operations, which would lead to an overall $\widetilde{O}(n)$ algorithm. In fact, by slightly adapting the algorithm for polynomial factorization presented in Kaltofen-Shoup [6], one finds that improving the exponent α in a $\widetilde{O}(n^\alpha)$ algorithm for modular composition directly improves the exponent on the (randomized) polynomial factorization algorithm. When $\alpha = 1$, the exponent on the polynomial factorization algorithm becomes 1.5, and this is what we achieve in this work[1].

We now focus on the *modular composition* problem: given degree n univariate polynomials $f(X), g(X)$, and $A(X)$ with coefficients in $\mathbb{F}_q$, we wish to produce the polynomial $f(g(X)) \bmod A(X)$. The best previous algorithm (Brent and Kung [2], and Huang and Pan [4]) dates to 1978, and achieves exponent 1.667. This number arises because the algorithm reduces the problem to matrix multiplication (and so the best upper bound on the exponent of matrix multiplication enters the running time). Even if the exponent of matrix multiplication is 2, this algorithm takes $\Omega(n^{1.5})$ operations, so a different strategy is needed. The following sequence of steps effectively reduces this problem to another problem, *multivariate multipoint evaluation*:

[1] In this discussion (and in the rest of this abstract), we ignore the dependence on q, the field size, which is always $\widetilde{O}(\log q)$ or $\widetilde{O}(\log^2 q)$.

- Produce the "multivariate version" of f, which is the polynomial
$$\underline{f}(X_0, X_1, \ldots, X_{m-1})$$
with degree at most $d-1$ in each variable for which
$$f(X) = \underline{f}(X^{d^0}, X^{d^1}, \ldots, X^{d^{m-1}}).$$
For concreteness it helps to think of $m = \log n$ and $d = 2$ (so $\underline{f}$ is a multilinear polynomial), although the actual choice of m, d will be different.
- Compute $g(X)^{d^i} \bmod A(X)$ for $i = 1, \ldots m-1$. Call these polynomials $g_i(X)$. This requires $\widetilde{O}(n)$ operations using repeated squaring and fact algorithms for polynomial multiplication and division with remainder.
- Observe that $\underline{f}(g_0(X), \ldots, g_{m-1}(X)) \equiv f(g(X)) \pmod{A(X)}$. Since the former polynomial has degree only $ndm = \widetilde{O}(n)$, we can afford to write it down and then reduce modulo $A(X)$ one final time. To do this we use an evaluation/interpolation approach.
- Select ndm distinct points in $\mathbb{F}_q$ and evaluate each $g_i(X)$ at all of them. This requires $\widetilde{O}(ndm)$ operations using fast univariate multipoint evaluation algorithms. Each evaluation point α yields the vector
$$(g_0(\alpha), \ldots, g_{m-1}(\alpha)) \in \mathbb{F}_q^m.$$
We evaluate $\underline{f}$ at each of these points, and finally perform fast univariate interpolation to recover the polynomial $\underline{f}(g_0(X), \ldots, g_{m-1}(X)) \equiv f(g(X))$.

Note that the only step that is not already within $\widetilde{O}(n)$ operations, is the final multipoint evaluation of the multivariate polynomial $\underline{f}$. Hence we have reduced the problem of modular composition to that of multivariate multipoint evaluation.

In contrast to the univariate case (where nearly-linear algorithms are well-known), very little was known about *multivariate* multipoint evaluation. Nüsken & Ziegler [7] have a non-trivial algorithm that again relies on fast matrix multiplication, but it is not enough to yield an improvement to modular composition algorithms via the above reduction. We devise a completely different algorithm for multivariate multipoint evaluation that runs in "nearly-linear" time in its input. It is interesting to note that our algorithm is not algebraic. We describe the main ideas next.

We are given a polynomial $h(X_0, X_1, \ldots, X_{m-1})$ with coefficients in $\mathbb{F}_q$, and $N = d^m$ evaluation points in $\mathbb{F}_q^m$. For simplicity, let us assume q is a prime. We first note that *if* the N points happened to be all of $\mathbb{F}_q^m$, then we could solve the problem in $\widetilde{O}(N)$ times by computing the evaluations of h over the entire domain via the multidimensional, finite-field FFT. We will essentially reach this case via the following transformation: first, lift the coefficients of h and the coordinates of the evaluation points to the *integers* $\{0, 1, 2, \ldots, q-1\}$. Note that in the integers, an evaluation has magnitude at most $d^m q^{dm} = M$. Select small primes $p_1, \ldots, p_k$ whose product exceeds M (so $p_i = O(\log M)$) and solve the multivariate multipoint evaluation problem modulo each p_i. Each evaluation can be reconstructed via the Chinese Remainder Theorem from these reduced instances. After several rounds of this multimodular reduction (it turns out 3 is sufficient), we end up with a small number of instances of the original problem relative to primes p of magnitude $\approx dm$. At this point we can use the aforementioned multidimensional FFT to evaluate over the entire domain. To see why this is near-optimal, observe that the *optimal* algorithm takes at least d^m time, while each of the small number of instance we end up with takes $\widetilde{O}(p^m) = \widetilde{O}((dm)^m)$ time, and we can make m small (say, a large constant). With slightly more effort, the same idea (lifting to the integers, followed by multimodular reduction) can be made to work when q is any prime power, and indeed for a class of extension rings that properly contains the finite fields.

Retracing the sequence of implications, we obtain $\widetilde{O}(n)$ time algorithms for modular composition, and in turn $\widetilde{O}(n^{1.5})$ time algorithms for polynomial factorization. It turns out that fast computation of $X^{q^i} \bmod A(X)$ lies at the core of other algebraic algorithms. Among other applications, we also obtain "exponent 1" algorithms for irreducibility testing and finding minimal polynomials. For all of these problems, our algorithms are (currently) the asymptotically fastest known. It is also possible to interpret our algorithm as giving a data structure supporting polynomial evaluation: given a degree n univariate polynomial $f(X)$ with coefficients in $\mathbb{F}_q$, we can produce in nearly-linear time, a nearly-linear sized data structure (namely, the tables of evaluations modulo the small primes) that answers evaluation queries (given $\alpha \in \mathbb{F}_q$, return $f(\alpha)$) in polylogarithmic time.

Open problems include improving the exponent of polynomial factorization, ideally to 1, and giving a nearly-linear *algebraic* algorithm for multivariate multipoint evalation and/or modular composition.

References

[1] E. R. Berlekamp, *Factoring polynomials over large finite fields*, Mathematics of Computation, 24 (1970), pp. 713–735.

[2] R. P. Brent and H. T. Kung, *Fast algorithms for manipulating formal power series*, J. ACM, 25 (1978), pp. 581–595.

[3] D.G. Cantor and H. Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Mathematics of Computation, 36 (1981), pp. 587–592.

[4] X. Huang and V. Y. Pan, *Fast rectangular matrix multiplication and applications.*, J. Complexity, 14 (1998), pp. 257–299.

[5] E. Kaltofen, *Polynomial factorization: a success story.*, in ISSAC, J. Rafael Sendra, ed., ACM, 2003, pp. 3–4.

[6] ———, *Subquadratic-time factoring of polynomials over finite fields.*, Mathematics of Computation, 67 (1998), pp. 1179–1197.

[7] M. Nüsken and M. Ziegler, *Fast multipoint evaluation of bivariate polynomials.*, in ESA, Susanne Albers and Tomasz Radzik, eds., vol. 3221 of Lecture Notes in Computer Science, Springer, 2004, pp. 544–555.

[8] J. von zur Gathen, *Who was who in polynomial factorization*, in ISSAC, Barry M. Trager, ed., ACM, 2006, p. 2.

[9] J. von zur Gathen and D. Panario, *Factoring polynomials over finite fields: A survey.*, J. Symb. Comput., 31 (2001), pp. 3–17.

[10] J. von zur Gathen and V. Shoup, *Computing Frobenius maps and factoring polynomials.*, Computational Complexity, 2 (1992), pp. 187–224.

Inaccessible Entropy

Salil Vadhan
(joint work with Iftach Haitner, Omer Reingold, and Hoeteck Wee)

Computational analogues of information-theoretic notions have given rise to some of the most interesting phenomena in the theory of computation. For example, a computational analogue of entropy, known as *pseudoentropy*, introduced by Håstad, Impagliazzo, Levin, and Luby [HILL], was the key to their fundamental result establishing the equivalence of pseudorandom generators and one-way functions, and has also now become a basic concept in complexity theory and cryptography.

In this work, we introduce another computational analogue of entropy, which we call *accessible entropy*, and present several applications of it to the foundations of cryptography. Before describing accessible entropy (and a complementary notion of *inaccessible entropy*), we recall the standard information-theoretic notion of entropy and the computational notion of pseudoentropy of Håstad et al.

Entropy and Pseudoentropy. Recall that the *entropy* of a random variable X is defined to be $\mathrm{H}(X) := \mathrm{E}_{x \overset{\mathrm{R}}{\leftarrow} X}[\log(1/\Pr[X = x])$, which measures the number of "bits of randomness" in X (on average). We will refer to $\mathrm{H}(X)$ as the *real entropy* of X to contrast with the computational analogues that we study. Håstad et al. [HILL] say that a random variable X has *pseudoentropy* (at least) k if there exists a random variable Y of entropy (at least) k such that X and Y are computationally indistinguishable.

The reason that pseudoentropy is interesting and useful is that there exist random variables X whose pseudoentropy is larger than their real entropy. For example, the output of a pseudorandom generator $G : \{0,1\}^\ell \to \{0,1\}^n$ on a uniformly random seed has entropy at most ℓ, but has pseudoentropy n (by definition). Håstad et al. proved that in fact, from *any* efficiently samplable distribution X whose pseudoentropy is noticeably larger than its real entropy, it is possible to construct a pseudorandom generator. By showing, in addition, how to construct such a distribution X from any one-way function, Håstad et al. prove their theorem that the existence of one-way functions implies the existence of pseudorandom generators.

The notion of pseudoentropy is only useful, however, as a lower bound on the "computational entropy" in a distribution. Indeed, it can be shown that every distribution on $\{0,1\}^n$ is computationally indistinguishable from a distribution of entropy at most $\mathrm{poly}(\log n)$. While several other computational analogues of entropy have been studied in the literature (cf., [BSW]), all of these are also meant to serve as ways of capturing the idea that a distribution "behaves like" one of higher entropy. In this paper, we explore a way in which a distribution can "behave like" one of much *lower* entropy.

Accessible Entropy. We motivate the idea of accessible entropy with an example. Consider an algorithm G that gets as input a random function h :

$\{0,1\}^n \to \{0,1\}^m$ from a family of collision-resistant hash functions (where $m \ll n$), chooses a random $x \overset{\mathrm{R}}{\leftarrow} \{0,1\}^n$, sets $y = h(x)$, and outputs the pair (y, x).

Now, information-theoretically, the second block of G's output (namely x) has entropy at least $n-m$ conditioned on the input h and the first block y, because $y = h(x)$ reveals only m bits of information about x. However, the collision-resistance property says that given the *state* of G after the first block, there is at most one consistent value of x that G can reveal with nonnegligible probability. (Otherwise, G would be able find two distinct messages $x \neq x'$ such that $h(x) = h(x')$.) This holds even if G is replaced by any polynomial-time adversary G^*. Thus, there is "real entropy" in x (conditioned on the history) but it is "computationally inaccessible" to G^*, to whom x effectively has entropy 0.

We generalize this basic idea to allow the upper bound on the "accessible entropy" to be a parameter k, and to consider both the real and accessible entropy accumulated over several blocks. In more detail, consider an m-block generator G that on input z, outputs a sequence $(y_1, \ldots, y_m)$ of blocks, and let $(Z, Y_1, \ldots, Y_m)$ be random variables denoting a random input Z to G and the output blocks of $\mathsf{G}(Z)$ (when G's coin tosses are chosen uniformly at random). We define the *real entropy* of G to be

$$\sum_i \mathrm{H}(Y_i | Z, Y_1, \ldots, Y_{i-1}),$$

where $\mathrm{H}(X|Y) = \mathrm{E}_{y \overset{\mathrm{R}}{\leftarrow} Y}[\mathrm{H}(X|_{Y=y})]$ is the standard notion of conditional entropy.

To define *accessible entropy*, consider a probabilistic polynomial-time adversary G^* that receives an input z, and then in sequence of m stages, tosses some fresh random coins s_i and computes and outputs a block y_i. At the end it should also justify that it has behaved consistently with the honest algorithm G by producing coin tosses r for G such that G would have output $(y_1, \ldots, y_m)$ on input z and coin tosses r. (For simplicity we restrict attention to G^* that always produce correct justifications, though our definitions and results can be generalized also to handle G^* that sometimes fail to do so.) Now, let $(Z, S_1, Y_1, S_2, Y_2, \ldots, S_m, Y_m)$ be random variables corresponding to the sequence of coins S_i and outputs Y_i of G^* on a random input Z. Then we define the *accessible entropy* achieved by G^* to be

$$\sum_i \mathrm{H}(Y_i | Z, S_1, \ldots, S_{i-1}).$$

The key point is that now we compute the entropy conditioned not just on the previous blocks, but on the entire local state of G^* prior to generating the i'th block. (We don't need to include Y_j for $j < i$ since these are determined by Z and $S_1, \ldots, S_j$.)

The collision resistance example given earlier shows that there can be generators G whose computationally accessible entropy is much smaller than the real Shannon entropy. Indeed, in that protocol, the real entropy of G's blocks is n (namely, the total entropy in x), but the computationally accessible entropy is at most $m + \mathrm{neg}(n)$, where $m \ll n$ is the output length of the collision-resistant hash function. (Here we are counting the conditional entropy in all of G's blocks for simplicity, but the definitions generalize naturally if we only want to sum the conditional entropies over some subset of blocks.) Thus, in contrast to pseudoentropy, accessible entropy is useful for expressing the idea that the "computational entropy" in a distribution is *smaller* than its real entropy. We refer to the difference (real entropy) − (accessible entropy) as the *inaccessible entropy* of G.

Applications. We have used the notion of inaccessible entropy and variants to:

- Give a much simpler and more efficient construction of statistically hiding commitment schemes from arbitrary one-way functions.
- Prove that constant-round statistically hiding commitments are necessary for constructing constant-round zero-knowledge proof systems for NP that remain secure under parallel composition (assuming the existence of one-way functions).
- Give a simpler construction of universal one-way hash functions and hence digital signature schemes from one-way functions. This appears in a follow-up subsequent paper [HRVW2]
- Inspire a simpler and more efficient construction of pseudorandom generators from one-way functions [HRV].

Bibliographic Note. Our paper [HRVW1] utilizes a more general (and more involved) notion of inaccessible entropy for *protocols*. The simpler notion of inaccessible entropy generators described above and the simple construction of such generators from one-way functions described in the talk will eventually be incorporated into the paper.

References

[BSW] B. Barak, R. Shaltiel, and A. Wigderson. Computational Analogues of Entropy. In *RANDOM-APPROX*, 2003.

[HRVW1] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Inaccessible entropy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pages 611–620, 31 May–2 June 2009. Full version available as *Electronic Colloquium on Computational Complexity* TR09-045, May 2009.

[HRVW2] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. Unpublished manuscript, October 2009.

[HRV] Iftach Haitner, Omer Reingold, Salil Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. Unpublished manuscript, November 2009.

[HILL] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. Preliminary versions in *STOC'89* and *STOC'90*.

Some Observations on Holographic Algorithms

Leslie G. Valiant

The theory of holographic algorithms is based on a notion of reduction that enables computational problems to be interrelated with unusual fluidity. The theory offers three basic reduction techniques:

(a) *Holographic transformations* that relate pairs of problems by simply taking a different view or *basis*,

(b) *Holographic gadgets* that use internal cancelations custom designed for the problems at hand, and

(c) *Interpolation techniques* for recovering information from the outputs of computations on a set of specially prepared variants of the problem instance at hand.

The overarching open question in the theory is whether this combination of techniques can bridge the gap between classical polynomial algorithms on the one hand, and the class of #P-complete (or NP- or $\oplus$P-complete) problems as defined by classical reductions, on the other.

In this talk we first review the basic notions of holographic algorithms as defined in [5, 7]. We then give an overview of our current understanding of this area, including some results from [1-7, 9].

We go on to introduce the notion of diversity for finite functions [8], in terms of which some limitations of the simplest kinds of holographic algorithms that we discussed in an earlier paper [6] can be explored more explicitly. These simplest holographic algorithms are those obtained from what we define as *elementary* reductions. We show that such algorithms do impose a limitation on the diversity of the functions that can be realized. It remains unresolved, however, whether holographic algorithms that are not bound by the constraints of elementarity, such as those given below, can evade this diversity limitation.

We then proceed to give some polynomial time holographic algorithms that evade this elementarity constraint, for three natural problems for undirected graphs of degree three [8]. These compute the parity of the number of solutions of each of the following three problems: feedback vertex sets (or, equivalently, induced forests), connected vertex covers, and vertex 3-colorings up to permutations of colors.

Besides evading the elementarity constraint our algorithms have other features that put them outside the currently better understood regions of holographic theory. For one thing the use of the three element basis **b3** from [7] puts them outside the collapse theorem of Cai and Lu [3], and hence outside any known classification such as that of symmetric signatures realized by two element bases [2]. Second, the results hold for parity rather than counting. For parity problems, or fixed finite fields in general, holographic transformations and interpolation both appear to offer less flexibility than they do for general counting problems. In particular, understanding the complexity of the counting problems modulo three for the structures we analyze here modulo two, appears to remain a challenge.

References

[1] J-Y. Cai, V. Choudhary and P. Lu, *On the Theory of Matchgate Computations.* Theory of Computing Systems, **45:1** (2009) 108-132.

[2] J-Y. Cai, P. Lu, *Holographic algorithms: from art to science.* Proc. 39th ACM Symp. on Theory of Computing, (2007) 401-410.

[3] J-Y. Cai, and P. Lu, *Holographic algorithms: The power of dimensionality resolved.* Theor. Comput. Sci. **410:18** (2009) 1618-1628.

[4] J-Y. Cai, P. Lu, and M. Xia. *Holographic Algorithms by Fibonacci Gates and Holographic Reductions for Hardness.* Proc. 49th Annual IEEE Symposium on Foundations of Computer Science, (2008) IEEE Press, 644-653.

[5] L. G. Valiant. *Holographic algorithms (extended abstract)*, Proc. 45th Annual IEEE Symposium on Foundations of Computer Science, Oct 17-19, Rome, Italy, (2004) IEEE Press, 306-315.

[6] L. G. Valiant. *Accidental algorithms*, Proc. 47th Annual IEEE Symposium on Foundations of Computer Science, Oct 22-24, Berkeley, CA, IEEE Press, (2006), 509-517.

[7] L. G. Valiant. *Holographic algorithms*, SIAM J. on Computing, **37:5**, (2008) 1565-1594. (Earlier version: Electronic Colloquium on Computational Complexity, Report TR05-099, (2005).)

[8] L. G. Valiant. *Some Observations on Holographic Algorithms*, Proc. 9th Latin American Theoretical Informatics Symposium, April 19-23, 2010, Oaxaca, Mexico, Springer-Verlag LNCS, To appear.

[9] M. Xia, P. Zhang, W. Zhao. *Computational complexity of counting problems on 3-regular planar graphs.* Theor. Comput. Sci. **384:1** (2007) 111-125.

Short Communications

The complexity of graph polynomials

Markus Bläser

(joint work with Christian Hoffmann, Johann A. Makowsky)

A graph polynomial P maps graphs to polynomials P_G over some ring R such that isomorphic graphs are mapped to the same polynomial. If we now fix some point ξ and map G to $P_G(\xi)$, we get a new graph invariant that maps graphs to elements of R, i.e, we evaluate the graph polynomial at ξ. For many graph polynomials that appear in the literature, results of the following types are known:

- The polynomial is $\#P$-hard to evaluate almost everywhere (in the Zariski sense).
- The polynomial can be evaluated in vertex-exponential time.
  (This is nontrivial in most cases, since the "obvious" ways of evaluating give edge-exponential running times.)
- The evaluation of the polynomial is fixed parameter tractable on graphs of bounded tree width.

The "order of quantifiers" is always: for all polynomials there is a proof that shows the three items. We are currently working on reversing the order of quantifiers, that is, we are looking for proofs that work for large classes of graph polynomials uniformly. This is work in progress. Possible classes are polynomials that are definable in some logic or polynomials that are p-definable (in Valiant's algebraic classes). For the third item, such a result was shown by Courcelle, Makowsky, and others for the class of monadic second order logic definable polynomials. For the first item, this is Makowsky's difficult point conjecture.

Counting decomposable univariate polynomials

Joachim von zur Gathen

A univariate polynomial f over a field is *decomposable* if it is the composition $f = g \circ h$ of two polynomials g and h whose degree is at least 2. We determine an approximation to the number of decomposables over a finite field. The tame case, where the field characteristic p does not divide the degree n of f, is reasonably well understood, and we obtain exponentially decreasing relative error bounds. The wild case, where p divides n, is more challenging and our error bounds are weaker.

The two central technical tools are a decomposition algorithm that works for most, but not all, inputs, and a normal form for the polynomials in Ritt's Second Theorem, where two essentially different pairs of polynomials yield the same composition.

The paper is available at http://arxiv.org/abs/0901.0054.

Complexity Theoretic Aspects of Property Testing

Oded Goldreich

Some complexity theorists may view property testers as PCPs of Proximity without the proof part. In general, property testing is concerned with approximate decisions, where the task is distinguishing between objects having a predetermined property and objects that are "far" from having this property. A potential tester is a randomized algorithm that queries the (representation of the) tested object at locations of its choice.

On the relation between adaptive and non-adaptive query complexity of graph properties in the adjacency matrix model. For any fixed property Π, let q denote the query complexity of (general, i.e., adaptive) testing of Π, and Q denote the corresponding non-adaptive query complexity (i.e., which refers to non-adaptive testers of Π). Following is a list of known and conjectured results, where $\widetilde{\Omega}$ and $\widetilde{\Theta}$ denote bounds with a slackness of a polylogarithmic factor.

- Theorem (see [3]): For any graph property in the adjacency matrix model, it holds that $Q = O(q^2)$.
- Theorem in [2]: There exist graph properties in the adjacency matrix model such that $Q = \widetilde{\Theta}(q)$. Actually, $Q = O(q)$ and even $Q = q$ are known too.
- Theorem in [2]: There exists a graph property in the adjacency matrix model such that $Q = \widetilde{\Theta}(q^{4/3})$.
- Theorem in [2]: There exists a graph property in the adjacency matrix model such that $Q = \widetilde{\Omega}(q^{3/2})$.

- Conjecture in [2]: For every integer $t > 2$, there exists a graph property in the adjacency matrix model such that $Q = \widetilde{\Theta}(q^{2-(2/t)})$. This conjecture is supported by a theorem that establish the same relation relation for a promise problem.

All existential results are proved using natural graph properties.

Hierarchy Theorems for Property Testing. Such results are proved for three central models of property testing: the general model of generic function, the model of bounded-degree graph properties, and the model of dense graph properties (in the adjacency matrix model). From a technical perspective, the treatment of the latter is most interesting, since it raises and resolves various natural questions regarding graph blow-up. For details, see [1].

References

[1] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg, *Hierarchy Theorems for Property Testing* in the proceedings of 13th RANDOM, Springer LNCS, Vol. 5687, pages 504-519, 2009.

[2] O. Goldreich and D. Ron, *Algorithmic Aspects of Property Testing in the Dense Graphs Model* in the proceedings of 13th RANDOM, Springer LNCS, Vol. 5687, pages 520–533, 2009.

[3] O. Goldreich and L. Trevisan, *Three Theorems regarding Testing Graph Properties* Random Structures and Algorithms, Vol. 23 (1), pages 23–57, August 2003.

Achieving capacity against additive errors and approximating almost-satisfiable Horn-SAT

Venkatesan Guruswami

Abstract

I gave a short report announcing two recent results, one from coding theory and another from hardness of approximation.

Explicit capacity-achieving binary codes for worst-case additive errors. We prove that explicit codes of optimal rate approaching capacity can be constructed against worst-case errors which are oblivious to the codeword (but not necessarily the message). Formally, we prove the following result [1]: there is an explicit, efficient stochastic encoding $E(\cdot, \cdot)$ of messages combined with a small number of auxiliary random bits, such that for *every* message m and *every* error vector e that contains at most a fraction p of ones, with high probability over the random bits r chosen by the encoder, m can be efficiently recovered from the corrupted codeword $E(m, r) + e$ by a decoder *without knowledge of the encoder's randomness r*. (Indeed such a result is rather easy to obtain if the encoder and decoder share random bits that are hidden from the channel.)

Our construction for additive errors also yields explicit *deterministic* codes of rate approaching $1 - H(p)$ for the "average error" criterion: for *every* error vector e of at most p fraction 1's, *most* messages m can be efficiently (uniquely) decoded from the corrupted codeword $C(m) + e$. Note that such codes cannot be linear, as the bad error patterns for all messages are the same in a linear code. We also give a new proof of the *existence* of such codes based on list decoding and certain algebraic manipulation detection codes. Our proof is simpler than the previous proofs from the literature on arbitrarily varying channels.

Tight inapproximability bound for almost-satisfiable Horn-SAT. By Schaefer's theorem, we know that linear equations mod 2, 2SAT, and Horn-SAT are essentially the only three distinct non-trivial constraint satisfaction problems over the Boolean domain for which satisfiability can de decided in polynomial time.

What if the instance is not perfectly satisfiable, but only "almost-satisfiable," i.e., it admits an assignment satisfying $(1-\epsilon)$ of the constraints for some small $\epsilon > 0$? Do there exist "robust" satisfiability algorithms that can find an assignment satisfying $1 - g(\epsilon)$ fraction of constraints for some $g(\epsilon) \to 0$ as $\epsilon \to 0$, given a $(1-\epsilon)$-satisfiable instance? Håstad's celebrated hardness result for Linear equations rules out such an algorithm for linear equations mod 2: it is NP-hard to satisfy even $(1/2+\epsilon)$ of the equations given a $(1-\epsilon)$-satisfiable instance. In sharp contrast, Zwick showed that robust satisfiability algorithms exist for 2SAT and Horn-SAT. For almost-satisfiable instances of 2SAT, he showed that using semidefinite programming one can efficiently satisfy $1 - O(\epsilon^{1/3})$ of the constraints. This bound was later improved by Charikar, Makarychev, and Makarychev to $1 - O(\sqrt{\epsilon})$ which is known to be best possible under the Unique Games conjecture of Khot. For Horn-SAT, Zwick gave a linear programming based algorithm that could satisfy a fraction $1 - O(\frac{\log\log(1/\epsilon)}{\log(1/\epsilon)})$ of the Horn clauses.

We prove that this exponentially worse bound on the fraction of unsatisfied clauses for Horn-SAT (as a function of ϵ, compared to the 2SAT case) is inherent. Specifically, in [2] we prove that it is Unique-Games hard to find an assignment satisfying $(1 - \frac{1}{O(\log(1/\epsilon))})$ of the constraints of a $(1-\epsilon)$-satisfiable instance of Horn-3SAT.

References

[1] V. Guruswami and A. Smith. *Explicit capacity-achieving codes against worst-case additive errors.* arXiv:0912.0965, 2009.

[2] V. Guruswami and Y. Zhou. *Tight inapproximability bounds for almost-satisfiable Horn-Sat and exact hitting set.* Manuscript, November 2009.

Multiplicities arising in Geometric Complexity Theory

Christian Ikenmeyer

(joint work with P. Bürgisser, M. Christandl)

In 1979 Valiant [Val79] conjectured the separation of complexity classes **VP** $\neq$ **VNP**. The Geometric Complexity Theory approach by Mulmuley and Sohoni (see for example [MS01, MS08]) tries to prove a conjecture, which implies a variant of **VP** $\neq$ **VNP**. To state the conjecture let $\det_n := \sum_{\pi\in S_n} \mathrm{sgn}(\pi) \prod_{i=1}^{n} X_{i\pi(i)}$,

$\mathrm{per}_m := \sum_{\pi\in S_m}\prod_{i=1}^m X_{i\pi(i)}$ and $z := X_{m+1,m+1}$ a variable that is not used by the permanent polynomial. Note that both $z^{n-m}\mathrm{per}_m$ and $\det_n$ are homogeneous polynomials of degree n. The group $\mathrm{GL}_{n^2} := \mathrm{GL}_{n^2}(\mathbb{C})$ acts on $\det_n$ and $z^{n-m}\mathrm{per}_m$ by replacing variables with linear combinations.

Conjecture ([MS01]). For all positive polynomials p and all $m_0 \in \mathbb{N}$ we find $m \geq m_0$ such that there does not exist a $\mathrm{GL}_{p(m)^2}$-equivariant surjection of the coordinate rings of orbit closures

$$\mathbb{C}[\overline{\mathrm{GL}_{p(m)^2}\cdot \det_{p(m)}}] \to \mathbb{C}[\overline{\mathrm{GL}_{p(m)^2}\cdot z^{p(m)-m}\mathrm{per}_m}].$$

Note that both coordinate rings are $\mathrm{GL}_{p(m)}$-representations. A proof certificate for the nonexistence of a surjection from left to right can theoretically be given by an irreducible representation that occurs on the right but not on the left. The occuring irreducible representations of the coordinate rings can be described in terms of so-called *Kronecker coefficients*. Thus an important step in this approach is the understanding of these coefficients. We gave asymptotic positivity properties for specific interesting families of Kronecker coefficients [BCI09], we showed that the computation of Kronecker coefficients is **#P**-hard [BI08] and we gave a combinatorial polynomial-time algorithm for computing "small" Kronecker coefficients that lie in an interesting subcase, namely the Littlewood-Richardson-coefficients [BI09].

References

[BCI09] Peter Bürgisser, Matthias Christandl, and Christian Ikenmeyer. Nonvanishing of Kronecker coefficients for rectangular shapes. arXiv:0910.4512, 2009.

[BI08] Peter Bürgisser and Christian Ikenmeyer. The complexity of computing Kronecker coefficients. In *FPSAC 2008, Valparaiso-Via del Mar, Chile, DMTCS proc. AJ*, pages 357–368, 2008.

[BI09] Peter Bürgisser and Christian Ikenmeyer. A max-flow algorithm for positivity of Littlewood-Richardson coefficients. In *FPSAC 2009, Hagenberg, Austria, DMTCS proc. AK*, pages 267–278, 2009.

[MS01] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. I. An approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526 (electronic), 2001.

[MS08] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

Combinatorial Constructions of Probabilistic Proof Systems

Or Meir

Probabilistic proof systems, and in particular interactive proofs and PCPs, are by now a flourishing research area that has lead to many interesting results and applications. Maybe the most famous results in this area are the PCP theorem [2, 1], stating that **NP** has a PCP that uses a constant number of queries to a proof of polynomial length, and the IP theorem [6, 9], stating that **PSPACE** has interactive proofs with polynomial number of rounds.

The original proofs of both those results, as well as many subsequent results in this area, were based on algebraic techniques: Given a claim to be verified, they construct a PCP for the claim by "arithmetizing" the claim, i.e., reducing the claim to a related "algebraic" claim about polynomials over finite fields, and then asking the prover to prove this algebraic claim. Proving the algebraic claim, in turn, requires an arsenal of tools that employ the algebraic structure of polynomials. While those algebraic techniques are very important and useful, it seems somewhat odd that one has to go through algebra in order to prove those theorems, since the theorems themselves say nothing about algebra. Furthermore, those techniques seem to give little intuition for why those theorems hold.

Given this state of affairs, it is an important goal to gain a better understanding of probabilistic proof systems and the fundamental reasons that make them possible. In her seminar paper, Dinur [4] has made a big step toward achieving this goal by giving an alternative proof for the PCP theorem using a combinatorial approach. Her proof is not only considerably simpler than the original proof, but also sheds more light on the intuitions that underlay the theorem.

In my research, I pursue this direction further, trying to prove all the main results of this area using a combinatorial approach. One published result of this flavor is a combinatorial PCPs that have verifiers that run in poly-logarithmic time [7], which also yields a combinatorial proof of **MIP** = **NEXP**. Two additional ongoing works concern with the combinatorial construction of PCPs that have short proof length (trying to match the result of [3]), and the combinatorial construction of PCPs that have sub-constant soundness (trying to match the result [8, 5]).

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and intractability of approximation problems. *Journal of ACM*, 45(3):501–555, 1998. Preliminary version in FOCS 1992.

[2] Sanjeev Arora and Shmuel Safra. Probabilistic checkable proofs: A new characterization of NP. *Journal of ACM volume*, 45(1):70–122, 1998. Preliminary version in FOCS 1992.

[3] Eli Ben-Sasson and Madhu Sudan. Simple PCPs with poly-log rate and query complexity. In *STOC*, pages 266–275, 2005. Full version can be obtained from Eli Ben-Sasson's homepage at `http://www.cs.technion.ac.il/~eli/`.

[4] Irit Dinur. The PCP Theorem by gap amplification. *Journal of ACM*, 54(3):241–250, 2007. Preliminary version in STOC 2006.

[5] Irit Dinur and Harsha Praladh. Composition of low-error 2-query PCPs using decodable PCPs. In *FOCS*, 2009.

[6] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[7] Or Meir. Combinatorial PCPs with efficient verifiers. In *FOCS*, 2009.

[8] Dana Moshkovitz and Ran Raz. Two query PCP with sub-constant error. In *FOCS*, 2008. Full version is available as ECCC TR08-071.

[9] Adi Shamir. IP=PSPACE. In *FOCS*, pages 11–15, 1990.

Average Time Fast SVP and CVP Algorithms for Low Density Lattices

Claus Peter Schnorr

Previous SVP and CVP algorithms of Kannan and Fincke, Pohst perform the stages of exhaustive enumeration of short/close lattice vectors in a straight forward order disregarding the success rate of stages. Our algorithm New Enum for SVP / CVP performs all stages in order of decreasing success rate, stages with high success rate are done first.

[S09] shows under **GSA** that New Enum runs in exponential time $n^{\frac{n}{32}+o(n)}$ and in polynomial time for moderately small $rd(\mathcal{L})$. We define the relative density $rd(\mathcal{L})$ of the lattice $\mathcal{L}$ by the equation $\lambda_1(\mathcal{L}) = rd(\mathcal{L})\,\gamma_n^{1/2}\det(\mathcal{L})^{1/n}$, where γ_n is the Hermite constant of dimension n and λ_1 is the length of a shortest nonzero lattice vector. Let the lattice basis $B = \in \mathbb{Z}^{m\times n}$ be given with the unique QR factorization, where $R = [r_{i,j}]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$ is upper triangular with positive diagonal entries.

GSA Let $B = QR = Q[r_{i,j}]$ satisfy $r_{i,i}^2/r_{i-1,i-1}^2 = q$ for $i = 2, ..., n$ for some $q > 0$.

Here $q < 1$, otherwise the basis $B = [\mathbf{b}_1, ..., \mathbf{b}_n]$ starts with a shortest lattice vector $\mathbf{b}_1$, $\|\mathbf{b}_1\| = \lambda_1$. The basis B satisfies **GSA** if its reduction is " locally uniform". Our worst case time bounds under GSA hold approximately in practice as all quotients $r_{i,i}/r_{i+1,i+1}$ of well reduced bases nearly coincide on the average. It is easier to work with the idealized requirement that the $r_{i,i}/r_{i-1,i-1}$ are all equal.

Theorem 1. *Given a lattice basis satisfying* **GSA** *and* $\|\mathbf{b}_1\| \le \sqrt{2e\pi}\, n^b\, \lambda_1$ *for some* $b \ge 0$*,* New Enum *runs in time* $n^{O(1)} + (O(n^{\frac{1}{2}+b} rd(\mathcal{L})\,)^{\frac{n+1}{4}}$.

Conclusions. New Enum runs in polynomial time if $rd(\mathcal{L}) \le n^{-\frac{1}{2}-\varepsilon}$ and $\varepsilon > b$. Then the vector $\mathbf{b}_1$ required in Theorem 1 satisfies

$$\|\mathbf{b}_1\| \le \sqrt{2e\pi}\, n^b\, rd(\mathcal{L})\, \gamma_n^{1/2} \det(\mathcal{L})^{\frac{1}{n}} = O(n^{\frac{1}{2}+b-\varepsilon})\det(\mathcal{L})^{\frac{1}{n}}.$$

Satisfying this bound for arbitrary lattices seems hard. But it is possible to simply extend the basis B and the lattice $\mathcal{L}(B)$ by the required nearly shortest vector $\mathbf{b}_1$ without solving SVP with approximation factor $\sqrt{2e\pi}\, n^b$. Therefore, SVP for $\mathcal{L}$ with $rd(\mathcal{L}) \le n^{-\frac{1}{2}-\varepsilon}$ is easy under GSA even if $\|\mathbf{b}_1\| > \sqrt{2e\pi}\, n^b \lambda_1$.

Lovász (1986) proves in section 1.2.21 a result that is similar to the case $rd(\mathcal{L}) \le n^{-\frac{1}{2}-\varepsilon}$, $\varepsilon > b$ of Theorem 1 where $\|\mathbf{b}_1\| = O(n^{\frac{1}{2}+b-\varepsilon})\det(\mathcal{L})^{\frac{1}{n}}$. Given such short vectors for all orthogonally projected lattices $\mathcal{L}_k = \pi_k(\mathcal{L})$ and some dual lattices $\overline{\mathcal{L}}_k^*$ for $k = 1, ..., n$ SVP for $\mathcal{L}$ can easily be solved in polynomial time.

References

[1] L. Lovász, *An Algorithmic Theory of Numbers, Graphs and Convexity*, SIAM, (1986).

[2] C.P. Schnorr and M. Euchner, *Lattce basis reduction: Improved practical algorithms and solving subset sum problems.* Mathematical Programming **66**, (1994) 181–199. //www.mi.informatik.uni-frankfurt.de.

[3] C.P. Schnorr, *Average Time Fast SVP and CVP Algorithms for Low Density Lattices.* Technical Report, University Frankfurt (2009). //www.mi.informatik.uni-frankfurt.de

Typically-correct derandomization

Ronen Shaltiel
(joint work with Jeff Kinne, Dieter van Melkebeek)

A fundamental open problem in Complexity Theory is whether BPP=P, that is can every polynomial time randomized algorithm be simulated by a polynomial time deterministic algorithm. Goldreich and Wigderson [1] considered a relaxed notion of deterministic simulation in which the deterministic algorithm may err on few inputs of every input length. We call such a simulation "typically-correct derandomization". Goldreich and Wigderson showed that this relaxed goal can be achieved under assumptions that are incomparable to those used in Hardness versus randomness tradeoffs [4, 2] to obtain BPP=P.

In [5, 3] we consider typically-correct derandomization in various algorithmic settings. For the case of BPP we show how to achieve typically correct derandomization under a weaker assumption than that used in [1]. Our assumption requires lower bounds for deterministic circuits while [1] requires lower bounds for nondeterministic circuits. While the assumption is still incomparable to that used in hardness versus randomness tradeoffs we argue that it seems weaker in the following sense: For randomized algorithms implemented by uniform poly-size constant depth circuits, plugging existing lower bounds for AC^0 into our technique give polynomial time typically-correct derandomization whereas plugging the known lower bounds in hardness versus randomness tradeoffs only yields a deterministic simulation that runs in quasi-polynomial time.

We also show how to achieve explicit typically-correct derandomization in algorithmic settings where derandomization on all inputs is impossible. Examples of such settings are communication protocols, decision trees and streaming algorithms.

References

[1] O. Goldreich and A. Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *Randomization and Approximation Techniques, 6th International Workshop, RANDOM 2002*, pages 209–223, 2002.

[2] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[3] J. Kinne, D. van Melkebeek, and R. Shaltiel. Pseudorandom generators and typically-correct derandomizatiom. In *Randomization and Approximation Techniques, 13th International Workshop, RANDOM 2009*, pages 574–587, 2009.

[4] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, Oct. 1994.

[5] R. Shaltiel. Weak derandomization of weak algorithms: explicit versions of Yao's lemma. In *Proceedings of the IEEE Conference on Computational Complexity*, 2009.

Semantic Communication

Madhu Sudan

(joint work with Brendan Juba and Oded Goldreich)

This talk reports on our efforts to formalize the notion of *meaning* of bits, especially in the context of communication. We assert that communication ought to be a means to achieving some end goal; and that achievement of the goal is the functional test of achieving *understanding* between parties. In earlier work with Brendan Juba [1], we proposed an example goal for communication and how it can be achieved even in the presence of potential misunderstanding. In the current work [2] we attempt to extend the study to all possible goals of communication. We propose a formal definition of a generic goal. We formalize the notion that in order to achieve the goal in the presence of misunderstanding, an interacting player must have the ability to *sense* progress. We show that sensing essentially is also sufficient to achieving the goal. We illustrate our model and theory by various examples

References

[1] Brendan Juba and Madhu Sudan. *Universal Semantic Communication I*, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pages 123-132, Victoria (BC), Canada, May 17-20, 2008.

[2] Oded Goldreich, Brendan Juba, and Madhu Sudan. *A Theory of Goal-Oriented Communication*, Electronic Colloquium on Computational Complexity, Technical Report TR 09-075, September 17, 2009.

Reporter: Christian Ikenmeyer